Maximum Entropy Models for Texture Synthesis

Arthur Leclaire
Institut de Mathématiques de Bordeaux

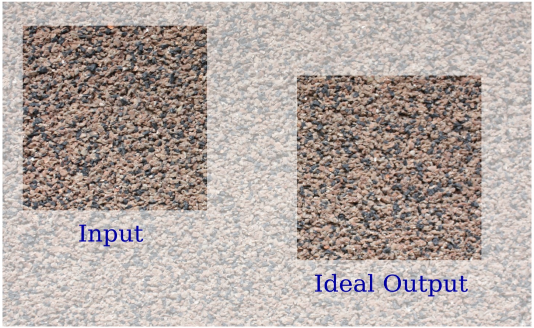
Joint work with **Valentin de Bortoli**, Agnès Desolneux, Bruno Galerne, and Alain Durmus

Workshop Statistical Models for Shapes and Images
Institut Henri Poincaré
Thursday, March 14th

By-example Texture Synthesis

Notation:

- $\Omega \subset \mathbb{Z}^2$ finite discrete rectangle.
- Image $x : \Omega \to \mathbb{R}^3$
  $x(i) = (x_R(i), x_G(i), x_B(i))$
- π probability distribution on $\mathbb{R}^d$, $d = 3|\Omega|$ (stationary random field).



Goal:

- Estimate a distribution π from an exemplar image x_0.
- Sample π.

Parametric Texture Synthesis

- Suppose that we have a family of statistical measurements (“features”)

$$f = (f_k)_{1 \leqslant k \leqslant p} : \mathbb{R}^d \longrightarrow \mathbb{R}^p$$

that captures the “perceptual aspect” of the texture.

- We want to design a random field X on Ω such that

$$\mathbb{E}[f(X)] = f(x_0) \quad \text{(macrocanonical model).}$$

or even

$$f(X) = f(x_0) \quad \text{a.s.} \quad \text{(microcanonical model).}$$

- We also need a model which is “as random as possible”
$\longrightarrow$ maximum entropy principle

Different Models for Different Statistics

- **Covariance/Fourier Spectrum**
  $\longrightarrow$ Sparse convolution, spectrum painting **[Lewis, 1984]**
  $\longrightarrow$ Spot noise, Random phase noise, Gaussian models **[Van Wijk, 1991]**, **[Galerne et al., 2011]**, **[Xia et al., 2014]**
  $\longrightarrow$ Local random phase noise **[Gilet et al., 2014]**

- **Wavelet statistics**
  $\longrightarrow$ Histograms of subbands **[Heeger & Bergen, 1995]**
  $\longrightarrow$ First-order responses to a bank of filters FRAME **[Zhu et al., 1998]**
  $\longrightarrow$ Second-order wavelet statistics **[Portilla & Simoncelli, 2000]**
  $\longrightarrow$ First-order dictionary statistics + spectrum **[Tartavel et al., 2014]**

- **Neural networks statistics**
  $\longrightarrow$ First-order neural statistics **[Lu et al., 2015]**
  $\longrightarrow$ Second-order neural statistics **[Gatys et al., 2015]**

- **Scattering statistics**
  $\longrightarrow$ First-order scattering statistics **[Zhang & Mallat, 2017]**, **[Bruna & Mallat, 2019]**

Different Models for Different Statistics

Red: Microcanonical models
Green: Macrocanonical Models

- **Covariance/Fourier Spectrum**
  $\longrightarrow$ Sparse convolution, spectrum painting **[Lewis, 1984]**
  $\longrightarrow$ Spot noise, **Random phase noise**, **Gaussian models**
  [Van Wijk, 1991], **[Galerne et al., 2011]**, **[Xia et al., 2014]**
  $\longrightarrow$ Local random phase noise **[Gilet et al., 2014]**

- **Wavelet statistics**
  $\longrightarrow$ **Histograms of subbands [Heeger & Bergen, 1995]**
  $\longrightarrow$ **First-order responses to a bank of filters FRAME [Zhu et al., 1998]**
  $\longrightarrow$ **Second-order wavelet statistics [Portilla & Simoncelli, 2000]**
  $\longrightarrow$ **First-order dictionary statistics + spectrum [Tartavel et al., 2014]**

- **Neural networks statistics**
  $\longrightarrow$ **First-order neural statistics [Lu et al., 2015]**
  $\longrightarrow$ **Second-order neural statistics [Gatys et al., 2015]**

- **Scattering statistics**
  $\longrightarrow$ **First-order scattering statistics [Zhang & Mallat, 2017]**, **[Bruna & Mallat, 2019]**

Motivation

Why studying macrocanonical models?

$\longrightarrow$ It is one principled formulation of by-example texture synthesis.

$\longrightarrow$ Link with the *modified* Julesz conjecture (1981):

> *"It seems that only the first-order statistics*
> *of these textons* [non-linear features] *have perceptual significance."*

$\longrightarrow$ Helps to better understand the chosen statistics/features.

$\longrightarrow$ Connections with nice results on MCMC and stochastic optimization.

Outline

Exponential Models

Langevin Dynamics and SOUL algorithm

Visual Results

Entropy

Let $\mathcal{P}$ be the set of probability distributions on $\mathbb{R}^d$.
Let μ be a reference probability measure on $\mathbb{R}^d$ (e.g. $\mu(dx) \propto e^{-J(x)}dx$ where $J(x) = \frac{\varepsilon}{2}\|x\|^2$)
The entropy $H : \mathscr{P} \to [-\infty, +\infty)$ (w.r.t. μ) is defined by

$$\forall \pi \in \mathcal{P}, \quad H(\pi) = \begin{cases} -\int_{\mathbb{R}^d} \log\left(\frac{\mathrm{d}\pi}{\mathrm{d}\mu}(x)\right)\frac{\mathrm{d}\pi}{\mathrm{d}\mu}(x)\mu(\mathrm{d}x) & \text{if } \frac{\mathrm{d}\pi}{\mathrm{d}\mu} \text{ exists} \\ -\infty & \text{otherwise.} \end{cases}.$$

Notice that

- $H(\pi) = -\mathrm{KL}(\pi|\mu)$
- H is strictly concave.

Macrocanonical/Microcanonical Models

Definition

Let $x_0 \in \mathbb{R}^d$ be the exemplar texture and $f : \mathbb{R}^d \to \mathbb{R}^p$ measurable.

- A **microcanonical** model associated with x_0 for the statistics f (with reference measure μ) is a probability distribution $\pi \in \mathcal{P}$ that solves

$$\max H(\pi)$$

over all $\pi \in \mathcal{P}$ such that $X \sim \pi \;\Rightarrow\; f(X) = f(x_0)$ a.s.

- A **macrocanonical** model associated with x_0 for the statistics f (with reference measure μ) is a probability distribution $\pi \in \mathcal{P}$ that solves

$$\max H(\pi)$$

over all $\pi \in \mathcal{P}$ such that $\mathbb{E}_{X\sim\pi}[f(X)] = f(x_0)$.

Maximum Entropy Principle

For $\theta \in \mathbb{R}^p$, if $e^{-\theta \cdot f} \in L^1(\mu)$, we define

$$\pi_\theta(dx) = \frac{1}{Z(\theta)} e^{-\theta \cdot f(x)} \mu(dx) = p_\theta(x)\mu(dx) \qquad \text{where} \qquad Z(\theta) = \int_{\mathbb{R}^d} e^{-\theta \cdot f(x)} \mu(dx).$$

Theorem (De Bortoli, Desolneux, Galerne, Leclaire, 2019)

Assume that

a) $\forall \theta \in \mathbb{R}^p, \quad \int_{\mathbb{R}^d} e^{\|\theta\| \|f(x)\|} \mu(dx) < \infty,$

b) $\forall \theta \in \mathbb{R}^p, \quad \mu\left(\left\{x \in \mathbb{R}^d \mid \theta \cdot f(x) < \theta \cdot f(x_0)\right\}\right) > 0.$

Then there exists $\theta_* \in \mathbb{R}^p$ such that π_{θ_*} is a macrocanonical model associated with x_0 for the statistics f. Besides, θ_* is a solution to the convex minimization problem

$$\underset{\theta \in \mathbb{R}^p}{\mathit{Argmin}} \left(\theta \cdot f(x_0) + \log Z(\theta)\right) = \underset{\theta \in \mathbb{R}^p}{\mathit{Argmin}} \log \left(\int_{\mathbb{R}^d} e^{-\theta \cdot (f(x) - f(x_0))} \mu(dx)\right) .$$

Proof: solving for θ_*

The parameter θ_* can be found by maximum-likelihood.

$$L(\theta) = \log p_\theta(x_0) = -\theta \cdot f(x_0) - \log Z(\theta).$$

Notice that

$$\frac{\partial L}{\partial \theta_k} = -f_k(x_0) - \frac{1}{Z(\theta)} \frac{\partial Z}{\partial \theta_k} = -f_k(x_0) + \frac{1}{Z(\theta)} \int_{\mathbb{R}^\Omega} f_k(x) e^{-\theta \cdot f(x)} \mu(dx) = -f_k(x_0) + \mathbb{E}_{\pi_\theta}\left[f_k(X)\right].$$

In other words,

$$\nabla L(\theta) = \mathbb{E}_{\pi_\theta}[f(X)] - f(x_0)\ .$$

Similarly,

$$\nabla^2 L(\theta) = -\mathbb{E}_{\pi_\theta}\left[(f(X) - \mathbb{E}_{\pi_\theta}[f(X)])(f(X) - \mathbb{E}_{\pi_\theta}[f(X)])^T\right] = -\text{Cov}_{\pi_\theta}(f(X))$$

$-L$ is a smooth convex function that can be minimized with gradient descent.

Model Estimation

A Monte-Carlo method is used to estimate the gradient

$$\nabla L(\theta) = \mathbb{E}_{\pi_\theta}\left[f(X)\right] - f(x_0)$$

Algorithm: Estimate θ from exemplar image x_0

- Compute observed statistics $f(x_0)$.
- Initialize $\theta \leftarrow 0$, $x \leftarrow 0$.
- For $n = 1, \dots, N$,
  - $x \leftarrow \text{Sample}(\pi_\theta)$
  - Compute estimated statistics $f(x)$.
  - Update $\theta \leftarrow \theta + \delta_n(f(x) - f(x_0))$
- Return θ.

After N iterations, we get a synthesized image x.

Exponential Models for Textures

- Stationary Gaussian model

Assume for simplicity that $x(i) \in \mathbb{R}$ for all $i \in \Omega$ (graylevel images).

$\longrightarrow$ Let us consider $f(x) = (\bar{x}, x * \tilde{x})$ with

$$\bar{x} = \frac{1}{|\Omega|} \sum_{i \in \Omega} x(i) \qquad \text{and} \qquad \forall i \in \Omega, \quad x * \tilde{x}(i) = \sum_{i' \in \Omega} x(i') x(i+i').$$

$\longrightarrow$ Then the associated macrocanonical model reads as

$$\pi_\theta(dx) = \frac{1}{Z(\theta)} \exp\left(-\theta_0 \bar{x} - \sum_{i,i' \in \Omega} \theta(i) x(i') x(i+i') - \frac{\varepsilon}{2} \|x\|^2 \right) dx.$$

Remark: If (k_j) is a bank of *linear* filters and

$$f_{j,j'}(x) = \frac{1}{|\Omega|} \sum_{i \in \Omega} k_j * x * \widetilde{k_{j'} * x}(i),$$

then the associated macrocanonical model is still a Gaussian distribution.

Exponential Models for Textures

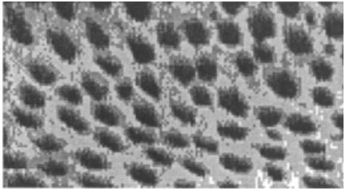
- Original FRAME Model **[Zhu, Wu, Mumford, 1998]**

  FRAME: "Filters, Random fields, And Maximum Entropy"

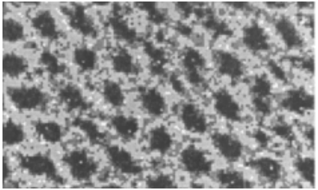
$\longrightarrow$ The features extract quantized responses of a set of linear (intensity, Laplacian of Gaussian, Gabor) and non-linear filters (modulus of Gabor):

$$f_{j,\alpha}(x) = \frac{1}{|\Omega|} \sum_{i \in \Omega} \mathbf{1}_{B_j^\alpha}(F_j * x(i))$$

where F_j is a filter and B_j^α are histogram bins.



Original



Synthesis

Exponential Models for Textures

- Original FRAME Model **[Zhu, Wu, Mumford, 1998]**

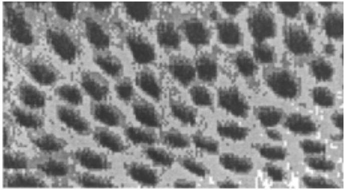
  FRAME: “Filters, Random fields, And Maximum Entropy”

$\longrightarrow$ Here, μ is the uniform distribution on $\{0, \ldots, 7\}^{\Omega}$.

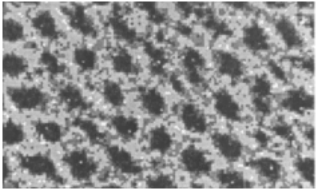
$\longrightarrow$ FRAME model is limited to quantized images (8 greylevels)

$\longrightarrow$ Synthesizing the FRAME model relies on Gibbs sampling.

$\longrightarrow$ A greedy procedure selects a small subset of filters ($\approx$ 6)



Original



Synthesis

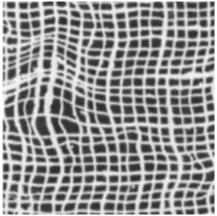
Exponential Models for Textures

- DeepFRAME: Model using CNN **[Lu, Zhu, Wu, 2016]**

$\longrightarrow$ The features extract responses to a given layer of a **pre-learned** convolutional neural network (CNN)

$$f_k(x) = \frac{1}{|\Omega|} \sum_{i \in \Omega} \mathcal{F}_k(x)(i)$$

where $(\mathcal{F}_k(x))_{1 \leqslant k \leqslant p}$ is the response at one particular layer of a CNN.



Original

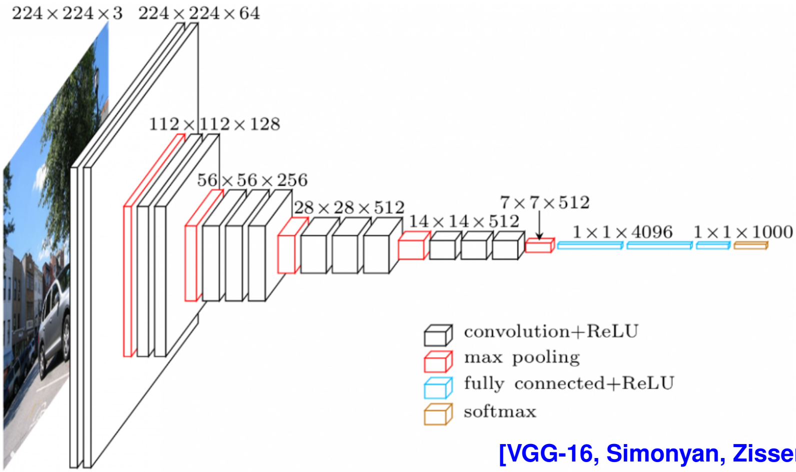
Synthesis

Original

Synthesis

Statistics used in DeepFrame

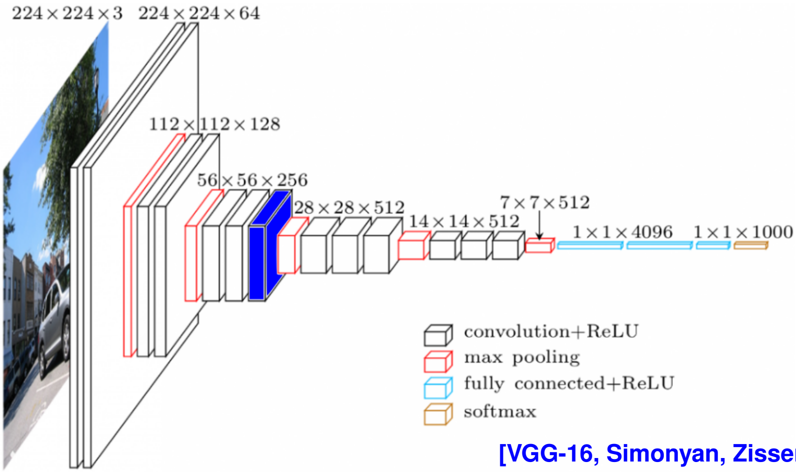
They use the CNN designed by the Visual Geometry Group (VGG) in Oxford.



[VGG-16, Simonyan, Zisserman, 2015]

Statistics used in DeepFrame

They use the CNN designed by the Visual Geometry Group (VGG) in Oxford.



[VGG-16, Simonyan, Zisserman, 2015]

Neural Network Features

Let us consider

$$\forall x \in \mathbb{R}^d, \quad \mathcal{F}(x) = (\mathcal{F}_1(x), \ldots, \mathcal{F}_p(x)) \in \prod_{k=1}^{p} \mathbb{R}^{d_k}$$

where $\mathcal{F}_k(x)$ is one response to a layer of a CNN with a non-linear unit $\varphi \in \mathscr{C}^1(\mathbb{R})$.
More precisely,

$$\mathcal{F}_j(x) = (\varphi \circ A_j \circ \varphi \circ A_{j-1} \circ \ldots \circ \varphi \circ A_1)(x)$$

where $A_j : \mathbb{R}^{n_j} \to \mathbb{R}^{n_{j+1}}$ is linear, and φ is applied on each component.
Example: for a convolutional neural network,

$$A_j(y) = k_j * y$$

where $k_j : \Omega_j \to \mathbb{R}^{n_{j+1} \times n_j}$ is a matrix convolution kernel.

Neural Network Features

We define

$$f(x) = \left(\sum_{i=1}^{d_1} \mathcal{F}_1(x)(i), \dots, \sum_{i=1}^{d_p} \mathcal{F}_p(x)(i) \right) .$$

The corresponding macrocanonical model is stationary (because of the spatial summation).

Proposition (De Bortoli, Desolneux, Galerne, Leclaire, 2019)

Let $x_0 \in \mathbb{R}^d$ and assume that $df(x_0)$ has rank $\min(d, p) = p$.
Assume that $\varphi \in \mathscr{C}^1(\mathbb{R})$ and that

$$\exists c > 0, \; \forall x \in \mathbb{R}, \quad |\varphi(x)| \leqslant c(1 + |x|).$$

Then the maximum entropy principle holds with $J(x) = \frac{\varepsilon}{2}\|x\|^2$ for any $\varepsilon > 0$.

Outline

Exponential Models

Langevin Dynamics and SOUL algorithm

Visual Results

How to sample π_θ ?

Let

$$V(x,\theta) = \theta \cdot (f(x) - f(x_0)) + J(x) \qquad \text{so that} \qquad \pi_\theta(x) \propto e^{-V(x,\theta)} dx.$$

We consider the **Langevin dynamics**

$$X_{n+1} = X_n - \gamma_{n+1} \nabla_x V(X_n, \theta) + \sqrt{2\gamma_{n+1}} Z_n$$

where

- (Z_n) is a collection of independent normalized Gaussian white noises
- $\gamma_n \geqslant 0$ is a sequence of step sizes

Equivalently, (X_n) is a inhomogeneous Markov chain with kernel

$$R_{\gamma_n}(x, \cdot) = \mathcal{N}(x - \gamma_n \nabla_x V(x,\theta), 2\gamma_n).$$

Theorem (Durmus, Moulines, 2016)

Under some hypotheses on V, and if $\sum \gamma_n = +\infty$ and $\sum \gamma_n^2 < \infty$, we have

$$X_n \xrightarrow[n\to\infty]{(d)} \pi_\theta$$

Sampling a GMM with Langevin Dynamics

A brief video interlude.

Combined Dynamics

We can now approximate $\nabla L(\theta)$ with a Langevin-based MCMC method.

$\longrightarrow$ Stochastic Optimization with Unadjusted Langevin (SOUL)

SOUL algorithm

Initialization: $X_0^0 \in \mathbb{R}^d$.

$$X_{k+1}^n = X_k^n - \gamma_{n+1} \nabla_x V(X_k^n, \theta_n) + \sqrt{2\gamma_{n+1}} Z_{k+1}^n$$
$$\text{for } k = 0, \dots, m_n - 1 \text{ , with } Z_{k+1}^n \sim \mathcal{N}(0, I)$$
$$\theta_{n+1} = \text{Proj}_\Theta \left(\theta_n - \frac{\delta_{n+1}}{m_n} \sum_{k=1}^{m_n} \nabla_\theta V(X_k^n, \theta_n) \right)$$
$$X_0^{n+1} = X_{m_n}^n$$

where Θ is a closed convex set of $\mathbb{R}^d$.

Convergence of SOUL algorithm

Notice that $-L$ is convex, $\mathscr{C}^1$ with Lipschitz gradient on Θ compact.

Theorem (De Bortoli, Durmus, Pereyra, Fernandez Vidal, 2019)

Assume that

1. Θ *is a convex compact set of* $\mathbb{R}^p$.
2. $J, f_1, \dots, f_p$ *are differentiable on* $\mathbb{R}^d$ *with Lipschitz gradients.*
3. *There exist* $\eta, c, M > 0$ *such that* $\forall \theta \in \Theta,\ \forall x \in \mathbb{R}^d, \quad \langle \nabla_x V(x, \theta), x \rangle \geqslant \eta \|x\|^2 \mathbf{1}_{|x|>M} - c$.
4. $(\delta_n), (\gamma_n)$ *are non-increasing positive with* δ_0, γ_0 *sufficiently small and*

$$\sum \delta_n = +\infty, \quad \sum \delta_{n+1} \sqrt{\gamma_n} < \infty, \quad \sum \frac{\delta_{n+1}}{m_n \gamma_n} < \infty.$$

Then $\theta_n \longrightarrow \theta_* \in \mathrm{Argmin}(-L)$ *almost surely and in* L^1.

NB: f may be non-convex (e.g. with differentiable neural networks).

Link with Microcanonical Model

For $V(x,\theta) = \theta \cdot (f(x) - f(x_0)) + J(x)$ and $J(x) = \frac{\varepsilon}{2}\|x\|^2$, the update reads

$$X_{k+1}^n = X_k^n - \gamma_{n+1} \sum_{j=1}^{p} \theta_{n,j} \nabla f_j(X_k^n) - \gamma_{n+1} \nabla J(X_k^n) + \sqrt{2\gamma_n} Z_{k+1}^n$$

$$= X_k^n - \gamma_{n+1} df(X_k^n)^T . \theta_n - \gamma_{n+1} \varepsilon X_k^n + \sqrt{2\gamma_n} Z_{k+1}^n .$$

$$\theta_{n+1} = \theta_n - \frac{\delta_{n+1}}{m_n} \sum_{k=1}^{m_n} (f(X_k^n) - f(x_0))$$

Taking $m_n = 1, \delta_n = 1, \gamma_{n+1} = \frac{1}{n}$, $\varepsilon = 0$, $\theta_0 = 0$, and removing the noise we get

$$X_{n+1} = X_n - df(X_n)^T \left(\frac{1}{n} \sum_{k=0}^{n-1} f(X_k) - f(x_0) \right) .$$

We get back a momentum-like gradient method to minimize $\Phi(x) = \|f(x) - f(x_0)\|_2^2$.

Outline

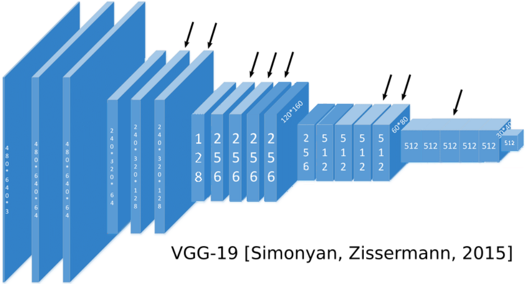
Exponential Models

Langevin Dynamics and SOUL algorithm

Visual Results

Experimental setup

- $f(x)$: spatially averaged reponses to *differentiable* VGG-19 at layers $3, 4, 5, 6, 7, 11, 12, 14$.
- Initialization: Gaussian random field with correct second-order statistics.
- $\delta_n = \mathcal{O}(\frac{1}{n})$, $\gamma_n = \mathcal{O}(\frac{1}{n})$, $m_n = 1$
- $\varepsilon = 0.1$ i.e. $\mu(dx) \propto e^{-0.05\|x\|^2}$
- $\Theta = \mathbb{R}^p$ (no projection)
- The color distribution is reimposed afterwards.



VGG-19 [Simonyan, Zissermann, 2015]

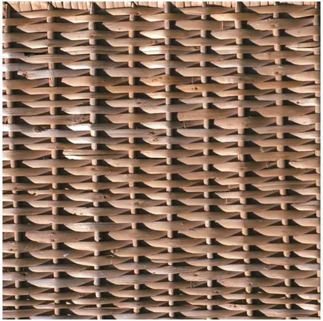
Synthesis Results

Original (256 $\times$ 256)

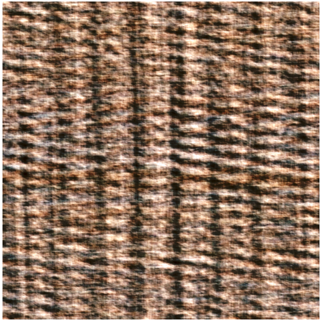
Initialization (Gaussian)

After 5000 iterations

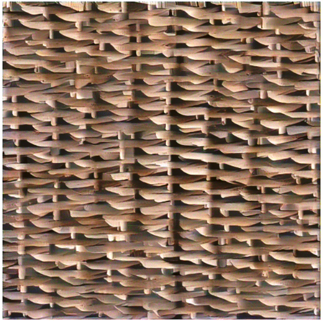
Synthesis Results



Original (512 $\times$ 512)

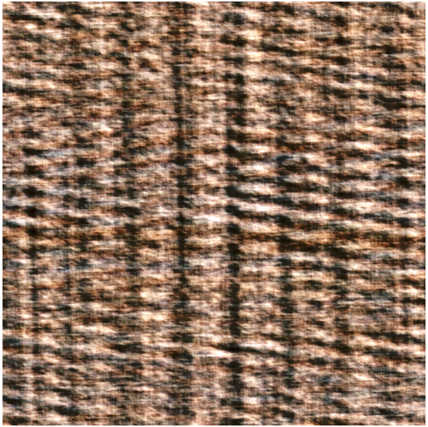


Initialization (Gaussian)



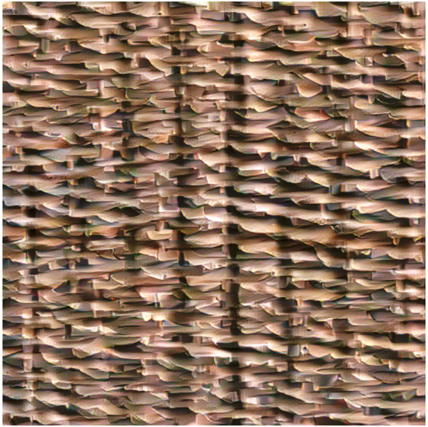
After 5000 iterations

Empirical Convergence



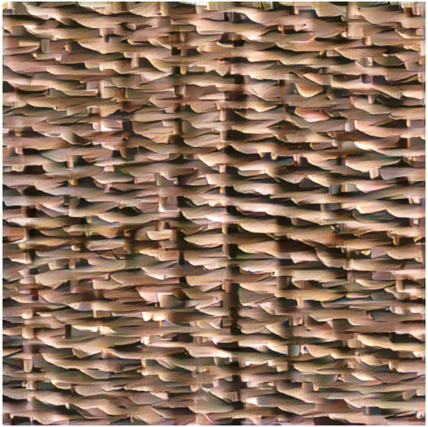
Iteration 0

Empirical Convergence



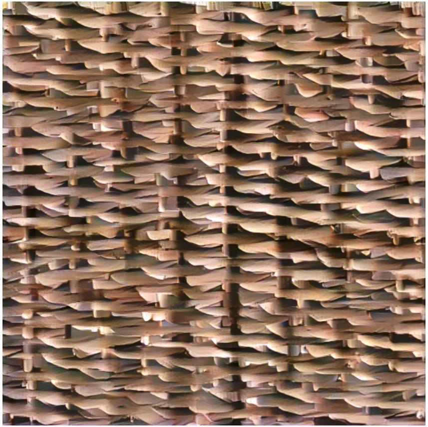
Iteration 100

Empirical Convergence



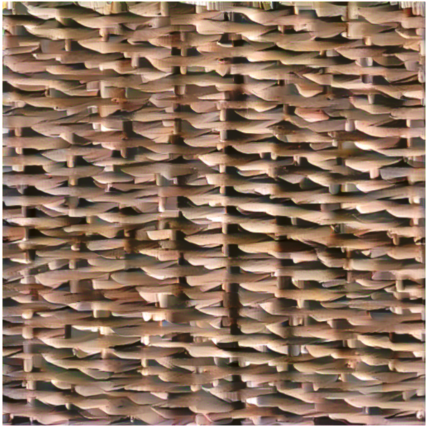
Iteration 200

Empirical Convergence



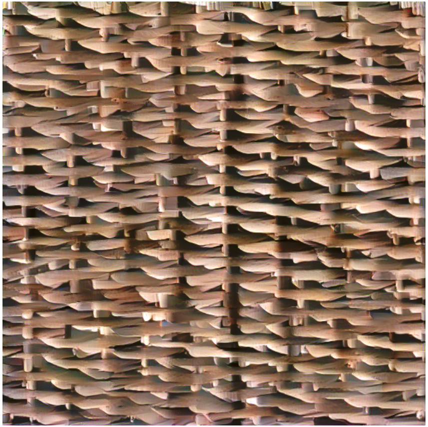
Iteration 300

Empirical Convergence



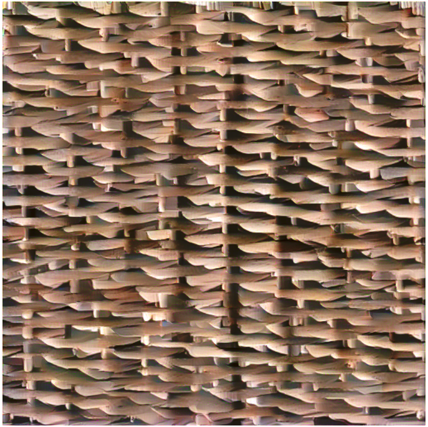
Iteration 400

Empirical Convergence



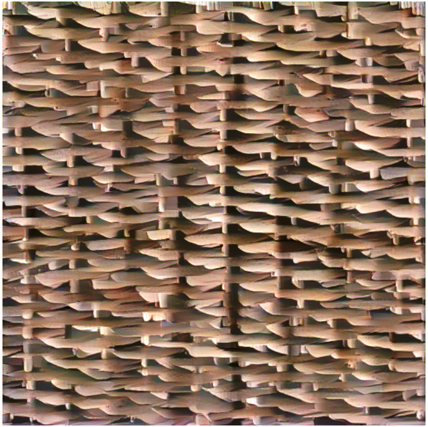
Iteration 500

Empirical Convergence



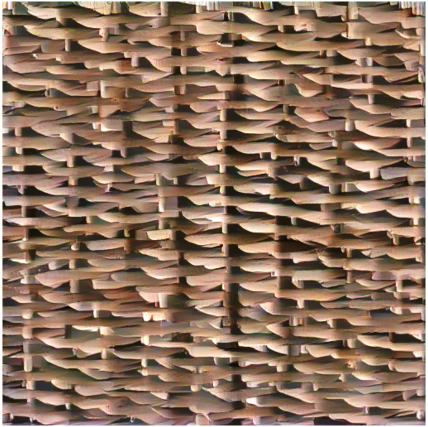
Iteration 600

Empirical Convergence



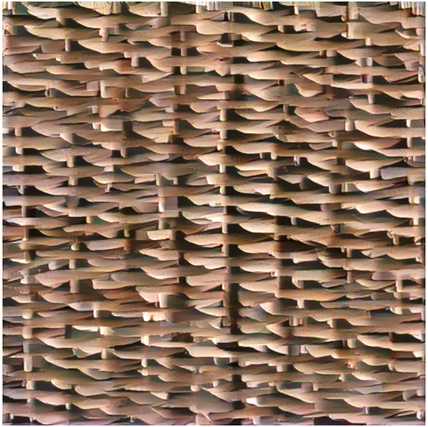
Iteration 700

Empirical Convergence



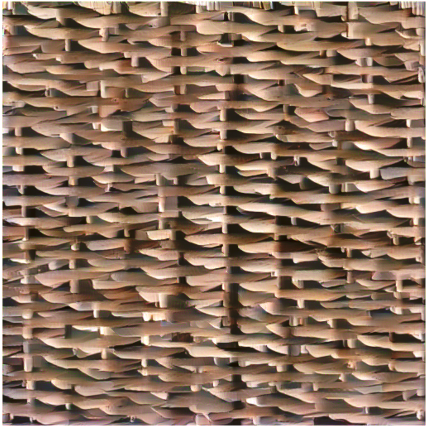
Iteration 800

Empirical Convergence



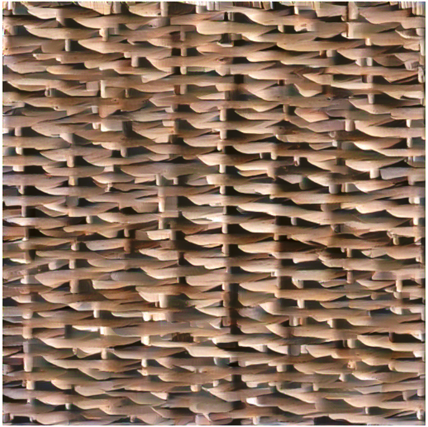
Iteration 900

Empirical Convergence



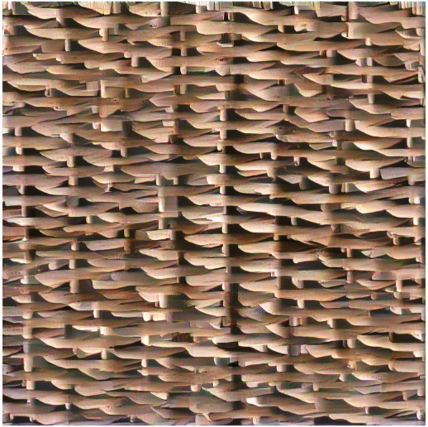
Iteration 1000

Empirical Convergence



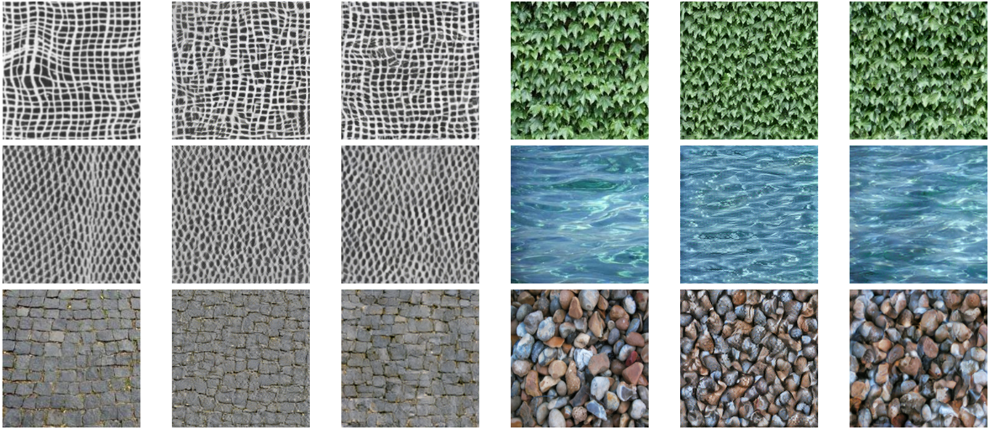
Iteration 2000

Empirical Convergence



Iteration 4000

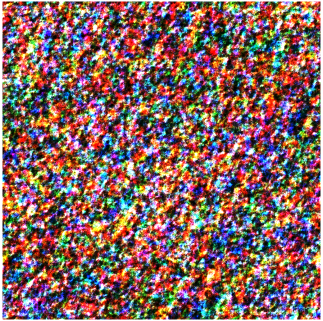
Comparison with DeepFrame



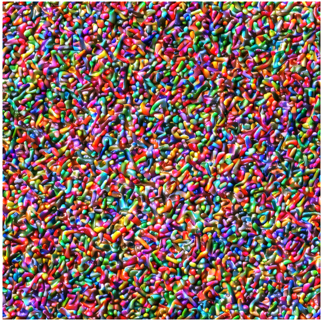
Original | DeepFrame [Lu et al.] | Our result | Original | DeepFrame [Lu et al.] | Our result

Synthesis Results

Original (512 $\times$ 512)



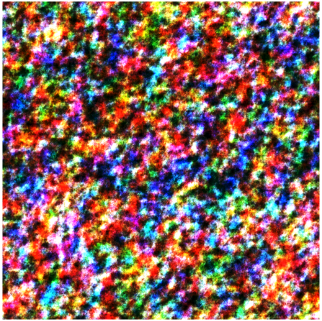
Initialization (Gaussian)



After 5000 iterations

Synthesis Results

Original (512 $\times$ 512)



Initialization (Gaussian)

After 5000 iterations

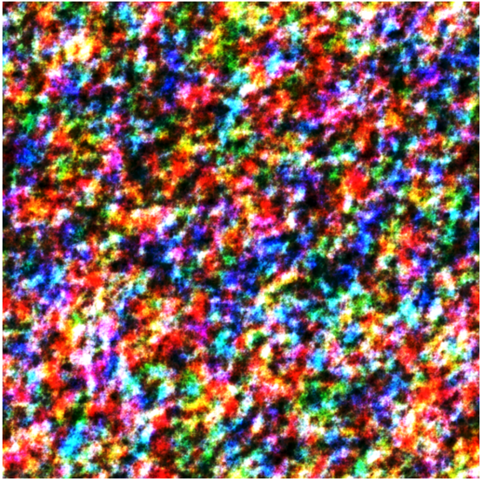
Synthesis Results

Original (512 × 512)

Synthesis Results

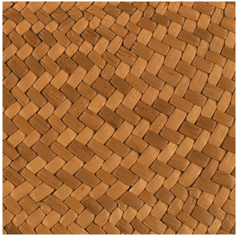
After 5000 iterations

Synthesis Results

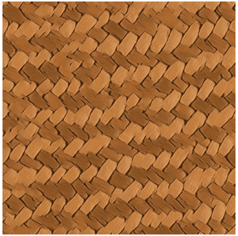


Initialization (Gaussian)

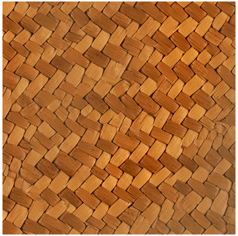
Comparison



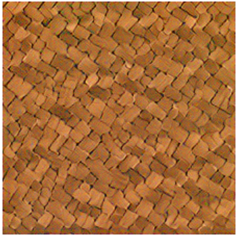
Original (512 $\times$ 512)



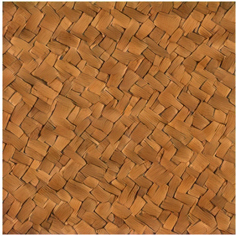
[Galerne et al.]



[Gatys et al.]



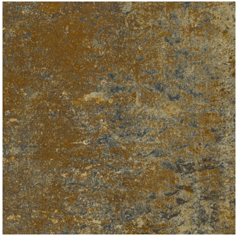
DeepFrame [Lu et al.]
(resolution /2)

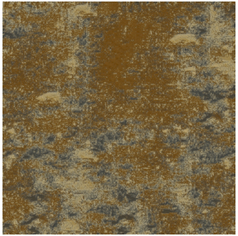


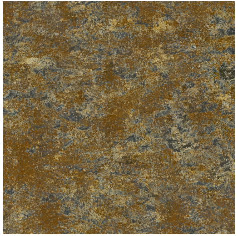
Our Result

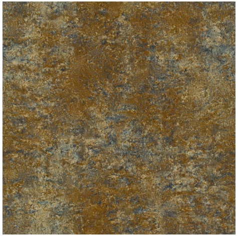
GAN [Jetchev et al.]

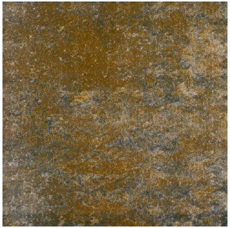
Comparison


Original


[Galerne et al.]


[Gatys et al.]


[Portilla & Simoncelli]


Our result


SGAN **[Jetchev et al.]**

Comparison

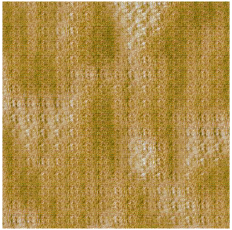
Original

[Galerne et al.]

[Gatys et al.]

[Portilla & Simoncelli]

Our result



SGAN [Jetchev et al.]

Conclusion - Perspectives

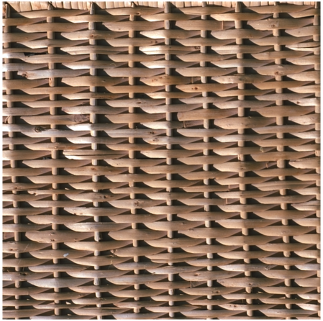
- Langevin sampling allows to design a generalization of FRAME
  $\longrightarrow$ with a continuous state-space
  $\longrightarrow$ only needs to differentiate the features (Auto-Diff)
- Provably convergent sampling and estimation algorithms (under hypotheses).
- Able to synthesize textures using VGG features (although mixing time is large).
- A model with only 2560 parameters.

PERSPECTIVES

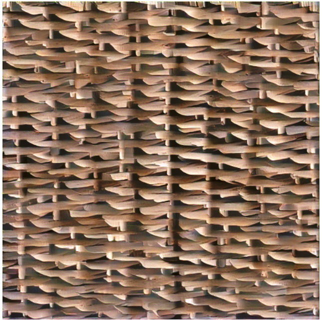
- Can we handle non-differentiable features?
- Include other statistics (wavelets, scattering, etc).
- Microcanonical and macrocanonical models asymptotically coincide when $\Omega \to \mathbb{Z}^2$?

THANK YOU FOR YOUR ATTENTION!

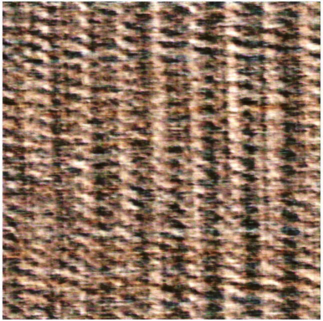
Randomly Weighted Network



Original (512 $\times$ 512)

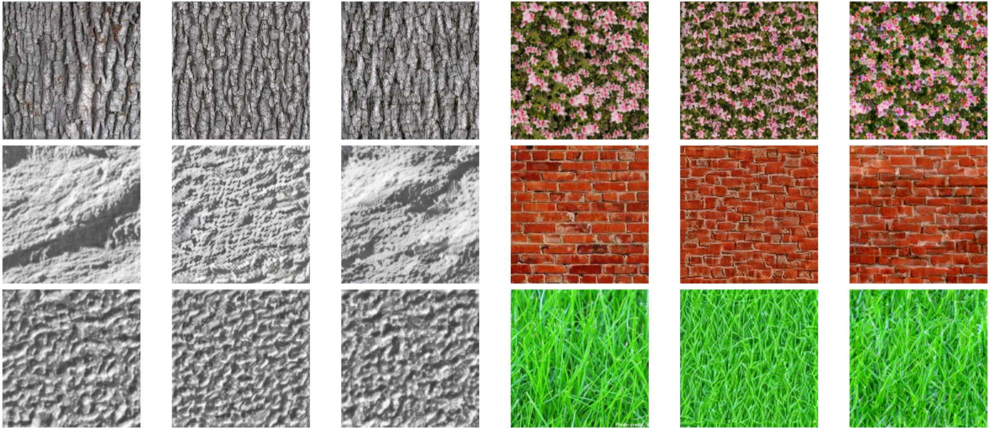


With VGG19



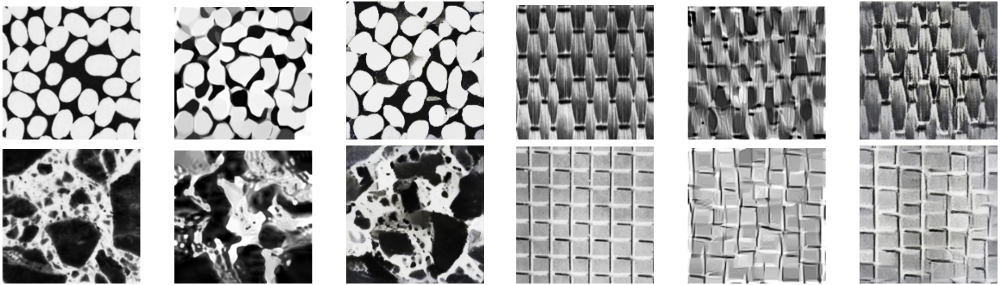
With Random weights

Comparison with DeepFrame



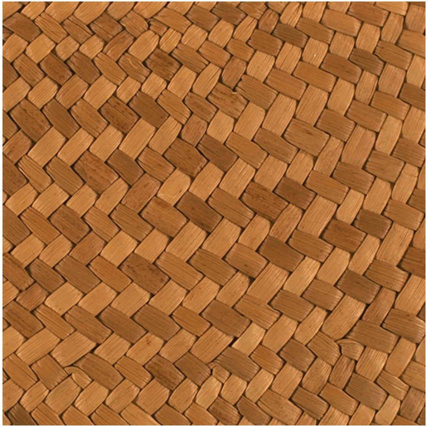
Original | DeepFrame [Lu et al.] | Our result | Original | DeepFrame [Lu et al.] | Our result

Comparison with Scattering



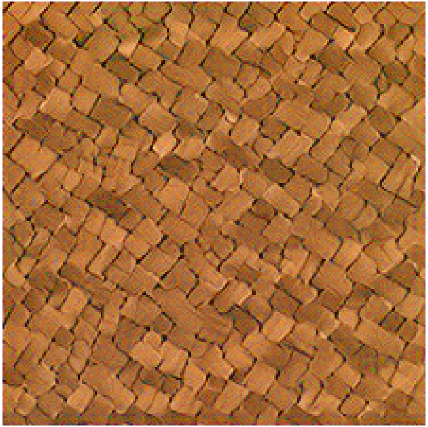
Original | Scattering [Bruna & Mallat] | Our result | Original | Scattering [Bruna & Mallat] | Our result

Comparison



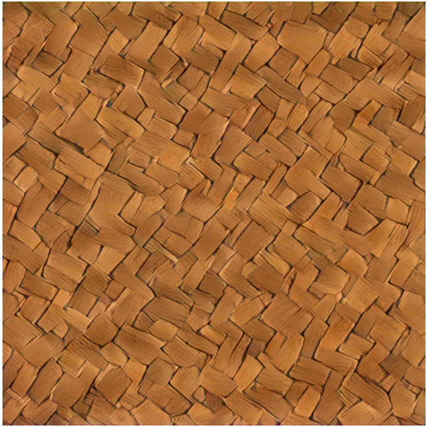
Original

Comparison



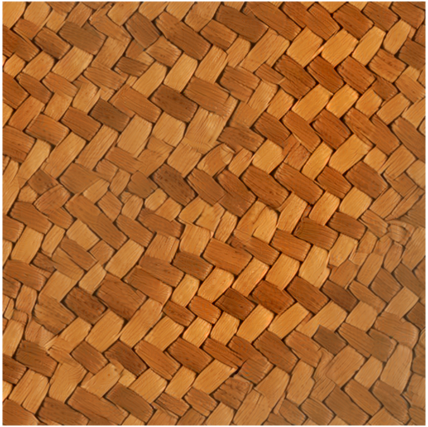
DeepFrame **[Lu et al.]** (warning : reduced resolution)

Comparison



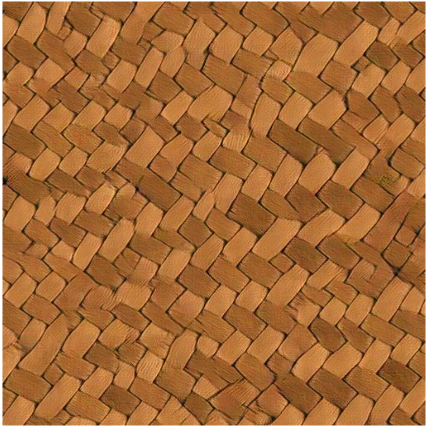
Our Result

Comparison



[Gatys et al.]

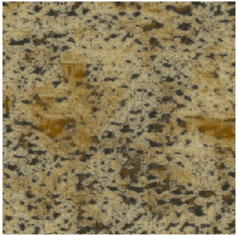
Comparison



GAN **[Jetchev et al.]**

Comparison

Original



[Galerne et al.]

[Gatys et al.]

[Portilla & Simoncelli]

Our result

SGAN [Jetchev et al.]

Comparison

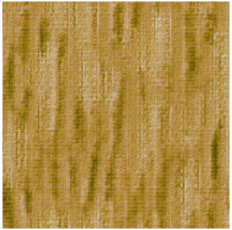

Original


[Galerne et al.]


[Gatys et al.]


[Portilla & Simoncelli]


Our result


SGAN [Jetchev et al.]