

COMPACT REPRESENTATIONS OF STATIONARY DYNAMIC TEXTURES

Gui-Song Xia, Sira Ferradans, Gabriel Peyré

Jean-François Aujol

CEREMADE, Univ. Paris-Dauphine

Univ. Bordeaux, IMB, UMR 5251

ABSTRACT

This paper addresses the problem of modeling stationary color dynamic textures with Gaussian processes. We detail two particular classes of such processes that are parameterized by a small number of compactly supported linear filters, so-called dynamical textons (*dynTextons*). The first class extends previous works on the spot noise texture model to the dynamical setting. It directly estimates the *dynTexton* to fit a translation-invariant covariance from the exemplar. The second class is a specialization of the auto-regressive (AR) dynamic texture method to the setting of space and time stationary textures. This allows one to parameterize the process covariance using only a few linear filters. Numerical experiments on a database of stationary textures shows that the methods, despite their extreme simplicity, provide state of the art results to synthesize space stationary dynamical texture.

Index Terms— Dynamic texture, texture synthesis, autoregressive process, spot noise.

1. INTRODUCTION

The modeling of dynamic textures, referring to image sequences that exhibit spatial and temporal regularities [1, 2], attracts much attention in image analysis [1, 2, 3, 4, 5]. This paper focuses on simple texture models, namely stationary Gaussian processes, to compute compact texture representation for synthesis.

Modeling stationary dynamic textures By “stationary dynamic textures” (SDTs), we refer to dynamic textures with stationarity in space and in time, as those studied in [6]. Early attempts to model SDTs include the spatio-temporal autoregressive (STAR) model [1], which creates locally space-time models for individual pixels relying on 3D causal neighbors. But there is no clear reason that the spatial neighbors should be causal. Bar-Joseph *et al.* [3] proposed a 3D wavelet transform to construct multi-resolution trees for synthesizing dynamic textures. However, the manipulation of such wavelet coefficients is not trivial. Following the idea of Efros and Leung [7], patch-based methods are adapted to synthesize stationary dynamic textures, see [8, 9]. But real modeling of

the texture content of videos is required to lead better understanding of dynamic textures. The most recent approach that is capable of modeling the spatio-temporal texture content of SDTs is the one proposed by Doretto *et al.* [6], by using dynamic multiscale autoregressive (AR) models and a linear dynamic system (LDS) [2]. The stationary nature of SDTs is not well investigated by this method and this results in *unnecessarily* large models.

Dynamic textures with Gaussian processes. Exploiting the spatial correlations of pixels indeed enables to build more compact representations of textures, see [1, 10, 11]. Recently, a compact Gaussian texton has been proposed for stationary 2D textures [12]. The investigation of such Gaussian textons for stationary dynamic textures is thus interesting but has never been addressed. The LDS models [6] often need a dimensional reduction step, e.g. PCA, to establish the AR processes. However, understanding the covariance operator of stationarity AR processes as a convolution, we are able to avoid the dimensional reduction step, reduce the model size and effectively speed up the computation.

Contributions This paper studies two kinds of Gaussian models for SDTs: the spot noise model and the stationary AR model. We investigate the parameter estimation problems and finally propose compact dynamic texture representations, which lead to fast analysis and synthesis of SDTs.

2. STATIONARY DYNAMIC TEXTURES

This paper concentrates on the modeling of dynamic textures, presented by deterministic videos $f \in \mathbb{R}^{N \times T \times d}$ where N is the number of spatial pixels in each frame, T is the number of frames, and d is the number of channels ($d = 1$ for gray-scale videos and $d = 3$ for color ones). We use the notation $f = (f_i^t(x))_{i,t,x}$, with $i = 1, \dots, d$ indexing the channel, $t = 1, \dots, T$ indexing time and $x = (x_1, x_2) \in \{0, \dots, n_1 - 1\} \times \{0, \dots, n_2 - 1\}$ indexing the 2-D pixel location where $N = n_1 \times n_2$. We also use $f^t = (f_i^t(x))_{i,x} \in \mathbb{R}^{N \times d}$ to denote a single frame.

We model a dynamic texture as a Gaussian random vector X in space and time, which is a mapping $X : \Omega \rightarrow \mathbb{R}^{N \times \mathbb{Z} \times d}$ with Ω as some probability space (observe we index time domain by \mathbb{Z} and do not introduce an artificial initial time.)

This work has been supported by the European Research Council (ERC project SIGMA-Vision) and the French National Research Agency (project NatImages).

For simplicity, we assume periodic boundary conditions in space in the following exposition, and we will tackle non-periodic input videos by computing their periodic components in Section 5. Stationarity of a dynamic texture implies that $X = (X^t)_{t \in \mathbb{Z}}$ has the same distribution as $(X^{t+\tau}(\cdot + d))$ for any shift $(\tau, d) \in \mathbb{Z} \times \mathbb{Z}^2$.

The 2D Fourier transform of a gray-scale image $g \in \mathbb{R}^N$ is defined as

$$\forall \omega = (\omega_1, \omega_2), \quad \hat{g}(\omega) = \sum_{x=(x_1, x_2)} g(x) e^{2i\pi \left(\frac{\omega_1 x_1}{n_1} + \frac{\omega_2 x_2}{n_2} \right)}.$$

The Fourier transform of a color image is obtained by concatenating the Fourier transforms of all the d channels of the image. This formula is also extended to videos $f \in \mathbb{R}^{N \times T \times d}$ to define $\hat{f}(\omega, \xi) \in \mathbb{R}^d$ where $\xi \in \{0, \dots, T-1\}$ is the time frequency.

3. SPOT NOISE GAUSSIAN DYNTEXTONS

3.1. Spot noise (SN) models

Given some deterministic input exemplar $f \in \mathbb{R}^{N \times T \times d}$, it makes sense to learn from f the parameters of a Gaussian model using the maximum likelihood estimator (MLE). It can be shown to be equal to the SN model introduced by [13]. A random field $X = (X_1, \dots, X_d)$ distributed according to the Gaussian SN $S \in \mathbb{R}^{N \times \mathbb{Z} \times d}$ associated to $f = (f_1, \dots, f_d) \in \mathbb{R}^{N \times T \times d}$ reads

$$\forall j = 1, \dots, d, \quad X_j = m_j + S_j \star W \quad (1)$$

where m_j is the space-time average, \star is the space-time convolution (infinite in time and periodic in space) defined for $h, g \in \mathbb{R}^{N \times \mathbb{Z} \times d}$ as, for all $j = 1, \dots, d$,

$$(h \star g)_j^t(x) = \sum_{t=-\infty}^{\infty} \sum_y h_j^{t-\tau}(x-y) g_j^\tau(y) \quad (2)$$

and W are i.i.d. Gaussian noises, *i.e.* $W^t(x) \sim \mathcal{N}(0, \frac{1}{\sqrt{NT}})$. To compute the space-time convolution with $f \in \mathbb{R}^{N \times T \times d}$, one should extend f to zero when $t \leq 0$ and $t > T$. The covariance of X can be estimated by using the empirical autocorrelation of f ,

$$\forall i, j = 1, \dots, d, \quad S_i \star \tilde{S}_j = f_i \star \tilde{f}_j \quad (3)$$

where $\tilde{S}_j^t(x) = \tilde{S}_j^{t-t}(-x)$.

3.2. Learning SN-dynTextons

In numerical applications, the input video is not periodic in space and only has a finite number of time frames. As detailed in Section 5, a simple preprocessing replaces this input with a space and time periodic video $f \in \mathbb{R}^{N \times T \times d}$. For the learning stage, we thus replace the SN model Equation (3)

by $S_i \star \tilde{S}_j = f_i \star \tilde{f}_j$ where $\bar{\cdot}$ is the space-time finite periodic convolution, replacing the integration of t over $(-\infty, +\infty)$ by $\{0, \dots, T-1\}$ in Equation (2). This is equivalent to imposing, for all $j \in \{1, \dots, d\}$, $\xi \in \{0, \dots, T-1\}$ and $\omega \in \{0, \dots, n_1-1\} \times \{0, \dots, n_2-1\}$,

$$\hat{S}_j(\omega, \xi) = \hat{f}_j(\omega, \xi) \hat{u}(\omega, \xi), \quad \text{s.t.} \quad |\hat{u}(\omega, \xi)| = 1. \quad (4)$$

Following [12], we restrict our attention to a small family $S = S^{[\delta]}$ of textons parameterized by $\delta \in \mathbb{R}^d$. This texton is defined by using $\hat{u} = \hat{u}_\delta$ in (4), where

$$\hat{u}_\delta(\omega, \xi) = \frac{\hat{c}_\delta(\omega, \xi)}{|\hat{c}_\delta(\omega, \xi)|} \quad \text{where} \quad \hat{c}_\delta(\omega, \xi) = \sum_{j=1}^d \hat{f}_j(\omega, \xi) \star \delta_j.$$

We define the SN-dynTextons $S = S^{[\delta]}$ where δ minimizes a quadratic spatial compactness criterion $E(\hat{S}^{[\delta]})$. This criterion equivalently measures the smoothness of \hat{S} . A classical choice, already used in [12], is a Sobolev norm $E(S) = \sum_{\xi, \omega} \|\nabla \hat{S}(\omega, \xi)\|^2$ where ∇ is a finite difference approximation of the color gradient operator. In practice, we minimize E by gradient descent using several random initialization. An example of such SN-dynTextons is displayed in Figure 1.

The obtained SN-dynTexton $S \in \mathbb{R}^{N \times T \times d}$ is a periodic 3-D filter. Numerical experiments show in Section 6 that in practice it has a fast temporal decay. It can thus be extended by zero padding when $t \leq 0$ and $t > T$. This produces an infinite time filter $S \in \mathbb{R}^{N \times \mathbb{Z} \times d}$ that can be used to define the model in (1).

4. AR GAUSSIAN DYNTEXTONS

4.1. Stationary AR processes

To reduce the number of parameters required to setup a stationary texture model, we follow [2] and assume X is an autoregressive Gaussian random field of order p (AR(p)). Since we assume space and time stationarity, such a field must have a finite variance and obey the following iterative relation, for each $i = 1, \dots, d$,

$$X_i^t = m_i + \sum_{\tau=1}^p \sum_{j=1}^d a_{i,j}^\tau \star X_j^{t-\tau} + \sum_{j=1}^d b_{i,j} \star W_j^{t-\tau} \quad (5)$$

where m_i is the average, \star is the 2-D spatial convolution in \mathbb{R}^N , and W_j^t are i.i.d Gaussian noises as $W_j^t(x) \sim \mathcal{N}(0, \frac{1}{\sqrt{N}})$, and $(a, b) = (a_{i,j}^\tau, b_{i,j})_{i,j,\tau}$ is a family of 2-D spatial filters.

4.2. Learning dynamic texture parameters

The parameters (a, b) of the model are learned from a single input video f by solving the Yule-Walker equations, adapted to the space-time stationary setting.

Learning the a parameters. This approach can be derived by first computing a least square fit of a , assuming that $b = 0$. Using the space stationarity, this can be re-written over the Fourier domain as solving, independently for each ω

$$\min_{\hat{a}(\omega)} \sum_{t=p}^T \sum_{i=1}^d |R_i^t(\omega)|^2, \quad \text{where}$$

$$\hat{R}_i^t(\omega) = \hat{X}_i^t(\omega) - \sum_{\tau=1}^p \sum_{j=1}^d \hat{a}_{i,j}^\tau(\omega) \hat{X}_j^{t-\tau}(\omega) \quad (6)$$

Dropping the dependency on ω to ease readability, introducing the block of frames $\hat{X}^{[\tau]} = (\hat{X}(\omega)^{t-\tau})_{t=p}^T \in \mathbb{C}^{(T-p) \times d}$ and $\hat{a}^\tau = (\hat{a}_{i,j}^\tau(\omega))_{i,j} \in \mathbb{C}^{d \times d}$, this minimization reads

$$\min_{\hat{a}(\omega)} \|\hat{X}^{[0]} - \sum_{\tau=1}^p \hat{X}^{[\tau]} \hat{a}^\tau\|^2$$

where $\|\cdot\|$ is the Frobenius norm of matrices in $\mathbb{C}^{(T-p) \times d}$.

The solution of this minimization requires to solve the following linear system of the variables $\hat{a}^{[\tau]}$

$$\forall \delta = 1, \dots, p, \quad \sum_{\tau=1}^p C_{\delta,\tau} \hat{a}^\tau = C_{\delta,0}$$

where $C_{\delta,\tau} = (\hat{X}^{[\delta]})^* (\hat{X}^{[\tau]}) \in \mathbb{C}^{d \times d}$. This can be achieved using a fast conjugate gradient solver (with hermitian system).

Learning the b parameters. Once a is learned, one computes the empirical residual \hat{R} as defined in (6), which is supposed to be, for each t , a realization of $\mathcal{N}(0, \hat{b}(\omega) \hat{b}(\omega)^*)$. One computes $\hat{b}(\omega) = (\hat{b}_{i,j}(\omega))_{i,j} \in \mathbb{C}^{d \times d}$ as any factorization $\hat{\Sigma}(\omega) = \hat{b}(\omega) \hat{b}(\omega)^*$ of the empirical covariance

$$\forall i, j = 1, \dots, d, \quad \hat{\Sigma}(\omega)_{i,j} = \frac{1}{T} \sum_{t=1}^T R_i^t(\omega) R_j^t(\omega)^*.$$

One can for instance use the Cholesky factorization, which ensures that $b_{i,j} = 0$ for $i > j$, thus reducing to $d(d+1)/2$ the number of filters to be stored.

Observe that (a, b) have compact support, meaning decay very fast, in space for stationary dynamic textures. Figure 1 shows the learned (a, b) for a dynamic texture video.

5. STATIONARY DYNAMIC TEXTURE SYNTHESIS

Notice that the learned SN-dynTextons has compact support in space and time and the filters associated with AR-dynTextons decay very fast in space. Thus, we can threshold both dynTextons for dynamic texture synthesis. The compact dynTextons will almost enable causal and online synthesis.

Synthesis with SN-dyntextons. Once the compact SN model S has been learned, the synthesis of a texture $g \in \mathbb{R}^{N \times \mathbb{Z} \times d}$ is obtained by using a realization of the Gaussian process, for instance, relying on the convolution formula (1).

Synthesis with AR-dyntextons. Given an exemplar video f , the synthesis of a texture $g \in \mathbb{R}^{N \times \mathbb{Z} \times d}$ is obtained by driving the AR process in Equation (5) with white Gaussian noise and learned filters (a, b) . The initial frames can be set to images filled by zeros, and the process evolves to a stationary state quickly. Observe that this model enables online synthesis thanks to the causal filters.

Non-periodic boundary conditions. For a non-periodic input video f , in order to meet the periodic boundary conditions, we compute its periodic component by relying on the FFT-based Poisson solver in [14]. More precisely, for SN-dyntextons, we compute the 3D space-time periodic components and for AR-dyntextons, we need to compute the 2D periodic component for each frame.

Texture synthesis with arbitrary size. Our framework enables to synthesize dynamic texture with arbitrary size in space and in time. The only change is to derive the dynamic processes with a Gaussian noise of the expected size.

6. NUMERICAL RESULTS

In order to test the proposed algorithms, we compiled a dataset of stationary dynamic textures¹ containing 27 different color dynamic textures. It includes dynamic sequences of *boiling water, clouds, fire, fog, fountain, waterfall, snow, ocean waves, ponds, and steam*. Figure 1 presents the results of analysis and synthesis on the exemplar texture *moving goldenlines* and *waterfall*. Figure 1(b) shows that the learned dynTextons decay very fast in space and time. Figure 1(c)-(d) and (e)-(f) compare the synthesized results of those using full-size dynTextons and truncated dynTextons. It demonstrates that thresholding the dynTextons does not affect the synthesized results, due to their compactness. Moreover, we observe that the synthesized results of these two Gaussian models are visually comparable. In particular, compared with the LDS model [6], in which case the a matrix is of size $N \times N$, the proposed AR-dyntextons are much more compact. More results and videos can be found in the link <http://www.enst.fr/~xia/dynTextures.html>.

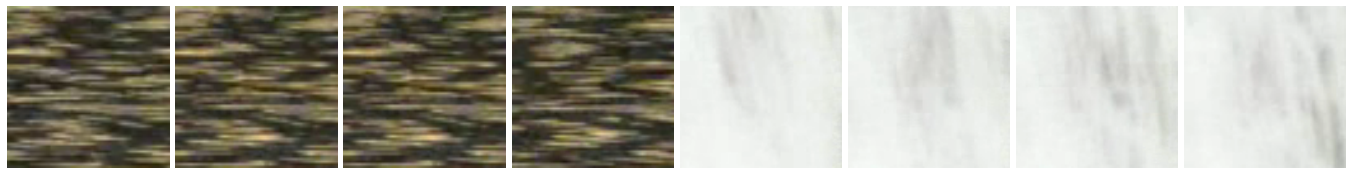
7. CONCLUSION

This paper introduced two compact representations of stationary dynamic textures using Gaussian processes. Both models enable fast analysis and synthesis of dynamic textures. Besides synthesis, these compact representations could also be used for dynamic texture recognition and video compression.

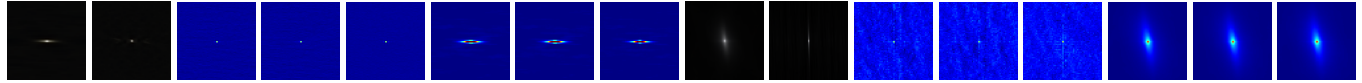
8. REFERENCES

- [1] Martin Szummer and Rosalind W. Picard, "Temporal texture modeling," in *Proc. Int. Conf. Image Processing*, Sep. 1996, vol. 3, pp. 823–826.

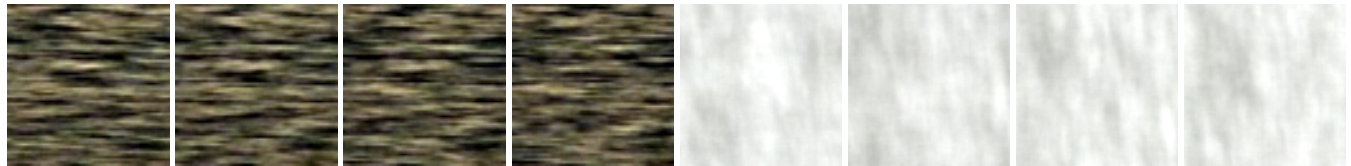
¹The dataset can be downloaded from <http://www.enst.fr/~xia/dynTextures.html>.



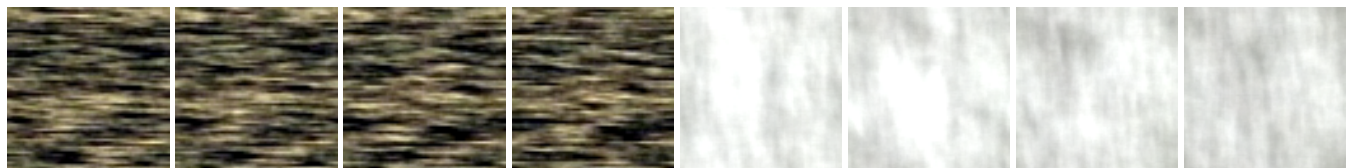
(a) 4 frames of the exemplar texture video f



(b) learned dynTextons: from left to right, the XY and XT plane of the SN-dynTextons, and the filters $a_{1,1}, a_{2,2}, a_{3,3}, b_{1,1}, b_{2,2}, b_{3,3}$ of the AR-dynTextons.



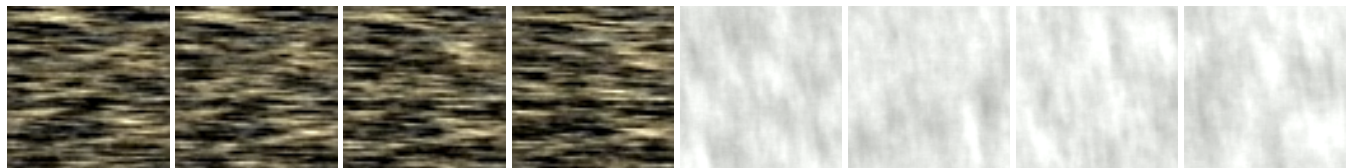
(c) 4 frames of a synthesized dynamic texture using the full-size SN-dynTextons learned from f



(d) 4 frames of a synthesized dynamic texture using truncated SN-dynTextons with size being half of f



(e) 4 frames of the synthesized dynamic textures using full-size AR-dynTextons learned from f



(f) 4 frames of the synthesized dynamic textures using truncated AR-dynTextons with size as half of f

Figure 1: Results on stationary dynamic texture synthesis. (a) displays 4 frames of the exemplar video f ; (b) shows the learned dynTextons; Observe that the learned SN-dynTexton is a 3D space-time filter. (b) only displays two planes, XY and XT, that intersect at the center of the space-time cuboid. Notice that the learned dynTextons have compact support, thus truncating dynTextons produces similar results. More synthesis results can be found at <http://www.enst.fr/~xia/dynTextures.html>.

- [2] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto, "Dynamic textures," *Int. J. Comput. Vision*, vol. 51, no. 2, pp. 91–109, 2003.
- [3] Ziv Bar-Joseph, Ran El-Yaniv, Dani Lischinski, and Michael Werman, "Texture mixing and texture movie synthesis using statistical learning," *IEEE Trans. Vis. Comput. Graphics*, vol. 7, pp. 120–135, 2001.
- [4] Roberto Costantini, Luciano Sbaiz, and Sabine Süsstrunk, "Higher Order SVD Analysis for Dynamic Texture Synthesis," *IEEE Trans. on Image Processing*, vol. 17, no. 1, pp. 42–52, 2008.
- [5] G. Peyré, "Dynamic texture synthesis with grouplets," in *Proc. MAPMO workshop on image processing*, 2009, pp. 103–117.
- [6] Gianfranco Doretto, Eagle Jones, and Stefano Soatto, "Spatially homogeneous dynamic textures," in *Proc. European Conf. Computer Vision*, 2004, pp. 591–602.
- [7] Alexei Efros and Thomas Leung, "Texture synthesis by non-parametric sampling," in *Proc. Int. Conf. Computer Vision*, 1999, pp. 1033–1038.
- [8] Li-Yi Wei and Marc Levoy, "Texture synthesis over arbitrary manifold surfaces," in *Proc. ACM SIGGRAPH*, 2001, pp. 355–360.
- [9] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick, "Graphcut textures: image and video synthesis using graph cuts," in *Proc. ACM SIGGRAPH*, 2003, pp. 277–286.
- [10] J. Portilla and E. P. Simoncelli, "A parametric texture model based on joint statistics of complex wavelet coefficients," *Int. J. Comput. Vision*, vol. 40, no. 1, pp. 49–70, 2000.
- [11] Yizhou Wang and Song-Chun Zhu, "Analysis and synthesis of textured motion: Particles and waves," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, pp. 1348–1363, 2004.
- [12] A. Desolneux, L. Moisan, and S. Ronsin, "Vers un texton pour les micro-textures," in *GRETSI*, 2011.
- [13] B. Galerne, Y. Gousseau, and J-M. Morel, "Random phase textures: Theory and synthesis," *IEEE Trans. on Image Processing*, vol. 20, no. 1, pp. 257–267, 2011.
- [14] Lionel Moisan, "Periodic plus smooth image decomposition," *J. Math. Imag. Vis.*, vol. 39, no. 2, pp. 161–179, 2011.