Spatially Homogeneous Dynamic Textures

Gianfranco Doretto, Eagle Jones, and Stefano Soatto

UCLA Computer Science Department
Los Angeles, CA 90095-1596
{doretto, eagle, soatto}@cs.ucla.edu

Abstract. We address the problem of modeling the spatial and temporal second-order statistics of video sequences that exhibit both spatial and temporal regularity, intended in a statistical sense. We model such sequences as dynamic multiscale autoregressive models, and introduce an efficient algorithm to learn the model parameters. We then show how the model can be used to synthesize novel sequences that extend the original ones in both space and time, and illustrate the power, and limitations, of the models we propose with a number of real image sequences.

1 Introduction

Modeling dynamic visual processes is a fundamental step in a variety of applications ranging from video compression/transmission to video segmentation, and ultimately recognition. In this paper we address a small but important component of the modeling process, one that is restricted to scenes (or portions of scenes) that exhibit both spatial and temporal regularity, intended in a statistical sense. Modeling visually complex dynamic scenes will require higher-level segmentation processes, which we do not discuss here, but having an efficient and general model of the spatio-temporal statistics at the low level is important in its own right.

While it has been observed that the distribution of intensity levels in natural images is far from Gaussian [1,2], the highly kurtotic nature of such a distribution is due to the presence of occlusions or boundaries delimiting statistically homogeneous regions. Therefore, *within such regions* it makes sense to employ the simplest possible model that can capture at least the second-order statistics. As far as capturing the temporal statistics, it has also been shown that linear Gaussian models of high enough order produce synthetic sequences that are perceptually indistinguishable from the originals, for sequences of natural phenomena that are well-approximated by stationary processes [3,4].

In this work, therefore, we seek to jointly model the spatio-temporal statistics of sequences of images that exhibit both spatial and temporal regularity using a simple class of dynamic multiscale autoregressive (MAR) models. We show how model parameters can be efficiently learned (iteratively, if the maximum likelihood solution is sought, or in closed-form if one can live with a sub-optimal estimate), and how they can be employed to synthesize sequences that extend in both space and time the original ones. This work builds on a number of existing contributions, which we summarize below.

1.1 Relation to previous work

While extensive research has been focused on 2D texture analysis, we bypass this literature and focus only on that work which models textures in time. Such dynamic textures were first explored by Nelson and Polana [5], who extract spatial and temporal features from regions of a scene characterized by complex, non-rigid motion.

Szummer and Picard [6] use a spatio-temporal autoregressive model that creates local models of individual pixels based on causal neighbors in space and time. Bar-Joseph [7] employs multiresolution analysis of the spatial structure of 2D textures, and extends the idea to dynamic textures. Multiresolution trees are constructed for dynamic textures using a 3D wavelet transform; mutliple trees from the input are statistically merged to generate new outputs. This technique is unable to generate an infinite length sequence, however, as the trees span the temporal axis.

Video textures were developed by Schödl et al. [8], in which a transition model for frames of the original sequence is developed, alowing the original frames to be looped in a manner that is minimally noticeable to the viewer. Wei and Levoy [9] developed a technique in which pixels are generated by searching for a matching neighborhood in the original sequence.

In [3,4] we propose a simplified system identification algorithm for efficient learning of the temporal statistics, but no explicit model of the spatial statistics. Temporal statistics is also exploited by Fitzgibbon [10] to estimate camera motion in dynamic scenes. Recent work in texture synthesis includes a 2D and 3D patch-based approach by Kwatra et al. [11], using graph cuts to piece together new images and sequences. Wang et al. employ Gabor and Fourier decompositions of images, defining “movetons” [12], combined with statistical analysis of motion.

Our work aims at developing models of *both* the spatial and temporal statistics of a sequence that exploit statistical regularity in both domains. We seek the simplest class of models that achieve the task of capturing arbitrary second-order statistics, in the spirit of [4]. We achieve the goal within the MAR framework, and provide an efficient, closed-form learning algorithm to estimate the parameters of the model. We show this approach to be effective in capturing a wide variety of phenomena, allowing efficient description, compression, and synthesis.

2 Image representation

In this section we introduce the class of multiscale stochastic models to be used in this paper, and describe how an image, that we model as a random field, can be represented in the multiscale framework. It has been shown that this framework can capture a very rich class of phenomena, ranging from one-dimensional Markov processes to $1/f$-like processes [13] and Markov random fields (MRFs) [14].

2.1 Multiscale autoregressive processes

The processes of interest are defined on a tree $\mathcal{T}$; we denote the nodes of $\mathcal{T}$ with an abstract index s. We define an upward shift operator γ such that $s\gamma$ is the parent node of s. We consider regular trees where each node has q children, and define a downward shift operator α, such that the children of s are indexed by $s\alpha_1, \ldots, s\alpha_q$ (see Fig. 1(a)). The nodes of the tree are organized in scales enumerated from 0 to M. The root node, $s = 0$, is the coarsest scale, while the finest scale consists of q^M nodes. We indicate the scale of node s with $d(s)$.

A multiscale autoregressive process $x(s) \in \mathbb{R}^{n(s)}$, $s \in \mathcal{T}$, is described via the scale-recursive dynamic model:

$$x(s) = A(s)x(s\gamma) + B(s)v(s) \quad , \tag{1}$$

under the assumptions that $x(0) \sim \mathcal{N}(0, P_0)$, and $v(s) \sim \mathcal{N}(0, I)$, where $v(s) \in \mathbb{R}^{k(s)}$ and $A(s)$ and $B(s)$ are matrices of appropriate size. The state variable $x(0)$ provides an initial condition for the recursion, while the driving noise $v(s)$ is white and independent of the initial condition.

Notice that model (1) is Markov from scale-to-scale. More importantly, any node s on the q-adic tree can be viewed as a boundary between $q+1$ subsets of nodes (corresponding to paths leading towards the parent and the q offspring nodes). If we denote with $\Upsilon_1(s), \ldots, \Upsilon_q(s), \Upsilon_{q+1}(s)$, the corresponding $q+1$ subsets of states, the following property holds:

$$p(\Upsilon_1(s), \ldots, \Upsilon_q(s), \Upsilon_{q+1}(s)|x(s)) = p(\Upsilon_1(s)|x(s)) \cdots p(\Upsilon_{q+1}(s)|x(s)) \quad . \tag{2}$$

This property implies that there are extremely efficient and highly parallelizable algorithms for statistical inference [15], which can be applied to noisy measurements $y(s) \in \mathbb{R}^{m(s)}$ of the process given by:

$$y(s) = C(s)x(s) + w(s) \quad , \tag{3}$$

where $w(s) \sim \mathcal{N}(0, R(s))$ represents the measurement noise, and $C(s)$ is a matrix of appropriate size, specifying the nature of the process observations as a function of spatial location and scale.

2.2 Multiscale representation of images

We now describe how a Markov random field may be exactly represented in a multiscale framework. A thorough development of the material in this section may be found in [14].

Let us consider a regular discrete lattice $\Omega \in \mathbb{Z}^2$. The essence of the definition of a MRF $y(\boldsymbol{x})$, $\boldsymbol{x} \in \Omega$ is that there exists a neighborhood set $\Gamma_{\boldsymbol{x}}$, such that [16]

$$p(y(\boldsymbol{x})|\{y(\boldsymbol{z})|\boldsymbol{z} \neq \boldsymbol{x}\}) = p(y(\boldsymbol{x})|\{y(\boldsymbol{z})|\boldsymbol{z} \in \Gamma_{\boldsymbol{x}}\}) \quad . \tag{4}$$

For example, the *first-order* neighborhood of a lattice point consists of its four nearest neighbors, and the *second-order* neighborhood consists of its eight nearest neighbors.

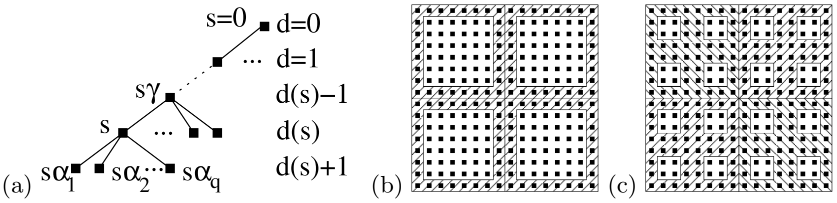

Fig. 1. (a) The state vector in a multiscale stochastic model is indexed by the nodes of a q-adic tree. The tree is a set of connected nodes rooted at 0. For a given node s, $s\gamma$ indexes the parent node of s, while $s\alpha_1, \ldots, s\alpha_q$ index its children. Finally, $d(s)$ indicates the scale of the node s. (b) The shaded region depicts the set $\mathit{\Gamma}(0)$ corresponding to the root of the tree for a 16×16 lattice. (c) To build the next level of the quad-tree for the multiscale representation, one proceeds recursively, defining $\mathit{\Gamma}(0\alpha_i)$, $i \in \{NW, NE, SE, SW\}$, where the subscripts refer to the spatial location (northwest, northeast, southeast, or southwest) of the particular child node.

We wish to use the multiscale framework to represent processes $y(\boldsymbol{x})$, $\boldsymbol{x} \in \mathit{\Omega}$ that are MRFs under second-order neighbors. If $\mathit{\Omega}$ is a lattice of $2^{M+2} \times 2^{M+2}$ points, a state at node s on the d-th level of the tree is representative of the values of the MRF at $16(2^{M-d+1} - 1)$ points. We denote this set of points as $\mathit{\Gamma}(s)$. The shaded region in Fig. 1(b) depicts the set $\mathit{\Gamma}(0)$ corresponding to the root of the tree of a 16×16 lattice. Moreover, each set $\mathit{\Gamma}(s)$ can be thought of as the union of four mutually exclusive subsets of $4(2^{M-d+1} - 1)$ points, and we denote these subsets as $\mathit{\Gamma}_i(s)$, $i \in \{NW, NE, SE, SW\}$, where the subscripts refer to the spatial location of the subset. In Fig. 1(b), $\mathit{\Gamma}(0)$ is the union of the boundaries of four squares of 8×8 points, located in the northwest, northeast, southeast, and southwest quadrants of $\mathit{\Omega}$. To build the next level of the quad-tree for the multiscale representation, we proceed recursively, defining $\mathit{\Gamma}(0\alpha_i)$, $i \in \{NW, NE, SE, SW\}$. The points corresponding to the four nodes at depth $d = 1$ are shown in Fig. 1(c).

If we define $y(s) = \{y(\boldsymbol{x}) | \boldsymbol{x} \in \mathit{\Gamma}(s)\}$, and $y_i(s) = \{y(\boldsymbol{x}) | \boldsymbol{x} \in \mathit{\Gamma}_i(s)\}$, $i \in \{NW, NE, SE, SW\}$, from (4) it follows that

$$\begin{aligned} & p(y(s\alpha_{NW}), y(s\alpha_{NE}), y(s\alpha_{SE}), y(s\alpha_{SW}) | y(s)) = \\ & \quad p(y(s\alpha_{NW}) | y(s)) \cdots p(y(s\alpha_{SW}) | y(s)) = \\ & \quad p(y(s\alpha_{NW}) | y_{NW}(s)) \cdots p(y(s\alpha_{SW}) | y_{SW}(s)) \ , \end{aligned} \tag{5}$$

and, given (2), it is straightforward that the MRF can be modelled as a multiscale stochastic process, and, in the Gaussian case, this leads to MAR models, given by equations (1) and (3).

3 A representation for space- and time-stationary dynamic textures

We seek a model that can capture the spatial and temporal "homogeneity" of video sequences that can also be used for extrapolation or prediction in both space and time. We remind the reader at this stage that our work, unlike [8,4], *cannot* capture scenes with complex layouts, due to the assumption of stationarity in space and time. In the experimental sections we show examples of sequences that obey the assumptions, as well as one that violates them.

We first observe that, depending on the modeling goals, a sequence of images $\{y(\boldsymbol{x},t)|t=1,\dots,\tau,\boldsymbol{x}\in\Omega\}$, can be viewed as any of the following: (a) a collection of τ realizations of a stochastic process defined on Ω; (b) a realization of a stochastic process defined in time; or (c) a realization of a stochastic process defined in both space and time. Let us consider the statistical description (for simplicity up to second-order), of $y(\boldsymbol{x},t)$, and define the mean of the process $m_y(\boldsymbol{x},t)\doteq E[y(\boldsymbol{x},t)]$, and the correlation function $r_y(\boldsymbol{x}_1,\boldsymbol{x}_2,t_1,t_2)\doteq E[y(\boldsymbol{x}_2,t_2)y^T(\boldsymbol{x}_1,t_1)]$. If in model (a) above we assume $m_y(\boldsymbol{x},t)=m_y\ \forall\ \boldsymbol{x},t$, $r_y(\boldsymbol{x}_1,\boldsymbol{x}_2,t_1,t_2)=r_y(\boldsymbol{x}_2-\boldsymbol{x}_1)\ \forall\ \boldsymbol{x}_1,\boldsymbol{x}_2,t_1,t_2$, then the images are realizations from a stochastic process that is stationary in space; statistical models for such processes correspond to models for what are commonly known as planar 2D textures. If instead we take (b) and make the assumptions $m_y(\boldsymbol{x},t)=m_y(\boldsymbol{x})\ \forall\ t$, $r_y(\boldsymbol{x}_1,\boldsymbol{x}_2,t_1,t_2)=r_y(\boldsymbol{x}_1,\boldsymbol{x}_2,t_2-t_1)\ \forall\ t_1,t_2$, then the sequence of images is a realization of a stochastic process that is stationary in time. There are statistical models that aim to capture the time stationarity of video sequences, such as [4,12]. If in (c) we make the following assumptions:

$$m_y(\boldsymbol{x},t)=m_y\ \ \forall\ t,\boldsymbol{x},\quad r_y(\boldsymbol{x}_1,\boldsymbol{x}_2,t_1,t_2)=r_y(\boldsymbol{x}_2-\boldsymbol{x}_1,t_2-t_1)\ \ \forall\ t_1,t_2,\boldsymbol{x}_1,\boldsymbol{x}_2, \tag{6}$$

then the sequence of images is a realization from a stochastic process that is stationary in time and space. Equation (6) represents the working conditions of this paper.

In order to model the second-order spatial statistics we consider the images $y(\boldsymbol{x},t)$, $t=1,\dots,\tau$, as realizations from a single stationary Gaussian MRF, which can be represented as the output of a MAR model. That is,

$$\begin{cases} x(s,t)=A(s)x(s\gamma,t)+B(s)v(s,t) \\ y(s,t)=C(s)x(s,t)+w(s,t)\ \ , \end{cases} \tag{7}$$

where $x(0,t)\sim\mathcal{N}(0,P_0)$, $v(s,t)\sim\mathcal{N}(0,I)$, $w(s,t)\sim\mathcal{N}(0,R(s))$. Notice that, since the process is stationary in space, we have $A(s_1)=A(s_2)$, $B(s_1)=B(s_2)$, $C(s_1)=C(s_2),R(s_1)=R(s_2)\ \forall\ s_1,s_2\in\mathcal{T}$, such that $d(s_1)=d(s_2)$. Therefore, with a slight abuse of notation, we now index every parameter previously indexed by s, with $d(s)$. For instance, $A(d(s))\doteq A(s)$.

Employing model (7), an image at time t is synthesized by independently drawing samples from the following random sources: $x(0,t)$ at the root of $\mathcal{T}$, and $v(s,t)$, for $s \in \mathcal{T} - \{0\}$. Since subsequent images of a video sequence exhibit strong correlation in time that is reflected as a temporal correlation of the random sources, we model $x(0,t)$ and $v(s,t)$ as the output of an autoregressive model. This means that, if we write $x(0,t) = B(0)v(0,t)$, $v(0,t) \sim \mathcal{N}(0,I)$, then the sources $v(s,t)$, $s \in \mathcal{T}$, evolve according to $v(s,t+1) = F(s,t)v(s,t) + u(s,t)$, where $u(s,t) \sim \mathcal{N}(0,Q(s,t))$ is a white process. Also, since we model sequences that are stationary in time, we have $F(s,t) = F(s)$, and $Q(s,t) = Q(s), \forall t$.

Finally, given a sequence of images $\{y(\boldsymbol{x},t) | t = 1,\ldots,\tau, \boldsymbol{x} \in \Omega\}$, satisfying conditions (6), we write the complete generative model as

$$\begin{cases} v(s,t+1) = F(s)v(s,t) + u(s,t) \\ x(0,t) = B(0)v(0,t) \\ x(s,t) = A(s)x(s\gamma,t) + B(s)v(s,t) \\ y(s,t) = C(s)x(s,t) + w(s,t) \ , \end{cases} \tag{8}$$

where $y(s,t) \in \mathbb{R}^{m(s)}$, $x(s,t) \in \mathbb{R}^{n(s)}$, $v(s) \in \mathbb{R}^{k(s)}$, $v(s,0) \sim \mathcal{N}(0,I)$; the driving noise $u(s,t) \sim \mathcal{N}(0,Q(s))$ and measurement noise $w(s,t) \sim \mathcal{N}(0,R(s))$ are independent across time and nodes of $\mathcal{T}$; and $A(s)$, $B(s)$, $C(s)$, $F(s)$, $Q(s)$, $R(s)$ are matrices of appropriate size.

4 Learning model parameters

Given a sequence of noisy images $\{y(\boldsymbol{x},t) | t = 1,\ldots,\tau, \boldsymbol{x} \in \Omega\}$, we wish to learn the model parameters $A(\cdot)$, $B(\cdot)$, $C(\cdot)$, $F(\cdot)$, along with the noise matrices $Q(\cdot)$ and $R(\cdot)$. This is a system identification problem [17] that can be posed as: given $y(\boldsymbol{x},1),\ldots,y(\boldsymbol{x},\tau)$, find

$$\hat{A}(\cdot),\hat{B}(\cdot),\hat{C}(\cdot),\hat{F}(\cdot),\hat{Q}(\cdot),\hat{R}(\cdot) = \arg \max_{\substack{A(\cdot),B(\cdot),C(\cdot),\\ F(\cdot),Q(\cdot),R(\cdot)}} p(y(1),\ldots,y(\tau)) \tag{9}$$

subject to (8). Problem (9) corresponds to the estimation of model parameters by maximizing the total likelihood of the measurements, and can be solved using an expectation-maximization (EM) procedure [18]. Note that the solution of problem (9) is not unique; in fact, once we choose the dimensions $n(s)$ and $k(s)$ of the hidden states $x(s,t)$ and $v(s,t)$, there is an equivalence class of models that maximizes the total likelihood. This class corresponds to the ensemble of every possible choice of basis for the states. Since we are interested in making broad use of model (8), while retaining its potential for recognition, we look for a learning procedure that selects one particular model from the equivalence class. To this end, we have designed a sub-optimal closed-form learning procedure, outlined in the following section.

4.1 A closed-form solution

In this section we outline a procedure to choose a particular representative from the equivalence class of models that explains the measurements $\{y(\boldsymbol{x},t) | t =$

$1, \ldots, \tau, \boldsymbol{x} \in \Omega\}$. Since describing the details of the procedure would require introduction of tedious notation, we retain only the most important ideas.

We first recall that $y(s,t) = \{y(\boldsymbol{x},t) | \boldsymbol{x} \in \Gamma(s)\}$, and define the set $\mathcal{Y}_\mathcal{D} \doteq \{y(s,t) | d(s) = \mathcal{D}\}$, where $\mathcal{D} = 0, \ldots, M$, and Ω is a regular lattice of $2^{M+2} \times 2^{M+2}$ points. We will indicate with $Y_\mathcal{D}$ a matrix that contains in each column one element $y(s,t)$ of the set $\mathcal{Y}_\mathcal{D}$. For a given t the measurements $y(s,t)$ are assumed to be lexicographically ordered in a column vector of $Y_\mathcal{D} \in \mathbb{R}^{m(s) \times 4^\mathcal{D}\tau}$, and we assume that the columns are suitably ordered in time. If we build the matrices $X_\mathcal{D}$ and $W_\mathcal{D}$ using $x(s,t)$ and $w(s,t)$, as we did with $Y_\mathcal{D}$, we note that $Y_\mathcal{D} = C(s) X_\mathcal{D} + W_\mathcal{D} \; \forall \; s$, such that $d(s) = \mathcal{D}$.

Since we work with images, we are interested in reducing the dimensionality of our data set, and we assume that $m(s) > n(s)$. Moreover, we seek a matrix $C(s)$ such that

$$C(s)^T C(s) = I \; . \tag{10}$$

These two assumptions allow us to fix the basis of the state space where $x(s,t)$ is defined, and eliminate the first source of ambiguity in the learning process. To see this, let us consider the estimation problem $\hat{C}(s), \hat{X}_\mathcal{D} = \arg\min_{C(s), X_\mathcal{D}} \|W_\mathcal{D}\|_F$, subject to (10). Take $Y_\mathcal{D} = U \Sigma V^T$ as the singular value decomposition (SVD) of $Y_\mathcal{D}$, where U and V are unitary matrices ($U^T U = I$ and $V^T V = I$). From the fixed rank approximation property of the SVD [19], the solution to our estimation problem is given by $\hat{C}(s) = U_{n(s)}$, and $\hat{X}_\mathcal{D} = \Sigma_{n(s)} V_{n(s)}^T$, where $U_{n(s)}$ and $V_{n(s)}$ indicate the first $n(s)$ columns of matrices U and V, while $\Sigma_{n(s)}$ is a square matrix obtained by taking the first $n(s)$ elements of each of the first $n(s)$ columns of Σ. At this stage, if desired, it is also possible to estimate the covariance of the measurement noise. If $\hat{W}_\mathcal{D} = Y_\mathcal{D} - \hat{C}(s) \hat{X}_\mathcal{D}$, then the sample covariance is given by $\hat{R}(s) = \hat{W}_\mathcal{D}^T \hat{W}_\mathcal{D} / (4^\mathcal{D} \tau)$.

Now suppose that we are given $X_\mathcal{D} \in \mathbb{R}^{n(s) \times 4^\mathcal{D}\tau}$ and $X_{\mathcal{D}+1} \in \mathbb{R}^{n(s) \times 4^{\mathcal{D}+1}\tau}$; we are interested in learning $A(s)$, such that $d(s) = \mathcal{D}$. Note that the number of columns in $X_{\mathcal{D}+1}$ is four times the number of columns in $X_\mathcal{D}$, and the columns of $X_{\mathcal{D}+1}$ can be reorganized such that $X_{\mathcal{D}+1} = [X_{NW,\mathcal{D}+1} \;\; X_{NE,\mathcal{D}+1} \;\; X_{SE,\mathcal{D}+1} \;\; X_{SW,\mathcal{D}+1}]$, where $X_{i,\mathcal{D}+1}$, $i \in \{NW, NE, SE, SW\}$, represents the values of the state of the i-th quadrant. (For the remainder of this section we omit the specification of i in the set $\{NW, NE, SE, SW\}$.) In this simplified version of the learning algorithm the rest of model parameters are actually sets of four matrices. In fact, $A(s)$ consists of a set of four matrices $\{A_i(S)\}$, that can easily be estimated in the sense of Frobenius by solving the linear estimation problem $\hat{A}_i(s) = \arg\min_{A_i(s)} \|X_{i,\mathcal{D}+1} - A_i(s) X_\mathcal{D}\|_F$. The closed-form solution can be computed immediately using the estimates $\hat{X}_\mathcal{D}$ and $\hat{X}_{i,\mathcal{D}+1}$ as: $\hat{A}_i(s) = \hat{X}_{i,\mathcal{D}+1} \hat{X}_\mathcal{D}^T (\hat{X}_\mathcal{D} \hat{X}_\mathcal{D}^T)^{-1}$.

Once we know $A_i(s)$, we can estimate $B_i(s)$. In order to further reduce the dimensionality of the model, we assume that

$$n(s) > k(s) \; , \quad E[v_i(s,t) v_i(s,t)^T] = I \; . \tag{11}$$

These hypotheses allow us to fix the second source of ambiguity in the estimation of model parameters by choosing a particular basis for the state space where

$v_i(s,t)$ is defined. Let us consider the SVD $X_{i,\mathcal{D}+1}-A_i(s)X_{i,\mathcal{D}} = U_v \Sigma_v V_v^T$, where U_v, and V_v are unitary matrices. We seek $\hat{B}_i(s), \hat{V}_{i,\mathcal{D}} = \arg\min_{B_i(s),V_{i,\mathcal{D}}} \|X_{i,\mathcal{D}+1}-A_i(s)X_{i,\mathcal{D}}-B_i(s)V_{i,\mathcal{D}}\|_F$, subject to (11), where $V_{i,\mathcal{D}}$ can be interpreted as $X_{i,\mathcal{D}}$. Again, from the fixed rank approximation property of the SVD [19], the solution follows immediately as: $\hat{B}_i(s) = U_{v,k(s)} S_{v,k(s)}/2^{\mathcal{D}}$, $\hat{V}_{i,\mathcal{D}} = 2^{\mathcal{D}} V_{v,k(s)}^T$, where $U_{v,k(s)}$ and $V_{v,k(s)}$ indicate the first $k(s)$ columns of the matrices U_v and V_v, while $\Sigma_{v(s)}$ is a square matrix obtained by taking the first $k(s)$ elements of each of the first $k(s)$ columns of Σ_v.

Once we have $V_{i,\mathcal{D}}$, we generate two matrices, $V_{1,i,\mathcal{D}} \in \mathbb{R}^{k(s)\times 4^{\mathcal{D}}(\tau-1)}$ and $V_{2,i,\mathcal{D}} \in \mathbb{R}^{k(s)\times 4^{\mathcal{D}}(\tau-1)}$, with a suitable selection and ordering of the columns of $V_{i,\mathcal{D}}$ such that $V_{2,i,\mathcal{D}} = F_i(s)V_{1,i,\mathcal{D}} + U_{i,\mathcal{D}}$ and $U_{i,\mathcal{D}}$ has in its columns the values of $u_i(s,t)$. Then $F_i(s)$ can be determined uniquely, in the sense of Frobenius, by the solution to $\hat{F}_i(s) = \arg\min_{F_i(s)} \|U_{i,\mathcal{D}}\|_F$, given by $\hat{F}_i(s) = \hat{V}_{2,i,\mathcal{D}}\hat{V}_{1,i,\mathcal{D}}^T(\hat{V}_{1,i,\mathcal{D}}\hat{V}_{1,i,\mathcal{D}}^T)^{-1}$. Finally, from the estimate $\hat{U}_{i,\mathcal{D}} = \hat{V}_{2,i,\mathcal{D}}-\hat{F}_i(s)\hat{V}_{1,i,\mathcal{D}}$, we learn the covariance of the driving noise $u_i(s,t)$: $\hat{Q}_i(s) = U_{i,\mathcal{D}}U_{i,\mathcal{D}}^T/4^{\mathcal{D}}(\tau-1)$.

5 A model for extrapolation in space

While extrapolation or prediction in time is naturally embedded in the first equation of model (8), extrapolation in space outside the domain Ω is not implied because of the lack of the notion of *space causality*. Nevertheless, given that model (8) captures both the spatial and temporal stationarity properties of the measurements, the question of whether it is possible to extrapolate in space also becomes natural.

Given $y_i(0,t)$, $i \in \{NW, NE, SE, SW\}$, with the parameters of model (8), we can synthesize the random field inside $\Gamma_i(0)$. In the same way, if from the values taken by the MRF on Ω, we predict the values $y_a(0,t)$ on the boundaries of the square a (see Fig. 2(a)), then we are able to fill it in. Using this idea we could tile the whole 2D space. Since we already have a model for the spatial and temporal statistics of a single tile, we need only a model for the spatio-temporal prediction of the borders of the tiles. The model that we propose is very similar to model (8) when $s = 0$, and for the square a it can be written as:

$$\begin{cases} v_a(0,t+1) = F_a(0)v_a(0,t) + u_a(0,t) \\ x_a(0,t) = A_a(0)x_\Omega(0,t) + B_a(0)v_a(0,t) \\ y_a(0,t) = C_a(0)x_a(0,t) + w_a(0,t) \ , \end{cases} \tag{12}$$

where the first and the last equations of the model, along with their parameters, have exactly the same meaning as the corresponding equations and parameters in model (8). The only difference is in the second equations of (8) and (12). While the meaning of $B_a(0)$ is the same as $B(0)$, (12) has an additional term $A_a(0)x_\Omega(0,t)$. With reference to Fig. 2(b), we take square a as an example case. If we know the values taken by the MRF on the shaded half-plane, then from the properties of the MRF, only the values on the boundary (the vertical line)

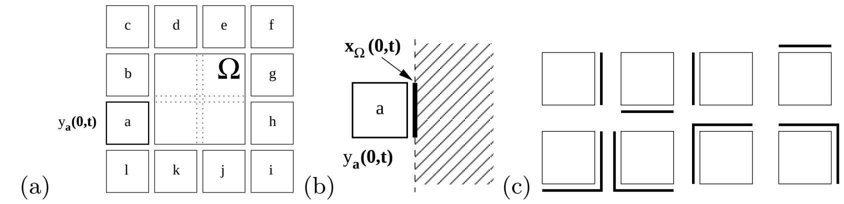

Fig. 2. Extrapolation in space. (a) Patches surrounding the original lattice Ω can be synthesized, if their borders can be inferred from the original data. (b) $y_a(0,t)$, the border of patch a, can be inferred from $x_\Omega(0,t)$, the portion of the original lattice which borders patch a. (c) The eight different ways in which borders must be inferred for a complete tiling of the 2D plane.

affect $y_a(0,t)$. In model (12) we make the further approximation that $y_a(0,t)$ is dependent only on the part of the boundary attached to it. Therefore, $x_\Omega(0,t)$ is a state space representation of the bolded piece of the vertical line in Fig. 2(b). The matrix $A_a(0)$ can be learned from the training set in closed-form, in the sense of Frobenius, similarly to the manner in which we learned $A(s)$; the same is valid for the other parameters of the model. Fig. 2(c) represents the eight cases that need to be handled in order to construct a tiling of the 2D space in all directions.

6 Experiments

We implemented the closed-form learning procedure in Matlab® and tested it with a number of sequences. Figures 3(a)-(e) contain five samples from a dataset we created (boiling water, smoke, ocean waves, fountain, and waterfall). All the sequences have $\tau = 150$ RGB frames, and each frame is 128×128 pixels, so $M = 5$. These sequences exhibit good temporal and spatial stationarity properties. To improve robustness, we normalize the mean and variance of each sequence before running the algorithm. Fig. 3(f) depicts a sequence of fire from the MIT Temporal Texture database. This sequence has $\tau = 100$ RGB frames, with the same dimensions as the other sequences. These images are highly non-Gaussian, and, more importantly, exhibit poor stationarity properties in space. This sequence has been included to "push the envelope" and show what happens when conditions (6) are only marginally satisfied. For the sequences just described, the closed-form algorithm takes less than one minute to generate the model on a high-end (2 GHz) PC.

So far we have made the unspoken assumption that the order of the model $\{n(s), k(s), s \in \mathcal{T}\}$ was given. In practice, we need to solve a model selection problem based on the training set. The problem can be tackled by using information-theoretic or minimum description length criteria [20,21]. Following [22], we automate the selection by looking at the normalized energy encoded in the singular values of Σ and Σ_v, and choose $n(s)$ and $k(s)$ so that we retain a given percentage of energy.

6.1 Synthesis

Using model (8) to extrapolate in time, jointly with model (12) to extrapolate in space, we synthesized new sequences that are 300 frames long, with each frame 256×256 pixels. Synthesis is performed by drawing random vectors $u(s, t)$ from Gaussian distributions, updating the states $v(s, t)$ and $x(s, t)$ and computing the pixel values. The process is implemented simply as matrix multiplications, and a 256×256 RGB image[1] takes as little as five seconds to synthesize. This computational complexity is relatively low, especially compared with MRF models that require the use of the Gibbs sampler. Besides being far less computationally expensive than such techniques, our algorithm does not suffer from the convergence issues associated with them.

Figures 3(a)-(e) show the synthesis results for five sequences. Our model captures the spatial stationarity (homogeneity) and the temporal stationarity of the training sequences[2]. We stress the fact that the training sequences are, of course, not perfectly stationary, and the model infers the "average" spatial structure of the original sequence.

Fig. 3(f) shows the synthesis results for a fire sequence. Since the spatial stationarity assumption is strongly violated, the model captures a "homogenized" spatial structure that generates rather different images from those of the training sequence[2]. Moreover, since the learning procedure factorizes the training set by first learning the spatial parameters ($C(s)$ and $A(s)$), and relies on the estimated state to infer the temporal parameters ($B(s)$ and $F(s)$), the temporal statistics (temporal correlation) appear corrupted if compared with the original sequence.

Finally, we bring the reader's attention to the ability of this model to compress the data set. To this end, we compare it with the classic dynamic texture model proposed in [4]. To a certain extent, this comparison is "unfair", because the classic dynamic texture model does not make any stationarity assumption in space. By making this assumption, however, we do expect to see substantial gains in compression. We generated the classic dynamic texture model, along with our model, for each training sequence, retaining the same percentage of energy in the model order selection step. We found that the ratio between the number of parameters in the classic model and our model ranged from 50 to 80.

7 Conclusion

We have presented a novel technique which integrates spatial and temporal modeling of dynamic textures, along with algorithms to perform learning from training data and generate synthetic sequences. Experimental results show that our method is quite effective at describing sequences which demonstrate temporal and spatial regularity. Our model provides for extension in time and space, and the implementation for both learning and synthesis is computationally inexpensive. Potential applications include compression, video synthesis, segmentation, and recognition.

[1] Multiple color channels are handled in the same manner as [4].

[2] Movies available on-line at http://www.cs.ucla.edu/∼doretto/projects/homogeneous-dynamic-textures.html

Acknowledgments

This work is supported by AFOSR F49620-03-1-0095, NSF ECS-02–511, CCR-0121778, ONR N00014-03-1-0850:P0001.

References

1. Huang, J., Mumford, D.: Statistics of natural images and models. In: Proc. CVPR. (1999) 541–547
2. Srivastava, A., Lee, A., Simoncelli, E., Zhu, S.: On advances in statistical modeling of natural images. J. of Math. Imaging and Vision **18** (2003) 17–33
3. Soatto, S., Doretto, G., Wu, Y.: Dynamic textures. In: Proc. ICCV, Vancouver (2001) 439–446
4. Doretto, G., Chiuso, A., Wu, Y., Soatto, S.: Dynamic textures. Int. J. Computer Vision **51** (2003) 91–109
5. Nelson, R., Polana, R.: Qualitative recognition of motion using temporal texture. CVGIP Image Und. **56** (1992) 78–89
6. Szummer, M., Picard, R.: Temporal texture modeling. In: Proc. ICIP. Volume 3. (1996) 823–826
7. Bar-Joseph, Z., El-Yaniv, R., Lischinski, D., Werman, M.: Texture mixing and texture movie synthesis using statistical learning. IEEE Trans. Vis. Comp. Graphics **7** (2001) 120–135
8. Schödl, A., Szeliski, R., Salesin, D., Essa, I.: Video textures. In: Proc. SIGGRAPH. (2000) 489–498
9. Wei, L., Levoy, M.: Texture synthesis over arbitrary manifold surfaces. In: Proc. SIGGRAPH. (2001) 355–360
10. Fitzgibbon, A.: Stochastic rigidity: Image registration for nowhere-static scenes. In: Proc. ICCV, Vancouver (2001) 662–670
11. Kwatra, V., Schödl, A., Essa, I., Turk, G., Bobick, A.: Graphcut textures: Image and video synthesis using graph cuts. In: Proc. SIGGRAPH. (2003) 277–286
12. Wang, Y., Zhu, S.: A generative method for textured motion: Analysis and synthesis. In: Proc. ECCV. (2002) 583–598
13. Daniel, M., Willsky, A.: The modeling and estimation of statistically self-similar processes in a multiresolution framework. IEEE Trans. Inf. T. **45** (1999) 955–970
14. Luettgen, M., Karl, W., Willsky, A., Tenney, R.: Multiscale representations of markov random fields. IEEE Trans. Sig. Proc. **41** (1993)
15. Chou, K., Willsky, A., Beveniste, A.: Multiscale recursive estimation, data fusion, and regularization. IEEE Trans. Automat. Contr. (1994) 464–478
16. Geman, S., Geman, D.: Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. IEEE Trans. PAMI **6** (1984) 721–741
17. Ljung, L.: System Identification - Theory for the User. Prentice Hall, Englewood Cliffs, NJ (1987)
18. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the em algorithm. J. R. Stat. Soc. B **39** (1977) 185–197
19. Golub, G., Van Loan, C.: Matrix Computations, Third Ed. Johns Hopkins Univ. Press, Baltimore, MD (1996)
20. Lütkepohl, H.: Introduction to Time Series Analysis. Second edn. Springer-Verlag, Berlin (1993)
21. Rissanen, J.: Modeling by shortest data description. Automatica **14** (1978) 465–471
22. Arun, K., Kung, S.: Balanced approximation of stochastic systems. SIAM J. on Matrix Analysis and Apps. **11** (1990) 42–68

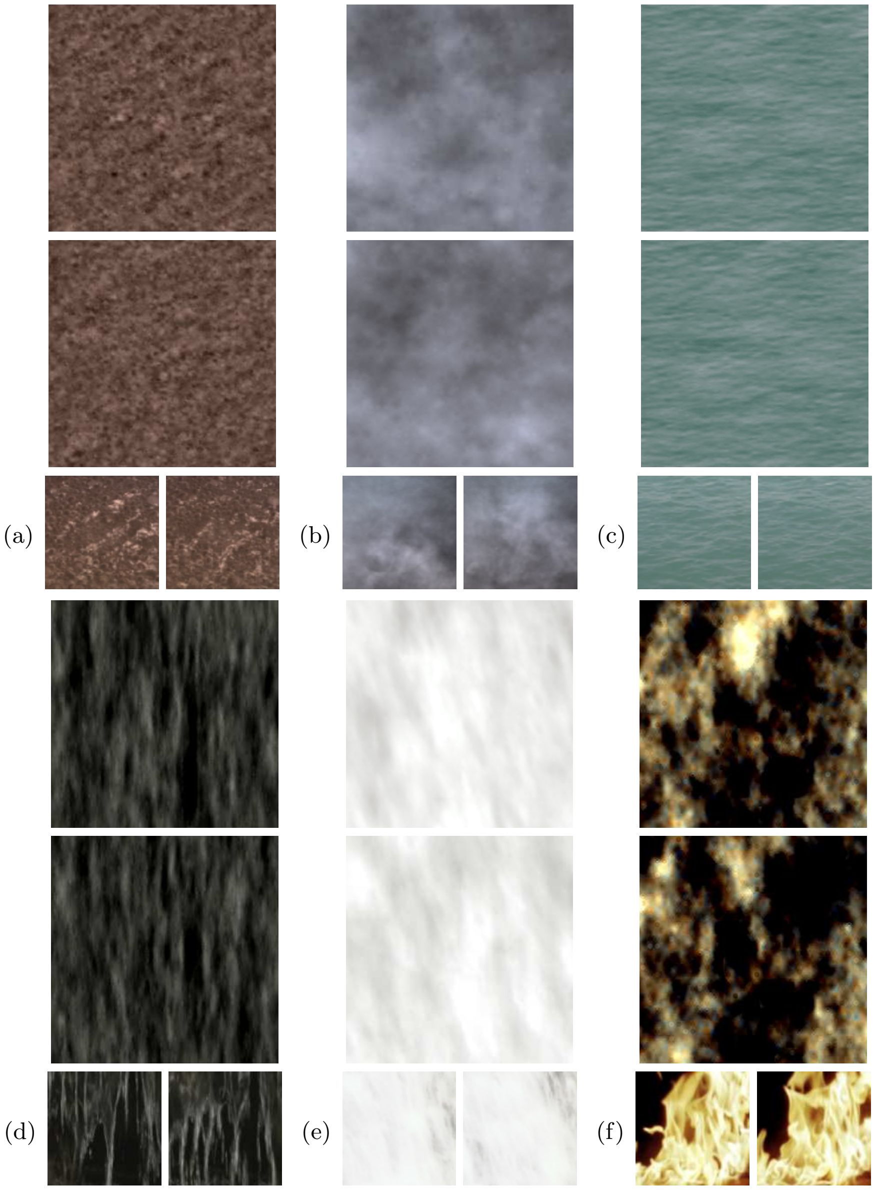

Fig. 3. (a)-(e) frames from five sequences which exhibit strong spatial and temporal regularity. Two frames of 128 × 128 pixels from the original sequences are shown on the bottom, and two synthesized frames (extended to 256 × 256 pixels using our spatial model) are shown on the top. Starting from top: (a) boiling water, (b) smoke, (c) ocean waves, (d) fountain, and (e) waterfall. (f) frames from a fire sequence with poor spatial regularity and non-Gaussian statistics. The frames synthesized by our technique correspond to a spatially "homogenized" version of the original sequence. All the data are available on-line at http://www.cs.ucla.edu/∼doretto/projects/homogeneous-dynamic-textures.html.