Dynamic Textures*

Stefano Soatto
UCLA
Computer Science
Los Angeles - CA 90095, and
Washington University, St.Louis
soatto@ucla.edu, soatto@ee.wustl.edu

Gianfranco Doretto
UCLA
Computer Science
Los Angeles - CA 90095
doretto@cs.ucla.edu

Ying Nian Wu
UCLA
Statistics
Los Angeles - CA 90095
ywu@stat.ucla.edu

Abstract

Dynamic textures are sequences of images of moving scenes that exhibit certain stationarity properties in time; these include sea-waves, smoke, foliage, whirlwind but also talking faces, traffic scenes etc. We present a novel characterization of dynamic textures that poses the problems of modelling, learning, recognizing and synthesizing dynamic textures on a firm analytical footing. We borrow tools from system identification to capture the "essence" of dynamic textures; we do so by learning (i.e. identifying) models that are optimal in the sense of maximum likelihood or minimum prediction error variance. For the special case of second-order stationary processes we identify the model in closed form. Once learned, a model has predictive power and can be used for extrapolating synthetic sequences to infinite length with negligible computational cost. We present experimental evidence that, within our framework, even low dimensional models can capture very complex visual phenomena.

1. Introduction

Consider a sequence of images of a moving scene. Each image is an array of positive numbers that depend upon the shape, pose and motion of the scene as well as upon its material properties (reflectance distribution) and on the light distribution of the environment. It is well known that the joint reconstruction of *photometry* and *geometry* is an intrinsically ill-posed problem: from any (finite) number of images it is not possible to uniquely recover all unknowns (shape, motion, reflectance and light distribution). Traditional approaches to scene reconstruction rely on fixing *some* of the unknowns either by virtue of assumption or by restricting the experimental conditions, while estimating the others[1].

However, such assumptions can never be validated from visual data, since it is always possible to construct scenes with different photometry and geometry that give rise to the same images[2]. The ill-posedness of the most general visual reconstruction problem and the remarkable consistency in the solution as performed by the human visual system reveals the importance of priors for images [29]. They are necessary to fix the arbitrary degrees of freedom and render the problem well-posed. In general, one can use the extra degrees of freedom to the benefit of the application at hand: one can fix photometry and estimate geometry (e.g. in robotic vision), or fix geometry and estimate photometry (e.g. in image-based rendering), or recover a combination of the two that satisfies some additional optimality criterion, for instance the minimum description length of the sequence of video data [23].

Given this arbitrariness in the reconstruction and interpretation of visual scenes, it is clear that there is no notion of a *true* interpretation, and the criterion for *correctness* is somewhat arbitrary. In the case of humans, the interpretation that leads to a correct Euclidean reconstruction (that can be verified by other sensory modalities, such as touch) has obvious appeal, but there is no way in which the correct Euclidean interpretation can be retrieved from visual signals alone.

In this paper we will analyze sequences of images of

*This research is supported in part by NSF grant IIS-9876145, ARO grant DAAD19-99-1-0139. We wish to thank Prabhakar Pundir, and Alessandro Chiuso.

[1] For instance, in stereo and structure from motion one assumes that (most of) the scene has Lambertian reflection properties, and exploits such an assumption to establish correspondence and estimate shape. Similarly, in shape from shading one assumes constant albedo and exploits changes in irradiance to recover shape.

[2] For example, a sequence of images of the sea at sunset could have been originated by a very complex and dynamic shape (the surface of the sea) with constant reflection properties (homogeneous material, water), and also by a very simple shape (e.g. the plane of the television monitor) with a non-homogeneous radiance (the televised spatio-temporal signal). Similarly, the appearance of a moving Lambertian cube can be mimicked by a spherical mirror projecting a light distribution to match the albedo of the cube.

moving scenes solely as visual signals. "Interpreting" and "understanding" a signal amounts to inferring a stochastic model that generates it. The "goodness" of the model can be measured in terms of the total likelihood of the measurements or in terms of its predicting power: a model should be able to give accurate predictions of future signals. Such a model will involve a combination of photometry, geometry and dynamics and will be designed for maximum likelihood or minimal prediction error variance. Notice that we will not require that the reconstructed photometry or geometry be *correct* (in the Euclidean sense), for that is intrinsically impossible without involving (visually) non-verifiable prior assumptions. But the model must be capable of predicting future measurements. In a sense, we look for an "explanation" of the image data that allows us to recreate and extrapolate it. It can therefore be thought of as the compressed version or the "essence" of the sequence of images.

1.1. Prior related work

There has been extensive work in the area of 2D texture analysis, recognition and synthesis. Most of the approaches use statistical models [13, 29, 21, 22, 5, 19, 4, 12] while few others rely on deterministic structural models [8, 28]. Another distinction is that some of them work directly on the pixel values while others project image intensity onto a set of basis functions[3]

There have been many physically based algorithms which target the visual appearance of specific phenomenons [7, 9, 20, 26]. These methods are computationally intensive, customized for particular textures and allow no parameters to control the simulation once a model is inferred.

On the other hand there has been comparatively little work in the specific area of dynamic textures. Schödl *et al.* [24] address the problem by finding transition points in the original video sequence where the video can be looped back on itself in a minimally obtrusive way. The process involves morphing techniques to smooth out visual discontinuities. Levoy and Wei [28] have also suggested extending their approach to dynamic textures by creating a repeatable sequence. The approach is clearly very restrictive and obtains a relatively quick solution for a small subset of problems without explicitly inferring a model.

Bar-Joseph [2] uses multi resolution analysis (MRA) tree merging for the synthesis and merging of 2D textures and extends the idea to dynamic textures. For 2D textures new MRA trees are constructed by merging MRA trees obtained from the input; the algorithm is different from De Bonet's [5] algorithm that operates on a single texture sample. The idea is extended to dynamic textures by constructing MRA trees using a 3D wavelet transform. Impressive results were obtained for the 2D case, but only a finite length sequence is synthesized after computing the combined MRA tree. Our approach captures the essence of a dynamic texture in the form of a dynamic model, and an infinite length sequence can be generated in real-time using the parameters computed off-line and, for the case of second-order process, in closed form.

Szummer and Picard's work [27] on temporal texture modelling uses a similar approach towards capturing dynamic textures. They use the spatio-temporal autoregressive model (STAR), which imposes a neighborhood causality constraint even for the spatial domain. This severely restricts the textures that can be captured. The STAR model fails to capture rotation, acceleration and other simple non translational motions. It works directly on the pixel intensities rather than a smaller dimensional representation of the image. We incorporate spatial correlation without imposing causal restrictions, as would be clear in the coming sections, and can capture more complex motions, including ones where the STAR model is ineffective (see [27], from which we borrow some of the data processed in Section 5).

1.2. Contributions of this work

This work presents several novel aspects in the field of dynamic textures. On the issue of *representation*, we present a novel definition of dynamic texture that is general (even the simplest instance can capture second-order processes with an arbitrary covariance sequence) and precise (it allows making analytical statements and drawing from the rich literature on system identification). On *learning*, we propose two criteria: total likelihood or prediction error. For the case of second-order model we give a closed-form solution of the learning problem. On *recognition*, we show how textures alike tend to cluster in model space. On *synthesis*, we show that even the simplest model (first-order ARMA with white IID Gaussian input) captures a wide range of textures. Our algorithm is simple to implement, efficient to learn and fast to simulate; it allows one to generate infinitely long sequences from short input sequences and to control parameters in the simulation.

2. Representation of dynamic textures

For a single image, one can say it is a texture if it is a realization from a stationary stochastic process with spatially invariant statistics [29]. This definition captures the intuitive notion of texture discussed earlier. For a sequence of images (dynamic texture), individual images are clearly not independent realizations from a stationary distribution, and there is a temporal coherence intrinsic in the process that needs to be captured. The underlying assumption, therefore, is that individual images are realizations of the output of a dynamical system driven by an independent and identically distributed (IID) process. We now make this concept precise as an operative definition of dynamic texture.

[3] Most common methods use Gabor filters [14, 3] and steerable filters [10, 13].

2.1. Definition of dynamic texture

Let $\{I(t)\}_{t=1...\tau}$ be a sequence of images. Suppose that at each instant of time t we can measure a noisy version of the image, $y(t) = I(t) + w(t)$ where $w(t)$ is an independent and identically distributed sequence drawn from a known distribution $p_w(\cdot)$ resulting in a positive measured sequence $y(t) \in \mathbb{R}^m$, $t = 1 \dots \tau$[4]. We say that *the sequence $\{I(t)\}$ is a (linear) dynamic texture* if there exists a set of n spatial filters ϕ_α, $\alpha = 1 \dots n$ and a stationary distribution $q(\cdot)$ such that, calling $x(t) \doteq \phi(I(t))$ we have $x(t) = \sum_{i=1}^{k} A_i x(t-i) + Bv(t)$, with $v(t)$ an IID realization from the density $q(\cdot)$, for some choice of matrices $A_1, \dots, A_k, B$ and initial condition $x(0) = x_0$. Without loss of generality, we can assume $k = 1$ since we can augment the state of the above model to be $\bar{x}(t) \doteq [x(t)^T \; x(t-1)^T \dots x(t-k)^T]^T$. Therefore, a dynamic texture is associated to an auto-regressive, moving average process (ARMA) with unknown input distribution

$$\begin{cases} x(t+1) = Ax(t) + Bv(t) \\ y(t) = \phi(x(t)) + w(t) \end{cases} \tag{1}$$

with $x(0) = x_0$, $v(t) \overset{IID}{\sim} q(\cdot)$ unknown, $w(t) \overset{IID}{\sim} p_w(\cdot)$ given, and $I(t) = \phi(x(t))$. One can obviously extend the definition to an arbitrary non-linear model of the form $x(t+1) = f(x(t), v(t))$, leading to the concept of *non-linear dynamic textures*.

2.2. Filters and dimensionality reduction

The definition of dynamic texture above entails a choice of filters ϕ_α, $\alpha = 1 \dots n$. These filters are also inferred as part of the learning process for a given dynamic texture.

There are several criteria for choosing a suitable class of filters, ranging from biological motivations to computational efficiency. In the trivial case, we can take ϕ to be the identity, and therefore look at the dynamics of individual pixels $x(t) = I(t)$ in (1). We view the choice of filters as a dimensionality reduction step, and seek for a decomposition of the image in the simple (linear) form

$$I(t) = \sum_{i=1}^{n} x_i(t)\theta_i \doteq Cx(t) \tag{2}$$

where $C = [\theta_1, \dots, \theta_n]$ and $\{\theta\}$ can be an orthonormal basis of $\mathcal{L}^2$, a set of principal components, or a wavelet filter bank.

An alternative non-linear choice of filters can be obtained by processing the image with a filter bank, and representing it with the collection of positions of the maximal response in the passband [18]. In this paper we will restrict our attention to linear filters. In [25] we discuss how to extend some of these results to special classes of nonlinear filters.

[4] This distribution can be inferred from the physics of the imaging device. For CCD sensors, for instance, a good approximation is a Poisson distribution with intensity related to the average photon count.

3. Learning dynamic textures

The maximum-likelihood formulation of the dynamic texture learning problem can be posed as follows:

$$\begin{aligned} &\text{given } y(1), \dots, y(\tau), \text{ find} \\ &\hat{A}, \hat{B}, \hat{C}, \hat{x}_0, \hat{q}(\cdot) = \arg\max_{A,B,q} \log p(y(1), \dots, y(\tau)) \\ &\text{subject to (1) and } v(t) \overset{IID}{\sim} q. \end{aligned}$$

The inference method depends crucially upon what type of representation we choose for q. Note that the above inference problem involves the hidden variables $x(t)$ multiplying the unknown parameter A and realizations $v(t)$ multiplying the unknown parameter B, and is therefore intrinsically non-linear even if the original state model is linear. In general one could use iterative techniques that alternate between estimating (sufficient statistic of) the conditional density of the state and maximizing the likelihood with respect to the unknown parameters, in a fashion similar to the expectation-maximization (EM) algorithm [6]. In order for such iterative techniques to converge to a unique minimum, *canonical model realizations* need to be considered, corresponding to particular forms for the matrices A and B. We discuss such realizations in Section 4, where we also present a *closed-form solution* for a wide class of linear dynamic textures.

3.1. Representation of the driving distribution

So far we have managed to defer addressing the fact that the unknown driving distribution belongs, in principle, to an infinite-dimensional space, and therefore something needs to be said about how this issue is dealt with algorithmically.

We consider three ways to approach this problem. One is to transform this into a finite-dimensional inference problem by choosing a parametric class of densities. This is done in the next section, where we postulate that the unknown driving density belongs to a finite-dimensional parameterization of a class of exponential densities, and therefore the inference problem is reduced to a finite-dimensional optimization. The exponential class is quite rich and it includes, in particular, multi-modal as well as skewed densities, although with experiments we show that even a single Gaussian model allows achieving good results. When the dynamic texture is represented by a second-order stationary process we will show in Section 4.2 that, contrary to popular belief, a *closed-form* solution can be obtained.

The second alternative is to represent the density q via a finite number of fair samples drawn from it; the model (1) can be used to represent the evolution of the conditional

density of the state given the measurements, and the density is evolved by updating the samples so that they remain a fair realization of the conditional density as time evolves. Algorithms of this sort are called "particle filters" [16], and in particular the CONDENSATION filter is the best known instance in the Computer Vision community.

The third alternative is to treat (1) as a semi-parametric statistical problem, where one of the parameters (q) lives in the infinite-dimensional manifold of probability densities that satisfy certain regularity conditions, endowed with a Riemannian metric (corresponding to Fisher's Information matrix), and to design gradient descent algorithms with respect to the natural connection, as it has been done in the context of independent component analysis (ICA). This avenue is considerably more laborious and we are therefore not considering it in this study.

3.2. Compression and denoising

Due to the equivalence between stationary second-order covariance sequences and first-order Gauss-Markov models, we can represent a dynamic texture with its model parameters and the input sequence. This, in general, will result in a lossy compression of the original sequence, where the error is given by the innovation process (see [25] for more details). In the experimental section we show a simple example of how a compressed sequence can be obtained.

The model we describe in this paper can also be used to perform suboptimal denoising of the original sequence. It is immediate to see that the denoised sequence is given by

$$\hat{I}(t) \doteq \hat{C}\hat{x}(t) \tag{3}$$

where $\hat{C}$ is the maximum likelihood estimates of C and $\hat{x}(t)$ is obtained from $\hat{x}(t+1) = \hat{A}\hat{x}(t) + \hat{B}\hat{v}(t)$.

4. A closed-form solution for learning second-order stationary processes

It is well known that a stationary second-order process with arbitrary covariance can be modelled as the output of a linear dynamical system driven by white, zero-mean Gaussian noise [17]. In our case, we will therefore assume that there exists a positive integer n, a process $\{x(t)\}$ with initial condition $x_0 \in \mathbb{R}^n$ and symmetric positive-definite matrices Q and R such that

$$\begin{cases} x(t+1) = Ax(t) + v(t) & x(0) = x_0 \ ; \ v(t) \sim \mathcal{N}(0, Q) \\ y(t) = Cx(t) + w(t) & w(t) \sim \mathcal{N}(0, R) \end{cases} \tag{4}$$

for some matrices $A \in \mathbb{R}^{n \times n}$ and $C \in \mathbb{R}^{m \times n}$. The problem of *model identification* consists in estimating the model parameters A, C, Q, R from measurements of $y(1), \ldots, y(\tau)$. Note that B in the model (1) is such that $BB^T = Q$.

4.1. Uniqueness and canonical model realizations

The first observation concerning the model (4) is that the choice of matrices A, C, Q is not unique, in the sense that there are infinitely many such matrices that give rise to exactly the same sample paths $y(t)$ starting from suitable initial conditions. This is immediately seen by substituting A with TAT^{-1}, C with CT^{-1} and Q with TQT^T, and choosing the initial condition Tx_0, where $T \in \mathcal{GL}(n)$ is any invertible $n \times n$ matrix. In other words, the basis of the state-space is arbitrary, and any given process has *not* a unique model, but an *equivalence class* of models $\mathcal{R} \doteq \{[A] = TAT^{-1}, [C] = CT^{-1}, [Q] = TQT^T, \mid T \in \mathcal{GL}(n)\}$. In order to be able to identify a unique model of the type (4) from a sample path $y(t)$, it is therefore necessary to choose a representative of each equivalence class: such a representative is called a *canonical model realization*, in the sense that it does not depend on the choice of basis of the state space (because it has been fixed).

While there are many possible choices of canonical models (see for instance [15]), we are interested in one that is "tailored" to the data, in the sense of having a diagonal state covariance. Such a model is called *balanced* [1]. Since we are interested in data dimensionality reduction, we will make the following assumptions about the model (4):

$$m >> n; \ \mathrm{rank}(C) = n \tag{5}$$

and choose the canonical model that makes the columns of C orthonormal:

$$C^T C = I_n \tag{6}$$

where I_n is the identity matrix of dimension $n \times n$. As we will see shortly, this assumption results in a unique model that is tailored to the data in the sense of defining a basis of the state space such that its covariance $P \doteq lim_{t \to \infty} E[x(t)x^T(t)]$ is asymptotically diagonal.

The problem we set out to solve can then be formulated as follows: *given* measurements of a sample path of the process: $y(1), \ldots, y(\tau)$; $\tau >> n$, estimate $\hat{A}, \hat{C}, \hat{Q}, \hat{R}$, a canonical model of the process $\{y(t)\}$. Ideally, we would want the maximum likelihood solution from the finite sample:

$$\hat{A}(\tau), \hat{C}(\tau), \hat{Q}(\tau), \hat{R}(\tau) = \arg \min_{A,C,Q,R} p(y(1) \ldots y(\tau)) \tag{7}$$

however, in this section we only derive a closed-form suboptimal solution in the sense of Frobenius; the asymptotically optimal solution can also be obtained in closed form, but is beyond the scope of this paper and is described in [25].

4.2. Closed-form solution

Let $Y_1^\tau \doteq [y(1), \ldots, y(\tau)] \in \mathbb{R}^{m \times \tau}$ with $\tau > n$, and similarly for X_1^τ and W_1^τ, and notice that

$$Y_1^\tau = CX_1^\tau + W_1^\tau; \quad C \in \mathbb{R}^{m \times n}; \ C^T C = I \tag{8}$$

by our assumptions (5) and (6). Now let $Y_1^\tau = U\Sigma V^T$; $U \in \mathbb{R}^{m\times n}$; $U^TU = I$; $V \in \mathbb{R}^{\tau\times n}$, $V^TV = I$ be the singular value decomposition (SVD) [11] with $\Sigma = \text{diag}\{\sigma_1, \ldots, \sigma_n\}$, and consider the problem of finding the best estimate of C in the sense of Frobenius: $\hat{C}(\tau), \hat{X}(\tau) = \arg\min_{C,X_1^\tau} \|W_1^\tau\|_F$ subject to (8). It follows immediately from the fixed rank approximation property of the SVD [11] that the unique solution is given by

$$\boxed{\hat{C}(\tau) = U} \quad \boxed{\hat{X}(\tau) = \Sigma V^T} \tag{9}$$

and $\hat{A}$ can be determined uniquely, again in the sense of Frobenius, by solving the following linear problem: $\hat{A}(\tau) = \arg\min_A \|X_1^\tau - AX_0^{\tau-1}\|_F$ which is trivially done in closed form using an estimate of X from (9):

$$\boxed{\hat{A}(\tau) = \Sigma V^T D_1 V (V^T D_2 V)^{-1}\Sigma^{-1}} \tag{10}$$

where $D_1 = \begin{bmatrix} 0 & 0 \\ I_{\tau-1} & 0 \end{bmatrix}$ and $D_2 = \begin{bmatrix} I_{\tau-1} & 0 \\ 0 & 0 \end{bmatrix}$. Notice that $\hat{C}(\tau)$ is uniquely determined up to a change of sign of the components of C and x. Also note that

$$E[\hat{x}(t)\hat{x}^T(t)] \equiv \lim_{\tau\to\infty} \frac{1}{\tau}\sum_{k=1}^{\tau} \hat{x}(t+k)\hat{x}^T(t+k) = \Sigma V^T V \Sigma = \Sigma^2 \tag{11}$$

which is diagonal. Thus the resulting model is *balanced*. Finally, the sample input noise covariance Q can be estimated from

$$\boxed{\hat{Q}(\tau) = \frac{1}{\tau}\sum_{i=1}^{\tau} \hat{v}(i)\hat{v}^T(i)} \tag{12}$$

where $\hat{v}(t) \doteq \hat{x}(t+1) - \hat{A}(\tau)\hat{x}(t)$. Should $\hat{Q}$ not be full rank, its dimensionality can be further reduced by computing the SVD $\hat{Q} = U_Q\Sigma_Q U_Q^T$ where $\Sigma_Q = \text{diag}\{\sigma_Q(1), \ldots, \sigma_Q(k)\}$ with $k \leq n$, and letting $\hat{B}$ be such that $\hat{B}\hat{B}^T = \hat{Q}$.

In the algorithm above we have assumed that the order of the model n was given. In practice, this needs to be inferred from the data. Following [1], we propose to determine the model order empirically from the singular values $\sigma_1, \sigma_2, \ldots$, by choosing n as the cutoff where the value of σ drops below a threshold. A threshold can also be imposed on the difference between adjacent singular values.

4.3. Asymptotic properties

The solution given above is, strictly speaking, *incorrect because the first SVD does not take into account the fact that X has a very particular structure* (i.e. it is the state of a linear dynamical model). It is possible, however, to adapt the algorithm to take this into account while still achieving a closed-form solution that can be proven to be asymptotically efficient, i.e. to approach the maximum likelihood solution. Such an optimal algorithm is beyond the scope of this paper and has been described in [25].

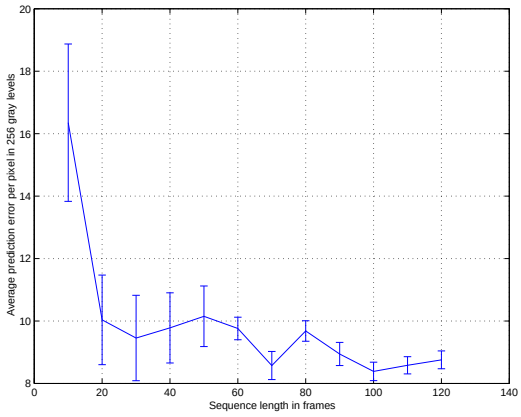

Figure 1. Model verification : *to verify the quality of the model learned, we have used a fixed number of principal components in the representation (20) and considered sub-sequences of the original data set of length varying from* 10 *to* 120*. We have used such sub-sequences to learn the parameters of the model in the Maximum-Likelihood sense, and then used the model to predict the next image. Using one criterion for learning (ML) and another one for validation (prediction error) is informative, for it challenges the model. The average prediction error per pixel is shown as a function of the length of the training sequence (for the* smoke *sequence), expressed in gray scale within a range of* 256 *levels. The average prediction error per pixel is shown as a function of the length of the training sequence (for the* smoke *sequence), expressed in gray scale within a range of 256 levels. The average error per pixel decreases and becomes stable after some critical length. Mean and standard deviation for* 100 *trials is shown as an error-bar plot.*

5. Experiments

We have developed a MATLAB® implementation of the algorithm described in Section 4: learning a sequence of 100 frames takes about 5 minutes on a 1GHz pentium® III PC. Synthesis can be performed at frame rate. In our implementation we have used τ between 50 and 150, n between 20 and 50 and k between 10 and 30.

5.1. Synthesis

Figures 3 to 4 show the behavior of the algorithm on a representative set of experiments. In each case of Figure 4, on the first row we show a few images from the original dataset, on the second row we show a few extrapolated

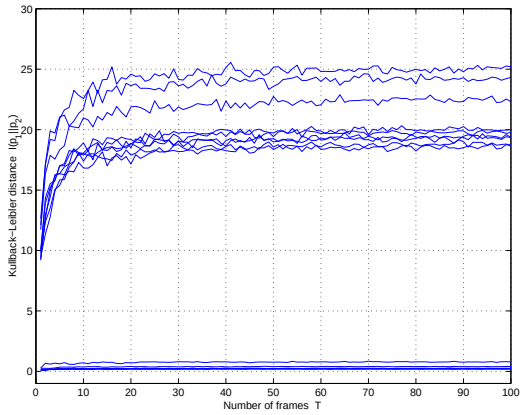

Figure 2. *The figure demonstrates that textures belonging to the same class tend to cluster together in the sense of Kullback-Leibler. In particular for this figure distances are computed amongst three realizations of the* `river` *sequence and three of the* `smoke` *sequence w.r.t. the former. The cluster of graphs on top refer to "smoke w.r.t. river" type of distances and the ones below refer to the "river w.r.t. river" type. The K-L divergences are computed using Monte-Carlo methods.*

samples. Figure 3 shows the overall compression error as a function of the dimension of the state space (top row) as well as the prediction error as a function of the length of the learning set (bottom row). For very regular sequences, the prediction error decreases monotonically; however, for highly complex scenes (e.g. a talking face, smoke), it is not monotonic.

As explained in Section 4, we choose the model order n and learn the parameters of the model. We do so for different lengths τ of the same training sequence; in order to cross-verify our models, we test the prediction error. For each length, τ, we predict the frame $\tau + 1$ (not part of the training set) and compute the prediction error per pixel in gray levels. The results are shown in Figure 1. The average error per pixel decreases and becomes stable after the 80^{th} frame. Thus, the predicting power of the model grows with the length of the training sequence and so does its ability to capture the spatio-temporal dynamics. Furthermore, after some "critical" length of the sequence, there is no improvement in the predicting power: the spatio-temporal dynamics has been captured and the use of a longer sequence does not provide additional information.

5.2. Recognition

We use the *Kullback-Leibler* divergence $I(\cdot\|\cdot)$ to compute the discrepancy between different dynamic textures (represented by the probability density functions p_1 and p_2).

In Figure 2 we display the quantity $I^\tau(p_1\|p_2)$, plotted against the length τ. We have taken different realizations of the textures `river` and `smoke` and have computed the distance of the former realizations against themselves and the latter. It is evident that alike textures tend to cluster together.

5.3. Compression

In this section we present a preliminary comparison between storage requirements for the estimated parameters w.r.t. the original space requirement of the texture sequences, to get an estimate of the sequence comparison capabilities of our model.

Just as anecdotal evidence of the potential of this method for video compression, we point out that the storage requirement of the original dataset is $\mathcal{O}(m\tau)$, while for the model stored is $\mathcal{O}(mn + n^2 + nk + k\tau)$ where $n << m$ and $\tau > n$ and k is the effective rank of $\hat{Q}$. For the sake of example, let $m = 100 \times 100$ (the size of the original image sequence) and $\tau = 100$ (its length). In order to store the original sequence one would need 10^6 numbers. The components of the model that are necessary to re-create an approximation of the sequence are A, C, Q and the input sequence $v(t)$. Typical numbers that result in "acceptable" lossy compression we have observed in our sequences are $n = 20$ and $k = 10$. Therefore, one would need n^2 numbers (for $\hat{A}$), $m \times n - n(n-1)/2$ (for $\hat{C}$, counting the orthogonality constraints), $n \times k$ numbers for $\hat{Q}$ and, finally, $k \times \tau$ numbers for the input sequence $\hat{v}(t)$. For the particular choices above, in order to store a model of the original sequence one would need about 2×10^4 numbers.

Of course, a more systematic evaluation of the potential of this model for compression is due. For very long sequences (large τ) , the algorithm presented above can be modified in order to avoid computing the SVD of a very large matrix. In particular, the model can be identified from a shorter subsequence, and then the identified model can be used to compute the input (in innovation form) using a simple linear Kalman filter. For details on how to do this see [25].

6. Discussion

We have introduced a novel representation of dynamic texture and associated algorithms to perform learning and synthesis of sequences from training data. We have demonstrated experimentally that even the simplest choice in the model (a linear stochastic system driven by Gaussian white noise) can capture complex visual phenomena. The algorithm is simple to implement, efficient to learn and fast to simulate. Some of these results may be useful for image compression and for image-based rendering and synthesis of image sequences.

References

[1] K. Arun and S. Kung. Balanced approximation of stochastic systems. *SIAM Matrix Analysis and Applications, 11*, pages

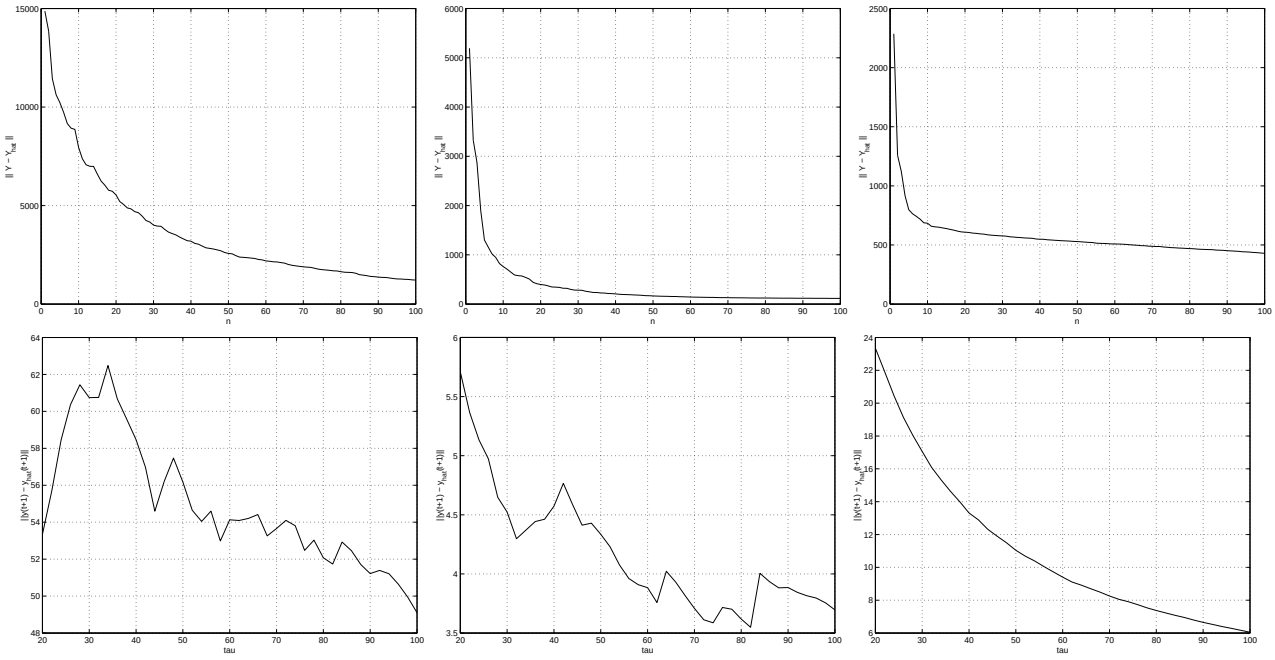

Figure 3. *Compression error as a function of the dimension of the state space n (top row), and extrapolation error as a function of the length of the training set τ (bottom row). Column (a)* `river` *sequence, column (b)* `face` *sequence, column (c)* `toilet` *sequence.*

42–68, 1990.

[2] Z. Bar-Joseph, R. El-Yaniv, D. Lischinski, and M. Werman. Texture mixing and texture movie synthesis using statistical learning. To appear in IEEE Transactions on Visualization and Computer Graphics.

[3] J. Bigun and J. M. du Buf. N-folded symmetries by complex moments in gabor space and their application to unsupervised texture segmentation. In *IEEE transactions on Pattern Analysis and Machine Intelligence,16(1)*, pages 80–87, january 1994.

[4] G. Cross and A. Jain. Markov random field texture models. In *IEEE transactions on Pattern Analysis and Machine Intelligence*, volume 5, pages 25–40, 1983.

[5] J. de Bonet and P. Viola. Multiresolution sampling procedure for analysis and synthesis of texture images. In *Proceedings IEEE Conf. On Computer Vision and Pattern Recognition*, 1998.

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Statist. Soc. B*, 39:185–197, 1977.

[7] D. Ebert, W. Carlson, and R. Parent. Solid spaces and inverse particle systems for controlling the animation of gases and fluids. *The Visual Computer,10(4)*, pages 179–190, September 1994.

[8] A. Efros and T. Leung. Texture synthesis by non-parametric sampling. In *Seventh International Conference on Computer Vision, Corfu, Greece*, 1999.

[9] A. Fournier and W. Reeves. A simple model of ocean waves. In *ACM SIGGRAPH Proceedings*, pages 75–84, 1986.

[10] W. Freeman and E. Adelson. The design and use of steerable filters. In *IEEE transactions on Pattern Analysis and Machine Intelligence, 13(9)*, pages 891–906, 1991.

[11] C. F. V. L. G. H. Golub. *Matrix computations*. Johns Hopkins University Press, Baltimore, MD, 1996.

[12] M. Hassner and J. Sklansky. The use of markov random fields as models of texture. *Image Modeling. Academic Press Inc.*, 1981.

[13] D. Heeger and J. Bergen. Pyramid-based texture analysis /synthesis. In *ACM SIGGRAPH Conference Proceedings*, August 1995.

[14] A. K. Jain and F. Farrokhnia. Unsupervised texture segmentation using gabor filters. *Pattern Recognition*, 24:1167–1186, 1991.

[15] T. Kailath. *Linear Systems*. Prentice Hall, Englewood Cliffs, NJ, 1980.

[16] J. Liu, R. Chen, and T. Logvinenko. A theoretical framework for sequential importance sampling and resampling. Technical report, Stanford University, Department of Statistics, January 2000.

[17] L. Ljung. *System Identification -Theory for the User*. Prentice Hall, Englewood Cliffs, NJ, 1987.

[18] S. Mallat. A theory of multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:674–693, 1989.

[19] R. Paget and D. Longstaff. A nonparametric multiscale markov random field model for synthesising natural textures. In *Fourth International Symposium on Signal Processing and its Applications*, volume 2, pages 744–747, August 1996.

[20] D. Peachey. Modeling waves and surf. In *SIGGRAPH Conference Proceedings*, pages 65–74, 1986.

[21] K. Popat and R. Picard. Novel cluster-based probability model for texture synthesis, classification, and compression. In *Proceedings SPIE visual Communications and Image Processing, Boston*, 1993.

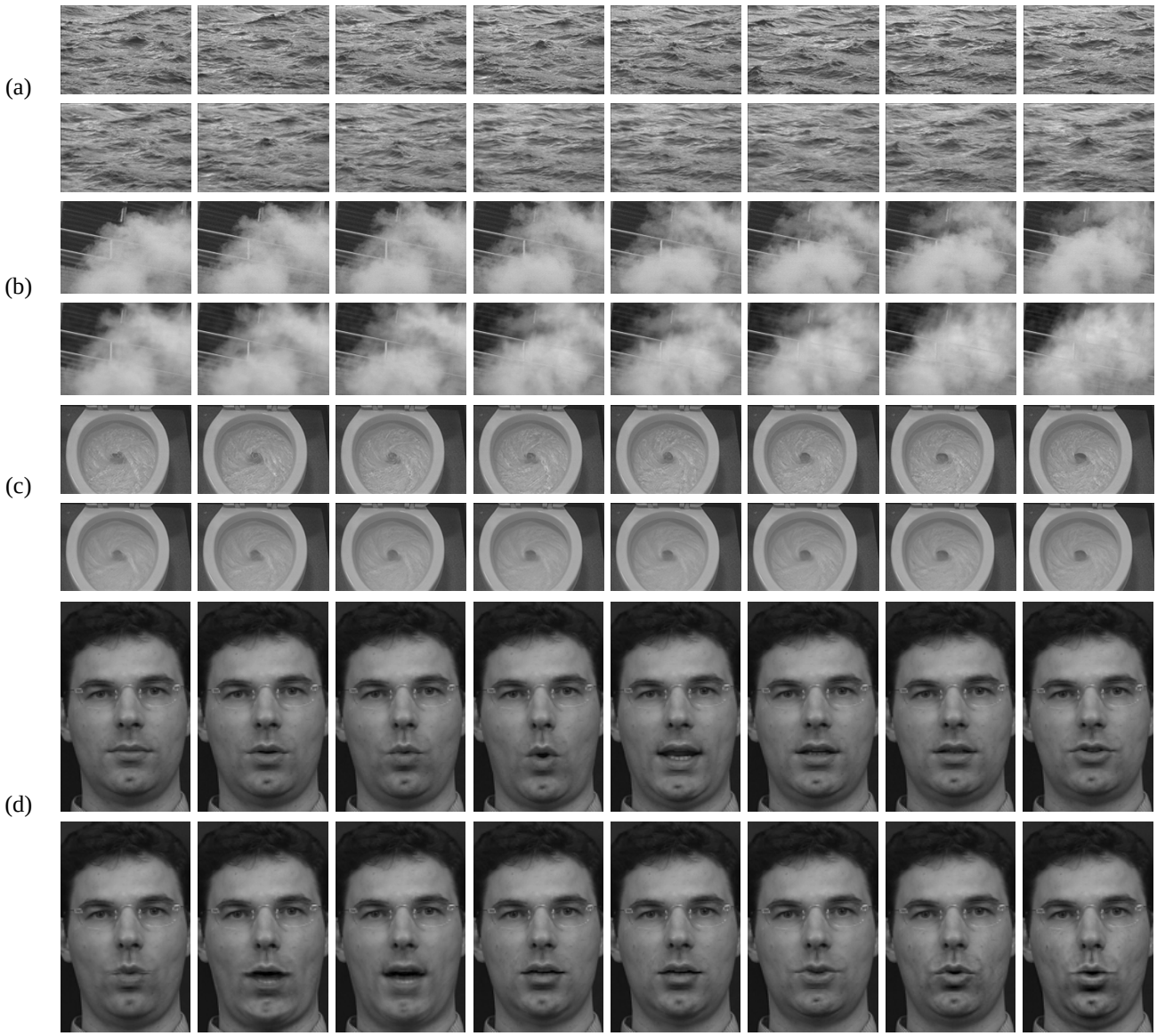

Figure 4. *(a)* `river` *sequence, (b)* `smoke` *sequence, (c)* `toilet` *sequence, (d)* `face` *sequence. For each of them the top row are samples of the original sequence (the* `toilet` *sequence has been borrowed from [27]), the bottom row shows samples of the extrapolated sequence. All the data are available on-line at* `http://www.vision.ucla.edu/projects/dynamic-textures.html`

[22] J. Portilla and E. Simoncelli. Texture representation and synthesis using correlation of complex wavelet coefficient magnitudes. In *CSIC,Madrid*, April 1999.

[23] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[24] A. Schodl, R. Szeliski, D. Salesin, and I. Essa. Video textures. In *Proceedings of ACM SIGGRAPH Conference, New Orleans, LA*, January 2000.

[25] S. Soatto and A. Chiuso. Dynamic data factorization. Technical Report CSD-2001/01, UCLA, Computer Science Department, March 2001.

[26] J. Stam and E. Fiume. Depicting fire and other gaseous phenomena using diffusion processes. In *SIGGRAPH Conference Proceedings*, pages 129–136, 1995.

[27] M. Szummer and R. W. Picard. Temporal texture modeling. In *IEEE International Conference on Image Processing, Lausanne, Switzerland*, volume 3, Sept 1996.

[28] L. Y. Wei and M. Levoy. Fast teture synthesis using tree structured vector quantization. In *SIGGRAPH Conference Proceedings*, 2000.

[29] S. Zhu, Y. Wu, and D. Mumford. Filters, random fields and maximum entropy (frame):towards the unified theory for texture modeling. In *Proceedings IEEE Conf. On Computer Vision and Pattern Recognition*, June 1996.