# Chapter 1

# From Dynamic Texture to Dynamic Shape and Appearance Models: An Overview

Gianfranco Doretto† and Stefano Soatto‡

†*GE Global Research*
*One Research Circle, Niskayuna, NY 12309*
*doretto@research.ge.com*

‡*UCLA Compute Science Department*
*405 Hilgard Ave, Los Angeles, CA 90095*
*soatto@ucla.edu*

In modeling complex visual phenomena one can employ rich models that characterize the global statistics of images, or choose simple classes of models to represent the local statistics of a spatiotemporal segment, together with the partition of the data into such segments. Each segment could be characterized by certain statistical regularity properties in space and/or time. The former approach is often pursued in Computer Graphics, where a global model is necessary to capture effects such as mutual illumination or cast shadows. However, such models cannot be uniquely inferred as they are far more complex than the data, and one has to revert to a much simpler representation that, for instance, models the visual complexity of single segments in terms of statistical variability from a nominal model. In this chapter we do so by modeling the image variability of dynamic scenes through the joint temporal variation of shape and appearance. We describe how this framework can be specialized to Dynamic Texture models for both static and moving cameras. The characterization poses the problems of modeling, learning, and synthesis of video sequences that exhibit certain temporal regularity properties (such as sea-waves, smoke, foliage, talking faces, flags in wind, etc.), using tools from time series analysis, system identification theory, and finite element methods.

## 1.1. Introduction

In modeling complex visual phenomena one can employ rich models that characterize the global statistics of images, or choose simple classes of models to represent the local statistics of a spatio-temporal "segment," together with the partition of the data into such segments. Each segment could be characterized by certain statistical regularity in space and/or time. The former approach is often pursued in computer graphics, where a global model is necessary to capture effects such as mutual illumination or cast shadows. However, such models are not well suited for inference, since they are far more complex than the data, meaning that from any number of images it is not possible to uniquely recover all the unknowns of a scene. In other words, it is always possible to construct scenes with different *photometry* (material reflectance properties, and light distribution), *geometry* (shape, pose, and viewpoint), and *dynamics* (changes over time of geometry and photometry) that give rise to the same images.[1] For instance, the complex appearance of sea waves can be attributed to a scene with simple reflectance and complex geometry, such as the surface of the sea, or with simple geometry and simple reflectance but complex illumination, for instance a mirror reflecting the radiance of a complex illumination pattern. The ill-posedness of the visual reconstruction problem can be turned into a well-posed inference problem within the context of a specific task, and one can also use the extra degrees of freedom to the benefit of the application at hand by satisfying some additional optimality criterion (e.g. the minimum description length (MDL) principle[2] for compression). This way, even though one cannot infer the "physically correct" model of a scene, one can infer a representation of the scene that can be sufficient to support, for instance, recognition tasks.

In this chapter we survey a series of recent papers that describe statistical models that can explain the measured video signal, predict new measurements, and extrapolate new image data. These models are not models of the scene, but statistical models of the video signal. We put the emphasis on sequences of images that exhibit some form of temporal regularity[a], such as sequences of fire, smoke, water, foliage, flags or flowers in wind, clouds, talking faces, crowds of waving people, etc., and we refer to them as *dynamic textures*.[4] In statistical terms, we assume that a dynamic texture is a sequence of images, that is a realization from a stationary stochastic process[b].

[a] The case of sequences that exhibit temporal and spatial regularity is treated in.[3]

[b] A stochastic process is stationary (of order $k$) if the joint statistics (up to order $k$) are time-invariant. For instance a process $\{I(t)\}$ is second-order stationary if its mean $\bar{I} \doteq E[I(t)]$ is constant and its covariance $E[(I(t_1) - \bar{I})(I(t_2) - \bar{I})]$ only depends upon

In order to capture the visual complexity of dynamic textures we model them in terms of statistical variability from a nominal model. The simplest instance of this approach is to use linear statistical analysis to model the variability of a data set as an affine variety; the "mean" is the nominal model, and a Gaussian density represents linear variability. This is done, for instance, in Eigenfaces[5] where appearance variation is modeled by a Gaussian process, in Active Shape Models[6] where shape variation is represented by a Gaussian Procrustean density,[7] and in *Linear Dynamic Texture Models*,[4,8] where motion is captured by a Gauss-Markov process. Active Appearance Models (AAM),[6] or linear morphable models,[9] go one step beyond in combining the representation of appearance and shape variation into a conditionally linear model, in the sense that if the shape is known then appearance variation is represented by a Gaussian process, and vice versa. Naturally, one could make the entire program more general and non-linear by "kernelizing" each step of the representation[10] in a straightforward way.[c]

In this chapter we present a more general modeling framework where we model the statistics of data segments that exhibit temporal stationarity using conditionally linear processes for shape, motion and appearance. In other words, rather than modeling only appearance (eigenfaces), only shape (active shape models) or only motion (linear dynamic texture models), using linear statistical techniques, we model all three simultaneously.[d] The result is the *Dynamic Shape and Appearance Model*,[11,12] a richer model that can specialize to the ones we mentioned before. In Section 1.3. we describe a variational formulation of the modeling framework. In Section 1.4. we show how this framework specializes into a model that explicitly accounts for view-point variability in planar scenes, and subsequently specializes into the linear dynamic texture model.[4,8] In Section 1.5. we set up the general learning problem for estimating dynamic shape and appearance models, and briefly discuss the main difficulties that arise from it. In Section 1.6. we reduce the general learning problem to the case of linear dynamic textures, and provide a closed-form solution, where the case of periodic video signals is also treated. In Section 1.7. the linear dynamic texture model is tested on simulation and prediction, showing that even the simplest instance of the model captures a wide range of dynamic textures. The algorithm is simple to implement, efficient to learn and fast to simulate; it allows gen-

$t_2 - t_1$.

[c]In principle linear processes can model arbitrary covariance sequences given a high enough order, so the advantage of a non-linear model is to provide lower complexity, at the expense of more costly inference.

[d]Eventually this will have to be integrated into a higher-level spatio-temporal segmentation scheme, but such a high-level model is beyond our scope, and here we concentrate on modeling and learning each segment in isolation.

erating infinitely long sequences from short input sequences, and to control the parameters in the simulation.[13] Section 1.8. describes how view-point variability in planar scenes is inferred and then simulated in a couple of real sequences. Finally, in Section 1.9. we test and simulate the more general dynamic shape and appearance model. We compare it to the linear dynamic texture model, and show significant improvement in both fidelity (RMS error) and complexity (model order). We do not show results on recognition tasks, and the interested reader can consult[14] for work in this area.

### 1.2. Related work

Statistical inference for analyzing and understanding general images has been extensively used for the last two decades. There has been a considerable amount of work in the area of 2D texture analysis, starting with the pioneering work of Julesz,[15] until the more recent statistical models (see[16] and references therein).

There has been comparatively little work in the specific area of dynamic (or time-varying) textures. The problem has been first addressed by Nelson and Polana,[17] who classify regional activities of a scene characterized by complex, non-rigid motion. Szummer and Picard's work[18] on temporal texture modeling uses the spatio-temporal auto-regressive model, which imposes a neighborhood causality constraint for both spatial and temporal domain. This restricts the range of processes that can be modeled, and does not allow to capture rotation, acceleration and other simple non translational motions. Bar-Joseph et al.[19] uses multi-resolution analysis and tree-merging for the synthesis of 2D textures and extends the idea to dynamic textures by constructing trees using a 3D wavelet transform.

Other related work[20] is used to register nowhere-static sequences of images, and synthesize new sequences. Parallel to these approaches there is the work of Wang and Zhu[21,22] where images are decomposed by using a dictionary of Gabor or Fourier bases to represent image elements called "movetons," or by computing their primal sketch. The model captures the temporal variability of movetons, or the graph describing the sketches. Finally, in[23] feedback control is used to improve the rendering performance of the linear dynamic texture model we describe in this chapter.

The problem of modeling dynamic textures for the purposes of synthesis has been tackled by the Computer Graphics community as well. The typical approach is to synthesize new video sequences using procedural techniques, entailing clever concatenation or repetition of training image data. The reader is referred to[24–27] and references therein.

The more general dynamic shape and appearance model is also related to the literature of Active Appearance Models. Unlike traditional AAM's, we do not use "landmarks," and our work follows the lines of the more recent efforts in AAM's, such as the work of Baker et al.[28] and Cootes et al.[29,30]

## 1.3. Modeling dynamic shape and appearance

In order to characterize the variability of images in response to changes in the geometry (shape), photometry (reflectance, illumination) and dynamics (motion, deformation) of the scene, we need a model of image formation. That is, we need to know how the image is related to the scene and its changes, and indeed what the "scene" *is*. This is no easy feat, because the complexity of the physical world is far superior to the complexity of the images, and therefore one can devise infinitely many models of the scene that yield the same images. Even the wildly simplified physical/phenomenological models commonly used in Computer Graphics are an overkill, because there are ambiguities in reflectance, illumination, shape and motion. In other words, if the physical scene undergoes changes in one of the factors (say shape), the images can be explained away with changes in another factor (say reflectance). In Appendix A we start with a simple physical model commonly used in Computer Graphics and argue that it can be reduced to a far simpler one where the effects of shape, reflectance and illumination are lumped into an "appearance" function, and shape and motion are lumped into a "shape" function, and dynamics is described by the temporal variation of such functions. Instead of modeling the variability of the images through the independent action of the different physical factors, we model it statistically using a conditionally linear process, that describes the variability from the nominal model.

### 1.3.1. *Image formation model*

In Appendix A we show that, under suitable assumptions, a collection of images $\{I_t(x)\}_{1\leq t\leq \tau}$, $x \in D \subset \mathbb{R}^2$, of a scene made of continuous (not necessarily smooth) surfaces with changing shape, changing reflectance and changing illumination, taken from a moving camera, can be modeled as follows:

$$\begin{cases} I_t(x_t) = \rho_t(x) \,, & x \in \Omega \subset \mathbb{R}^2 \\ x_t = w_t(x) \,, & t = 1, 2, \ldots, \tau \end{cases} \tag{1.1}$$

where $\rho_t : \Omega \subset \mathbb{R}^2 \to \mathbb{R}_+$ is a positive integrable function, which we call *appearance*, and $w_t : \Omega \subset \mathbb{R}^2 \to \mathbb{R}^2$ is a homeomorphism[e] which we call *shape*. In other words, if we think of an image as a function defined on a domain $D$, taking values in the range $\mathbb{R}_+$, the domain and range are called shape and appearance respectively, and their changes are called *dynamics*.

### 1.3.2. *Variability of shape, appearance, and dynamics*

Rather than modeling the variability of the image from physical models of changes of the scene, we are going to learn a statistical model of the variability of the images directly, based on model (1.1). In particular, we are going to assume a very simple model that imposes that changes in shape, appearance and dynamics are conditionally affine. This means that shape is modeled as a Gaussian shape space; given shape, appearance variation is modeled by a Gaussian distribution, and given shape and appearance, motion is modeled by a Gauss-Markov model. Specifically, we assume that

$$w_t(x) = w_0(x) + W(x)s_t \ , \quad x \in \Omega \tag{1.2}$$

where $w_0 : \mathbb{R}^2 \to \mathbb{R}^2$ is a vector-valued function called *nominal warp*, and $W : \mathbb{R}^2 \to \mathbb{R}^{2\times k}$ is a matrix-valued function whose columns are called *principal warps*. The time-varying vector $s_t \in \mathbb{R}^k$ is called the *shape parameter*. Similarly, we assume that

$$\rho_t(x) = \rho_0(x) + P(x)\alpha_t \ , \quad x \in \Omega \tag{1.3}$$

where $\rho_0 : \mathbb{R}^2 \to \mathbb{R}_+$ is called *nominal template*, and the columns of the vector-valued function $P : \mathbb{R}^2 \to \mathbb{R}^{1\times l}$ are the *principal templates*. The time-varying vector $\alpha_t \in \mathbb{R}^l$ is called the *appearance parameter*. The temporal changes of the shape and appearance parameters are modeled by a Gauss-Markov model. This means that there exist matrices $A \in \mathbb{R}^{m\times m}$, $B \in \mathbb{R}^{m\times n}$, $C \in \mathbb{R}^{(k+l)\times m}$, $Q \in \mathbb{R}^{n\times n}$ and a Gaussian process $\{\xi_t \in \mathbb{R}^m\}$ with initial condition $\xi_0$, driven by $n_t \in \mathbb{R}^n$ such that

$$\begin{cases} \xi_{t+1} = A\xi_t + Bn_t \ , & n_t \overset{IID}{\sim} \mathcal{N}(0, Q) \\ \begin{bmatrix} s_t \\ \alpha_t \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \xi_t \ , & \xi_0 \sim \mathcal{N}(\bar{\xi}_0, Q_0) \end{cases} \tag{1.4}$$

[e]A homeomorphism is a continuously invertible map, which we also call "warp." Assuming that $w_t$ is a homeomorphism corresponds to assuming that physical changes in the scene do not result in self-occlusions. However, since we aim at using model (1.1) for deriving a statistical model for the variability of the images, we will see that occlusions can be modeled by changes in appearance, and therefore the assumption is not restrictive. Note that, according to (1.1), the image $I_t(x_t)$ is only defined on $x_t \in w_t(\Omega)$, which may be a subset or a superset of $D$. In the former case, it can be extended to $D$ by regularity, as we describe in this chapter, or by "layering," as described in.[31,32]

where $\{n_t\}$ is a white, zero-mean Gaussian process with covariance $Q$. For convenience we have broken the matrix $C$ into two blocks, $C_1 \in \mathbb{R}^{k\times m}$ and $C_2 \in \mathbb{R}^{l\times m}$, corresponding to the shape and appearance parameters. Also note that without loss of generality one can lump the effect of $B$ into $Q$ and therefore assume $B$ to be the identity matrix.[33] In addition to modeling the temporal variability in (1.4), another property that differentiates this framework is that in traditional active appearance models the variable $x$, in (1.2), belongs to $\{x_1, \dots, x_N\}$, a set of "landmark points," and then it is extended to $D$ in order to perform linear statistical analysis in (1.3). On the other hand, in our model $x$ is defined on the same domain in both equations; the user is not required to define landmarks, and all the shape parameters are estimated during the inference process. Note that the functions $\rho_0$, $P$, $w_0$, and $W$ are not arbitrary and will have to satisfy additional geometric and regularity conditions that we will describe shortly. The complete model of phenomenological image formation can be summarized as follows:

$$\begin{cases} \xi_{t+1} = A\xi_t + n_t \ , \qquad \xi_0 \sim \mathcal{N}(\bar{\xi}_0, Q_0) \ , \quad n_t \overset{IID}{\sim} \mathcal{N}(0, Q) \\ y_t(w_0(x) + W(x)C_1\xi_t) = P(x)C_2\xi_t + \eta_t(x) \ , \quad x \in \Omega \subset \mathbb{R}^2 \end{cases} \tag{1.5}$$

where we assume that only a noisy version of the image $y_t(x) = I_t(x) + \tilde{\eta}_t(x)$ is available on $x \in D$, with noise $\tilde{\eta}_t(x) \overset{IID}{\sim} \mathcal{N}(0, \tilde{R}(x))$. By defining $\eta_t(x) \doteq \rho_0(x) + \tilde{\eta}_t(w_0(x) + W(x)s_t)$, we obtain $\eta_t(x) \overset{IID}{\sim} \mathcal{N}(\rho_0(x), R(x))$, where $R(x) = \tilde{R}(w_0(x) + W(x)s_t)$, and we have absorbed the nominal template as the mean of the noise. We will refer to model (1.5) as the *Dynamic Shape and Appearance (DSA) Model*.

## 1.4. Specializations of the model

The model (1.5) can be further simplified or specialized for particular scenarios. For instance, one may want to model changes in the viewpoint explicitly. As we argue in Appendix A, these are ambiguous if the scene is allowed to deform and change reflectance arbitrarily. However, occasionally one may have knowledge that the scene is rigid in the coarse scale, and variability in the images is only due to changes in albedo (or fine-scale shape) and viewpoint, for instance in moving video of a fountain, or foliage.[20] In this case, following the notation of Appendix A, $w_t(x) = \pi(g_t S(x))$. Depending on $\rho_t(x)$, one may have enough information to infer an estimate of camera motion $g_t$ and shape $S$ up to a finite-dimensional group of transformations, sort of an equivalent of "structure from motion" for a dynamic scene.[20] Note that $S$ is an infinite-dimensional unknown, and therefore inference can be posed in a variational framework following the guidelines of.[1]

One simple case where viewpoint variation can be inferred with a simple finite-dimensional model is when the scene is *planar*, so that $\pi(g_t S(x)) = H_t x$ where $H_t \in \mathbb{GL}(3)/\mathbb{R}$ is an homography[f] (a projective transformation) and $x$ is intended in homogeneous coordinates. The model therefore becomes

$$\begin{cases} \xi_{t+1} = A\xi_t + n_t \\ H_{t+1} = F_t H_t + n_{H_t} \\ y_t(H_t x) = P(x)C_2\xi_t + \eta_t(x) \end{cases} \tag{1.6}$$

where $F_t \in \mathbb{R}^{9\times 9}$ is a (possibly) time-varying matrix and $n_{H_t}$ is a driving noise designed to guarantee that $H_t$ remains an homography. Note that since $P(x)$ and $C_2$ can only be determined as a product, we can substitute them with $C(x) \doteq P(x)C_2$. Moreover, the assumption of a planar scene can be made without loss of generality, since all modeling responsibility for deviations from planarity can be delegated to the appearance $\rho_t(\cdot)$.

Model (1.6) can be further reduced by assuming that not only the scene is planar, but that such a plane is not moving and coincides with the image plane ($H_t$ constant and equal to the identity). This yields

$$\begin{cases} \xi_{t+1} = A\xi_t + n_t \\ y_t(x) = C(x)\xi_t + \eta_t(x) \end{cases} \tag{1.7}$$

where changes in shape are not modeled explicitly and all the modeling responsibility falls on the appearance parameters and principal templates. This is the *Linear Dynamic Texture (LDT) Model*, which is a particular instance of the more general model proposed in.[4,8] It is a linear Gauss-Markov model and it is well known that it can capture the second-order properties of a generic stationary stochastic process.[4]

In the next Section 1.5. we will setup the learning problem and sketch the solution for the case of the DSA model (1.5). For a full derivation of the learning procedure, as well as the learning of model (1.6), the interest reader is referred to.[11,12]

## 1.5. Learning dynamic shape and appearance models

Given a noisy version of a collection of images $\{y_t(x)\}_{1\le t\le \tau}$, $x \in D$, learning the model (1.5) amounts to determining the functions $w_0(\cdot)$ (nominal warp), $W(\cdot)$ (principal warps), $\rho_0(\cdot)$ (nominal template), $P(\cdot)$ (principal templates), the dynamic parameters $A$, $C$ and covariance $Q$ that minimize

[f] $\mathbb{GL}(3)$ is the general linear group of invertible $3 \times 3$ matrices. Homographies can be represented as invertible matrices up to a scale.[34]

a discrepancy measure between the data and the model. In formulas we are looking for[g]

$$\begin{cases} \arg\min_{w_0,W,\rho_0,P,A,C,Q} E\left[\int_\Omega |\eta_t(x) - \rho_0(x)|^2 dx + \nu \|n_t\|^2\right] \\ \text{subject to (1.5) and } \int_\Omega P_{\cdot i}(x) P_{\cdot j}(x)\, \mathrm{d}x = \delta_{ij} = \int_\Omega W_{\cdot i}(x) W_{\cdot j}(x)\, \mathrm{d}x \end{cases} \tag{1.8}$$

The last set of constraints, where $\delta_{ij}$ denote the Kronecker's delta and $P_{\cdot i}$ and $W_{\cdot i}$ represent the $i$-th column of $P$ and $W$ respectively, imposes orthogonality of the shape and appearance bases, and could be relaxed under suitable conditions. The cost function comprises a data fidelity term, and another term that accounts for the linear dynamics in (1.5), weighted according to a regularizing constant $\nu$.

Needless to say, solving (1.8) is a tall order. One of the main difficulties is that it entails performing a minimization in an infinite-dimensional space. To avoid this, in[11,12] we reduce the problem using finite-element methods (FEM),[35] which provide with a straightforward way to regularize the unknowns[h]. The result is an alternating minimization procedure that solves (1.8) iteratively with a minimization in a finite-dimensional space.

An important ambiguity that arises in solving (1.8) is related to the shape and appearance state dimensionality $k$, and $l$. In fact, one could decide a priory how much image variability should be modeled by the shape, and how much by the appearance. For instance, the linear dynamic texture model implicitly assumes that all the modeling responsability is delegated to the appearance ($k = 0$). However, in designing an automatic procedure that infers all the unknowns, this is a fundamental problem. In[11,12] we use model complexity as the arbiter that automatically selects model dimensionality, and assigns how modeling responsibility is shared among appearance, shape, and motion.

Since describing the details of the solution of problem (1.8) is outside the scope of this overview chapter, we refer the interest reader to[11,12] to probe further, and the next Section 1.6. will setup and solve the simpler problem of learning linear dynamic texture models.

## 1.6. Learning linear dynamic texture models

Given a sequence of noisy images $\{y_t(x)\}_{1 \le t \le \tau}$, $x \in D$, learning the linear dynamic texture model (1.7) amounts to identifying the model parameters

[g]In principle the domain of integration $\Omega$ should also be part of the inference process; for comments on this issue, the reader is referred to.[11,12]

[h]Note that for solving problem (1.8) one has to introduce another regularization term, which ensures the shape function $w_t(x)$ to be an homeomorphism, see[11,12] for details.

$A$, $C(x)$, and $Q$. This is a *system identification problem*,[36] where one has to infer a dynamical model from a time series. The maximum-likelihood formulation of the linear dynamic texture learning problem can be posed as follows:

$$\begin{aligned} &\text{given } y_1(x), \dots, y_\tau(x), \text{ find} \\ &\hat{A}, \hat{C}(x), \hat{Q} = \arg\max_{A,C,Q} \log p(y_1(x), \dots, y_\tau(x)) \\ &\text{subject to (1.7) and } n_t \overset{IID}{\sim} \mathcal{N}(0, Q). \end{aligned} \tag{1.9}$$

While we refer the reader to[4] for a more complete discussion about how to solve problem (1.9), how to set out the learning via prediction error methods, and for a more general definition of the dynamic texture model, here we summarize a number of simplifications that lead us to a simple closed-form procedure.

In (1.9) we have to make assumptions on the class of filters $C(x)$ that relate the image measurements to the state $\xi_t$. There are many ways in which one can choose them. However, in texture analysis the dimension of the signal is huge (tens of thousands components) and there is a lot of redundancy. Therefore, we view the choice of filters as a dimensionality reduction step and seek for a decomposition of the image in the simple (linear) form $I_t(x) = \sum_{i=1}^{l} \xi_{i,t}\theta_i(x) \doteq C(x)\xi_t$, where $C(x) = [\theta_1(x), \dots, \theta_l(x)] \in \mathbb{R}^{p \times l}$ and $\{\theta_i\}$ can be an orthonormal basis of $L^2$, a set of principal components, or a wavelet filter bank, and $p \gg l$, where $p$ is the number of pixels in the image.

The first observation concerning model (1.7) is that the choice of matrices $A$, $C(x)$, $Q$ is not unique, in the sense that there are infinitely many such matrices that give rise to exactly the same sample paths $y_t(x)$ starting from suitable initial conditions. This is immediately seen by substituting $A$ with $TAT^{-1}$, $C(x)$ with $C(x)T^{-1}$ and $Q$ with $TQT^T$, and choosing the initial condition $Tx_0$, where $T \in \mathbb{GL}(l)$ is any invertible $l \times l$ 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 $\mathcal{R} \doteq \{[A] = TAT^{-1}, [C(x)] = C(x)T^{-1}, [Q] = TQT^T, \mid T \in \mathbb{GL}(l)\}$. In order to identify a unique model of the type (1.7) from a sample path $y_t(x)$, it is 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[37]), we will make the assumption that $\text{rank}(C(x)) = l$ and choose the canonical model that makes the columns of $C(x)$ orthonormal: $C(x)^T C(x) = I_l$, where $I_l$ is the identity matrix of dimension $l \times l$. 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 is asymptotically diagonal (see Equation (1.14)).

With the above simplifications one may use *subspace identification* techniques[36] to learn model parameters in closed-form in the maximum-likelihood sense, for instance with the well known N4SID algorithm.[33] Unfortunately this is not possible. In fact, given the dimensionality of our data, the requirements in terms of computation and memory storage of standard system identification techniques are far beyond the capabilities of the current state of the art workstations. For this reason, following,[4] we describe a closed-form sub-optimal solution of the learning problem, that takes few seconds to run on a current low-end PC when $p = 170 \times 110$ and $\tau = 120$.

### 1.6.1. *Closed-form solution*

Let $Y_1^\tau \doteq [y_1, \dots, y_\tau] \in \mathbb{R}^{p\times\tau}$ with $\tau > l$, and similarly for $\Xi_1^\tau \doteq [\xi_1, \dots, \xi_\tau] \in \mathbb{R}^{l\times\tau}$ and $N_1^\tau \doteq [\eta_1, \dots, \eta_\tau] \in \mathbb{R}^{p\times\tau}$, and notice that

$$Y_1^\tau = C\Xi_1^\tau + N_1^\tau \ . \tag{1.10}$$

Now let $Y_1^\tau = U\Sigma V^T$; $U \in \mathbb{R}^{p\times l}$; $U^TU = I_l$; $V \in \mathbb{R}^{\tau\times l}$, $V^TV = I_l$ be the singular value decomposition (SVD)[38] with $\Sigma = \text{diag}\{\sigma_1, \dots, \sigma_l\}$, and $\{\sigma_i\}$ be the singular values, and consider the problem of finding the best estimate of $C$ in the sense of Frobenius: $\hat{C}_\tau, \hat{\Xi}_\tau = \arg\min_{C,\Xi_1^\tau} \|N_1^\tau\|_F$ subject to (1.10). It follows immediately from the fixed rank approximation property of the SVD[38] that the unique solution is given by

$$\hat{C}_\tau = U \ , \quad \hat{\Xi}_\tau = \Sigma V^T \ , \tag{1.11}$$

$\hat{A}$ can be determined uniquely, again in the sense of Frobenius, by solving the following linear problem:

$$\hat{A}_\tau = \arg\min_A \|\Xi_1^\tau - A\Xi_0^{\tau-1}\|_F \ , \tag{1.12}$$

where $\Xi_0^{\tau-1} \doteq [\xi_0, \dots, \xi_{\tau-1}] \in \mathbb{R}^{l\times\tau}$ which is trivially done in closed-form using the state estimated from (1.11):

$$\hat{A}_\tau = \Sigma V^T D_1 V (V^T D_2 V)^{-1} \Sigma^{-1} \ , \tag{1.13}$$

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 $\xi$. Also note that

$$E[\hat{\xi}_t \hat{\xi}_t^T] \equiv \lim_{\tau\to\infty} \frac{1}{\tau} \sum_{k=1}^{\tau} \hat{\xi}_{t+k} \hat{\xi}_{t+k}^T = \Sigma V^T V \Sigma = \Sigma^2 \ , \tag{1.14}$$

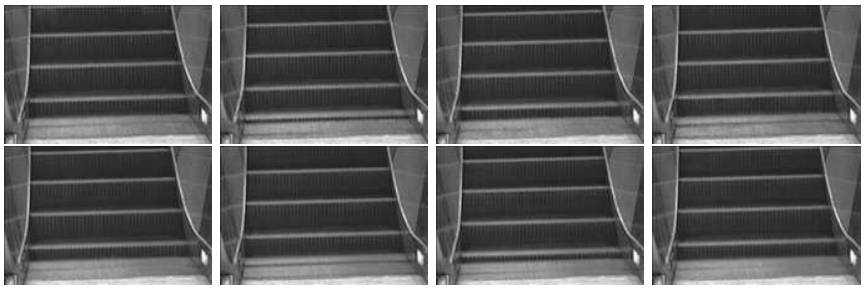

Fig. 1.1. **Escalator.** Example of a dynamic texture that is a periodic signal. Top row: samples from the original sequence (120 training images of 168×112 pixels). Bottom row: extrapolated samples (using $l = 21$ components). The original data set comes from the MIT Temporal Texture database.[18] (From Doretto.[39])

which is diagonal as mentioned in the first part of Section 1.6.. Finally, the sample input noise covariance $Q$ can be estimated from

$$\hat{Q}_\tau = \frac{1}{\tau}\sum_{i=1}^{\tau} \hat{n}_i \hat{n}_i^T \ , \tag{1.15}$$

where $\hat{n}_t \doteq \hat{\xi}_{t+1} - \hat{A}_\tau \hat{\xi}_t$. 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 $\Sigma_Q = \text{diag}\{\sigma_{Q,1}, \ldots, \sigma_{Q,n_v}\}$ with $n_v \leq l$, and one can set $n_t \doteq \hat{B} v_t$, with $v_t \sim \mathcal{N}(0, I_{n_v})$, and $\hat{B}$ such that $\hat{B}\hat{B}^T = \hat{Q}$.

In the algorithm above we have assumed that the order of the model $l$ was given. In practice, this needs to be inferred from the data. Following,[4] one can determine the model order from the singular values $\sigma_1, \sigma_2, \ldots$, by choosing $l$ as the cutoff where the singular values drop below a threshold. If the singular values are normalized according to their total energy,[11,12] the threshold assumes relative meaning and can be consistently used to learn and compare different models. A threshold can also be imposed on the difference between adjacent singular values.

### 1.6.2. *Learning periodic dynamic textures*

The linear dynamic texture model (1.7) is suitable for dynamic visual processes that are periodic signals over time. This can be achieved if $Q = 0$, which means that the model is not excited by driving noise, and all the eigenvalues of $A$ (the poles of the linear dynamical system) are located on the unit circle of the complex plane. In order to learn a model of this kind one can use a slight variation of the procedure highlighted in Section 1.6.1..[39] In fact, in estimating $A$ one has to take into account the eigenvalue

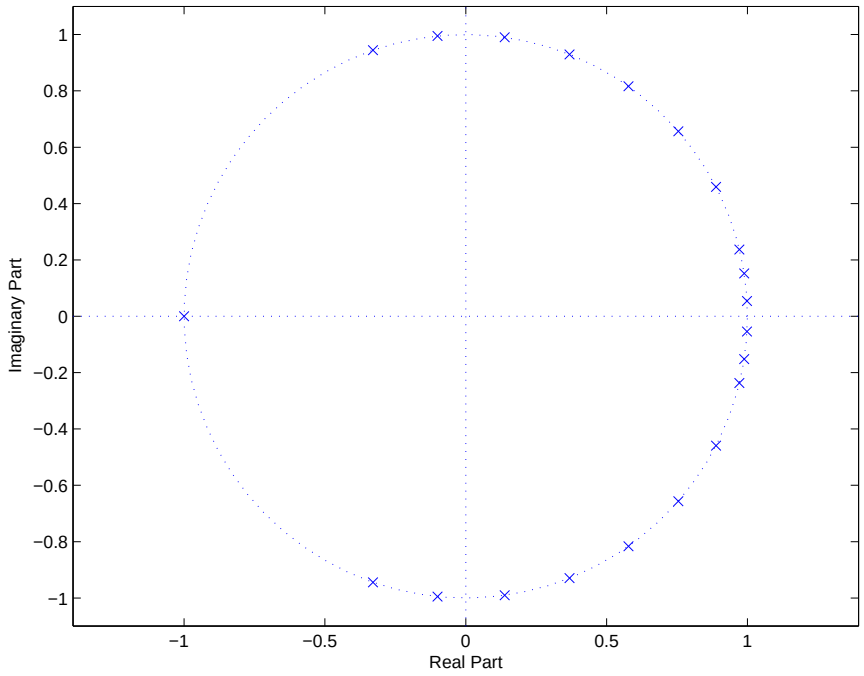


Fig. 1.2. Plot of the complex plane with the eigenvalues of $\hat{A}$ for the escalator sequence (from Doretto[39]).

property, which means that $A$ has to be orthogonal. Adding this constraint transforms problem (1.12) in a Procrustes problem,[38] which can still be solved in closed-form. More precisely, if the SVD of $\Xi_1^\tau {\Xi_0^\tau}^{-1^T}$ is given by $U_A \Sigma_A V_A^T$, one can estimate $A$ as

$$\hat{A}_\tau = \arg \min_{\{A | A^T A = I_n\}} \|\Xi_1^\tau - A\Xi_0^{\tau -1}\|_F = U_A V_A^T \ . \tag{1.16}$$

The top row of Figure 1.1. shows a video sequence of an escalator, which is a periodic signal. The bottom row shows some synthesized frames. The reader may observe that the quality of the synthesized frames makes them indistinguishable from the original ones. Figure 1.2. instead, shows that all the eigenvalues of the matrix $\hat{A}$ lie on the unit circle of the complex plane.

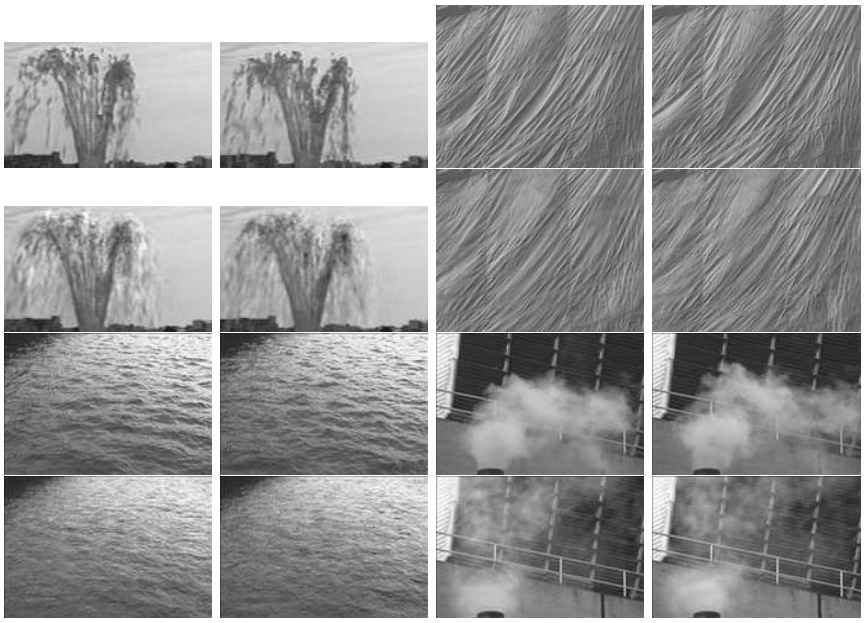

Fig. 1.3. Top row: **Fountain** ($\tau = 100$, $p = 150 \times 90$), **Plastic** ($\tau = 119$, $p = 190 \times 148$). Bottom row: **River** ($\tau = 120$, $p = 170 \times 115$), **Smoke** ($\tau = 150$, $p = 170 \times 115$). For every sequence: Two samples from the original sequence (top row), and two samples from a synthesized sequence (bottom row) (from Doretto et al.,[4] © 2003 Springer-Verlag).

## 1.7. Validation of the linear dynamic texture model

One of the most compelling validations for a dynamic texture model is to simulate it to evaluate to what extent the synthesis captures the essential perceptual features of the original data. Given a typical training sequence of about one hundred frames, using the procedure described in Section 1.6.1. one can learn model parameters in a few seconds, and then synthesize a potentially infinite number of new images by simulating the linear dynamic texture (LDT) model (1.7). To generate a new image one needs to draw a sample $n_t$ from a Gaussian distribution with covariance $Q$, update the state $\xi_{t+1} = A\xi_t + n_t$, and compute the image $I_t = C\xi_t$. This can be done in real time.

Even though the result is best shown in movies[i], Figure 1.3. and Figure 1.4. provide some examples of the kind of output that one can get. They

[i]The interested reader is invited to visit the website `http://vision.ucla.edu/~doretto/` for demos on dynamic texture synthesis.

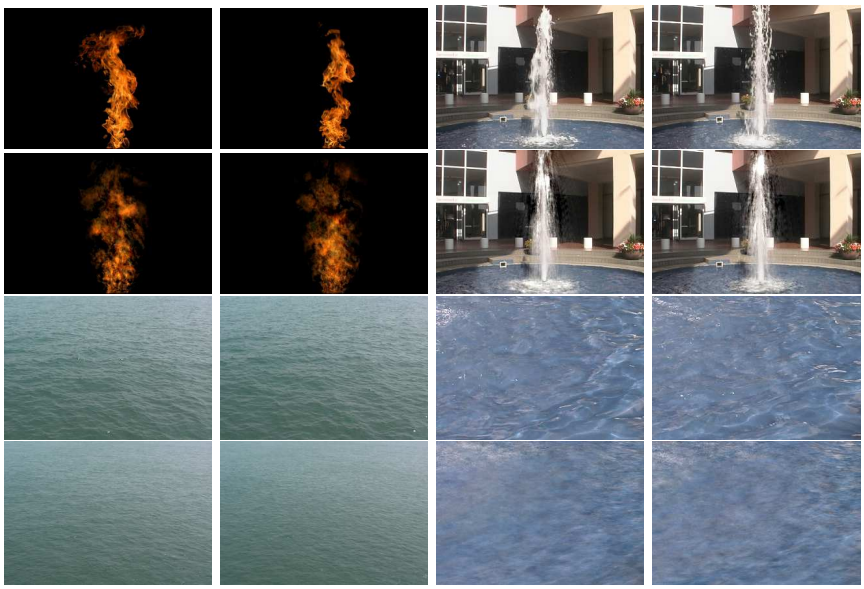

Fig. 1.4. Top row: **Fire** ($\tau = 150$, $p = 360 \times 243$), **Color-fountain** ($\tau = 150$, $p = 320 \times 220$). Bottom row: **Ocean** ($\tau = 150$, $p = 320 \times 220$), **Water** ($\tau = 150$, $p = 320 \times 220$). For every sequence: Two samples from the original sequence (top row), and two samples from a synthesized sequence (bottom row) (from Doretto et al.,[4] © 2003 Springer-Verlag).

show that even the simple model (1.7), which captures only the second-order temporal statistics of a video sequence, is able to capture most of the perceptual features of sequences of images of natural phenomena, such as fire, smoke, water, flowers or foliage in wind, etc. In particular, here the dimension of the state was set to $l = 50$, and $\xi_0$ was drawn from a zero-mean Gaussian distribution with covariance inferred from the estimated state $\hat{\Xi}_1^\tau$. In Figure 1.3., the training sequences were borrowed from the MIT Temporal Texture database,[18] the length of these sequences ranges from $\tau = 100$ to $\tau = 150$ frames, and the synthesized sequences are 300 frames long. In Figure 1.4., the training sets are color sequences that were captured by the authors except for the fire sequence that comes from the Artbeats Digital Film Library[j]. The length of the sequences is $\tau = 150$ frames, the frames are $320 \times 220$ pixels, and the synthesized sequences are 300 frames long.

An important question is how long should the input sequence be in order to capture the dynamics of the process. To answer this question ex-

[j] `http://www.artbeats.com`

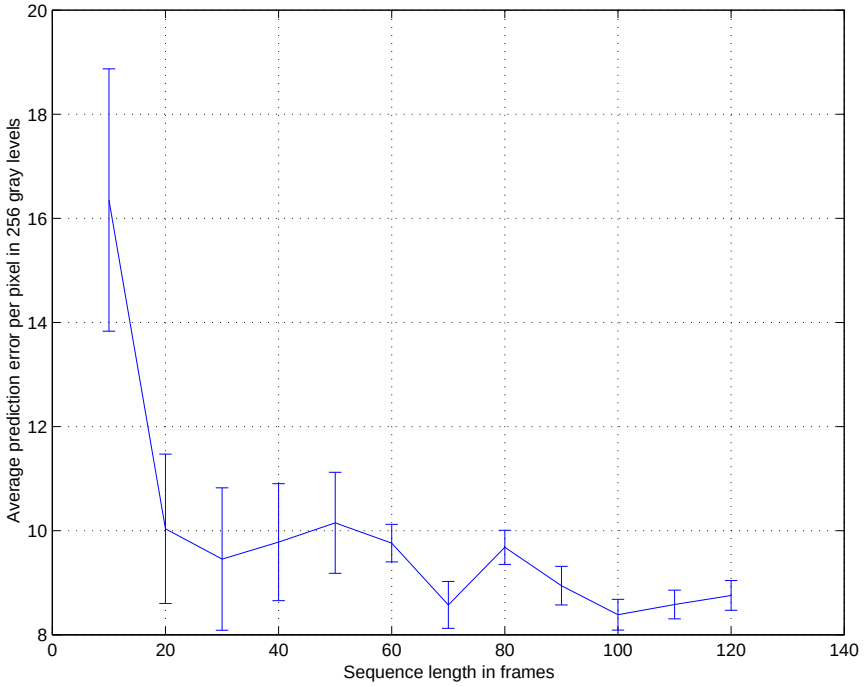


Fig. 1.5. Error-bar plot of the average prediction error and standard deviation (for 100 trials) per pixel (expressed in gray levels with a range of $[0, 255]$), as a function of the length of the steam training sequence. The state dimension is set to $l = 20$ (from Doretto et al.,[4] © 2003 Springer-Verlag).

perimentally, for a fixed state dimension, we consider the prediction error as a function of the length $\tau$, of the input (training) sequence. This means that for each length $\tau$, we predict the frame $\tau + 1$ (not part of the training set) and compute the prediction error per pixel in gray levels. We do so many times in order to infer the statistics of the prediction error, i.e. mean and variance at each $\tau$. Using one criterion for learning (the procedure in Section 1.6.1.), and another one for validation (prediction error) is informative for challenging the model. Figure 1.5. shows an error-bar plot including mean and standard deviation of the prediction error per pixel for the steam sequence. The average error decreases and becomes stable after approximately 70 frames. The plot of Figure 1.5. validates *a-posteriori* the model (1.7) inferred with the procedure described in Section 1.6.1.. Other dynamic textures have similar prediction error plots.[4]

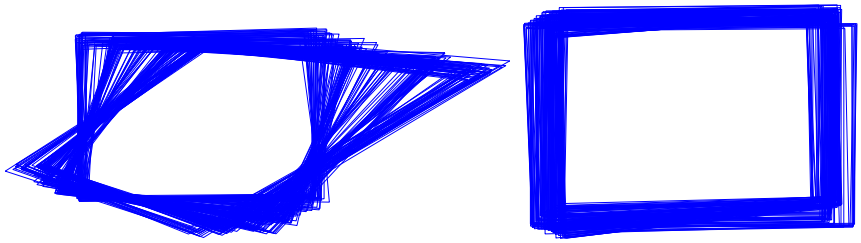

Fig. 1.6. Generation of the appearance domain $\Omega$ for the pool sequence (left), and for the waterfall sequence (right), as the result of the intersection of the domain $D$ mapped according to the inverse of the estimated homographies $\{H_i\}$ (from Doretto and Soatto,[12] © 2006 IEEE).

## 1.8. Simulation of viewpoint variability

We tested model (1.6) with two sequences that we call pool and waterfall. The former has 170, and the latter 130 color frames of $350 \times 240$ pixels. The shape state dimension was set to $k = 8$, whereas the appearance state dimension was learnt with a relative cutoff threshold $\gamma_\rho = 0.01$, giving $l = 34$ for the pool sequence, and $l = 42$ for the waterfall sequence. The interested reader is referred to[11,12] for details on learning model (1.6). Figure 1.6. illustrates the generation of the appearance domain $\Omega$ of the two sequences as the intersection of the original image domain $D$ mapped according to the inverse of the estimated homographies $\{H_i\}$. Figure 1.7. shows two samples of the pool and waterfall sequences along with the same samples after the rectification with respect to the estimated homographies. For each sequence, Figure 1.7. also shows two extrapolated frames obtained by simulating the models and by imposing a synthetic motion[k]. The extrapolated movies are 200 frames long, and the frame dimension is $175 \times 120$ pixels. Notice that only the pixels in the domain $\Omega$ are displayed. For the pool sequence, the synthetic camera motion is such that the camera first zooms in, then translates to the left, turns to the left, right, and finally zooms out. For the waterfall sequence, the synthetic camera motion is such that the camera first zooms in, then translates to the left, down, right, up, and finally rotates to the left.

Although model (1.6) does not capture the physics of the scene, it is sufficient to "explain" the measured data and to extrapolate the appearance of the images in space and time. This model can be used for instance, for the purpose of video editing, as it allows controlling the motion of the

[k]The interested reader is invited to visit the website `http://vision.ucla.edu/~doretto/` for demos on dynamic texture synthesis.

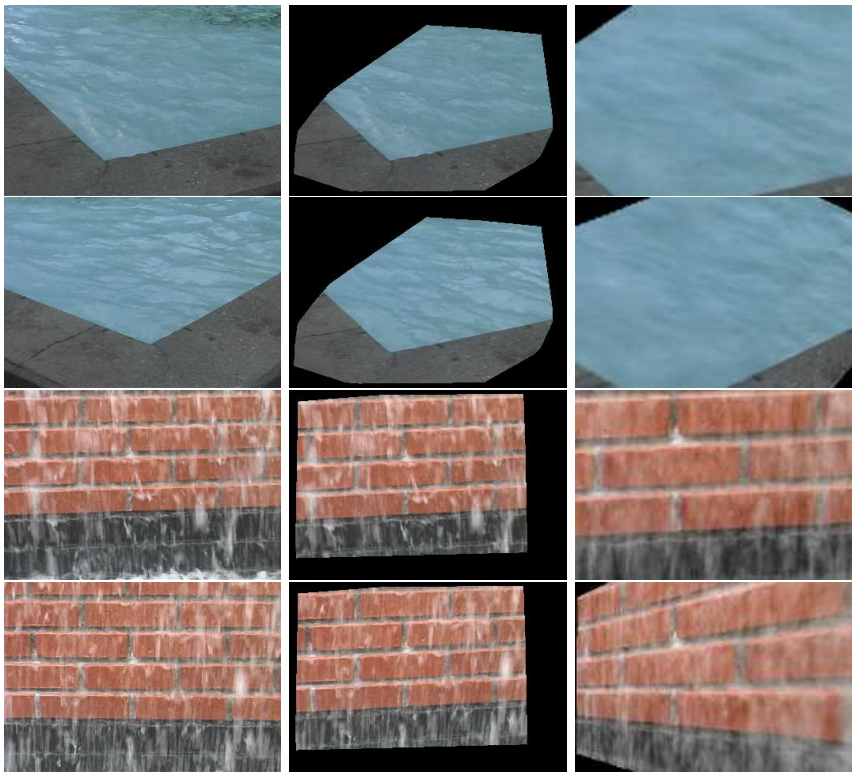

Fig. 1.7. **Pool, Waterfall.** For each sequence: Two samples of the original sequence (left column) and the same samples after the homography registration (middle column), and two samples of a synthesized sequence with synthetic camera motion (right column) (from Doretto and Soatto,[12] © 2006 IEEE).

vantage point of a virtual camera looking at the scene, but also for video stabilization of scenes with complex dynamics.

## 1.9. Validation of the dynamic shape and appearance model

Table 1.1. summarizes some differences between the linear dynamic texture (LDT) model and the dynamic shape and appearance (DSA) model extracted from four different real sequences that we call flowers, duck, flag, and candle. For each of the sequences, the LDT and the DSA models were learnt with the following choice of normalized cutoff thresholds: $\gamma_\rho = 0.01$

Table 1.1. Model complexity and fidelity. For every sequence: $l_{\mathrm{LDT}}$ is the state space dimension of the LDT model, $l_{\mathrm{DSA}}$ and $k_{\mathrm{DSA}}$ are the appearance and shape state dimensions of the DSA model, $\mathrm{RMSE}_{\mathrm{LDT}}$ and $\mathrm{RMSE}_{\mathrm{DSA}}$ are the normalized root mean square reconstruction errors per pixel using the LDT and DSA models respectively.

| Sequence | $l_{\mathrm{LDT}}$ | $l_{\mathrm{DSA}}$ | $k_{\mathrm{DSA}}$ | $\mathrm{RMSE}_{\mathrm{LDT}}$ | $\mathrm{RMSE}_{\mathrm{DSA}}$ | $\mathrm{RMSE}_{\mathrm{LDT2}}$ |
|---|---|---|---|---|---|---|
| flowers | 22 | 19 | 6 | 1.57% | 1.61% | 1.73% |
| candle | 11 | 7 | 7 | 0.83% | 0.84% | 1.14% |
| duck | 16 | 11 | 6 | 0.66% | 0.63% | 0.73% |
| flag | 18 | 10 | 8 | 1.17% | 1.27% | 1.42% |

for the appearance space, and $\gamma_w = 0.03$ for the shape space[l]. In Table 1.1., $l_{\mathrm{LDT}}$ indicates the dimension of the state of the LDT model, whereas $l_{\mathrm{DSA}}$ and $k_{\mathrm{DSA}}$ indicate the appearance and shape state dimensions of the DSA model. Since the majority of the model parameters is used to encode either the principal components of the LDT model, or the principal templates of the DSA model, comparing $l_{\mathrm{LDT}}$ and $l_{\mathrm{DSA}}$ is informative of the reduction of the complexity of the DSA model. As expected, at this reduction corresponds an increase of the shape state dimension, going from zero to $k_{\mathrm{DSA}}$. In particular, Table 1.1. suggests the following empirical relationship: $l_{\mathrm{LDT}} \approx l_{\mathrm{DSA}} + k_{\mathrm{DSA}}$.

Table 1.1. also reports data about the fidelity in the reconstruction of the training sequences from the inferred models. In particular, the last three columns report the normalized root mean square reconstruction errors (RMSE) per pixel. $\mathrm{RMSE}_{\mathrm{LDT}}$ and $\mathrm{RMSE}_{\mathrm{LDT2}}$ are the errors for the LDT model with state dimension $l_{\mathrm{LDT}}$ and $l_{\mathrm{DSA}}$ respectively, whereas $\mathrm{RMSE}_{\mathrm{DSA}}$ is the error for the DSA model. One may notice that $\mathrm{RMSE}_{\mathrm{DSA}}$ and $\mathrm{RMSE}_{\mathrm{LDT}}$ are fairly similar. This is not surprising since the models are inferred while retaining principal components and templates that are above the same cutoff threshold. On the other hand, the comparison between $\mathrm{RMSE}_{\mathrm{LDT}}$ and $\mathrm{RMSE}_{\mathrm{LDT2}}$ highlights the degradation of the reconstruction error when the LDT model is forced to have the same state dimensionality of the appearance state.

Like in Section 1.7., Figure 1.8. and Figure 1.9. show results on the ability of the DSA model to capture the spatio-temporal properties of a video sequence by using the model to extrapolate new video clips[m]. For the four test sequences the figures show frames of the original sequence

[l]Note that for learning the LDT model the threshold is not used because the shape space is assumed to have dimension $k = 0$.

[m]The interested reader is invited to visit the website `http://vision.ucla.edu/~doretto/` for demos on dynamic texture synthesis.

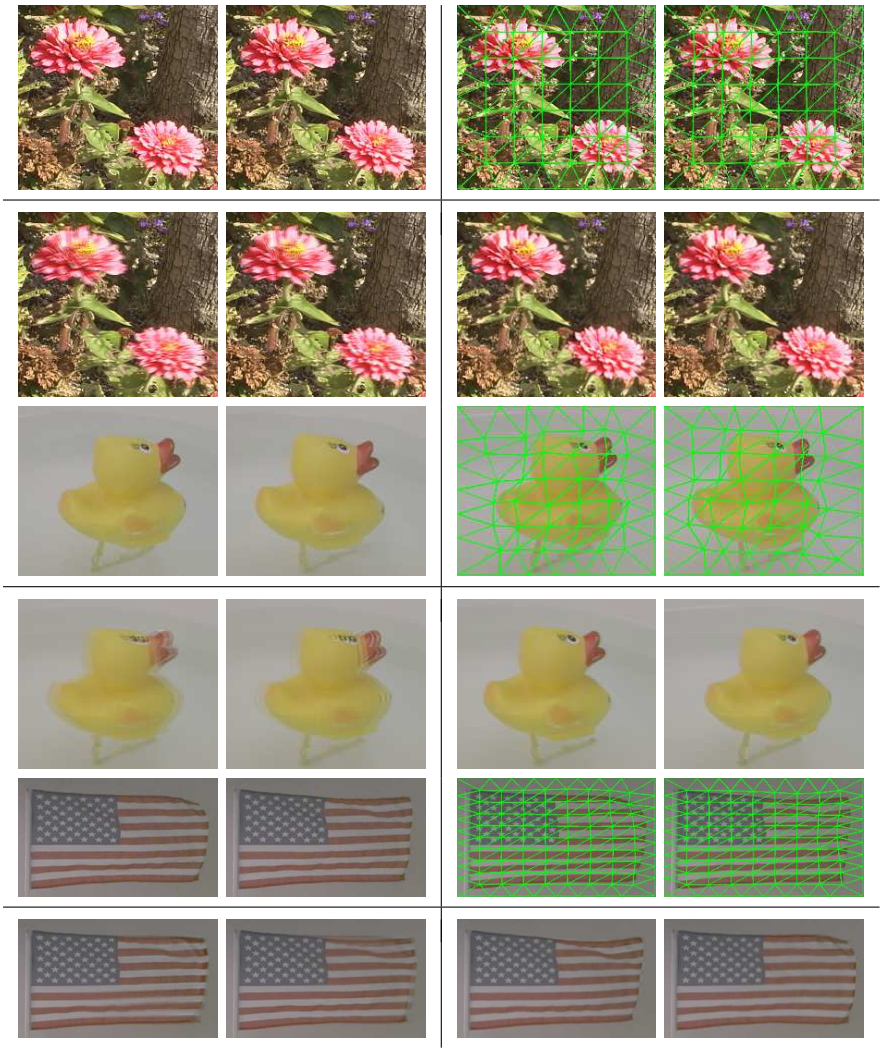

Fig. 1.8. **Flowers, Duck, Flag.** For each sequence: Original frames (top left), original frames with estimated shape $w_t$ (top right), frames synthesized with the LDT model (bottom left), frames synthesized with the DSA model (bottom right) (from Doretto,[11] © 2005 IEEE).

(top left), the same frames with the triangulated mesh representing the estimated shape $w_t$ (top right), frames synthesized with the LDT model (bottom left), and frames synthesized with the DSA model (bottom right).

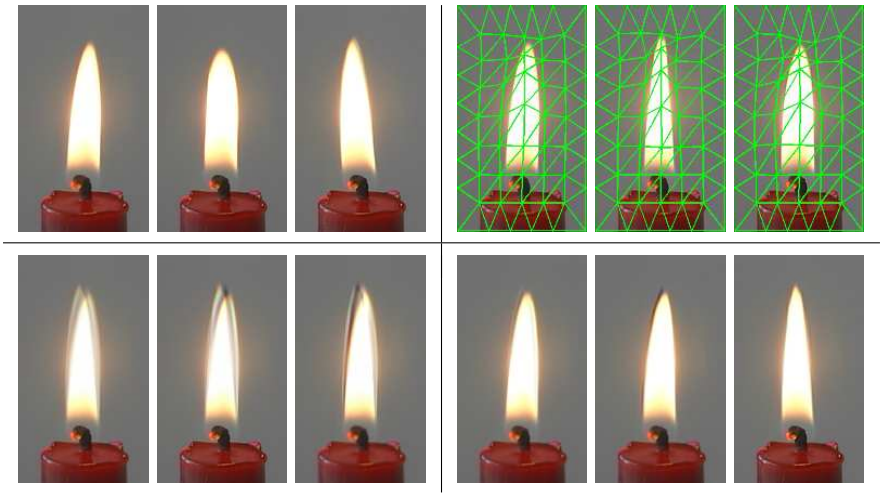

Fig. 1.9. **Candle:** Original frames (top left), original frames with the estimated shape $w_t$ (top right), frames synthesized with the LDT model (bottom left), frames synthesized with the DSA model (bottom right) (from Doretto,[11] © 2005 IEEE).

Even if the reconstruction errors of the two models are comparable, the simulation reveals that the DSA model outperforms the simpler LDT model. This is true especially when a video sequence contains moving objects with defined structure and sharp edges, suggesting that the DSA model can capture the higher-order temporal statistical properties of a video sequence.

The fact that the DSA model has superior generative power and less complexity of the LDT model does not come for free. In fact, the two models have a different algorithmic complexity with respect to learning. While the LDT model can be inferred very efficiently with a closed-form procedure that takes a few seconds to run,[4] the procedure highlighted in Section 1.5. typically requires a few dozens of iterations to converge, which translates into a couple of hours of processing on a high-end PC. The situation is different with respect to reconstruction or extrapolation. The DSA model, as well as the LDT model, is a parametric model with a per-frame simulation cost dominated by the generation of the appearance of an image $\rho_t$, which involves $O(pl)$ multiplications and additions. The LDT model has the same simulation cost, which can be higher if the state dimension $l_{LDT}$ is significantly higher than $l_{DSA}$. This complexity, as mentioned before, enables real time simulation.

## 1.10. Discussion

This chapter, which draws on a series of works published recently,[4,8,11,12,39,40] illustrates a model for portions of image sequences where shape, motion and appearance can be represented by conditionally linear models, and illustrates how this model can specialize into linear dynamic texture models, or models that account for view-point variation. We have seen how the linear dynamic texture model have proven successful at capturing the phenomenology of some very complex physical processes, such as water, smoke, fire etc., indicating that such models may be sufficient to support detection and recognition tasks and, to a certain extent, even synthesis and animation.[13] We have also seen that the general dynamic shape and appearance model can be used to model large enough regions of the image (in fact, the entire image), including significant changes in shape (e.g. a waving flag), motion (e.g. a floating duck), and appearance (e.g. a flame). This model can be thought of as extending the work on Active Appearance Models[6,28] to the temporal domain, or extending Dynamic Texture Models[4] to the spatial domain.

Eventually, this framework will be used to model segments of videos, which can be found by a segmentation procedure, which we have not addressed here. The interested reader can consult[3,31,32,41] for seed work in that direction, but significant work remains to be done in order to integrate the local models we describe into a more general modeling framework.

## Appendix A. Image formation model and assumptions

The goal of this appendix is to describe the conditions under which model (1.1) is valid. We start from a model that is standard in Computer Graphics: A collection of "objects" (closed, continuous but not necessarily smooth surfaces embedded in $\mathbb{R}^3$) $S_i$, $i = 1, \cdots, N_o$, the number of objects. Each surface is described relative to a Euclidean reference frame $g_i \in SE(3)$, a rigid motion in space,[34] which together with $S_i$, describes the *geometry* of the scene. In particular we call $g_i$ the *pose* of the $i$-th object, and $S_i$ its *shape*. Objects interact with light in ways that depend upon their material properties. We make the assumption that the light leaving a point $p \in S_i$ towards any direction depends solely on the incoming light directed towards $p$: Then each point $p$ on $S_i$ has associated with it a function $\beta_i : \mathbb{H}^2 \times \mathbb{H}^2 \to \mathbb{R}_+$; $(v, l) \mapsto \beta_i(v, l)$ that determines the portion of light coming from a direction $l$ that is reflected in the direction $v$, each of them represented as a point on the hemisphere $\mathbb{H}^2$ centered at the point $p$. This *bidirectional reflectance distribution function* (BRDF), describes the reflective properties

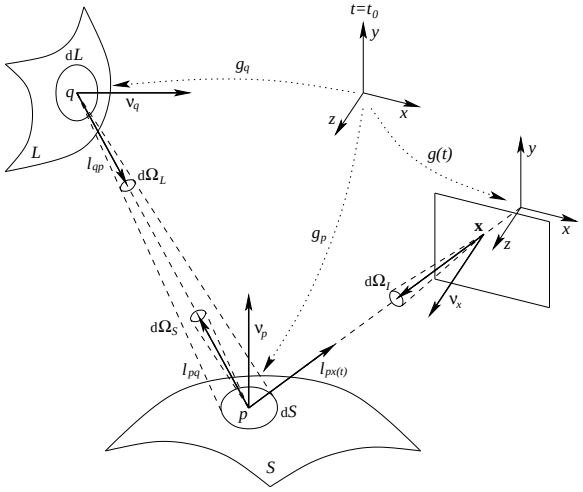


Fig. A.1. Geometric relation between light source, shape of the scene, and camera view point (from Doretto and Soatto,[12] © 2006 IEEE).

of the materials, neglecting diffraction, absorbtion, subsurface scattering and other aberrations. The *light source* is the collection of objects that can radiate energy, i.e. the scene itself, $L = \bigcup_{i=1}^{N_o} S_i$. The light element $\mathrm{d}E(q, l)$ accounts for light radiated by $q \in L$ in a direction $l \in \mathbb{H}^2$. $\mathrm{d}E$ can be described by a distribution on $L \times \mathbb{H}^2$ with values in $\mathbb{R}_+$. It depends on the properties of the light source that are described by its *radiance*. The collection $\beta_i : \mathbb{H}^2 \times \mathbb{H}^2 \to \mathbb{R}_+,\ i = 1, \cdots, N_o$ , and $\mathrm{d}E : L \times \mathbb{H}^2 \to \mathbb{R}_+$ describes the *photometry* of the scene (reflectance and illumination).

In principle, we would want to allow $S_i$, $g_i$, $\beta_i$, and $\mathrm{d}E$ to be functions of time. In practice, instead of allowing the surface $S_i$ to deform arbitrarily in time according to $S_i(t)$, and moving rigidly in space via $g_i(t) \in SE(3)$, we lump the pose $g_i(t)$ into $S_i$ and, without loss of generality, describe the surface in the fixed reference frame via $S_i(t)$. Therefore, we use $S_i = S_i(t)$, $\beta(v, l) = \beta(v, l, t)$, and $\mathrm{d}E(q, l) = \mathrm{d}E(q, l, t)$, $t = 1, \cdots, \tau$, to describe the *dynamics* of the scene. Now that we have defined geometry, photometry, and dynamics of the scene, we want to establish how they are related to the measured images.

As it is customary in computer vision, we make the assumption that the set of objects that act as light sources and those that act as light sinks are disjoint, i.e. we ignore inter-reflections. This means that we can divide the objects in two groups, the light source $L = \bigcup_{i=1}^{N_L} S_i$, and the shape $S = \bigcup_{i=N_L+1}^{N_o} S_i$ with its corresponding BRDF $\beta = \bigcup_{i=N_L+1}^{N_o} \beta_i$, where

$S \cap L = \emptyset$. Note that $S$ needs not be simply connected. We can also choose, as fixed reference frame, the one corresponding to the position and orientation of the viewer at the initial time instant $t_0$, and describe the position and orientation of the camera at time $t$, relative to the camera at time $t_0$, using a moving Euclidean reference frame $g(t) \in SE(3)$.

The image $I(\mathbf{x}, t)$ is obtained by summing the energy coming from the scene:

$$\begin{cases} I(\mathbf{x}, t) = \int_{L(t)} \beta(g_p x, g_p q, t) \langle \nu_p, l_{pq} \rangle \, \mathrm{d}E(q, g_q p, t) \\ \mathbf{x} = \pi(g(t)p) \ , \ p \in S(t) \end{cases} \tag{A.1}$$

where $q \in L(t)$, and $x$ is a point in the three-dimensional Euclidean space that corresponds to the position of the pixel $\mathbf{x}$ (see Figure A.1.). The quantities $g_p x$, $g_p q$, $g_q p \in \mathbb{H}^2$, represent unit vectors that indicate the directions from $p$ to $x$, from $p$ to $q$, and from $q$ to $p$ respectively. The unit vectors $\nu_p, l_{pq} \in \mathbb{H}^2$, represent the outward normal vector of $S(t)$ at point $p$, and the direction from $p$ to $q$; $\pi : \mathbb{R}^3 \to \mathbb{R}^2$ denotes the *standard (or "canonical") perspective projection*, which consists in scaling the coordinates of $p$ in the reference frame $g(t)$ by its depth, which naturally depends on $S(t)$.

Note that Equation (A.1) does not take into account the visibility of the viewer, and the light source. In fact, one should add to the equation two characteristic function terms: $\chi_v(x, t)$ outside the integral, which models the visibility of the scene from the pixel $\mathbf{x}$, and $\chi_s(p, q)$ inside the integral to model the visibility of the light source from a scene point (cast shadow). We are omitting these terms here for simplicity, and assume that there are no self-occlusions.

The image formation model (A.1), although derived with some approximations, is still an overkill because the variability in the image can be attributed to different factors. In particular, there is an ambiguity between reflectance and illumination if we allow either one to change arbitrarily, since only their convolution, or *radiance* affects the images. In this case, one can assume without loss of generality that the scene is Lambertian, or even self-luminous, and model deviations from the model as temporal changes in albedo, or radiance. Therefore, we can forego modeling illumination altogether and concentrate on modeling radiance directly. More precisely we have $\beta(v, l) \doteq \frac{\rho_a(p)}{\pi}$ , $p \in S$, where $\rho_a(p) : \mathbb{R}^3 \to \mathbb{R}_+$ is a scalar function called *surface albedo*, which is the percentage of incident irradiance reflected in any direction. Therefore, the first equation in (A.1) becomes $I(\mathbf{x}, t) = \frac{\rho_a(p,t)}{\pi} \int_{L(t)} \langle \nu_p, l_{pq} \rangle \, \mathrm{d}E(q, g_q p, t)$. With constant illumination we have $L(t) = L$ and $\mathrm{d}E(q, g_q p, t) = \mathrm{d}E(q, g_q p)$. A good approximation of the concept of *ambient light* can be produced through large sources that have diffusers whose purpose is to scatter light in all directions, which, in turn, gets reflected by the surfaces of the scene (inter-reflection). Instead

of modeling such a complicated situation, we can look at the desired effect of the sources: to achieve a uniform light level in the scene. Therefore, as a further simplification, we postulate an ambient light intensity which is the same at each point in the environment. This hypothesis corresponds to saying that every surface point $p \in S(t)$ receives the same irradiance from every possible direction. In formulas, this means that the integral in the Lambertian model becomes a constant $E_0$, i.e. $\int_L \langle \nu_p, l_{pq} \rangle \, \mathrm{d}E(q, g_q p) \doteq E_0$. By setting $\rho_I(p) \doteq E_0 \rho_a(p)/\pi$, we obtain the following reduced image formation model

$$\begin{cases} I(\mathbf{x}, t) = \rho_I(p, t) \\ \mathbf{x} = \pi(g(t)p) \ , \ p \in S(t) \ . \end{cases} \tag{A.2}$$

In addition to the reflectance/illumination ambiguity, which is resolved by modeling their product, i.e. radiance, there is an ambiguity between shape and motion. First, we parameterize $S$: a point $p \in S$ can be expressed, using a slight abuse of notation, by a parametric function $S(t) : B \subset \mathbb{R}^2 \to \mathbb{R}^3$; $u \mapsto S(u, t)$. This parametrization could be learned during the inference process according to a certain optimality property, like we do in Section 1.5.. For now, we can choose a parametrization induced by the image plane $\Omega \subset \mathbb{R}^2$ at a certain instant of time $t_0$, where a point $p \in S(t)$ is related to a pixel position $\mathbf{x}_0 \in \Omega$ according to $\mathbf{x}_0 \doteq \pi(p)$, and the parametric function representing the shape is given by $S(t) : \Omega \to \mathbb{R}^3$; $\mathbf{x}_0 \mapsto S(\mathbf{x}_0, t)$. With this assumption the second equation in model (A.2) becomes $\mathbf{x} = \pi(g(t)S(\mathbf{x}_0, t))$. This equation highlights an ambiguity between shape $S(t)$ and motion $g(t)$. More precisely, *the motion of the point* $\mathbf{x}$ *in the image plane at time* $t$ *could be attributed to the motion of the camera, or to the shape deformation.* Unfortunately we have access only to their composition. For this reason we lump these two quantities into one that we call $w(t) : \Omega \to \Omega$; $\mathbf{x}_0 \mapsto w(\mathbf{x}_0, t) \doteq \pi(g(t)S(\mathbf{x}_0, t))$.

The parametrization of the shape induces a parametrization of the irradiating albedo, which can be expressed as $\rho_I(S(\mathbf{x}_0, t), t) = I(\mathbf{x}, t)$. This equation highlights another ambiguity between irradiating albedo $\rho_I(t)$ and shape $S(t)$. In particular, *the variability of the value of the pixel at position* $\mathbf{x}$ *could be attributed to the variability of the irradiating albedo, or to the shape deformation.* Again, since we measure only their composition, we lump these two quantities into one, and define $\rho(t) : \Omega \to \mathbb{R}_+$; $\mathbf{x}_0 \mapsto \rho(\mathbf{x}_0, t) \doteq \rho_I(S(\mathbf{x}_0, t), t)$. Note that the domain of $\rho(t)$ is $\Omega \subset \mathbb{R}^2$, while the domain of $\rho_I(t)$ is $S(t) \subset \mathbb{R}^3$. Finally, we rewrite model (A.2) in the form that we use throughout the chapter[n]:

$$\begin{cases} I(\mathbf{x}(t), t) = \rho(\mathbf{x}_0, t) \ , \ \mathbf{x}_0 \in \Omega \subset \mathbb{R}^2 \\ \mathbf{x}(t) = w(\mathbf{x}_0, t) \ , \qquad t = 1, \dots, \tau \end{cases} \tag{A.3}$$

[n]In order to lighten the notation, in the body of the chapter the time variable will appear as a subscript, and pixel positions will not be in boldface.

If we think of an image $I(t)$ as a function defined on a domain $\Omega$ and with a range $\mathbb{R}_+$, the model states that shape and motion are warped together to model the domain deformation $w(t)$, while shape and irradiating albedo are merged together to model the range deformation $\rho(t)$. We will refer to these two quantities as the *shape*[o] and *appearance* of the image $I(t)$, respectively.

We conclude by making explicit one last assumption, that becomes obvious once we try to put together shape (or warping) $w(t)$, and appearance $\rho(t)$ to generate an image $I(t)$. In fact, to perform this operation we need the warping to be invertible. More precisely, it has to be a homeomorphism. This ensures that the spaces $\Omega$ and $w(\Omega, t)$ are topologically equivalent, so the scene does not get crinkled, or folded by the warping. This condition is verified if the shape $S(t)$ is smooth with no self-occlusions. Therefore, in (A.3) the temporal variation due to occlusions is not modeled by variations of the shape (warping), but by variations of the appearance.

## Acknowledgements

This work is supported by NSF grant EECS-0622245.

## Bibliography

1. H. Jin, S. Soatto, and A. J. Yezzi, Multi-view stereo reconstruction of dense shape and complex appearance, *International Journal of Computer Vision*. **63**(3), 175–189, (2005).
2. J. Rissanen, Modeling by shortest data description, *Automatica*. **14**, 465–471, (1978).
3. G. Doretto, E. Jones, and S. Soatto. Spatially homogeneous dynamic textures. In *Proceedings of European Conference on Computer Vision*, vol. 2, pp. 591–602, (2004).
4. G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto, Dynamic textures, *International Journal of Computer Vision*. **51**(2), 91–109, (2003).
5. M. Turk and A. Pentland, Eigenfaces for recognition, *Journal of Cognitive Neuroscience*. **3**(1), 71–86, (1991).
6. T. F. Cootes, G. J. Edwards, and C. J. Taylor, Active appearance models, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **23**(6), 681–685, (2001).
7. T. K. Carne, The geometry of shape spaces, *Proceedings of the London Mathematical Society*. **3**(61), 407–432, (1990).
8. S. Soatto, G. Doretto, and Y. N. Wu. Dynamic textures. In *Proceedings of*

[o]Note that this concept of shape does not have to be confused with the concept of three-dimensional shape $S(t)$ that we have introduced at the beginning of the appendix.

*IEEE International Conference on Computer Vision*, vol. 2, pp. 439–446, (2001).
9. T. Vetter and T. Poggio, Linear object classes and image synthesis from a single example image, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **19**(7), 733–742, (1997).
10. B. Schölkopf and A. Smola, *Learning with kernels: SVM, regularization, optimization, and beyond*. (The MIT press, 2002).
11. G. Doretto. Modeling dynamic scenes with active appearance. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 66–73, (2005).
12. G. Doretto and S. Soatto, Dynamic shape and appearance models, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **28**(12), 2006–2019, (2006). ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2006.243.
13. G. Doretto and S. Soatto. Editable dynamic textures. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 137–142, (2003).
14. P. Saisan, G. Doretto, Y. N. Wu, and S. Soatto. Dynamic texture recognition. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 58–63, (2001).
15. B. Julesz, Visual pattern discrimination, *IEEE Transactions on Information Theory*. **8**(2), 84–92, (1962).
16. J. Portilla and E. Simoncelli, A parametric texture model based on joint statistics of complex wavelet coefficients, *International Journal of Computer Vision*. **40**(1), 49–71, (2000).
17. R. C. Nelson and R. Polana, Qualitative recognition of motion using temporal texture, *Computer Vision, Graphics, and Image Processing: Image Understanding*. **56**(1), 78–89, (1992).
18. M. Szummer and R. W. Picard. Temporal texture modeling. In *Proceedings of IEEE International Conference on Image Processing*, vol. 3, pp. 823–826, (1996).
19. Z. Bar-Joseph, R. El-Yaniv, D. Lischinski, and M. Werman, Texture mixing and texture movie synthesis using statistical learning, *IEEE Transactions on Visualization and Computer Graphics*. **7**(2), 120–135, (2001).
20. A. Fitzgibbon. Stochastic rigidity: image registration for nowhere-static scenes. In *Proceedings of IEEE International Conference on Computer Vision*, vol. 1, pp. 662–669, (2001).
21. Y. Z. Wang and S. C. Zhu. A generative method for textured motion: analysis and synthesis. In *Proceedings of European Conference on Computer Vision*, pp. 583–598, (2002).
22. Y. Z. Wang and S. C. Zhu. Modeling complex motion by tracking and editing hidden Markov graphs. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 856–863, (2004).
23. L. Yuan, F. Wen, C. Liu, and H. Y. Shum. Synthesizing dynamic texture with closed-loop linear dynamic systems. In *Proceedings of European Conference on Computer Vision*, vol. 2, pp. 603–616, (2004).

24. A. Schödl, R. Szeliski, D. H. Salesin, and I. Essa. Video textures. In *Proceedings of SIGGRAPH*, pp. 489–498, (2000).
25. L. Y. Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of SIGGRAPH*, pp. 479–488, (2000).
26. V. Kwatra, A. Schödl, I. Essa, G. Turk, and B. A. F. Graphcut textures: image and video synthesis using graph cuts. In *Proceedings of SIGGRAPH*, pp. 277–286, (2003).
27. K. S. Bhat, S. M. Seitz, J. K. Hodgins, and P. K. Khosla. Flow-based video synthesis and editing. In *Proceedings of SIGGRAPH*, pp. 360–363, (2004).
28. S. Baker, I. Matthews, and J. Schneider, Automatic construction of active appearance models as an image coding problem, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **26**(10), 1380–1384, (2004).
29. T. Cootes, S. Marsland, C. Twining, K. Smith, and C. Taylor. Groupwise diffeomorphic non-rigid registration for automatic model building. In *Proceedings of European Conference on Computer Vision*, pp. 316–327, (2004).
30. N. Campbell, C. Dalton, D. Gibson, and B. Thomas. Practical generation of video textures using the autoregressive process. In *Proceedings of the British Machine Vision Conference*, pp. 434–443, (2002).
31. J. Y. A. Wang and E. H. Adelson, Representing moving images with layers, *IEEE Transactions on Image Processing*. **3**(5), 625–638, (1994). ISSN 1057-7149. doi: 10.1109/83.334981.
32. J. D. Jackson, A. J. Yezzi, and S. Soatto. Dynamic shape and appearance modeling via moving and deforming layers. In *Proceedings of the Workshop on Energy Minimization in Computer Vision and Pattern Recognition (EMMCVPR)*, pp. 427–448, (2005).
33. P. Van Overschee and B. De Moor, Subspace algorithms for the stochastic identification problem, *Automatica*. **29**(3), 649–660, (1993).
34. Y. Ma, S. Soatto, J. Kosecká, and S. S. Sastry, *An invitation to 3D vision: from images to geometric models*. (Springer-Verlang New York, Inc., 2004).
35. T. J. R. Hughes, *The Finite Element Method - linear static and dynamic finite element analysis*. (Dover Publications, Inc., 2000).
36. L. Ljung, *System identification: theory for the user*. (Prentice-Hall, Inc., 1999), 2nd edition.
37. T. Kailath, *Linear systems*. (Prentice Hall, Inc., 1980).
38. G. H. Golub and C. F. Van Loan, *Matrix computations*. (The Johns Hopkins University Press, 1996), 3rd edition.
39. G. Doretto. *DYNAMIC TEXTURES: modeling, learning, synthesis, animation, segmentation, and recognition*. PhD thesis, University of California, Los Angeles, CA (March, 2005).
40. G. Doretto and S. Soatto. Towards plenoptic dynamic textures. In *Proceedings of the 3rd International Workshop on Texture Analysis and Synthesis*, pp. 25–30, (2003).
41. G. Doretto, D. Cremers, P. Favaro, and S. Soatto. Dynamic texture segmentation. In *Proceedings of IEEE International Conference on Computer Vision*, vol. 2, pp. 1236–1242, (2003).