Modeling Dynamic Textures Using Subspace Mixtures

Che-Bin Liu†, Ruei-sung Lin†, and Narendra Ahuja‡
†Dept. of Computer Science, ‡Dept. of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign, Urbana, IL, USA

Abstract

In this paper, we aim at modeling video sequences that exhibit temporal appearance variation. The dynamic texture model proposed in [6] is effective to model simple dynamic scenes. However, because of its over-simplified appearance model and under-constrained dynamics model, the visual quality of its synthesized video sequences is often not satisfactory. This leads to our new model. We parameterize the nonlinear image manifold using mixtures of probabilistic principal component analyzers. We then align coefficients from different mixture components in a global coordinate system, and model the image dynamics in the global coordinate using an autoregressive process. The experimental results show that our method is capable of capturing complex temporal appearance variation and offers improved synthesis results over previous works.

1. Introduction

Examples of dynamic textures/scenes include flowing river, boiling water, waving trees, shifting smoke, etc. There are three main approaches in literature to model dynamic textures. First, one can directly extend methods of static 2D texture synthesis, e.g. [12]. These methods ignore the underlying texture dynamics. Also, they cannot synthesize dynamic scenes with more than one texture, or non-texture objects. Second, one can use a spatio-temporal model at the pixel level to represent the relationships between a pixel and its neighborhoods [8]. Such pixel-level dynamical models experience difficulties in selecting the appropriate neighborhood size and topology. A good model of this approach also requires a large number of model parameters. Most importantly, such a model is not capable of synthesizing rotation-like motion patterns. Third, one can use a dynamical model at the image level. For example, Soatto et al [6] model dynamic textures/scenes using a linear dynamical system (LDS), which represents each image as a point in a linear subspace (e.g. PCA) and uses an autoregressive model to learn the dynamics of the trajectory in the image subspace as

$$\begin{cases} x_t = Ax_{t-1} + v_t, & v_t \sim \mathcal{N}(0, Q) \\ y_t = Cx_t + w_t, & w_t \sim \mathcal{N}(0, R) \end{cases} \quad (1)$$

where y is the observed image, x is the hidden state variable, C is the output matrix mapping observations to state variables, A is the transition matrix of AR process, and v and w are zero-mean Guassian noise sources. Compared to the pixel-level dynamical model, this approach requires much fewer model parameters and has a greater capability of capturing different motion types. Furthermore, Yuan et al [13] analyze the stability of the LDS through its pole placement and propose a dynamical model with feedback control, which improves synthesis results.

However, by using either [6] or [13], the visual quality of synthesized video sequences is not satisfactory when the scenes contain large temporal appearance variation and/or shape variation. In [13], they address the problem of under-constrained dynamics in [6]. But one key to modeling complex texture motion is a better appearance model. As will be shown in our experiments, a linear dimensionality reduction scheme such as PCA is too simple to capture complex appearance changes.

In this paper, we address problems in both appearance and dynamics models. In appearance model, we propose to use a mixture of PCAs to characterize image manifolds which is usually nonlinear for images with large appearance variation. Using a mixture model for appearance manifold seems intuitive, but then deriving a dynamics model could be very difficult. This is because different mixture components have their own coordinate systems based on their eigenvectors spanning the subspace, but dynamics models have to be operated on a single coordinate system of the manifold [1]. Therefore, we adopt a global coordination model, which provides a mapping between coordinates of different mixture components and the global manifold coordinate, and extend it to a dynamic model.

[1] A switching LDS [1] does not operate on a single coordinate system, but it does not ensure appearance continuity.

0-7803-9332-5/05/$20.00 ©2005 IEEE

2. Global Parameterization of Appearance Manifold

There are generally two categories of nonlinear dimensionality reduction schemes: (1) locally linear mapping [11] and (2) nonlinear embedding [3, 10]. A mixture of locally linear models offers a two-way mapping, but lacks a coherent global coordinate system. Nonlinear embedding offers a global coordinate, but lacks the mapping for the inference of an observation from a global coordinate. Therefore, we map a mixture of locally linear models into a new coordinate system to achieve ideal manifold mapping for modeling dynamic textures.

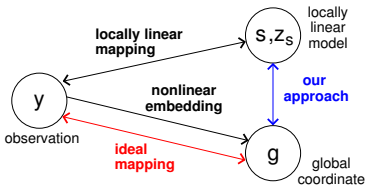

Figure 1: Mappings of nonlinear dimensionality reduction schemes.

2.1. Global Coordination Model

Roweis [4] propose a global coordination model that maps a mixture of locally linear models into a global coordinate system (Figure 2). Given a local model s and its local coordinate z_s, the mappings from z_s to observation y and to global coordinate g are both linear. And since s and z_s are unknown, mapping between y and g is nonlinear through the inference.

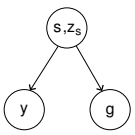

Figure 2: The global coordination model.

To learn this model, we use a post-coordination method proposed in [9]. Given a learned mixture of S PCA models, for each data point y_n, the s-th PCA has an d dimensional internal coordinate z_{ns} for y_n and an associated responsibility r_{ns}, where $r_{ns} = P(y_n|s)$ and $\sum_s r_{ns} = 1$. We assume there is a linear mapping between local representations and global coordinates, with linear projection L_s and mean l_s^0. The global coordinates g_n is defined as the weighted sum of the projections by each local model:

$$\begin{aligned} g_n &= \sum_s r_{ns} g_{ns} = \sum_s r_{ns}(L_s z_{ns} + l_s^0) \\ &= \sum_s \sum_{i=0}^{d} r_{ns} z_{ns}^i l_s^i = \sum_j u_{nj} l_j, \end{aligned} \tag{2}$$

$$G = UL \qquad j = j(i,s), \quad u_{nj} = r_{ns} z_{ns}^i, \quad l_j = l_s^i \tag{3}$$

where l_s^i is the i-th column of L_s, z_{ns}^i is the i-th entry of z_{ns}, and $z_{ns}^0 = 1$. After vectorizing index pair (i,s) into a single index j and defining matrix U as u_{nj} and j-th row of L as $l_j = l_s^i$, we have a linear equation system (3) with fixed U and unknown L.

To determine L, we need to minimize a cost function that incorporates the topological constraints that govern g_n. Hence, the cost function is selected based on LLE's idea [3]: preserving the same neighborhood structure between the high dimensional input space and the low dimensional embedding. For each data point y_n, we denote its nearest neighbors as y_m ($m \in \mathcal{N}_n$) and minimize

$$\mathcal{E}(Y,W) = \sum_n \parallel y_n - \sum_{m \in \mathcal{N}_n} w_{nm} y_m \parallel^2 \tag{4}$$

with respect to W subject to $\sum_{m \in \mathcal{N}_n} w_{nm} = 1$. The weights w_{nm} are unique and can be solved by constrained least squares. These weights represent the locally linear relationships between y_n and its neighbors. Accordingly, we define the same cost function

$$\begin{aligned} \mathcal{E}(G,W) &= \sum_n \parallel g_n - \sum_{m \in \mathcal{N}_n} w_{nm} g_m \parallel^2 \\ &= \text{trace}(G^T(I-W^T)(I-W)G) \\ &= \text{trace}(L^T A L) \end{aligned} \tag{5}$$

with respect to G, where $A = U^T(I-W^T)(I-W)U$. Since $\mathcal{E}$ is invariant to translations and rotations of G, and $\mathcal{E}$ scales as G is scaled, we define the following two constraints

$$\frac{1}{N}\sum_n g_n = \frac{1}{N}\vec{1}^T G = \frac{1}{N}\vec{1}^T UL = 0 \tag{6}$$

and

$$\frac{1}{N}\sum_n g_n g_n^T = \frac{1}{N} G^T G = L^T B L = I_d, \tag{7}$$

where $B = \frac{1}{N}U^T U$. Now that the cost function (5) and the constraint (7) are both quadratic, we can determine the optimal L, without local minima problems, by solving generalized eigenvalue system $Av = \lambda Bv$ subject to $\frac{1}{N}\vec{1}^T UL = 0$. The solution for L is the matrix with its columns formed by the second to $(d+1)$-th smallest generalized eigenvectors.

3. Globally Coordinated Dynamic Network

Now that we have the nonlinear mapping between images and their low dimensional global coordinates, we can then model the image dynamics in the low dimensional space (global coordinates). Here we adopt the Markovian assumption, $P(g_t|g_{t-1}, \ldots, g_1) = P(g_t|g_{t-1})$. With this property, our dynamic texture model is a generative model depicted in Figure 3.

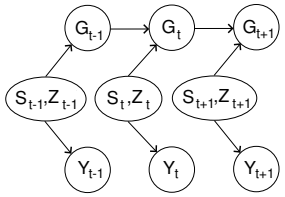

Figure 3: Our generative model for dynamic textures.

3.1. Autoregressive Process

We can treat the image dynamics as a realization of a stochastic process estimated by an autoregressive (AR) model. The AR model is used based on the assumption that each term in the time series depends linearly on several previous terms [2]. Therefore, the AR model of order p, denoted as AR(p), for dynamic textures is expressed as

$$g_t = \sum_{i=1}^{p} A_i g_{t-i} + v_t, \tag{8}$$

where the matrices A_i are the coefficient matrices of the AR(p) model, and v_t is an uncorrelated random noise. To select the optimum order of the AR model, we adopt Schwarz's Bayesian Criterion [5] which chooses the order of the model so as to minimize the forecast mean-squared error.

As Yuan et al [13] point out, the LDS based method produces good-quality dynamic textures only if it is an oscillatory system. That is, for the AR(1) model, for all eigenvalues σ_i of A, $|\sigma_i| \leq 1$ and there exists j such that $|\sigma_j| = 1$. Otherwise, the synthesized dynamic textures will gradually decay or diverge. To overcome this problem, they incorporate feedback control that results in a non-causal system. Therefore, they first need to generate reference states, and then iteratively smooth out the discontinuity. Although using this method one will obtain better results, it does not predict new states on the fly, which is a desirable feature for many real-world applications (e.g. video games).

To prevent g_t generated by AR model from drifting away, we sample a certain number of v_t and pick the one that pulls g_t toward the manifold. Our experimental results show that this method ensures a stable dynamic texture in a long synthesized image sequence.

4. Experimental Results

The image sequences used in our experiments are taken from MIT temporal texture database [7]. Most image sequences in the database have resolutions of 170 by 115 and contain 120 to 150 frames. We train mixture models with the method proposed by Tipping [11].

4.1. Dynamic Texture Reconstruction

For the application in video compression, our mixture of PCA method always yields less reconstruction errors than the single PCA method in different dynamic texture sequences. While the difference of reconstruction errors between two methods are usually about 5%, the differences of the visual quality of reconstructed images are always very obvious (See Figure 4).

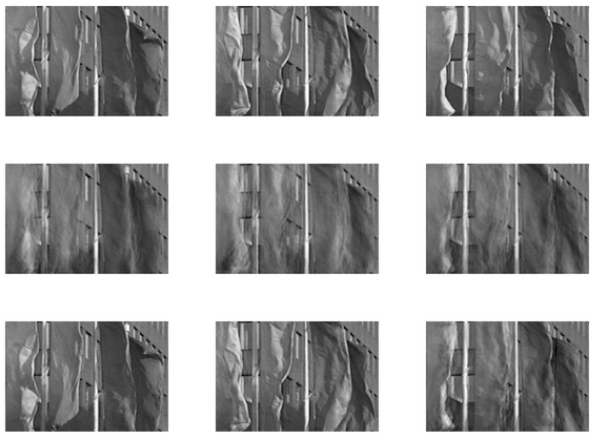

Figure 4: The images on the top row are from the original sequence. The middle row is reconstructed by single PCA method. The bottom row is reconstructed by our method with a mixture of three PCA models.

4.2. Dynamic Texture Synthesis

For the application in synthesis, we demonstrate a river sequence that would allow temporally continuous and infinitely synthesized images. Note that some original sequences do not show repeated patterns, so it is impossible to generate infinite synthesized sequences.

Figure 5 shows the synthesis results by the single PCA method and our proposed method. The images are corresponding frames selected from 200 synthesized images. As can be seen, the single PCA method yields

a decreasing quality as the synthesized sequence becomes longer. Our method generates a much improved result. Our synthesis process is performed in real-time.

5. Conclusion

In this paper, we model dynamic textures with mixtures of locally linear subspaces. We adopt a global coordination model that provides a coherent coordinate mapping between images and their low-dimensional appearance embedding. We then model the texture dynamics on the appearance embedding. Compared to the relevant works, our method yields less reconstruction error and generates higher-quality dynamic textures.

References

[1] Z. Ghahramani and G. E. Hinton. Variational learning for switching state-space models. *Neural Computation*, 12:831–864, 2000.

[2] H. Lütkepohl. *Introduction to Multiple Time Series Analysis*. Springer-Verlag, 1991.

[3] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, Dec 2000.

[4] S. Roweis, L. Saul, and G. E. Hinton. Global coordination of local linear models. In *Neural Information Processing Systems*, volume 14, pages 889–896, 2001.

[5] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

[6] S. Soatto, G. Doretto, and Y. Wu. Dynamic textures. In *IEEE International Conference on Computer Vision*, volume 3, pages 439–446, 2001.

[7] M. Szummer. MIT temporal texture database.

[8] M. Szummer and R. W. Picard. Temporal texture modeling. In *IEEE International Conference on Image Processing*, volume 3, pages 823–826, 1996.

[9] Y. W. Teh and S. Roweis. Automatic alignment of local representations. In *Neural Information Processing Systems*, volume 15, pages 841–848, 2002.

[10] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, Dec 2000.

[11] M. E. Tipping and C. M. Bishop. Mixtures of probabilistic principal component analysers. *Neural Computation*, 11(2):443–482, 1999.

[12] L.-Y. Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of ACM SIGGRAPH 2000*, pages 479–488, 2000.

[13] L. Yuan, F. Wen, C. Liu, and H.-Y. Shum. Synthesizing dynamic texture with closed-loop linear dynamic system. In *ECCV*, volume 9999, pages 0–0, 2004.

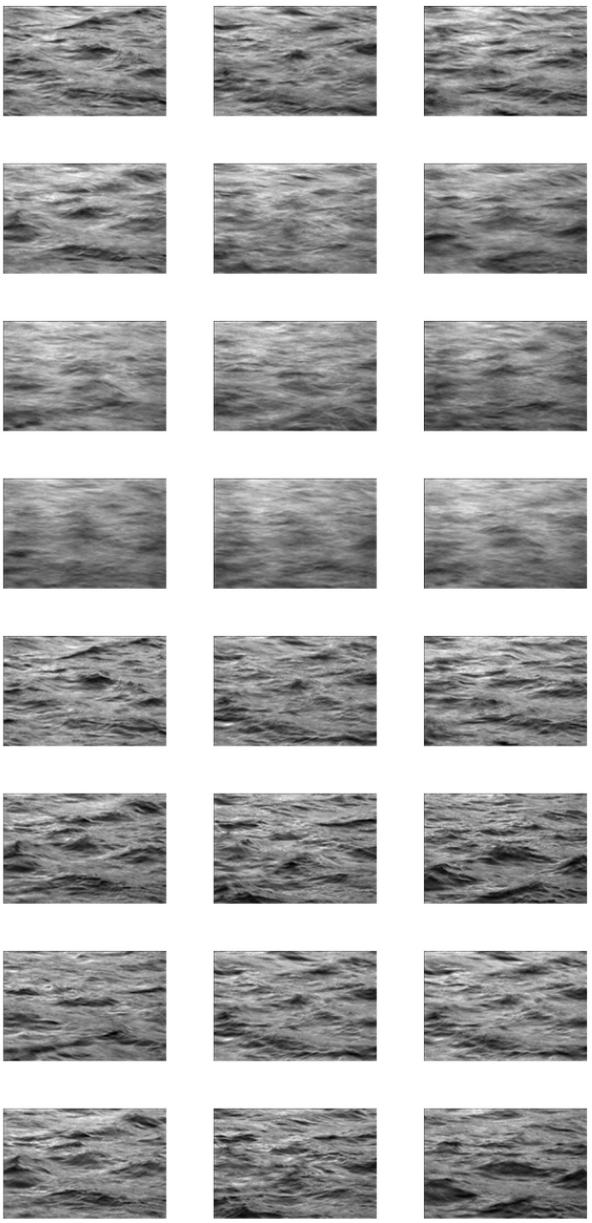

Figure 5: The top four rows are synthesized by single PCA method. The bottom four rows are synthesized by our method with a mixture of two PCA models. All PCA models for both synthesized sequences have a dimensionality of 15. An AR(1) model is used in each case.