Generating Video Textures by PPCA and Gaussian Process Dynamical Model

Wentao Fan and Nizar Bouguila

Institute for Information Systems Engineering
University of Concordia
Montreal, Canada
{wenta_fa,bouguila}@ciise.concordia.ca

Abstract. Video texture is a new type of medium which can provide a continuous, infinitely varying stream of video images from a recorded video clip. It can be synthesized by rearranging the order of frames based on the similarities between all pairs of frames. In this paper, we propose a new method for generating video textures by implementing probabilistic principal components analysis (PPCA) and Gaussian Process Dynamical model (GPDM). Compared to the original video texture technique, video texture synthesized by PPCA and GPDM has the following advantages: it might generate new video frames that have never existed in the input video clip before; the problem of "dead-end" is totally avoided; it could also provide video textures that are more robust to noise.

Keywords: Video texture, computer graphics, computer vision, dimensionality reduction, autoregressive process, Gaussian process, PPCA.

1 Introduction

Video textures, first introduced by Schödl *et al.* [1], is a new type of medium between static image and dynamic video. It can create a continuous, infinitely changing stream of images from a recorded video. Following the work of video texture, Schödl *et al.* also extended this technique on video sprites [2] [3]. Recently, a number of extensions and applications of video texture have emerged. Dong *et al.* [4] proposed a novel method of generating video texture based on wavelet coefficients which are computed from the decomposition of the pixel values of neighboring frames. In the work of Fitzgibbon [5], video texture is synthesized first by applying the principal components analysis (PCA) to obtain the signatures of each frame, then autoregressive process (AR) is used to predict new frames. In [6], Campbell *et al.* extended this approach to work with strongly non-linear sequences.

Our work is inspired from [5], where the author has shown that video texture may be created by implementing regression methods such as AR process which allow the prediction of new video frames. Accordingly, new video textures are obtained by appending synthesized frames. Gaussian process [7] [8] [9] is another approach which can be exploited to solve regression problems. Via Gaussian

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 801–808, 2009.

process, we can define probability distributions over functions directly, and a Gaussian process prior can be combined with a likelihood to acquire a posterior over functions. In our work, we adopt an extension of Gaussian process namely Gaussian process dynamical model (GPDM) [10] [11] which is a latent variable model that can be applied for nonlinear time series analysis. GPDM extended the Gaussian process latent variable model (GPLVM) [12] with a latent dynamical model. In GPDM, it includes a low-dimensional space account for dynamics in the time series data, as well as a mapping from the latent space to observation space. Since video sequence is a time series data, in principle, GPDM is a suitable method to synthesize new video textures. Fitzgibbon [5] has applied PCA as a dimensionality reduction technique to obtain the frames signatures. However, we have shown in a previous works [13] that probabilistic principal components analysis (PPCA) [14] is more robust to noise and provide better results. Thus, our video texture generation framework will be based on both PPCA and GPDMs.

The remainder of this paper is organized as follows. First we introduce Gaussian processes regression in Section 2. Then GPDM for video texture is discussed in Section 3. Section 4 is devoted to the experimental results. The conclusion and future work are included in Section 5.

2 Gaussian Processes Regression

A Gaussian process is defined as a probability distribution over some functions $y(\mathbf{x})$, such that the set of values of $y(\mathbf{x})$ evaluated at an arbitrary set of points $\mathbf{x}_1, ..., \mathbf{x}_N$ jointly have a Gaussian distribution. Here, we will illustrate how Gaussian process can be applied on general regression problems. We consider a model where the observed target values t_n are corrupted with some random noise

$$t_n = y_n + \epsilon_n \tag{1}$$

where $y_n = y(\mathbf{x}_n)$ for input data $\mathbf{x}$. ϵ_n is the random noise which has Gaussian distribution with zero mean and β^{-1} variance. Since the noise is independent for each data point, given the values of $\mathbf{y} = (y_1, ...y_N)^T$, the joint distribution of target values $\mathbf{t} = (t_1, ..., t_N)$ is an isotropic Gaussian

$$p(\mathbf{t}|\mathbf{y}) = \mathcal{N}(\mathbf{t}|\mathbf{y}, \beta^{-1}\mathbf{I}_N) \tag{2}$$

After obtaining the marginal distribution of $\mathbf{t}$, the next job is to evaluate the conditional distribution $p(t_{N+1}|\mathbf{t})$ where t_{N+1} is the next target value that we wish to predict. In order to find $p(t_{N+1}|\mathbf{t})$, we first need to find the joint distribution of $p(\mathbf{t}_{N+1})$ for $t_1, ..., t_{N+1}$

$$p(\mathbf{t}_{N+1}) = \mathcal{N}(\mathbf{t}_{N+1}|0, \mathbf{C}_{N+1}) \tag{3}$$

where $\mathbf{C}_{N+1}$ is an $(N+1) \times (N+1)$ covariance matrix. The covariance matrix $\mathbf{C}_{N+1}$ needs to be partitioned as

$$\mathbf{C}_{N+1} = \begin{pmatrix} \mathbf{C}_N & \mathbf{k} \\ \mathbf{k}^T & c \end{pmatrix} \tag{4}$$

where $\mathbf{C}_N$ is the $N \times N$ covariance matrix of the training data, vector $\mathbf{k}$ represents the $N \times 1$ covariance matrix of training data and the predictive target t_{N+1}, and the scalar c denotes the variance of t_{N+1}. As shown in [8], since the joint distribution $p(\mathbf{t}_{N+1})$ is also a Gaussian distribution, we can obtain the mean and covariance of the conditional distribution $p(t_{N+1}|\mathbf{t})$ as

$$m(\mathbf{x}_{N+1}) = \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{t} \tag{5}$$

$$\sigma^2(\mathbf{x}_{N+1}) = c - \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{k} \tag{6}$$

These results represent the core idea of Gaussian process regression. More details and discussion about Gaussian processes can be found in [7].

3 Gaussian Processes Dynamical Models

The Gaussian process dynamical model (GPDM) [11] is a latent variable model with two nonlinear mappings. One mapping is from the latent space to the observation space and the other is the dynamical mapping in the latent space. Suppose $\{\mathbf{y}_1, ..., \mathbf{y}_N\}$ denotes the D-dimensional observation data set and $\mathbf{y}_t$ represents a particular observation output at the specific time t, $\mathbf{y}_t \in \mathbb{R}^D$. $\mathbf{x}_1, ..., \mathbf{x}_N$ is a data set in the latent space, $\mathbf{x}_t$ represents the d-dimensional latent coordinate of the observation data at time index t, $\mathbf{x}_t \in \mathbb{R}^d$. The first-order Markov dynamics and the latent space mapping are given by

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}; \mathbf{A}) + \mathbf{n}_{x,t} \tag{7}$$

$$\mathbf{y}_t = g(\mathbf{x}_t; \mathbf{B}) + \mathbf{n}_{y,t} \tag{8}$$

here, the dynamical mapping function f is parameterized by $\mathbf{A}$ and latent space mapping function g is is parameterized by $\mathbf{B}$. $\mathbf{n}_{x,t}$ and $\mathbf{n}_{y,t}$ are zero-mean, isotropic, white Gaussian noise processes. Two basis functions ϕ_i and φ_j are used for f and g are given by

$$f(\mathbf{x}; \mathbf{A}) = \sum_i \mathbf{a}_i \phi_i(\mathbf{x}) \tag{9}$$

$$g(\mathbf{x}; \mathbf{B}) = \sum_j \mathbf{b}_j \varphi_i(\mathbf{x}) \tag{10}$$

where weights $\mathbf{A} \equiv [a_1, a_2, ...]^T$ and $\mathbf{B} \equiv [b_1, b_2, ...]^T$. f and g are nonlinear functions of $\mathbf{x}$, but the dependencies of f and g on the parameters $\mathbf{A}$ and $\mathbf{B}$ are linear. For the mapping from latent space to the observation space, after marginalizing over g, the joint distribution of $\mathbf{Y}$ can be represented as

$$p(\mathbf{Y}|\mathbf{X}, \bar{\beta}, \mathbf{W}) = \frac{|\mathbf{W}|^N}{\sqrt{(2\pi)^{ND}|\mathbf{K}_Y|^D}} \exp(-\frac{1}{2} tr(\mathbf{K}_Y^{-1} \mathbf{Y} \mathbf{W}^2 \mathbf{Y}^T)) \tag{11}$$

here, $\mathbf{K}_Y$ is the kernel matrix of the mapping g and $\bar{\beta}$ are the hyperparameters of the kernel. $\mathbf{W}$ represents the scale parameters which account for the overall scale in each data dimension. The elements of $\mathbf{K}_Y$ are defined by a kernel function $(K_Y)_{ij} \equiv \mathbf{k}_Y(\mathbf{x}_i, \mathbf{x}_j)$. We choose the radial basis function (RBF) as the kernel function for the latent mapping g

$$k_Y(\mathbf{x}, \mathbf{x}') = \beta_1 \exp(-\frac{\beta_2}{2}\|\mathbf{x} - \mathbf{x}'\|^2) + \beta_3^{-1}\delta_{\mathbf{x},\mathbf{x}'} \tag{12}$$

where the hyperparameter β_1 represents the output scale of the kernel function, β_2 represents the inverse width of the RBF, and β_3 gives the variance of the isotropic noise term $\mathbf{n}_{y,t}$. The dynamic mapping for latent coordinate is similar to the latent space mapping. The joint probability density over the latent coordinates can be represent as

$$p(\mathbf{X}|\bar{\alpha}) = \frac{p(\mathbf{x}_1)}{\sqrt{(2\pi)^{(N-1)d}|\mathbf{K}_X|^d}} \exp(-\frac{1}{2}tr(\mathbf{K}_X^{-1}\mathbf{X}_{2:N}\mathbf{X}_{2:N}^T)) \tag{13}$$

here, $\mathbf{X}_{2:N} = [\mathbf{x}_2, ...\mathbf{x}_N]^T$ denotes the input data that except the first element. $\mathbf{K}_X$ is the kernel matrix build from $[\mathbf{x}_1, ...\mathbf{x}_{N-1}]$. In this dynamic mapping, the form "RBF + linear" is defined for the kernel function

$$k_X(\mathbf{x}, \mathbf{x}') = \alpha_1 \exp(-\frac{\alpha_2}{2}\|\mathbf{x} - \mathbf{x}'\|^2) + \alpha_3 x^T x' + \alpha_4^{-1}\delta_{\mathbf{x},\mathbf{x}'} \tag{14}$$

In order to discourage overfitting, prior distributions are placed on hyperparameters $\bar{\alpha}$, $\bar{\beta}$ and $\mathbf{W}$.[1] Then a generative model for time-series observations can be obtained through a latent space mapping, a dynamic mapping and prior distributions:

$$p(\mathbf{X}, \mathbf{Y}, \bar{\alpha}, \bar{\beta}, \mathbf{W}) = p(\mathbf{Y}|\mathbf{X}, \bar{\beta}, \mathbf{W})p(\mathbf{X}|\bar{\alpha})p(\mathbf{W})p(\bar{\alpha})p(\bar{\beta}) \tag{15}$$

This represents the general form of the GPDM. Details of how to evaluate the parameters for GPDM can be found in [10].

4 Experimental Results

In our work, the goal is to apply GPDM to synthesize video textures. The performance of our approach is evaluated by comparing our results with the video textures generated by AR approach in [5]. In the AR approach for synthesizing video textures, frame signatures are first calculated by adopting the dimension reduction technique: principal components analysis (PCA), followed by the synthesis of new video textures using AR process. In our case, in order to test our approach under different scenarios, several input video clips are selected. First,

[1] $p(\bar{\alpha}) \propto \prod_i \alpha_i^{-1}$, $p(\bar{\beta}) \propto \prod_i \beta_i^{-1}$ and $p(\mathbf{W}) = \prod_{m=1}^{D} \frac{2}{k\sqrt{2\pi}} \exp(-\frac{w_m^2}{2k^2})$, where w_m are the variances that contain the elements of $\mathbf{W}$, and in practice, k is set to 10^3.

the input video clip is decomposed into a sequence of frames. Each individual frame is an input vector $\mathbf{x}$, with dimensionality D. The value of D is the number of pixels contained in each frame. Second, these input vectors are mean-subtracted and the latent coordinates are initialized with PPCA. Last, GPDM is applied to synthesize new video frames which are then composed together to generate a new video texture.

4.1 Generation of New Frames

As described above, new video frames are predicted using GPDM. In other words, it is to predict the next video frame $\mathbf{x}_{N+1}$ conditioned on the previous frame $\mathbf{x}_N$. The marginal distribution of the new frame $p(\mathbf{x}_{N+1})$ derived from the conditional distribution $p(\mathbf{x}_{N+1}|\mathbf{x}_N)$ is also a Gaussian distribution

$$\mathbf{x}_{N+1} \sim \mathcal{N}(\mu_X(\mathbf{x}_N); \sigma_X^2(\mathbf{x}_N)) \tag{16}$$

We can solve this prediction problem by applying the similar ideas as in Gaussian process regression. According to results in (5) and (6), the mean and covariance can be calculated as

$$\mu_X(\mathbf{x}) = \mathbf{X}_{2:N}^T \mathbf{K}_X^{-1} \mathbf{k}_X(\mathbf{x}) \tag{17}$$

$$\sigma_X^2(\mathbf{x}) = k_X(\mathbf{x}, \mathbf{x}) - \mathbf{k}_X(\mathbf{x})^T \mathbf{K}_X^{-1} \mathbf{k}_X(\mathbf{x}) \tag{18}$$

In the above equations, $\mathbf{k}_X(\mathbf{x})$ represents a vector that contains the covariance $\mathbf{k}_X(\mathbf{x}, \mathbf{x}_i)$ in the i-th entry and $\mathbf{x}_i$ denotes the i-th training vector. Then, the next frame in the latent space is: $\mathbf{x}_{N+1} = \mu_X(\mathbf{x}_N)$. Therefore, the new video frames can be generated by $\mathbf{y}_{N+1} = \mu_Y(\mathbf{x}_{N+1})$.

New video textures are successfully generated from input video clips by applying PPCA and GPDM with 50 frames in each video texture. They can be played without any visual discontinuity but with similar motions as the original one. Moreover, all resulted frames have never appeared before in the input videos. Fig.1∼ Fig.6 show the first three frames generated by PPCA and GPDM for several input video clips ((a), (b) and (c) represent the first, second and third frame, respectively).

(a) (b) (c)

Fig. 1. The first three synthesized frames for a movie of a man moving a pen

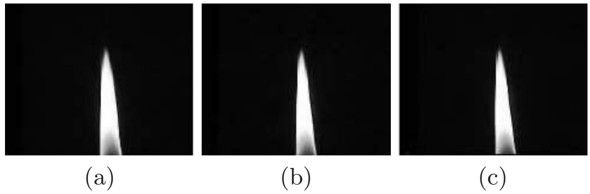

Fig. 2. The first three synthesized frames for a movie a candle flame

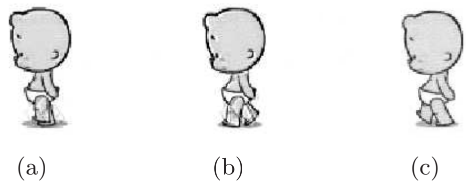

Fig. 3. The first three synthesized frames for an animation of cartoon

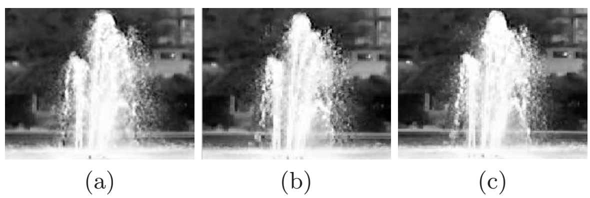

Fig. 4. The first three synthesized frames for a movie of fountain

Fig. 5. The first three synthesized frames for a movie of flag

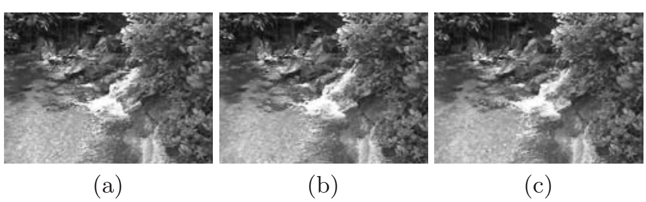

Fig. 6. The first three synthesized frames for a movie of waterfall

4.2 Comparison of the Results

In this section, we compare the performance of synthesizing video textures by GPDM and AR process. Via the AR process, although the result seems very good, there is still one problem which is the occurrence of noise. For all results, after a certain time, the noise will start to become visible and make the video blur. However, through GPDM, it is more robust to noise compared to AR process since it contains a latent space account for the dynamics in the input data. As shown in Fig.7, the 20th, 25th and 30th frames generated by PCA and AR process contain much more noise than the ones produced by PPCA and GPDM at each corresponding frame number. Based on our experimental results, we may conclude that video textures generated by PPCA and GPDM can provide better results with more robustness to noise than AR approach. The synthesized new video textures contain similar motions as the input video clips and all frames in the new video textures are completely new.

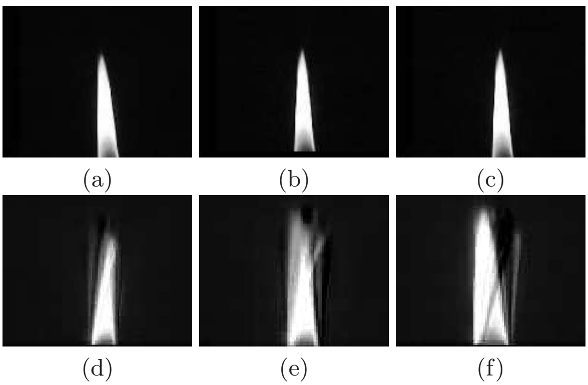

Fig. 7. (a), (b) and (c) illustrate the 20th, 25th and 30th frames synthesized by PPCA and GPDM; (d), (e) and (f) demonstrate the 20th, 25th and 30th frames generated by PCA and AR process

5 Conclusion and Future Works

In this paper, we proposed a new approach for generating video textures using PPCA and GPDM. GPDM is a nonparametric model for learning high-dimensional nonlinear dynamical data sets. We have tested PPCA and GPDM on several movie clips, it can generate video textures containing frames that never appeared before with similar motions as the original video. Compared with PCA and AR process, PPCA and GPDM can produce better results with more robustness to noise. Unfortunately, video textures synthesized by PPCA and GPDM still have visual discontinuities for some highly structured and variable motions (such as dancing and fighting). Thus, there might be some more potential improvements on generating video textures. Since GPDM is highly dependent on the kernel functions, selection of a better function would be a key factor for improving the predictive power. Besides this, We also would like to modify the statistical model of the GPDM in order to acquire the ability of modelling highly variable motion sequences in the future.

Acknowledgments. The completion of this research was made possible thanks to the Natural Sciences and Engineering Research Council of Canada (NSERC) and a NATEQ Nouveaux Chercheurs Grant. Finally, the authors thank CIARP reviewers for their constructive comments.

References

1. Schödl, A., Szeliski, R., Salesin, D., Essa, I.: Video textures. In: Proceedings of the 27th annual conference on Computer graphics and interactive techniques, pp. 489–498. ACM Press/Addison-Wesley Publishing Co., New York (2000)
2. Schödl, A., Essa, I.: Machine learning for video-based rendering. Advances in Neural Information Processing Systems 13, 1002–1008 (2001)
3. Schödl, A., Essa, I.: Controlled animation of video sprites. In: ACM SIGGRAPH Symposium on Computer Animation, pp. 121–128 (2002)
4. Dong, J.Y., Zhang, C.X., Wang, Y.J., Zhou, H.Y.: A video texture synthesis method based on wavelet transform. In: Wavelet Analysis and Pattern Recognition, ICWAPR 2007. International Conference, Beijing, pp. 1177–1181 (2007)
5. Fitzgibbon, A.W.: Stochastic rigidity: image registration for nowhere-static scenes. In: Proceedings of International Conference on Computer Vision (ICCV) 2001, vol. 1, pp. 662–669 (2001)
6. Campbell, N., Dalton, C., Gibson, D., Thomas, B.: Practical generation of video textures using the auto-regressive process. Image and Vision Computing 22, 819–827 (2004)
7. Rasmussen, C., Williams, C.: Gaussian Processes for Machine Learning. The MIT Press, Cambridge (2006)
8. MacKay, D.J.C.: Information Theory, Inference, and Learning Algorithms. Cambridge University Press, Cambridge (2003)
9. MacKay, D.J.C.: Introduction to Gaussian processes. Neural Networks and Machine Learning 168, 133–165 (1998)
10. Wang, J.M., Fleet, D.J., Hertzmann, A.: Gaussian process dynamical models for human motion. IEEE Transactions on Pattern Analysis and Machine Intelligence, 283–298 (2008)
11. Wang, J.M., Fleet, D.J., Hertzmann, A.: Gaussian process dynamical models. In: Advances in Neural Information Processing Systems 18, pp. 1441–1448. MIT Press, Cambridge (2006)
12. Lawrence, N.: Probabilistic non-linear principal component analysis with Gaussian process latent variable models. The Journal of Machine Learning Research 6, 1783–1816 (2005)
13. Fan, W., Bouguila, N.: On Video Textures Generation: A comparison Between Different Dimensionality Reduction Techniques. In: IEEE International Conference on System, Man, and Cybernetics (2009)
14. Bishop, C.M.: Probabilistic principal component analysis. Journal of the Royal Statistical Society, Series B 61, 611–622 (1999)