AN ADAPTIVE DICTIONARY LEARNING APPROACH FOR MODELING DYNAMICAL TEXTURES

Xian Wei, Hao Shen, Martin Kleinsteuber

Department of Electrical Engineering and Information Technology
Technische Universität München, Arcisstr. 21, 80333 Munich, Germany
{xian.wei, hao.shen, kleinsteuber}@tum.de

ABSTRACT

Video representation is an important and challenging task in the computer vision community. In this paper, we assume that image frames of a moving scene can be modeled as a Markov random process. We propose a sparse coding framework, named adaptive video dictionary learning (AVDL), to model a video adaptively. The developed framework is able to capture the dynamics of a moving scene by exploring both sparse properties and the temporal correlations of consecutive video frames. The proposed method is compared with state of the art video processing methods on several benchmark data sequences, which exhibit appearance changes and heavy occlusions.

Index Terms— Dynamic textures modeling, sparse representation, dictionary learning, linear dynamical systems.

1. INTRODUCTION

Temporal or dynamic textures (DT) are image sequences that exhibit spatially repetitive and certain stationarity properties in time. This kind of sequences are typically videos of processes, such as moving water, smoke, swaying trees, moving clouds, or a flag blowing in the wind. Study and analysis of DT is important in several applications such as video segmentation [3], video recognition [12], and DT synthesizing [5].

One classical approach is to model dynamic scenes via the optical flow [10]. However, such methods require a certain degree of motion smoothness and parametric motion models [3]. Non-smoothness, discontinuities, and noise inherence to rapidly varying, non-stationary DTs (e.g. fire) pose a challenge to develop optical flow based algorithms. Another technique, called particle filter [4], models the dynamical course of DTs as a Markov process. A reasonable assumption in DT modeling is that each observation is correlated to an underlying latent variable, or "state", and then derive the parameter transition operator between these states.

This work has been supported by the Cluster of Excellence CoTeSys - Cognition for Technical Systems, funded by the German Research Foundation (DFG).

Some approaches directly view each observation as a state, and then focus on transitions between the observations in the time domain. For instance, the work in [13] treats this transition as an associated probability problem, and other methods construct a spatio-temporal autoregressive model (STAR) or position affine operator for this transition [14, 11].

Differently, feature-based models capture the intrinsic law and underlying structures of the data by projecting the original data onto a low-dimensional feature space via feature extracted techniques, such as principle component analysis (PCA). G. Doretto et al. [12, 5] model the evolution of the dynamic textured scenes as a linear dynamical system (LDS) under a Gaussian noise assumption. As a popular method in dynamic textures, LDS and its derivative algorithms have been successfully used for various dynamic texture applications [5, 12]. However, constraints are imposed on the types of motion and noise that can be modeled in LDS. For instance, it is sensitive to input variations due to various noise. Especially, it is vulnerable to non-Gaussian noise, such as missing data or occlusion of the dynamic scenes. Moreover, stability is also a challenging problem for LDS [2].

To tackle these challenges, the approach taken here is to explore an alternative method to model the DTs by appealing to the principle of sparsity. Instead of using the Principle Components (PCs) as the transition "states" in LDS, sparse coefficients over a learned dictionary are imposed as the underlying "states". In this way, the dynamical process of DTs exhibits a transition course of corresponding sparse events. These sparse events can be obtained via a recent technique on linear decomposition of data, called dictionary learning [6, 9]. Formally, these sparse representations $x \in \mathbb{R}^k$ to a signal $y \in \mathbb{R}^m$, can be written as

$$y = Dx$$

where $D \in \mathbb{R}^{m \times k}$ is a dictionary, and x is sparse, i.e. most of its entries are zero or small in magnitude. That is, the signal y can be sparsely represented only using a few elements from some dictionary D.

In this work, we start with a brief review of the dynamic texture model from the viewpoint of convex ℓ_2 optimization, and then deduce a combined regression associated with

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

several regularizations for a joint process—"states extraction" and "states transition". Then we treat the solution of the above combined regression as an adaptive dictionary learning problem, which can achieve two distinct yet tightly coupled tasks— efficiently reducing the dimensionality via sparse representation and robustly modeling the dynamical process. Finally, we cast this dictionary learning problem as the optimization of a smooth non-convex objective function, which is efficiently resolved via a gradient descent method.

2. ADAPTIVE VIDEO DICTIONARY LEARNING

In this section, we start with a brief introduction to the linear dynamical systems (LDS) model and develop an adaptive dictionary learning framework for sparse coding.

2.1. Linear Dynamical Systems

Let us denote a given sequence of $(n+1)$ frames by $Y := [y_0, \ldots, y_n] \in \mathbb{R}^{m\times(n+1)}$, where the time is indexed by $i = 0, 1, \ldots, n$. The evolution of a LDS is often described by the following two equations

$$\begin{cases} x_{i+1} = Ax_i + w_i \\ y_i = Dx_i + v_i, \end{cases} \tag{1}$$

where $y_i \in \mathbb{R}^m$, $x_i \in \mathbb{R}^k$, $w_i \in \mathbb{R}^k$ and $v_i \in \mathbb{R}^m$ denote the observation, its hidden state or feature, state noise, and observation noise, respectively. The system is described by the dynamics matrix $A \in \mathbb{R}^{k\times k}$, and the modeling matrix $D \in \mathbb{R}^{m\times k}$. Here we are interested in estimating the system parameters A and D, together with the hidden states, given the sequence of observations Y.

The problem of learning the LDS (1) can be considered as a coupled linear regression problem [2]. Let us denote $X = [x_0, \ldots, x_n] \in \mathbb{R}^{k\times(n+1)}$, $X_0 = [x_0, \ldots, x_{n-1}] \in \mathbb{R}^{k\times n}$, and $X_1 = [x_1, \ldots, x_n] \in \mathbb{R}^{k\times n}$. The system dynamics and modeling matrix are expected to be caught by solving the following minimization problem,

$$\min_{A,D,X} \left\|X_1 - AX_0\right\|_F^2 \quad s.t. \left\|Y - DX\right\|_F^2 \le \varepsilon, \tag{2}$$

where ε is a small positive constant. In our approach, we assume that all observations y_i admit a sparse representation with respect to an unknown dictionary $D \in \mathbb{R}^{m\times k}$, i.e.

$$y_i = Dx_i, \qquad \text{for all } i = 0, 1, \ldots, n, \tag{3}$$

where $x_i \in \mathbb{R}^k$ is sparse. Without loss of generality, we further assume that all columns of the dictionary D have unit norm. We then define the set

$$\mathcal{S}(m,k) := \{D \in \mathbb{R}^{m\times k} | \operatorname{ddiag}(D^\top D) = I_k\}, \tag{4}$$

where $\operatorname{ddiag}(Z)$ is the diagonal matrix whose entries on the diagonal are those of Z, I_k denotes the identity matrix. The set $\mathcal{S}(m,k)$ is the product of k unit spheres, and is hence a $k(m-1)$ dimensional smooth manifold. Finally, by adopting the common sparse coding framework to problem (2), we have the following minimization problem

$$\min_{A,D,X} \left\|X_1 - AX_0\right\|_F^2 + \mu_1\left\|Y - DX\right\|_F^2 + \mu_2\|X\|_1, \tag{5}$$

where $D \in \mathcal{S}(m,k)$, $\|\cdot\|_F$ denotes the Frobenius norm of matrices, and $\|\cdot\|_1$ is the ℓ_1 norm, which measures the overall sparsity of a matrix. The parameter $\mu_2 > 0$ weighs the sparsity measurement against the residual errors.

2.2. A Dictionary Learning Model for Dynamical Scene

Solving the minimization problem as stated in Eq. (5) is a very challenging task. In this work, we employ an idea similar to *subspace identification methods* [2], which treat the state as a function of (A, D). Here, we confine ourselves to the sparse solution of an elastic-net problem, which is proposed in [16], as

$$x^* := \operatorname*{argmin}_{x\in\mathbb{R}^k} \tfrac{1}{2}\|y - Dx\|_2^2 + \lambda_1\|x\|_1 + \tfrac{\lambda_2}{2}\|x\|_2^2, \tag{6}$$

where $\lambda_1 > 0$ and $\lambda_2 > 0$ are regularization parameters, which play an important role in ensuring stability and uniqueness of the solutions. Let us define the set of indices of the non-zero entries of the solution $x^* = [x_1^*, \ldots, x_k^*]^\top \in \mathbb{R}^k$ as

$$\Lambda := \{i \in \{1, \ldots, k\} | x_i^* \neq 0\}. \tag{7}$$

Then the solution x^* has a closed-form expression as

$$x_y^*(D) := \left(D_\Lambda^\top D_\Lambda - \lambda_2 I_m\right)^{-1}\left(D_\Lambda^\top y - \lambda_1 s_\Lambda\right), \tag{8}$$

where $s_\Lambda \in \{\pm 1\}^{|\Lambda|}$ carries the signs of x_Λ^*, D_Λ is the subset of D in which the index of atoms (rows) fall into support Λ. Furthermore, it is known that the solution $x_y^*(D)$ as given in (8) is a locally twice differentiable function at D. By an abuse of notation, we define

$$\begin{aligned} X_0 \colon \mathcal{S}(m,k) &\to \mathbb{R}^{k\times n} \\ D &\mapsto [x_{y_0}^*(D), \ldots, x_{y_{n-1}}^*(D)]. \end{aligned} \tag{9}$$

In a similar way, $X_1 \colon \mathcal{S}(m,k) \to \mathbb{R}^{k\times n}$ is defined. Thus, the cost function reads as

$$\begin{aligned} f \colon \mathbb{R}^{k\times k} \times \mathcal{S}(m,k) &\to \mathbb{R} \\ (A, D) &\mapsto \tfrac{1}{2}\left\|X_1(D) - AX_0(D)\right\|_F^2. \end{aligned} \tag{10}$$

It is known that an LDS with the dynamic matrix A is said to be stable, if the largest eigenvalue of A is bounded by 1 [2]. Let σ be the largest eigenvalue of A, then $|\sigma| \le \|A\|_F$. Thus, we enforce the small σ via imposing a penalty $\|A\|_F^2$ on (10), and then end up with the cost function as

$$\begin{aligned} \widetilde{f} \colon \mathbb{R}^{k\times k} \times \mathcal{S}(m,k) &\to \mathbb{R} \\ (A, D) &\mapsto f(A, D) + \tfrac{\gamma}{2}\|A\|_F^2, \end{aligned} \tag{11}$$

2.3. Development of the Algorithm

In this section, we firstly derive a gradient descent algorithm to minimize (11) and then discuss some details of the choice of the parameters in the final implementation.

We start with the computation of the first derivative of the sparse solution of the elastic-net problem $x_y^*(D)$ as given in (8). Given the tangent space of $\mathcal{S}(m,k)$ at D as

$$T_D\mathcal{S}(m,k) := \{X \in \mathbb{R}^{m\times k} | \operatorname{ddiag}(X^\top D) = 0\}, \quad (12)$$

the orthogonal projection of a matrix $H \in \mathbb{R}^{m\times k}$ onto the tangent space $T_D\mathcal{S}(p,n)$ with respect to the inner product $\langle X, Y\rangle = \operatorname{tr}(X^\top Y)$ is given by

$$\Pi_D(H) := H - D\operatorname{ddiag}(D^\top H). \quad (13)$$

Let us denote $K := D_\Lambda^\top D_\Lambda - \lambda_2 I_k$. The first derivative of x_y^* in the direction $H \in T_D\mathcal{S}(m,k)$ is

$$\begin{aligned}\mathrm{D}\, x_y^*(D)H =& K^{-1}H_\Lambda^\top y - K^{-1}(D_\Lambda^\top H_\Lambda \\ &+ H_\Lambda^\top D_\Lambda)\cdot K^{-1}\left(D_\Lambda^\top y - \lambda_1 s_\Lambda\right).\end{aligned} \quad (14)$$

By the product structure of $\mathbb{R}^{k\times k}\times\mathcal{S}(m,k)$, the Riemannian gradient of the function $\widetilde{f}$ is

$$\operatorname{grad}\widetilde{f}(A,D) = \left(\nabla_{\widetilde{f}}(A), \Pi_D\left(\nabla_{\widetilde{f}}(D)\right)\right). \quad (15)$$

Here, the Euclidean gradient $\nabla_{\widetilde{f}}(A)$ of $\widetilde{f}$ with respect to A is computed as

$$\nabla_{\widetilde{f}}(A) = (AX_0(D) - X_1(D))\,X_0(D) + \gamma A, \quad (16)$$

with e_i being the i-th standard basis vector of $\mathbb{R}^n$. Using the shorthand notation, $r_i := D_{\Lambda_i}^\top y_i - \lambda_1 s_{\Lambda_i}$, $\Delta x_i := x_{y_i}^*(D) - A_{\Lambda_i}x_{y_{i-1}}^*(D)$, and $q_i := r_i\Delta x_i^\top$, the Euclidean gradient $\nabla_{\widetilde{f}}(D)$ of $\widetilde{f}$ with respect to D is

$$\begin{aligned}\nabla_{\widetilde{f}}(D) =& \sum_{i=1}^{n} y_i(\Delta x_i)^\top K_i^{-1} - D_{\Lambda_i}K_i^{-1}(q_i + q_i^\top) \\ &\cdot K_i^{-1} - y_{i-1}(\Delta x_i)^\top A_{\Lambda_i}(K_{i-1})^{-1} + D_{\Lambda_{i-1}} \\ &\cdot (K_{i-1})^{-1}(A_{\Lambda_{i-1}}q_{i-1} + q_{i-1}^\top A_{\Lambda_{i-1}}^\top)(K_{i-1})^{-1}.\end{aligned} \quad (17)$$

For a gradient search iteration on manifolds, we employ the following smooth curve on $\mathcal{S}(m,k)$ through $D \in \mathcal{S}(m,k)$ in direction $H \in T_D\mathcal{S}(m,k)$

$$\begin{aligned}&\tau\colon (-\lambda,\lambda) \to \mathcal{S}(m,k) \\ &t \mapsto (D+tH)\left(\operatorname{ddiag}((D+tH)^\top(D+tH))\right)^{-\frac{1}{2}}\end{aligned} \quad (18)$$

with $\lambda > 0$. It essentially normalizes all columns of $D + tH$. For a detailed overview on optimization on matrix manifold, refer to [1].

Algorithm 1: Adaptive Video Dictionary Learning

1: Training data Y
2: Initialize the parameters λ_1,λ_2,γ, initial dictionary D, and initial transition matrix A.
3: **for** $i = 1, 2, \ldots, T$ **do**
4: *Sparse Coding Stage*
Use Lasso algorithm to compute x via $x \leftarrow \min_x \frac{1}{2}\|y - Dx\|_2^2 + \lambda_1\|x\|_1 + \frac{\lambda_2}{2}\|x\|_2^2$
Compute the active set Λ for each x.
5: Compute the gradient of $\widetilde{f}(A,D)$ according to (16) and (17).
6: Update the parameters A and D

$$A_i \leftarrow A_{i-1} - \rho_i\nabla_{\widetilde{f}}(A_{i-1}),$$
$$D_i \leftarrow D_{i-1} - \rho_i\nabla_{\widetilde{f}}(D_{i-1}).$$

7: **end for**
8: **return** A and D

Until now, we have computed the gradient of $\widetilde{f}$ as defined in (11) with respect to its two arguments D and A. An iterative scheme (such as the gradient descent method or conjugate gradient method) can be used to find the optimal D and A, using the gradient expression above. The procedure displayed in Algorithm (1) is the version of AVDL based on gradient descent procedure. The learning rate ρ_i can be computed via the well-known backtracking line search method, similar to [9]. Here, considering the high coherence among the temporal frames, we prefer non-redundant dictionary, that is, $k \ll m$ for the dictionary $D \in \mathbb{R}^{m\times k}$. For parameters (λ_1, λ_2) in the elastic net, we put an emphasis on sparse solutions and choose $\lambda_2 \in (0, \frac{\lambda_1}{10})$, as proposed in [16].

3. NUMERICAL EXPERIMENTS

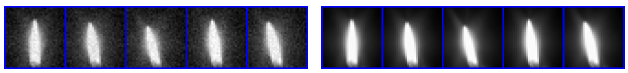
We carry out a few experiments on natural image sequences data, and demonstrate the practicality of the proposed algorithm. Our test dataset comprises of videos from DynTex++ [7], and data from internet sources (for instance, YouTube). Firstly, we show the performance on reconstruction and synthesizing with a grayscale video of burning candle, which is corrupted by Gaussian noise or occlusion. This video has 1024 frames with size of 32×32, see figure 1. The initial dictionary is 1024×512. After the acquisition of the dictionary D and the transition A, the synthesized data can be generated easily by $x_{i+1} = Ax_i x_i^\top x_i (x_i^\top x_i)^{-1}$, or more precisely, using a convex formulation

$$\min_{x_{i+1}} \frac{1}{2}\|x_{i+1} - Ax_i\|_2^2 + \lambda\|x_{i+1}\|_1.$$

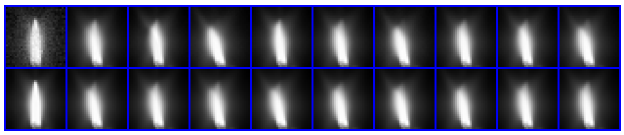
Table 1 shows the performance of synthesizing on burning candle with Gaussian noise. The error pairs (e_x, e_y) are

Table 1. Synthesizing results on sequence of burning candle.

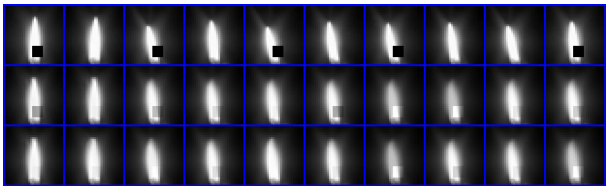
Instance	LDS, (PCs)			AVDL, $\gamma = 0.5$, (loops)				
	64	128	256	1	50	100	200	400
Compression rate (%)	6.25	12.50	25.00	1.02	3.29	3.41	3.50	3.55
σ	0.9802	0.9833	0.9849	1.78	1.06	0.9992	0.9994	0.9994
e_y	1.35×10^5	1.35×10^5	1.35×10^5	1.36×10^3	60.29	58.82	55.97	71.27
e_x	101.58	135.88	168.95	3.75×10^4	171.99	75.52	61.96	46.18



(a) Corrupted original sequence (b) Reconstructed sequence



(c) Synthesized video using LDS and AVDL on DTs with Gaussian noise



(d) Synthesized video using LDS and AVDL on DTs with missing data

Fig. 1. Reconstruction and synthesizing on the candle scene. (a), (b) are $(i = 1, 64, 128, 512, 1024)th$ frame of the corrupted data by Gaussian noisy and the reconstructed data using AVDL, respectively. (c) The top row is the synthesized sequence using LDS (128PCs), and the bottom row is the synthesized sequence using AVDL ($(i = 2, 1024, 3072, 5120, \ldots, 20480)th$ frame). (d) The top row is the sequence with missing data. The middle row the synthesized sequence using LDS, and the bottom row is the synthesized sequence using AVDL.

defined as $e_y = \sum_i \|y_i - Dx_i\|$, $e_x = \sum_i \|x_{i+1} - Ax_i\|$, and the largest eigenvalue of A is denoted by σ. The compression rate for AVDL is sparsity of x to $m \times (n+1)$, and for LDS is number of PCs to m. Table 1 shows AVDL can obtain the stable dynamic matrix A ($\sigma \leq 1$), smaller compression rate and smaller error (e_x, e_y) of cost function (5), by increasing the numbers of main loops in Algorithm 1.

Figure 1 $(a \sim c)$ is the visual comparison between LDS and AVDL. AVDL performs well on denoising against corruption by Gaussian noise. In the case of occlusion in figure 1 (d), random 50 frames of the 1024 burning candle video are corrupted by a (6×7) rectangle. The length of both synthesizing data is 1024, based on first frame of the burning candle. 87.01% of the synthesizing data from LDS are corrupted by this rectangle, but 9.47% for AVDL.

The second experiment is about scenes classification on DynTex++, which contains DTs from 36 classes. Each class has 100 subsequences of length 50 frames with 50×50 pixels. 20 videos are randomly chosen in each class and total 720 videos are used for our experiments. Classification for LDS is performed using the Martin distance with a nearest-neighbor classifier on its parameters pair (A, D) [12]. Another classifier is AVDL associated with the sparse representation-based classifier (SRC) [15, 8], in which the class of a test sequence is determined by the smallest reconstruction error e_y and transition error e_x. Table 2 provides the recognition results with increasing occlusion rates for test data. Compared to LDS with nearest-neighbor classifier (LDS-NN), Table 2 shows the proposed AVDL with SRC (AVDL-SRC) performs better while the test videos are corrupted by increasing occlusion.

Table 2. DT recognition rates for videos with occlusion.

Occlusion rate (%)	0	5	15	30
LDS-NN (128PCs)	69.72	45.00	25.14	14.17
AVDL-SRC	70.28	64.72	44.44	22.36

4. CONCLUSIONS

This paper proposes an alternative method, called AVDL, to model the dynamic process of DTs. In AVDL, the sparse events over a dictionary are imposed as transition states. The proposed method show a robust performance for synthesizing, reconstruction and recognition on DTs corrupted by Gaussian noise. Especially, AVDL exhibits more powerful in the case of test data with non-Gaussian noise, such as occlusion. One possible future extension is to learn a dictionary for large scale DT sequences based on AVDL.

5. REFERENCES

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[2] B. Boots, G. J. Gordon, and S. M. Siddiqi. A constraint generation approach to learning stable linear dynamical systems. In *Advances in Neural Information Processing Systems*, pages 1329–1336, 2007.

[3] A. B. Chan and N. Vasconcelos. Layered dynamic textures. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(10):1862–1879, 2009.

[4] P. M. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez. Particle filtering. *Signal Processing Magazine, IEEE*, 20(5):19–38, 2003.

[5] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto. Dynamic textures. *International Journal of Computer Vision*, 51(2):91–109, 2003.

[6] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *Image Processing, IEEE Transactions on*, 15(12):3736–3745, 2006.

[7] B. Ghanem and N. Ahuja. Maximum margin distance learning for dynamic texture recognition. In *European Conference on Computer Vision*, pages 223–236. Springer, 2010.

[8] B. Ghanem and N. Ahuja. Sparse coding of linear dynamical systems with an application to dynamic texture recognition. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 987–990. IEEE, 2010.

[9] S. Hawe, M. Seibert, and M. Kleinsteuber. Separable dictionary learning. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 438–445, June 2013.

[10] B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial intelligence*, 17(1):185–203, 1981.

[11] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: image and video synthesis using graph cuts. In *Graphics (TOG), ACM Transactions on*, volume 22, pages 277–286. ACM, 2003.

[12] P. Saisan, G. Doretto, Y. N. Wu, and S. Soatto. Dynamic texture recognition. In *Computer Vision and Pattern Recognition. IEEE Computer Society Conference on*, volume 2, pages II–58. IEEE, 2001.

[13] A. Schödl, R. Szeliski, D. H. Salesin, and I. Essa. Video textures. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 489–498. ACM Press/Addison-Wesley Publishing Co., 2000.

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

[15] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(2):210–227, 2009.

[16] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.