

# Variational Layered Dynamic Textures

Antoni B. Chan      Nuno Vasconcelos

Department of Electrical and Computer Engineering  
University of California, San Diego

abchan@ucsd.edu, nuno@ece.ucsd.edu

## Abstract

The layered dynamic texture (LDT) is a generative model, which represents video as a collection of stochastic layers of different appearance and dynamics. Each layer is modeled as a temporal texture sampled from a different linear dynamical system, with regions of the video assigned to a layer using a Markov random field. Model parameters are learned from training video using the EM algorithm. However, exact inference for the E-step is intractable. In this paper, we propose a variational approximation for the LDT that enables efficient learning of the model. We also propose a temporally-switching LDT (TS-LDT), which allows the layer shape to change over time, along with the associated EM algorithm and variational approximation. The ability of the LDT to segment video into layers of coherent appearance and dynamics is also extensively evaluated, on both synthetic and natural video. These experiments show that the model possesses an ability to group regions of globally homogeneous, but locally heterogeneous, stochastic dynamics currently unparalleled in the literature.

## 1. Introduction

Traditional motion representations, based on optical flow, are inherently local and have significant difficulties when faced with aperture problems and noise. The classical solution to this problem is to regularize the optical flow field [1, 2, 3, 4], but this introduces undesirable smoothing across motion edges or regions where the motion is, by definition, not smooth (e.g. vegetation in outdoors scenes). It also does not provide any information about the objects that compose the scene, although the optical flow field could be subsequently used for motion segmentation. More recently, there have been various attempts to model video as a superposition of layers subject to homogeneous motion. While layered representations exhibited significant promise in terms of combining the advantages of regularization (use of global cues to determine local motion) with the flexibility of local representations (little undue smoothing), and a

truly object-based representation, this potential has so far not fully materialized. One of the main limitations is their dependence on parametric motion models, such as affine transforms, which assume a piece-wise planar world that rarely holds in practice [5, 6]. In fact, layers are usually formulated as “cardboard” models of the world that are warped by such transformations and then stitched to form the frames in a video stream [5]. This severely limits the types of video that can be synthesized: while the concept of layering showed most promise for the representation of scenes composed of ensembles of objects subject to homogeneous motion (e.g. leaves blowing in the wind, a flock of birds, or highway traffic), very little progress has so far been demonstrated in actually modeling such scenes.

Recently, there has been more success in modeling complex scenes as *dynamic textures* or, more precisely, samples from stochastic processes defined over space and time [7, 8, 9]. This work has demonstrated that *global stochastic modeling* of both video dynamics and appearance is much more powerful than the classic global modeling as “cardboard” figures under parametric motion. In fact, the dynamic texture (DT) has shown a surprising ability to abstract a wide variety of complex patterns of motion and appearance into a *simple* spatio-temporal model. One major current limitation is, however, its inability to account for visual processes consisting of *multiple, co-occurring, dynamic textures*, for example, a flock of birds flying in front of a water fountain, highway traffic moving at different speeds, and video containing both trees in the background and people in the foreground. In such cases, the existing DT model is ill-equipped to model the video, since it must represent multiple motion fields with a single dynamic process.

To address this problem, various extensions of the DT have been recently proposed in the literature [8, 10, 11]. These extensions have emphasized the *application* of the standard DT model to video segmentation, rather than exploiting the probabilistic nature of the DT representation to propose a *global generative model for video*. They represent the video as a collection of *localized* spatio-temporal patches (or pixel trajectories), which are modeled with dy-

dynamic textures or similar time-series representations, and *clustered* to produce the desired segmentations. Due to their local character, these representations cannot account for *globally homogeneous* textures that exhibit substantial *local heterogeneity*. These types of textures are common in both urban settings, where the video dynamics frequently combine global motion and stochasticity (e.g. vehicle traffic around a square, or pedestrian traffic around a landmark), and natural scenes (e.g. a flame that tilts under the influence of the wind, or water rotating in a whirlpool).

These limitations were addressed in [12] through the introduction of a global generative model, denoted as the *layered dynamic texture* (LDT). This model augments the DT with a discrete *hidden* variable that enables the assignment of different dynamics to different regions of the video. The hidden variable is modeled as a Markov random field (MRF), to ensure spatial smoothness of the segmentation, and conditioned on its state, each video region is a standard DT. An EM algorithm for maximum-likelihood estimation of LDT parameters from an observed video sample was also derived in [12]. The problem of the intractability of exact inference during the E-step (due to the MRF) was addressed with the use of a Gibbs sampler. This, however, results in a slow learning algorithm, limiting the application of the model to very small video samples. In this work, we propose a *variational approximation* for the LDT that enables *efficient learning* of its parameters. We further propose an LDT extension, the *temporal-switching* LDT, that allows the shape of the layers to *change over time*, enabling segmentation in both *space and time*. Finally, we apply the LDT to motion segmentation of challenging video sequences, and report state-of-the-art results on the synthetic texture database from [13].

The paper is organized as follows. In Section 2, we review the LDT and the EM learning algorithm. The variational approximation is proposed in Section 3, and the temporally-switching LDT in Section 4. Finally, in Section 5, the variational LDT is applied to motion segmentation of both synthetic and real videos.

## 2. Layered dynamic textures

Consider a video composed of various textures, e.g. the combination of fire, smoke, and water, shown on the right side of Figure 1. In this case, a single DT cannot simultaneously account for the appearance and dynamics of the three textures, because each texture moves distinctly, e.g. fire changes faster and is more chaotic than smoke. This type of video can be modeled by encoding each texture as a separate *layer*, with its own state-sequence and observation matrix (see Figure 1). Different regions of the spatiotemporal video volume are assigned to each texture and, conditioned on this assignment, each region evolves as a standard DT. The video is a composite of the various layers.

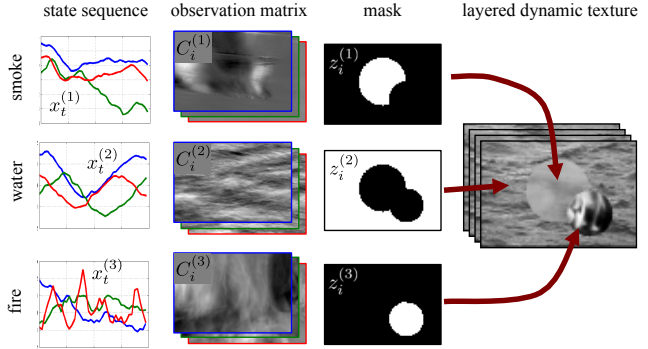


Figure 1. Generative model for a video with multiple dynamic textures (smoke, water, and fire). The three textures are modeled with separate state sequences and observation matrices. The textures are then masked, and composited to form the layered video.

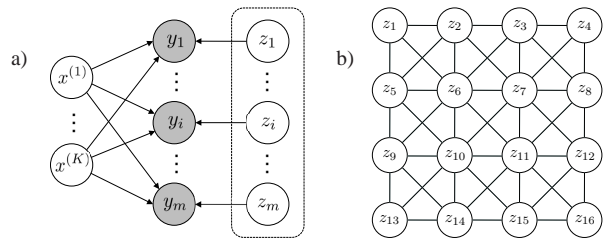


Figure 2. a) Graphical model for the LDT; b) Example of a  $4 \times 4$  layer assignment MRF.

The graphical model for the LDT [12] is shown in Figure 2a. Each of the  $K$  layers has a state process  $x^{(j)} = \{x_t^{(j)}\}$  that evolves separately, where  $x_t^{(j)} \in \mathbb{R}^n$  is the state vector at time  $t$  and  $n$  is the dimension of the hidden state-space. A pixel trajectory  $y_i = \{y_{i,t}\}$ , where  $y_{i,t} \in \mathbb{R}$  is the pixel value at location  $i$  at time  $t$ , is assigned to one of the layers through the hidden variable  $z_i$ , and the collection of hidden variables  $Z = \{z_i\}$  is modeled as an MRF grid to ensure spatial smoothness of the layer assignments (e.g. Figure 2b). We will assume that each pixel  $y_i$  has zero-mean over time (*i.e.* mean-subtracted). The model equations are

$$\begin{cases} x_t^{(j)} = A^{(j)}x_{t-1}^{(j)} + v_t^{(j)} & , j \in \{1, \dots, K\} \\ y_{i,t} = C_i^{(z_i)}x_t^{(z_i)} + w_{i,t} & , i \in \{1, \dots, m\} \end{cases} \quad (1)$$

where  $C_i^{(j)} \in \mathbb{R}^{1 \times n}$  is the transformation from the hidden state to the observed pixel for *each* pixel  $y_i$  and *each* layer  $j$ . The noise processes and initial state are distributed as Gaussians, *i.e.*  $v_t^{(j)} \sim \mathcal{N}(0, Q^{(j)})$ ,  $w_{i,t} \sim \mathcal{N}(0, r^{(z_i)})$ , and  $x_1^{(j)} \sim \mathcal{N}(\mu^{(j)}, Q^{(j)})$ , where  $Q^{(j)}$  is a  $n \times n$  covariance matrix, and  $r^{(j)} > 0$ . Each layer is parameterized by  $\Theta_j = \{A^{(j)}, Q^{(j)}, C^{(j)}, r^{(j)}, \mu^{(j)}\}$ . Finally, the MRF  $Z$  has potential functions

$$V_i(z_i) = \alpha_i^{(z_i)}, \quad V_{i,i'}(z_i, z_{i'}) = \begin{cases} \gamma_1, & z_i = z_{i'} \\ \gamma_2, & z_i \neq z_{i'} \end{cases} \quad (2)$$

where  $V_i$  is the self-potential function with  $\alpha_i^{(j)}$  the prior probability for assigning  $z_i = j$ , and  $V_{i,i'}$  is the potential function between connected nodes  $z_i$  and  $z_{i'}$  that attributes higher probability to configurations with neighboring pixels in the same layer. In this work, we treat the MRF as a prior on  $Z$ , which controls the smoothness of the layers.

Given layer assignments, the LDT is a superposition of DTs defined over different regions of the video volume. In this case, estimating LDT parameters reduces to estimating those of the DT of each region. When layer assignments are unknown, the LDT parameters can be estimated with the EM algorithm [12].

### 2.1. Parameter estimation with EM

Given a training video  $Y = \{y_{i,t}\}$ , the parameters  $\Theta$  of the LDT are learned by maximum-likelihood [14]

$$\Theta^* = \operatorname{argmax}_{\Theta} \log p(Y) = \operatorname{argmax}_{\Theta} \log \sum_{X,Z} p(Y, X, Z). \quad (3)$$

Since the data likelihood depends on hidden variables (state sequences  $X = \{x^{(j)}\}$  and layer assignments  $Z$ ), (3) can be found with the EM algorithm [15], which iterates between

$$\text{E - Step: } \mathcal{Q}(\Theta; \hat{\Theta}) = \mathbb{E}_{X,Z|Y;\hat{\Theta}}[\log p(X, Y, Z; \Theta)] \quad (4)$$

$$\text{M - Step: } \hat{\Theta}' = \operatorname{argmax}_{\Theta} \mathcal{Q}(\Theta; \hat{\Theta}), \quad (5)$$

where  $p(X, Y, Z; \Theta)$  is the complete-data likelihood, parameterized by  $\Theta$ , and  $\mathbb{E}_{X,Z|Y;\hat{\Theta}}$  the expectation with respect to  $X$  and  $Z$ , conditioned on  $Y$ , parameterized by the current estimates  $\hat{\Theta}$ . As is typical for mixture models, we use an indicator variable  $z_i^{(j)}$  of value 1 if and only if  $z_i = j$ , and 0 otherwise. In the E-step [12], the following conditional expectations are computed

$$\begin{aligned} \hat{x}_t^{(j)} &= \mathbb{E}_{X|Y}[x_t^{(j)}], & \hat{z}_i^{(j)} &= \mathbb{E}_{Z|Y}[z_i^{(j)}], \\ \hat{P}_{t,t}^{(j)} &= \mathbb{E}_{X|Y}[P_{t,t}^{(j)}], & \hat{P}_{t,t-1}^{(j)} &= \mathbb{E}_{X|Y}[P_{t,t-1}^{(j)}], \\ \hat{x}_{t|i}^{(j)} &= \mathbb{E}_{X|Y, z_i=j}[x_t^{(j)}], & \hat{P}_{t,t|i}^{(j)} &= \mathbb{E}_{X|Y, z_i=j}[P_{t,t}^{(j)}], \end{aligned} \quad (6)$$

where  $\mathbb{E}_{X|Y, z_i=j}$  is the conditional expectation of  $X$  given the observation  $Y$  and that the  $i$ -th pixel belongs to layer  $j$ . Next a number of statistics are aggregated over time,

$$\begin{aligned} \Gamma_i^{(j)} &= \sum_{t=1}^{\tau} y_{i,t} \hat{x}_{t|i}^{(j)}, & \Phi_i^{(j)} &= \sum_{t=1}^{\tau} \hat{P}_{t,t|i}^{(j)}, \\ \phi_1^{(j)} &= \sum_{t=1}^{\tau-1} \hat{P}_{t,t}^{(j)}, & \phi_2^{(j)} &= \sum_{t=2}^{\tau} \hat{P}_{t,t}^{(j)}, \\ \psi^{(j)} &= \sum_{t=2}^{\tau} \hat{P}_{t,t-1}^{(j)}, & \hat{N}_j &= \sum_{i=1}^m \hat{z}_i^{(j)}, \end{aligned} \quad (7)$$

where  $\tau$  is the number of video frames. In the M-step [12], the parameter estimates are recomputed

$$\begin{aligned} C_i^{(j)*} &= \Gamma_i^{(j)T} \Phi_i^{(j)-1}, & A^{(j)*} &= \psi^{(j)} \phi_1^{(j)-1}, & \mu^{(j)*} &= \hat{x}_1^{(j)}, \\ r^{(j)*} &= \frac{1}{\tau \hat{N}_j} \sum_{i=1}^m \hat{z}_i^{(j)} \left( \sum_{t=1}^{\tau} y_{i,t}^2 - C_i^{(j)*} \Gamma_i^{(j)} \right), \\ Q^{(j)*} &= \frac{1}{\tau} \left( \hat{P}_{1,1}^{(j)} - \mu^{(j)*} (\mu^{(j)*})^T + \phi_2^{(j)} - A^{(j)*} \psi^{(j)T} \right). \end{aligned}$$

### 2.2. Related work

A number of applications of DT (or similar) models to segmentation have been reported in the literature [8, 10, 11], but do not exploit the probabilistic nature of the DT representation for the segmentation itself. More related to the extensions proposed is the *dynamic texture mixture* (DTM) of [13]. This is a model for collections of video sequences, and has been successfully used for motion segmentation through clustering of spatio-temporal patches. The main difference with respect to the LDT is that (like all clustering models) the DTM is not a *global generative model* for video of *co-occurring* textures (as is the case of the LDT). Hence, the application of the DTM to segmentation requires decomposing the video into a collection of small spatio-temporal patches, which are then clustered. The *localized* nature of this video representation is problematic for the segmentation of textures which are *globally* homogeneous but exhibit substantial variation between neighboring locations, such as the rotating blades of a windmill. Furthermore, patch-based segmentations have poor boundary accuracy, due to the artificial boundaries of the underlying patches, and the difficulty of assigning a patch that overlaps multiple regions to any of them. On the other hand, the LDT models video as a collection of layers, offering a truly *global* model of the appearance and dynamics of each layer, and avoiding boundary uncertainty.

With respect to time-series models, the LDT is related to switching linear dynamical models, which are LDSs that can switch between different parameter sets over time [16, 17, 18, 19]. In particular, it is most related to the switching state-space LDS [19], which models the observed variable by switching between the outputs of a set of independent LDSs. The fundamental difference between the two models is that, while [19] switches parameters in *time* using a hidden-Markov model (HMM), the LDT switches parameters in *space* (i.e. within the dimensions of the observed variable) using an MRF grid. This substantially complicates all statistical inference, leading to different algorithms for learning and inference with the LDT.

### 3. Inference by variational approximation

Computing the exact E-step for the LDT is intractable because the expectations of (6) require marginalizing over the states of the MRF. In [12], these expectations are approximated using a Gibbs sampler, which is slow and limits the learning algorithm to small videos. A popular low-complexity alternative to exact inference is to rely on a variational approximation. This consists of directly approximating the posterior distribution  $p(X, Z|Y)$  with a distribution  $q(X, Z)$  within some class of tractable probability distributions  $\mathcal{F}$ . Given an observation  $Y$ , the optimal variational approximation minimizes the Kullback-Leibler diver-

gence between the approximate and exact posteriors [20]

$$q^*(X, Z) = \operatorname{argmin}_{q \in \mathcal{F}} D(q(X, Z) \| p(X, Z|Y)) \quad (8)$$

$$= \operatorname{argmin}_{q \in \mathcal{F}} \mathcal{L}(q(X, Z)), \quad (9)$$

where

$$\mathcal{L}(q(X, Z)) = \int q(X, Z) \log \frac{q(X, Z)}{p(X, Y, Z)} dX dZ. \quad (10)$$

To obtain a tractable approximate posterior, we assume statistical independence between pixel assignments  $z_i$  and state variables  $x^{(j)}$ , i.e.

$$q(X, Z) = \prod_{j=1}^K q(x^{(j)}) \prod_{i=1}^m q(z_i), \quad (11)$$

and note that optimizing  $\mathcal{L}$  (i.e. finding the best approximate posterior) will induce a set of *variational parameters* that models the dependencies between  $x^{(j)}$  and  $z_i$ . Substituting (11) into (10), the  $\mathcal{L}$  function is minimized by sequentially optimizing each of the factors  $q(x^{(j)})$  and  $q(z_i)$ , while holding the others constant [20]. The optimal factorial distributions are (see [21] for derivations)

$$\log q(x^{(j)}) = \sum_{i=1}^m h_i^{(j)} \log p(y_i | x^{(j)}, z_i = j) \quad (12)$$

$$+ \log p(x^{(j)}) - \log \mathcal{Z}_q^{(j)},$$

$$\log q(z_i) = \sum_{j=1}^K z_i^{(j)} \log h_i^{(j)}, \quad (13)$$

where  $\mathcal{Z}_q^{(j)}$  is a normalization constant (see [21]),  $h_i^{(j)}$  are the variational parameters

$$h_i^{(j)} = \mathbb{E}_q[z_i^{(j)}] = \frac{\alpha_i^{(j)} g_i^{(j)}}{\sum_{k=1}^K \alpha_i^{(k)} g_i^{(k)}}, \quad (14)$$

$$\log g_i^{(j)} = \mathbb{E}_q \left[ \log p(y_i | x^{(j)}, z_i = j) \right] \quad (15)$$

$$+ \sum_{(i, i') \in \mathcal{E}} h_{i'}^{(j)} \log \frac{\gamma_1}{\gamma_2},$$

$\mathbb{E}_q$  is the expectation with respect to  $q(X, Z)$ , and  $\mathcal{E}$  is the set of edges in the MRF.

The optimal factorial distributions can be interpreted as follows. The variational parameters  $\{h_i^{(j)}\}$ , which appear in both  $q(z_i)$  and  $q(x^{(j)})$ , account for the dependence between  $X$  and  $Z$ .  $h_i^{(j)}$  is the posterior probability of assigning pixel  $y_i$  to layer  $j$ , and is estimated by the expected log-likelihood of observing pixel  $y_i$  from layer  $j$ , with an additional boost of  $\log \frac{\gamma_1}{\gamma_2}$  per neighboring pixel also assigned to layer  $j$ .  $h_i^{(j)}$  also weighs the contribution of each pixel  $y_i$  to the factor

$q(x^{(j)})$ , which effectively acts as a soft assignment of pixel  $y_i$  to layer  $j$ . Also note that in (12),  $h_i^{(j)}$  can be absorbed into  $p(y_i | x^{(j)}, z_i = j)$ , making  $q(x^{(j)})$  the distribution of an LDS parameterized by  $\tilde{\Theta}_j = \{A^{(j)}, Q^{(j)}, C^{(j)}, R_j, \mu^{(j)}\}$ , where  $R_j$  is a diagonal matrix with entries  $[\frac{r_1^{(j)}}{h_1^{(j)}}, \dots, \frac{r_m^{(j)}}{h_m^{(j)}}]$ .

The optimal  $q^*(X, Z)$  is found by iterating through each pixel  $i$ , recomputing the variational parameters  $h_i^{(j)}$  according to (14) and (15), until convergence. This might be computationally expensive, because it requires running a Kalman smoothing filter to update each  $h_i^{(j)}$ . The computational load can be reduced by updating batches of variational parameters at a time, e.g. the set of nodes in the MRF with non-overlapping Markov blankets (as in [22]). In practice, batch updating typically converges to the solution reached by serial updating, but is significantly faster.

Given the optimal approximate posterior  $q^*(X, Z)$ , the approximation to (6) of the E-step is

$$\begin{aligned} \hat{x}_t^{(j)} &\approx \mathbb{E}_{q^*}[x_t^{(j)}], & \hat{P}_{t,t}^{(j)} &\approx \mathbb{E}_{q^*}[x_t^{(j)} x_t^{(j)T}], \\ \hat{z}_i^{(j)} &\approx h_i^{(j)}, & \hat{P}_{t,t-1}^{(j)} &\approx \mathbb{E}_{q^*}[x_t^{(j)} x_{t-1}^{(j)T}], \\ \hat{x}_{t|i}^{(j)} &= \mathbb{E}_{X|Y, z_i=j}[x_t^{(j)}] \approx \mathbb{E}_{q^*}[x_t^{(j)}], \\ \hat{P}_{t,t|i}^{(j)} &= \mathbb{E}_{X|Y, z_i=j}[x_t^{(j)} x_t^{(j)T}] \approx \mathbb{E}_{q^*}[x_t^{(j)} x_t^{(j)T}]. \end{aligned} \quad (16)$$

Note that for the expectation  $\mathbb{E}_{X|Y, z_i=j}$ , we assume that, if  $m$  is large (as is the case with images), fixing the value of a single  $z_i = j$  will have little effect on the posterior, due to the combined evidence from the large number of pixels in the layer. Finally, the approximation for the *maximum a posteriori* layer assignment (i.e. segmentation),  $Z^* = \operatorname{argmax}_Z p(Z|Y)$ , is  $z_i^* \approx \operatorname{argmax}_j h_i^{(j)}$ ,  $\forall i$ .

## 4. Temporally-switching LDT

In this section, we propose an extension of the LDT, which we denote as the *temporally-switching layered dynamic texture* (TS-LDT). The TS-LDT contains a *temporally-switching* MRF that allows for the layer regions to change over time, and hence enables segmentation in *both space and time*. In the TS-LDT, a pixel  $y_{i,t}$  is assigned to one of the layers *at each time instance*, through the hidden variable  $z_{i,t}$ , and the collection of assignment variables  $Z = \{z_{i,t}\}$  is modeled as a MRF to ensure both spatial and temporal smoothness. The model equations are

$$\begin{cases} x_t^{(j)} = A^{(j)} x_{t-1}^{(j)} + v_t^{(j)} & , j \in \{1, \dots, K\} \\ y_{i,t} = C_i^{(z_{i,t})} x_t^{(z_{i,t})} + w_{i,t} + \gamma_i^{(z_{i,t})} & , i \in \{1, \dots, N\} \end{cases}$$

where  $C_i^{(j)} \in \mathbb{R}^{1 \times n}$ ,  $v_t^{(j)} \sim \mathcal{N}(0, Q^{(j)})$ , and  $x_1^{(j)} \sim \mathcal{N}(\mu^{(j)}, Q^{(j)})$  are the same as the LDT. For the TS-LDT, the observation noise processes is now distributed as  $w_{i,t} \sim \mathcal{N}(0, r^{(z_{i,t})})$ , and the mean value,  $\gamma_i^{(j)} \in \mathbb{R}$ , for pixel  $i$



in layer  $j$  is now explicitly included. Note that we must specify the mean for each layer, since a pixel may switch between layers at any time. Finally, each frame of the 3D MRF grid has the same structure as the LDT MRF, with additional edges connecting nodes between frames (e.g.  $z_{i,t}$  and  $z_{i,t+1}$ ) according to the potential function

$$V_{t,t'}(z_{i,t}, z_{i,t'}) = \begin{cases} \beta_1, & z_{i,t} = z_{i,t'} \\ \beta_2, & z_{i,t} \neq z_{i,t'} \end{cases}. \quad (17)$$

#### 4.1. Parameter estimation with EM

The EM algorithm for the TS-LDT is similar to that of the LDT. The E-step computes the expectations, now conditioned on  $z_{i,t} = j$  (see [21] for derivations),

$$\begin{aligned} \hat{x}_{t|j}^{(j)} &= \mathbb{E}_{X|Y}[x_t^{(j)}], & \hat{z}_{i,t|j}^{(j)} &= \mathbb{E}_{Z|Y}[z_{i,t}^{(j)}], \\ \hat{P}_{t,t|j}^{(j)} &= \mathbb{E}_{X|Y}[P_{t,t}^{(j)}], & \hat{P}_{t,t-1|j}^{(j)} &= \mathbb{E}_{X|Y}[P_{t,t-1}^{(j)}], \\ \hat{x}_{t|i}^{(j)} &= \mathbb{E}_{X|Y, z_{i,t}=j}[x_t^{(j)}], & \hat{P}_{t,t|i}^{(j)} &= \mathbb{E}_{X|Y, z_{i,t}=j}[P_{t,t}^{(j)}]. \end{aligned} \quad (18)$$

Next, the aggregated statistics are computed

$$\begin{aligned} \phi_1^{(j)} &= \sum_{t=1}^{\tau-1} \hat{P}_{t,t}^{(j)}, & \phi_2^{(j)} &= \sum_{t=2}^{\tau} \hat{P}_{t,t}^{(j)}, \\ \Phi_i^{(j)} &= \sum_{t=1}^{\tau} \hat{z}_{i,t}^{(j)} \hat{P}_{t,t|i}^{(j)}, & \psi^{(j)} &= \sum_{t=2}^{\tau} \hat{P}_{t,t-1}^{(j)}, \\ \hat{N}_j &= \sum_{t=1}^{\tau} \sum_{i=1}^m \hat{z}_{i,t}^{(j)}, & \xi_i^{(j)} &= \sum_{t=1}^{\tau} \hat{z}_{i,t}^{(j)} \hat{x}_{t|i}^{(j)}, \\ \Gamma_i^{(j)} &= \sum_{t=1}^{\tau} \hat{z}_{i,t}^{(j)} (y_{i,t} - \gamma_i^{(j)}) \hat{x}_{t|i}^{(j)}. \end{aligned} \quad (19)$$

In the M-step, the parameters are updated according to

$$\begin{aligned} C_i^{(j)*} &= \Gamma_i^{(j)T} \Phi_i^{(j)-1}, & A^{(j)*} &= \psi^{(j)} \phi_1^{(j)-1}, & \mu^{(j)*} &= \hat{x}_1^{(j)}, \\ r^{(j)*} &= \frac{1}{N_j} \sum_{i=1}^m \left[ \sum_{t=1}^{\tau} \hat{z}_{i,t}^{(j)} (y_{i,t} - \gamma_i^{(j)})^2 - C_i^{(j)*} \Gamma_i^{(j)} \right], \\ Q^{(j)*} &= \frac{1}{\tau} \left[ \hat{P}_{1,1}^{(j)} - \mu^{(j)*} (\mu^{(j)*})^T + \phi_2^{(j)} - A^{(j)*} \psi^{(j)T} \right], \\ \gamma_i^{(j)*} &= \frac{1}{\sum_{i=1}^m \hat{z}_{i,t}^{(j)}} \left( \sum_{t=1}^{\tau} \hat{z}_{i,t}^{(j)} y_{i,t} - C_i^{(j)} \xi_i^{(j)} \right), \end{aligned}$$

which now take into account the mean of each layer  $\gamma_i^{(j)}$ .

#### 4.2. Inference by variational approximation

Similar to the LDT, the variational approximation for the TS-LDT assumes statistical independence between pixel assignments  $z_{i,t}$  and state variables  $x^{(j)}$ , i.e.

$$q(X, Z) = \prod_{j=1}^K q(x^{(j)}) \prod_{i=1}^m \prod_{t=1}^{\tau} q(z_{i,t}). \quad (20)$$

The optimal factorial distributions are (derivations in [21])

$$\begin{aligned} \log q(x^{(j)}) &= \sum_{t=1}^{\tau} \sum_{i=1}^m h_{i,t}^{(j)} \log p(y_{i,t} | x_t^{(j)}, z_{i,t} = j) \\ &\quad + \log p(x^{(j)}) - \log Z_q^{(j)}, \end{aligned} \quad (21)$$

$$\log q(z_{i,t}) = \sum_{j=1}^K z_{i,t}^{(j)} \log h_{i,t}^{(j)}, \quad (22)$$

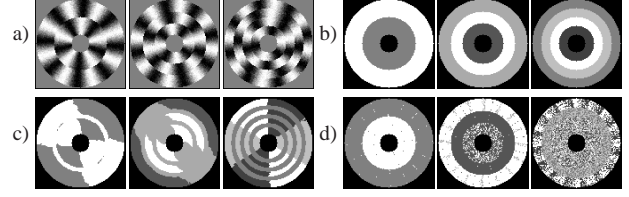


Figure 3. Segmentation of synthetic circular motion: a) video frames; segmentations using: b) LDT, c) DTM, and d) GPCA.

where  $Z_q^{(j)}$  is a normalization constant,  $h_{i,t}^{(j)}$  are the variational parameters

$$h_{i,t}^{(j)} = \mathbb{E}_q[z_{i,t}^{(j)}] = \frac{\alpha_{i,t}^{(j)} g_{i,t}^{(j)}}{\sum_{k=1}^K \alpha_{i,t}^{(k)} g_{i,t}^{(k)}}, \quad (23)$$

$$\begin{aligned} \log g_{i,t}^{(j)} &= \mathbb{E}_q \left[ \log p(y_{i,t} | x_t^{(j)}, z_{i,t} = j) \right] \\ &\quad + \sum_{(i,i') \in \mathcal{E}_t} h_{i',t}^{(j)} \log \frac{\gamma_1}{\gamma_2} + \sum_{(t,t') \in \mathcal{E}_i} h_{i,t'}^{(j)} \log \frac{\beta_1}{\beta_2}, \end{aligned} \quad (24)$$

and  $\mathbb{E}_q$  is the expectation with respect to  $q(X, Z)$ . Note that the TS-LDT variational parameters are similar to those of the LDT, except that (24) now also includes a boost of  $\log \frac{\beta_1}{\beta_2}$  from pixels in adjacent frames, within the same layer. The optimal  $q^*(X, Z)$  is used to approximate the E-step and MAP segmentation, in a manner similar to the LDT.

### 5. Application to motion segmentation

In this section, we present experiments on motion segmentation of both synthetic and real video using the LDT. All segmentations were obtained by learning an LDT with the variational EM algorithm, and computing the posterior layer assignments  $Z^* = \operatorname{argmax}_Z p(Z|Y)$  with the variational approximation. We compare the LDT segmentations with those produced by various state-of-the-art methods in the literature, including DTM [13] with a patch-size of  $5 \times 5$ , generalized PCA (GPCA) [10], and level-sets with Ising models [11] (for  $K = 2$  only). Segmentations are evaluated by computing the Rand index [23], which is a measure of clustering performance, with the ground-truth. We begin by presenting results on a synthetic textures containing circular motion, followed by quantitative results on the database from [13], and conclude with results on real video. Videos of all results are available in [24].

#### 5.1. Results on synthetic circular motion

We first demonstrate LDT segmentation of sequences containing several rings of distinct circular motion, as shown in Figure 3a. Each video contains 2, 3, or 4 circular rings, with each ring rotating at a different speed. The sequences were segmented with LDT, DTM, and GPCA with

Method	$K = 2$	$K = 3$	$K = 4$
LDT	<b>0.944</b> (05)	<b>0.894</b> (12)	<b>0.916</b> (20)
DTM [13]	0.912 (17)	0.844 (15)	0.857 (15)
Ising [11]	0.927 (12)	n/a	n/a
AR [11]	0.922 (10)	n/a	n/a
AR0 [11]	0.917 (20)	n/a	n/a
GPCA [10]	0.538 (02)	0.518 (10)	0.538 (10)

Table 1. Average Rand index for various segmentation algorithms on the synthetic texture database.

$n = 2$ . LDT (Figure 3b) correctly segments all the rings, favoring global homogeneity over localized grouping of segments by texture orientation. On the other hand, DTM (Figure 3c) tends to find incorrect segmentations based on local direction of motion. In addition, on the 4-ring video, DTM incorrectly assigns one segment to the boundaries between rings, illustrating how the poor boundary accuracy of the patch-based segmentation framework can create substantial problems. Finally, GPCA (Figure 3d) is able to correctly segment 2 rings, but fails when there are more. In these cases, GPCA correctly segments one of the rings, but randomly segments the remainder of the video. These results illustrate how LDT can correctly segment sequences whose motion is globally (at the ring level) homogeneous, but locally (at the patch level) heterogeneous. Both DTM and GPCA fail to exhibit this property. Quantitatively, this is reflected by the much higher average Rand scores of the segmentations produced by LDT (1.00, as compared to 0.491 for DTM, and 0.820 for GPCA).

## 5.2. Results on synthetic texture database

We next present results on the synthetic texture database from [13], which contains 299 sequences with  $K = \{2, 3, 4\}$  regions of different video textures (e.g. water, fire, vegetation), as illustrated in Figure 5a. In [13], the database was segmented with DTM, using a fixed initial contour. Although DTM was shown to be superior to other state-of-the-art methods [11, 10], the segmentations contain some errors due to the poor boundary localization discussed above. In this experiment, we show that using the LDT to refine the DTM segmentations *substantially improves* the results from [13]. For comparison, we apply the level-set methods of [11], Ising, AR (auto-regressive models), and AR0 (AR with zero-mean), also initializing with the DTM segmentation. We also compare with GPCA [10], which requires no initialization. Each method was run for several values of  $n$ , and the average Rand index was computed for each  $K$ . No post-processing was applied to the segmentations. We note that this large-scale experiment on the LDT, with hundreds of video, is infeasible with the Gibbs sampler [12], where EM runs for several hours. The variational approximation is significantly faster, with EM taking only a few minutes.

Table 1 shows the performance obtained, with the best  $n$ ,

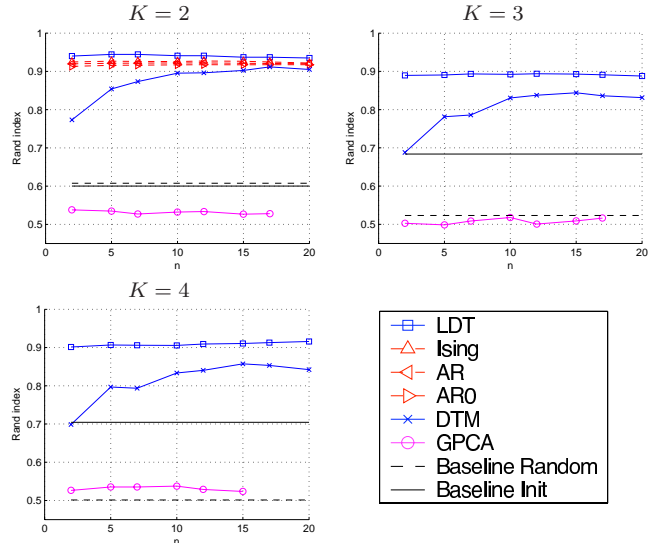


Figure 4. Texture database results: average Rand index vs. the state-space dimension ( $n$ ) for video with  $K = \{2, 3, 4\}$  textures.

by each algorithm. It is clear that LDT segmentation significantly improves the initial segmentation produced by DTM: the average Rand increases from 0.912 to 0.944, from 0.844 to 0.894, and from 0.857 to 0.916, for  $K = \{2, 3, 4\}$  respectively. LDT also performs best among all algorithms, with Ising as the closest competitor (Rand 0.927). Figure 4 shows a plot of the Rand index versus the dimension  $n$  of the segmentation models, demonstrating that LDT segmentation is robust to the choice of  $n$ .

Qualitatively, LDT improves the DTM segmentation in three ways: 1) segmentation boundaries are more precise, due to the region-level modeling (rather than patch-level); 2) segmentations are less noisy, due to the inclusion of the MRF prior; and 3) gross errors, e.g. texture borders marked as segments, are eliminated. Several examples of these improvements are presented in Figures 5b and 5c. From left to right, the first example is a case where the LDT corrects a noisy DTM segmentation (imprecise boundaries and spurious segments). The second and third examples are cases where the DTM produces a poor segmentation (e.g. the border between two textures is erroneously marked as a segment), which the LDT corrects. The final two examples are very difficult cases. In the fourth example, the initial DTM segmentation is very poor. Albeit a substantial improvement, the LDT segmentation is still noisy. In the fifth example, the DTM splits the two water segments incorrectly (the two textures are very similar). The LDT substantially improves the segmentation, but the difficulties due to great similarity of water patterns prove too difficult to overcome completely.

Finally, we present results on the ocean-fire video from [8], which contains a water background and moving regions of fire in the foreground, in Figure 6. The video was segmented with the TS-LDT, using the DTM segmentation

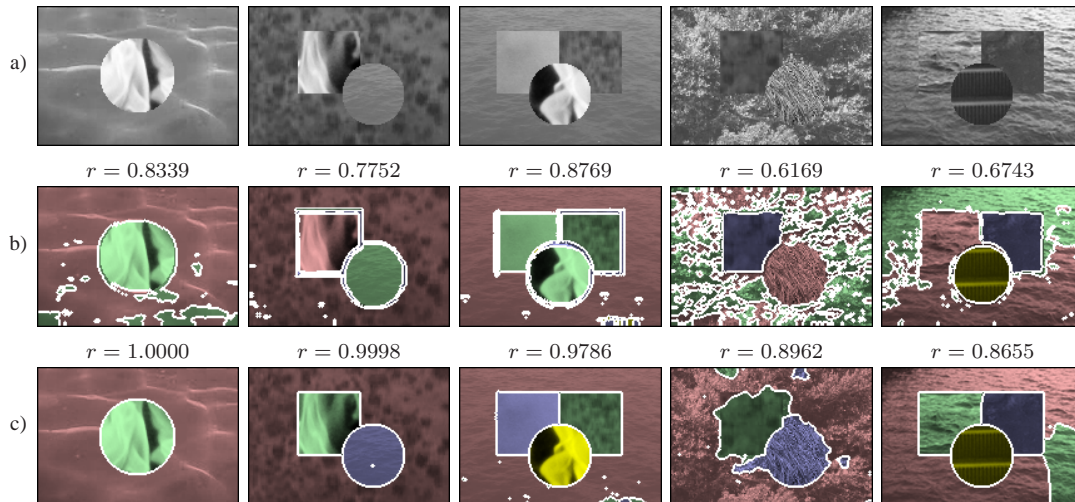


Figure 5. Texture database examples: a) video frames; the b) DTM and c) LDT segmentations.  $r$  is the the Rand index of the segmentation.

as initialization. The TS-LDT successfully segments the changing regions of fire. In addition, the TS-LDT improves localization (tighter boundaries) and corrects noise (spurious segments) of the DTM segments.

### 5.3. Results on real video

We conclude the segmentation experiments with results on real video sequences. Figure 7a presents the segmentation of a moving ferris wheel, using LDT and DTM for  $K = \{2, 3\}$ . For  $K = 2$ , both LDT and DTM segment the static background from the moving ferris wheel. However, for 3 regions, the plausible segmentation, by LDT, of the foreground into two regions corresponding to the ferris wheel and a balloon moving in the wind, is not matched by DTM. Instead, DTM segments the ferris wheel into two regions, according to the dominant direction of its local motion (either moving up or down), ignoring the balloon motion. This is identical to the problems found for the synthetic sequences of Figure 3: *the inability to uncover global homogeneity when the video is locally heterogeneous*. On the other hand, the preference of LDT for two regions of very different sizes, illustrates its robustness to this problem. The strong local heterogeneity of the optical flow in the region of the ferris wheel is well explained by the global homogeneity of the corresponding layer dynamics. Figure 7b shows another example of this phenomenon. For 3 regions, LDT segments the windmill into regions corresponding to the moving fan blades, parts of the shaking tail piece, and the background. When segmenting into 4 regions, LDT splits the fan blade segment into two regions, which correspond to the fan blades and the internal support pieces. On the other hand, the DTM segmentations for  $K = \{3, 4\}$  split the fan blades into different regions based on the orientation (vertical or horizontal) of the optical flow. Additional

segmentations, which further demonstrate the robustness of the LDT and its applicability to a wide range of scenes, are available from [24].

## 6. Conclusions

In this work, we proposed a variational approximation for inference on the LDT, which enables efficient learning of the model parameters. We further proposed an extension of the LDT, the temporally-switching LDT that can model changes in the shape of each layer over time. We have conducted extensive experiments, with both synthetic mosaics of real textures and real video sequences, that tested the ability of the variational LDT to segment video into regions of coherent dynamics and appearance. The variational LDT has been shown to outperform a number of state-of-the-art methods for video segmentation. In particular, it was shown to possess a unique ability to group regions of *globally homogeneous but locally heterogeneous stochastic dynamics*. We believe that this ability is unmatched by any video segmentation algorithm currently available in the literature. The new method also has consistently produced segmentations with better spatial-localization than those possible with the *localized representations*, such as the mixture of dynamic textures, that have previously been prevalent in the area of dynamic texture segmentation.

## Acknowledgements

This work was partially funded by NSF awards IIS-0534985, IIS-0448609, and DGE- 0333451.

## References

- [1] B. K. P. Horn, *Robot Vision*. McGraw-Hill Book Company, New York, 1986.



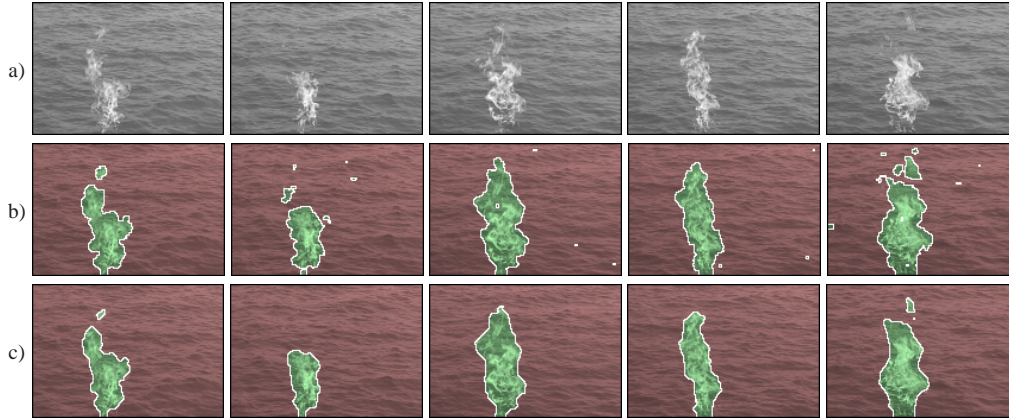


Figure 6. Segmentation of ocean-fire from [8]: (a) video frames; and the (b) DTM, and (c) TS-LDT segmentations.

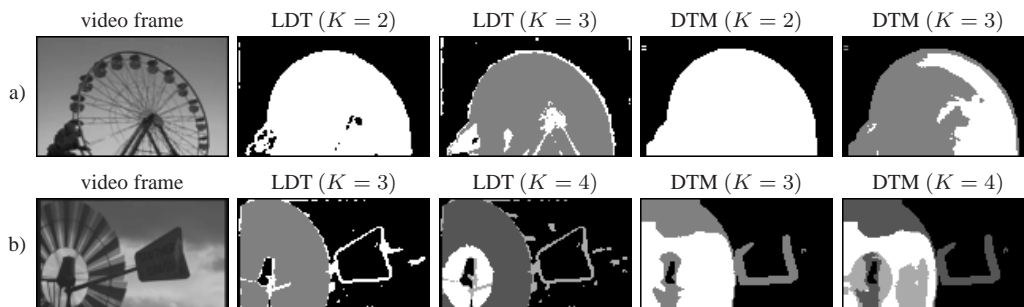


Figure 7. Segmentation of a (a) ferris wheel and (b) windmill using LDT and DTM, for different number of segments ( $K$ ).

- [2] B. Horn and B. Schunk, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–204, 1981.
- [3] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. DARPA Image Understanding Workshop*, 1981, pp. 121–130.
- [4] J. Barron, D. Fleet, and S. Beauchemin, "Performance of optical flow techniques," *Intl. J. of Computer Vision*, vol. 12, pp. 43–77, 1994.
- [5] J. Wang and E. Adelson, "Representing moving images with layers," *IEEE Trans. on Image Proc.*, vol. 3, no. 5, pp. 625–38, 1994.
- [6] B. Frey and N. Jovic, "Estimating mixture models of images and inferring spatial transformations using the EM algorithm," in *CVPR*, 1999, pp. 416–22.
- [7] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto, "Dynamic textures," *Intl. J. Computer Vision*, vol. 51, no. 2, pp. 91–109, 2003.
- [8] G. Doretto, D. Cremers, P. Favaro, and S. Soatto, "Dynamic texture segmentation," in *ICCV*, vol. 2, 2003, pp. 1236–42.
- [9] P. Saisan, G. Doretto, Y. Wu, and S. Soatto, "Dynamic texture recognition," in *CVPR*, vol. 2, 2001, pp. 58–63.
- [10] R. Vidal and A. Ravichandran, "Optical flow estimation & segmentation of multiple moving dynamic textures," in *CVPR*, vol. 2, 2005, pp. 516–21.
- [11] A. Ghoreyshi and R. Vidal, "Segmenting dynamic textures with Ising descriptors, ARX models and level sets," in *Dynamical Vision Workshop in the European Conf. on Computer Vision*, 2006.
- [12] A. B. Chan and N. Vasconcelos, "Layered dynamic textures," in *Neural Information Processing Systems 18*, 2006, pp. 203–10.
- [13] —, "Modeling, clustering, and segmenting video with mixtures of dynamic textures," *IEEE Trans. PAMI*, vol. 30, no. 5, pp. 909–926, May 2008.
- [14] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice-Hall, 1993.
- [15] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society B*, vol. 39, pp. 1–38, 1977.
- [16] Y. Wu, G. Hua, and T. Yu, "Switching observation models for contour tracking in clutter," in *CVPR*, 2003, pp. 295–302.
- [17] V. Pavlović, J. Rehg, and J. MacCormick, "Learning switching linear models of human motion," in *NIPS*, 2000.
- [18] S. M. Oh, J. M. Rehg, T. Balch, and F. Dellaert, "Learning and inferring motion patterns using parametric segmental switching linear dynamic systems," *IJCV*, vol. 77, no. 1-3, pp. 103–24, 2008.
- [19] Z. Ghahramani and G. E. Hinton, "Variational learning for switching state-space models," *Neural Comp.*, vol. 12, no. 4, pp. 831–64, 2000.
- [20] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
- [21] A. B. Chan and N. Vasconcelos, "Derivations for the layered dynamic texture and temporally-switching layered dynamic texture," Statistical Visual Computing Lab, Tech. Rep. SVCL-TR-2009-01, 2009.
- [22] J. Besag, "Spatial interaction and the statistical analysis of lattice systems," *Journal of the Royal Statistical Society, Series B (Methodological)*, vol. 36, no. 2, pp. 192–236, 1974.
- [23] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, pp. 193–218, 1985.
- [24] "Layered dynamic textures." [Online]. Available: <http://www.svcl.ucsd.edu/projects/layerdytex>