Layered Dynamic Textures

Antoni B. Chan and Nuno Vasconcelos
Department of Electrical and Computer Engineering
University of California, San Diego
abchan@ucsd.edu, nuno@ece.ucsd.edu

Abstract

A dynamic texture is a generative model for video that treats the video as a sample from spatio-temporal stochastic process. One problem associated with the dynamic texture is that it cannot model video where there are regions of motion with different dynamics, e.g. a scene with smoke and fire. In this work, we introduce the layered dynamic texture model, which addresses this problem by introducing a separate state process for each region of motion. We derive the EM algorithm for learning the parameters of the model, and demonstrate the efficacy of the proposed model for the tasks of segmentation and synthesis of video.

1. Introduction

Traditional motion representations, based on optic 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 [1–4], but this introduces undesirable smoothing across motion edges or regions where the motion is, by definition, not smooth (e.g. vegetation in outdoors scenes). 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), 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 layers showed most promise as models for scenes composed of ensembles of objects subject to homogeneous motion (e.g. leaves blowing in the wind, a flock of birds, a picket fence, 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 [7–10]. This work has demonstrated that modeling both the dynamics and appearance of video as stochastic quantities leads to a much more powerful generative model for video than that of a "cardboard" figure subject to parametric motion. In fact, the dynamic texture model 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 of the dynamic texture framework, however, is 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, video containing both trees in the background and people in the foreground, and so forth. In such cases, the existing dynamic texture model is inherently incorrect, since it must represent multiple motion fields with a single dynamic process.

In this work, we address this limitation by introducing a new generative model for video, which we denote by the *layered dynamic texture* (LDT). This consists of augmenting the dynamic texture with a discrete *hidden* variable, that enables the assignment of different dynamics to different regions of the video. Conditioned on the state of this hidden variable, the video is then modeled as a simple dynamic texture. By introducing a shared dynamic representation for all the pixels in the same region, the new model is a layered representation. When compared with traditional layered models, it replaces the process of layer formation based on "warping of cardboard figures" with one based on sampling from the generative model (for both dynamics and appearance) provided by the dynamic texture. This enables a much richer video representation. Since each layer is a dynamic texture, the model can also be seen as a multi-state dynamic texture, which is capable of assigning different dynamics and appearance to different image regions.

Recently, some of the limitations of the dynamic tex-

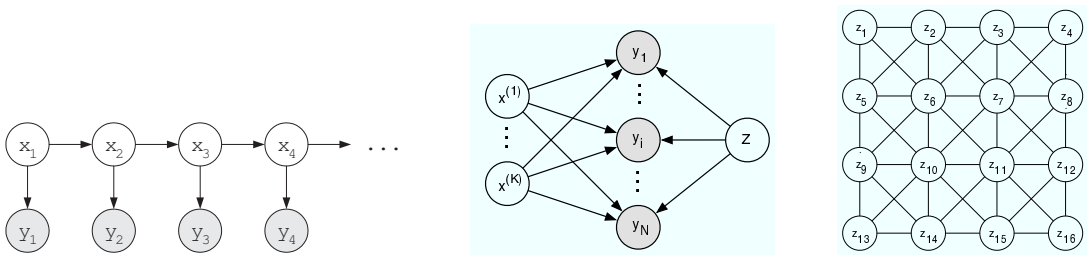


Figure 1. (a) The graphical model for the dynamic texture. x_t is the hidden state at time t, and y_t is the observed frame at time t; (b) The graphical model for the layered dynamic texture. y_i is an observed pixel process and $x^{(j)}$ is a hidden state process. Z is the collection of layer assignment variables z_i that assigns each pixels to one of the state processes, and is modeled as an MRF; (c) An example of a 4×4 MRF used for layer assignment.

ture were also addressed in [11]. The layered formulation now proposed has various differences with respect to this work. First, it enables probabilistic pixel assignments, as opposed to the hard assignments of [11]. Second, the learning method of [11] is exact only when the noise term of the appearance component of the dynamic texture is zero. As usual in computer vision, allowing this term to be different than zero is important not only because it enables the processing of video with noise, but also because it introduces flexibility with respect to model mismatches. Finally, the model of [11] does not enforce spatial consistency of the assignments of pixels to regions. We show that this can be naturally done with the layered dynamic texture model, and can lead to significant improvements of segmentation accuracy when the different regions have similar dynamics.

The paper is organized as follows. In Section 2, we introduce the layered dynamic texture model. In Section 3 we present the EM algorithm for learning the model from training data. Finally, in Section 4 we present an experimental evaluation in the context of segmentation and video synthesis.

2. Layered dynamic textures

We start with a brief review of dynamic textures, and then introduce the layered dynamic texture model.

2.1. Dynamic texture

A dynamic texture [7] is a generative model for video, which treats the video as a sample from a linear dynamical system. The model, shown in Figure 1 (a), separates the visual component and the underlying dynamics into two stochastic processes. The dynamics of the video are represented as a time-evolving state process $x_t \in \mathbb{R}^n$, and the appearance of the frame $y_t \in \mathbb{R}^m$ is a linear function of the current state vector with some observation noise. Formally, the system is described by

$$\begin{cases} x_t = Ax_{t-1} + Bv_t \\ y_t = Cx_t + \sqrt{r}w_t \end{cases} \quad (1)$$

where $A \in \mathbb{R}^{n \times n}$ is a transition matrix, $C \in \mathbb{R}^{m \times n}$ a transformation matrix, $Bv_t \sim_{iid} \mathcal{N}(0, Q,)$ and $\sqrt{r}w_t \sim_{iid} \mathcal{N}(0, rI_m)$ the state and observation noise processes parameterized by $B \in \mathbb{R}^{n \times n}$ and $r \in \mathbb{R}$, and the initial state $x_0 \in \mathbb{R}^n$ is a constant. One interpretation of the dynamic texture model is that the columns of C are the principal components of the video frames, and the state vectors are the PCA coefficients for each video frame. This is the case when the model is learned with the method of [7].

An alternative interpretation considers a single pixel as it evolves over time. Each coordinate of the state vector x_t defines a one-dimensional random trajectory in time. A pixel is then represented as a weighted sum of random trajectories, where the weighting coefficients are contained in the corresponding row of C. This is analogous to the discrete Fourier transform in signal processing, where a signal is represented as a weighted sum of complex exponentials although, for the dynamic texture, the trajectories are not necessarily orthogonal. This interpretation illustrates the ability of the dynamic texture to model the same motion under different intensity levels (e.g. cars moving from the shade into sunlight) by simply scaling the rows of C. Regardless of interpretation, the simple dynamic texture model has only one state process, which restricts the efficacy of the model to video where the motion is homogenous.

2.2. Layered dynamic textures

We now introduce the *layered dynamic texture* (LDT), which is shown in Figure 1 (b). The model addresses the limitations of the dynamic texture by relying on a set of state processes $X = \{x^{(j)}\}_{j=1}^{K}$ to model different video dynamics. The layer assignment variable z_i assigns pixel y_i to one of the state processes (layers), and conditioned on the layer assignments, the pixels in the same layer are modeled as a dynamic texture. In addition, the collection of layer assignments $Z = \{z_i\}_{i=1}^{N}$ is modeled as a Markov random field (MRF) to ensure spatial layer consistency (an example is shown in Figure 1 (c)). The linear system equations for

the layered dynamic texture are

$$\begin{cases} x_t^{(j)} = A^{(j)} x_{t-1}^{(j)} + B^{(j)} v_t^{(j)} & j \in \{1, \cdots, K\} \\ y_{i,t} = C_i^{(z_i)} x_t^{(z_i)} + \sqrt{r^{(z_i)}} w_{i,t} & i \in \{1, \cdots, N\} \end{cases} \quad (2)$$

where $C_i^{(j)} \in \mathbb{R}^{1 \times n}$ is the transformation from the hidden state to the observed pixel domain for each pixel y_i and each layer j, the noise parameters are $B^{(j)} \in \mathbb{R}^{n \times n}$ and $r^{(j)} \in \mathbb{R}$, the iid noise processes are $w_{i,t} \sim_{iid} \mathcal{N}(0, 1)$ and $v_t^{(j)} \sim_{iid} \mathcal{N}(0, I_n)$, and the initial states are drawn from $x_1^{(j)} \sim \mathcal{N}(\mu^{(j)}, S^{(j)})$.

As a generative model, the layered dynamic texture assumes that the state processes X and the layer assignments Z are independent, i.e. the layer motion is independent of layer location, and vice versa. Given the layer assignments, the LDT is a collection of dynamic textures over different regions of the video. As a result, learning the LDT reduces to learning several dynamic textures, when given the segmentation of the video into regions of distinct motion. For the more general case, the segmentation and the dynamics can be learned simultaneously using the EM algorithm.

2.3. Modeling layer assignments

An MRF is used to model the layer assignments to ensure spatial consistency of the layer (see Figure 1 (c) for an example of the grid). The MRF has the following distribution

$$p(Z) = \frac{1}{\mathcal{Z}} \prod_i \psi_i(z_i) \prod_{(i,j) \in \mathcal{E}} \psi_{i,j}(z_i, z_j) \quad (3)$$

where $\mathcal{E}$ is the set of edges in the MRF grid, $\mathcal{Z}$ a normalization constant (partition function), and ψ_i and $\psi_{i,j}$ potential functions of the form

$$\psi_i(z_i) = \begin{cases} \alpha_1 & , z_i = 1 \\ \vdots & \vdots \\ \alpha_K & , z_i = K \end{cases} \quad (4)$$

$$\psi_{i,j}(z_i, z_j) = \begin{cases} \gamma_1 & , z_i = z_j \\ \gamma_2 & , z_i \neq z_j \end{cases} \quad (5)$$

The potential function ψ_i defines a prior likelihood for each layer, while $\psi_{i,j}$ attributes higher probability to configurations where neighboring pixels are in the same layer. Rather than learn the parameters of the potential functions for each model, we will treat the MRF as a prior on Z that regularizes the smoothness of the layers.

3. Parameter estimation using EM

The parameters of the layered dynamic texture are learned using the Expectation-Maximization (EM) algorithm [12], which iterates between estimating hidden state variables X and hidden layer assignments Z from the current parameters, and updating the parameters given the current hidden variable estimates. One iteration of the EM algorithm contains the following two steps

- E-Step: $\mathcal{Q}(\Theta; \hat{\Theta}) = \mathrm{E}_{X,Z|Y;\hat{\Theta}}(\log p(X, Y, Z; \Theta))$
- M-Step: $\hat{\Theta}^* = \mathrm{argmax}_{\Theta} \mathcal{Q}(\Theta; \hat{\Theta})$

In the remainder of this section, we derive the joint log-likelihood of the model, followed by the derivations of the E-step and M-step of the learning algorithm. See Table 1 for notation.

N	number of pixels in a frame
τ	length of the observed video sequence
K	number of state processes
i	index over the pixel sequences
j	index over the state processes
t	time index of a sequence
y_i	the i^{th} pixel sequence
$y_{i,t}$	the observation at time t of y_i
$x^{(j)}$	the j^{th} state sequence
$x_t^{(j)}$	the state at time t of $x^{(j)}$
z_i	the layer assignment variable for y_i
$z_i^{(j)}$	the indicator variable that y_i is labeled j

Table 1. Notation for EM for layered dynamic textures

3.1. Log-likelihood

The state processes X and layer assignments Z are independent, and hence the joint log-likelihood factors as

$$\begin{aligned} \ell(X, Y, Z) &= \log p(X, Y, Z) && (6) \\ &= \log p(Y|X, Z) + \log p(X) + \log p(Z) && (7) \\ &= \sum_{i,j} z_i^{(j)} \log p(y_i | x^{(j)}, z_i = j) && (8) \\ &\quad + \sum_j \log p(x^{(j)}) + \log p(Z) \\ &= \sum_{i,j} z_i^{(j)} \sum_{t=1}^{\tau} \log p(y_{i,t} | x_t^{(j)}, z_i = j) && (9) \\ &\quad + \sum_j \left(\sum_{t=2}^{\tau} \log p(x_t^{(j)} | x_{t-1}^{(j)}) + \log p(x_1^{(j)}) \right) \\ &\quad + \log p(Z) \end{aligned}$$

where $z_i^{(j)}$ is the indicator variable that $z_i = j$. Substituting for the probability distributions and dropping the constant terms yields the log-likelihood given in (20).

3.2. E-Step

Taking the conditional expectation of (20), the E-step requires the computation of the following terms:

$$\begin{aligned}
\hat{x}_t^{(j)} &= \mathrm{E}_{X|Y}(x_t^{(j)}) \\
\hat{x}_{i,t}^{(j)} &= \mathrm{E}_{Z,X|Y}(z_i^{(j)} x_t^{(j)}) \\
\hat{P}_{t,t}^{(j)} &= \mathrm{E}_{X|Y}(x_t^{(j)} (x_t^{(j)})^T) \\
\hat{P}_{i|t,t}^{(j)} &= \mathrm{E}_{Z,X|Y}(z_i^{(j)} x_t^{(j)} (x_t^{(j)})^T) \\
\hat{P}_{t,t-1}^{(j)} &= \mathrm{E}_{X|Y}(x_t^{(j)} (x_{t-1}^{(j)})^T) \\
\hat{z}_i^{(j)} &= \mathrm{E}_{Z|Y}(z_i^{(j)}) = p(z_i = j|Y)
\end{aligned} \tag{10}$$

These expectations are intractable to compute in closed-form since it is not known to which state process each of the pixels y_i is assigned, and hence it is necessary to marginalize over all configurations of Z. This problem also appears for the computation of the posterior layer assignment probability $p(z_i = j|Y)$. While other inference approximation methods, e.g. variational methods or belief propagation, could be used, the current method that we adopt for approximating these expectations is to simply average over draws from the posterior $p(X, Z|Y)$ using a Gibbs sampler (see Appendix for details).

3.3. M-Step

The optimization in the M-Step is obtained by taking the partial derivative of the $\mathcal{Q}$ function with respect to each of the parameters. For convenience, we first define the following quantities,

$$\begin{array}{ll}
\phi_1^{(j)} = \sum_{t=1}^{\tau-1} \hat{P}_{t,t}^{(j)} & \phi_2^{(j)} = \sum_{t=2}^{\tau} \hat{P}_{t,t}^{(j)} \\
\Phi^{(j)} = \sum_{t=1}^{\tau} \hat{P}_{t,t}^{(j)} & \Phi_i^{(j)} = \sum_{t=1}^{\tau} \hat{P}_{i|t,t}^{(j)} \\
\psi^{(j)} = \sum_{t=2}^{\tau} \hat{P}_{t,t-1}^{(j)} & \Gamma_i^{(j)} = \sum_{t=1}^{\tau} y_{i,t} \hat{x}_{i,t}^{(j)} \\
\hat{N}_j = \sum_i \hat{z}_i^{(j)} & \Lambda_i^{(j)} = \sum_{t=1}^{\tau} \hat{z}_i^{(j)} y_{i,t}^2
\end{array} \tag{11}$$

Taking the partial derivative with respect to each parameter and setting to zero yields the parameter updates:

$$\begin{aligned}
A^{(j)*} &= \psi^{(j)} (\phi_1^{(j)})^{-1} \\
Q^{(j)*} &= \frac{1}{\tau - 1} (\phi_2^{(j)} - A^{(j)*} (\psi^{(j)})^T) \\
\mu^{(j)*} &= \hat{x}_1^{(j)} \\
S^{(j)*} &= \hat{P}_{1,1}^{(j)} - \mu^{(j)*} (\mu^{(j)*})^T \\
C_i^{(j)*} &= (\Gamma_i^{(j)})^T (\Phi_i^{(j)})^{-1} \\
r^{(j)*} &= \frac{1}{\tau \hat{N}_j} \sum_{i=1}^{N} (\Lambda_i^{(j)} - C_i^{(j)*} \Gamma_i^{(j)})
\end{aligned} \tag{12}$$

The M-step parameter updates are analogous to those required to learn a regular linear dynamical system [13, 14], with minor modifications for transformation matrices and observation noise.

4. Experiments

In this section, we show the efficacy of the proposed model for segmentation and synthesis of several videos with multiple regions of distinct motion. Figure 2 (a) shows the three video sequences used in testing. The first (top) is a composite of three distinct video textures of water, smoke, and fire. The second (middle) is of laundry spinning in a dryer. The laundry in the bottom left of the video is spinning in place in a circular motion, and the laundry around the outside is spinning faster. The final video (bottom) is of a highway [15] where the traffic in each lane is traveling at a different speed. The first, second and fourth lanes (from left to right) move faster than the third and fifth. All three videos have multiple regions of motion and are therefore properly modeled by the models proposed in this paper, but not by a regular dynamic texture.

A layered dynamic texture (LDT) was fit to each of the three videos. For comparison, a layered dynamic texture with the layer assignments z_i distributed as iid multinomials (LDT-iid) was also learned. In all the experiments, the dimension of the state space was $n = 10$. The MRF grid was based on the eight-neighbor system (with cliques of size 2), and the parameters of the potential functions were $\gamma_1 = 0.99$, $\gamma_2 = 0.01$, and $\alpha_j = 1/K$. The expectations required by the EM algorithm were approximated using Gibbs sampling. We first present segmentation results, to show that the models can effectively separate layers with different dynamics, and then discuss results relative to video synthesis from the learned models.

4.1. Segmentation

The videos were segmented by assigning each of the pixels to the most probable layer conditioned on the observed video, i.e. $z_i^* = \mathrm{argmax}_j \, p(z_i = j|Y)$. Another possibility would be to assign the pixels by maximizing the posterior of all the pixels $p(Z|Y)$. While this maximizes the true posterior, in practice we obtained similar results with the two methods. The former method was chosen because the individual posterior distributions are already computed during the E-step of EM.

Figures 2 (b) and (c) show the segmentation results obtained using the LDT and LDT-iid models, respectively. The segmented video is also available at [16]. From the segmentations produced by LDT-iid, it can be concluded that the laundry video can be reasonably well segmented without the MRF prior. The segmentation of the composite video using LDT-iid is slightly worse, and contains several regions of noise. Nonetheless, this confirms the intuition that the various video regions contain very distinct dynamics that can only be modeled with separate state processes. Otherwise, the pixels should be either randomly assigned among the various layers, or uniformly assigned to one of

them. The segmentations of the traffic video using LDT-iid are poor. While the dynamics are different, the differences are significantly more subtle, and segmentation requires stronger enforcement of layer consistency. As expected, the introduction of the MRF prior improves the segmentations for all three videos. For example, in the composite sequence all erroneous segments in the water region are removed, and in the traffic sequence, most of the speckled segmentation also disappears.

In terms of the overall segmentation quality, the LDT is able to segment the composite video perfectly. The segmentation of the laundry video is plausible, as the laundry tumbling around the edge of the dryer moves faster than that spinning in place. The model also produces a reasonable segmentation of the traffic video, with the segments roughly corresponding to the different lanes of traffic. Much of the errors correspond to regions that either contain intermittent motion (e.g. the region between the lanes) or almost no motion (e.g. truck in the upper-right corner and flat-bed truck in the third lane). Some of these errors could be eliminated by filtering the video before segmentation, but we have attempted no pre or post-processing. Finally, we note that the laundry and traffic videos are not trivial to segment with standard computer vision techniques, namely methods based on optical flow. This is particularly true in the case of the traffic video where the abundance of straight lines and flat regions makes computing the correct optical flow difficult due to the aperture problem.

4.2. Synthesis

The layered dynamic texture is a generative model, and hence a video can be synthesized by drawing a sample from the learned model. A synthesized composite video comparing the LDT and the normal dynamic texture can be found at [16]. When modeling a video with multiple motions, the regular dynamic texture will average different dynamics. This is noticeable in the synthesized video, where the fire region does not flicker at the same speed as in the original video. Furthermore, the motions in different regions are coupled, e.g. when the fire begins to flicker faster, the water region ceases to move smoothly. In contrast, the video synthesized from the layered dynamic texture is more realistic, as the fire region flickers at the correct speed, and the different regions follow their own motion patterns.

5. Conclusions and Future Work

In this paper we have introduced a new model, the layered dynamic texture, that can model video that contains regions of motion with different dynamics. For this class of video, we showed that the layered dynamic texture is more appropriate for synthesis than the regular dynamic texture. In addition, the model provides a natural framework for segmenting video into regions of motion. One disadvantage of the model is that the current implementation of the E-step in the learning algorithm requires sampling methods, which are computationally intensive. Future work will be directed towards faster approximation methods, such as variational approximation or belief propagation.

Appendix

A sample from the layered dynamic texture can be obtained using the Gibbs sampler [17], which is a method for sampling from complicated probability distributions. Noting that it is much easier to sample conditionally from the *collection* of variables X and Z than on any individual $x^{(j)}$ or $z_i^{(j)}$, the Gibbs sampler is first initialized with $X \sim p(X)$, followed by alternating between sampling from $Z \sim p(Z|X,Y)$ and sampling from $X \sim p(X|Y,Z)$.

The layer assignment distribution $p(Z|X,Y)$ is given by

$$\begin{aligned} p(Z|X,Y) &= \frac{p(Y|X,Z)p(X|Z)p(Z)}{p(Y|X)p(X)} && (13)\\ &\propto p(Y|X,Z)p(Z) && (14)\\ &\propto p(Z)\prod_i p(y_i|X,z_i)) && (15)\end{aligned}$$

If the z_i are modeled as independent multinomials, then sampling z_i involves sampling from the posterior of the multinomial $p(z_i|X,y_i) \propto p(y_i|X,z_i)p(z_i)$. If Z is modeled as an MRF, then the $p(y_i|X,z_i)$ terms are absorbed into the self potentials ψ_i of the MRF, and sampling can be done using the MCMC algorithm [18].

The state processes are independent of each other when conditioned on the video and the pixel assignments, i.e.

$$p(X|Y,Z) = \prod_j p(x^{(j)}|Y,Z) = \prod_j p(x^{(j)}|Y_j) \quad (16)$$

where $Y_j = \{y_i|z_i = j\}$ are all the pixels that are assigned to layer j. Using the Markovian structure of the state process, the joint probability factors into the conditionals probabilities,

$$p(x_1^{(j)},\cdots,x_\tau^{(j)}|Y_j) = p(x_1^{(j)}|Y_j)\prod_{t=2}^{\tau} p(x_t^{(j)}|x_{t-1}^{(j)},Y_j) \quad (17)$$

The parameters of each conditional Gaussian is obtained with the conditional Gaussian theorem [19],

$$\begin{aligned} &\mathrm{E}(x_t^{(j)}|x_{t-1}^{(j)},Y_j) = && (18)\\ &\quad \mu_t^{(j)} + \Sigma_{t,t-1}^{(j)}(\Sigma_{t-1,t-1}^{(j)})^{-1}(x_{t-1}^{(j)} - \mu_{t-1}^{(j)})\\ &\mathrm{cov}(x_t^{(j)}|x_{t-1}^{(j)},Y_j) = && (19)\\ &\quad \Sigma_{t,t}^{(j)} - \Sigma_{t,t-1}^{(j)}(\Sigma_{t-1,t-1}^{(j)})^{-1}\Sigma_{t-1,t}^{(j)}\end{aligned}$$

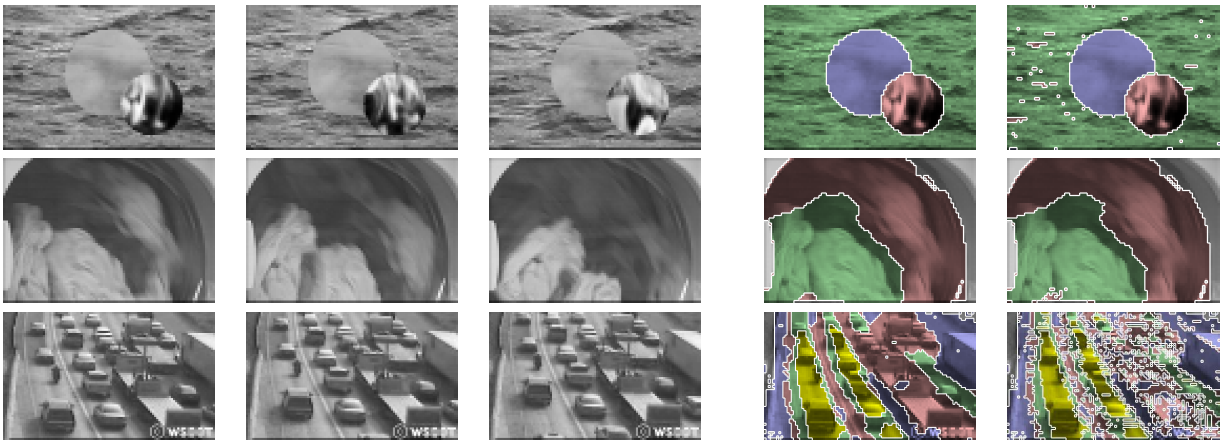
where the marginal mean, marginal covariance, and one-step covariance are $\mu_t^{(j)} = \mathrm{E}(x_t^{(j)}|Y_j)$, $\Sigma_{t,t}^{(j)} =$

$$\ell(X,Y,Z) = -\frac{1}{2}\sum_{i,j} z_i^{(j)} \sum_{t=1}^{\tau} \frac{1}{r^{(j)}} \left(y_{i,t}^2 - 2C_i^{(j)} x_t^{(j)} y_{i,t} + \mathrm{tr}\left(C_i^{(j)} x_t^{(j)} (x_t^{(j)})^T (C_i^{(j)})^T \right) \right) \tag{20}$$

$$- \frac{1}{2}\sum_{j} \mathrm{tr}\left((S^{(j)})^{-1} \left(x_1^{(j)}(x_1^{(j)})^T - x_1^{(j)}(\mu^{(j)})^T - \mu^{(j)}(x_1^{(j)})^T + \mu^{(j)}(\mu^{(j)})^T \right) \right)$$

$$- \frac{1}{2}\sum_{j}\sum_{t=2}^{\tau} \mathrm{tr}\left((Q^{(j)})^{-1} \left(x_t^{(j)}(x_t^{(j)})^T - x_t^{(j)}(x_{t-1}^{(j)})^T (A^{(j)})^T - A^{(j)} x_{t-1}^{(j)}(x_t^{(j)})^T + A^{(j)} x_{t-1}^{(j)}(x_{t-1}^{(j)})^T (A^{(j)})^T \right) \right)$$

$$- \frac{\tau}{2}\sum_{i,j} z_i^{(j)} \log r^{(j)} - \frac{\tau - 1}{2}\sum_{j} \log \left| Q^{(j)} \right| - \frac{1}{2}\sum_{j} \log \left| S^{(j)} \right| + p(Z)$$



(a) (b) (c)

Figure 2. Frames from the test video sequences (a): (top) composite of water, smoke, and fire video textures; (middle) spinning laundry in a dryer; and (bottom) highway traffic with lanes traveling at different speeds. Segmentation results for each of the test videos using: (b) the layered dynamic texture, and (c) the layered dynamic texture without MRF.

$\mathrm{cov}(x_t^{(j)}|Y_j)$, and $\Sigma_{t,t-1}^{(j)} = \mathrm{cov}(x_t^{(j)}, x_{t-1}^{(j)}|Y_j)$, which can be obtained using the Kalman smoothing filter [13, 14]. The sequence $x^{(j)}|Y_j$ is then sampled by drawing $x_1^{(j)}|Y_j$, followed by drawing $x_2^{(j)}|x_1^{(j)}, Y_j$, and so on.

References

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

[2] B. Horn and B. Schunk. Determining Optical Flow. *Artificial Intelligence*, vol. 17, 1981.

[3] B. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. *Proc. DARPA Image Understanding Workshop*, 1981

[4] J. Barron, D. Fleet, and S. Beauchemin. Performance of Optical Flow Techniques. *International Journal of Computer Vision*, vol. 12, 1994.

[5] J. Wang and E. Adelson. Representing Moving Images with Layers. *IEEE Trans. on Image Processing*, vol. 3, September 1994.

[6] B. Frey and N. Jojic. Estimating Mixture Models of Images and Inferring Spatial Transformations Using the EM Algorithm. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1999.

[7] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto. Dynamic textures. *International Journal of Computer Vision*, vol. 2, pp. 91-109, 2003.

[8] G. Doretto, D. Cremers, P. Favaro, S. Soatto. Dynamic texture segmentation. In *IEEE ICCV*, vol. 2, pp. 1236-42, 2003.

[9] P. Saisan, G. Doretto, Y. Wu, and S. Soatto. Dynamic texture recognition. In *IEEE CVPR*, vol. 2, pp. 58-63, 2001.

[10] A. B. Chan and N. Vasconcelos. Probabilistic kernels for the classification of auto-regressive visual processes. In *CVPR*, 2005.

[11] R. Vidal and A. Ravichandran Optical flow estimation & segmentation of multiple moving dynamic textures. In *CVPR*, 2005.

[12] 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.

[13] R.H. Shumway and D.S. Stoffer. An approach to time series smoothing and forecasting using the EM algorithm. *Journal of Time Series Analysis*, vol. 3(4), pp. 253-64, 1982.

[14] S. Roweis and Z. Ghahramani. A unifying review of linear Gaussian models. *Neural Computation*, vol. 11, pp. 305-45, 1999.

[15] `http://www.wsdot.wa.gov`

[16] `http://www.svcl.ucsd.edu`

[17] D.J.C. MacKay. Introduction to Monte Carlo Methods. In *Learning in Graphical Models*, pp. 175-204, Kluwer Academic Press, 1998.

[18] S. Geman and D. Geman. Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. *IEEE PAMI*, vol. 6(6), 1984.

[19] S. M. Kay *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice-Hall Signal Processing Series, 1993.