The EM Algorithm for Layered Dynamic Textures

Antoni Chan and Nuno Vasconcelos
Department of Electrical and Computer Engineering
University of California, San Diego

Technical Report SVCL-TR-2005-03

June 3rd, 2005

Abstract

A layered dynamic texture is an extension of the dynamic texture that contains multiple state processes, and hence can model video with multiple regions of different motion. In this technical report, we derive the EM algorithm for the two models of layered dynamic textures.

1 Introduction

A layered dynamic texture is an extension of the dynamic texture [1] that can model multiple regions (layers) of different motion. The motion in each layer is abstracted using a time-evolving state process, and each pixel formed from one of the state processes, which is assigned through a layer assignment variable. In Section 2 and Section 3, we derive the EM algorithm for learning the layered dynamic texture (LDT) and the approximate layered dynamic texture (ALDT), respectively.

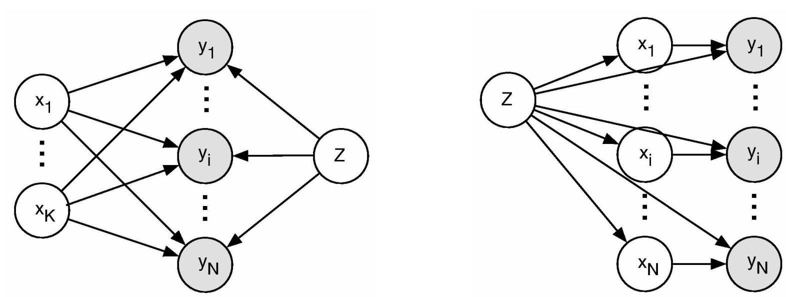

Figure 1: The layered dynamic texture (left), and the approximate layered dynamic texture (right). The observed pixel process y_i dependent on the underlying state processes x_j. Each pixel y_i is assigned to one of the state processes through the layer assignment variable z_i, which are the elements of Z.

2 Layered dynamic texture

The model for the layered dynamic texture is shown in Figure 1 (left). Each of the K layers has a Gauss-Markov state process x_j that abstracts the motion of the layer. The N observed pixels are assigned to one of the layers through the assignment variable z_i, which are the elements of Z. The linear system equations for this model are

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

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 layer j, the noise processes are $B^{(j)} w_{i,t} \sim_{iid} \mathcal{N}(0, Q^{(j)})$ and $\sqrt{r^{(j)}} v_t^{(j)} \sim_{iid} \mathcal{N}(0, r^{(j)} I_n)$, and the initial state is drawn from $x_1^{(j)} \sim \mathcal{N}(\mu^{(j)}, S^{(j)})$. In the remainder of this section, we discuss the modeling of Z, derive the joint log-likelihood function, and derive the EM algorithm for layered dynamic textures.

2.1 Markov Random Field

In order to insure spatial consistency of each layer, the elements of the layer assignment Z are modeled as a Markov Random Field (MRF) [2], given by the distribution

$$p(Z) = \frac{1}{\mathcal{Z}} \prod_i \psi_i(z_i) \prod_{(i,j) \in \mathcal{E}} \psi_{i,j}(z_i, z_j) \tag{2}$$

where $\mathcal{E}$ is the set of edges in the MRF grid, ψ_i and $\psi_{i,j}$ are the potential functions, and $\mathcal{Z}$ is the normalization constant (partition function) that makes the distribution sum to 1. The potential functions ψ_i and $\psi_{i,j}$ are

$$\psi_i(z_i) = \begin{cases} \alpha_1 & , z_i = 1 \\ \vdots & \vdots \\ \alpha_K & , z_i = K \end{cases} \qquad \psi_{i,j}(z_i, z_j) = \begin{cases} \gamma_1 & , z_i = z_j \\ \gamma_2 & , z_i \neq z_j \end{cases} \tag{3}$$

which correspond to some prior likelihood of each layer, and favoring neighboring pixels that are in the same layer, respectively. While the parameters for the potential functions could be learned for each model, we instead treat them as constants, which can be obtained by estimating their values over a corpus of manually segmented images.

2.2 Log-likelihood function

The joint log likelihood factors as

$$\begin{aligned} \ell(X, Y, Z) &= \log p(X, Y, Z) && (4) \\ &= \log p(Y|X, Z) + \log p(X) + \log p(Z) && (5) \\ &= \sum_{i,j} z_i^{(j)} \log p(y_i | x^{(j)}, z_i = j) + \sum_j \log p(x^{(j)}) + \log p(Z) && (6) \\ &= \sum_{i,j} z_i^{(j)} \sum_{t=1}^{\tau} \log p(y_{i,t} | x_t^{(j)}, z_i = j) && (7) \\ &\quad + \sum_j \left(\sum_{t=2}^{\tau} \log p(x_t^{(j)} | x_{t-1}^{(j)}) + \log p(x_1^{(j)}) \right) + \log p(Z) \end{aligned}$$

where $z_i^{(j)}$ is the indicator variable that $z_i = j$. Substituting for the probability distributions and dropping the constant terms, we have

$$\begin{aligned}
\ell(X,Y,Z) &= -\frac{1}{2}\sum_{i,j} z_i^{(j)} \sum_{t=1}^{\tau}\left(\left\|y_{i,t} - C_i^{(j)}x_t^{(j)}\right\|^2_{r^{(j)}} + \log\left|r^{(j)}\right|\right) && (8)\\
&\quad -\frac{1}{2}\sum_j\left(\sum_{t=2}^{\tau}\left(\left\|x_t^{(j)} - A^{(j)}x_{t-1}^{(j)}\right\|^2_{Q^{(j)}} + \log\left|Q^{(j)}\right|\right) + \left\|x_1^{(j)} - \mu^{(j)}\right\|^2_{S^{(j)}} + \log\left|S^{(j)}\right|\right)\\
&\quad + \log p(Z)\\
&= -\frac{1}{2}\sum_{i,j} z_i^{(j)} \sum_{t=1}^{\tau}\frac{1}{r^{(j)}}\left(y_{i,t}^2 - 2C_i^{(j)}x_t^{(j)}y_{i,t} + \mathrm{tr}\left(C_i^{(j)}x_t^{(j)}(x_t^{(j)})^T(C_i^{(j)})^T\right)\right) && (9)\\
&\quad - \frac{\tau}{2}\sum_{i,j} z_i^{(j)}\log r^{(j)}\\
&\quad - \frac{1}{2}\sum_j\sum_{t=2}^{\tau}\mathrm{tr}\left((Q^{(j)})^{-1}\left(x_t^{(j)}(x_t^{(j)})^T - x_t^{(j)}(x_{t-1}^{(j)})^T(A^{(j)})^T - A^{(j)}x_{t-1}^{(j)}(x_t^{(j)})^T\right.\right.\\
&\quad \left.\left. + A^{(j)}x_{t-1}^{(j)}(x_{t-1}^{(j)})^T(A^{(j)})^T\right)\right) - \frac{\tau-1}{2}\sum_j \log\left|Q^{(j)}\right|\\
&\quad - \frac{1}{2}\sum_j \mathrm{tr}\left((S^{(j)})^{-1}\left(x_1^{(j)}(x_1^{(j)})^T - x_1^{(j)}(\mu^{(j)})^T - \mu^{(j)}(x_1^{(j)})^T + \mu^{(j)}(\mu^{(j)})^T\right)\right)\\
&\quad - \frac{1}{2}\sum_j \log\left|S^{(j)}\right| + p(Z)
\end{aligned}$$

2.3 Learning with EM

The parameters of the model are learned using the EM algorithm [3], which is an iterative algorithm with the following steps

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

In the remainder of this section, we derive the E and M steps for the layered dynamic texture.

2.3.1 E-Step

Looking at the log-likelihood function (8), we see that the E-step requires the computation of the following conditional expectations:

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

Unfortunately these expectations are intractable to compute in closed-form, since it is not known which layer each pixel y_i is assigned to, and hence we must marginalize over the configurations of Z.

2.3.2 Inference using the Gibbs sampler

Using the Gibbs sampler [4], the expectations can be approximated by averaging over samples from the posterior distribution $p(X, Z|Y)$. 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 then as follows

1. Initialize $X \sim p(X)$
2. Repeat
   (a) Sample $Z \sim p(Z|X, Y)$
   (b) Sample $X \sim p(X|Y, Z)$

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

$$p(Z|X,Y) = \frac{p(Y,X|Z)p(Z)}{p(Y,X)} \quad (11)$$

$$= \frac{p(Y|X,Z)p(X|Z)p(Z)}{p(Y|X)p(X)} \quad (12)$$

$$\propto p(Y|X,Z)p(Z) \quad (13)$$

$$\propto p(Z)\prod_i p(y_i|X,z_i)) \quad (14)$$

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 [2].

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

$$p(X|Y,Z) = \prod_j p(x_j|Y,Z) = \prod_j p(x_j|Y_j) \quad (15)$$

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 probabilites,

$$p(x_1^{(j)}, \cdots, x_\tau^{(j)}|Y_j) = p(x_1^{(j)}|Y_j)\prod_{t=2}^{\tau} p(x_t^{(j)}|x_{t-1}^{(j)}, Y_j) \quad (16)$$

The parameters of each conditional Gaussian is obtained with the conditional Gaussian theorem [5],

$$\mathrm{E}(x_t^{(j)}|x_{t-1}^{(j)}, Y_j) = \mu_t^{(j)} + \Sigma_{t,t-1}^{(j)}(\Sigma_{t-1,t-1}^{(j)})^{-1}(x_{t-1}^{(j)} - \mu_{t-1}^{(j)}) \quad (17)$$

$$\mathrm{cov}(x_t^{(j)}|x_{t-1}^{(j)}, Y_j) = \Sigma_{t,t}^{(j)} - \Sigma_{t,t-1}^{(j)}(\Sigma_{t-1,t-1}^{(j)})^{-1}\Sigma_{t-1,t}^{(j)} \quad (18)$$

where the marginal mean, marginal covariance, and one-step covariance are

$$\mu_t^{(j)} = \mathrm{E}(x_t^{(j)}|Y_j) \quad (19)$$

$$\Sigma_{t,t}^{(j)} = \mathrm{cov}(x_t^{(j)}|Y_j) \quad (20)$$

$$\Sigma_{t,t-1}^{(j)} = \mathrm{cov}(x_t^{(j)}, x_{t-1}^{(j)}|Y_j) \quad (21)$$

which can be obtained using the Kalman smoothing filter [7, 8]. 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.

2.3.3 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,

$$\begin{array}{lll} \phi_1^{(j)} = \sum_{t=1}^{\tau-1} \hat{P}_{t,t}^{(j)} & \phi_2^{(j)} = \sum_{t=2}^{\tau} \hat{P}_{t,t}^{(j)} & \Phi^{(j)} = \sum_{t=1}^{\tau} \hat{P}_{t,t}^{(j)} \\ \Phi_i^{(j)} = \sum_{t=1}^{\tau} \hat{P}_{i|t,t}^{(j)} & \psi^{(j)} = \sum_{t=2}^{\tau} \hat{P}_{t,t-1}^{(j)} & \Gamma_i^{(j)} = \sum_{t=1}^{\tau} y_{i,t}\hat{x}_{i,t}^{(j)} \\ \hat{N}_j = \sum_i \hat{z}_i^{(j)} & \Lambda_i^{(j)} = \sum_{t=1}^{\tau} \hat{z}_i^{(j)} y_{i,t}^2 & \end{array} \tag{22}$$

The parameter updates are

- Transition matrix:

$$\frac{\partial \mathcal{Q}}{\partial A^{(j)}} = -\frac{1}{2}\sum_{t=2}^{\tau}\left(-2(Q^{(j)})^{-1}\hat{P}_{t,t-1}^{(j)} + 2(Q^{(j)})^{-1}A^{(j)}\hat{P}_{t-1}^{(j)}t-1\right) \tag{23}$$

$$\Rightarrow A^{(j)*} = \left(\sum_{t=2}^{\tau}\hat{P}_{t,t-1}^{(j)}\right)\left(\sum_{t=2}^{\tau}\hat{P}_{t-1,t-1}^{(j)}\right)^{-1} = \psi^{(j)}(\phi_1^{(j)})^{-1} \tag{24}$$

- State noise covariance:

$$\begin{aligned} \frac{\partial \mathcal{Q}}{\partial Q^{(j)}} &= -\frac{1}{2}\sum_{t=2}^{\tau}\left(-(Q^{(j)})^{-T}\left(\hat{P}_{t,t}^{(j)} - \hat{P}_{t,t-1}^{(j)}(A^{(j)})^T - \right.\right. \\ &\left.\left. A^{(j)}\hat{P}_{t-1,t}^{(j)} + A^{(j)}\hat{P}_{t-1,t-1}^{(j)}(A^{(j)})^T\right)(Q^{(j)})^{-T}\right) - \frac{\tau-1}{2}(Q^{(j)})^{-T} \end{aligned} \tag{25}$$

$$\Rightarrow Q^{(j)*} = \frac{1}{\tau-1}\left(\left(\sum_{t=2}^{\tau}\hat{P}_{t,t}^{(j)}\right) - A^{(j)*}\left(\sum_{t=2}^{\tau}\hat{P}_{t,t-1}^{(j)}\right)^T\right) = \frac{1}{\tau-1}(\phi_2^{(j)} - A^{(j)*}(\psi^{(j)})^T) \tag{26}$$

- Initial state:

$$\frac{\partial \mathcal{Q}}{\partial \mu^{(j)}} = -\frac{1}{2}\left(-2\hat{x}_1^{(j)} + 2\mu^{(j)}\right) \tag{27}$$

$$\Rightarrow \mu^{(j)*} = \hat{x}_1^{(j)} \tag{28}$$

- Initial covariance:

$$\begin{aligned} \frac{\partial \mathcal{Q}}{\partial S^{(j)}} &= -\frac{1}{2}\left(-(S^{(j)})^{-T}\left(\hat{P}_{1,1}^{(j)} - \hat{x}_1^{(j)}(\mu^{(j)})^T - \right.\right. \\ &\left.\left. \mu^{(j)}(\hat{x}_1^{(j)})^T + \mu^{(j)}(\mu^{(j)})^T\right)(S^{(j)})^{-T} + (S^{(j)})^{-T}\right) \end{aligned} \tag{29}$$

$$\Rightarrow S^{(j)*} = \hat{P}_{1,1}^{(j)} - \mu^{(j)*}(\mu^{(j)*})^T \tag{30}$$

- Transformation matrix:

$$\frac{\partial \mathcal{Q}}{\partial C_i^{(j)}} = -\frac{1}{2}\sum_t \left(-2(r^{(j)})^{-1} y_{i,t} (\hat{x}_{i,t}^{(j)})^T + 2(r^{(j)})^{-1} C_i^{(j)} \hat{P}_{i|t,t}^{(j)} \right) \tag{31}$$

$$\Rightarrow C_i^{(j)*} = \left(\sum_t y_{i,t} \hat{x}_{i,t}^{(j)} \right)^T \left(\sum_t \hat{P}_{i|t,t}^{(j)} \right)^{-1} = (\Gamma_i^{(j)})^T (\Phi_i^{(j)})^{-1} \tag{32}$$

- Pixel noise variance:

$$\frac{\partial \mathcal{Q}}{\partial r^{(j)}} = -\frac{1}{2}\sum_{i,t} \left(-(r^{(j)})^{-2} \left(\hat{z}_i^{(j)} y_{i,t}^2 - 2C_i^{(j)} \hat{x}_{i,t}^{(j)} y_{i,t} + C_i^{(j)} \hat{P}_{i|t,t}^{(j)} (C_i^{(j)})^T \right) \right) - \frac{\tau \hat{N}_j}{2} (r^{(j)})^{-1} \tag{33}$$

$$\begin{aligned} \Rightarrow r^{(j)*} &= \frac{1}{\tau \hat{N}_j} \left(\sum_{i,t} \hat{z}_i^{(j)} y_{i,t}^2 - 2\sum_{i,t} C_i^{(j)} \hat{x}_{i,t}^{(j)} y_{i,t} + \sum_{i,t} C_i^{(j)} \hat{P}_{i|t,t}^{(j)} (C_i^{(j)})^T \right) & (34) \\ &= \frac{1}{\tau \hat{N}_j} \sum_i (\Lambda_i^{(j)} - C_i^{(j)*} \Gamma_i^{(j)}) & (35) \end{aligned}$$

In summary, the M-step is

$$\begin{array}{ll} A^{(j)*} = \psi^{(j)} (\phi_1^{(j)})^{-1} & Q^{(j)*} = \frac{1}{\tau-1} (\phi_2^{(j)} - A^{(j)*} (\psi^{(j)})^T) \\ \mu^{(j)*} = \hat{x}_1^{(j)} & S^{(j)*} = \hat{P}_{1,1}^{(j)} - \mu^{(j)*} (\mu^{(j)*})^T \\ C_i^{(j)*} = (\Gamma_i^{(j)})^T (\Phi_i^{(j)})^{-1} & r^{(j)*} = \frac{1}{\tau \hat{N}_j} \sum_{i=1}^N (\Lambda_i^{(j)} - C_i^{(j)*} \Gamma_i^{(j)}) \end{array} \tag{36}$$

3 Approximate layered dynamic texture

We now consider a slightly different model shown in Figure 1 (right). Each pixel y_i is associated with its own state process x_i, whose parameters are chosen using the layer assignment z_i. The new model is given by the following linear system equations

$$\left\{ \begin{array}{l} x_{i,t} = A^{(z_i)} x_{i,t-1} + B^{(z_i)} v_{i,t} \quad , i \in \{1, \cdots, N\} \\ y_{i,t} = C_i^{(j)} x_{i,t} + \sqrt{r^{(z_i)}} w_{i,t} \end{array} \right. \tag{37}$$

where the noise processes are $w_{i,t} \sim_{iid} \mathcal{N}(0,1)$ and $v_{i,t} \sim_{iid} \mathcal{N}(0, I_n)$. The log-likelihood factors as

$$\log p(X,Y,Z) = \log p(Y|X,Z) + \log p(X|Z) + \log p(Z) \tag{38}$$

If the pixels are assigned independently and identically distributed as a multinomial distribution, then the model is similar to a mixture of dynamic textures [6]. The only difference is that $C_i^{(j)}$ is different for each pixel and each layer, rather than for each state. Again, we will model Z as an MRF grid as in the previous section.

3.1 Joint log-likelihood

The joint log-likelihood of the model is given by

$$\begin{aligned}
\ell(X,Y,Z) &= \log p(Y|X,Z) + \log p(X|Z) + \log p(Z) && (39)\\
&= \log \prod_{i,j} p(y_i|x_i, z_i = j)^{z_i^{(j)}} + \log \prod_{i,j} p(x_i|z_i = j)^{z_i^{(j)}} + \log p(Z) && (40)\\
&= \sum_{i,j} \sum_{t=1}^{\tau} z_i^{(j)} \log p(y_{i,t}|x_{i,t}, z_i = j) && (41)\\
&\quad + \sum_{i,j} z_i^{(j)} \left(\log p(x_{i,1}|z_i = j) + \sum_{t=2}^{\tau} \log p(x_{i,t}|x_{i,t-1}) \right) + \log p(Z)
\end{aligned}$$

Substituting for each of the distributions and ignoring the constant terms, we have

$$\begin{aligned}
\ell(X,Y,Z) &= -\frac{1}{2} \sum_{i,j} \sum_{t=1}^{\tau} z_i^{(j)} \left(\log r^{(j)} + \left\| y_{i,t} - C_i^{(j)} x_{i,t} \right\|^2_{r^{(j)}} \right) && (42)\\
&\quad - \frac{1}{2} \sum_{i,j} \sum_{t=2}^{\tau} z_i^{(j)} \left(\log \left| Q^{(j)} \right| + \left\| x_{i,t} - A^{(j)} x_{i,t-1} \right\|^2_{Q^{(j)}} \right)\\
&\quad - \frac{1}{2} \sum_{i,j} z_i^{(j)} \left(\log \left| S^{(j)} \right| + \left\| x_{i,1} - \mu^{(j)} \right\|^2_{S^{(j)}} \right) + \log p(Z)\\
&= -\frac{1}{2} \sum_{i,j} z_i^{(j)} \sum_{t=1}^{\tau} \frac{1}{r^{(j)}} \left(y_{i,t}^2 - 2 C_i^{(j)} x_t^{(j)} y_{i,t} + \operatorname{tr} \left(C_i^{(j)} x_t^{(j)} (x_t^{(j)})^T (C_i^{(j)})^T \right) \right) && (43)\\
&\quad - \frac{\tau}{2} \sum_{i,j} z_i^{(j)} \log r^{(j)}\\
&\quad - \frac{1}{2} \sum_{i,j} z_i^{(j)} \sum_{t=2}^{\tau} \operatorname{tr} \left((Q^{(j)})^{-1} \left(x_t^{(j)} (x_t^{(j)})^T - x_t^{(j)} (x_{t-1}^{(j)})^T (A^{(j)})^T - A^{(j)} x_{t-1}^{(j)} (x_t^{(j)})^T \right.\right.\\
&\quad \left.\left. + A^{(j)} x_{t-1}^{(j)} (x_{t-1}^{(j)})^T (A^{(j)})^T \right) \right) - \frac{\tau - 1}{2} \sum_{i,j} z_i^{(j)} \log \left| Q^{(j)} \right|\\
&\quad - \frac{1}{2} \sum_{i,j} z_i^{(j)} \operatorname{tr} \left((S^{(j)})^{-1} \left(x_1^{(j)} (x_1^{(j)})^T - x_1^{(j)} (\mu^{(j)})^T - \mu^{(j)} (x_1^{(j)})^T + \mu^{(j)} (\mu^{(j)})^T \right) \right)\\
&\quad - \frac{1}{2} \sum_{i,j} z_i^{(j)} \log \left| S^{(j)} \right| + p(Z)
\end{aligned}$$

3.2 Learning with EM

The parameters of the simplified model are learned using the EM algorithm. In the remainder of this section we derive the E and M steps for the ALDT model.

3.2.1 E-Step

The E-step for the simple model is very similar to that of the mixture of dynamic textures [6]. The expectations that need to be computed are

$$\mathrm{E}_{X,Z|Y}(z_i^{(j)} x_{i,t}) = \hat{z}_i^{(j)} \hat{x}_{i,t}^{(j)} \qquad (44)$$

$$\mathrm{E}_{X,Z|Y}(z_i^{(j)} x_{i,t}(x_{i,t})^T) = \hat{z}_i^{(j)} \hat{P}_{i|t,t}^{(j)} \quad (45)$$

$$\mathrm{E}_{X,Z|Y}(z_i^{(j)} x_{i,t}(x_{i,t-1})^T) = \hat{z}_i^{(j)} \hat{P}_{i|t,t-1}^{(j)} \quad (46)$$

where

$$\hat{x}_{i,t}^{(j)} = \mathrm{E}_{x_i^{(j)}|y_i,z_i^{(j)}=1}\left(x_{i,t}^{(j)}\right) \quad (47)$$

$$\hat{P}_{i|t,t}^{(j)} = \mathrm{E}_{x_i^{(j)}|y_i,z_i^{(j)}=1}\left(x_{i,t}^{(j)} x_{i,t}^{(j)T}\right) \quad (48)$$

$$\hat{P}_{i|t,t-1}^{(j)} = \mathrm{E}_{x_i^{(j)}|y_i,z_i^{(j)}=1}\left(x_{i,t}^{(j)} x_{i,t-1}^{(j)}{}^T\right) \quad (49)$$

$$\hat{z}_i^{(j)} = p(z_i = j|Y) \quad (50)$$

The first three terms can be computed using the Kalman smoothing filter [7, 8]. The posterior of z_i can be estimated by averaging over samples drawn from the full posterior $p(Z|Y)$ MRF, using Markov-Chain Monte Carlo (MCMC) [2],

$$p(Z|Y) \propto p(Y|Z)p(Z) \quad (51)$$

$$= \prod_i p(y_i|z_i)p(Z) \quad (52)$$

$$= \prod_i p(y_i|z_i)\psi_i(z_i) \prod_{(i,j)\in\mathcal{E}} \psi_{i,j}(z_i, z_j) \quad (53)$$

where $p(y_i|z_i)$ is computed using the innovations of the Kalman filter [7].

3.2.2 M-Step

Most of the M-step is similar to that of the mixture of dynamic textures, except that $C_i^{(j)}$ is different for each pixel and layer. For convenience, we define the quantities,

$$\begin{array}{ll} \Phi^{(j)} = \sum_{i=1}^N \hat{z}_i^{(j)} \sum_{t=1}^\tau \hat{P}_{i|t,t}^{(j)} & \phi_1^{(j)} = \sum_{i=1}^N \hat{z}_i^{(j)} \sum_{t=1}^{\tau-1} \hat{P}_{i|t,t}^{(j)} \\ \phi_2^{(j)} = \sum_{i=1}^N \hat{z}_i^{(j)} \sum_{t=2}^\tau \hat{P}_{i|t,t}^{(j)} & \psi^{(j)} = \sum_{i=1}^N \hat{z}_i^{(j)} \sum_{t=2}^\tau \hat{P}_{i|t,t-1}^{(j)} \\ \Gamma_i^{(j)} = \hat{z}_i^{(j)} \sum_{t=1}^\tau y_{i,t}(\hat{x}_{i,t}^{(j)})^T & \Phi_i^{(j)} = \hat{z}_i^{(j)} \sum_{t=1}^\tau \hat{P}_{i|t,t}^{(j)} \end{array} \quad (54)$$

and $\hat{N}_j = \sum_i \hat{z}_i^{(j)}$ and $\Lambda_i^{(j)} = \sum_{t=1}^\tau \hat{z}_i^{(j)} y_{i,t}^2$. The updates for the parameters are

- State Transition Matrix

$$\frac{\partial \mathcal{Q}}{\partial A^{(j)}} = -\frac{1}{2}\sum_i \sum_{t=2}^\tau \hat{z}_i^{(j)}\left[-2(Q^{(j)})^{-1}\hat{P}_{i|t,t-1}^{(j)} + 2(Q^{(j)})^{-1}A^{(j)}\hat{P}_{i|t-1,t-1}^{(j)}\right] = 0 \quad (55)$$

$$\sum_i \sum_{t=2}^\tau \hat{z}_i^{(j)}\hat{P}_{i|t,t-1}^{(j)} - A^{(j)}\sum_i \sum_{t=2}^\tau \hat{z}_i^{(j)}\hat{P}_{i|t-1,t-1}^{(j)} = 0 \quad (56)$$

$$\Rightarrow A^{(j)*} = \psi^{(j)}(\phi_1^{(j)})^{-1} \quad (57)$$

- State Noise Covariance

$$\frac{\partial \mathcal{Q}}{\partial Q^{(j)}} = \frac{1}{2}\sum_i \sum_{t=2}^{\tau} \hat{z}_i^{(j)} (Q^{(j)})^{-1} \Big[\hat{P}_{i|t,t}^{(j)} - \hat{P}_{i|t,t-1}^{(j)} (A^{(j)})^T - A^{(j)} (\hat{P}_{i|t,t-1}^{(j)})^T + A^{(j)} \hat{P}_{i|t-1,t-1}^{(j)} (A^{(j)})^T \Big] (Q^{(j)})^{-1} - \frac{\tau-1}{2} \sum_i \hat{z}_i^{(j)} (Q^{(j)})^{-1} = 0 \tag{58}$$

$$= \sum_i \sum_{t=2}^{\tau} \hat{z}_i^{(j)} \hat{P}_{i|t,t}^{(j)} - \sum_i \sum_{t=2}^{\tau} \hat{z}_i^{(j)} \hat{P}_{i|t,t-1}^{(j)} (A^{(j)})^T - \sum_i \sum_{t=2}^{\tau} \hat{z}_i^{(j)} A^{(j)} (\hat{P}_{i|t,t-1}^{(j)})^T + \sum_i \sum_{t=2}^{\tau} \hat{z}_i^{(j)} A^{(j)} \hat{P}_{i|t-1,t-1}^{(j)} (A^{(j)})^T - (\tau-1)\hat{N}_j Q^{(j)} = 0 \tag{59}$$

$$\Rightarrow Q^{(j)} = \frac{1}{(\tau-1)\hat{N}_j} \Big[\phi_2^{(j)} - \psi^{(j)} (A^{(j)})^T - A^{(j)} (\psi^{(j)})^T + A^{(j)} \phi_1^{(j)} (A^{(j)})^T \Big] \tag{60}$$

$$Q^{(j)*} = \frac{1}{(\tau-1)\hat{N}_j} \Big[\phi_2^{(j)} - A^{(j)*} (\psi^{(j)})^T \Big] \tag{61}$$

- Initial State Mean

$$\frac{\partial \mathcal{Q}}{\partial \mu^{(j)}} = \frac{1}{2}\sum_i \hat{z}_i^{(j)} \Big[2(S^{(j)})^{-1} \hat{x}_{i,1}^{(j)} - 2(S^{(j)})^{-1} \mu^{(j)} \Big] = 0 \tag{62}$$

$$= \sum_i \hat{z}_i^{(j)} \hat{x}_{i,1}^{(j)} - \sum_i \hat{z}_i^{(j)} \mu^{(j)} = 0 \tag{63}$$

$$\Rightarrow \mu^{(j)*} = \frac{1}{\hat{N}_j} \sum_i \hat{z}_i^{(j)} \hat{x}_{i,1}^{(j)} \tag{64}$$

- Initial State Covariance

$$\frac{\partial \mathcal{Q}}{\partial S^{(j)}} = \frac{1}{2}\sum_i \hat{z}_i^{(j)} (S^{(j)})^{-1} \Big[\hat{P}_{i|1,1}^{(j)} - \hat{x}_{i,1}^{(j)} (\mu^{(j)})^T - \mu^{(j)} (\hat{x}_{i,1}^{(j)})^T + \mu^{(j)} (\mu^{(j)})^T \Big] (S^{(j)})^{-1} - \frac{1}{2}\sum_i \hat{z}_i^{(j)} (S^{(j)})^{-1} = 0 \tag{65}$$

$$= \sum_i \hat{z}_i^{(j)} \Big[\hat{P}_{i|1,1}^{(j)} - \hat{x}_{i,1}^{(j)} (\mu^{(j)})^T - \mu^{(j)} (\hat{x}_{i,1}^{(j)})^T + \mu^{(j)} (\mu^{(j)})^T \Big] - \sum_i \hat{z}_i^{(j)} S^{(j)} = 0 \tag{66}$$

$$\Rightarrow S^{(j)} = \frac{1}{\hat{N}_j} \sum_i \hat{z}_i^{(j)} \Big[\hat{P}_{i|1,1}^{(j)} - \hat{x}_{i,1}^{(j)} (\mu^{(j)})^T - \mu^{(j)} (\hat{x}_{i,1}^{(j)})^T + \mu^{(j)} (\mu^{(j)})^T \Big] \tag{67}$$

$$S^{(j)*} = \left(\frac{1}{\hat{N}_j} \sum_i \hat{z}_i^{(j)} \hat{P}_{i|1,1}^{(j)} \right) - \mu^{(j)*} (\mu^{(j)*})^T \tag{68}$$

- Transformation matrix:

$$\frac{\partial \mathcal{Q}}{\partial C_i^{(j)}} = -\frac{1}{2} \sum_t \frac{\hat{z}_i^{(j)}}{r^{(j)}} \Big[-2\hat{x}_{i,t}^{(j)} y_{i,t} + 2\hat{P}_{i|t,t}^{(j)} (C_i^{(j)})^T \Big] = 0 \tag{69}$$

$$\Rightarrow C_i^{(j)*} = \left(\sum_t y_{i,t} \hat{x}_{i,t}^{(j)} \right)^T \left(\sum_t \hat{P}_{i|t,t}^{(j)} \right)^{-1} = (\Gamma_i^{(j)})^T (\Phi_i^{(j)})^{-1} \tag{70}$$

- Pixel noise variance:

$$\frac{\partial Q}{\partial r^{(j)}} = -\frac{\tau}{2}\sum_i \hat{z}_i^{(j)} \frac{1}{r^{(j)}} + \frac{1}{2}\sum_{i,t} \frac{\hat{z}_i^{(j)}}{(r^{(j)})^2}(y_{i,t}^2 - 2C_i^{(j)}\hat{x}_{i,t}^{(j)} y_{i,t} + C_i^{(j)}\hat{P}_{i|t,t}^{(j)}(C_i^{(j)})^T) = 0 \quad (71)$$

$$\Rightarrow r^{(j)} = \frac{1}{\tau \hat{N}_j}\sum_{i,t} \hat{z}_i^{(j)}(y_{i,t}^2 - 2C_i^{(j)}\hat{x}_{i,t}^{(j)} y_{i,t} + C_i^{(j)}\hat{P}_{i|t,t}^{(j)}(C_i^{(j)})^T) \quad (72)$$

$$\Rightarrow r^{(j)*} = \frac{1}{\tau \hat{N}_j}\sum_i \left(\hat{z}_i^{(j)}(\sum_t y_{i,t}^2) - C_i^{(j)*}\hat{z}_i^{(j)}(\sum_t \hat{x}_{i,t}^{(j)} y_{i,t})\right) = \frac{1}{\tau \hat{N}_j}\sum_i (\Lambda_i^{(j)} - C_i^{(j)*}\Gamma_i^{(j)}) \quad (73)$$

In summary, the M-step is

$$\begin{array}{ll} A^{(j)*} = \psi^{(j)}(\phi_1^{(j)})^{-1} & Q^{(j)*} = \frac{1}{(\tau-1)\hat{N}_j}(\phi_2^{(j)} - A^{(j)*}(\psi^{(j)})^T) \\ \mu^{(j)*} = \frac{1}{\hat{N}_j}\sum_i \hat{z}_i^{(j)}\hat{x}_{i,1}^{(j)} & S^{(j)*} = \frac{1}{\hat{N}_j}(\sum_{i=1}^N \hat{z}_i^{(j)}\hat{P}_{i|1,1}^{(j)}) - \mu^{(j)*}(\mu^{(j)*})^T \\ C_i^{(j)*} = (\Gamma_i^{(j)})^T(\Phi_i^{(j)})^{-1} & r^{(j)*} = \frac{1}{\tau\hat{N}_j}\sum_{i=1}^N(\Lambda_i^{(j)} - C_i^{(j)*}\Gamma_i^{(j)}) \end{array} \quad (74)$$

References

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

[2] S. Geman and D. Geman. Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6(6), pp. 721-41, 1984.

[3] 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.

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

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

[6] A. B. Chan and N. Vasconcelos Mixtures of dynamic textures. In *IEEE International Conference on Computer Vision*, vol. 1, pp. 641-47, 2005.

[7] 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.

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