Mixtures of 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 linear dynamical system used to model a single video as a sample from a spatio-temporal stochastic process. In this work, we introduce the mixture of dynamic textures, which models a collection of videos consisting of different visual processes as samples from a set of dynamic textures. We derive the EM algorithm for learning a mixture of dynamic textures, and relate the learning algorithm and the dynamic texture mixture model to previous works. Finally, we demonstrate the applicability of the proposed model to problems that have traditionally been challenging for computer vision.

1. Introduction

One family of visual processes that has relevance for various applications of computer vision is that of, what could be loosely described as, visual processes composed of *ensembles of particles subject to stochastic motion*. The particles can be microscopic, e.g plumes of smoke, macroscopic, e.g. leaves and vegetation blowing in the wind, or even objects, e.g. a human crowd, a flock of birds, a traffic jam, or a bee-hive. The applications range from remote monitoring for the prevention of natural disasters, e.g. forest fires, to background subtraction in challenging environments, e.g. outdoors scenes with vegetation, and various types of surveillance, e.g. traffic monitoring, homeland security applications, or scientific studies of animal behavior.

Despite their practical significance, and the ease with which they are perceived by biological vision systems, the visual processes in this family still pose tremendous challenges for computer vision. In particular, the *stochastic nature* of the associated motion fields tends to be *highly challenging for traditional motion representations* such as optical flow [1–4], which requires some degree of motion smoothness, parametric motion models [5–7], which assume a piece-wise planar world, or object tracking [8–10], which tends to be impractical when the number of subjects to track is large and these objects interact in a complex manner.

The main limitation of all these representations is that they are inherently *local*, aiming to achieve understanding of the whole by modeling the motion of the individual particles. This is *contrary to how these visual processes are perceived by biological vision*: smoke is usually perceived as a whole, a tree is normally perceived as a single object, and the detection of traffic jams rarely requires tracking individual vehicles. Recently, there has been an effort to advance towards this type of *holistic modeling*, by viewing video sequences derived from these processes as *dynamic textures* or, more precisely, samples from stochastic processes defined over space and time [11–14]. In fact, the dynamic texture framework has been shown to have great potential for video synthesis [11], motion segmentation [12], and video classification [13, 14]. This is, in significant part, due to the fact that the underlying generative probabilistic framework is capable of 1) abstracting a wide variety of complex motion patterns into a *simple* spatio-temporal process, and 2) synthesizing samples of the associated time-varying texture.

One major current limitation of the dynamic texture framework 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 in opposite directions, video containing both smoke and fire, and so forth. While, in such cases, existing dynamic texture models are inherently incorrect, the underlying generative framework is not. In fact, co-occurring textures can be easily accounted for by augmenting the probabilistic generative model with a discrete *hidden* variable, that has a number of states equal to the number of textures, and encodes which of them is responsible for a given piece of the spatio-temporal video volume. Conditioned on the state of this hidden variable, the video is then modeled as a simple dynamic texture.

This leads to an extension of the dynamic texture model, a *mixture of dynamic textures*, that we study in this work. In addition to introducing the model itself, we report on three main contributions. First, the expectation maximiza-

tion (EM) algorithm is derived for maximum likelihood estimation of the parameters of a mixture of dynamic textures. Second, the relationships between this mixture model and various other models previously proposed in the machine learning and computer vision literatures, including mixtures of factor analyzers, linear dynamical systems, and switched linear dynamic models, are analyzed. Finally, we demonstrate the applicability of the new model to the solution of traditionally difficult vision problems that range from clustering traffic video sequences to segmentation of sequences containing multiple dynamic textures.

The paper is organized as follows. In Section 2, we formalize the dynamic texture mixture model. In Section 3 we present the EM algorithm for learning its parameters from training data. In Section 4, we relate it to previous models and discuss its application to video clustering and segmentation. Finally, in Section 5 we present an experimental evaluation in the context of these applications.

2. Mixtures of dynamic textures

In this section, we introduce the dynamic texture mixture model. For completeness, we start with a brief review of the dynamic texture model.

2.1. Dynamic texture

A dynamic texture [11] is a generative video model defined by a random process with an observed variable y_t, which encodes the video frame at a specific time, and a hidden state variable x_t, which encodes the evolution of the video over time. The state and observed variables are related through the following linear dynamical system (LDS) equations

$$\begin{cases} x_{t+1} = Ax_t + v_t \\ y_t = Cx_t + w_t \end{cases} \quad (1)$$

where $x_t \in \mathbb{R}^n$ and $y_t \in \mathbb{R}^m$ (typically $n \ll m$). The parameter $A \in \mathbb{R}^{n\times n}$ is the state transition matrix and $C \in \mathbb{R}^{m\times n}$ a matrix containing the principal components of the video sequence. The driving noise process is $v_t \sim \mathcal{N}(0, Q,)$ with $Q \in \mathbb{R}^{n\times n}$, and the observed noise is $w_t \sim \mathcal{N}(0, R,)$ with $R \in \mathbb{R}^{m\times m}$, where $\mathcal{N}(\mu, \Sigma,)$ is a Gaussian distribution with mean μ and covariance Σ. We extend the definition of [11] by allowing the initial state x_1 to be distributed as $x_1 \sim \mathcal{N}(\mu, S)$. The dynamic texture is completely specified with the parameters $\Theta = \{A, Q, C, R, \mu, S\}$, and is represented as a graphical model in Figure 1a.

It can be shown [15] that the probability of the initial state, the conditional state distribution, and the conditional observation distribution are given by

$$p(x_1) = G(x_1, \mu, S) \quad (2)$$

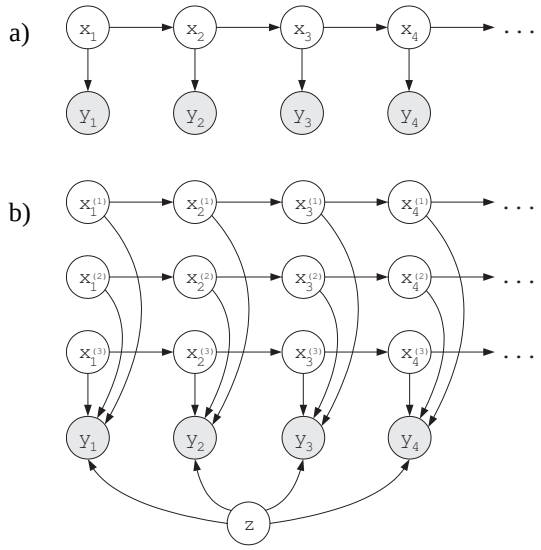

Figure 1. a) The dynamic texture. b) The dynamic texture mixture with 3 components. The variable $x_t^{(j)}$ is the state of the j^{th} dynamic texture at time t, and the hidden variable z selects from the three dynamic textures.

$$p(x_t|x_{t-1}) = G(x_t, Ax_{t-1}, Q) \quad (3)$$
$$p(y_t|x_t) = G(y_t, Cx_t, R) \quad (4)$$

where $G(x, \mu, \Sigma) = (2\pi)^{-n/2}|\Sigma|^{-1/2}e^{-\frac{1}{2}\|x-\mu\|^2_\Sigma}$ is the n-dimensional Gaussian distribution and $\|x\|^2_\Sigma = x^T\Sigma^{-1}x$. Letting $x_1^\tau = (x_1, \cdots, x_\tau)$ and $y_1^\tau = (y_1, \cdots, y_\tau)$ be a sequence of states and observations, the joint distribution is

$$p(x_1^\tau, y_1^\tau) = p(x_1)\prod_{t=2}^{\tau} p(x_t|x_{t-1})\prod_{t=1}^{\tau} p(y_t|x_t). \quad (5)$$

A number of methods are available to learn the parameters of the dynamic texture from a training video sequence, including asymptotically maximum likelihood methods such as N4SID [16] or expectation-maximization [17], and a suboptimal (but computationally efficient) solution [11].

2.2. Mixture of dynamic textures

Consider a generative video model, where the observation of a video sequence y_1^τ is generated from one of K dynamic textures, each with probability α_j of occurring. Given component probabilities $\{\alpha_1, \ldots, \alpha_K\}$ with $\sum_{j=1}^K \alpha_j = 1$ and dynamic texture components $\{\Theta_1, \ldots, \Theta_K\}$, the generative model is:

1. Sample component j from the distribution $\{\alpha_1, \ldots, \alpha_K\}$.
2. Sample an observation y_1^τ from the dynamic texture component Θ_j.

N	number of observed sequences
τ	length of an observed sequence
K	number of mixture components
i	index over the set of observed sequences
j	index over the components of the mixture
t	time index of a sequence
y_i	the i^{th} observed sequence
$y_{i,t}$	the observation at time t of y_i
$x_i^{(j)}$	the state sequence of y_i under component j
$x_{i,t}^{(j)}$	the state at time t of $x_i^{(j)}$
$z_i^{(j)}$	the indicator variable that y_i is from component j
α_j	the probability of the j^{th} component
Θ_j	the parameters of the j^{th} component

Table 1. Notation for EM for mixture of dynamic textures

The probability of the sequence y_1^τ sampled from this generative model is

$$p(y_1^\tau) = \sum_{j=1}^{K} \alpha_j p_j(y_1^\tau) \tag{6}$$

where $p_j(y_1^\tau) = p(y_1^\tau; \Theta_j)$ is the class conditional probability of the j^{th} dynamic texture. This is the mixture model for dynamic textures.

An equivalent model, shown for $K = 3$ components in Figure 1b, is to explicitly represent the state of *each* component j by the conditional distribution

$$p(x_t^{(j)}|x_{t-1}^{(j)}) = G(x_t^{(j)}, A^{(j)}x_{t-1}^{(j)}, Q^{(j)}) \tag{7}$$

The observed variable y_t is conditioned on *all* the component states and the hidden variable z,

$$p(y_t|x_t^{(1)}, \cdots, x_t^{(K)}, z = j) = G(y_t, C^{(j)}x_t^{(j)}, R^{(j)}) \tag{8}$$

where z selects the appropriate state $x_t^{(j)}$ and the $C^{(j)}$ and $R^{(j)}$ parameters to form the j^{th} dynamic texture.

3. Parameter estimation using EM

The EM algorithm [18] is a method for estimating the parameters of a probability distribution when the distribution depends on hidden variables (i.e. there is missing data). For the dynamic texture mixture, the observed information is a set of video sequences $\{y_i\}$, and the missing data consists of 1) the assignments of sequences to mixture components (the assignment of sequence y_i to the j^{th} mixture component is encoded by the state of the indicator variable $z_i^{(j)}$), and 2) the hidden state sequence $x_i^{(j)}$ for y_i under component j (see Table 1 for notation). The EM solution is found using an iterative procedure that alternates between estimating the missing information with the current parameters, and computing new parameters given the estimate of the missing information. The EM iteration is

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

where $p(x, y, z; \Theta)$ is the complete data likelihood, parameterized by Θ, of the observation, hidden state, and hidden assignment variables.

We assume that the training data is a set of independent video sequences. The EM algorithm for a mixture of dynamic textures is presented in Algorithm 1 (see [19] for derivation). The expectation step computes the conditional expectations of the state variables

$$\hat{x}_{i,t}^{(j)} = \mathrm{E}(x_{i,t}^{(j)}|y_i, z_i^{(j)} = 1) \tag{9}$$

$$\hat{V}_{i|t,t}^{(j)} = \mathrm{cov}(x_{i,t}^{(j)}, x_{i,t}^{(j)}|y_i, z_i^{(j)} = 1) \tag{10}$$

$$\hat{V}_{i|t,t-1}^{(j)} = \mathrm{cov}(x_{i,t}^{(j)}, x_{i,t-1}^{(j)}|y_i, z_i^{(j)} = 1) \tag{11}$$

where the conditional expectations are taken with respect to the distribution of the hidden state x_t, given the observed sequence y_i, and parameterized by the j^{th} mixture component. In addition, the E-step computes the conditional likelihood of the observation given the j^{th} mixture component, $p_j(y_i)$, for all j. These quantities are computed using the Kalman smoothing filter [15, 17, 19]. The maximization step computes the maximum likelihood parameter values for each dynamic texture component, by averaging over all sequences $\{y_i\}$, weighted by the probability that the sequence y_i belongs to the j^{th} mixture component.

For the purposes of our experiments, each dynamic texture component Θ_j was initialized by using the suboptimal learning method of [11] on a random video sequence from the training set. The component probabilities were initialized to a uniform distribution, $\alpha_j = 1/K$. Since the EM algorithm can terminate on a local minimum, the algorithm was run several times using different initialization seeds, and the parameters which best fit the training data (in the maximum likelihood sense) were kept. Finally, the covariance matrices Q, S, and R were regularized by forcing their eigenvalues to be larger than a minimum value, and by restricting S and R to be diagonal.

4. Connections to the literature and applications

The proposed EM learning algorithm and dynamic texture mixture model are related to several previous works. In this section we briefly describe these relations and discuss two applications of the dynamic texture mixture to the problems of video clustering and motion segmentation.

Algorithm 1 EM for Dynamic Texture Mixture

Input: N sequences $\{y_i\}_{i=1}^N$, num. of components K.
Initialize $\{\Theta_j\}_{j=1}^K = \{\alpha_j, A_j, Q_j, R_j, C_j, \mu_j, S_j\}_{j=1}^K$.
repeat
{Expectation Step}
for $i = 1$ to N and $j = 1$ to K **do**
Compute $\hat{x}_{i,t}^{(j)}$, $\hat{V}_{i|t,t}^{(j)}$, $\hat{V}_{i|t,t-1}^{(j)}$, and $\log p_j(y_i)$ using the Kalman smoothing filter with y_i and Θ_j.
$\hat{P}_{i|t,t}^{(j)} = \hat{V}_{i|t,t}^{(j)} + \hat{x}_{i,t}^{(j)}(\hat{x}_{i,t}^{(j)})^T$
$\hat{P}_{i|t,t-1}^{(j)} = \hat{V}_{i|t,t-1}^{(j)} + \hat{x}_{i,t}^{(j)}(\hat{x}_{i,t-1}^{(j)})^T$
$\hat{z}_i^{(j)} = \frac{\alpha_j p_j(y_i)}{\sum_{k=1}^K \alpha_k p_k(y_i)}$
end for
{Maximization Step}
for $j = 1$ to K **do**
$\Phi_j = \sum_{i=1}^N \hat{z}_i^{(j)} \sum_{t=1}^\tau \hat{P}_{i|t,t}^{(j)}$
$\varphi_j = \sum_{i=1}^N \hat{z}_i^{(j)} \sum_{t=2}^\tau \hat{P}_{i|t,t}^{(j)}$
$\phi_j = \sum_{i=1}^N \hat{z}_i^{(j)} \sum_{t=2}^\tau \hat{P}_{i|t-1,t-1}^{(j)}$
$\Psi_j = \sum_{i=1}^N \hat{z}_i^{(j)} \sum_{t=2}^\tau \hat{P}_{i|t,t-1}^{(j)}$
$\Gamma_j = \sum_{i=1}^N \hat{z}_i^{(j)} \sum_{t=1}^\tau y_{i,t}(\hat{x}_{i,t}^{(j)})^T$
$\Lambda_j = \sum_{i=1}^N \hat{z}_i^{(j)} \sum_{t=1}^\tau y_{i,t}(y_{i,t})^T$
$\hat{N}_j = \sum_{i=1}^N \hat{z}_i^{(j)}$, $\alpha_j^* = \frac{\hat{N}_j}{N}$
$C_j^* = \Gamma_j(\Phi_j)^{-1}$, $A_j^* = \Psi_j(\phi_j)^{-1}$
$R_j^* = \frac{1}{\tau \hat{N}_j}\left(\Lambda_j - C_j^*\Gamma_j\right)$
$Q_j^* = \frac{1}{(\tau-1)\hat{N}_j}\left(\varphi_j - A_j^*\Psi_j^T\right)$
$\mu_j^* = \frac{1}{\hat{N}_j}\sum_{i=1}^N \hat{z}_i^{(j)}\hat{x}_{i,1}^{(j)}$
$S_j^* = \hat{V}_{1,1}^{(j)} + \frac{1}{\hat{N}_j}\sum_{i=1}^N \hat{z}_i^{(j)}(\hat{x}_{i,1}^{(j)} - \mu_j^*)(\hat{x}_{i,1}^{(j)} - \mu_j^*)^T$
$\Theta_j = \{\alpha_j^*, A_j^*, Q_j^*, R_j^*, C_j^*, \mu_j^*, S_j^*\}$
end for
until convergence
Output: $\Theta = \{\Theta_j\}_{j=1}^K$

4.1. Relationship to prior work

For a single component ($j = 1$) and a single observation ($N = 1$), the EM algorithm for a dynamic texture mixture reduces to the classical EM algorithm for learning a linear dynamical system [17, 20, 21]. A linear dynamical system (1) is a generalization of the factor analysis model [15], a statistical model which explains an observed vector as a combination of measurements which are driven by independent factors. In the LDS framework, the time index t becomes the index of the independent observations y_t. The factors x_t (the hidden state) are independent (hence $A = 0$) and distributed as $\mathcal{N}(0, I)$ (i.e. $S = Q = I$ and $\mu = 0$). The observation y_t is then a function of the factors x_t, the factor loading matrix C (which explains how each factor influences the observation vector), and the observation noise $\mathcal{N}(0, R)$ where R is a diagonal matrix. With the appropriate restrictions on the mixture parameters, the EM algorithm for a dynamic texture mixture reduces to the EM algorithm used for learning a mixture of factor analyzers [22]. In particular, this requires setting $S_j = Q_j = I$ and $A_j = 0$ for each factor analysis component, and $\tau = 1$ since there are no temporal dynamics.

The dynamic texture mixture is also related to "switching" linear dynamical models, where the system parameters are selected via a separate Markovian switching variable as the time series progresses. Variations of these models include [23] where only the observation matrix C switches, [24] where the state parameters switch (A and Q), and [25] where the observation and state parameters switch (C, R, A, and Q). These three models are not mixtures of linear dynamical systems, and only have one state variable which evolves according to the active system parameters at each time step.

In contrast to switching models with a single state variable, the model proposed by Ghahramani [26] switches the observed variable between the output of different linear dynamic systems at each time step. Each LDS has its own observation matrix and state variable, which evolves according to its own system parameters. The difference between the Ghahramani model and the mixture of dynamic textures is that the Ghahramani model can switch between LDS outputs *at each time step*, whereas the mixture of dynamic textures selects an LDS *only once* at time $t = 1$, and never switches from it. Hence, the mixture of dynamic textures can be seen as a special case of the Ghahramani model, where the initial probabilities of the switching variable are the mixture component probabilities α_j, and the Markovian transition matrix of the switching variable is equal to the identity matrix.

This has consequences of significant practical importance. In particular, the ability to switch at each time step in the Ghahramani model results in a posterior distribution that is a Gaussian mixture with a number of terms that increases exponentially with time [26]. Thus, exact inference on the Ghahramani model is intractable, and the EM-style of learning requires approximate methods (e.g. variational approximations). In contrast, because the dynamic texture mixture selects only one LDS for an observed sequence, the posterior is a mixture with a constant number of Gaussians and exact inference in the dynamic texture mixture model is tractable, and hence the EM algorithm introduced above is exact.

Applications of switching linear models are numerous, including tracking of multiple objects with sensor data [23], human motion modeling [24], economic growth modeling [25], and respiration modeling of people with sleep apnea [26].

4.2. Clustering and motion segmentation

Video clustering is an important problem in various areas of computer vision. For example, it can be used to uncover high-level patterns of structure in a video stream (e.g. recurring events, events of high and low probability, outlying events, etc.) and has, therefore, application to problems such as surveillance, novelty detection, video summarization (by shot clustering), or remote monitoring of various types of environments. It can also be applied to the entries of a video database in order to automatically create a taxonomy of video classes that can then be used for database organization or video retrieval. Under the mixture of dynamic textures representation, a set of video sequences can be naturally clustered by first learning the mixture that best fits the entire collection of sequences, and then assigning each sequence to the mixture component with largest posterior probability of having generated it,

$$\ell_i = \operatorname*{argmax}_j \log p(y_i; \Theta_j) + \log \alpha_j. \tag{12}$$

In addition to clustering different video sequences, the mixture of dynamic textures is also a natural representation for the problem of segmenting a single video sequence into various components of homogeneous appearance and motion. In particular, these components can be segmented by dividing the sequence into a set of localized spatio-temporal patches and then clustering these patches. For example, the segmentation results of the following section were obtained by collecting video patches from the video sequence using a $p \times p$ spatially-sliding window (that fills the entire temporal volume), and clustering them into K classes. A segmented image was then produced using a voting scheme, where each pixel in a patch receives a vote for the class of that patch, as given by the clustering. The pixels were then assigned to the class with the most votes. Finally, a 3×3 maximum vote filter was used to smooth the segmented video regions.

While the idea of using EM for clustering or motion segmentation is not novel [6, 7, 27–29], the mixture of dynamic textures representation enables its application to a class of visual processes that has traditionally been quite challenging for clustering and motion segmentation algorithms. This is illustrated in the subsequent section.

5. Experimental evaluation

We evaluated the performance of the dynamic texture mixture model through experiments with clustering of traffic video, and motion segmentation on both synthetic and real video sequences.

5.1. Video clustering results

Clustering was performed on 133 video sequences of vehicle highway traffic [30]. Each video sequence was preprocessed by converting it into grayscale, downsampling it by four, subtracting the mean, normalizing the pixels to unit variance, and clipping the video frames to 48×48 pixels. The video sequences contained a variety of moving and stopped traffic, and were clustered into 4 classes. Figure 2 shows four typical sequences for each of the four clusters. These examples, and further analysis of the sequences in each cluster, reveal that the clusters are in agreement with classes frequently used in the perceptual categorization of traffic: stopped traffic ("traffic jam"), light traffic, slow traffic, and medium traffic.

Note that this sort of "perceptually plausible" clustering would be extremely difficult to obtain with traditional motion representations based on optical flow or parametric motion representations. Vehicle tracking and counting would be more likely to produce results equivalent to those achieved by dynamic texture mixture modeling, but would entail both 1) significant technical challenges (most vehicles occupy a very small number of image pixels and would be quite difficult to track) and 2) tremendous computational complexity (because there can be many vehicles to track). Furthermore, assuming that tracking is feasible, there would be a need to cluster the collections of tracks produced by each sequence. It is not clear that this problem, by itself, could be solved in a more natural or efficient way than the solution based on the dynamic texture mixture model.

Figure 2. Example of clustering traffic video into four classes, corresponding to (top to bottom) stopped traffic, light traffic, slow traffic, and medium traffic. Four typical sequences are shown for each of the four clusters.

5.2. Motion segmentation results

For the segmentation experiments, all video was converted into grayscale, 5×5 video patches were used, and the number of principal components was $n = 10$. Patches with an average pixel variance (in time) of less than 50 were marked as static background. Figure 3 shows the segmentation of a composite video containing regions of water, smoke, and fire using $K = 3$ clusters.

Segmentation of the motion in a highway traffic scene using $K = 4$ clusters is shown in Figure 4. The algorithm has segmented the video into regions of traffic which are moving away from the camera (the two large regions on the right) and moving towards the camera (the regions on the left). The region with traffic moving towards the camera has been segmented into two regions because of perspective effects due to car motion towards the camera.

While not perfect, these results are, once again, significantly better than what could be achieved with traditional representations. Note that 1) the motion information is quite sparse (there are significant gaps between cars), and 2) the perspective effects are extreme (cars at a distance occupy little more than a single pixel). The dynamic texture model could also be explicitly extended to account for some of the problems, e.g. by explicitly accounting for the drastic perspective deformation to which the dynamic texture components are subject. We intend to consider such extensions in the future.

Finally, Figure 5 shows the segmentation of a waterfall scene using a window size of 15×15 pixels and $K = 4$ clusters. The different segmented regions correspond to regions of different water dynamics (e.g. fast moving water, turbulent water, and slow moving water). Once again, the segmentation is plausible from a perceptual point of view (video is available at [31]) and would be difficult to achieve with classical motion models.

References

[1] B. K. P. Horn. *Robot Vision*. McGraw-Hill Book Company, 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] P. Anandan, J. Bergen, K. Hanna, and R. Hingorani. Hierarchical Model-Based Motion Estimation. *Motion Analysis and Image Sequence Processing*, Kluwer Academic Press, 1993.

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

[7] H. Sawhney and S. Ayer. Compact Representations of Videos Through Dominant and Multiple Motion Estimation. *IEEE Transactions in Pattern Analysis and Machine Intelligence*, vol. 18, August 1996.

[8] M. Isard and A. Blake. Condensation – Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision*, Vol. 29(1), pp. 5-28, 1998.

[9] M. Irani, B. Rousso, and S. Peleg. Detecting and Tracking Multiple Moving Objects Using Temporal Integration. *Proc. ECCV*, 1992.

[10] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-Based Object Tracking. *IEEE Transactions in Pattern Analysis and Machine Intelligence*, vol 25(5), pp 564-575, 2003.

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

[12] G. Doretto, D. Cremers, P. Favaro, S. Soatto. Dynamic texture segmentation. In *IEEE International Conference on Computer Vision*, vol. 2, pp 1236-42, 2003.

[13] P. Saisan, G. Doretto, Y. Wu, and S. Soatto. Dynamic texture recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, Proceedings*, volume 2, pages 58–63, 2001.

[14] A. B. Chan and N. Vasconcelos. Probabilistic kernels for the classification of auto-regressive visual processes. In *IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, 2005.

[15] S. Roweis and Z. Ghahramani. A unifying review of linear Gaussian models. *Neural Computation*, Vol. 11, pp 305-345, 1999.

[16] P. Van Overschee and B. De Moor. N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30:75–93, 1994.

[17] 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 253-264, 1982.

[18] 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.

[19] A. B. Chan and N. Vasconcelos The EM algorithm for mixtures of dynamic textures. Technical Report SVCL-TR-2005-02, `http://www.svcl.ucsd.edu`, UCSD, 2005.

[20] V. Digalakis, J. R. Rohlicek, and M. Ostendorf. ML estimation of a stochastic linear system with the EM algorithm and its application to speech recognition. *IEEE Transactions on Speech and Audio Processing*, Vol. 1(4), pp 431-442, 1993.

[21] Z. Ghahramani and G. Hinton. Parameter estimation for linear dynamical systems. Tech Report CRG-TR-96-2, Department of Computer Science, University of Toronto, 1996.

[22] Z. Ghahramani and G. Hinton. The EM algorithm for mixtures of factor analyzers. Tech Report CRG-TR-96-1, Department of Computer Science, University of Toronto, 1997.

[23] R. H. Shumway and D. S. Stoffer. Dynamic linear models with switching. *Journal of the American Statistical Association*, Vol 86, pp 763-769.

[24] V. Pavlović, J. M. Rehg, and J MacCormick. Learning switching linear models of human motion. In *Neural Information Processing Systems 13*, 2000.

[25] C.-J. Kim. Dynamic linear models with Markov-switching. *Journal of Econometrics*, Vol. 60, pp 1-22, 1994.

[26] Z. Ghahramani and G. Hinton. Switching state-space models. Tech Report CRG-TR-96-3, Department of Computer Science, University of Toronto, 1996.

[27] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley and Sons, 2001.

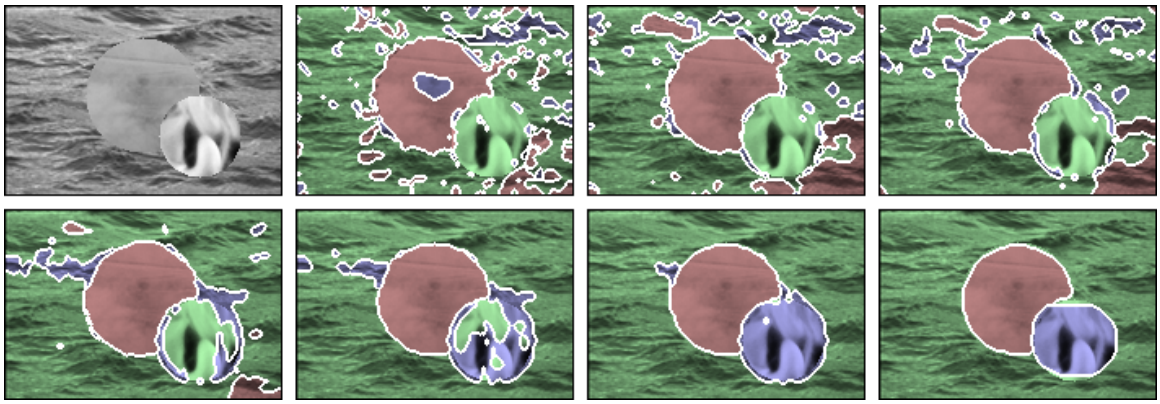

Figure 3. Segmentation of a composite video containing regions of water, smoke, and fire. (top-left to bottom-right) A frame from the video, the segmentation at iterations 1 through 6, and the final segmentation (bottom-right).

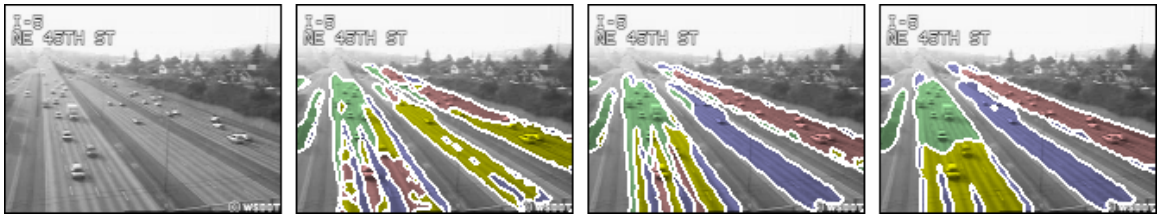

Figure 4. Segmentation of a highway traffic video. (left to right) A frame from the video, segmentation at iterations 1 and 10, and the final segmentation.

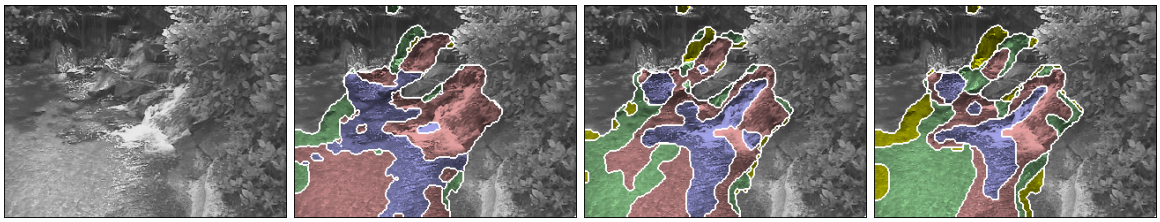

Figure 5. Segmentation of a waterfall scene. (left to right) frame from the video, segmentation at iterations 1 and 4, and the final segmentation.

[28] A. Jepson and M. Black. Mixture Models for Optical Flow. *Proc. CVPR*, 1993.

[29] Y. Weiss. Smoothness in Layers: Motion Segmentation Using Non-parametric Mixture Estimation. *Proc. CVPR*, 1997.

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

[31] `http://www.svcl.ucsd.edu/projects/motiondytex/`