Joint Motion Segmentation and Background Estimation in Dynamic Scenes

Adeel Mumtaz Weichen Zhang Antoni B. Chan
Department of Computer Science, City University of Hong Kong
adeel.mumtaz@my.cityu.edu.hk, wczhang4-c@my.cityu.edu.hk, abchan@cityu.edu.hk

Abstract

We propose a joint foreground-background mixture model (FBM) that simultaneously performs background estimation and motion segmentation in complex dynamic scenes. Our FBM consist of a set of location-specific dynamic texture (DT) components, for modeling local background motion, and set of global DT components, for modeling consistent foreground motion. We derive an EM algorithm for estimating the parameters of the FBM. We also apply spatial constraints to the FBM using an Markov random field grid, and derive a corresponding variational approximation for inference. Unlike existing approaches to background subtraction, our FBM does not require a manually selected threshold or a separate training video. Unlike existing motion segmentation techniques, our FBM can segment foreground motions over complex background with mixed motions, and detect stopped objects. Since most dynamic scene datasets only contain videos with a single foreground object over a simple background, we develop a new challenging dataset with multiple foreground objects over complex dynamic backgrounds. In experiments, we show that jointly modeling the background and foreground segments with FBM yields significant improvements in accuracy on both background estimation and motion segmentation, compared to state-of-the-art methods.

1. Introduction

Separating the background from foreground is a fundamental step in computer vision applications. Current methods for background subtraction work well on scenes where the background is mostly static over short periods of time [1–3]. For scenes with dynamic backgrounds (*e.g.*, moving tree leaves, water, fountain), the background motion field can be represented using dynamic textures (DTs) [4–6], a probabilistic motion model that treats the video a sample from a linear dynamical system. To separate foreground from background, the majority of background models require manually setting a threshold on the background score, which can vary significantly across scenes [7]. In addition, most methods require prior knowledge in the form of a "clean" training video containing only the background.

Dynamic texture models have also shown promise in clustering the microscopic and macroscopic motion patterns present in dynamic scenes [8–10]. [8] performs motion segmentation by clustering video patches using a mixture of DTs. However, one drawback is that this method is based purely on motion, and hence will fail to segment an object that has stopped moving. For example, the crowd segmentation used in [11] treats temporarily stopped pedestrians as background, and hence cannot count stationary people.

In this paper, we propose a joint foreground-background mixture (FBM) model for simultaneous motion segmentation and background estimation in dynamic scenes (see Fig. 1). The FBM consists of a set of location-specific background DTs, for modeling local background motion, and a set of global foreground DTs, for modeling global consistent motion of the foreground. A Markov random field (MRF) grid is used to add spatial constraints to the segmentation and reduce spurious noise. Our proposed joint model addresses the above problems associated with performing background estimation and motion segmentation separately: 1) our model does not require a threshold since the background model can be directly compared with the foreground motion models; 2) our model does not require a dedicated training video to learn the background; 3) our model can segment temporarily stopped objects. Finally, because both the background and foreground are jointly estimated, our model can more accurately separate foreground segments and background in complex dynamic scenes, compared to separately performing background subtraction or motion segmentation.

The contributions of our work are three-fold. First, we propose a novel foreground-background mixture (FBM) model, based on dynamic textures, for jointly representing the background and foreground motions in dynamic scenes. Second, we derive an EM algorithm to learn the parameters of the FBM, as well as a variational approximation to the posterior, and develop an adaptive threshold-based initialization strategy. Third, we evaluate the performance of FBM on background subtraction and motion segmentation in challenging dynamic scenes. Because most previously available datasets consist of a single foreground object and

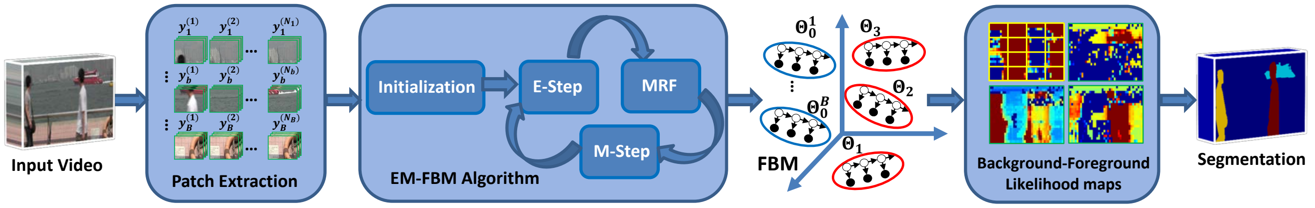

Input Video
$y_1^{(1)}$ $y_1^{(2)}$ ... $y_1^{(N_1)}$
$y_b^{(1)}$ $y_b^{(2)}$ ... $y_b^{(N_b)}$
$y_B^{(1)}$ $y_B^{(2)}$ ... $y_B^{(N_B)}$
Patch Extraction
Initialization
E-Step
MRF
M-Step
EM-FBM Algorithm
Θ_0^1
Θ_3
Θ_2
Θ_0^B
Θ_1
FBM
Background-Foreground Likelihood maps
Segmentation

Figure 1: Joint learning procedure for the foreground-background mixture model (FBM). The input video is divided into a set of video patches (spatiotemporal cubes). An FBM is trained using the EM algorithm with MRF constraints, resulting in B location-specific background DTs and K global foreground DTs. Finally, likelihood maps for the background and foregrounds are compared to perform joint motion segmentation and background estimation.

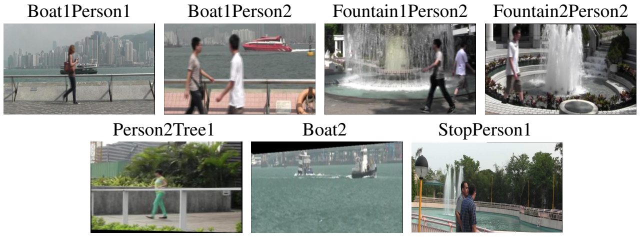

Boat1Person1 Boat1Person2 Fountain1Person2 Fountain2Person2
Person2Tree1 Boat2 StopPerson1

Figure 2: Example frames from our *FBDynScn* dataset.

relatively simple background [5, 6, 12], we introduce a new challenging dynamic scenes dataset *FBDynScn*, which consists of seven sequences with multiple foreground objects (*e.g*., boat, person) over complex backgrounds (*e.g*., fountain, trees, water) (see Fig. 2).

2. Related work

A number of techniques for adaptive background subtraction are available, exemplified by the seminal work of Stauffer and Grimson (SG) [1], which uses an adaptive Gaussian mixture model (GMM). Since then a number of extensions to the SG mixture, which, for example, include properties of local image neighborhoods and global consistency, have been proposed [2, 3]. These methods assume that the background is relatively *static* over short time scales, which makes them perform poorly in highly dynamic scenes. Similarly static backgrounds are assumed in some moving object detection methods [7]. Joint domain-range methods [13, 14] use a joint feature space to model the foreground and background at each pixel, and perform background subtraction by comparing the foreground and background scores. However, [13, 14] are based on color distributions, and hence are not suitable for complex dynamic scenes. They also cannot perform segmentation of the foreground into multiple motions.

Several methods have also been proposed for modeling dynamically moving backgrounds. [15] performs background subtraction by separating "salient" (foreground) motion from the background motion, while [16] integrates moving object detection and background learning into a single process using a low-rank representation of the background to accommodate the global variations. Other methods for modeling dynamic backgrounds are based on dynamic textures (DT) [17]. In [6], a DT models the entire video frame, and a robust Kalman filter is used to mark pixels that are not well explained by the DT as foreground. In [5], a local PCA-based DT method is proposed where each patch in the current frame is marked as foreground if it is not well modeled by the PCA basis. Finally, [4] proposes an extension of the adaptive SG model, where the mixture components are DTs, and a corresponding online learning algorithm to account for changes in scene over time.

DT models have also been applied to motion segmentation. [8] proposes a mixture of DTs for clustering spatio-temporal video patches to obtain a motion segmentation, and yields improved accuracy on complex motions, compared with traditional motion representations such as optical flow [18]. The layered dynamic texture (LDT) [9] models the whole video as a composition of layers, each modeled by a separate DT. Note that [8, 9] cannot be directly used for background subtraction. Each segment must correspond to a unique motion, and hence backgrounds with mixed motions (*e.g*., water, trees, and static) will be over-segmented. Other layered models [19, 20] perform segmentation by representing a video as a superposition of subject layers, undergoing homogeneous motion over a background layer. These models are based on optical flow and parametric motion that assume a piece-wise planar world, and hence are not applicable to scenes with backgrounds or foregrounds with complex dynamic appearance.

Our proposed FBM is a natural combination of location-specific dynamic background models (*e.g*., [4]) and DT motion segmentation [8], but with the following 3 challenges: 1) merging location-specific background DTs and global foreground DTs into a unified mixture, and proper handling of motion and non-motion areas; 2) proper initialization of background/foreground components for EM; 3) smoothness constraints (MRF) to regularize the model. To the best of our knowledge there exists no previous method that performs joint learning of foreground motions and background motions in dynamic scenes. Our FBM can be seen as an extension of [13, 14] to use *dynamic* appearance models. However, in contrast to [13, 14], our FBM also segments the foreground into *multiple* motions. In contrast to traditional background models (e.g., [4]), our FBM does not require a manually selected threshold to perform the background separation, and does not need a separate training video. In contrast to the motion segmentation of [8], our FBM can segment stopped objects and can segment complex backgrounds with mixed motions. In contrast to [16], which only produces a single foreground segment, our FBM can segment the foreground into multiple motions.

Our FBM is inspired by [21], which does feature selection by augmenting a GMM with extra components to model non-selected features as noise, and by [12] which does joint object categorization and motion segmentation.

3. Foreground-background mixture models

In this section, we propose our foreground-background mixture model. We begin with a brief review of the the dynamic texture (DT) and dynamic texture mixture (DTM).

3.1. Dynamic textures

A dynamic texture [22, 23] is a generative model for both the appearance and the dynamics of video sequences. It consists of a random process containing an *observation variable* y_t, which encodes the appearance of the video frame at time t, and a *hidden state variable* x_t, which encodes the dynamics of the video over time. The state and observation variables are related through the *linear dynamical system* (LDS) defined by

$$\begin{cases} x_{t+1} = Ax_t + v_t \\ y_t = Cx_t + w_t \end{cases}, \tag{1}$$

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 a *state transition matrix* and $C \in \mathbb{R}^{m\times n}$ is an *observation matrix* (e.g. containing the *principal components* of the video sequence when learned with [23]). The *driving noise process* v_t is normally distributed with zero mean and covariance Q, i.e. $v_t \sim \mathcal{N}(0, Q,)$ where $Q \in \mathbb{S}^n_+$ is a positive-definite $n\times n$ matrix. The *observation noise* w_t is also zero mean and Gaussian, with covariance R, i.e. $w_t \sim \mathcal{N}(0, R,)$ where $R \in \mathbb{S}^m_+$. The dynamic texture is specified by the parameters $\Theta = \{A, Q, C, R, \mu, S\}$.

While a DT models a time-series as a single sample from a LDS, the dynamic texture mixture (DTM) [8] models multiple time-series as samples from a set of K DTs. The probability of a given video sequence $y_{1:\tau}$ under a DTM with K dynamic texture components $\{\Theta_1, \ldots, \Theta_K\}$ having prior probabilities $\alpha = \{\alpha_1, \ldots, \alpha_K\}$ is

$$p(y_{1:\tau}) = \sum_{j=1}^{K} \alpha_j p(y_{1:\tau}|\Theta_j), \tag{2}$$

where $p(y_{1:\tau}|\Theta_j)$ is the observation likelihood function of a DT with parameters Θ_j.

3.2. Foreground-background mixture model

The foreground-background mixture model (FBM) consists of two sets of DTs for simultaneous background estimation and motion segmentation: 1) a set of location-specific DTs that model local background motions; 2) a set of non-location-specific DTs that model global consistent motion in the foreground. Our proposed FB mixture and joint estimation procedure is summarized in Fig. 1.

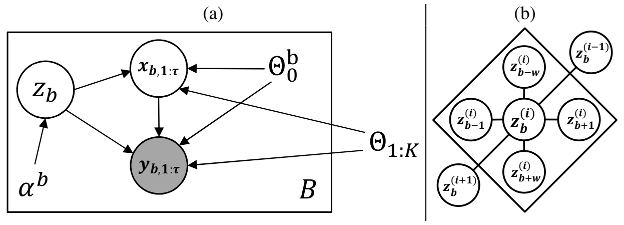


Figure 3: (a) Graphical model of the foreground-background mixture model. There are B replicas of the original dynamic texture (DT) mixture model (one for each location b). The parameters for the foreground components $\Theta_{1:K}$ are shared across all locations b, whereas each location has its own background DT Θ_0^b and mixture weights α^b; (b) MRF neighborhood for $z_b^{(i)}$.

The video ($W \times H \times T$) is split into a set of overlapping video patches ($p \times p \times \tau$ spatio-temporal cubes), extracted along a regularly spaced grid. There are a total of B background locations in the video frame, each having a total of N_b video patches along temporal dimension. In the FBM, each location b is associated with one location-specific background DT component Θ_0^b, while the foreground is modeled with K DT components $\{\Theta_1, \ldots, \Theta_K\}$. Note that we use the index 0 for the background component at b, and indices 1 to K for the foreground components. Under the FBM, the video patch $y_{b,1:\tau}$ observed at location b is a sample from a mixture of its background DT and the K global foreground DTs, *i.e.*, $\{\Theta_0^b, \Theta_1, \ldots, \Theta_K\}$,

$$p(y_{b,1:\tau}) = \alpha_0^b p(y_{b,1:\tau}|\Theta_0^b) + \sum_{j=1}^{K} \alpha_j^b p(y_{b,1:\tau}|\Theta_j), \tag{3}$$

where $\alpha^b = \{\alpha_0^b, \alpha_1^b, \ldots, \alpha_K^b\}$ are the component weights, with $\sum_{m=0}^{K} \alpha_m^b = 1$. $p(y_{b,1:\tau}|\Theta_0^b)$ is the class conditional density of the b^{th} background DT, parameterized by $\Theta_0^b = \{A_0^b, Q_0^b, C_0^b, R_0^b, \mu_0^b, S_0^b\}$, while $p(y_{b,1:\tau}|\Theta_j)$ is the class conditional density of the j^{th} foreground DT, parameterized by $\Theta_j = \{A_j, Q_j, C_j, R_j, \mu_j, S_j\}$.

The system of equations that define the mixture of foreground and background DTs is

$$\begin{cases} x_{b,t+1} = A_{z_b} x_{b,t} + v_{b,t} \\ y_{b,t} = C_{z_b} x_{b,t} + w_{b,t} \end{cases} \tag{4}$$

where $z_b \sim \text{multinomial}(\alpha_0^b, \alpha_1^b, \ldots, \alpha_K^b)$ is the assignment variable that indicates the mixture component from which the observation is drawn. The initial condition is given by $x_{b,1} \sim \mathcal{N}(\mu_{z_b}, S_{z_b})$, and the noise processes by $v_{b,t} \sim \mathcal{N}(0, Q_{z_b})$ and $w_{b,t} \sim \mathcal{N}(0, R_{z_b})$. When $z_b = 0$, then the DT parameters are selected from Θ_0^b, while for $z_b > 0$ the DT parameters are from Θ_{z_b}. The graphical model for the FBM is presented in Fig. 3. Since there are K foreground DTs and B background DTs, the complexity of exact inference on the FBM is $K + B$ times that of the underlying DT. Finally, the complete set of parameters for the FBM is $\boldsymbol{\Theta} = \{\{\alpha^b, \Theta_0^b\}_{b=1}^{B}, \{\Theta_j\}_{j=1}^{K}\}$.

3.3. EM algorithm for parameter estimation

Given a set of video patches $\{\mathbf{y}_b^{(i)}\}_{i=1}^{N_b}$ at each background location b, we aim to estimate the parameters $\mathbf{\Theta}$ of the FBM that maximizes the likelihood of the data [24],

$$\mathbf{\Theta}^* = \underset{\mathbf{\Theta}}{\operatorname{argmax}} \sum_{b=1}^{B} \sum_{i=1}^{N_b} \log p(\mathbf{y}_b^{(i)}; \mathbf{\Theta}). \tag{5}$$

When the probabilistic model depends on hidden variables (*e.g.*, the output of the system is observed, but its state is unknown), the maximum-likelihood solution can be found with the EM algorithm [25]. For FBM, each observation $\mathbf{y}_b^{(i)}$ at location b is associated with the missing data: 1) assignment $z_b^{(i)}$ to one of the global foreground or local background mixture components, and 2) hidden state sequence $\mathbf{x}_b^{(i)}$ that produces $\mathbf{y}_b^{(i)}$. Each EM iteration consists of:

$$\mathrm{E-Step}: \mathcal{Q}(\mathbf{\Theta}; \hat{\mathbf{\Theta}}) = \mathbb{E}_{X,Z|Y;\hat{\mathbf{\Theta}}}[\log p(X, Y, Z; \mathbf{\Theta})], \tag{6}$$

$$\mathrm{M-Step}: \hat{\mathbf{\Theta}}^* = \underset{\mathbf{\Theta}}{\operatorname{argmax}}\, \mathcal{Q}(\mathbf{\Theta}; \hat{\mathbf{\Theta}}), \tag{7}$$

where $p(X, Y, Z; \mathbf{\Theta})$ is the complete-data likelihood of the observations (video patches) $Y = \{\{\mathbf{y}_b^{(i)}\}_{i=1}^{N_b}\}_{b=1}^{B}$, the corresponding hidden state sequences $X = \{\{\mathbf{x}_b^{(i)}\}_{i=1}^{N_b}\}_{b=1}^{B}$, and the assignment variables $Z = \{\{z_b^{(i)}\}_{i=1}^{N_b}\}_{b=1}^{B}$.

As is usual in the EM literature [25], we introduce an indicator $\mathbf{z}_{b,i,m} \in \{0, 1\}$, such that $\mathbf{z}_{b,i,m} = 1$ if and only if $z_b^{(i)} = m$. The complete-data likelihood is then

$$\begin{aligned} &p(X, Y, Z) \\ &= p(Z) \prod_{b=1}^{B} \prod_{i=1}^{N_b} p(\mathbf{x}_b^{(i)}, \mathbf{y}_b^{(i)} | \Theta_0^b)^{\mathbf{z}_{b,i,0}} \prod_{j=1}^{K} p(\mathbf{x}_b^{(i)}, \mathbf{y}_b^{(i)} | \Theta_j)^{\mathbf{z}_{b,i,j}}, \end{aligned} \tag{8}$$

where $p(\mathbf{x}_b^{(i)}, \mathbf{y}_b^{(i)} | \Theta)$ is the density for a DT Θ, and

$$p(Z) = \prod_{b=1}^{B} \prod_{i=1}^{N_b} \prod_{m=0}^{K} (\alpha_m^b)^{\mathbf{z}_{b,i,m}}. \tag{9}$$

Applying the expectation of (6) to the log of the complete-data likelihood in (8) yields a $\mathcal{Q}$ function similar to that of the DTM in [8]. The E and M steps for FBM can then be derived by following a procedure similar to [8] (see supplemental for complete derivation).

The E-step consists of computing the conditional expectations with the Kalman smoothing filter [26],

$$\hat{x}_{b,t|m}^{(i)} = \mathbb{E}_{\mathbf{x}_b^{(i)} | \mathbf{y}_b^{(i)}, z_b^{(i)} = m} \left[x_{b,t}^{(i)} \right], \tag{10}$$

$$\hat{P}_{b,t,t|m}^{(i)} = \mathbb{E}_{\mathbf{x}_b^{(i)} | \mathbf{y}_b^{(i)}, z_b^{(i)} = m} \left[x_{b,t}^{(i)} (x_{b,t}^{(i)})^T \right], \tag{11}$$

$$\hat{P}_{b,t,t-1|m}^{(i)} = \mathbb{E}_{\mathbf{x}_b^{(i)} | \mathbf{y}_b^{(i)}, z_b^{(i)} = m} \left[x_{b,t}^{(i)} (x_{b,t-1}^{(i)})^T \right], \tag{12}$$

and the assignment probabilities,

$$\hat{\mathbf{z}}_{b,i,m} = \frac{\alpha_m^b p(\mathbf{y}_b^{(i)} | z_b^{(i)} = m)}{\sum_{k=0}^{K} \alpha_k^b p(\mathbf{y}_b^{(i)} | z_b^{(i)} = k)}, \tag{13}$$

where $p(\mathbf{y}_b^{(i)} | z_b^{(i)} = j)$ is the observation likelihood, which is calculated with the Kalman filter (see [26]). The expectations for each component $m \in \{0, \cdots, K\}$ are then aggregated over all video patches at location b, and then over all locations for foreground components $j \in \{1, \cdots, K\}$,

$$\begin{array}{ll} \hat{N}_m^b = \sum_i \hat{\mathbf{z}}_{b,i,m}, & \hat{N}_j = \sum_b \hat{N}_j^b, \\ \xi_m^b = \sum_i \hat{\mathbf{z}}_{b,i,m} \hat{x}_{b,1|m}^{(i)}, & \xi_j = \sum_b \xi_j^b, \\ \eta_m^b = \sum_i \hat{\mathbf{z}}_{b,i,m} \hat{P}_{b,1,1|m}^{(i)}, & \eta_j = \sum_b \eta_j^b, \\ \Phi_m^b = \sum_i \hat{\mathbf{z}}_{b,i,m} \sum_{t=1}^{\tau} \hat{P}_{b,t,t|m}^{(i)}, & \Phi_j = \sum_b \Phi_j^b, \\ \phi_m^b = \sum_i \hat{\mathbf{z}}_{b,i,m} \sum_{t=2}^{\tau} \hat{P}_{b,t-1,t-1|m}^{(i)}, & \phi_j = \sum_b \phi_j^b, \\ \varphi_m^b = \sum_i \hat{\mathbf{z}}_{b,i,m} \sum_{t=2}^{\tau} \hat{P}_{b,t,t|m}^{(i)}, & \varphi_j = \sum_b \varphi_j^b, \\ \Psi_m^b = \sum_i \hat{\mathbf{z}}_{b,i,m} \sum_{t=2}^{\tau} \hat{P}_{b,t,t-1|m}^{(i)}, & \Psi_j = \sum_b \Psi_j^b, \\ \Gamma_m^b = \sum_i \hat{\mathbf{z}}_{b,i,m} \sum_{t=1}^{\tau} y_{b,t}^{(i)} (\hat{x}_{b,t|m}^{(i)})^T, & \Gamma_j = \sum_b \Gamma_j^b, \\ \Lambda_m^b = \sum_i \hat{\mathbf{z}}_{b,i,m} \sum_{t=1}^{\tau} y_{b,t}^{(i)} (y_{b,t}^{(i)})^T, & \Lambda_j = \sum_b \Lambda_j^b. \end{array} \tag{14}$$

For the M-step, the parameters of each foreground components $j \in \{1, \cdots, K\}$ is updated with

$$\begin{array}{ll} \hat{C}_j = \Gamma_j (\Phi_j)^{-1}, & \hat{R}_j = \frac{1}{\tau \hat{N}_j} (\Lambda_j - \hat{C}_j \Gamma_j), \\ \hat{A}_j = \Psi_j (\phi_j)^{-1}, & \hat{Q}_j = \frac{1}{(\tau-1)\hat{N}_j} (\varphi_j - \hat{A}_j \Psi_j^T), \\ \hat{\mu}_j = \frac{1}{\hat{N}_j} \xi_j, & \hat{S}_j = \frac{1}{\hat{N}_j} \eta_j - \hat{\mu}_j (\hat{\mu}_j)^T, \end{array} \tag{15}$$

and the parameters for each background b is updated,

$$\begin{array}{l} \hat{C}_0^b = \Gamma_0^b (\Phi_0^b)^{-1}, \;\; \hat{R}_0^b = \frac{1}{\tau \hat{N}_0^b} (\Lambda_j - \hat{C}_0^b \Gamma_j), \\ \hat{A}_0^b = \Psi_0^b (\phi_0^b)^{-1}, \;\; \hat{Q}_0^b = \frac{1}{(\tau-1)\hat{N}_0^b} (\varphi_0^b - \hat{A}_0^b \Psi_j^T), \\ \hat{\mu}_0^b = \frac{1}{\hat{N}_0^b} \xi_0^b, \;\; \hat{S}_0^b = \frac{1}{\hat{N}_0^b} \eta_0^b - \hat{\mu}_0^b (\hat{\mu}_0^b)^T, \;\; \hat{\alpha}_m^b = \frac{\hat{N}_m^b}{N_b}. \end{array} \tag{16}$$

4. FBM with MRF constraints

In this section we add an MRF to the hidden assignment variables to encourage spatially smooth segmentations.

4.1. MRF constraints

The model we will consider is a FBM where the assignment variables $Z = \{z_b^{(i)}\}$ have MRF constraints based on their positions. Rather than assume that the $z_b^{(i)}$ are independent as in (9), we apply an MRF so that the assignments obey neighborhood constraints similar to [9],

$$p(Z) = \frac{1}{\mathcal{Z}} \left[\prod_{b=1}^{B} \prod_{i=1}^{N_b} V(z_b^{(i)}) \right] \cdot \prod_{((b,i),(d,n)) \in \mathcal{E}} V(z_b^{(i)}, z_d^{(n)}), \tag{17}$$

where $V(z_b^{(i)})$ is the self potential, and $V(z_b^{(i)}, z_d^{(n)})$ is the neighbor potential,

$$V(z_b^{(i)} = m) = \alpha_m^b, \tag{18}$$

$$V(z_b^{(i)}, z_d^{(n)}) = \begin{cases} \gamma_1, & z_b^{(i)} = z_d^{(n)} \\ \gamma_2, & z_b^{(i)} \neq z_d^{(n)} \end{cases}. \tag{19}$$

The set $\mathcal{E}$ contains all edges between neighbors, with each node indexed by the pair (b, i). In this paper we use the six-connected neighborhood, as shown in Fig. 3(b). Finally, $\mathcal{Z}$ is the normalization constant. Since an MRF is introduced on Z, there is no closed-form solution for inference.

4.2. Variational approximation to the posterior

We present a variational approximation to the posterior $p(X, Z|Y)$ (see supplemental for derivation). Define the approximate posterior $q(X, Z)$, which factorizes by sample,

$$p(X, Z|Y) \approx q(X, Z) = \prod_{b=1}^{B} \prod_{i=1}^{N_b} q(\mathbf{x}_b^{(i)}, z_b^{(i)}). \tag{20}$$

The optimal variational distribution is obtained by iterating between updating the variational parameters $h_{b,i,m}$,

$$\Delta_{b,i,m} = \sum_{((b,i),(d,n)) \in \mathcal{E}} \hat{\mathbf{z}}_{d,n,m}, \tag{21}$$

$$\log g_{b,i,m} = \log \alpha_m^b + \Delta_{b,i,m} \log \frac{\gamma_1}{\gamma_2}, \tag{22}$$

$$h_{b,i,m} = \frac{g_{b,i,m}}{\sum_{k=0}^{K} g_{b,i,k}}, \tag{23}$$

and the variational assignment probabilities,

$$\hat{\mathbf{z}}_{b,i,m} = \frac{h_{b,i,m} p(\mathbf{y}_b^{(i)} | z_b^{(i)} = m)}{\sum_{k=0}^{K} h_{b,i,k} p(\mathbf{y}_b^{(i)} | z_b^{(i)} = k)}. \tag{24}$$

In (21), $\Delta_{b,i,m}$ is the soft number of neighbors of $z_b^{(i)}$ assigned to component m. Finally, the variational posterior $q^*(\mathbf{x}_b^{(i)}, z_b^{(i)})$ is equivalent to the FBM posterior with independent $z_b^{(i)}$ (as in Section 3.2), but with prior probabilities as $h_{b,i,m}$ that are different for each sample i. The variational approximation is summarized in Alg. 1.

4.3. Summary

A summary of the EM algorithm for FBM using MRF is presented in Alg. 2. For initialization of background and foreground DTs, we use an adaptive threshold scheme where minimum variance and motion likelihood thresholds ($\mathfrak{T}$ and $\mathfrak{L}$) select patches for learning the initial DTs with [23] (see supplemental). After initialization, EM is run on all the patches. During EM, we assume that foreground DTs should only model patches with motion (foreground motion is always dynamic), while background DTs should model motion and non-motion patches (background can be static or dynamic). After EM converges, the segmentation is produced by assigning each video patch to the most likely mixture component (either background or labeled foreground), according to the posterior probability $\hat{\mathbf{z}}_{b,i,m}$. Stopped objects are detected by identifying non-motion patches that do not have high-likelihood under the background component.

Algorithm 1 Variational posterior assignments

```
Input: Set of video patches $Y$, FBM $\mathbf{\Theta}$.
Initialize $h_{b,i,m} = \frac{1}{K+1}$, $\forall\{b, i, m\}$.
Using (24), calculate $\hat{\mathbf{z}}_{b,i,m}$, $\forall\{b, i, m\}$.
repeat
    for $b = \{1, \dots, B\}$ and $i = \{1, \dots, N_b\}$ do
        {Update variational parameters of node $(b, i)$}
        Using (21)-(24), update $h_{b,i,m}$ and $\hat{\mathbf{z}}_{b,i,m}$, $\forall m$.
    end for
until convergence of $h_{b,i,m}$.
Output: variational parameters $\{h_{b,i,m}\}$, assignment probabilities $\{\hat{\mathbf{z}}_{b,i,m}\}$.
```

Algorithm 2 Variational EM for FBM

```
Input: Set of video patches $Y$, number of foreground components $K$, MRF parameters $\{\gamma_1, \gamma_2\}$.
Initialize FBM $\mathbf{\Theta} = \{\{\alpha^b, \Theta_0^b\}_{b=1}^B, \{\Theta_j\}_{j=1}^K\}$.
repeat
    {Expectation Step}
    Calculate variational approximation to $\{\hat{\mathbf{z}}_{b,i,m}\}$ using Algorithm 1.
    for $b = \{1, \dots, B\}$ and $i = \{1, \dots, N_b\}$ do
        Calculate the expectations in (10-12) for $\mathbf{y}_b^{(i)}$ and each DT in $\{\Theta_0^b, \Theta_1, \dots, \Theta_K\}$.
    end for
    Calculate aggregate expectations (14), $\forall b$, $\forall j$.
    {Maximization Step}
    for $j = \{1, \dots, K\}$ do
        Update foreground DT $\Theta_j$ with (15).
    end for
    for $b = \{1, \dots, B\}$ do
        Update background DT $\Theta_0^b$ and $\alpha^b$ with (16).
    end for
until convergence
Output: background models $\{\Theta_0^b\}_{b=1}^B$, foreground models $\{\Theta_j\}_{j=1}^K$, priors $\{\alpha^b\}_{b=1}^B$.
```

5. Experiments and results

In this section, we present applications of FBM on background estimation and motion segmentation.

5.1. Datasets

To evaluate the performance of FBM, we collect a new challenging dataset *FBDynSyn*, consisting of 7 videos containing multiple foreground objects over a complex background (e.g. boats and people over water, fountains, and trees), as depicted in Fig. 2. The videos are in grayscale with varying sizes (average size of $160 \times 304 \times 316$). We annotated each video with a ground truth segmentation of the foreground objects and background. We also tested our algorithm on the most challenging video (in terms of quantitative measures) "Sailing02" from [4].

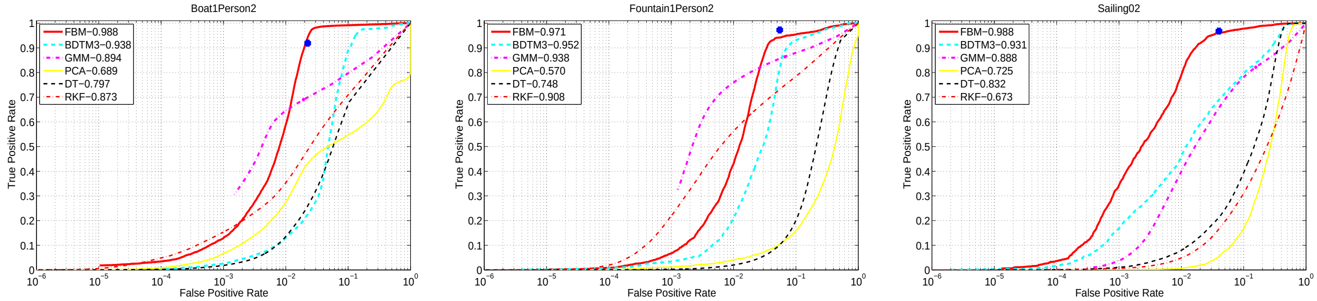


Figure 4: The ROC curves for background estimation on two videos from *FBDynScn* and Sailing02 from [4]. The AUC for each method is listed in the legend. The blue circle is the operating point of the FBM†.

†The FBM operating point does not lie on the ROC curve. The ROC curve is based on thresholding the log-likelihood of the background DT, whereas the operating point of FBM is based on comparing the posterior probabilities of background and foreground segments.

	AUC						FPR					
video	FBM	BDTM3 [4]	GMM [27]	PCA [4]	DT [5]	RKF [6]	FBM	BDTM3 [4]	GMM [27]	PCA [4]	DT [5]	RKF [6]
Sailing02	**0.988**	0.931	0.888	0.725	0.832	0.673	**0.016**	0.271	0.555	0.495	0.353	0.782
Boat1Person1	**0.994**	0.974	0.824	0.853	0.814	0.812	**0.004**	0.031	0.007	**0.004**	0.013	0.106
Boat1Person2	**0.988**	0.938	0.894	0.689	0.797	0.873	0.009	0.052	**0.005**	0.103	0.069	0.033
Fountain1Person2	**0.971**	0.952	0.938	0.570	0.748	0.908	**0.034**	0.073	0.175	0.847	0.518	0.332
Fountain2Person2	**0.973**	0.947	0.962	0.525	0.846	0.930	0.064	0.069	**0.035**	0.997	0.194	0.231
Person2Tree1	**0.985**	0.960	0.977	0.675	0.911	0.927	0.030	0.071	**0.013**	0.986	0.164	0.244
Boat2	**0.989**	0.977	0.784	0.971	0.917	0.707	**0.004**	0.007	0.110	0.005	0.014	0.250
average	**0.984**	0.954	0.895	0.715	0.838	0.833	**0.023**	0.082	0.129	0.491	0.189	0.283

Table 1: Background estimation results. The left side shows the AUC, while the right side is the FPR for TPR=0.90 (0.55 for Boat1Person1, Boat1Person2, and Boat2). Bold values indicate the best performance on each video.

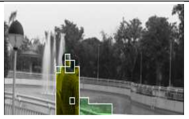
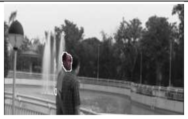
	FBM		DECOLOR[16]	
video	TPR	FPR	TPR	FPR
Sailing02	0.968	0.040	0.947	0.164
Boat1Person1	0.973	0.019	0.967	0.007
Boat1Person2	0.919	0.022	0.977	0.018
Fountain1Person2	0.972	0.055	0.791	0.007
Fountain2Person2	0.892	0.038	0.946	0.086
Person2Tree1	0.953	0.056	0.967	0.017
Boat2	0.955	0.022	0.931	0.008
StopPerson1	0.945	0.026	0.642	0.003
average	**0.947**	**0.035**	0.896	0.039
Example of stop case				

Table 2: Background estimation performance for FBM and DECOLOR at the operating point of the algorithms, and an example of detecting a temporarily stopped object.

5.2. Experimental setup

For our FBM, we divide each video into spatiotemporal overlapping patches with dimensions $10 \times 10 \times 15$ (step: $5 \times 5 \times 10$). The number of global foreground components K is set according to the number of motion components present in each video[1]. For the MRF, we use the neighborhood shown in Fig. 3(b), and set $\log \frac{\gamma_1}{\gamma_2} = 50$. To segment a video, an FBM with $n = 10$ is learned from the video using the EM algorithm (Alg. 2). For the initialization procedure, we set the minimum variance threshold $\mathfrak{T} = 1$ and the motion likelihood threshold $\mathfrak{L} = 100$.

We compare our FBM with several state-of-the-art methods in both background subtraction and motion segmentation[2]. For background subtraction, we compare with the adaptive GMM of [27], which automatically selects the number of components. We also consider the DT-based method [5] (denoted as DT) using a patch size of 7×7, and the robust Kalman filter (RKF) [6] (both using $n = 10$). From [4], we test the best performing adaptive background DTM method with 3 components (denoted as BDTM3). We also used the PCA model from [4] with patch size 7×7 and $n = 10$. We test DECOLOR [16], a recent moving object detector that also runs in batch mode like FBM. Finally, our dataset does not have a separate training video for each scene. To make a fair comparison, for background models that require training, we first train the model on the video, and then run background subtraction on the same video.

For motion segmentation, we compare against the DTM [8] with $K + 1$ components (the extra component is for the background motion). We extend the DTM by adding the same MRF constraints as the FBM. We also compare with the temporal-switching LDT [28], again using $K + 1$ components. Other parameter settings are the same as FBM.

To measure the accuracy of background estimation, ROC curves are calculated by sweeping a threshold on the background score image (e.g., from the background component in FBM), and calculating the true positive rate (TPR) and

[1] Similar background estimation results were obtained when setting $K = 1$, which collapses all foreground motions into a single class.

[2] Note that these models perform either background subtraction or motion segmentation tasks, whereas our FBM performs both simultaneously.

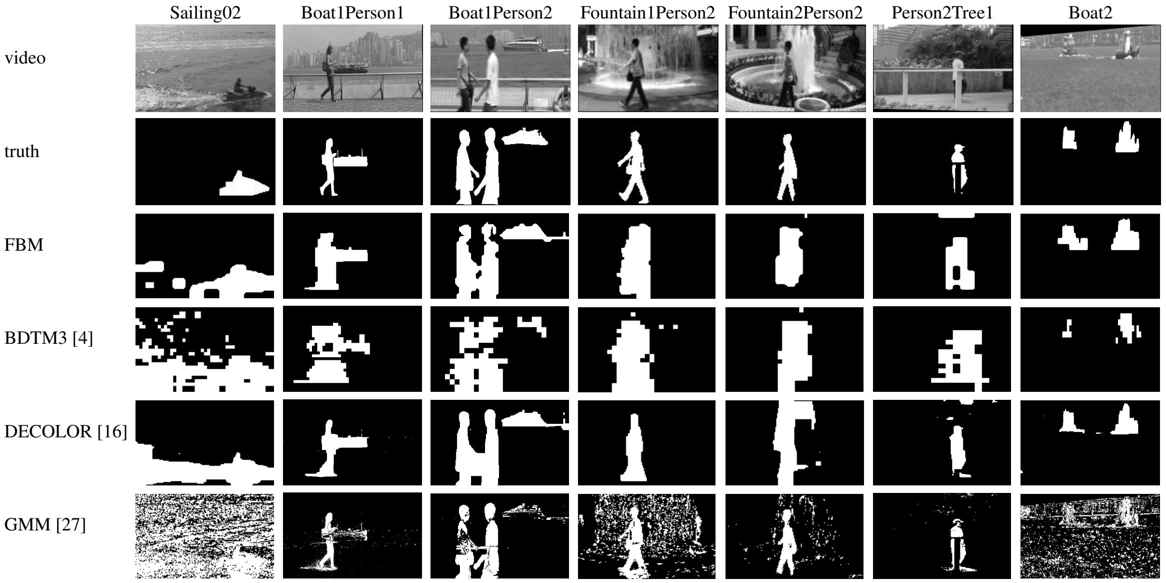


Figure 5: Example frames of background estimation using FBM and other methods. The results of FBM are based on the operating point (Table 2). For the other methods, the thresholds are set to yield a TPR of 0.90 or 0.55 (see Table 1).

false positive rate (FPR) with respect to the ground-truth background segment. The overall performance is measured by the area under the ROC curve (AUC). The motion segmentation results are evaluated using the Rand index (RI) [29] to measure the percentage of agreement between the ground-truth and segmentation masks.

5.3. Results on background estimation

Table 1 compares the AUC for FBM and the tested background subtraction methods. FBM has the highest average AUC of 0.984, while the next best method BDTM3 has an AUC of 0.954. Note that BDTM3 uses 3 background DT components at each location, where as FBM uses only a single background DT at each location. Despite this, FBM is able to achieve higher AUC by also modeling the global foreground motion. Fig. 4 shows the ROC curves for 3 videos. As the FPR is lowered, FBM typically maintains a higher TPR than other methods, especially in the high TPR regime (upper-right). Table 2 shows the performance of FBM and DECOLOR at the operating point of the algorithms. The operating point of FBM is typically in the high TPR regime (average of 0.947) with a corresponding low FPR of 0.035[3]. Compared to DECOLOR, FBM has higher average TPR (0.947 vs. 0.896) while maintaining a similar FPR (0.035 vs. 0.039). DECOLOR does poorly on a few videos with complex backgrounds (Fountain1Person2) or with stopped objects (StopPerson1).

Table 1 presents the FPR for a fixed TPR of 0.90 (or 0.55 for more difficult videos). For the same setting of TPR, our FBM achieves the lowest average FPR of 0.023, compared with other methods, e.g., 0.082 and 0.129 for BDTM3 and GMM. Fig. 5 presents examples of background estimation on each video. Since FBM is a patch-based framework, we do not get a fine-detailed foreground mask. FBM has the least noise as compared to other methods, which sometimes learn portions of the background as foreground. For BDTM3, the segmented foreground is typically larger than the actual foreground, creating more false positives than FBM. GMM obtains good details on the foreground mask, but also has a significant amount of false positive noise.

5.4. Results on motion segmentation

Table 3 shows the Rand index results on motion segmentation, while Fig. 6 presents examples of segmentation masks for each video. FBM significantly outperforms other motion segmentation methods with an average RI of 0.94 versus 0.76 and 0.51 for LDT and DTM. Even with an extra DT component, DTM is not able to model the complex background as a single segment. Instead, it oversegments the background and puts multiple foreground motions into the same segment (e.g., Boats2Person2). LDT performs well on some scenes where the background is homogeneous (e.g., Person2Tree1), and thus can be modeled well with one DT layer. However, LDT also fails on scenes with complex backgrounds with different dynamics (e.g., Fountain2Person2). In contrast, FBM can correctly segment both the complex background and different foreground motions.

Finally, FBM can successfully segment stopped objects (e.g., StopPerson1 in Fig. 6,), whereas pure motion segmentation methods, DTM and LDT, cannot segment these.

6. Conclusion

In this paper, we proposed a novel foreground-background mixture model that jointly performs motion segmentation and background estimation. We derive an EM algorithm for estimating the parameters of FBM, and

[3] The average TPR/FPR for FBM without MRF is 0.936/0.092.

	Boat1Person1	Boat1Person2	Fountain1Person2	Fountain2Person2	Person2Tree1	Boat2	StopPerson1	average RI
DTM [8]	0.7030	0.4890	0.3638	0.3885	0.4325	0.5379	0.6716	0.5123
LDT [9]	0.9524	0.7021	0.7769	0.3833	0.8646	0.7986	0.8668	0.7635
FBM	**0.9632**	**0.9428**	**0.9156**	**0.9388**	**0.9270**	**0.9610**	**0.9482**	**0.9424**

Table 3: Motion segmentation results on the *FBDynScn* data set.

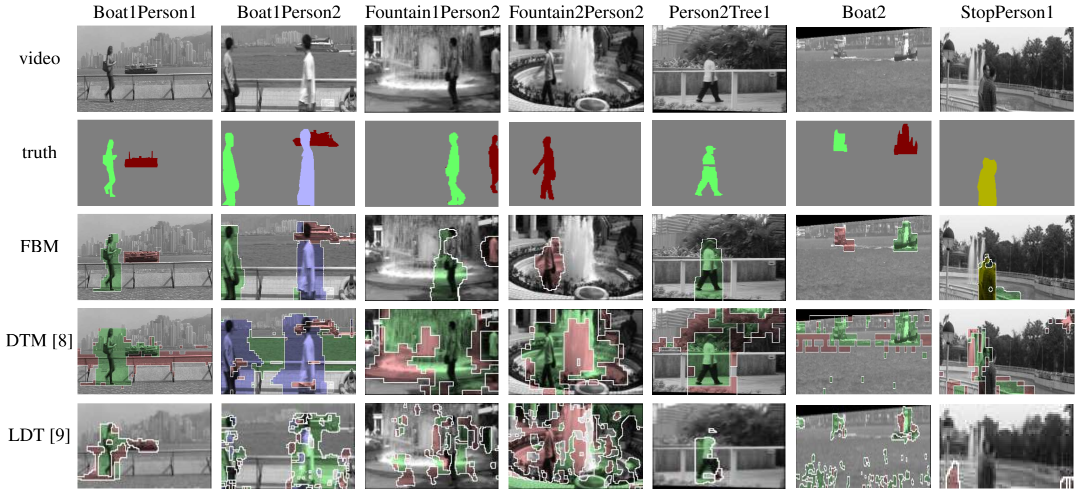


Figure 6: Example frames of motion segmentation on each video in *FBDynScn*. Foreground segments are colored as red, green, or indigo, while the background segment has no coloring. Stopped objects are colored yellow.

also derive a variational posterior for FBM with MRF constraints. Experiment results show that jointly estimating the background and foreground segments with the FBM can improve the accuracy of both background estimation and motion segmentation, compared to state-of-the-art methods. Once FBM is trained from a video, it can do online background estimation and motion segmentation on any new video frames. Future work will consider online updating, similar to [4], and automatically selecting the number of components, e.g. using a variational Bayesian framework.

References

[1] C. Stauffer and E. Grimson, "Learning patterns of activity using real-time tracking," *IEEE TPAMI*, vol. 22(8), pp. 747–57, 2000.

[2] M. Heikkila and M. Pietikainen, "A texture-based method for modeling the background and detecting moving objects," *IEEE TPAMI*, vol. 28, no. 4, 2006.

[3] G. Dalley, J. Migdal, and W. Grimson, "Background subtraction for temporally irregular dynamic textures," in *WACV 2008. IEEE Workshop on*, 2008.

[4] A. B. Chan, V. Mahadevan, and N. Vasconcelos, "Generalized stauffer-grimson background subtraction for dynamic scenes." *Mach. Vis. Appl.*, vol. 22, no. 5, pp. 751–766, 2011.

[5] A. Monnet, A. Mittal, N. Paragios, and V. Ramesh, "Background modeling and subtraction of dynamic scenes," in *ICCV*, 2003.

[6] J. Zhong and S. Sclaroff, "Segmenting foreground objects from a dynamic textured background via a robust kalman filter," in *ICCV*, 2003.

[7] G. Shu, A. Dehghan, and M. Shah, "Improving an object detector and extracting regions using superpixels," in *CVPR*, 2013.

[8] A. B. Chan and N. Vasconcelos, "Modeling, clustering, and segmenting video with mixtures of dynamic textures," *IEEE TPAMI*, 2008.

[9] A. Chan and N. Vasconcelos, "Layered dynamic textures," *IEEE TPAMI*, vol. 31, no. 10, pp. 1862–1879, 2009.

[10] A. Mumtaz, E. Coviello, G. Lanckriet, and A. Chan, "Clustering dynamic textures with the hierarchical em algorithm for modeling video," *IEEE TPAMI*, vol. 35(7), pp. 1606–1621, 2013.

[11] A. Chan and N. Vasconcelos, "Counting people with low-level features and bayesian regression," *Image Processing, IEEE Transactions on*, vol. 21, no. 4, pp. 2160–2177, 2012.

[12] D. Singaraju and R. Vidal, "Using global bag of features models in random fields for joint categorization and segmentation of objects," in *CVPR*, 2011.

[13] Y. Sheikh and M. Shah, "Bayesian modeling of dynamic scenes for object detection," *IEEE TPAMI*, 2005.

[14] M. Narayana, A. R. Hanson, and E. G. Learned-Miller, "Improvements in joint domain-range modeling for background subtraction," in *BMVC*, 2012.

[15] Y.-L. Tian and A. Hampapur, "Robust salient motion detection with complex background for real-time video surveillance," in *WACV/MOTIONS, IEEE Workshops on*, 2005.

[16] X. Zhou, C. Yang, and W. Yu, "Moving object detection by detecting contiguous outliers in the low-rank representation," *IEEE TPAMI*, vol. 35, no. 3, 2013.

[17] G. Doretto, D. Cremers, P. Favaro, and S. Soatto, "Dynamic texture segmentation," in *ICCV*, vol. 2, 2003.

[18] B. Horn and B. Schunk, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–204, 1981.

[19] D. Sun, J. Wulff, E. B. Sudderth, H. Pfister, and M. J. Black, "A fully-connected layered model of foreground and background flow," in *CVPR*, 2013.

[20] N. Jojic and B. Frey, "Learning flexible sprites in video layers," in *CVPR*, 2001.

[21] Y. Li, M. Dong, and J. Hua, "Simultaneous localized feature selection and model detection for gaussian mixtures," *IEEE TPAMI*, vol. 31, no. 5, pp. 953–960, 2009.

[22] S. Soatto, G. Doretto, and Y. N. Wu, "Dynamic textures," in *ICCV*, 2001.

[23] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto, "Dynamic textures," *IJCV*, 2003.

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

[25] 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.

[26] R. H. Shumway and D. S. Stoffer, "An approach to time series smoothing and forecasting using the EM algorithm," *Journal of Time Series Analysis*, 1982.

[27] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *ICVR*, 2004.

[28] A. B. Chan and N. Vasconcelos, "Variational layered dynamic textures," in *CVPR*. IEEE, 2009.

[29] L. Hubert and P. Arabie, "Comparing partitions," *Journal of classification*, vol. 2, no. 1, pp. 193–218, 1985.