Extracting a Fluid Dynamic Texture and the Background from Video

Bernard Ghanem and Narendra Ahuja
Beckman Institute for Advanced Science and Technology
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
bghanem2,ahuja@vision.ai.uiuc.edu

Abstract

Given the video of a still background occluded by a fluid dynamic texture (FDT), this paper addresses the problem of separating the video sequence into its two constituent layers. One layer corresponds to the video of the unoccluded background, and the other to that of the dynamic texture, as it would appear if viewed against a black background. The model of the dynamic texture is unknown except that it represents fluid flow. We present an approach that uses the image motion information to simultaneously obtain a model of the dynamic texture and separate it from the background which is required to be still. Previous methods have considered occluding layers whose dynamics follows simple motion models (e.g. periodic or 2D parametric motion). FDTs considered in this paper exhibit complex stochastic motion. We consider videos showing an FDT layer (e.g. pummeling smoke or heavy rain) in front of a static background layer (e.g. brick building). We propose a novel method for simultaneously separating these two layers and learning a model for the FDT. Due to the fluid nature of the DT, we are required to learn a model for both the spatial appearance and the temporal variations (due to changes in density) of the FDT, along with a valid estimate of the background. We model the frames of a sequence as being produced by a continuous HMM, characterized by transition probabilities based on the Navier-Stokes equations for fluid dynamics, and by generation probabilities based on the convex matting of the FDT with the background. We learn the FDT appearance, the FDT temporal variations, and the background by maximizing their joint probability using Interactive Conditional Modes (ICM). Since the learned model is generative, it can be used to synthesize new videos with different backgrounds and density variations. Experiments on videos that we compiled demonstrate the performance of our method.

1. Introduction

Separation of a video into its constituent motion layers is a problem that has received significant attention. Typically, it involves separation of a still background from an occluding layer that is formed by moving objects. This problem occurs commonly in real life, usually in the context of a scene of interest (background) being obstructed by the foreground. The difficulty of the problem depends on the nature of the moving objects in each layer (e.g. large and rigid vs. small), their motions (e.g. rigid vs. non-rigid), and their optical characteristics (e.g. opaque vs. translucent). In this paper we, assume that the given video sequence can be represented as a convex sum of the layers. Then, the separation task consists of extracting each individual layer along with both their spatial and temporal supports. In the absence of underlying assumptions regarding the motion or appearance of each layer, this separation problem is ill-defined. We consider the problem of separating a video containing a fluid dynamic texture, FDT, moving in front of a still background, into two distinct layers, as well as learning a spatiotemporal model for the FDT. For example, such a video can consist of heavy smoke or fountain water occluding a building, or fog/clouds blocking a panoramic vacation scene.

Numerous approaches have been proposed for dynamic layer separation from video. They differ according to the motion or appearance models assumed for each layer. In [15,17], dense spatial and temporal correspondences are required, thus, restricting the scope of these methods to relatively simple 2D parametric motions, which are relatively easy to extract. More complex non-rigid motions can be separated using the "information exchange" approach proposed in [11], but this method assumes that one layer obeys 2D parametric motion. The approach in [12] extends the previous work by relaxing the assumption of simple parametric motion to accept periodic motion. The authors use a global space-time alignment method followed by local

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

space-time refinement to align consecutive video frames. After alignment, the periodic motion is extracted using a median filter applied on spatial and temporal image derivatives. Despite the advantages these methods offer, they are unable to separate an FDT from its background for the following reasons: **(1)** FDT motion is non-periodic in general and is characterized by rapid temporal variations, both in density and flow. Consequently, current separation methods will produce significant errors especially in regions where the texture density is high. **(2)** Many of the past methods use frame alignment which is not a well defined task for FDT images.

In general, image-based models of dynamic texture [2, 8, 13] tend to incorporate the background into the model, thus, rendering it too specific to use in general purpose applications (e.g. extrapolating the dynamic texture into novel backgrounds). Also, this drawback hinders the generalization performance of recognition systems that are built based on these models. Therefore, separating the dynamic texture from its background support becomes necessary. Attempts at simultaneously separating dynamic textures from video and modeling include [1, 5]; however, they use specific physical models of the motion and appearance of the dynamic texture (e.g., rain and snow). These models therefore do not serve as general mechanisms for FDT's. To motivate our problem, we consider a sample smoke sequence, from which we seek to extract both the FDT and the building in the background. Figure 1 shows results of separating the FDT layer from the static background layer, using **(i)** a temporal median filter (1(d)), which fails to capture the background, due to the smoke's temporal persistence, **(ii)** the dynamic layer separation method of [11, 12], which fails to separate the layers especially in regions of high smoke density, and **(iii)** the algorithm we propose in this paper.

Contributions: In this paper, we present a novel approach to model and separate FDT's from video sequences, which addresses the aforementioned limitations of previous methods. The contributions of the proposed method are twofold. **(I)** It simultaneously separates an FDT from its static background and learns a general fluid model of the FDT's appearance and dynamics. For simplicity, we assume that the FDT's appearance is temporally stationary and that the temporal variations of its density are governed by basic laws of fluid dynamics. In fact, we model the frames of a video sequence as being produced by an HMM, characterized by transition probabilities based on the Navier-Stokes equations for fluid dynamics and by generation probabilities based on convex matting of the FDT with the background. Both the FDT and background layers are estimated by formulating and maximizing an appropriate joint probability using Iterative Conditional Modes (ICM). **(II)** Due to the generative nature of our DT model and the separability of its underlying components (i.e. the FDT appearance, FDT dynamics, and background), higher level applications can be performed (e.g. synthesis and recognition). This separability property allows us to avoid the shortcomings of current DT models that couple the aforementioned components. In this paper, we use this model to synthesize novel FDT sequences.

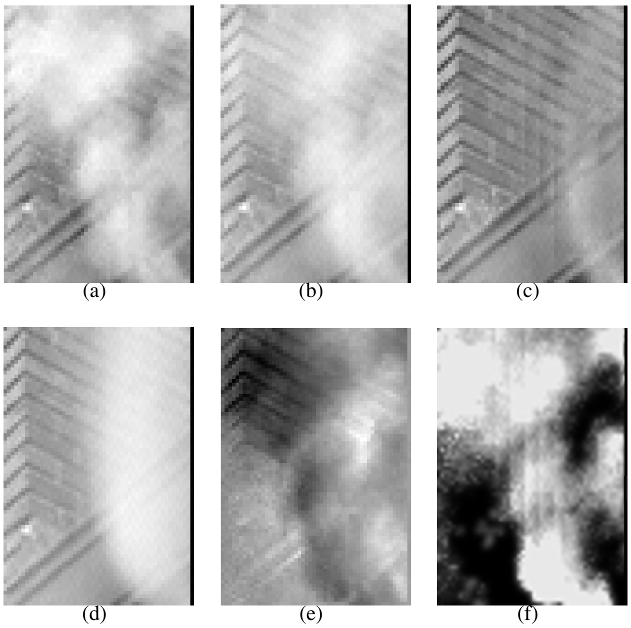


Figure 1. 1(a) is an original frame from a smoke sequence and 1(d) is the median image of this sequence. 1(b) and 1(e) show the results of layer separation (extracted background and smoke, respectively) using the information exchange method of [11, 12], while 1(c) and 1(f) show the corresponding separation results produced by our algorithm.

The paper is organized as follows. Section 2 describes the FDT dynamics and appearance models used in formulating the probabilistic framework of Section 3. In this section, we establish a joint probability of the variables to be estimated and explain how to maximize it using ICM. We derive iterative update equations and describe how the initialization is performed. Section 4 presents experimental results of applying the proposed method to real and synthetic video sequences.

2. Problem Formulation and Overview

Given a video sequence $\{\mathcal{I}_t\}_{t=1}^{F}$, which includes an FDT moving in front of a static background $\mathcal{B}$, we assume that each frame is a convex combination of the FDT and $\mathcal{B}$. In other words, for an individual frame of $M \times N$ pixels, we have $\forall x = 1, \cdots, M,\ y = 1, \cdots, N$

$$\begin{cases} \mathcal{I}_t\left(x,y\right) = \rho_t\left(x,y\right)\mathcal{D}\left(x,y\right) + \left[1-\rho_t\left(x,y\right)\right]\mathcal{B}\left(x,y\right) \\ \rho_t\left(x,y\right) \in [0,1],\ \mathcal{I}_t\left(x,y\right) \leq \max\left(\mathcal{B}\left(x,y\right),\mathcal{D}\left(x,y\right)\right) \end{cases} \tag{1}$$

where $\rho_t\left(x,y\right)$ is the density (α matte) of the FDT at spatiotemporal location (x,y,t) and $\mathcal{D}\left(x,y\right)$ is its appearance. The first term in the sum designates the contribution of FDT to $\mathcal{I}_t$, while the second term designates that of the background.

This model makes two fundamental assumptions. **(1)** The textured appearance of the FDT does not change with time (i.e. $\mathcal{D}$ is static). This implies that the appearance of an FDT at a specific time is the result of applying a soft mask (ρ_t) to a static 2D texture. This is valid for the vast majority of dynamic textures that behave as fluids (e.g. fog, smoke, water, etc.). For example, in the case of fog, $\mathcal{D}$ is approximately constant. In the following sections, we also see that this assumption can facilitate the estimation of the FDT model. **(2)** The model defining an FDT's dynamics can be made independent of its appearance model. Equivalently, $\{\rho_t\}_{t=1}^{F}$ and $\mathcal{D}$ are statistically independent. This assumption is valid and widely used in computer graphics to render FDT's (e.g. smoke) with different texture maps or colors. Based on this claim, we can learn $\{\rho_t\}_{t=1}^{F}$ and $\mathcal{D}$ from one sequence and synthesize a novel FDT sequence with the same temporal variations but with an appearance $\mathcal{D}^{'} \neq \mathcal{D}$, or vice versa.

We model the temporal variations of an FDT by the Navier-Stokes differential equations for fluid dynamics. In their general form, they govern the change of a fluid's density and flow over time. In fact, they have been used extensively in synthesizing images of general stable fluids in computer graphics [3, 4, 14], since they have been shown to effectively model both the spatial and temporal coherence of the FDT. In Eq (2), we present the vector form of these equations.

$$\begin{cases} (E_1): & \frac{\partial \rho}{\partial t} = -\left(\vec{u}.\nabla\right)\rho + \kappa\nabla^2\rho + \mathcal{S} \\ (E_2): & \frac{\partial \vec{u}}{\partial t} = -\left(\vec{u}.\nabla\right)\vec{u} + \nu\nabla^2\vec{u} + \vec{\mathcal{F}} \end{cases} \tag{2}$$

where ρ, $\vec{u}$, κ and ν are the density, flow, diffusion rate and viscosity of the fluid respectively. $\mathcal{S}$ represents the auxiliary sources of fluid density, while $\vec{\mathcal{F}}$ represents the external forces applied to the fluid. In developing the proposed model, we assume $S = 0$ and $\vec{\mathcal{F}} = \vec{0}$, thus rendering a source-free uncompressed fluid, with $0 \leq \rho \leq 1$.

We discretize these differential equations as shown in Eq (3). Time derivatives are replaced by differences and spatial derivatives are replaced by suitable derivative filters to render the transition arrays B_T, S_t^x, S_t^y, $\vec{s}_t^{\,x}$, and $\vec{s}_t^{\,y}$. We used $\vec{h}_x$ and $\vec{h}_y$ as the first order derivative filters in the x and y directions respectively. Similarly, we define $H_{xx} = H_{yy}^T$ as the second order derivatives. We choose these basic filters so that the transition matrices (B_t, S_t^x, and S_t^y) are sparse. Note that B_t is a function of $\vec{u}_t$ and κ_t, while S_t^x, and S_t^y are functions of ν. Also, denote the vectorized version of $\rho_t\left(x,y\right)$ by $\vec{\rho}_t$ and that of $\vec{u}_t\left(x,y\right)$ by $\vec{u}_t$, where the x components are stacked on top of the y components.

$$\begin{cases} (E_1): \vec{\rho}_{t+1} = B_t\vec{\rho}_t \\ (E_2): \vec{u}_{t+1}\left(x,y\right) = \begin{bmatrix} \vec{u}_t^T S_t^x \vec{u}_t + \vec{u}_t^T \vec{s}_t^{\,x} \\ \vec{u}_t^T S_t^y \vec{u}_t + \vec{u}_t^T \vec{s}_t^{\,y} \end{bmatrix} \\ \vec{h}_x = \vec{h}_y^T = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix};\ H_{xx} = \begin{bmatrix} 1 & -2 & 1 \\ 1 & -2 & 1 \\ 1 & -2 & 1 \end{bmatrix} \end{cases} \tag{3}$$

Below, we propose a method to separate the background from the FDT (i.e. compute $\{\vec{\rho}_t\}_{t=1}^{F}$), as well as estimate the FDT appearance ($\mathcal{D}$) and the background ($\mathcal{B}$). Eqs (1) and (3) are the premises for building the joint probability model in Section 3.

3. Proposed Model

In this section, we embed the model of Section 2 into a probabilistic framework to incorporate the noise that might arise. In Figure 2, we illustrate the graphical representation of this framework. It is in the form of a continuous hidden Markov model (HMM), where **(1)** the generation probability conforms to the convex matting constraint of Eq (1) and **(2)** the transition probability is based on the discretized Navier-Stokes equations of Eq (3). For now, the hidden state is $x_t = \{\vec{\rho}_t, \vec{u}_t\}$.

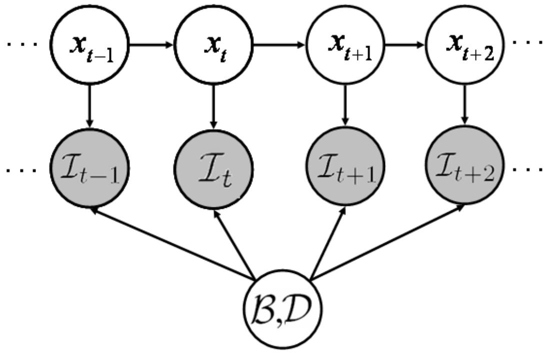


Figure 2. Graphical representation of our proposed model. The hidden state x_t encodes the FDT's temporal variations. Given the current state x_t, $\mathcal{B}$, and $\mathcal{D}$, the current video frame $\mathcal{I}_t$ is independent of all other frames.

Using the Markovian property and the independence of $\{\rho_t\}_{t=1}^{F}$ and $\mathcal{D}$, we formulate the joint probability of this model as in Eq (4). Let $\vec{i}_t$, $\vec{b}$, and $\vec{d}$ be the vectorized versions of $\mathcal{I}_t$, $\mathcal{B}$, and $\mathcal{D}$ respectively. Here, $K =$

$P(\mathcal{B})P(\mathcal{D})P(x_1)$ due to the joint independence of $\mathcal{B}$ and $\mathcal{D}$. Assuming uniform priors on $\mathcal{B}$, $\mathcal{D}$, and x_1, K is a constant with respect to these variables.

$$P\left(\{I_t\}_{t=1}^F, \{x_t\}_{t=1}^F, \mathcal{B}, \mathcal{D}\right) =$$
$$K\left[\prod_{t=1}^{F-1} P\left(\vec{i}_t|\vec{\rho}_t, \vec{b}, \vec{d}\right) P\left(\vec{\rho}_{t+1}|\vec{\rho}_t, \vec{u}_t\right) P\left(\vec{u}_{t+1}|\vec{u}_t\right)\right] \quad (4)$$

where $P\left(\vec{u}_{t+1}|\vec{u}_t\right) = \prod_{(x,y)} P\left(\vec{u}_{t+1}(x,y)|\vec{u}_t\right)$. We model the generation probability and the transition probabilities as follows, where $\Lambda_t = diag\left(\vec{\rho}_t\right)$ and $A = diag\left(\vec{d} - \vec{b}\right)$.

$$P\left(\vec{i}_t|\vec{\rho}_t, \mathcal{B}, \mathcal{D}\right) \sim \left\{\begin{array}{c} \mathcal{N}\left(\Lambda_t\vec{d} + \left(I - \Lambda_t\right)\vec{b}, \sigma_{\mathcal{I}}^2 I\right) \\ \Updownarrow \\ \mathcal{N}\left(A\vec{\rho}_t + \vec{b}, \sigma_{\mathcal{I}}^2 I\right) \end{array}\right.$$
$$P\left(\vec{\rho}_{t+1}|\vec{\rho}_t, \vec{u}_t, \kappa_t\right) \sim \mathcal{N}\left(B_t\vec{\rho}_t, \sigma_\rho^2 I\right)$$
$$P\left(\vec{u}_{t+1}(x,y)|\vec{u}_t, \nu\right) \sim \mathcal{N}\left(\left[\begin{array}{c} \vec{u}_t^T S_t^x \vec{u}_t + \vec{u}_t^T \vec{s}_t^x \\ \vec{u}_t^T S_t^y \vec{u}_t + \vec{u}_t^T \vec{s}_t^y \end{array}\right], \sigma_u^2 I\right)$$

We propose to estimate $\{x_t\}_{t=1}^F$, $\mathcal{B}$, $\mathcal{D}$, and the model parameters (κ_t, ν, $\sigma_{\mathcal{I}}^2$, and σ_ρ^2) by maximizing the joint probability in Eq (4) using coordinate ascent along these variables (i.e. ICM). As we will see, this leads to a set of update equations formulated as convex quadratic programming (QP) problems. In what follows, we decouple the state variables by keeping the flow variables $\{\vec{u}_t\}_{t=1}^F$ equal to their initial estimates (i.e. these variables are not updated). This allows for numerical stability, reduces computational complexity, and removes the nonlinearity in the model.

3.1. Maximizing Eq (4)

In Eq (4), the background ($\vec{b}$), FDT appearance ($\vec{d}$), FDT densities ($\{\vec{\rho}_t\}_{t=1}^F$), FDT diffusion rate (κ_t), and the model parameters ($\sigma_{\mathcal{I}}$ and σ_ρ) are coupled, so maximizing the joint probability renders a large-scale, non-linear, and non-convex optimization problem. Consequently, we resort to solving this problem suboptimally, using ICM, which will lead to a local maximum in general. In each ICM iteration, the estimate of an individual variable is computed by maximizing the joint probability with all other variables fixed. In Section 3.2, we describe how we initialize these variables. Here, we note that we refrained from using a complete EM formulation because it adds considerable computational expense with limited performance improvement.

Let $\vec{b}^{(k)}$, $\vec{d}^{(k)}$, $\left\{\vec{\rho}_t^{(k)}\right\}_{t=1}^F$, $\kappa_t^{(k)}$, $\sigma_{\mathcal{I}}^{(k)}$, and $\sigma_\rho^{(k)}$ be the estimates at the k^{th} iteration. By minimizing the negative logarithm of the joint probability, we can derive the following update equations.

Update Model Parameters and κ_t: Setting $\alpha_t^{(k)}(x,y) = \rho_{t+1}^{(k)}(x,y) - \rho_t^{(k)}(x,y) + \vec{u}_t(x,y)^T\nabla\rho_t^{(k)}(x,y)$ and $\beta_t^{(k)}(x,y) = \nabla^2\rho_t^{(k)}(x,y)$, we compute the ML estimates of $\sigma_{\mathcal{I}}^{(k+1)}$, $\sigma_\rho^{(k+1)}$, and κ_t as follows.

$$\sigma_{\mathcal{I}}^{(k+1)} = \sqrt{\frac{1}{F}\sum_{t=1}^{F} ||A^{(k)}\vec{\rho}_t^{(k)} + \vec{b}^{(k)} - \vec{i}_t||^2} \quad (5)$$

$$\sigma_\rho^{(k+1)} = \sqrt{\frac{1}{F}\sum_{t=1}^{F-1} ||\vec{\rho}_{t+1}^{(k)} - B_t\vec{\rho}_t^{(k)}||^2} \quad (6)$$

$$\kappa_t^{(k+1)} = \frac{\vec{\alpha}_t^T\vec{\beta}_t}{||\vec{\beta}_t||^2} \quad (7)$$

Update Background: We update the background by solving the QP below, where $l_1(j) = \min_{1\leq t\leq F} i_t(j)$, $L_2^{(k)} = diag\left(\vec{l}_2^{(k)}\right) = \sum_{t=1}^F\left(I - \Lambda_t^{(k)}\right)^2$, and $\vec{l}_3^{(k)} = \sum_{t=1}^F\left[\left(I - \Lambda_t^{(k)}\right)\left(\Lambda_t^{(k)}\vec{d}^{(k)} - \vec{i}_t\right)\right]$.

$$\vec{b}^{(k+1)} = \underset{\vec{0}\leq\vec{z}\leq\vec{l}_1}{\arg\min}\left[\vec{z}^T L_2^{(k)}\vec{z} + 2\vec{z}^T\vec{l}_3^{(k)}\right] \quad (8)$$

Since $L_2^{(k)}$ is diagonal, the above QP is equivalent to MN scalar QP's, which can be solved in closed form: $b^{(k+1)}(j) = \min\left(\max\left(-\frac{l_3^{(k)}(j)}{l_2^{(k)}(j)}, 0\right), l_1(j)\right)$.

Update FDT Appearance: A similar QP is formulated for $\vec{d}^{(k+1)}$, where $L_2^{(k)} = diag\left(\vec{l}_2^{(k)}\right) = \sum_{t=1}^F\left(\Lambda_t^{(k)}\right)^2$, and $\vec{l}_3^{(k)} = \sum_{t=1}^F\left[\Lambda_t^{(k)}\left(\left(I - \Lambda_t^{(k)}\right)\vec{b}^{(k)} - \vec{i}_t\right)\right]$. Consequently, the closed form solution is $d^{(k+1)}(j) = \min\left(\max\left(-\frac{l_3^{(k)}(j)}{l_2^{(k)}(j)}, 0\right), l_1(j)\right)$.

Update FDT Densities: Here, we update $\vec{\rho}_t^{(k)}$ sequentially $\forall t = 1, \cdots, F$. Let $\alpha_1 = I_{\{t\neq 1\}}$ and $\alpha_2 = I_{\{t\neq F\}}$ be the indicators for the first and last frames. The update equation for the FDT density at a particular time t becomes

$$\vec{\rho}_t^{(k+1)} = \underset{\vec{0}\leq\vec{z}\leq\vec{1}}{\arg\min}\ \frac{1}{2\sigma_{\mathcal{I}}^2}||A^{(k)}\vec{z} + \vec{b}^{(k)} - \vec{i}_t||^2$$
$$+ \frac{1}{2\sigma_\rho^2}\left(\alpha_2||\vec{\rho}_{t+1}^{(k)} - B_t^{(k)}\vec{z}||^2 + \alpha_1||\vec{z} - B_{t-1}^{(k)}\vec{\rho}_{t-1}^{(k)}||^2\right)$$

which is equivalent to Eq (9). Here, we define $L_2^{(k)} = \frac{\left(A^{(k)}\right)^2}{\sigma_{\mathcal{I}}^{(k)2}} + \frac{\alpha_2 B_t^{(k)T}B_t^{(k)} + \alpha_1 I}{\sigma_\rho^{(k)2}}$ and $\vec{l}_3^{(k)} = \frac{A^{(k)}\left(\vec{b}^{(k)} - \vec{i}_t\right)}{\sigma_{\mathcal{I}}^{(k)2}} -$

$\frac{\alpha_2 B_t^{(k)} \vec{\rho}_{t+1}^{(k)} + \alpha_1 B_{t-1}^{(k)} \vec{\rho}_{t-1}^{(k)}}{\sigma_\rho^{(k)2}}$. Since $L_2^{(k)}$ is sparse and $L_2^{(k)} \succeq 0$, the resulting problem is convex and can be solved efficiently via suitable QP solvers (e.g. active set or interior point methods). In our implementation, we use a basic active set method, whose fundamental step requires the solution of a sparse linear system using preconditioned conjugate gradients. To speed up the update process, we initialize the solution of the linear system corresponding to frame t with the final solution of frame $t-1$.

$$\vec{\rho}_t^{(k+1)} = \underset{\vec{0} \leq \vec{z} \leq \vec{1}}{\arg\min} \left[\vec{z}^T L_2^{(k)} \vec{z} + 2\vec{z}^T \vec{l}_3^{(k)} \right] \tag{9}$$

3.2. Initialization

Since the original optimization problem is non-linear and non-convex, the solution we obtain from the previous update scheme will be a local minimum, in general. Therefore, initializing the variables with meaningful values is crucial. We propose to use spectral matting [7] to initialize the FDT densities and a phase-based optical flow method [6] to determine the FDT flow vectors. Furthermore, we assume that during the entire sequence the background solely appears (i.e. with zero FDT density) at least once at every pixel. This is a valid assumption because, in the absence of a background appearance model, no feasible estimate exists for the true intensity of a background pixel, which does not solely appear in a sequence at least once.

3.2.1 Determine $\{\vec{u}_t\}_{t=1}^F$

To initialize the fluid flow, we can choose any optical flow algorithm from the large set of generic algorithms or those specific to fluid flow [9, 18]. In [6], a phase-based algorithm is proposed to estimate the optical flow fields of an image sequence by tracking contours of constant phase over time. In this paper, we use this method to estimate the flow vectors between every pair of consecutive frames in the sequence.

3.2.2 Determine $\left\{\vec{\rho}_t^{(0)}\right\}_{t=1}^F$

To compute the initial FDT densities, we estimate the α matte for every individual frame in the sequence, using spectral matting defined in [7]. This method is chosen, since it is unsupervised and is proven to be optimal under certain conditions. To make the paper self-contained, we briefly review this method and then modify it for our purposes. For each frame $\mathcal{I}_t$, the above method decomposes it into a convex combination of K layers: $\mathcal{I}_t = \sum_{i=1}^K \alpha_i \mathcal{F}_i$. Each matte $\vec{\alpha}_i$ is a linear combination of the eigenvectors of the Laplacian matrix ($\mathcal{L}_t$) corresponding to the graph formed from the pixels of $\mathcal{I}_t$ (i.e. $\vec{\alpha}_i = E(\mathcal{L}_t)\,\vec{y}_i$). The authors define a pairwise cost between two α mattes as: $w_t(i,j) = \vec{\alpha}_i^T \mathcal{L}_t \vec{\alpha}_j$. The foreground matte is determined as the sum of a subset of the extracted α mattes ($\{\vec{\alpha}_i\}_{i=1}^K$) that minimizes $C = \sum_{i,j\in\mathcal{S}} w_t(i,j)$ with a lower bound on the percentage of pixels labeled as background and foreground. The authors perform an exhaustive search over all 2^K subsets to find $\mathcal{S}$. The computational complexity of this search grows exponentially in the number of layers, so we propose to form an equivalent min-cut problem, which can be solved sub-optimally yet efficiently. The equivalent problem is formulated in Eq (10). We have relaxed the binary constraint on $\vec{z}$ to take on real values with $||\vec{z}||^2 = K$. a_1 and a_2 are defined as the minimum and maximum percentages of allowable foreground pixels. We use a simple gradient descent method to find a local minimum of this QP, which is then discretized using kmeans ($k=2$) on $\vec{z}_t^*$. The discretized $\vec{z}_t^*$ selects the set of α mattes that form the foreground matte, which in turn determines $\vec{\rho}_t^{(0)}$. From the initial density estimates, we compute $\kappa^{(0)}$ and $\sigma_\rho^{(0)}$ using Eqs (5,6,7).

$$\begin{aligned} \vec{z}_t^* &= \arg\min \left[\vec{z}^T W_t \vec{z} + 2\vec{z}^T W_t \vec{1} \right] \\ &\text{s.t.} \begin{cases} ||\vec{z}||^2 = K \\ (2a_1 - 1)MN \leq \vec{1}^T \vec{z} \leq (2a_2 - 1)MN \end{cases} \end{aligned} \tag{10}$$

3.2.3 Determine $\vec{b}^{(0)}$

After computing $\left\{\vec{\rho}_t^{(0)}\right\}_{t=1}^F$, we determine the initial background estimate at each pixel as the image intensity at that pixel corresponding to the frame that contains the minimum density of that pixel over time (i.e. $b^{(0)}(j) = i_{t_0}(j)$, where $t_0 = \arg\min_t \rho_t^{(0)}(j)$). From this initial estimate, we compute $\vec{d}^{(0)}$ using the FDT appearance update equation. Then, we initialize $\sigma_{\mathcal{I}}$, using Eq (5).

4. Experimental Results

We conducted a set of experiments to verify the correctness of our formulation and the proposed algorithm. We collected FDT sequences from various sources: the MIT temporal texture dataset [16], the Dyntex dataset [10], and online sources. All these video sequences include an FDT moving infront of a static background. They show significant variations in the nature of the FDT (e.g. sparse fountain water and thick exhaust smoke) and the complexity of the background (e.g. highly textured or constant intensity regions). In order to compute the initial α matte of each frame within 60 seconds, we crop out the spatial support of the FDT from the initial video and resize it (if necessary) to a maximum size of 120×120 pixels. We use $K = 30$

layers, $a_1 = 0.3$, and $a_2 = 0.7$ to initialize the FDT densities. The experiments were executed using MATLAB on a 2.8 GHz, 2GB RAM PC. We allowed 2-3 ICM iterations for each video sequence, where the execution time of each iteration ranged from 15-60 seconds per frame. The majority of this time is taken by updating the FDT density of a given frame. In Section 4.1, we show an example of how the estimated variables evolve with the ICM iterations. In Section 4.2, we give a quantitative comparison between the information exchange method of [11,12] and our own, when applied to a synthetic sequence. Finally, Section 4.3 illustrates how our generative model can be used to synthesize novel FDT sequences with varying appearance and/or background.

4.1. ICM Iterations

During each ICM iteration, we update each variable to the value that maximizes the joint probability with all other variables fixed. This leads to incremental improvement in the estimation of each variable. In Figure 3, we show the results of applying our algorithm to two smoke sequences: smoke$_2$ ([10]) and smoke$_{MIT}$ ([16]). The first three and last three images of the first row smoke$_2$ and smoke$_{MIT}$ respectively. In the second row, we show the initial background (columns (a) and (c)), initial FDT appearance (columns (b) and (e)), and initial FDT density (columns (c) and (f)) estimates, while the third row shows their corresponding final estimates. The appearances of the two FDT's are of high intensity and spatially close to constant, except in regions where the FDT only appears in a few frames (i.e. regions of low temporal persistence). Similarly, the background estimates incur errors at regions where the FDT density is high and temporally persistent. Since the FDT model enforces spatial and temporal coherence, fitting it to the observed frames significantly improves the initial estimates of all the variables. In fact, this coherence is crucial for individual FDT densities, since some initial estimates include considerable errors. This is the case for the smoke$_{\mathrm{MIT}}$ sequence, where even though the initial density estimate is inverted (column (f) of 2nd row), our algorithm rectifies it (column (f) of 3rd row). For visual comparison, we supply the median images of the two sequences in Figures 3(g) and (h), respectively. Clearly, the median filter fails to capture the background correctly. In Figures 3(i) and (j), we plot the estimated diffusion rates (κ_t) of each sequence. These values are positive and show minor temporal variations. This indicates a nearly constant outward diffusion from the fluid source, which correctly supports the underlying dynamics of each sequence.

4.2. Comparative Analysis

In this section, we perform a quantitative comparison between our proposed method and the information exchange separation method used in [11, 12]. Given two images formed from linear combinations of two layers, the information exchange method attempts to extract these two layers by iteratively minimizing the structural correlation between the two original images. So, to implement this method on an FDT sequence, we first extract its median image and then sequentially apply the method on each frame and the median image, as done in [11]. We applied both the information exchange method and our own to a synthetic FDT sequence (2nd row of Figure 4), whose ground truth FDT densities and background are known beforehand. These densities were acquired from a chimney smoke sequence (1st row of Figure 4), where the FDT moves in front of a black background and has a constant spatial appearance. The synthetic sequence contains 250 frames, each of which is 68×128 pixels in size. We executed the information exchange method using a 5×5 pixel window and a maximum of 30 iterations. As recommended by [11], we performed a coarse to fine analysis to incorporate the spatially varying FDT densities. For our algorithm, we used 2 ICM iterations. Figure 4(a) plots the ℓ_2 norm of the absolute difference between the estimated and ground truth densities, while Figure 4(b) plots their corresponding normalized correlations. It is clear that our algorithm outperforms the other method. Here, we mention that the information exchange method produced negative pixel intensities, which were suppressed in our experiment. Also, this method does not enforce temporal coherence between layers over time, so, for example, the first layer at time t may not correspond to the first layer at $t + 1$. Hence, to choose the estimated density layer from the two possible layers, we select the one closest to the ground truth density at that frame.

4.3. FDT Synthesis

Our generative model enforces statistical independence between the three components of the video sequence (the FDT appearance, FDT dynamics, and the background). So, after learning each of these components, we can readily synthesize novel FDT sequences, by varying each component separately. For example, in Figure 5, we show sample frames produced by transferring the FDT densities learned from three original video sequences (1st, 3rd, and 5th rows) to a new background and/or appearance.

5. Conclusion and Future Work

In this paper, we have presented a novel method to simultaneously separate a fluid dynamic texture (FDT) from its static background and learn a generative model of the texture's spatiotemporal characteristics. The proposed model combines the FDT's temporal density variations and spatial appearance with the static background model in a continuous HMM framework. We learn these variables by maxi-

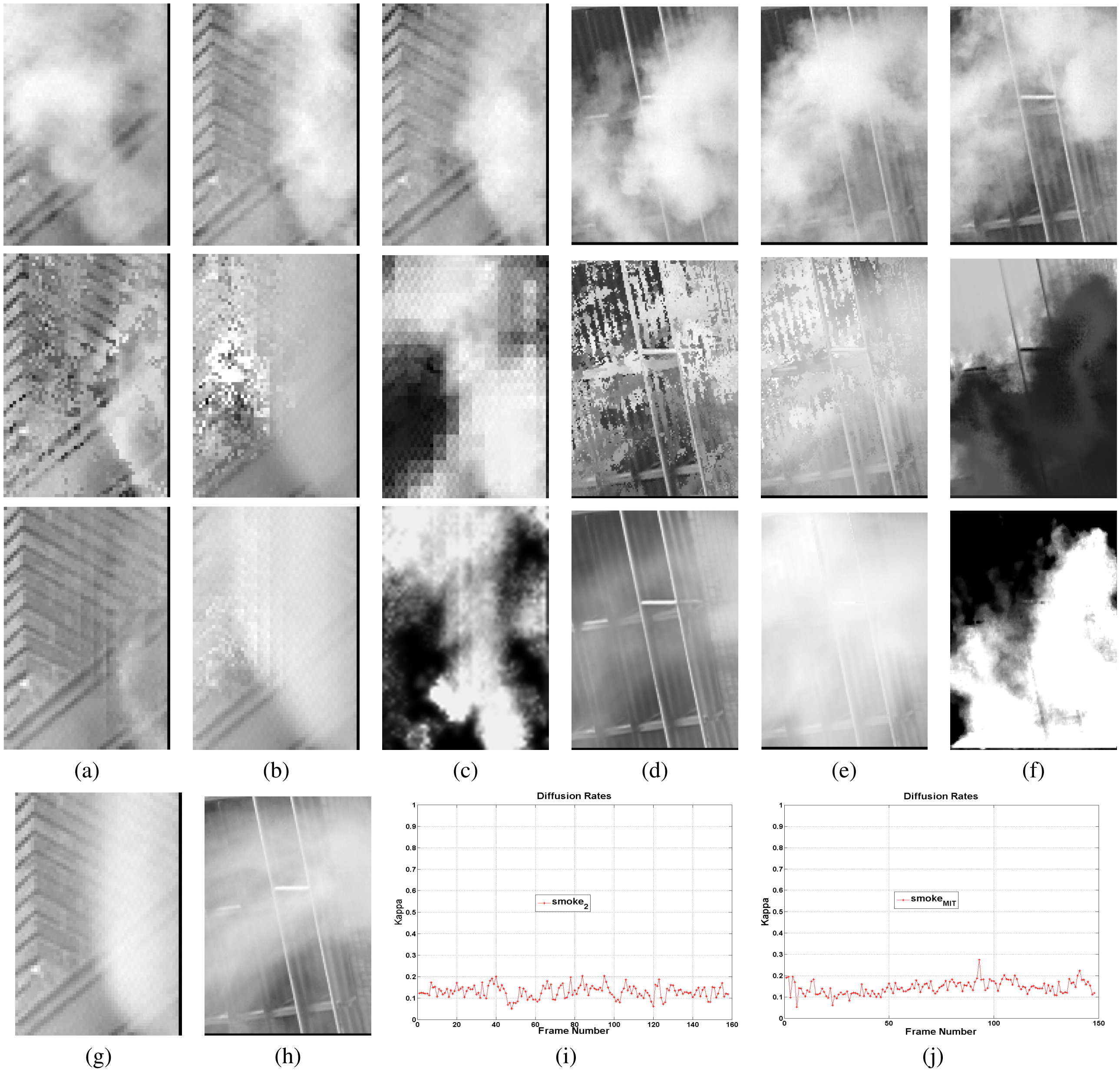
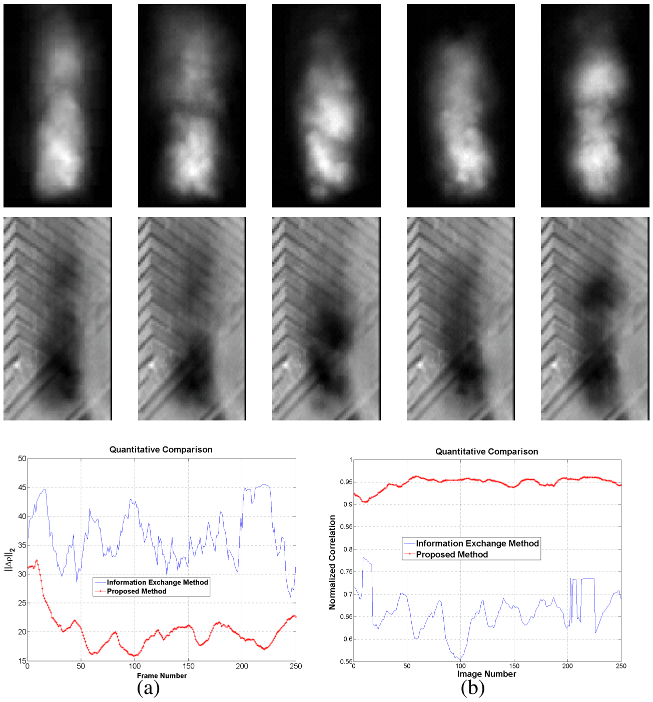


Figure 3. The first row shows frames from the original smoke$_2$ (first three) and smoke$_2$ (last three) FDT sequences. The second row shows the initial estimates of $\mathcal{B}$ in columns (a) and (d), $\mathcal{D}$ in (b) and (e), and the density ($\vec{\rho}_t$) of a sample frame for both sequences respectively. This density corresponds to the first frame of each sequence, shown in the first row. The third row shows the final estimates of the previous variables in the same order. Figures (g) and (h) show the median images of both sequences. Figures (i) and (j) plot the estimated diffusion rates respectively.

mizing their joint probability using ICM. We validate our method by applying it to real and synthetic sequences, as well as, comparing it to a current dynamic layer separation method. Furthermore, we exploited the generative nature of the learned model to produce synthetic FDT sequences. In the future, we plan to extend this work to learn the FDT local flow model ($\{\vec{u}_t\}_{t=1}^{F}$), which was held constant here. This will make non-repetitive extrapolation of synthetic sequences possible. Furthermore, we aim at extending the current work to non-static background models and using it to build an FDT recognition system.

6. Acknowledgement

This research was supported by the Office of Naval Research under grant: N00014-06-1-0101.



(a) (b)

Figure 4. Compares our algorithm to the information exchange separation method of [11], when applied to the synthetic sequence (2^{nd} row). 4(a) plots the ℓ_2 norm of the error between the ground truth FDT densities (1^{st} row) and those estimated by each method. In 4(b), we plot the normalized correlation between these values.

References

[1] P. Barnum, T. Kanade, and S. G. Narasimhan. Spatio-temporal frequency analysis for removing rain and snow from videos. In *Proc. of ICCV*, 2007.

[2] D. Chetverikov and R. Peteri. A brief survey of dynamic texture description and recognition. In *Proc. of the International Conference on Computer Recognition Systems*, 2005.

[3] R. Fedwik, J. Stam, and H. W. Jensen. Visual simulation of smoke. In *Proc. of the International Conference on Computer Graphics and Interactive Techniques*, 2001.

[4] N. Foster and D. Metaxas. Controlling fluid animation. In *Proc. of Computer Graphics International*, pages 178–188, 1997.

[5] K. Garg and S. K. Nayar. Detection and removal of rain from videos. In *Proc. of CVPR*, 2004.

[6] T. Gautama and M. A. V. Hulle. A phase-based approach to the estimation of the optical flow field using spatial filtering. *IEEE Transactions on Neural Networks*, 13(5):1127–1136, 2002.

[7] A. Levin, A. Rav-Acha, and D. Lischinski. Spectral matting. In *Proc. of CVPR*, 2007.

[8] R. Li, T.-P. Tian, and S. Sclaroff. Simultaneous learning of nonlinear manifold and dynamical models for high-dimensional time series. In *Proc. of ICCV*, 2007.

[9] E. Memin and P. Perez. Fluid motion recovery by coupling dense and parametric vectorfields. In *Proc. of ICCV*, 1999.

[10] R. Peteri, M. Huiskes, and S. Fazekas. Dyntex: www.cwi.nl/projects/dyntex/ at the Centre for Mathematics and Computer Science (CWI), Amsterdam, The Netherlands. 2006.

[11] B. Sarel and M. Irani. Separating transparent layers through layer information excxhange. In *Proc. of ECCV*, 2004.

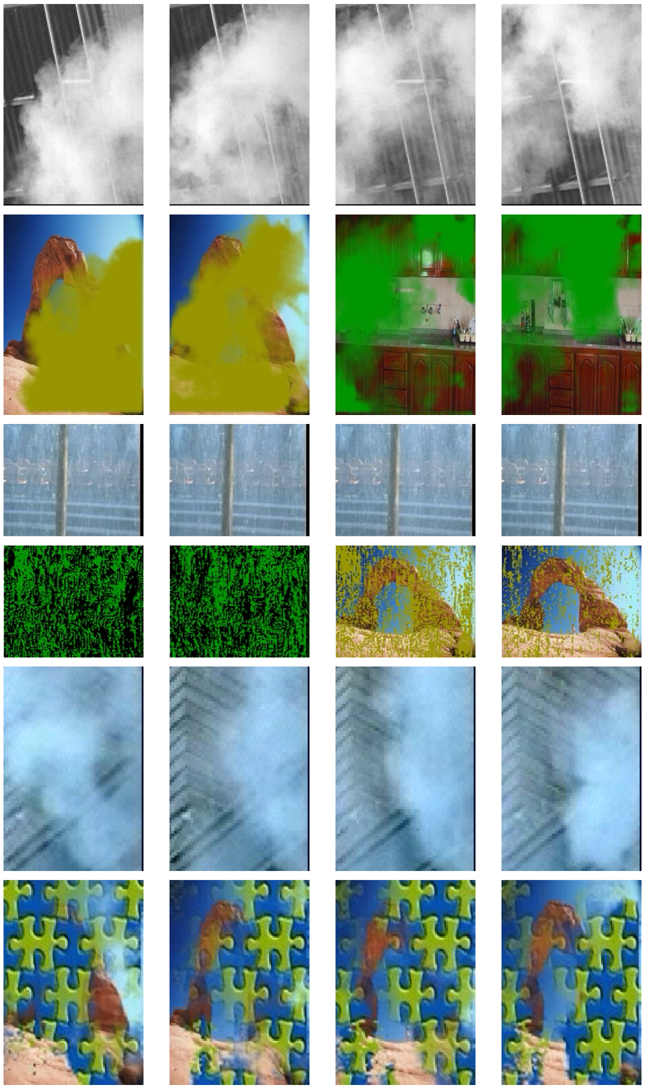


Figure 5. Frames from synthetic FDT sequences (refer to http://vision.ai.uiuc.edu/∼bghanem2/Shared/FDT Results/)

[12] B. Sarel and M. Irani. Separating transparent layers of repetitive dynamic behaviors. In *Proc. of ICCV*, 2005.

[13] S. Soatto, G. Doretto, and Y. N. Wu. Dynamic textures. *IJCV*, 51:91–109, 2003.

[14] J. Stam. Real-time fluid dynamics for games. In *Proc. of the Game Developer Conference*, 2003.

[15] R. Szeliski, S. Avidan, and P. Anandan. Layer extraction from multiple images containing reflections and transparency. In *Proc. of CVPR*, 2000.

[16] M. Szummer and R. W. Picard. Temporal texture modeling. In *Proc. of ICIP*, volume 3, pages 823–826, Sept. 1996.

[17] Y. Wexler, A. Fitzgibbon, and A. Zisserman. Bayesian estimation of layers from multiple images. In *Proc. of the ECCV*, 2002.

[18] R. P. Wildes, M. J. Amabile, A.-M. Lanzillotto, and T.-S. Leu. Recovering estimates of fluid flow from image sequence data. *Computer Vision and Image Understanding*, 80(2):246–266, 2000.