The Dependent Dirichlet Process Mixture of Objects for Detection-free Tracking and Object Modeling

Abstract

This paper explores the question of how to find, track, and learn models of arbitrary objects in a video without a predefined method for object detection. We present a model that localizes objects via unsupervised tracking while learning a representation of each object, avoiding the need for pre-built detectors. Our model uses a dependent Dirichlet process mixture to capture the uncertainty in number of objects and requires only spatial and color video data that can be efficiently extracted via frame differencing. We give two algorithms for performing inference and demonstrate our method on multiple videos. Results illustrate its ability to accurately find, track, and model diverse objects moving over non-uniform backgrounds and through occlusion. We demonstrate the model in difficult detection scenarios, on a video containing a large number of objects, and on a recent human-tracking benchmark where we show performance comparable to state of the art detector-based methods.

1. Introduction

Algorithms for automated object detection and tracking in video have found application in a wide range of fields, including robotic vision, cell tracking, sports analysis, video indexing, and video surveillance (Turaga et al., 2008; Yilmaz et al., 2006). The goal of these algorithms is to find the sequences of positions held by each object of interest in a video. A majority of modern methods require a pre-trained object detector or make use of prior knowledge about the objects' physical characteristics (such as their color or shape) to perform detection (Breitenstein et al., 2009b). Often, these methods will apply the detector in each frame of a video, and then use the detection results in tracking or data association algorithms. Other algorithms use heuristics to find, or require manual initialization of, object positions and then search for similar image patches in consecutive frames to perform tracking (Meer, 2003). Both techniques require some predefined detection strategy for each type of object they intend to find and track.

When the objects to be tracked have highly variable appearance, if one wishes to track many different types of objects, or if one simply does not know the types of objects in advance, it is often hard to find a suitable detection strategy (Babenko et al., 2011). Furthermore, common video conditions such as variable lighting, low quality images, non-uniform backgrounds, and object occlusions can all reduce the accuracy of detection (Wu & Nevatia, 2007).

Cases such as these, where constructing an object detector may be infeasible, prompt the need for a method to automatically localize and track arbitrary objects. Some classic methods towards this end have involved background subtraction and blob tracking, which segment foreground patches to localize objects, and optical flow-based tracking, which separate objects based on their relative motion. Both have trouble consistently and accurately segmenting objects and tracking through occlusion (Veeraraghavan et al., 2006; Barron et al., 1994). A recent work introduced the term "detection-free tracking" for this task, and proposed a method based on spectral clustering of trajectories (Fragkiadaki & Shi, 2011).

Bayesian models have also been employed to capture the components of a video, and a number of recent works have incorporated nonparametric Bayesian priors for finding the patterns of motion in scenes (Wang et al., 2007; Emonet et al., 2011). However, there has been little towards building Bayesian models that model arbitrary video objects in order to perform high quality detection-free tracking.

In this paper, we develop a nonparametric Bayesian model for jointly learning a representation of each object and performing unsupervised tracking, thereby allowing for accurate localization of arbitrary objects. We combine a dependent Dirichlet process mixture with object and motion models to form the dependent Dirichlet process mixture of objects (DDPMO). The advantages of our model are that it can (a) accurately discern arbitrary video objects and track them in a fully unsupervised fashion, (b) jointly learn a model for each object and use these models to aid in object localization/tracking, (c) infer a distribution over the num-

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

ber of distinct objects present in a video, (d) incorporate a model for the motion of each object, and (e) begin tracking as objects enter the video frame, stop when they exit, and track through periods of partial or full occlusion.

2. Dependent Dirichlet Process Mixture of Objects (DDPMO)

2.1. Data Extraction

Our model operates on spatial and color pixel data that can be efficiently extracted via frame differencing: in each frame t, we find the pixel values that have changed beyond some threshold, and record pixel positions $\mathbf{x}^s_{t,n} = (x^{s_1}_{t,n}, x^{s_2}_{t,n})$ (where $n = 1,\ldots,N_t$ indexes the extracted pixels at time t). Additionally, the spectrum of RGB color values is discretized into V bins, and the local color distribution around each pixel is described by counts of surrounding pixels (in an $m \times m$ grid) that fall into each color bin, denoted $\mathbf{x}^c_{t,n} = (x^{c_1}_{t,n}, \ldots, x^{c_V}_{t,n})$. Observations are therefore of the form

$$\mathbf{x}_{t,n} = (\mathbf{x}^s_{t,n}, \mathbf{x}^c_{t,n}) = (x^{s_1}_{t,n}, x^{s_2}_{t,n}, x^{c_1}_{t,n}, \ldots, x^{c_V}_{t,n}) \quad (1)$$

Examples of spatial pixel data extracted via frame differencing are shown in Figure 1.

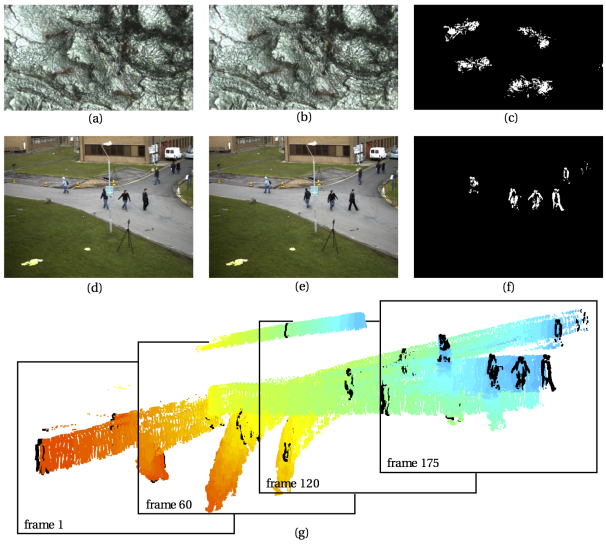

Figure 1. Two pairs of consecutive frames and the spatial observations $\mathbf{x}^s_{t,n}$ extracted by taking the pixel-wise frame difference between each pair (a - f). The final image shows the results of frame differencing over a sequence of images (from the PETS2010 dataset).

2.2. Object Model

An object model F is a distribution over observations

$$\mathbf{x}_{t,n} \sim \mathrm{F}(\theta^k_t) \quad (2)$$

where θ^k_t denotes the parameters of the k^{th} object at time t. We wish to keep our object model general enough to be applied to arbitrary video objects, but specific enough to learn a representation that can aid in tracking. In this paper, we model each object with

$$\mathbf{x}^s_{t,n} \sim \mathrm{Normal}(\mu_t, \Sigma_t) = \frac{e^{-\frac{1}{2}(\mathbf{x}^s_{t,n}-\mu_t)^\top \Sigma_t^{-1}(\mathbf{x}^s_{t,n}-\mu_t)}}{(2\pi)^{d/2}|\Sigma_t|^{1/2}} \quad (3)$$

$$\mathbf{x}^c_{t,n} \sim \mathrm{Mult}(\delta_t) = \frac{m^2!}{x^{c_1}_{t,n}! \cdots x^{c_V}_{t,n}!} (\delta^1_t)^{x^{c_1}_{t,n}} \ldots (\delta^V_t)^{x^{c_V}_{t,n}} \quad (4)$$

where object parameters $\theta_t = \{\mu_t, \Sigma_t, \delta_t\}$ and $\sum_{j=1}^V \delta^j_t = 1$. The object model captures the objects' locus and extent with the multivariate Gaussian and color distribution with the multinomial. This representation can capture the physical characteristics of a wide range of objects while providing distinctions between objects with different shapes, positions, and appearances.

2.3. Dependent Dirichlet Process Prior

Dirichlet process (DP) priors for component weights in mixture models have long been used as nonparametric Bayesian tools to estimate the number of clusters in data (Antoniak, 1974). Dependent Dirichlet process (DDP) mixtures extend this by allowing cluster parameters to vary with some covariate. In our case, a DDP object mixture lets us estimate, and capture the uncertainty in, the number of objects while modeling their time-varying parameters.

A DDP known as a generalized Polya urn (GPU) (Caron et al., 2007) has the desired properties that, when used in a mixture model, clusters can be created and die off and cannot merge or split. In this model, each data point $\mathbf{x}_{t,n}$ has an assignment $c_{t,n}$ to a cluster $k \in \{1,\ldots,K_{t,n}\}$. Each assignment increases the cluster's size $m^k_{t,n}$ by one. After each time step, cluster sizes may decrease when objects are "unassigned" in a deletion step. The GPU can be formulated as

1. At time $t = 1$
   (a) Set $c_{1,1} = 1$, $m^1_{1,1} = 1$, $K_{1,1} = 1$
   (b) for $n = 2,\ldots,N_1$
      i. Draw $c_{1,n} \sim \mathrm{Cat}(\frac{m^1_{1,n-1}}{n-1+\alpha}, \ldots, \frac{m^{K_{1,n-1}}_{1,n-1}}{n-1+\alpha}, \frac{\alpha}{n-1+\alpha})$
      ii. If $c_{1,n} \leq K_{1,n-1}$, set $m^{c_{1,n}}_{1,n} = m^{c_{1,n}}_{1,n-1} + 1$, $m^{\setminus c_{1,n}}_{1,n} = m^{\setminus c_{1,n}}_{1,n-1}$, and $K_{1,n} = K_{1,n-1}$
      iii. If $c_{1,n} > K_{1,n-1}$, set $m^{c_{1,n}}_{1,n} = 1$, $m^{\setminus c_{1,n}}_{1,n} = m^{\setminus c_{1,n}}_{1,n-1}$, and $K_{1,n} = K_{1,n-1} + 1$.
2. At time $t \geq 2$
   (a) Set $K_{t,0} = K_{t-1,N_{t-1}}$
   (b) For $k = 1,\ldots,K_{t,0}$

i. Draw $\Delta m^k_{t-1} \sim \text{Binom}(m^k_{t-1,N_{t-1}}, \rho)$

ii. Set $m^k_{t,0} = m^k_{t-1,N_{t-1}} - \Delta m^k_{t-1}$

(c) For $n = 1, \ldots, N_t$

i. Draw

$$c_{t,n} \sim \text{Cat}\left(\frac{m^1_{t,n-1}}{\alpha+\sum_k m^k_{t,n-1}}, \ldots, \frac{m^{K_{t,n-1}}_{t,n-1}}{\alpha+\sum_k m^k_{t,n-1}}, \frac{\alpha}{\alpha+\sum_k m^k_{t,n-1}}\right)$$

ii. If $c_{t,n} \leq K_{t,n-1}$, set $m^{c_{t,n}}_{t,n} = m^{c_{t,n}}_{t,n-1} + 1$, $m^{\setminus c_{t,n}}_{t,n} = m^{\setminus c_{t,n}}_{t,n-1}$, and $K_{t,n} = K_{t,n-1}$

iii. If $c_{t,n} > K_{t,n-1}$, set $m^{c_{t,n}}_{t,n} = 1$, $m^{\setminus c_{t,n}}_{t,n} = m^{\setminus c_{t,n}}_{t,n-1}$, and $K_{t,n} = K_{t,n-1} + 1$.

where Cat is the categorical distribution, $m^{\setminus c_{t,n}}_{t,n}$ is the set $\{m^1_{t,n}, \ldots, m^{K_{t,n}}_{t,n}\} \setminus \{m^{c_{t,n}}_{t,n}\}$, Binom is the binomial distribution, α is the DP concentration parameter, and ρ is a deletion parameter that controls temporal dependence of the DDP. We refer to step 1 as GPU($t = 1$) and step 2 as GPU($t \geq 2$).

2.4. Motion Model and Object Prior

We would also like to model the motion of objects. Assuming as little as possible, we take each object's parameters θ^k_t to be a noisy version of the previous parameters θ^k_{t-1} (if the object existed at the previous time) and define

$$\theta^k_t | \theta^k_{t-1} \sim \begin{cases} \text{T}(\theta^k_{t-1}) \text{ if } k \leq K_{t-1,N_{t-1}} \\ \mathbb{G}_0 \text{ if } k > K_{t-1,N_{t-1}} \end{cases} \tag{5}$$

where T denotes a transition kernel, the $k > K_{t-1,N_{t-1}}$ case is when a new cluster has been created at time t, and $\mathbb{G}_0$ is the base distribution of the dependent Dirichlet process, which represents the prior distribution over object parameters. We define $\mathbb{G}_0$ to be

$$\mathbb{G}_0(\theta^k_t) = \text{NiW}(\mu^k_t, \Sigma^k_t | \mu_0, \kappa_0, \nu_0, \Lambda_0)\text{Dir}(\delta^k_t | q_0) \tag{6}$$

where NiW denotes the normal-inverse-Wishart distribution and Dir denotes the Dirichlet distribution; these act as a conjugate prior to the object model. To meet technical requirements of the GPU, the transition kernel T must satisfy

$$\int \mathbb{G}_0(\theta^k_{t-1})\text{T}(\theta^k_t | \theta^k_{t-1}) d\theta^k_{t-1} = \mathbb{G}_0(\theta^k_t) \tag{7}$$

or, equivalently, its invariant distribution must equal the base distribution (Gasthaus, 2008). One way to satisfy this while providing a reasonable transition kernel is to introduce a set of M auxiliary variables $\mathbf{z}^k_t = (z^k_{t,1}, \ldots, z^k_{t,M})$ for cluster k at time t such that

$$P(\theta^k_t | \theta^k_{t-1}) = \int P(\theta^k_t | \mathbf{z}^k_t) P(\mathbf{z}^k_t | \theta^k_{t-1}) d\mathbf{z}^k_t \tag{8}$$

With this addition, object parameters do not directly depend on their values at a previous time, but are instead dependent through an intermediate sequence of variables. This allows the cluster parameters at each time step to be marginally distributed according to the base distribution $\mathbb{G}_0$ while maintaining simple time varying behavior. We therefore define the transition kernel $T(\theta^k_t | \theta^k_{t-1})$ generatively as

$$\begin{aligned} z^k_{t,1:M} &\sim T_1(\theta^k_{t-1}) \\ &= \text{Normal}(\mu^k_{t-1}, \Sigma^k_{t-1})\text{Mult}(\delta^k_{t-1}) \\ \mu^k_t, \Sigma^k_t, \delta^k_t &\sim T_2(z^k_{t,1:M}) \\ &= \text{NiW}(\mu_M, \kappa_M, \nu_M, \Lambda_M)\text{Dir}(q_M) \end{aligned} \tag{9}$$

where $\mu_M, \kappa_M, \nu_M, \Lambda_M$ and q_M are posterior updates given the auxiliary variables $z_{t,1:M}$ (defined in equations 13-17).

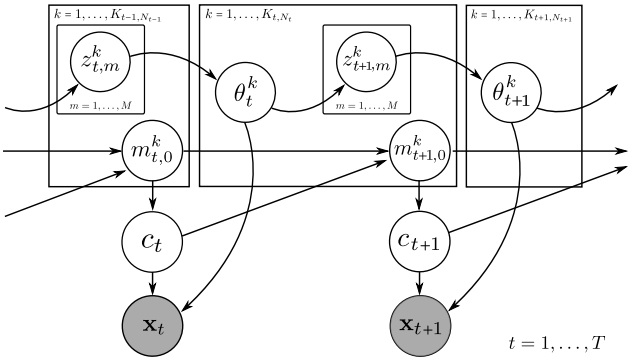

Figure 2. Graphical model of the dependent Dirichlet process mixture of objects (DDPMO). All observations at time t are denoted as $\mathbf{x}_t$ and their assignments as c_t.

2.5. DDPMO

The full generative process of the DDPMO is

1. At time t=1
   (a) Draw $\{c_{1,1:N_1}, K_{1,N_1}\} \sim \text{GPU}(t = 1)$
   (b) For $k = 1, \ldots, K_{1,N_1}$
      i. Draw $\theta^k_1 \sim \mathbb{G}_0(\mu_0, \kappa_0, \nu_0, \Lambda_0, q_0)$
   (c) For $n = 1, \ldots, N_1$
      i. Draw $\mathbf{x}_{1,n} \sim \text{F}(\theta^{c_{1,n}}_1)$
2. At time $t \geq 2$
   (a) Draw $\{c_{t,1:N_t}, K_{t,N_t}, m^{1:K_{t-1,N_{t-1}}}_{t,0}\} \sim \text{GPU}(t \geq 2)$
   (b) For $k = 1, \ldots, K_{t,N_t}$
      i. Draw
      $$\theta^k_t \sim \begin{cases} \text{T}(\theta^k_{t-1}) \text{ if } k \leq K_{t-1,N_{t-1}} \\ \mathbb{G}_0(\mu_0, \kappa_0, \nu_0, \Lambda_0, q_0) \text{ if } k > K_{t-1,N_{t-1}} \end{cases}$$
   (c) For $n = 1, \ldots, N_t$
      i. Draw $\mathbf{x}_{t,n} \sim \text{F}(\theta^{c_{t,n}}_t)$

where the notation $c_{1,1:N_1} = \{c_{1,1}, \ldots, c_{1,N_1}\}$. A graphical model for the DDPMO is shown in Figure 2.

3. Inference

We describe two inference algorithms for the DDPMO: sequential Monte Carlo (SMC) with local Gibbs iterations, and Particle Markov Chain Monte Carlo (PMCMC).

3.1. Sequential Monte Carlo

We first derive an SMC (particle filter) inference algorithm where we draw samples from a proposal distribution by iterating through local Gibbs updates (detailed in Section 3.1.1). SMC allows us to make a single pass through the data and draw posterior samples in an online fashion.

Algorithm 1 SMC for the DDPMO

Input: Extracted pixel data $\{\mathbf{x}_{1,1:N_1},\dots,\mathbf{x}_{T,1:N_T}\}$, number of particles L, number of local Gibbs iterations S

Output: Posterior samples $\left\{\theta_1^{1:K_{1,N_1}},\dots,\theta_T^{1:K_{T,N_T}}\right\}^{(1:L)}$ of the object model parameters

for $t=1$ **to** T **do**
  for $l=1$ **to** L **do**
    for iter $=1$ **to** S **do**
      Sample $(c_{t,1:N_t})^{(l)} \sim Q_1$ and $(\theta_t^{1:K_{t,N_t}})^{(l)} \sim Q_2$
    end for
    for $k=1$ **to** K_{t,N_t} **do**
      Sample $(\Delta m_t^k)^{(l)} \sim \text{Binom}((m_{t,N_t}^k)^{(l)},\rho)$
      Set $(m_{t+1,0}^k)^{(l)} = (m_{t,N_t}^k)^{(l)} - (\Delta m_t^k)^{(l)}$
      Sample $(z_{t+1,1:M}^k)^{(l)} \sim T_1((\theta_t^k)^{(l)})$
    end for
    Compute particle weight $\tilde{w}_t^{(l)}$
  end for
  Normalize particle weights and resample particles
end for

3.1.1. Local Gibbs Updates

We perform Gibbs sampling on the assignments and object parameters (at a given t) to draw SMC proposals; this allows for the proposal of well-mixed samples given newly introduced data in a particular frame. For an assignment $c_{t,n}$, we can compute a value proportional to the posterior for each possible assignment value $1,\dots,K_{t,n}$, and then sample from the resulting categorical distribution (after normalizing). The first proposal distribution Q_1 is the probability of an assignment $c_{t,n}$ given current cluster sizes, cluster parameters, and concentration parameter α, written

$$\begin{aligned} Q_1\left(c_{t,n}|m_{t,n-1}^{1:K_{t,n-1}},\theta_t^{1:K_{t,n-1}},\alpha\right) \propto \\ \text{Cat}(m_{t,n-1}^1,\dots,m_{t,n-1}^{K_{t,n-1}},\alpha) \\ \times \begin{cases} \text{F}(\mathbf{x}_{t,n}|\theta_t^{c_{t,n}}) & \text{if } c_{t,n} \le K_{t,n-1} \\ \int P(\mathbf{x}_{t,n}|\theta)\mathbb{G}_0(\theta)d\theta & c_{t,n} > K_{t,n-1} \end{cases} \end{aligned} \tag{10}$$

where we set the number of clusters $K_{t,n}$ and their sizes $m_{t,n}^{1:K_{t,n}}$ appropriately as each $c_{t,n}$ is assigned, and assume $K_{1,0}=0$ for consistency at $t=1$. The integral in the case of a new cluster ($k > K_{t,n-1}$) has an analytic solution

$$\begin{aligned} \int P(\mathbf{x}_{t,n}|\theta)\mathbb{G}_0(\theta)d\theta = \\ t_{\nu_0-1}\left(\mathbf{x}_{t,n}^s \mid \mu_0, \frac{\Lambda_0(\kappa_0+1)}{\kappa_0(\nu_0-1)}\right) \\ \times \prod_{j=1}^{V} \frac{\Gamma(\mathbf{x}_{t,n}^c)}{\Gamma(q_0)} \times \frac{\Gamma(\sum_{j=1}^V q_0)}{\Gamma(\sum_{j=1}^V \mathbf{x}_{t,n}^c)} \end{aligned} \tag{11}$$

where t_{ν_0-1} denotes the multivariate t-distribution with ν_0-1 degrees of freedom, where we follow the three-value parameterization (Gelman, 2004), and Γ denotes the gamma function.

The conjugacy of appearance model and transition kernel allow us to sample from the second proposal distribution Q_2, which is the posterior distribution over the object parameters given current observations, auxiliary variables, and previous time object parameters, written

$$\begin{aligned} Q_2(\theta_t^k|\theta_{t-1}^k,\mathbf{x}_{t,1:N_t}^k,z_{t,1:M}^k) = \\ \text{F}(\mathbf{x}_{t,1:N_t}^k|\theta_t^k)T_2(\theta_t^k|z_{t,1:M}^k) \\ = \text{NiW}(\mu_t^k,\Sigma_t^k|\mu_N,\kappa_N,\nu_N,\Lambda_N) \\ \times \text{Dir}(\delta_t^k|q_N) \end{aligned} \tag{12}$$

where $\mathbf{x}_{t,1:N_t}^k = \{x_{t,n} \in x_{t,1:N_t} | c_{t,n}=k\}$ and the parameters for the NiW and Dir distributions are given when $\mathbf{x}_{t,1:N_t}^k$ and $z_{t,1:M}^k$ are taken to be the "observations" in the following Bayesian updates

$$\kappa_N = \kappa_0 + N \tag{13}$$

$$\nu_N = \nu_0 + N \tag{14}$$

$$\mu_N = \frac{\kappa_0}{\kappa_0+N}\mu_0 + \frac{N}{\kappa_0+N}\bar{\mathbf{x}}^s \tag{15}$$

$$\Lambda_N = \Lambda_0 + S_{\mathbf{x}^s} \tag{16}$$

$$q_N = q_0 + \sum_{i=1}^{N} \mathbf{x}_i^c \tag{17}$$

where N is the number of observations, $\{\mu_0,\kappa_0,\nu_0,\Lambda_0\}$ are the NiW prior parameters, q_0 is the Dir prior parameter, $\mathbf{x}^s$ and $\mathbf{x}^c$ respectively denote the spatial and color features of the observations, and $\bar{\mathbf{x}}$ and $S_{\mathbf{x}}$ respectively denote the sample mean and sample covariance of the observations.

3.1.2. Particle Weights

At each time, the particle weights are set to be

$$\tilde{w}_t^{(l)} = \frac{P\left((c_{t,1:N_t})^{(l)},(\theta_t^{1:K_{t,N_t}})^{(l)},\mathbf{x}_{t,1:N_t}|(\theta_{t-1}^{1:K_{t-1,N_{t-1}}})^{(l)},(m_{t,0}^{1:K_{t-1,N_{t-1}}})^{(l)}\right)}{P\left((c_{t,1:N_t})^{(l)},(\theta_t^{1:K_{t,N_t}})^{(l)}|(\theta_{t-1}^{1:K_{t-1,N_{t-1}}})^{(l)},(m_{t,0}^{1:K_{t-1,N_{t-1}}})^{(l)}\right)} \tag{18}$$

where the numerator decomposes into

$$P\left(\mathbf{x}_{t,1:N_t}|(c_{t,1:N_t})^{(l)},(\theta_t^{1:K_{t,N_t}})^{(l)}\right) \times P\left((c_{t,1:N_t})^{(l)},(\theta_t^{1:K_{t,N_t}})^{(l)}|(\theta_{t-1}^{1:K_{t-1,N_{t-1}}})^{(l)},(m_{t,0}^{1:K_{t-1,N_{t-1}}})^{(l)}\right) \tag{19}$$

which can be computed using the DDPMO local probability equations defined in Section 2.5, and the denominator can be computed using equations 10 and 12. After the particle weights are computed, they are normalized; particles are then redrawn based on their normalized weights in a multinomial resampling procedure (Douc & Cappé, 2005).

3.1.3. Computational Cost

Assume N extracted pixels per frame, T frames, L particles, M auxiliary variables, S local Gibbs iterations, and fewer than K sampled objects ($K_{T,N_T}^{(1:L)} \leq K$). In the SMC inference algorithm, each local Gibbs iterations is $\mathrm{O}(KN+M)$ and evaluating each particle weight is $\mathrm{O}(K+N)$; the SMC algorithm therefore scales as $\mathrm{O}(TL(K(SN+M)+SM+N))$. If we neglect the number of auxiliary variables M, as it is almost always fixed at a small value, the algorithm scales as $\mathrm{O}(TLKSN)$. We have empirically found that an SMC implementation in MATLAB, while not tuned for speed, usually requires 4-20 seconds for every 1 second of video, depending on the number of objects (after frame-rate has been subsampled to approximately 3 images/second in all cases). It is not unreasonable to believe that this could be scaled to real time tracking, given parallel computation and efficient image processing.

3.2. Particle Markov Chain Monte Carlo

SMC provides an efficient, online method for posterior inference, but can suffer from degeneracy; notably, a large majority of the returned particles correspond to a single, non-optimal tracking hypothesis. Ideally, we would like to infer a full posterior over object paths. MCMC methods are guaranteed to yield true posterior samples as the number of samples tends to infinity; however, we have found batch Gibbs sampling to be impractical for inference in the DDPMO, as samples tend to remain stuck in local posterior optima (often when a track begins on one object before switching to another) and cannot converge to a high accuracy tracking hypothesis in a reasonable amount of time.

PMCMC (Andrieu et al., 2010) is a Markov chain Monte Carlo method that attempts to remedy these problems by using SMC as an intermediate sampling step to move efficiently through high dimensional state spaces. We implement a specific case known as the Particle Gibbs (PG) algorithm, where we sample from the conditional distributions used in Gibbs sampling via a modified version of Algorithm 1 referred to as conditional sequential Monte Carlo.

3.2.1. Conditional SMC

Conditional SMC (Andrieu et al., 2010) allows for SMC to be used as a proposal distribution in a Gibbs sampling algorithm. We must first introduce the notion of a particle's lineage. Let $A_t^{1:L}$ denote the indices of the L particles chosen during the resampling step in time t (in Algorithm 1). The lineage $B_{1:T}^{(l)}$ of a particle is recursively defined as $B_T^{(l)} = l$ and for $t = (T-1),\ldots,1$, $B_t^{(l)} = A_t^{B_{t+1}^{(l)}}$. More intuitively, for the l^{th} particle, which contains the variables $\Theta_{1:T}^{(l)}$ at the final time T, $B_t^{(l)}$ denotes the index of the particle that contained the variables $\Theta_{1:t}^{(l)} \subset \Theta_{1:T}^{(l)}$ at time t.

Conditional SMC uses lineages to ensure that a given particle will survive all resampling steps, whereas the remaining particles are generated as before. We define conditional SMC for the DDPMO in Algorithm 2. Note that computation of particle weights and resampling (for relevant particles) is performed in the same manner as in SMC inference (Algorithm 1).

Algorithm 2 Conditional SMC for the DDPMO

Input: Extracted pixel data $\{\mathbf{x}_{1,1:N_1},\ldots,\mathbf{x}_{T,1:N_T}\}$, number of particles L, number of local Gibbs iterations S, condition particle $\Phi_{1:T}^{(\eta)}$ with lineage $B_{1:T}^{(\eta)}$ $(\eta \in \{1,\ldots,L\})$

Output: Particle-conditional posterior samples $\{\Theta_{1:T}\}^{(1:L)}$ of all latent model variables

for $t = 1$ **to** T **do**
  for $l = 1$ **to** L **do**
    if $l \neq B_t^{(\eta)}$ **then**
      for iter $= 1$ **to** S **do**
        Sample $(c_{t,1:N_t})^{(l)} \sim Q_1$ and $(\theta_t^{1:K_{t,N_t}})^{(l)} \sim Q_2$
      end for
      for $k = 1$ **to** K_{t,N_t} **do**
        Sample $(\Delta m_t^k)^{(l)} \sim \mathrm{Binom}((m_{t,N_t}^k)^{(l)},\rho)$
        Set $(m_{t+1,0}^k)^{(l)} = (m_{t,N_t}^k)^{(l)} - (\Delta m_t^k)^{(l)}$
        Sample $(z_{t+1,1:M}^k)^{(l)} \sim T_1((\theta_t^k)^{(l)})$
      end for
      Set $\Theta_t^{(l)} = \left\{(c_{t,1:N_t})^{(l)},(\theta_t^{1:K_{t,N_t}})^{(l)},(m_{t+1,0}^{1:K_{t,N_t}})^{(l)},(z_{t+1,1:M}^{1:K_{t,N_t}})^{(l)}\right\}$
    else
      Set $\Theta_t^{(l)} = \Phi_t^{(\eta)}$
    end if
    Compute particle weight $\tilde{w}_t^{(l)}$
  end for
  for $l = 1$ **to** L **do**
    if $l \neq B_t^{(\eta)}$ **then**
      Normalize weights and resample particles
    end if
  end for
end for

3.2.2. Particle Gibbs

In the particle Gibbs (PG) algorithm (Andrieu et al., 2010), the model variables are first initialized, and then conditional SMC (Algorithm 2) is run for a number of iterations. More specifically, at the end of each iteration, a sample is drawn from the set of weighted particles returned by conditional SMC, and this sample is conditioned upon in the next iteration. The PG algorithm for the DDPMO is formalized in Algorithm 3.

As the PG inference requires all variables to be initialized, SMC inference (Algorithm 1) can be used as a quick way to provide near-MAP initialization of variables.

Algorithm 3 PMCMC (Particle Gibbs) for the DDPMO

Input: Extracted pixel data $\{\mathbf{x}_{1,1:N_1},\ldots,\mathbf{x}_{T,1:N_T}\}$, number of global Gibbs iterations G, number of particles L, number of local Gibbs iterations S

Output: Posterior samples $\left\{\theta_1^{1:K_{1,N_1}},\ldots,\theta_T^{1:K_{T,N_T}}\right\}^{1:G}$ of the object model parameters

1: Initialize all model variables to $\Phi_0^{(L)}$
2: **for** $g = 1$ **to** G **do**
3: Run conditional SMC (Algorithm 2) with input $\{\mathbf{x}_{1,1:N_1},\ldots,\mathbf{x}_{T,1:N_T}\}$, L, S, and conditional on particle $\Phi_{g-1}^{(L)}$ to get particle set $\{\Theta_{1:T}^{(1)},\ldots,\Theta_{1:T}^{(L)}\}$
4: Draw $\Phi_g^{(L)} \sim \text{Unif}(\{\Theta_{1:T}^{(1)},\ldots,\Theta_{1:T}^{(L)}\})$
5: **end for**
6: Return $\left\{\theta_1^{1:K_{1,N_1}},\ldots,\theta_T^{1:K_{T,N_T}}\right\}^{1:G} \in \Phi_{1:G}^{(L)}$

4. Experiments

The multivariate Gaussian components of the inferred object model parameters are used to provide centroid sequences (tracks), approximate ovals, and bounding boxes for the objects found in a video.

We first demonstrate the DDPMO on synthetic videos in order to illustrate its ability to track objects through occlusion and infer models of diverse objects. We then apply this model to three true videos to demonstrate finding and tracking objects in difficult detection scenarios, detecting a large number of objects, and using the unsupervised results to automatically train a supervised object detector. We also show how our results compare against detector-based methods on a recent human-tracking benchmark video.

4.1. Synthetic Videos

4.1.1. Tracking through Occlusion

We first construct simple synthetic videos to demonstrate tracking objects through occlusion (via the learned object models). The first three synthetic videos are each 200 frames and contain a number of colored squares of different (and potentially time-varying) sizes moving at varied speeds and trajectories over a black background. The synthetic videos contain instances of occlusion (of one or more squares) and objects with time-varying appearances, as these notoriously decrease the accuracy of detection and tracking.

The first two videos contain a red and a blue square, which begin at opposite sides of the scene at frame $f = 1$ and occlude (blue over red) in the center of the frame at $f = 100$ (Figure 3(a)). In the first video, both squares then continue in the same direction, and in the second they reverse directions and end at their initial starting positions. Data extraction yields identical spatial features in both videos; hence, successful tracking depends fully on the model of object color. The third video is the same as the first, except a third green square is added (Figure 3(b)). This square is 50 pixels/side at $f = 1$, travels across the frame while linearly shrinking to 10 pixels/side at $f = 100$ (where it is occluded), and grows back to 50 pixels/side to end at the opposite side of the frame.

The DDPMO successfully tracked the squares via the SMC inference algorithm in all three occlusion scenarios and correctly modeled the time-varying square in the third video (Figure 4 (a)-(c)).

4.1.2. Learning Diverse Object Models

In this experiment, we constructed a video from images of four physically different objects (in shape, scale, and appearance)—a truck, human, cat, and beetle—which randomly walk around the scene. The DDPMO fully tracks all four objects, and accurately learns a diverse set of object models. Inferred results for each object are shown in Figure 4(d)-(g).

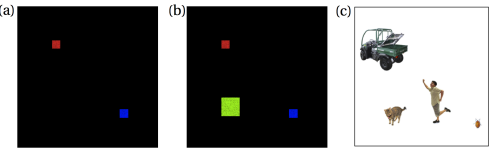

Figure 3. Still video frames from synthetic experiments 1 and 2 (a), 3 (b), and 4 (c).

4.2. Ants Video

In this experiment, we aim to demonstrate the ability of the DDPMO to find and track objects in a difficult detection scenario. The video contains six ants with a similar texture and color distribution as the background. The ants are hard to discern (even for the human authors), and it is unclear how a predefined detection criteria might be constructed. Futher, the ants move erratically and the spatial data extracted per ant via frame differencing is inconsistent between frames,

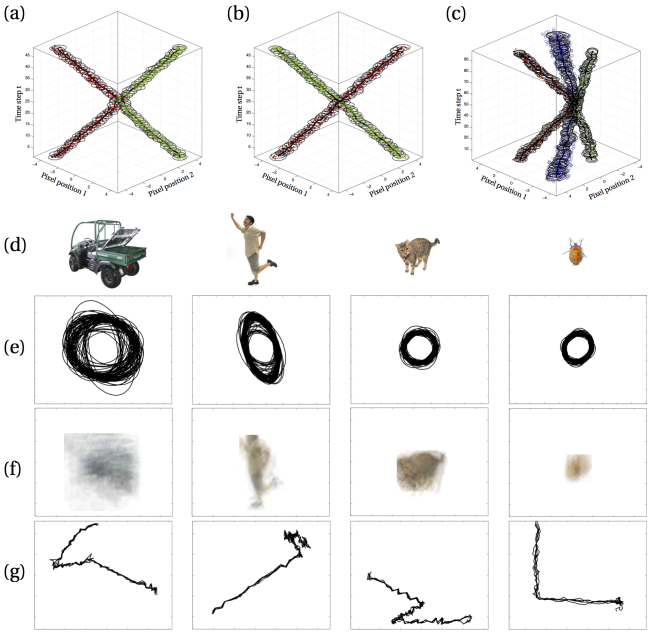

Figure 4. Inference results are plotted for synthetic experiments 1-3 (a-c). We show the four objects in synthetic experiment 4 (d), samples from the posterior over their spatial parameters (e), averages over extracted color observations for each inferred object (f), and samples from the posterior over the objects' tracks (g).

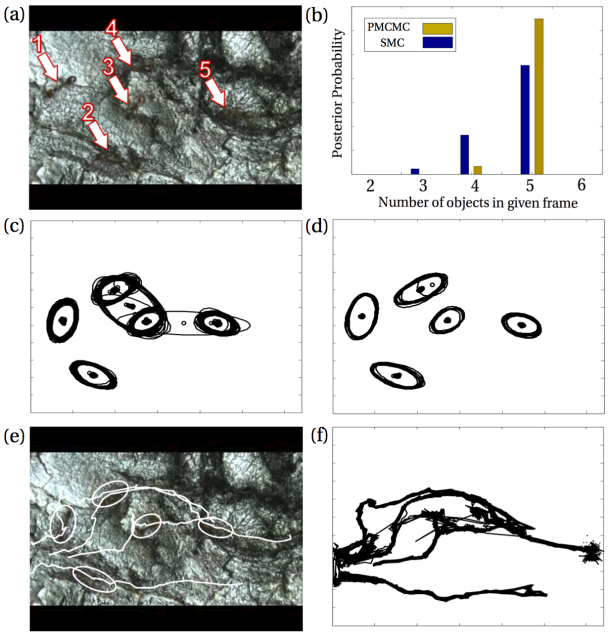

Figure 5. Ants in (a) are difficult to discern (positions labeled). We plot 100 samples from the inferred posterior over object parameters (using SMC (c) and PMCMC (d)) and over object tracks (f). PMCMC proves to give more accurate object parameter samples, which can also be seen by graphing the posterior distribution over the number of objects in the frame (b).

making it difficult to attempt a simple detection-free tracking method involving independent per-frame segmentation. A still image from the video, with ant locations shown, is given in Figure 5(a).

We compare SMC with PMCMC on this dataset, and find that PMCMC yields more accurate posterior samples (Figure 5(d)) than SMC (5(c)), and infers a more accurate posterior distribution over the number of objects (the posterior on a single frame for both methods is shown in 5(b)).

4.3. T Cells Video

Automated tracking tools for cells are useful for cell biologist and immunologists studying cell behavior. We present results on a video containing T cells that are hard to detect using conventional methods due to their low constrast appearance against a background (Figure 6(a)). Furthermore, there are a large number of cells (roughly 60 per frame, 92 total). In this experiment, we aim to demonstrate the ability of the DDPMO to perform a tough detection task while scaling up to large numbers of objects. Inference results are shown in Figure 6(c-d).

Futhermore, manually hand-labeling cell positions to train a detector is feasible but time consuming; we show how unsupervised detection results from the DDPMO can be used to automatically train a supervised cell detector (a linear SVM),

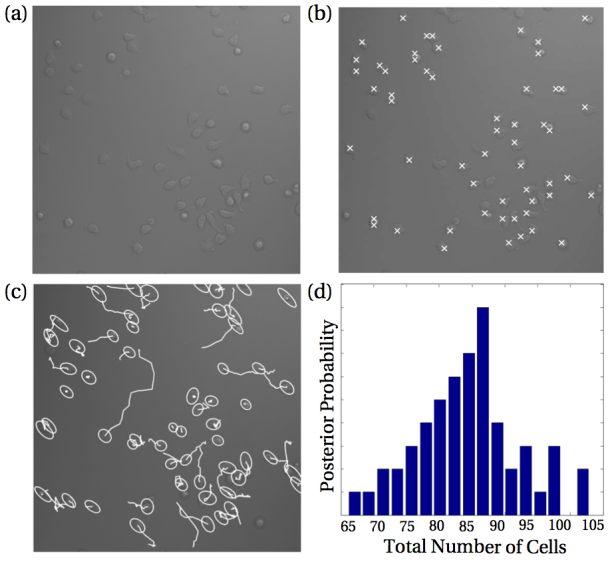

Figure 6. T cells are numerous, and hard to detect due to low contrast images (a). Inferred detection and tracking results are overlaid in (c). The posterior over number of cells is shown in (d), which peaks near the true value of 92 cells. Inferred cell positions (unsupervised) were used to automatically train an SVM for supervised cell detection; SVM detected positions are shown in (b).

METHOD NAME	SFDA	ATA
Breitenstein (Breitenstein et al., 2009a)	0.57	0.30
Yang (Yang et al., 2009)	0.55	0.45
Conte (Conte et al., 2010)	0.53	0.06
DDPMO	**0.51**	**0.30**
Berclaz (Berclaz et al., 2009)	0.48	0.15
Alahi 1 (Alahi et al., 2009)	0.43	0.04
Alahi 2 (Alahi et al., 2009)	0.42	0.05
Bolme 1 (Bolme et al., 2009)	0.41	NA
Ge (Ge & Collins, 2009)	0.38	0.04
Bolme 2 (Bolme et al., 2009)	0.34	NA
Arsic (Arsic et al., 2009)	0.18	0.02

Table 1. SFDA and ATA performance metric results for the DDPMO and ten human-specific detection-based tracking algorithms on the PETS2010 benchmark dataset. Results are listed in descending order of the SFDA value. The DDPMO performs detection-free tracking of arbitrary objects with comparable accuracy to these human-specific methods. Results provided by the authors of (Ellis & Ferryman, 2010).

which can then be applied (via a sliding window across each frame) as a secondary, speedy method of detection (Figure 6(b)).

4.4. Comparison with Detector-based Methods

Benchmark video datasets have been produced to provide standard scenes on which researchers can compare detection and tracking results. These videos have been primarily made for surveillance-related workshops, and notably for the International Workshop on Performance Evaluation of Tracking and Surveillance (PETS).

A video dataset used in the PETS2009-2013 conferences (referred to here as PETS2010) was chosen for experimentation due to its prominence in a number of studies (Table 1). This video consists of a monocular, stationary camera, 794 frame video sequence containing a number of walking humans. Due to the large number of frames and objects in this video, the SMC inference algorithm was used.

For comparison, we report two commonly used performance metrics for object detection and tracking, known as the sequence frame detection accuracy (SFDA) and average tracking accuracy (ATA) (Kasturi et al., 2008). These metrics compare detection and tracking results against human-authored ground-truth, where SFDA$\in [0,1]$ corresponds to detection performance and ATA$\in [0,1]$ corresponds to tracking performance. We authored all ground-truth with the Video Performance Evaluation Resource (ViPER) tool (Doermann & Mihalcik, 2000).

In Figure 7(a-d), the MAP sample from the posterior distribution over the object parameters is overlayed on the extracted data over a sequence of frames. Figure 7(e) shows the first 50 frames from the video, where the assignment of each data point is represented by color and marker type. The DDPMO is compared against ten state of the art detector-based methods from the PETS2010 conference, and yields comparable results (Figure 7(f)), receiving the fourth highest SFDA score out of eleven, and second highest (tied) ATA score out of 9 (Table 1).

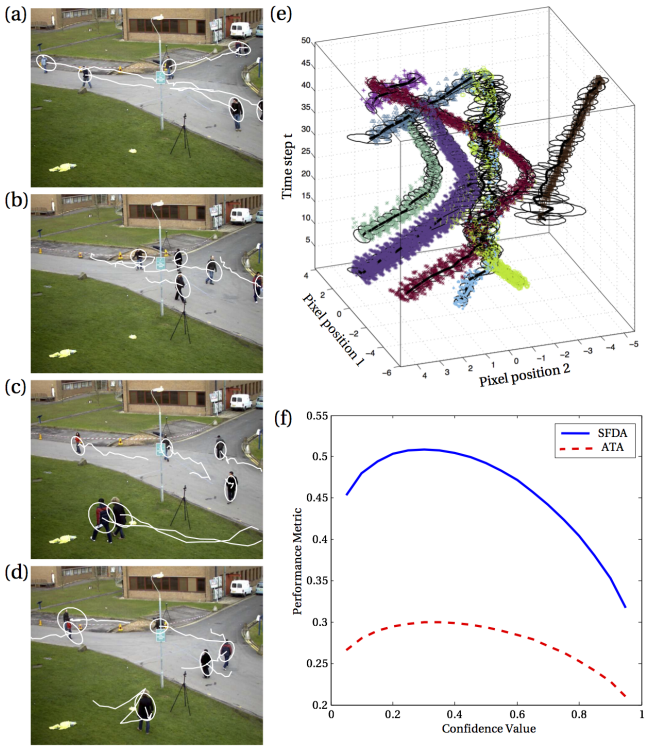

Figure 7. The PETS2010 human tracking benchmark for comparison with object-detector-based methods. MAP object parameter samples are overlaid on still video frames (a-d), and shown along with spatial observations for the first 50 frames (e) (where object assignment variables $c_{t,n}$ are denoted by marker type and color). We plot SFDA and ATA curves over different covariance matrix confidence thresholds (f), which dictate bounding box width.

5. Conclusion

The DDPMO provides the ability to find, track, and learn a representation for arbitrary objects in video, in a single model framework, to accomplish accurate detection-free tracking. We detail SMC and PMCMC algorithms for efficient inference and provide results on a number of synthetic and real video datasets that show the ability to track through occlusion, learn models for diverse objects whose appearances may drift, localize objects in difficult detection scenarios, scale to track large numbers of objects, and track certain detectable objects with performance comparable to state of the art detector-based methods.

References

Alahi, A., Jacques, L., Boursier, Y., and Vandergheynst, P. Sparsity-driven people localization algorithm: Evaluation in crowded scenes environments. In *Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop on*, pp. 1–8. IEEE, 2009.

Andrieu, Christophe, Doucet, Arnaud, and Holenstein, Roman. Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72 (3):269–342, 2010.

Antoniak, Charles E. Mixtures of dirichlet processes with applications to bayesian nonparametric problems. *The annals of statistics*, pp. 1152–1174, 1974.

Arsic, D., Lyutskanov, A., Rigoll, G., and Kwolek, B. Multi camera person tracking applying a graph-cuts based foreground segmentation in a homography framework. In *Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop on*, pp. 1–8. IEEE, 2009.

Babenko, Boris, Yang, Ming-Hsuan, and Belongie, Serge. Robust object tracking with online multiple instance learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33 (8):1619–1632, 2011.

Barron, John L, Fleet, David J, and Beauchemin, SS. Performance of optical flow techniques. *International journal of computer vision*, 12(1):43–77, 1994.

Berclaz, J., Fleuret, F., and Fua, P. Multiple object tracking using flow linear programming. In *Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop on*, pp. 1–8. IEEE, 2009.

Bolme, D.S., Lui, Y.M., Draper, BA, and Beveridge, JR. Simple real-time human detection using a single correlation filter. In *Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop on*, pp. 1–8. IEEE, 2009.

Breitenstein, M.D., Reichlin, F., Leibe, B., Koller-Meier, E., and Van Gool, L. Markovian tracking-by-detection from a single, uncalibrated camera. *Work*, 2009a.

Breitenstein, Michael D, Reichlin, Fabian, Leibe, Bastian, Koller-Meier, Esther, and Van Gool, Luc. Robust tracking-by-detection using a detector confidence particle filter. In *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 1515–1522. IEEE, 2009b.

Caron, F., Davy, M., and Doucet, A. Generalized Polya urn for time-varying Dirichlet process mixtures. In *23rd Conference on Uncertainty in Artificial Intelligence (UAI'2007), Vancouver, Canada, July 2007*, 2007.

Conte, D., Foggia, P., Percannella, G., and Vento, M. Performance evaluation of a people tracking system on pets2009 database. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pp. 119–126. IEEE, 2010.

Doermann, D. and Mihalcik, D. Tools and techniques for video performance evaluation. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 4, pp. 167–170. IEEE, 2000.

Douc, Randal and Cappé, Olivier. Comparison of resampling schemes for particle filtering. In *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*, pp. 64–69. IEEE, 2005.

Ellis, A. and Ferryman, J. Pets2010 and pets2009 evaluation of results using individual ground truthed single views. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pp. 135–142. IEEE, 2010.

Emonet, Rémi, Varadarajan, Jagannadan, and Odobez, J-M. Extracting and locating temporal motifs in video scenes using a hierarchical non parametric bayesian model. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 3233–3240. IEEE, 2011.

Fragkiadaki, Katerina and Shi, Jianbo. Detection free tracking: Exploiting motion and topology for segmenting and tracking under entanglement. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 2073–2080. IEEE, 2011.

Gasthaus, Jan. Spike sorting using time-varying Dirichlet process mixture models, 2008.

Ge, W. and Collins, R.T. Evaluation of sampling-based pedestrian detection for crowd counting. In *Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop on*, pp. 1–7. IEEE, 2009.

Gelman, A. *Bayesian data analysis*. CRC press, 2004.

Kasturi, R., Goldgof, D., Soundararajan, P., Manohar, V., Garofolo, J., Bowers, R., Boonstra, M., Korzhova, V., and Zhang, J. Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 319–336, 2008.

Meer, Peter. Kernel-based object tracking. *IEEE Transactions on pattern analysis and machine intelligence*, 25(5), 2003.

Turaga, Pavan, Chellappa, Rama, Subrahmanian, Venkatramana S, and Udrea, Octavian. Machine recognition of human activities: A survey. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(11):1473–1488, 2008.

Veeraraghavan, Harini, Schrater, Paul, and Papanikolopoulos, Nikos. Robust target detection and tracking through integration of motion, color, and geometry. *Computer Vision and Image Understanding*, 103(2):121–138, 2006.

Wang, Xiaogang, Ma, Xiaoxu, and Grimson, Eric. Unsupervised activity perception by hierarchical bayesian models. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1–8. IEEE, 2007.

Wu, Bo and Nevatia, Ram. Cluster boosted tree classifier for multi-view, multi-pose object detection. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–8. IEEE, 2007.

Yang, J., Vela, PA, Shi, Z., and Teizer, J. Probabilistic multiple people tracking through complex situations. In *11th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2009.

Yilmaz, Alper, Javed, Omar, and Shah, Mubarak. Object tracking: A survey. *Acm Computing Surveys (CSUR)*, 38(4):13, 2006.