Unsupervised Detection and Tracking of Multiple Objects with Dependent Dirichlet Process Mixtures

Willie Neiswanger and Frank Wood

September 27, 2011

Abstract

main points

1 Introduction

this section should introduce our work and situate it relative to the literature

This work is related to the study of algorithms that perform automated detection and tracking of arbitrary objects in videos; we define such algorithms as those with the ability to decide which spatiotemporal regions of a video constitute objects (extraction), find the positions and/or shapes of distinct objects (localization), and maintain the positions of the detected objects over time (tracking). Robust algorithms which can carry out these tasks are applicable to problems in the fields of surveillance, autonomous systems and robotics, and video summarization and compression. Approaches to these tasks have been proposed under different headings and in a variety of contexts. This section serves to introduce relevant terminology and topics of research.

The task of extraction is closely related to the topics of foregound segmentation, background modeling, motion detection, and appearance feature extraction. In general, these methods aim to extract features from regions of a video frame that are considered objects or are distinguished in some way from the background of a scene. Areas of research in this field include attempts to handle situations where the background displays motion or other dynamic characteristics, reduce noise caused by the false extraction of backgrounds, handle abberant image effects present in some videos, and operate accurately on scenes involving illumination changes.

The task of localization is closely related to the topics of object detection, position-finding, target segmentation, contour-detection, template matching, and recognition. These methods are mainly concerned with finding the positions and/or shapes of objects in a given video frame, or with discerning between objects similar in position or appearance. There has been a focus in this field on developing techniques for representing an object's appearance, both to allow for localization of objects on a per-frame basis and to allow for consistent localization of a given object over time. Research into the unsupervised detection of arbitrary objects (which we define as methods that accomplish both extraction and localization) has included work involving image segmentation, parsing, and pixel clustering. These topics usually relate to the automated partitioning of images into meaningful regions. Note that the popular phrase 'object detection' is also typically used to refer to procedures that carry out extraction and localization; however, it is often reserved to denote methods that are tailored to find the position of specified objects or object types (as opposed to arbitrary object detection), or to denote supervised methods, where extraction is peformed implicitely during searches for specified objects and localization procedures.

The task of tracking is closely related to the topics of video tracking, data association, filtering, point matching, and 'tracking of non-rigid objects'. In general, these tasks aim to maintain localization of the positions and/or shapes of objects over time. Areas of interest in this field involve developing metrics for judging the similarity between two object representations (for example, potential representations of the same object at adjacent frames in a video), techniques that provide a way to conduct efficient searches over a frame for regions that are similar to a given object, and methods for predicting the future position or appearance of an object. A great deal of research in this field over the past decade has focused on the development of algorithms to track multiple objects simultaneously. There has been a particular emphasis on developing ways to deal with problems such as object occlusions (where one object blocks another from the view of a video camera), complex object interactions, object coalescence (where one object is tracked by multiple times/redundantly), objects with similar appearances, variable (and potentially high) numbers of objects, and objects that enter and exit a field of view at different times.

There does not yet exist a major body of research concerned primarily with methods for combining extraction, localization, and tracking to perform cohesive unsupervised detection of arbitrary objects in videos. Many papers that attempt to accomplish this goal focus on one of extraction, localization, or tracking, and attempt to integrate in methods from the other two tasks. However, a few key schemes for unsupervised detection and tracking do indeed exist. A widely applied, though unsophisticated, technique to carry out such a procedure is called blob tracking. Blob tracking typically refers to methods that perform extraction and localization at each frame independently–for example, by assuming that object pixels have a different intensity than background pixels, setting a fixed pixel intensity threshold for extraction of object pixel locations, and segmenting the extracted data into 'blobs' to accomplish localization. Afterwards, tracking is often carried out by finding the centroid of each blob and applying standard multiple target tracking algorithms (described in [section]). Note that blob tracking carries out the three main tasks described above, so it satisfies the criteria for the type of unsupervised detection and tracking algorithms studied in this work; there are, however, many challenges associated with blob tracking that render it impractical in many settings (detailed in [section]).

This paper presents a new method for unsupervised multiple object detection and tracking. We introduce, and provide details on the theory behind, a technique involving dependent Dirichlet process mixture models. Specifically, this work applies the model to data gained during the extraction process; note that data of this type is refered to at different points in the paper as 'extraction data', 'data points', 'points', or 'observations'. The model is first presented in an abstract form, which could be adapted to a wide range of extraction procedures and object appearance representations (mainly left for future studies). We then describe inference algorithms that allow object localization and tracking to be carried out. Afterwards, we formulate a definite model for data gained during a specific extraction procedure and implement the described inference procedures to carry out multiple object detection and tracking in a number of experiments.

1.1 Related Work

Related work can be broken down into studies relating to extraction procedures, unsupervised localization procedures, single object tracking methods, multiple object tracking methods, supervised multiple object detection and tracking methods, and unsupervised multiple object detection and tracking methods.

Extraction procedures can usually be classified as either model-based methods or heuristic-based methods. Model-based methods emphasize the detection of shapes of regions containing objects accurately and reducing background noise. For example, foreground extraction has been robustly carried out by modeling each pixel over time in order to infer a probability distribution over pixel values; if the pixel takes on a very unlikely value at a future time, it is marked as a foreground pixel at that point [31, 12, 13]. These pixel modeling techniques have been adapted to allow for time-dependent probability distributions over pixel values, which help account for slighlty changing backgrounds, such as those caused from brief movements of background objects (for example, trees in outdoor scenes). Note that these techniques treat each pixel independently; this has a tendency to cause errors in foreground detection, resulting in fragmentation of foreground images (which is problematic as many of these methods intend to blob the foreground objects after detection in order to perform segmentation and locate objects' centroids). Post processing has been applied to add information regarding the edges of objects in an attempt to decrease fragmentation [32]. Others have attempted to achieve better foreground modeling in cases where the foreground and background exhibit similar color distributions (the so called 'color similarity problem') by, for example, incorporating models of the foreground into a typical background model (such as the one described above) [17, 24].

On the other hand, heuristic-based methods perform extracting by looking for temporal changes in pixel value characteristics. While these methods locate regions with objects less accurately than model-based methods, they are often much less computationally expensive. Frame differencing and background subtraction are two such methods, and involve comparing pixel values, or groups of pixel values, between two images, to detect a change surpassing some threshold. The goal of these techniques is to find locations in a scene that display motion, and to deem these moving areas the foreground of a scene. Often, background subtraction refers to comparing an image containing targets with an image of the background without targets or with some model of the background that is learned as the video progresses (in these cases, objects that do not move throughout a scene will be treated as the background), while frame or image differencing refers to comparing two subsequent images in a video. Frame differencing has been used as the sole extraction method for object localization or tracking schemes with success [28, 1, 7], and also as a secondary data extraction method to help improve the accuracy of object tracking schemes [29]. In addition to the two techniques mentioned above, there exists other ways of extracting movement from an image, such as through techniques involving optical flow (a calculation based on a so called 'generalized gradient model' of an image, which attempts to capture the speed and direction of movement over areas in the image) [20, 3]. In [2], a motion detection and object tracking method is developed based on the relative motion of moving objects' boundaries, which are again found through analysis of optical flow.

There are an extensive collection of methods for tracking single, non-rigid, objects. For example, color has long been used for tracking. In this case, tracking refers to the task of maintaining the location of an object after its initial position has been specified. In general, these schemes operate by modeling an object with some color-based appearance model, and using this model to find the object in subsequent frames. One classic approach involves modeling the color of a target (in particular, a distribution over the hue-saturation space of a target region) with a Gaussian mixture model (GMM), and choosing subsequent target positions by searching surrounding areas for regions that yield similar GMMs. Furthermore, the GMM is often allowed to adapt over time, and slowly change to model changes in lighting or other smooth time-based variations in the target's color [30, 25, 21]. Others have attempted to abstract this work with a non-parametric color modeling approach based on kernel density estimation, which does not assume a specific underlying distribution (such as the Gaussian mixture in the previous case) and instead converges to reasonable distribution that depends on the data [14]. Additionally, color has been successfully applied as the feature in the mean-shift procedure (kernel-based method) for tracking [10, 29, 27, 23].

The mean-shift procedure (also known as 'kernel-based object tracking') attempts to provide a robust way for non-rigid objects to be tracked. This method is beneficial because it optimizes the search for the 'next' position of an object (i.e. the search for the region in a frame which is most similar to a target region in the previous frame). This procedure–a derivation and overview of which can be found in [16, 6]–involves an iterative algorithm that repeatedly shifts each data point to the weighted average of data points in its neighborhood; it has been proven that this process converges for each data point. Additionally, the process (in particular, the specification of the neighborhood and the weight-distribution when calculating the weighted mean of nearby data points) is generalized so that multiple kernels can be specified and used to allow for varied clustering behaviors. It can be shown that, through this procedure, each data point becomes associated with a local point of high density (dependent upon the underlying weight distribution specified by the kernel) which naturally allows for clustering [6]. The mean shift algorithm has been implemented successfully to allow for a sort of kernel-based object tracking [10, 8, 9]. Given the current position of an object at a given frame, the goal of tracking is often to find a nearby position in the next frame that has the most similar distribution over some common set of features. Usually this must be done through an exhaustive search, comparing the similarity of distributions at each nearby position with that at the current position. However, if a certain type of 'isotropic kernel' (mean-shift kernel) known as the Bhattacharyya coefficient is chosen as a similarity metric between the feature distributions at two positions, it creates a smooth function where gradient descent techniques can be used to quickly converge upon an optimal subsequent position without using an exhaustive search. Many papers are concerned with applying this kernel-based object tracking scheme to different sets of features or with different kernels. Additionally, the scheme has been attempted with an adaptive kernel whose shape, scale, and orientation is influenced by a target being tracked [34], with a kernel adjusted by the estimated centroid of a tracked target [26], and with a heirarchical version of the mean-shift procedure [11].

Some have tried to extent single object tracking schemes to tracking multiple targets; at the most basic level, this involves initializing multiple targets and running an instance of (single) VOT for each, either simultaneously or in succession [29]. Furthermore, if performed simultaneously, the tracking of each target can be made dependent upon characteristics of other targets (such as their proximity) to resolve errors and improve tracking (such as those caused by the incorrect merging of two targets) [22, 33] .

1.2 Contributions

This paper applies a time dependent Dirichlet process mixture to model data gained during extraction procedures, provides background as to how this Bayesian nonparametric model allows for the incorporation of prior knowledge about object appearances and motion behaviors, describes methods for unsupervised detection and tracking of arbitrary objects in terms of Bayesian inference procedures, and presents a specific implementation of the model and inference procedures to carry out detection and tracking on a number of target-types in different settings.

This paper is heavily model-based, which differs from many papers in the field of multiple object tracking that do not explicitely describe models which underlie various procedures that are carried out. Solutions in many papers are, instead, algorithm oriented. The model-based approach allows rigor to be kept as inference procedures are developed. Furthermore, development of inference procedures are performed separately from development of the model, and various well studied inference and approximate inference methods can be carried out to achieve algorithmic results that are (in some cases) known to be optimal.

This idea presents a framework which allows for time-varying models of objects' appearances and motion behaviors to be incorporated, and is formulated in a Bayesian manner that allows distributions underlying the parameters of these models to be specified in a principled manner. It has been shown to work successfully in conjunction with simple (noisy) extraction methods. More robust, expensive, or in general accurate methods can be used for further accuracy. This method is able to handle the appearance and disappearance of objects from a scene, variable numbers of objects, cases where objects show time varying changes in behavior, and cases of occlusion.

Section two of this paper provides an overview of the model and describes how aspects can be modified for for modeling object appearance and motion behavior. Section three presents a specific implementation of the model and a number of inference procedures for carrying out object detection and tracking. Section four describes a number of experiments carried out with this implementation. Section five describes results of these experiments, comments on further directions of study, and provides concluding remarks.

2 Model Overview

This section provides details on the nature of the data to be modeled, Dirichlet process mixture (DPM) models, and dependent Dirichlet process mixture (DDPM) models. We end by making explicit the connection between elements of the model and object characteristics that these elements represent (for example, the appearance and motion behavior of objects are specifically captured). The provided model is formulated here in a sufficently abstract manner to allow for various representations of objects to be specified in future studies.

2.1 Data

We defined the task of extraction to be the unsupervised decision of which spatiotemporal regions of a video constitute objects; hence, the extraction data modeled in this work are D-dimensional observations that in some way provide information about the regions of the video frame that contain objects. A simple example of this information is shown by the three-dimensional extraction data $x = (d_1, d_2, t) \in \mathbb{R}^2 \times \mathbb{Z}_+$, where d_1 and d_2 are the spatial dimensions of pixels that comprise regions containing objects in a given frame, and t is the discrete time index of the frame. These three features are always available once regions containing objects are discerned, and should always be kept as features of the data for modeling (as position information is key for reasonable tracking and time information is necessary for our intention to model the temporal dependencies of the data).

Examples of additional features that could potentially be extracted from video frame regions containing objects include color information, pixel intensity values, feature point (such as corner, shape, or edge) locations or spatial characteristics, texture representations, transform results (such as results of hough or fourier transforms), and quantiative region descriptions derived from image segmentation procedures; the main idea is to choose features which are able to be easily extracted or computed, capture variability in the appearance of objects, are applicable to a wide variety of object types, and, when viewed in cumulation for multiple observations, are distributed in a way that allows for modeling and tractable inference (this is discussed further in section 3).

The baseline three-dimensional extraction data $x = (d_1, d_2, t) \in \mathbb{R}^2 \times \mathbb{Z}_+$ from pixels present in regions containing objects are distributed roughly as worm-like shapes that follow the paths of individual objects as they move, when they are plotted in three dimensions (where we let the frame number, our discrete representation of time, ascend the vertical axis and the spatial positions occupy the horizontal axes). An example of data with such a distribution can been seen in Figure 1. The data consists of the three-dimensional baseline features and was gained by extraction peformed on a 50 frame sequence showing five ants moving across the video scene. Inference performed on the time dependent Dirichlet process mixture model (described in the following sections) carries out clustering of the data, which allows each worm-like data cloud to be isolated and independent objects to be localized and tracked.

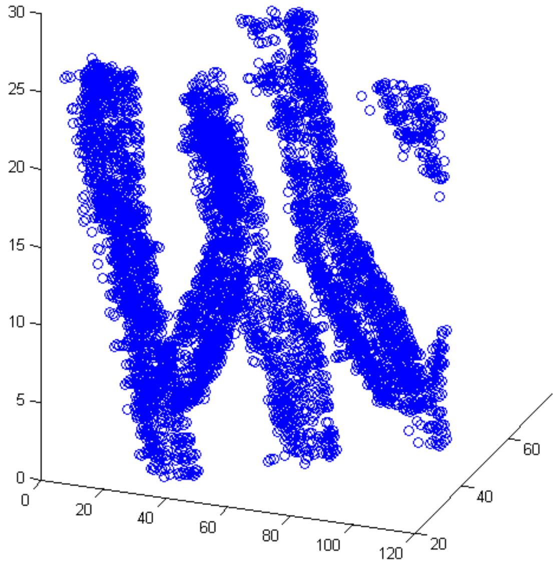

Figure 1: Baseline three-dimensional extraction data gained from a video of multiple moving targets. The vertical axis denotes the frame number and the horizontal axes denote d_1 and d_2 spatial coordinates, respectively, of pixels within a given frame. The data comes from a 50 frame video segment showing 5 moving ants (note that one ant enters the scene near the middle of the segment).

2.2 Dirichlet Process Mixtures

Dirichlet process mixtures fall under the heading of Bayesian nonparametric (BNP) models. These models have been widely used over the past decade to perform nonparametric density estimation and cluster analysis. They are, in particular, useful for estimating the number of latent classes (clusters) in mixture models. In this work, estimating the number of clusters in the data generated via extraction is shown in the following sections to be equivalent to estimating the number of distinct objects in a video. Consequentially, a model which is able to perform robust nonparametric estimation of the number of cluster is well suited to be applied to the task of unsupervised object detection and tracking.

The following sections are intended to provide a solid introduction to DPMs without a major foray into the theoretical aspects of the model that require a more rigorus technical setup. We give background on finite mixture models, Bayesian finite mixture models, and Dirichlet processes, presently, which allows us to subsequently describe DPMs in context.

2.2.1 Finite Mixture Model

A finite mixture model can be thought of as a probability distribution for an observation x_i formulated as a linear combination of K mixture components (where the coefficients of the linear combination sum to one); each mixture component, in turn, is a probability distribution for x_i, given some parametric form. The finite mixture model can be written as

$$P(x) = \sum_{k=1}^{K} P(c_i = k) P(x|\theta_k) \tag{1}$$

where $c_i \in \{1, \dots, K\}$ denotes the assignment of x_i to a given mixture, which allows the coefficients of the linear combination to be written as $P(c_i = k)$ (note that these coefficients now sum to one naturally due to the nature of probability distributions). Additionally, θ_k denotes the parametric form of the k^{th} mixture component. We also define $p_k = P(c_i = k)$ for all $k \in \{1, \dots, K\}$. This model can also be written hierarchically as

$$\begin{aligned} x_i|c_i, \theta_{c_i} &\sim F(\theta_{c_i}) \\ c_i|p_1, \dots, p_K &\sim \text{Discrete}(p_1, \dots, p_K) \end{aligned} \tag{2}$$

where the x_i (for $i \in \{1, \dots, N\}$) are observations, the c_i are the mixture component assignments associated with each observation, and the θ_{c_i} are parameters defining the c_i^{th} mixture component (i.e. the distribution to be mixed, $F(\theta_{c_i})$).

2.2.2 Bayesian (Finite) Mixture Model

The finite mixture model of the previous section can be extended to a Bayesian perspective by viewing the distribution parameters that were previously point values, θ_{c_i} (the mixture component parameters) and $p_1, \dots, p_K$ (the mixture component assignment weights), as random variables and providing each with a prior distribution. In this case, the prior distribution $\mathbb{G}_0$ is placed on the mixture component parameters, and the prior distribution $\text{Dir}(\alpha/K, \ldots, \alpha/K)$ is placed on the K mixture component assignment weights. The resulting Bayesian mixture model can be formulated in a hierarchical representation as

$$\begin{aligned} x_i|c_i, \theta_{c_i} &\sim F(\theta_{c_i}) \\ c_i|p_1, \ldots, p_K &\sim \text{Discrete}(p_1, \ldots, p_K) \\ \theta_{c_i} &\sim \mathbb{G}_0 \\ p_1, \ldots, p_K &\sim \text{Dir}(\alpha/K, \ldots, \alpha/K) \end{aligned} \tag{3}$$

where the x_i (for $i \in \{1, \ldots, N\}$) are observations, the c_i are the mixture component assignments associated with each observation, the θ_{c_i} are parameters defining the c_i^{th} mixture component (i.e. the distribution to be mixed, $F(\theta_{c_i})$), the θ_{c_i} are drawn from their prior distribution $\mathbb{G}_0$, and $p_1, \ldots, p_K$ are drawn from their prior distribution, where 'Dir' refers to a standard Dirichlet distribution.

2.2.3 Dirichlet Process

The Dirichlet process (DP), first introduced by [15] in 1973, may be intuitively viewed as a probability distribution over discrete probability distributions. Accordingly, draws from a DP are probability mass functions (PMFs). A DP is parameterized by a base distribution $\mathbb{G}_0$, which is a probability distribution over a set Θ, and a concentration parameter $\alpha \in \mathbb{R}_+$. We say that G is a random PMF distributed according to a DP, written $G \sim \text{DP}(\alpha, \mathbb{G}_0)$, if the following holds for all finite partitions $A_1, \ldots, A_p$ of Θ:

$$(G(A_1), \ldots, G(A_p)) \sim \text{Dir}(\alpha\mathbb{G}_0(A_1), \ldots, \alpha\mathbb{G}_0(A_p)) \tag{4}$$

Where 'Dir' denotes a standard Dirichlet distribution. The parameters $\mathbb{G}_0$ and α may be intuitively viewed as the mean and precision of the DP. This is due to the fact that if the base distribution $\mathbb{G}_0$ is a distribution over Θ, $A \subset \Theta$, and $G \sim \text{DP}(\alpha, \mathbb{G}_0)$, then the following holds:

$$\mathbb{E}[G(A)] = \mathbb{G}_0(A) \tag{5}$$

$$\text{Var}[G(A)] = \mathbb{G}_0(A)(1 - \mathbb{G}_0(A))/(\alpha + 1) \tag{6}$$

Hence, the expectation of $G(A)$ is $\mathbb{G}_0$, the variance of $G(A) \to 0$ as $\alpha \to \infty$, and G converges pointwise to $\mathbb{G}_0$ when α is unbounded.

2.2.4 Dirichlet Process (Infinite) Mixture Model

A Dirichlet process mixture model, also refered to as an infinite mixture model, is an extension of the Bayesian mixture model described previously. When using a DP as a prior in a Bayesian mixture model, Θ can be viewed as the set of parameters of the component mixture distributions. A DPM is formulated by allowing the prior mixing distribution (distribution over component weights?) in a standard mixture model to be distributed according to a DP; this allows for modeling data where the true number of latent mixture components is unknown and arbitrarily large by assuming an infinite number of components, of which only a finite amount are expressed by the data. In particular, the DPM can be defined hierarchically as

$$\begin{aligned} x_i|\phi_i &\sim F(\phi_i) \\ \phi_i|\mathbb{G} &\sim \mathbb{G} \\ \mathbb{G}|\alpha, \mathbb{G}_0 &\sim \mathrm{DP}(\alpha, \mathbb{G}_0) \end{aligned} \tag{7}$$

where the x_i (for $i \in \{1, \ldots, N\}$) are observations, the ϕ_i are parameters defining the mixture component from which the i_{th} observation is drawn (i.e. the distribution to be mixed, $F(\phi_i)$), and the ϕ_i are drawn from $\mathbb{G}$, which is in turn drawn from a DP with base distribution $\mathbb{G}_0$ and parameter α. See [18] and [19] for more details on this formulation. Note that difference between the formulation and indexing of the clusters in this model compared to the previous two models. This definition can be shown to be equivalent to the Bayesian mixture model defined in (3), when K is taken to be unbounded. This is the origin for references to this model as the infinite mixture model.

An alternative formultion of the DPM is known as the Chinese resturant process (CRP) sampling representation. If we let K be the number of distinct mixture components in the above model, we can write the distinct mixture components as $\theta_1, \ldots, \theta_K$. Let $c_1, \ldots, c_N$ (where $c_i \in \{1, \ldots, K\}$) be class assignment variables that indicate the cluster to which observation x_i is assigned. This allows the DPM to be represented by the CRP, a discrete-time stochastic process that defines a partition of the set $\{x_1, \ldots, x_N\}$ (via the elements' assignments $c_1, \ldots, c_N$). The CRP allows samples to be drawn from the conditional distribution of the indictor variables c_i, and can be expressed as

$$\begin{aligned} P(c_i = c_j \text{ for some } j < i) &= \frac{m_k}{i - 1 + \alpha} \\ P(c_i \neq c_j \text{ for all } j < i) &= \frac{\alpha}{i - 1 + \alpha} \end{aligned} \tag{8}$$

where m_k is the cardinality of the set $\{c_j | (c_j = c_i = k) \wedge (j < i)\}$ and α is the parameter of the DP prior on $\mathbb{G}$.

2.3 Dependent Dirichlet Process Mixtures

The goal of dependent Dirichlet process mixtures (DDPMs) is to allow modeling of data that is not independent and identically distributed (i.i.d) but instead has some underlying dependencies. For example, data generated during extraction procedures from videos have some associated temporal structure, since tracked objects display time dependent characteristics (in the features that denote position, as well as those which denote other appearance characteristics).

To account for the dependent behavior of data, research has been conducted on developing models involving a sequence of DPMs, where components of the mixtures are dependent upon (or may be considered 'tied to') corresponding components at adjacent positions in the sequence. For example, if the data shows temporal dependence, the goal might be to create a sequence of DPMs, one for each time-step, where the components of the mixture at each step are dependent upon corresponding components in the both the following and previous time steps. Research towards models which accomplish such goals are described in (list citations).

More rigorously, we take the definition of a DDPM to be a stochastic process defined on the space of probability distributions over a domain, which are indexed by time, space, or a selection of other covariates in such a way that the marginal distribution at any point in the domain follows a Dirichlet process (adapted from definitions found in [cite griffin and steel and gasthaus-thesis]).

2.3.1 Generalized Polya Urn Dependent Dirichlet Process Mixture

Data gained by performing extraction on videos containing objects is significantly time-dependent. Futhermore, the clusters of data, each of which represents an object in the video, might change in number as time progresses, since objects can enter and exit a scene. The specific DDPM used to model extraction data in this work while handling the above challenges is known as the Generalized Polya Urn Dependent Dirichlet Process Mixture (GPUDDPM), and was introduced by [4] in 2006.

The GPUDDPM, when applied to data over T discrete time steps, can be viewed as a collection of DPMMs (one for each $t \in \{1, \ldots, T\}$), which are linked together by dependencies among their respective parameters. For example, the mixture component parameters at a given time step are dependent upon the mixture component parameters in adjacent time steps and the distribution and number of distinct cluster assignments at a given time step are dependent upon the same over a wider range of time steps, upon the distribution of observations' cluster assignments over this range, and upon a deletion procedure, described below.

To link together the parameters of mixture components in adjacent time-steps, a transition kernel $P(\phi_k^t | \phi_k^{t-1})$ is implemented, which provides a distribution over parameters of a mixture component in a given time-step (as a function of equivalent parameters in adjacent time-steps). One caveat is that each mixture component must be drawn independently from $\mathbb{G}_0$ (the base distribution of the DP, which acts as a prior distribution for the cluster parameters) which we achieve by making $\mathbb{G}_0$ the invariant distribution of $P(\phi_k^t | \phi_k^{t-1})$ (note that the transition kernel is a markov chain). Additionally, to account for varying numbers of clusters, there is a deletion procedure (described in [5]) by which observations are considered ”removed” from their assigned clusters at a given time-step (which modifies the value of m_k defined above in the CRP representation). We introduce variables $d_1, \ldots, d_N$ which denote the times at which the assignments of given observations are considered to be removed. With these variables, we can now express the m_k value at time t as

$$m_{k,t} = \sum_{t'=1}^{t} \mathbb{I}[(c_{t'} = k) \wedge (t < d_{t'})] \tag{9}$$

where $\mathbb{I}[\cdot]$ is an indicator function that evaluates to 1 if its argument is true, and 0 otherwise. Additionally, for an observation x_i at a given timestep t, the deletion time d_i can be defined to be $d_i = t + l_i$, where l_i is considered the lifespan of an assignment, is distributed geometrically, and can be expressed as

$$l_i|\rho \sim \rho(1-\rho)^{l_i} \tag{10}$$

with parameter ρ. This process adheres to what [5] calls a 'uniform deletion strategy' over all the observations' assignments (since assignments to each cluster have equal probability of being deleted), though a more complex deletion strategy, dependent upon cluster size, can also be implemented.

Using the cluster size and observation deletion terms introduced above, we can define the GPUDDPM hierarchically as, for each time step $t = \{1, \ldots, T\}$

$$\begin{aligned} x_i^{(t)}|c_i^{(t)}, \theta_{c_i^{(t)}}^{(t)} &\sim F(\theta_{c_i^{(t)}}^{(t)}) \\ c_i^{(t)}|m^{(t-1)}, \alpha &\sim \text{CRP}(m^{(t-1)}, \alpha) \\ m^{(t-1)}|m^{(t-2)}, c_i^{(t-1)} &\sim \text{DEL}(m^{(t-2)}, c_i^{(t-1)}, \rho, \gamma) \\ \theta_k^{(t)}|\theta_k^{(t-1)} &\sim \begin{cases} P(\theta_k^{(t)}|\theta_k^{(t-1)}) \\ \mathbb{G}_0 \end{cases} \end{aligned} \tag{11}$$

The graphical corresponding with this formulation is shown in Figure 2.

2.3.2 Object Characteristics Modeled

We would like make explicit the connection between certain elements of the GPUDDPM model and the characteristics of objects in vidoes that these elements represent.

First, the appearances of objects are moded by what we term the 'likelihood and object appearance model', which we denoted by F. In the above modeling formulation, F is the probability distribution for an observation, given that it is assigned to specified cluster (i.e. given that it is associated with a given object in a video). The likelihood and object appearance model is therefore

$$F(\theta_{c_i}) = P(x_i|\theta_{c_i}) \tag{12}$$

where x_i is an observation, c_i is its associated assignment, θ_{c_i} specifies the parameters of the parametric form of the c_i^{th} cluster. The specific form of this model is dependent upon the observations x_i; recall that the observations, which as a baseline include spatial and temporal features, could include an arbitrary amount of additional features which need to be incorporated into this appearance model. A specific formulation of the appearance model can be seen in the proceeding section.

Object behavior is modeled by the dependencies between corresponding clusters at adjacent time steps. In particular, this relationship is captured by the transition kernel, $P(\theta_k^t|\theta_k^{t-1})$ (which captures the dependence of cluster k at time t on the same cluster at time $t-1$), described

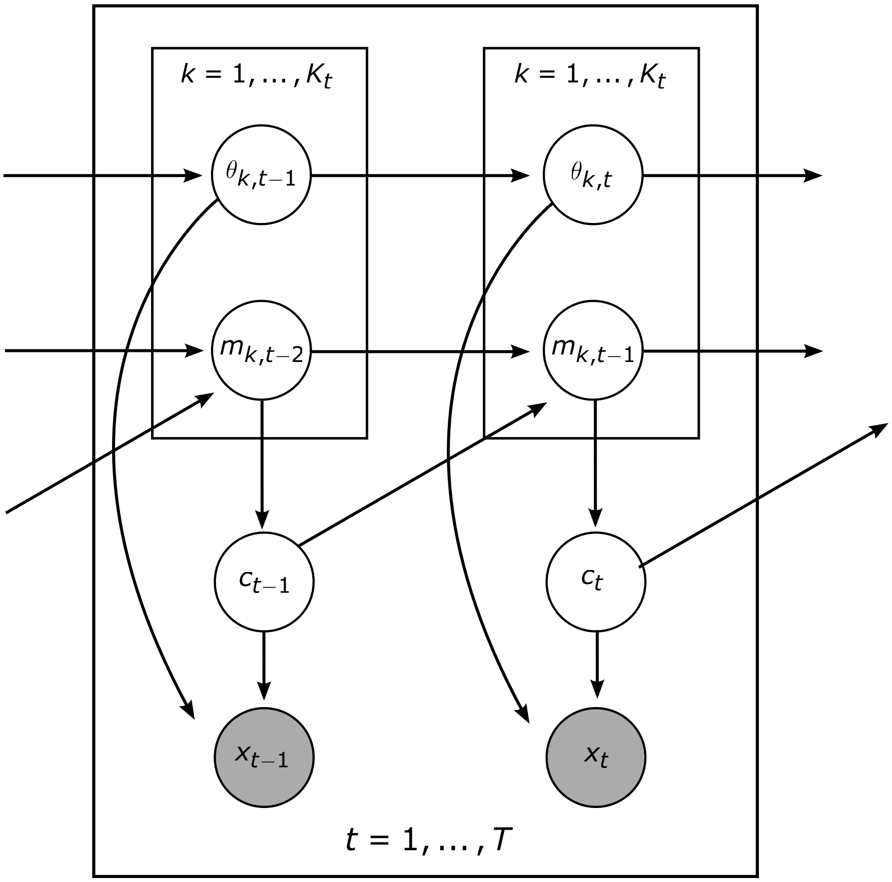

Figure 2: Graphical Model of the Generalized Poly Urn Dependent Dirichlet Process Mixture

previously. The motion behavior of an object, i.e. the specific dependence of the parameters of cluster k at a time t on its parameters at time $t-1$, is therefore

$$Tr(\theta_{k,t}|\theta_{k,t-1}) = P(\theta_{k,t}|\theta_{k,t-1}) \tag{13}$$

Additionally, the base distribution $\mathbb{G}_0$ of the DP acts as a prior on the cluster parameters. This allows one to place a prior probability over appearances of objects. It is important that this prior is not chosen in such a specific manner that it limits this method from performing unsupervised detection of arbitrary objects.

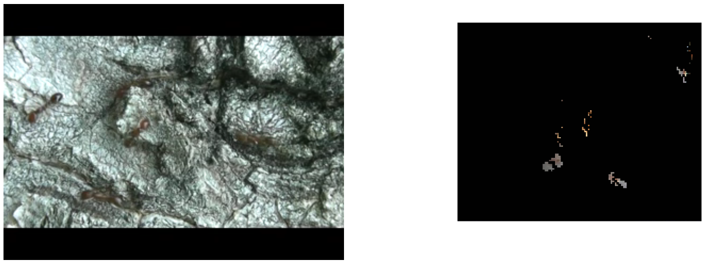

Figure 3: (a.) Frame of a video containing five ants. (b.) Frame differencing extraction results from the same frame (note: this is just a demo of the type of image i will have here... these aren't actually from the same frame, and also formatting is screwed up)

3 Model Specification and Inference

A particular method for extraction from videos was chosen to generate data and a specific formulation of the model described in the previous section was constructed. Inference on this model was carried out to perform unsupervised detection and tracking of multiple arbitrary objects. Model choices are discussed and inference schemes are described in the following sections.

3.1 Extraction

Out of the numerous extraction procedures discussed at the beginning of this paper, we desired one that was as unsophisticated as possible, both to test the robustness of this method on potentially noisy extraction data and to ensure that the extraction procedure was applicable to a wide range of videos. We decided upon the extraction procedure of simple frame-differencing, which involves looking at the difference in pixel intensity or value in successive frames, and recording the positions of the pixels (or pixel groups) that have exhibited frame-wise change beyond a given threshold.

Frame differencing is simple, quick, able to be applied to a wide range of static, single-camera videos (note that videos used in the experiments described in the following sections were chosen to be static; moving-camera videos should be used in conjunction with applicable extraction methods that allow camera movement). We have found that, when correctly implemented, it is sufficiently general to extract the worm-like data clouds from a wide variety of videos containing multiple moving objects. A few examples of this extraction procedure on single video frames can be found in Figure 3.

3.2 Model Implementation in Experiments

3.2.1 Likelihood and Object Appearance Model

For a given object k at video frame t, the set of positions of the object's motion points are modeled with a product (over each position dimension) of one-dimensional Gaussians

$$P(x_{t_i}^{pos}|\mu_{t,k},\lambda_{t,k}) = \prod_{d=1}^{D} \mathcal{N}(x_{t_i}^{pos_d}|\mu_{t,k}^d,\lambda_{t,k}^d) \tag{14}$$

where $x_{t_i}^{pos} = \{x_{t_i}^{pos_1},\cdots,x_{t_i}^{pos_D}\}$, $\mu_{t,k}^d$ is the mean of the k^{th} Gaussian at time t for dimension d, and $\lambda_{t,k}^d$ is the precision of the k^{th} Gaussian at time t for dimension d. Likewise, for a given object k at video frame t, the set of color vectors of the object's motion points are modeled with a multinomial

$$P(x_{t_i}^{col}|m_{t,k}) = Mult(x_{t_i}^{col}|m_{t,k}) \tag{15}$$

where $x_{t_i}^{col} = \{x_{t_i}^{col_1},\cdots,x_{t_i}^{col_V}\}$, $m_{t,k} = \{m_{t,k}^1,\cdots,m_{t,k}^V\}$, $\sum_{v=1}^{V} m_{t,k}^v = 1$, and $m_{t,k}^v > 0 \;\; \forall v \in \{1,\cdots,V\}$. The likelihood for a motion point observation is thus

$$P(x_{t_i}|\mu_{t,k},\lambda_{t,k},m_{t,k}) = Mult(x_{t_i}^{col}|m_{t,k}) \prod_{d=1}^{D} \mathcal{N}(x_{t_i}^{pos_d}|\mu_{t,k}^d,\lambda_{t,k}^d) \tag{16}$$

I want to justify distribution families chosen for appearance modeling in this implementation. To include: the edges of an object are detected well with frame differencing, so while extraction data densities aren't necessarily Gaussians with maxima over the targets, they are usually radially centered around the centroids of targets. Furthermore (thing frank said about gaussians and what they do—minimize least squared distance or something).

3.2.2 Base Distribution $\mathbb{G}_0$ and Appearance Prior

$\mathbb{G}_0$ is the base distribution of the underlying time-dependent Dirichlet process mixture; it also serves as a prior distribution for the parameters present in the likelihood. We make use of conjugate priors in the base distribution to allow for more efficient computation. In particular, for an object k at time t, we let the prior distribution over the parameters of the motion point position distribution, $\mu_{t,k}$ and $\lambda_{t,k}$, be

$$\mathbb{G}_0^{pos}(\mu_{t,k},\lambda_{t,k}|\mu_0,n_0,a,b) = \prod_{d=1}^{D} [\mathcal{N}(\mu_{t,k}^d|\mu_0,n_0\lambda_{t,k}^d)Ga(\lambda_{t,k}^d|a,b)] \tag{17}$$

where μ_0, n_0, a, and b are parameters of the base distribution. Furthermore, for object k at time t, we let the prior distribuion over the parameters of the motion point color vector distribution be

$$\mathbb{G}_0^{col}(m_{t,k}|q) = Dir(m_{t,k}|q) \tag{18}$$

where $q = \{q^1, \cdots, q^V\}$ is a parameter of the base distribution, where $q^v > 0 \quad \forall v \in \{1, \cdots, V\}$. The base distribution is thus

$$\mathbb{G}_0(\mu_{t,k}, \lambda_{t,k}, m_{t,k}|\mu_0, n_0, a, b, q) = Dir(m_{t,k}|q) \prod_{d=1}^{D} [\mathcal{N}(\mu_{t,k}^d|\mu_0, n_0\lambda_{t,k}^d) Ga(\lambda_{t,k}^d|a, b)] \tag{19}$$

Transition Kernels and Motion Behavior

Let $\phi_{t,k}^{pos} = \{\mu_{t,k}, \lambda t, k\}$ and $\phi_{t,k}^{col} = \{m_{t,k}\}$. We define two transition kernels, $P(\phi_{t,k}^{pos}|\phi_{t-1,k}^{pos})$ and $P(\phi_{t,k}^{col}|\phi_{t-1,k}^{col})$, which dictate, respectively, the random walk of the motion point position parameters and the motion point color vector parameters over time. The transition kernels must be chosen so that their invariation distributions are $\mathbb{G}_0$, i.e. such that the following hold:

$$\int \mathbb{G}_0^{pos}(\phi_{t-1,k}^{pos}) P(\phi_{t,k}^{pos}|\phi_{t-1,k}^{pos}) d\phi_{t-1,k}^{pos} = \mathbb{G}_0^{pos}(\phi_{t,k}^{pos}) \tag{20}$$

$$\int \mathbb{G}_0^{col}(\phi_{t-1,k}^{col}) P(\phi_{t,k}^{col}|\phi_{t-1,k}^{col}) d\phi_{t-1,k}^{col} = \mathbb{G}_0^{col}(\phi_{t,k}^{col}) \tag{21}$$

3.3 Inference Implementations in Experiments

Include a graphical model for each of the specified inference methods here? Or do I only need one graphical model for the model specified above?

Include algorithms for each of the inference procedures.

Experiments

3.4 Synthetic Videos

3.4.1 Occlusions

3.4.2 Similar Appearances

3.4.3 Different Numbers of Objects

3.4.4 Objects wih Diverse Characteristics

3.5 Real-life Videos

3.5.1 Complex Behavior

Ants

3.5.2 Occlusions

Various people scenes

3.5.3 Objects with Diverse Characteristics

Traffic and Humans (PETS scenes)

3.5.4 Standard Data for Comparison

PETS and other standard datasets

Results and Discussion

Conclusion

References

[1] C. Beleznai, B. Fr ”uhst ”uck, and H. Bischof. Human tracking by fast mean shift mode seeking. *Journal of Multimedia*, 1(1):1–8, 2006.

[2] M.J. Black and D.J. Fleet. Probabilistic detection and tracking of motion boundaries. *International Journal of Computer Vision*, 38 (3):231–245, 2000.

[3] A.F. Bobick and J.W. Davis. The recognition of human movement using temporal templates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(3):257–267, 2001.

[4] F. Caron, M. Davy, and A. Doucet. Generalized polya urn for time-varying dirichlet process mixtures. In *23rd Conference on Uncertainty in Artificial Intelligence (UAI2007), Vancouver, Canada*. Citeseer, 2007.

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

[6] Y. Cheng. Mean shift, mode seeking, and clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(8):790–799, 1995.

[7] H. Chu, S. Ye, Q. Guo, and X. Liu. Object tracking algorithm based on camshift algorithm combinating with difference in frame. In *Automation and Logistics, 2007 IEEE International Conference on*, pages 51–55. IEEE, 2007.

[8] D. Comaniciu and P. Meer. Mean shift analysis and applications. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1197–1203. Ieee, 1999.

[9] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 142 –149 vol.2, 2000. doi: 10.1109/CVPR.2000.854761.

[10] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(5):564 – 577, may 2003. ISSN 0162-8828. doi: 10.1109/TPAMI.2003.1195991.

[11] D. DeMenthon, R. Megret, and Md.). Computer Vision Laboratory University of Maryland (College Park. *Spatio-temporal segmentation of video by hierarchical mean shift analysis*. Citeseer, 2002.

[12] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. *Computer VisionECCV 2000*, pages 751–767, 2000.

[13] A. Elgammal, R. Duraiswami, D. Harwood, and L.S. Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of the IEEE*, 90(7): 1151–1163, 2002.

[14] Ahmed Elgammal, Ramani Duraiswami, and Larry S. Davis. Efficient non-parametric adaptive color modeling using fast gauss transform. In *in Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 563–570, 2001.

[15] T.S. Ferguson. A bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230, 1973.

[16] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *Information Theory, IEEE Transactions on*, 21(1):32–40, 1975.

[17] J. Gallego, M. Pardas, and G. Haro. Bayesian foreground segmentation and tracking using pixel-wise background model and region based foreground model. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 3205–3208. IEEE, 2009.

[18] J. Gasthaus, F. Wood, D. Görür, and Y. W. Teh. Dependent Dirichlet process spike sorting. In *Advances in Neural Informations Processing Systems 22*, 2008.

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

[20] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

[21] A.D. Jepson, D.J. Fleet, and T.F. El-Maraghi. Robust online appearance models for visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1296–1311, 2003.

[22] Z. Khan, T. Balch, and F. Dellaert. An mcmc-based particle filter for tracking multiple interacting targets. *Computer Vision-ECCV 2004*, pages 279–290, 2004.

[23] S.H. Lee, E. Choi, and M.G. Kang. Illumination change adaptive tracking based on color centroid shifting. *Optical Engineering*, 50: 057205, 2011.

[24] J.M. McHugh, J. Konrad, V. Saligrama, and P.M. Jodoin. Foreground-adaptive background subtraction. *Signal Processing Letters, IEEE*, 16(5):390–393, 2009.

[25] S.J. McKenna, Y. Raja, and S. Gong. Tracking colour objects using adaptive mixture models. *Image and Vision Computing*, 17(3-4): 225–231, 1999.

[26] R. Mehmood, M.U. Ali, and I.A. Taj. Applying centroid based adjustment to kernel based object tracking for improving localization. In *Information and Communication Technologies, 2009. ICICT '09. International Conference on*, pages 209 –214, aug. 2009. doi: 10.1109/ICICT.2009.5267188.

[27] K. Nummiaro, E. Koller-Meier, and L. Van Gool. An adaptive color-based particle filter. *Image and Vision Computing*, 21(1):99–110, 2003.

[28] A.E.C. Pece. Generative-model-based tracking by cluster analysis of image differences. *Robotics and Autonomous Systems*, 39(3-4):181–194, 2002.

[29] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. *Computer VisionECCV 2002*, pages 661–675, 2002.

[30] Y. Raja, S.J. McKenna, and S. Gong. Tracking and segmenting people in varying lighting conditions using colour. In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pages 228–233. IEEE, 1998.

[31] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2. IEEE, 1999.

[32] D. Turdu and H. Erdogan. Improved post-processing for gmm based adaptive background modeling. In *Computer and information sciences, 2007. iscis 2007. 22nd international symposium on*, pages 1–6. IEEE, 2007.

[33] J. Vermaak, A. Doucet, and P. Pérez. Maintaining multimodality through mixture tracking. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1110–1116. IEEE, 2003.

[34] J. Wang, B. Thiesson, Y. Xu, and M. Cohen. Image and video segmentation by anisotropic kernel mean shift. *Computer Vision-ECCV 2004*, pages 238–249, 2004.