Object Detection and Tracking with Dependent Dirichlet Processes

Willie and Frank

September 3, 2011

Introduction

An unsupervised algorithm for the automated localization and tracking of arbitrary objects in videos is desirable; an algorithm of this sort is one with the capability to decide which parts of a video constitute objects or regions of interest (detection), find the positions of distinct objects (localization), and maintain the positions of all detected objects over time (tracking).

The march towards such an algorithm can be seen in the approaches taken to solve a number of related problems over the past three decades. Multiple target tracking (MTT), also know as data association or point matching, involves providing a general localization and tracking solution for objects represented as two-dimensional points; a few algorithms have been well established since the 1980s. Single object tracking (SOT), also known as video tracking, 2D tracking, or 'tracking of non-rigid objects', refers to the task of finding the position of an object at a later time, given a current instantiated position. VOT often focuses on the

At the same time, there has been much recent development in algorithms involving general methods for extracting parts of a video which may be considered objects or regions of interest often fall under the heading of foreground segmentation, background modeling, motion detection, or feature point detection. Schemes attempting to provide an automated method for the detection and tracking of objects or interest regions are varied, though often make use one common method falls under the general heading of blob tracking.

One topic involves algorithms which attempt to perform combined object detection and tracking.

Furthermore, research has focused on performing the above functions on videos with diverse object types, time varying object appearances, diverse numbers of objects, objects which may enter and exit the scene, occlusion of objects, presence of clutter, variable background characteristics, and unpredictable object behaviors.

Details of specific methods in these topics will be covered presently. (?)

MTT and VOT

Discussion of tracking: MTT and VOT. Maybe I could go more into the discussion frank and I had here (instead of opening paragraph). More breadth, less depth on some of the VOT techniques (really read through the Survey on Object Tracking, ie include contour and template stuff).

note: putting this here, not sure where it goes but needs to be hammered out in this intro: MTT (how i currently view it) treats each object as points, but does general tracking which leads to object detection (decides on number, structure, shape, etc. of paths, each of which represents an object over time). Current video detection and tracking schemes do this: at each frame, they try to detect an object (position, rough size/shape–need to define detection as something like this), and use these detections at each frame in object tracking (ranging from representing each detected object as a point and doing MTT methods, or a wide range of similarity-measure schemes of target characteristics/color which i call VOT methods. Also, object detection at further frames often aided by current detection/tracking...but even so, is still a detection at each frame, then tracking viewpoint). [note: this is almost a general tracking problem where each data point is a detected object]. A slightly different paradigm when approaching this problem involves general tracking [i need to define tracking better] which leads to object detection, like MTT but where objects are not considered points but instead as having some spatial shape and other characteristics too (eg color). Sorta new ideas: The goal of this project is to demonstrate a model that provides a stable framework for general unsupervised detection and tracking of multiple objects of arbitrary nature.

General Object Detection

Discussion on foreground/background modeling, feature characteristics of interest-objects, such as feature points and motion points, other general object detection methods (related to detection in video...any purely image-based object detection methods?).

Discussion of combined detection and tracking schemes. Example: blob tracking. The simplest involves detection at each time and then tracking (more particularly: feature extraction [foreground modeling], object detection [segmentation], then object tracking). Some have tried to do this in a different order: generalized tracking over time, which leads to object detection (more particularly: feature extraction [foreground modeling], generalized object tracking [most akin to MTT [as i view it...learn more about MTT before i say this] but for VOT], results complete object detection). Talk about conception benefits of this reverse order over detection then tracking.

Discussion on this project. This project attempts to do the final method presented in the background (general multiple object tracking leading to object detection). Introduce overall goals and benefits of this idea, terms/concepts/methods/models that will be discussed in future sections.

Model Overview

Discuss types of models necessary for this generalized object tracking goal described above (e.g. models that allow for nonparametric decisions as to number of targets, size of targets, time over which targets are tracked/in scene).

Discuss Dirichlet Mixture Model and Dirichlet Process.

Discuss dependent Dirichlet process concept, and time-dependent Dirichlet process mixtures, and gpudpm (gpuddp frank calls it?).

Model Specification and Inference

Experiments

Results

Conclusion

Outline and Terminology

1. Multiple Target Tracking (MTT)
    (a) a.k.a. Point MTT, Multitarget Tracking, Data Association, Point Matching, Plot Association
    (b) Definition: (will include later)
    (c) Multiple Hypothesis Tracking (MHT), Joint Probabilistic Data Association Filter (JPDAF), and their variants and spin-offs
        i. These algorithms may be used to track targets in videos if they can be represented as points.
2. Visual Object Tracking (VOT)
    (a) a.k.a. Video Tracking, Target Tracking, Visual Single Object Tracking, 2D Tracking, 'Tracking of non-rigid objects'
    (b) Definition: (will include later)
    (c) Recursive Bayesian filtering
    (d) Mean-shift methods
    (e) Color tracking
    (f) These can be extended to multiple targets if detection / initialization is given (sometimes involve more sophisticated techniques when multiple targets introduced).
3. Foreground Extraction and Object Detection

(a) a.k.a. Foreground segmentation, background modeling, frame differencing, background subtraction, motion detection, and
a.k.a. Target localization, target representation, target segmentation

(b) Definition: (will include later)

(c) Foreground / Background Modeling

   i. GMM for background / foreground modeling (many studies)

(d) Motion detection, frame differencing, background subtraction

(e) Segmentation

4. Multiple Visual Object Tracking and Detection

   (a) a.k.a Multiple Target Tracking and Localization

   (b) Definition: (will include later)

   (c) blob tracking

      i. foreground detection (motion, thresholding, subtraction, color), segmentation (pixel assignment), (sometimes) centroid placement (localization?), and (sometimes) point MTT.

   (d) clustering methods for automated segmentation and tracking

      i. Pece cluster tracker on motion (time-varying GMM with additions, applied to frame differencing)

      ii. Mean-shift time-dependent clustering (1 paper)

Related Work

The follow sections provide details on work related to the automated detection and tracking of objects. The following numbered sections correspond to the numbers in the outline above.

1. Multiple Target Tracking (MTT)

(a)-(c)

At its most basic, MTT can be approached with the Global Nearest Neighbor (GNN) technique [**?**], where tracks are formed by associating each observation with its nearest neighbor, under the constraint that an observation can be associated with no more than one track. Often, GNN is performed until a problem such as a conflicting track assignment occurs, whereupon a more sophisticated technique is used. As for these more-complex methods, there are two classic MTT algorithms that form a basis for many modern techniques. The first is called multiple hypothesis tracking (MHT) [**???**], in which, for each observation, a set of neighbors is collected as potentials for association; the process is then repeated recursively for each potential neighbor, forming a tree-like structure of possible paths. Once the possible branches for a given path are enumerated, one branch is considered most-likely and is chosen as the true path (where branches may be favored if, for example, they are longer or adhere to some expected behavior of the target). MHT is computationally expensive, but has gained popularity due to increases in computational capabilities [**?**]. The second classic MTT algorithm is known as the Joint Probabilistic Data Association (JPDA) method [**??**]. In contrast to GNN, where the nearest potential neighbor is chosen as the associated position, and MHT, where all potential neighbors are followed until only one remains, JPDA computes a probability that each potential neighbor (that pass an initial validation test) is the next position of the current track, and predicts the next position as a probability-weighted combination of all potential positions. In this way, JPDA acts similar to another MTT technique known as Recursive Bayesian Filtering (RBF), where the position of a target at a future step is predicted, and this prediction allows for subsequent points to be associated to a track with higher accuracy. Techniques involving RBF often make use of a Kalman or particle filter, and these methods are often incorporated with the other algorithms mentioned in this section. Pulford gives a thorough overview of 'classic' and 'modern' data association algorithms (and gives the details of 35 different MTT algorithms) [**?**]. Taking a different approach to estimating the number targets, Fox et al. uses a Dirichlet Process prior to estimate the number of targets in an MTT scheme [**?**].

2. Visual Object Tracking (VOT)

(a)-(b)

Given the position of a target in the current frame of a video, VOT is the task of predicting the target's position in the following frame. VOT is not concerned with detecting new targets, but instead with maintaining the location of an object after its initial position has been specified. VOT is also sometimes used to refer to the task of maintaining the shape or size of a target over time, in addition to its position. Most VOT schemes operate by considering the region that represents a target in a given frame of a video, and looking in the vicinity of this region in a subsequent frame for the most similar region. Areas of research are primarily concerned with ways to represent the appearance of targets, predict future positions of targets, accurately or quickly find similar regions in successive frames, and handle troublesome targets (for example, targets which may appear, dissapear, or have time-varying characteristics) [**?**].

(c)

Recursive Bayesian filtering involves the task of estimating the position of a target from noisy measurements. (This section is not finished. Need to discuss numerous methods which use particle filtering techniques for object tracking. Are these techniques even very similar / under the same umbrella?)

(d)

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 [**??**]–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 [?]. The mean shift algorithm has been implemented successfully to allow for a sort of kernel-based object tracking [???]. 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 [?], with a kernel adjusted by the estimated centroid of a tracked target [?], and with a heirarchical version of the mean-shift procedure [?].

(e)

Color has long been used to track non-rigid objects; again, 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 [???]. 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 [?]. Additionally, color has been successfully applied as the feature in the mean-shift procedure (kernel-based method) for tracking [????].

(f)

VOT schemes are easily extendable 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 [?]. 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) [??] .

3. Foreground Extraction and Object Detection

(a)-(b)

The following methods are not concerned with tracking, but instead with the detection of and differentiation between individual objects in a given frame (or over a sequence of frames) of a video. Usually, objects of interest for tracking are in the foreground of a scene; as such many of the following techniques involve attempts to separate the foreground of a scene from the background. Another issue is that of segmentation, or deciding which pixels are associated with which object; this is also what determines the number of objects present in a scene.

(c)

Foreground extraction, foreground segmentation, foreground detection, and background modeling all refer to the task of deducing which pixels represent foreground objects (i.e. objects likely to be desired for tracking), and which represent background objects. Foreground extraction, though possible to carry out by following certain heuristics involving temporal changes in pixel value characteristics (see section (d)), is often achieved, instead, using model-oriented techniques. 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 [**???**]. 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 [**?**]. 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) [**??**].

(d)

Frame differencing and background subtraction 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 data extraction method for object tracking schemes with success [**???**], and also as a secondary data extraction method to help improve the accuracy of object tracking schemes [**?**]. 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) [**??**]. In [**?**], 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.

(e)

Target localization, target representation, and target segmentation all relate to the task of finding the position of objects in a scene. In the context of the foreground extraction described previously, the process of segmenting foreground pixels into distinct foreground objects allows for object localization. Segmentation can be very simple: the watershed transform, mean-shift procedure, and other basic clustering algorithms have been used directly on foreground extracted pixels to accomplish segmentation [**?**].

4. Multiple Visual Object Tracking and Detection

(a)-(b)

There are certain methods which attempt to combine object detection and VOT in order to provide a start-to-finish technique for temporal object detection (detecting a given object throughout its time in a scene). Some methods involve foreground extraction and segmentation as initialization for VOT, which results in the tracking of all objects present at the beginning of a scene, while others attempt more sophisticated methods in order to handle scenes where objects enter and exit throughout.

(c)

Blob tracking often refers to the task of foreground extraction (in blob trackers this is often done by extracting pixels whose value is above some threshold, which only works for videos where foreground objects have pixel values that are very dissimilar to those of background objects), segmentation of foreground pixels, and MTT, where each 'blob' created during segmentation is treated as a point particle. More sophisticated blob tracking may involve statistical appearance models for robust foreground extraction, such as those described in 3.(c), which allow for background subtraction in cases where the foreground and background pixels have similar colors or where the characteristics of the foreground targets show strong temporal dependence [**???**]. Blob trackers have even incorporated the ability to estimate the correct number of targets in a scene and introduce recursive Bayesian filtering for improved VOT [**?**].

(d)

Clustering based object tracking and detection schemes typically attempt to either detect objects or simultaneously detect and track objects using minimal, easily extracted, or general information about a scene and the targets within it. These techniques often start with foreground (e.g. movement) pixels, color data from each frame in a video, or so called 'feature points' (described in the following section) and attempt to automatically perform segmentation and tracking with temporal clustering (where clusters are somehow present at or shared between multiple frames). This differs from blob-based tracking approaches, where a pixel is assigned to the foreground or to a specific, segmented blob (each of which represents a target) at each frame.

i In [**?**], foreground positions are found via frame differencing and modeled with a sequence of GMMs. Pece provides criteria for eliminating, merging, and estimating the correct number of clusters. Additionally, he uses simple heuristics for the initialization of clusters. Tracking is carried out on a dataset consisting of varied objects, and this method is able to track the various types of objects, though fails when objects become too close in space.

ii The mean-shift procedure was used by **?** to perform clustering on foreground pixels extracted via frame-differencing. The authors promote this technique over that in [**?**], and say that it has a better ability to keep nearby density maxima (in the difference pixels data) separate.