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

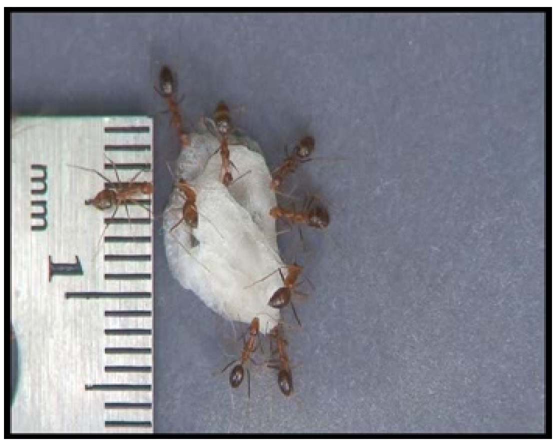
Statistical Machine Learning for Computer Vision

Willie Neiswanger

Object Detection and Tracking

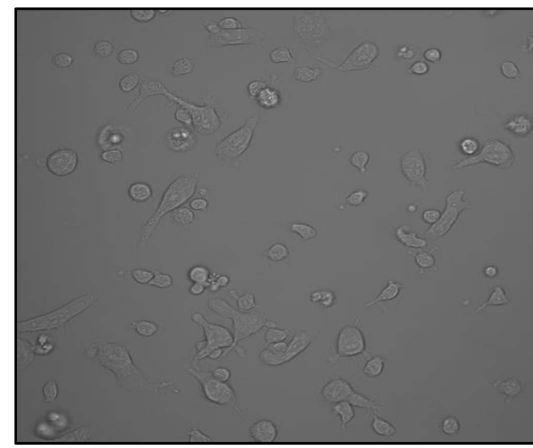
A task in the field of computer vision

Zoology / Ecology



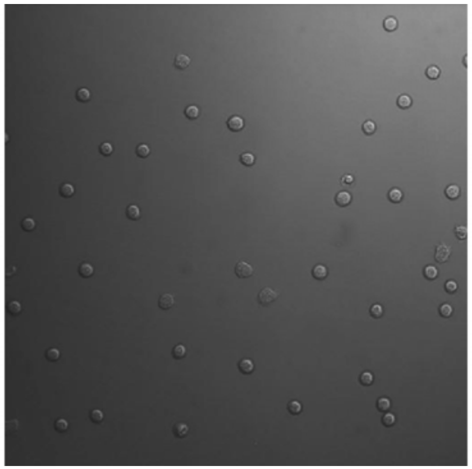
Surveillance

Cell Biology



Also: Behavioral psychology, robotics, and video summarization

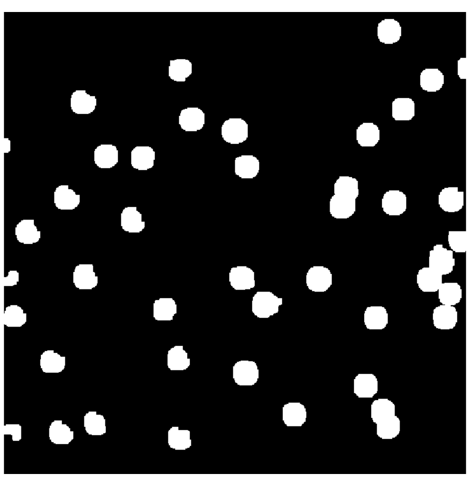
Object Detection and Tracking



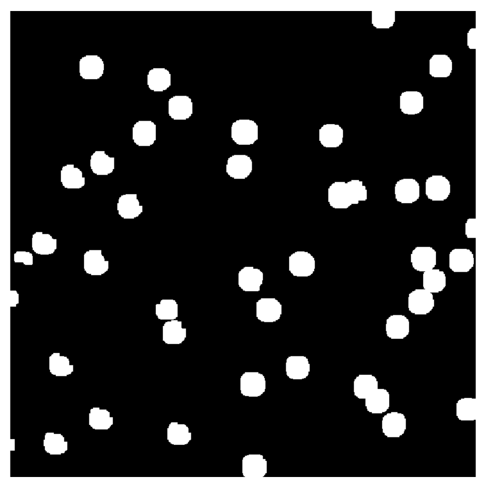
Typical methods consist of :

1. - Object Detection
2. - Object Tracking

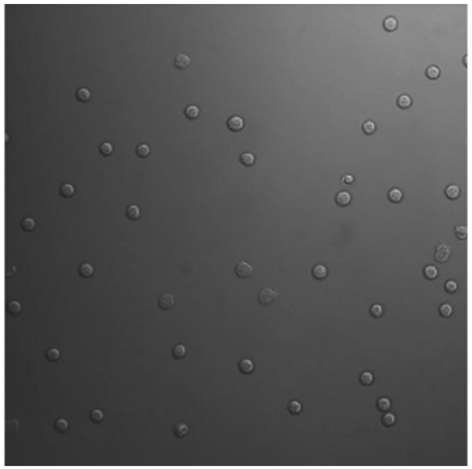
1.



2.



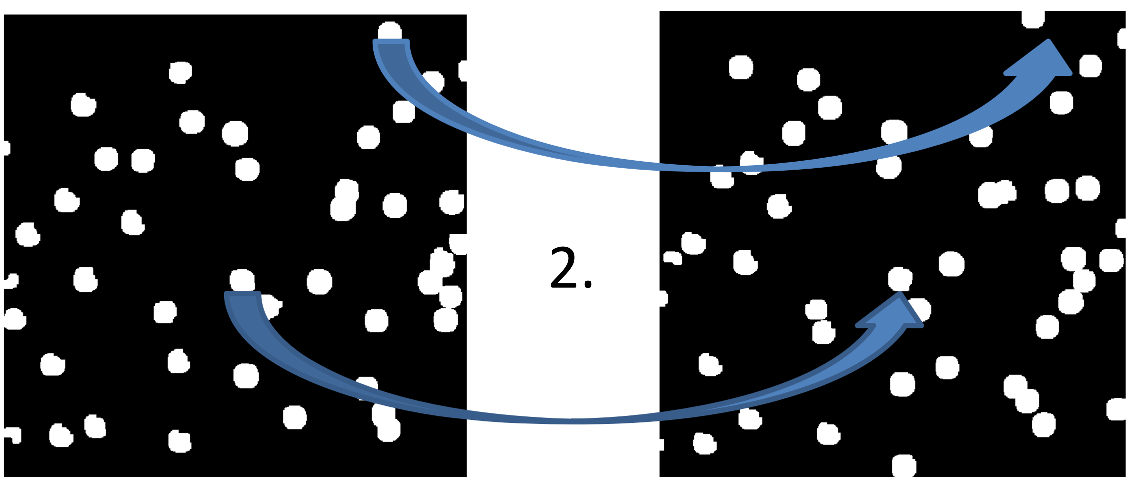
Object Detection and Tracking



Typical methods consist of :

1. - Object Detection
2. - Object Tracking

1.



Object Detection and Tracking

Object detection needed for:

- Segmentation / localization
- Counting number of objects
- Initialization of tracking
- Tracking
- Finding entering / exiting objects

Difficulties in detection

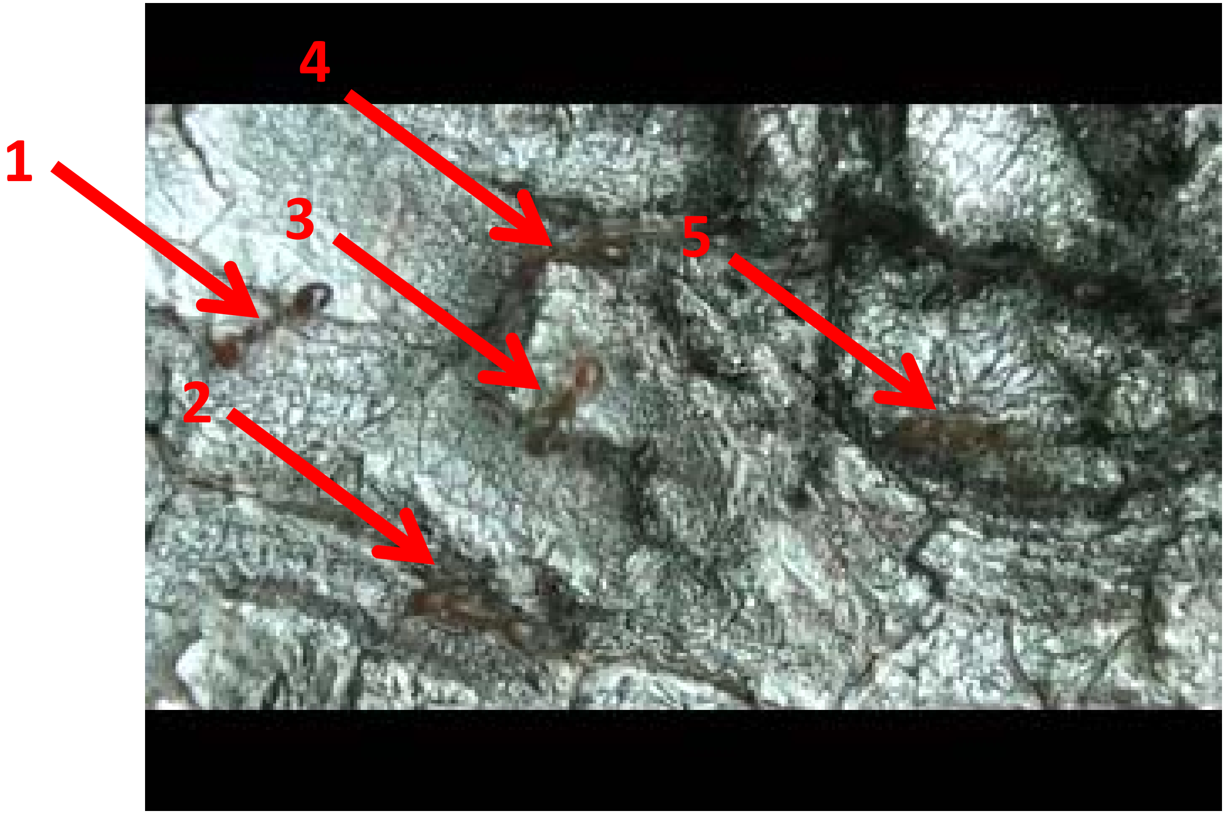
- Need ad hoc method for each object type
- Complex backgrounds and film qualities
- Spurious / missing objects

Example: Ants

The ants in this image are difficult to detect.

Example: Ants

The ants in this image are difficult to detect.



Project Goal

To develop a general, unsupervised object detection and tracking method that can be applied to videos with arbitrary objects, backgrounds, and film qualities.

We propose:

1. Extract data “sloppily” from videos.
2. Use a sophisticated model to perform detection and tracking

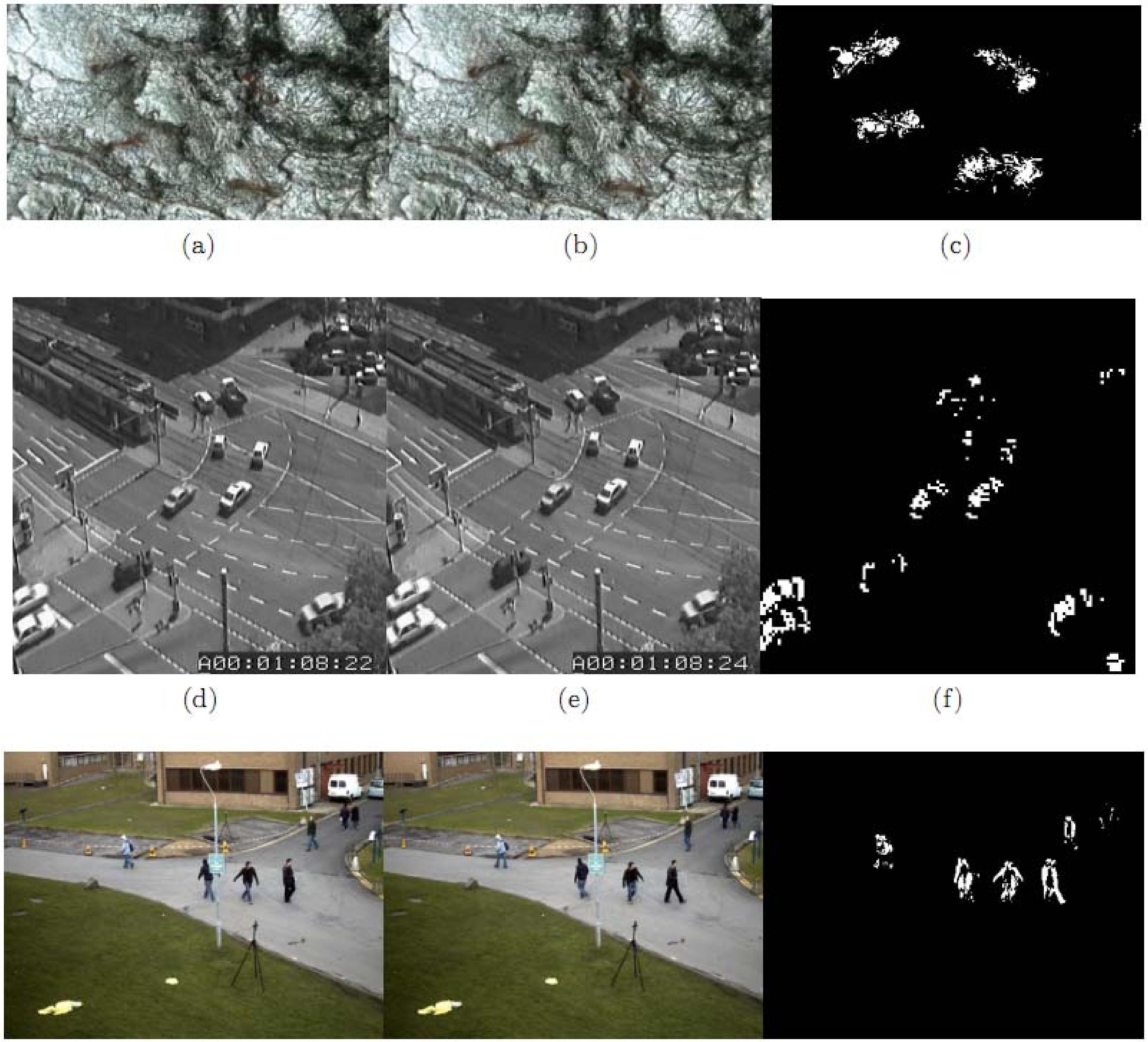
Extraction Examples

Extraction:

- There exist methods for gathering data from foreground objects

- For example, these methods involve:
  - Motion (pixel difference, optical flow)
  - Foreground / background modeling
  - Feature points / texture

Extraction Examples



(a) (b) (c)

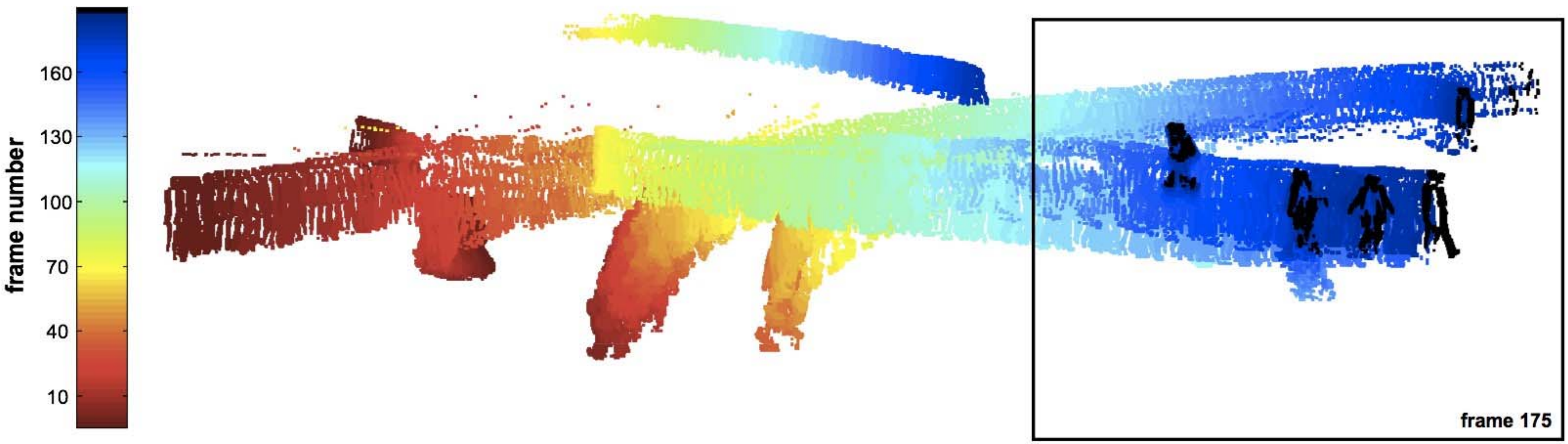
(d) (e) (f)

(g) (h) (i)

E.g.: Frame differencing shown to the left

From each moving pixel, we extract its position and “color counts” from surrounding pixels.

Extraction Examples



Model

We aim to cluster extracted data:

- *Clustering*: an unsupervised learning method to partition a dataset.

- We want our clusters to correspond to distinct video objects

We will use a mixture model to perform clustering:

- *Mixture model*: a probability distribution with components ("clusters") that can be viewed to represent subpopulations of a dataset.

- Clustering is accomplished when we carry out inference algorithms that assign data points to clusters.

Model

Basic intuitions behind a few mixture model based clustering methods:

1. Assume data are samples from a parameterized mixture model (a family of functions) and estimate parameters of the model to find clusters.

2. Assume prior distributions over parameters of the mixture model, and, given the data, use Bayesian inference to infer a posterior distribution over the parameters.

Model

We're going one farther:

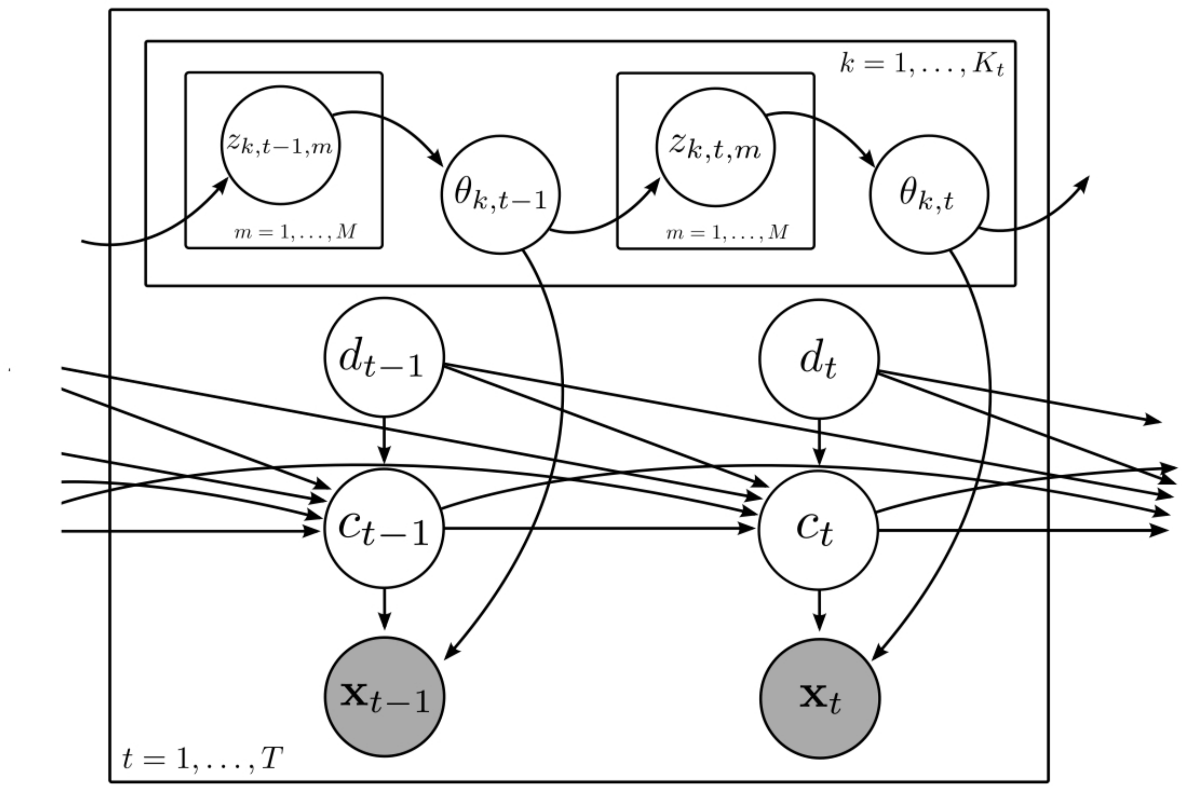
3. Do a "nonparametric" version of the previous mixture model (called: a Bayesian nonparametric model).

Nonparametric model: model structure is not "fixed", can grow in size or complexity based on the data.

In our case: our model allows for different numbers of clusters, so we can use on videos with however many objects.

Model

You will learn lots about graphical models, but here is a taster:



Model

One more thing to note:

- the parametric form of the clusters
  (i.e. of the components in the mixture model).

In our mixture model, each cluster has the form:

$$P(\mathbf{x}|\theta) = \mathcal{N}(\mathbf{x}^s|\boldsymbol{\mu}, \Sigma)\mathcal{M}n(\mathbf{x}^c|\mathbf{p})$$

So, at each frame, subpopulations of extraction data are represented by:

- a normal distribution over position features
- a multinomial distribution over color features

Bayesian Inference

Bayesian inference used to infer distribution over cluster parameters (and thus cluster the data).

We use an inference algorithm that involves "sampling" or "random number sampling"

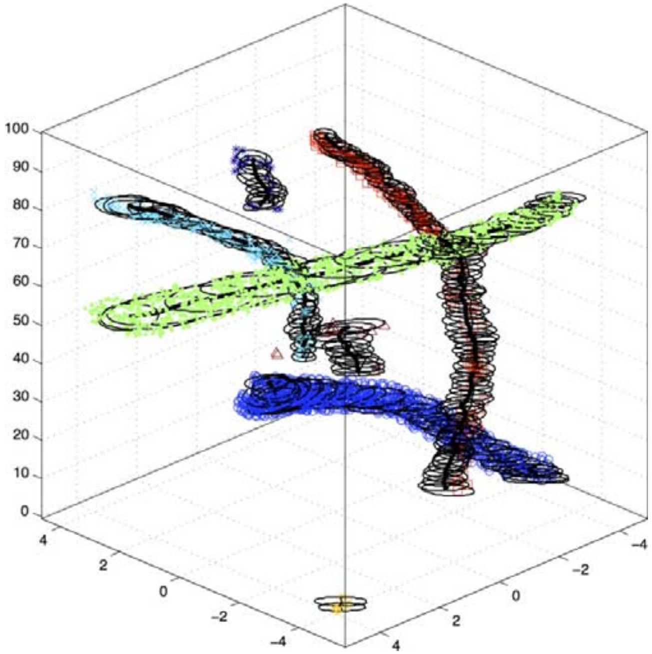
- *Sampling*: generate pseudo-random numbers distributed according to a specified distribution.

In our case: we use sampling to generate samples distributed according to the posterior distribution over our model.

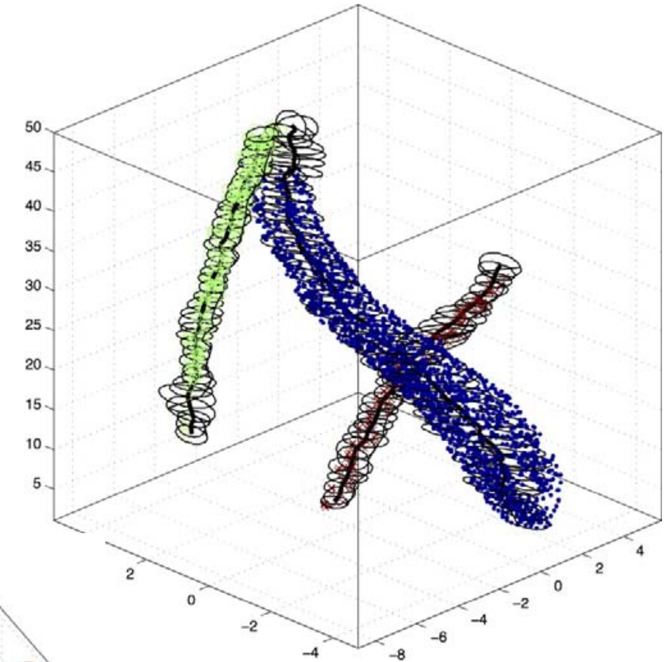
See Sampler-In-Action Video

via my Youtube page:
www.columbia.edu/~wdn2101/myresearch.html

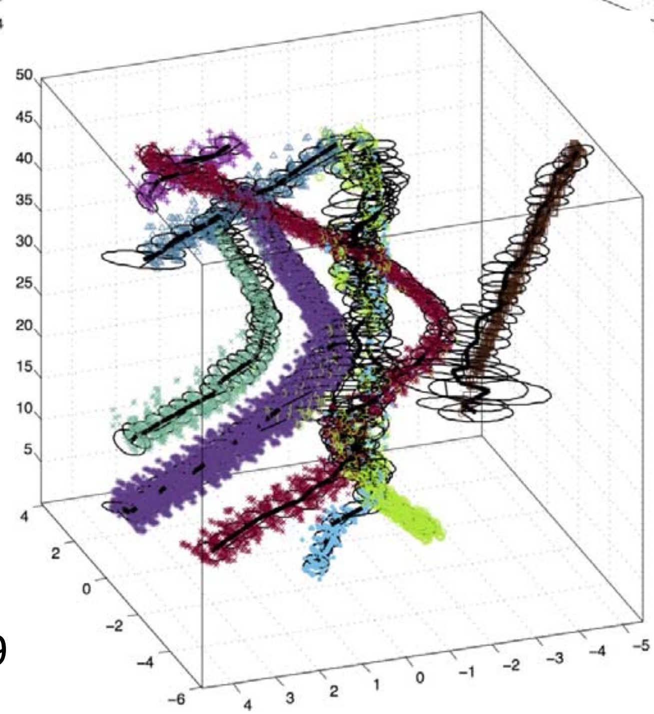
Results



PETS2000



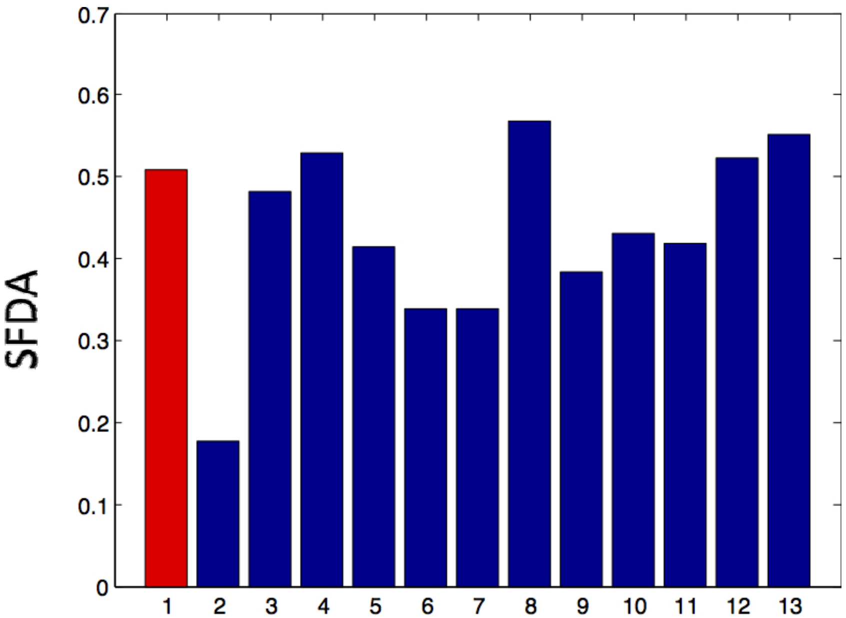
PETS2001

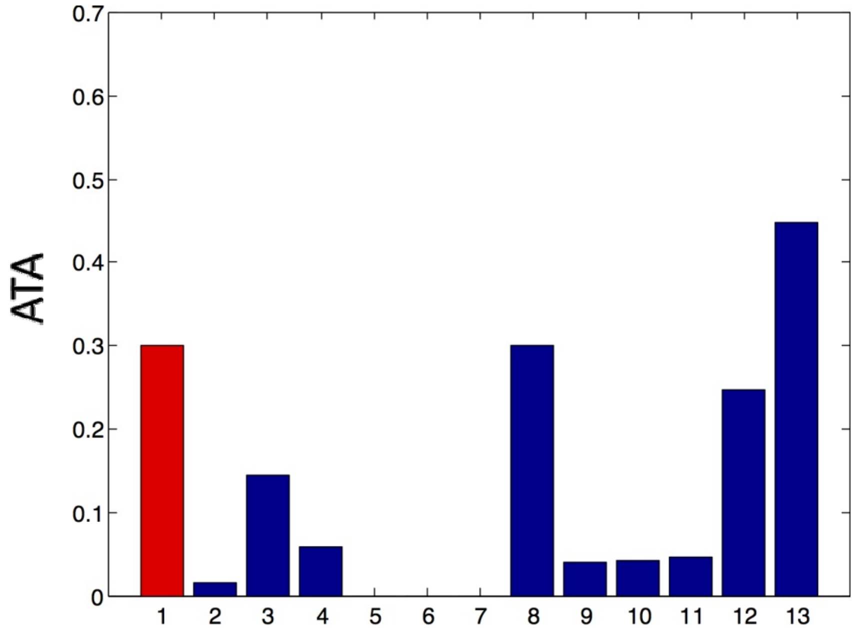


PETS2009

Performance Results

We can compute standard object detection/tracking metrics and compare our performance with state of the art object-specific trackers.





Ours is the red bar; our general method is comparable with these object-specific trackers.