# Combining Discriminative Features to Infer Complex Trajectories

David Ross, Simon Osindero, Richard Zemel
University of Toronto, Machine Learning

ICML 2006
June 28, 2006

# Inferring Complex Trajectories

- Problem: time series regression
  - Estimate continuous state variables from a sequence of observations
- Running example: Visual tracking of a target object
- Standard approach: Generative state-space model (Kalman filter, etc.)
  - strong likelihood, generates observations
  - weak prior, describes trajectory

# Combining Discriminative Features

- Discriminative conditional model
- Model Pr(state|observations) as a (log-linear) combination of dynamics & observation features
- “Pile on” features
  - Learn which are useful
  - Switch features on and off dynamically

# Discriminative Features

- Dynamics features: $f_j(\mathbf{x}_{t-1}, \mathbf{x}_t)$
  - how well do two states match?
  - (non) linear dynamical models
- Observation features: $g_k(\mathbf{x}_t, \mathbf{Y}, t)$
  - is the target at $x_t$?
  - Any appearance model/object detector
  - Can include information from the entire observation sequence
- Robustify by switching features on and off
  - Hidden switch variables $u_{jt}$ $v_{kt}$

# Features & Switch Potentials

- Weighted distance between state and prediction

$$f_j(\mathbf{x}_{t-1}, \mathbf{x}_t) = -\frac{1}{2}\left(\mathbf{x}_t - \phi_j(\mathbf{x}_{t-1})\right)^T \boldsymbol{\alpha}_j \left(\mathbf{x}_t - \phi_j(\mathbf{x}_{t-1})\right)$$

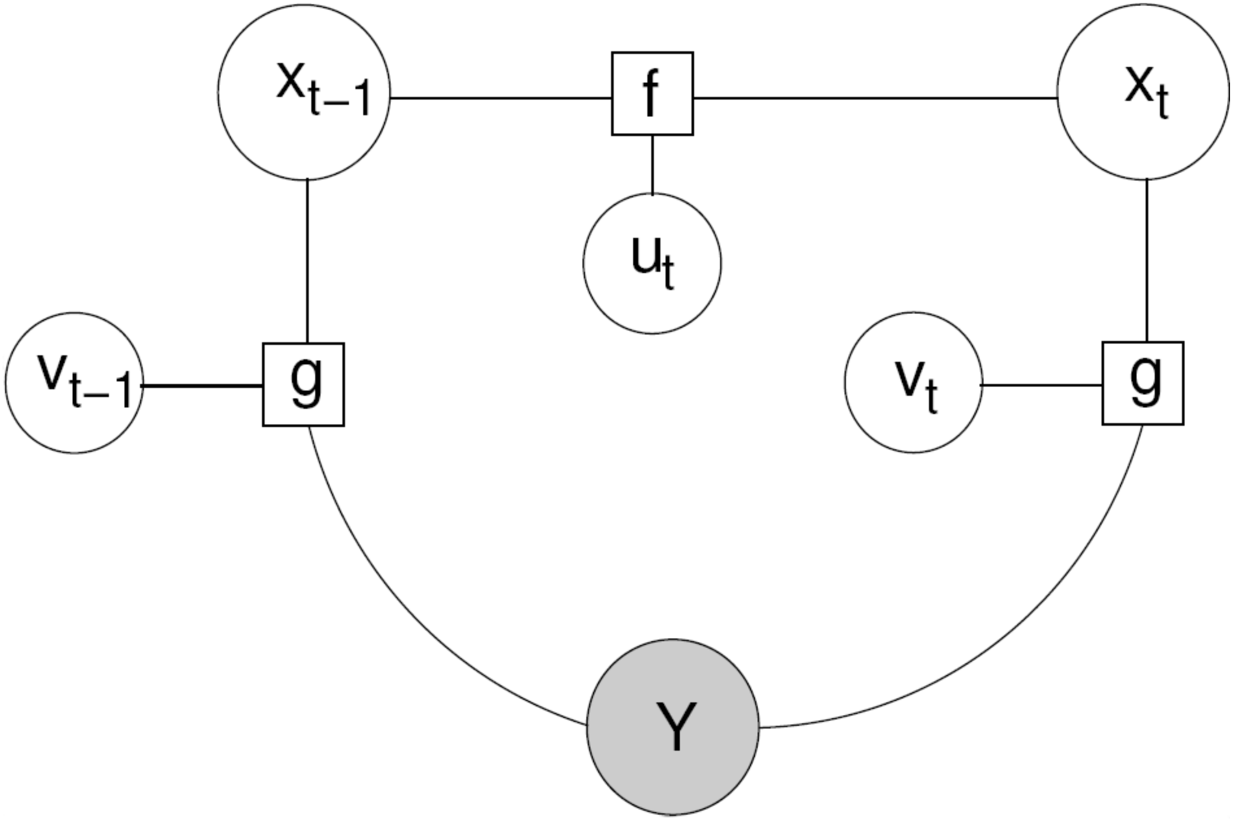
$$g_k(\mathbf{x}_t, \mathbf{Y}) = -\frac{1}{2}\left(\mathbf{x}_t - \gamma_k(\mathbf{Y}, t)\right)^T \boldsymbol{\beta}_k \left(\mathbf{x}_t - \gamma_k(\mathbf{Y}, t)\right)$$

$$\phi_j(\mathbf{x}_{t-1}) = \mathbf{T}_j \mathbf{x}_{t-1} + \mathbf{d}_j$$

- Switch Potentials: extra features help decide if switches should be on or off
- Any classifier (logistic / softmax regression)

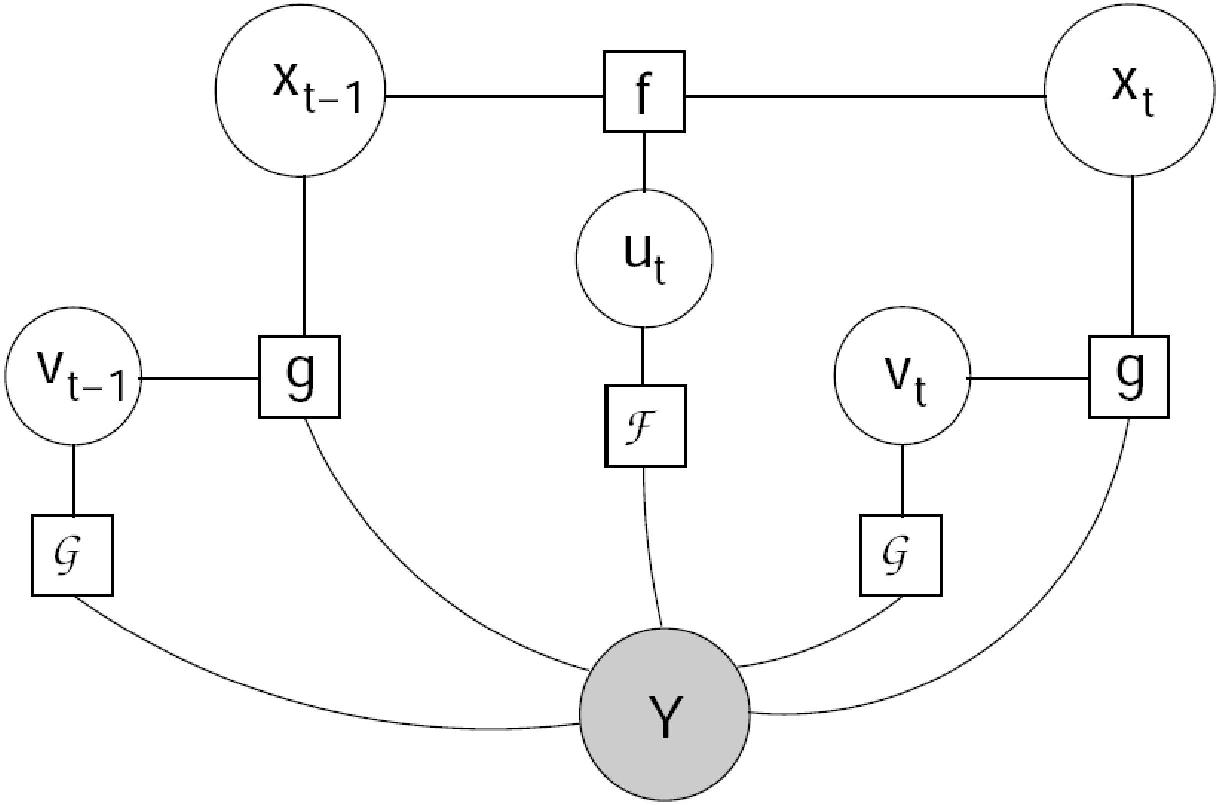
# Probability Model

$$\mathrm{P}(\mathbf{X}|\mathbf{Y}) \propto \exp\left(\sum_{t,j} f_j(\mathbf{x}_{t-1},\mathbf{x}_t)u_{j,t} + \sum_{t,k} g_k(\mathbf{x}_t,\mathbf{Y})v_{k,t}\right)$$



# Probability Model

$$\mathrm{P}(\mathbf{X}|\mathbf{Y}) \propto \exp\left( \sum_{t,j} f_j(\mathbf{x}_{t-1}, \mathbf{x}_t) u_{j,t} + \sum_{t,k} g_k(\mathbf{x}_t, \mathbf{Y}) v_{k,t} + \sum_{t,j} \mathcal{F}_j(\mathbf{Y}, t) u_{j,t} + \sum_{t,k} \mathcal{G}_k(\mathbf{Y}, t) v_{k,t} \right)$$



# Inference

- P(X|Y) is hard
- P(X|U,V,Y) and P(U,V|X,Y) are easy
- Infer state sequence using belief propagation
- Sample switch probabilities:

$$\mathrm{P}(v_{kt} = 1) = \sigma\left(g_k(\mathbf{x}_t, \mathbf{Y}) + \mathcal{G}_k(\mathbf{Y}, t)\right)$$

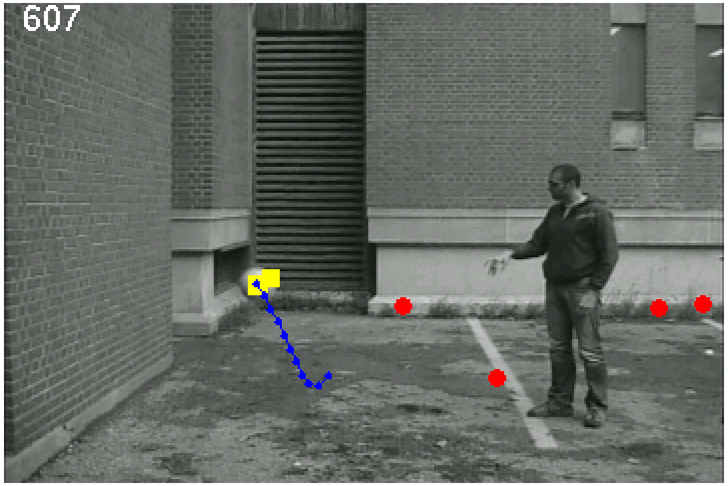
$$\mathrm{P}(u_{jt} = 1) = \frac{\exp(f_j(\mathbf{x}_{t-1}, \mathbf{x}_t) + \mathcal{F}_j(\mathbf{Y}, t))}{\sum_{j'} \exp(f_{j'}(\mathbf{x}_{t-1}, \mathbf{x}_t) + \mathcal{F}_{j'}(\mathbf{Y}, t))}$$

# Learning

- Separable into several easy sub-problems
- learn each observation and dynamics feature separately
- learning switch potential parameters for each feature (classification problem)
- jointly learn feature precisions (weights), using Contrastive Divergence

# Application: Tracking in Video

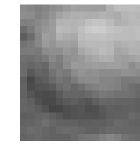
- Combine unreliable dyn/obs features
- 6d state (position, velocity, acceleration)
- Linear dynamics features
- Observation features predict (x,y) position
- Train: video labeled with ground truth



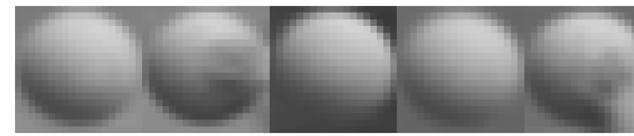

- expected ball position (last 10 frames)
- observation features
  - switched on
  - switched off

# Features Used

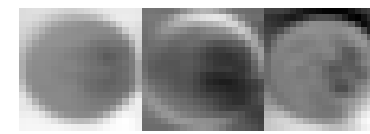
- Template from one frame



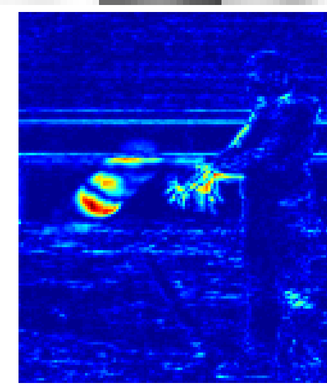
- Template from K-Means



- PCA, 3 components

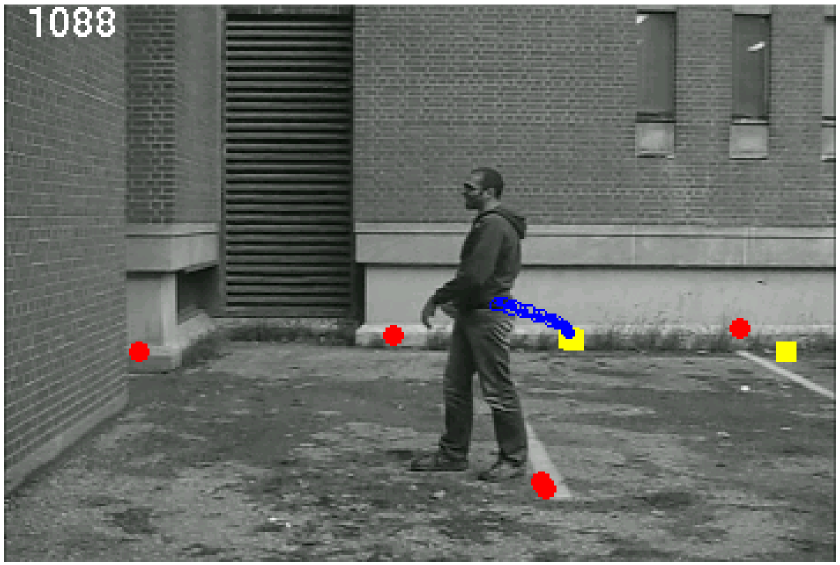


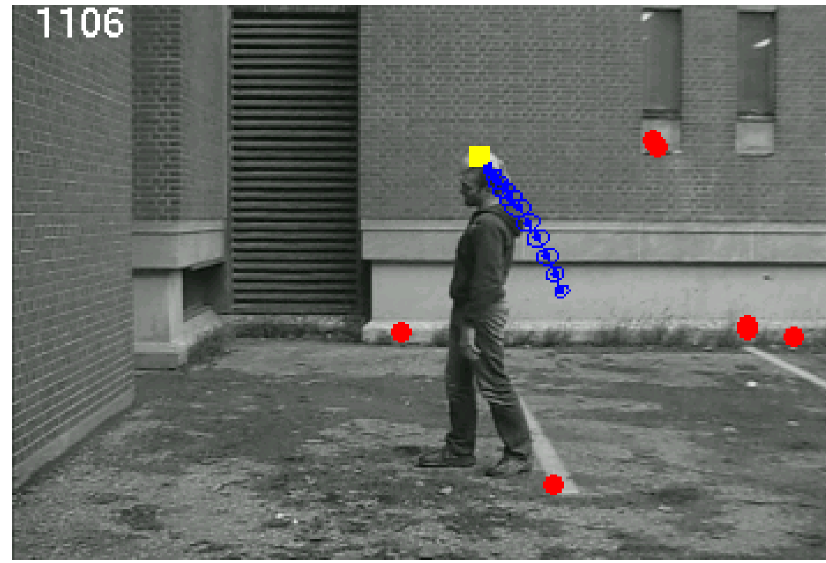
- Local background subtraction

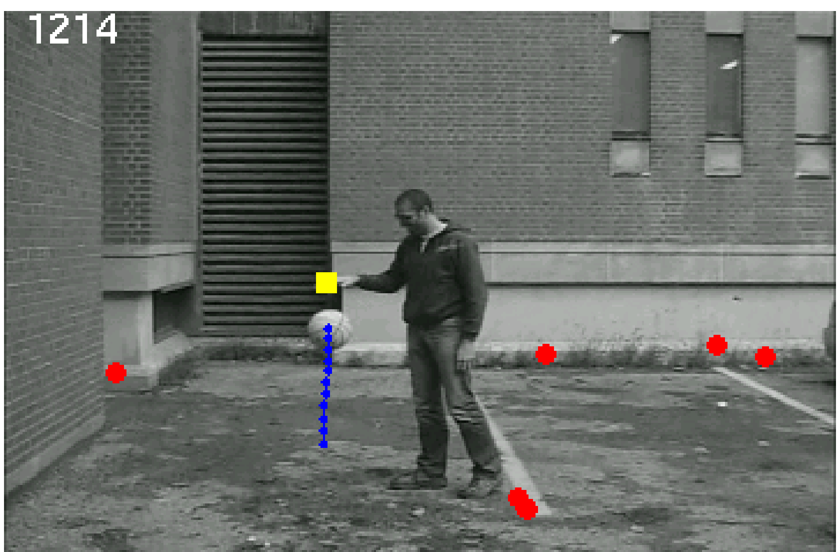


- 4 Linear dynamics (fly, hold, bounce:ground, bounce:wall)

# Within Sequence Generalization



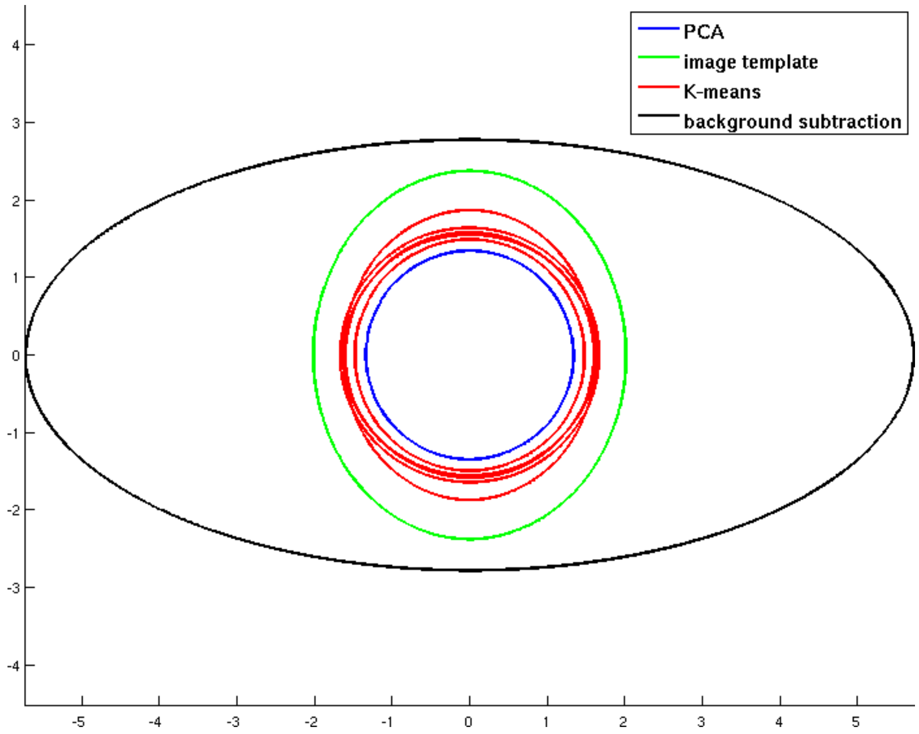

- expected ball position (last 10 frames)
- observation features
  - switched on
  - switched off

# How do other trackers do?

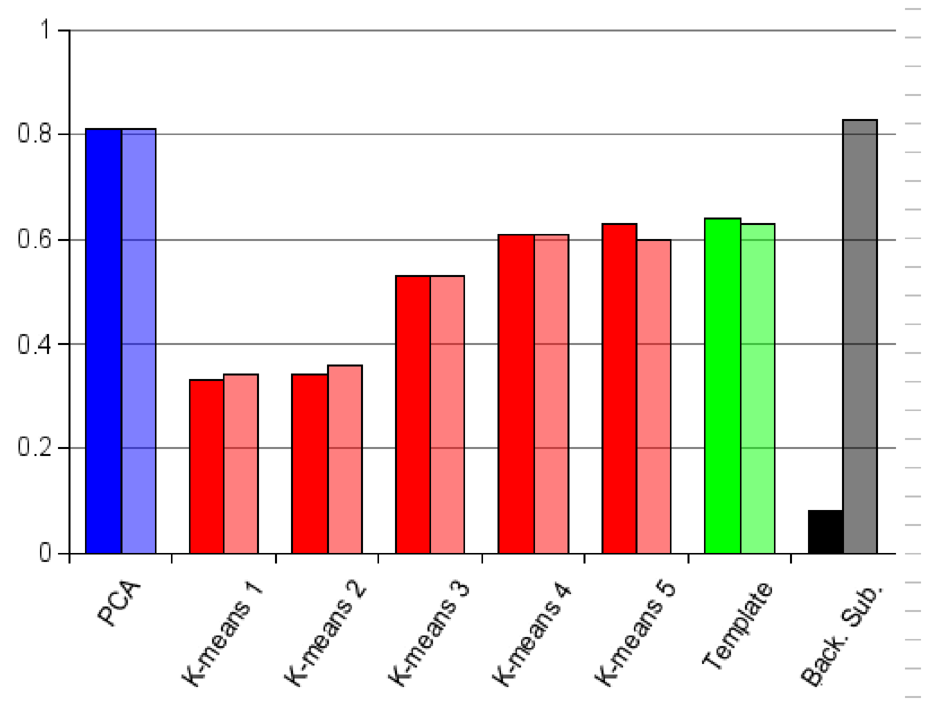
- error rate = fraction of frames where predicted state is > 5 pixels away from ground truth
- Kalman Filter er=0.73
- Adaptive Eigentracker+particle filter fails at frame 688, can't recover er=0.61
- THIS MODEL er=0.11

# How is it working?

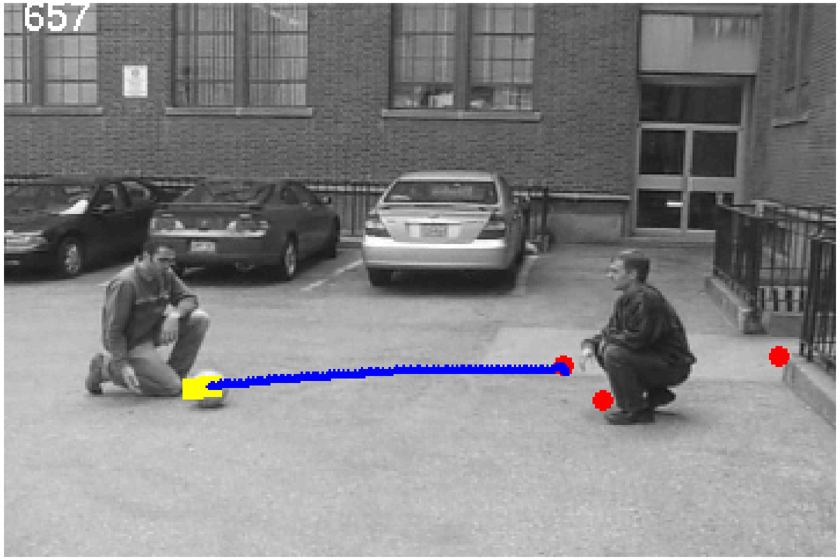
standard deviation

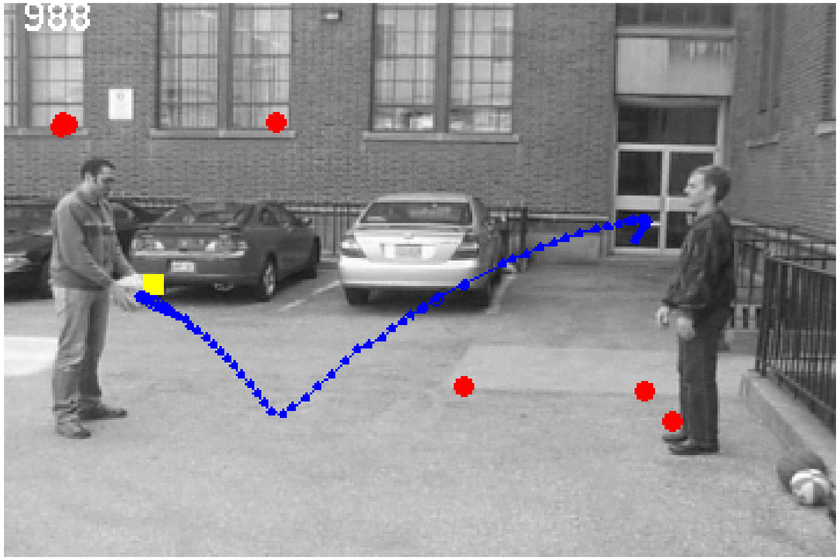


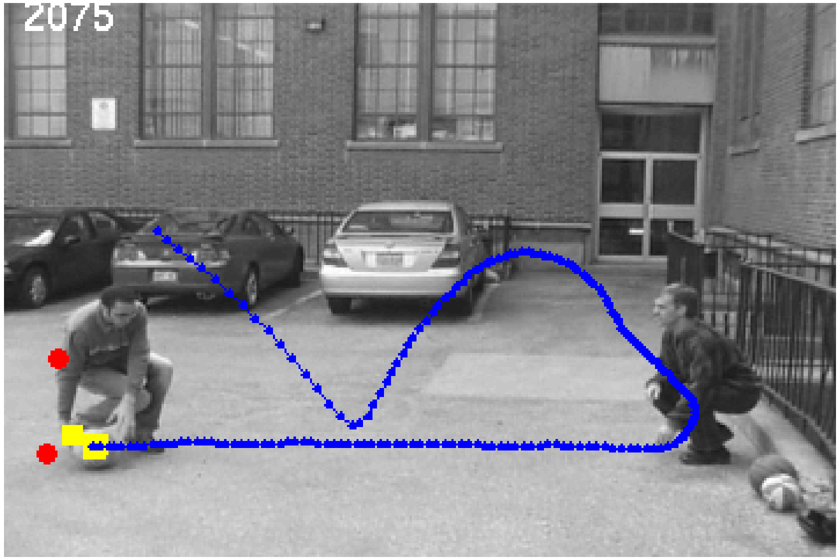

“on the ball”



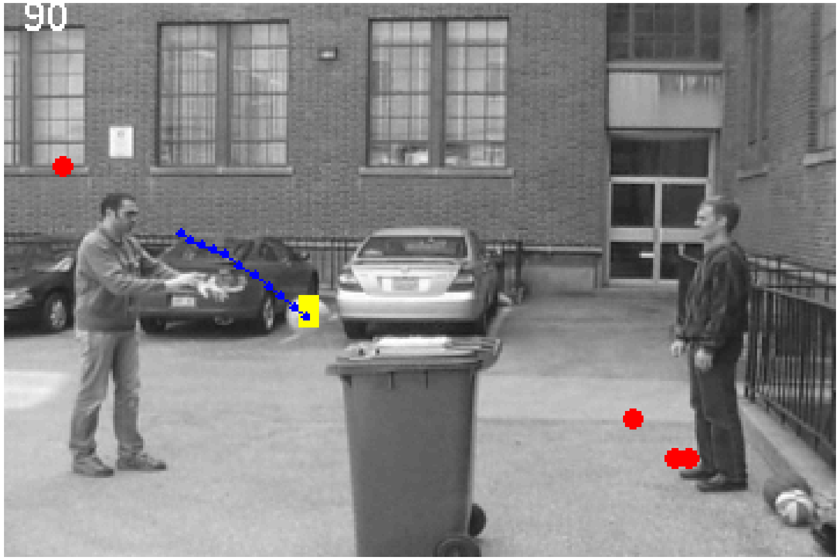
# Test #2: Rolling & Bouncing



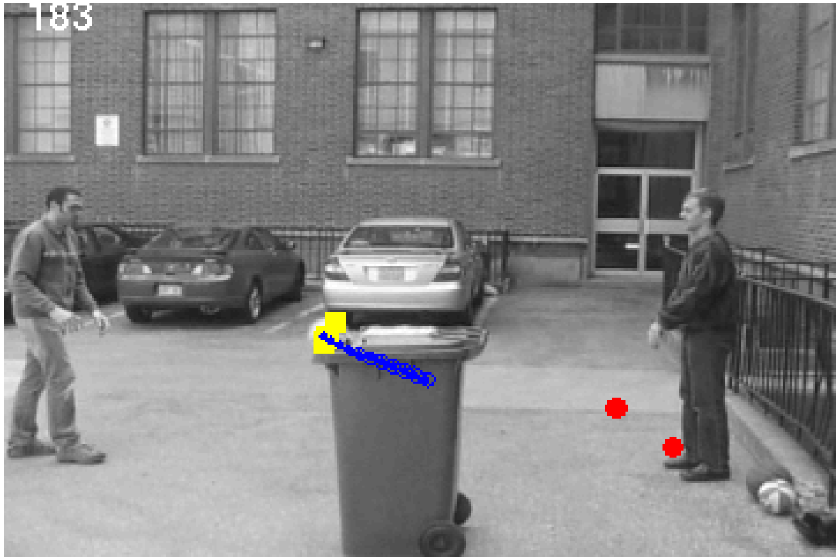



# Test #3: Total Occlusion







# Related Work

- Conditional Random Fields
- Switching linear dynamical systems (and particle filters)
- Discriminative Trackers (Sminchisescu)
- Unsupervised Product Models (Gehler)

# Future Directions

- Expanding the range of features
  - Optic flow, SIFT, State-of-the-art trackers
  - Different modalities (e.g. sound)
- Unsupervised learning of dynamics
- Other applications (financial time-series, robot localization, any suggestions?)