Object Tracking, Trajectory Analysis and Event Detection in Intelligent Video Systems

Student: Hsu-Yung Cheng
Advisor: Jenq-Neng Hwang, Professor
Department of Electrical Engineering
University of Washington

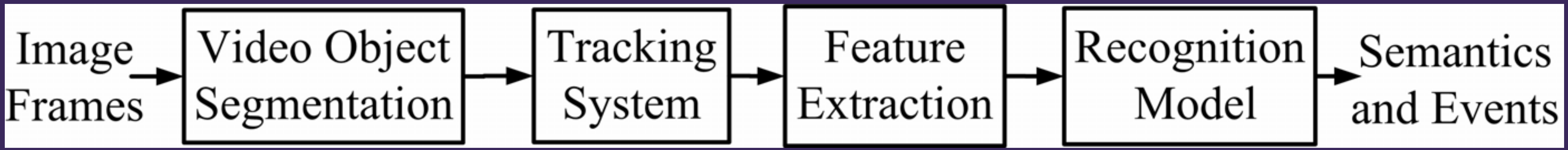
Outlines

- Motivation
- Object Tracking
- Trajectory Analysis
- Event Detection
- Conclusions and Future Work

Motivation

- Advantage of Video-based systems
  - Being able to capture a large variety of information
  - Relatively inexpensive
  - Easier to install, operate, and maintain
- Applications
  - Security surveillance
  - Home care surveillance
  - Intelligent transportation systems
- There is an urgent need for **intelligent video systems** to replace human operators to monitor the areas under surveillance.

System Modules for Intelligent Event Detection Systems



Challenges for Robust Tracking

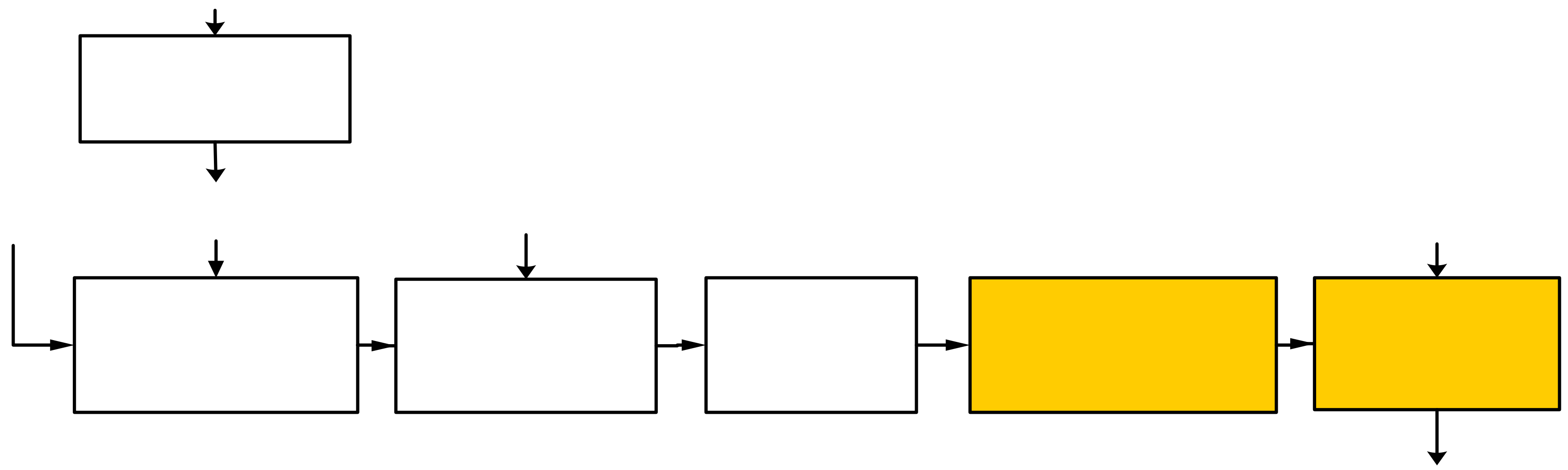
- Segmentation errors
- Change of lighting conditions
- Shadows
- Occlusion

Inter-Object Occlusion

Initial Occlusion

Background Occlusion

Proposed Tracking Mechanism



Background Estimation and Updating

- Based on **Gaussian mixture models** [Stauffer 1999]
- Model the recent history of each pixel by a mixture of K Gaussian distributions.
- Every pixel value is checked among the existing K Gaussian distributions for a match.
- Update the weights for the K distributions and the parameters of the matched distribution
- The k^{th} Gaussian is ranked by w_k / σ_k ($\Sigma_k = \sigma_k{}^2 I$)
- The top-ranked Gaussians are selected as the background models.
- Pixel values that belong to background models are accumulated and averaged as the background image.
- The background image is updated for every certain interval of time.

Moving Object Segmentation

- Based on background subtraction
- Fourth order moment

[S. Colonnese et al. *Proc. of SPIE 2003*]

$$\mu_d^{(4)}(x,y) = \frac{1}{N_\eta} \sum_{(s,t)\in\eta(x,y)} (diff\_img(s,t) - \hat{m}_d)^4$$

- Thresholding

$$S(x,y) = \begin{cases} 1, & if \quad \mu_d^{(4)}(x,y) \geq \theta \\ 0, & if \quad \mu_d^{(4)}(x,y) < \theta \end{cases}$$

Kalman Filter

- Kalman filters are modeled on a Markov chain built on linear operators perturbed by Gaussian noises.
- At time *k*, each target has state x_k

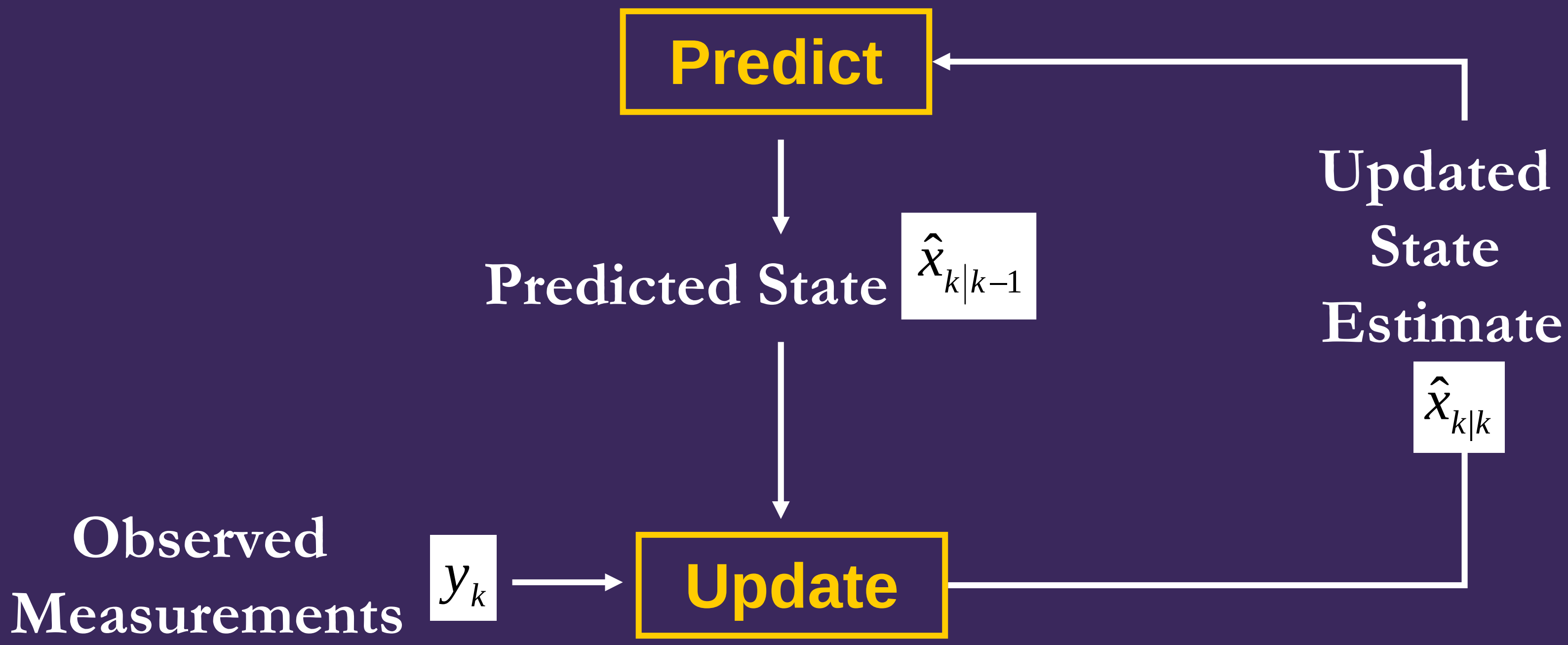
  $x_k = F_k x_{k-1} + w_k$, where $w_k \sim \mathrm{N}(0, Q_k)$

  and observation (measurement) y_k

  $y_k = H_k x_k + v_k$, where $v_k \sim \mathrm{N}(0, R_k)$

Kalman, R. E. "A New Approach to Linear Filtering and Prediction Problems,“ *Transactions of the ASME - Journal of Basic Engineering* Vol. 82: pp. 35-45, 1960.

Kalman Filter Phases



$$x_k = y_k = [u_k \; v_k \; \dot{u}_k \; \dot{v}_k]^T$$

$$F_k = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Kalman Filter Phases

Predict Phase

- Predicted State

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1}$$

- Predicted Estimate Covariance

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k$$

Update Phase

- Updated State Estimate

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k$$

- Updated Estimate Covariance

$$P_{k|k} = (I - K_k H_k) P_{k|k-1}$$

- Kalman Gain

$$K_k = P_{k|k-1} H_k^T S_k^{-1}$$

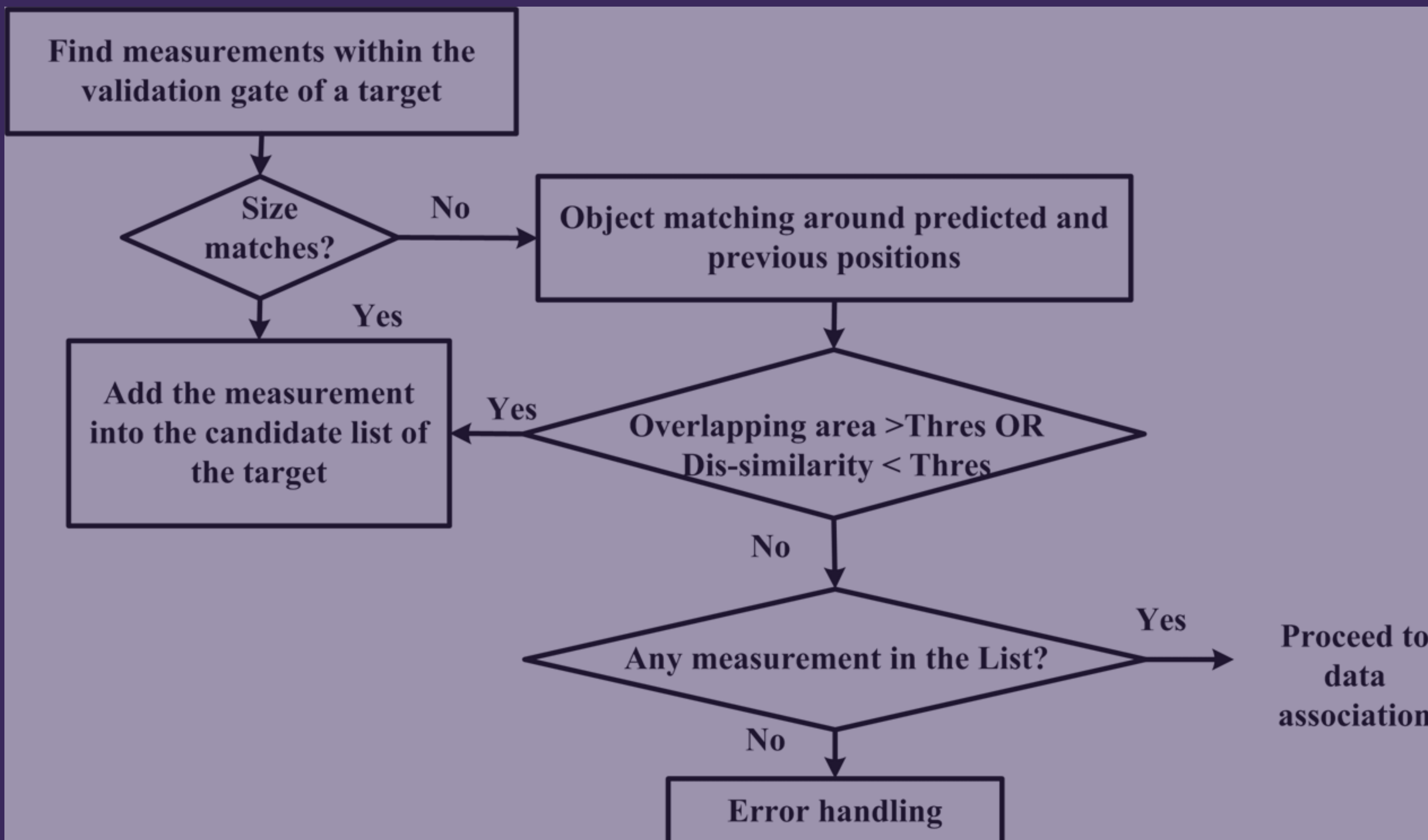
- Innovation Covariance

$$S_k = H_k P_{k|k-1} H_k^T + R_k$$

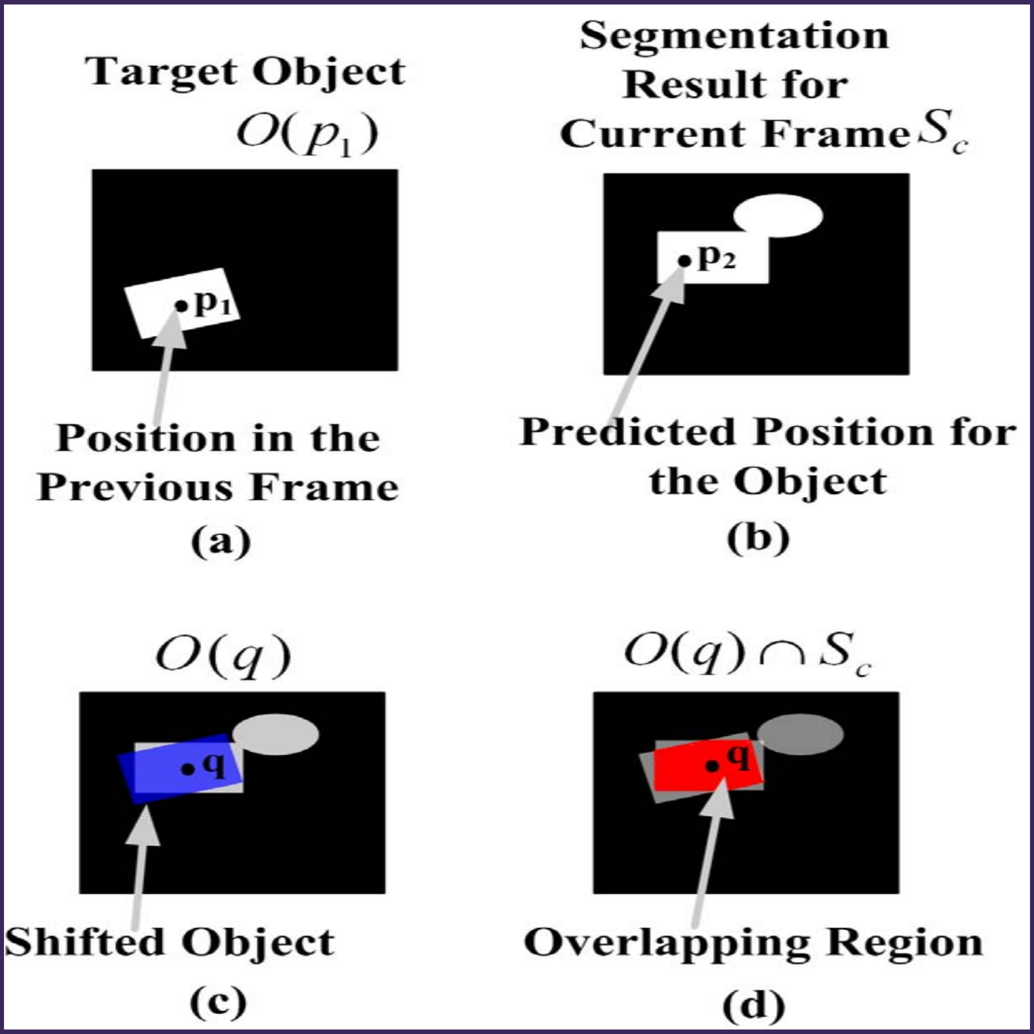
- Innovation (Measurement) Residual

$$\tilde{y}_k = y_k - H_k \hat{x}_{k|k-1}$$

Constructing Measurement Candidate List



Searching for measurement candidate representation points



- Search for q_1 and q_2 in the two n x n windows centered around p_1 and p_2, respectively.

$$q_1 = \arg \max_{q \in \eta_1(x,y)} Area(O(q) \cap S_c)$$

$$q_2 = \arg \max_{q \in \eta_2(x,y)} Area(O(q) \cap S_c)$$

- Compute the dissimilarities between the target object and the potential measurement candidates.

Data Association

- To associate measurements with targets when performing updates
- Nearest Neighbor Data Association

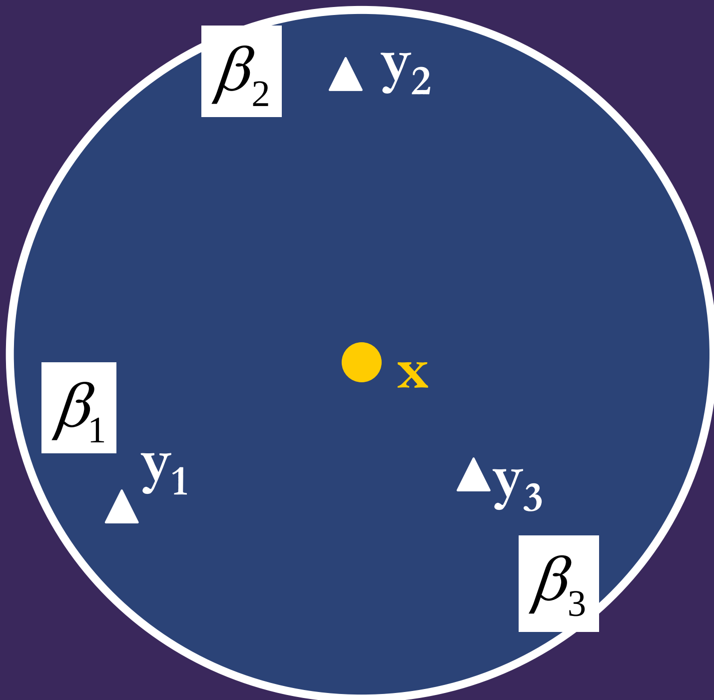
  For all the measurement in the validation gate of a target, select the nearest measurement.

$$[y_k - H_k x_k]^T S_k^{-1} [y_k - H_k x_k] \le \gamma^2$$

- Probabilistic Data Association (PDA)
- Joint Probabilistic Data Association (JPDA)

Probabilistic Data Association

$$\beta_j = P\{X_j \mid Y^k\}$$



Consider a single target independently of others

χ_j denotes the event that the j^{th} measurement belongs to that target.

Combined (Weighted) Innovation

$$\tilde{y}_k = \sum_{j=1}^{m} \beta_j \tilde{y}_{kj} = \sum_{j=1}^{m} \beta_j (y_{kj} - H_k \hat{x}_{k|k-1})$$

Y. Bar-Shalom and E. Tse, "Tracking in a cluttered environment with probabilistic data association," *Automatica*, vol. 11, pp. 451-460, Sept. 1975.

Modified PDA for Video Object Tracking

- To handle video objects (regions), incorporate the following factor when computing β_j

$$\alpha \frac{Similarity_j}{\sum_{i=1}^{m} Similarity_i} + (1-\alpha)\frac{OverlapArea_j}{\sum_{i=1}^{m} OverlapArea_i}$$

$$0 < \alpha < 1$$

- Similarity measure: cross correlation function

$$C_R = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{(\sum_m \sum_n (A_{mn} - \bar{A})^2)(\sum_m \sum_n (B_{mn} - \bar{B})^2)}}$$

Experimental Videos

Vehicle Tracking Results 1

Vehicle Tracking Results 2

Human Tracking Results

Object Tracking Statistics

Video	Sequence 1	Sequence 2	Sequence 3	Sequence 4
Ground Truth	71	64	92	130
Object Detected	72	61	93	128
Miss	0	3	0	5
False Alarm	1	0	1	3
Correctly Detected	71	61	92	125
Correctly Tracked	70	58	92	120
Detection Precision	0.986	1.000	0.989	0.977
Detection Recall	1.000	0.953	1.000	0.962
Tracking Success Rate	0.986	0.951	1.000	0.960

$$\text{Detection Precision} = \frac{\text{Correctly Detected}}{\text{Object Detected}}$$

$$\text{Detection Recall} = \frac{\text{Correctly Detected}}{\text{Ground Truth}}$$

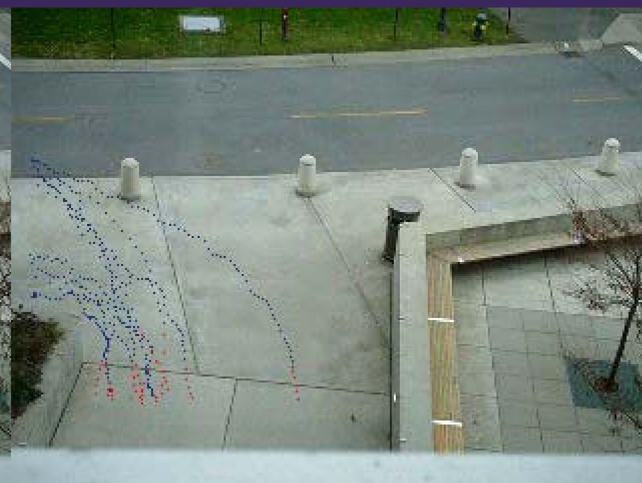
$$\text{Tracking Success Rate} = \frac{\text{Correctly Tracked}}{\text{Correctly Detected}}$$

Occluded Object Tracking Success Rate: 0.855

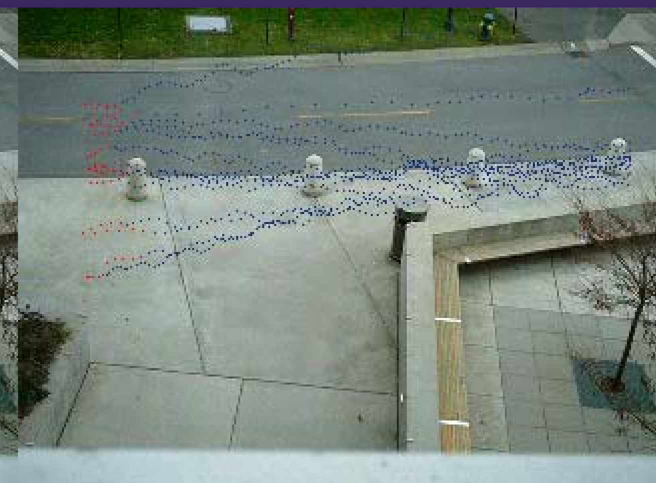
Trajectory Analysis

Class 1

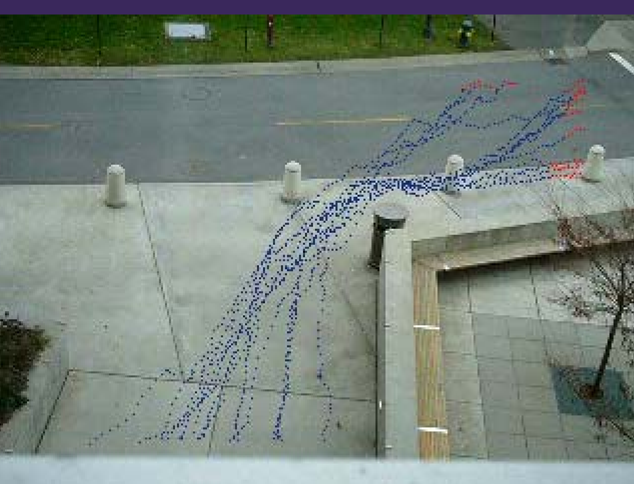
Class 3



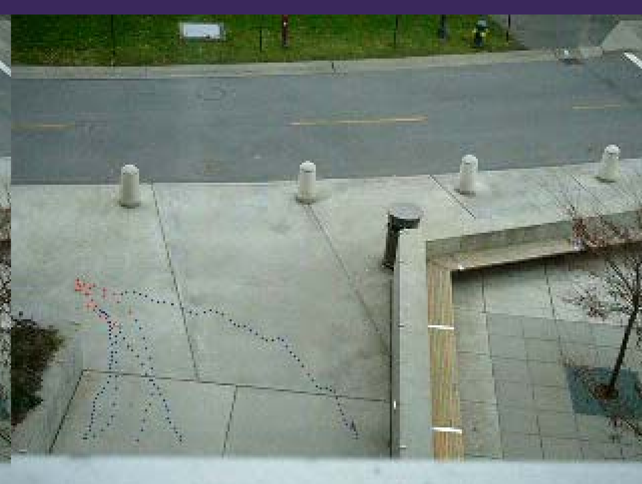
Class 5



Class 2



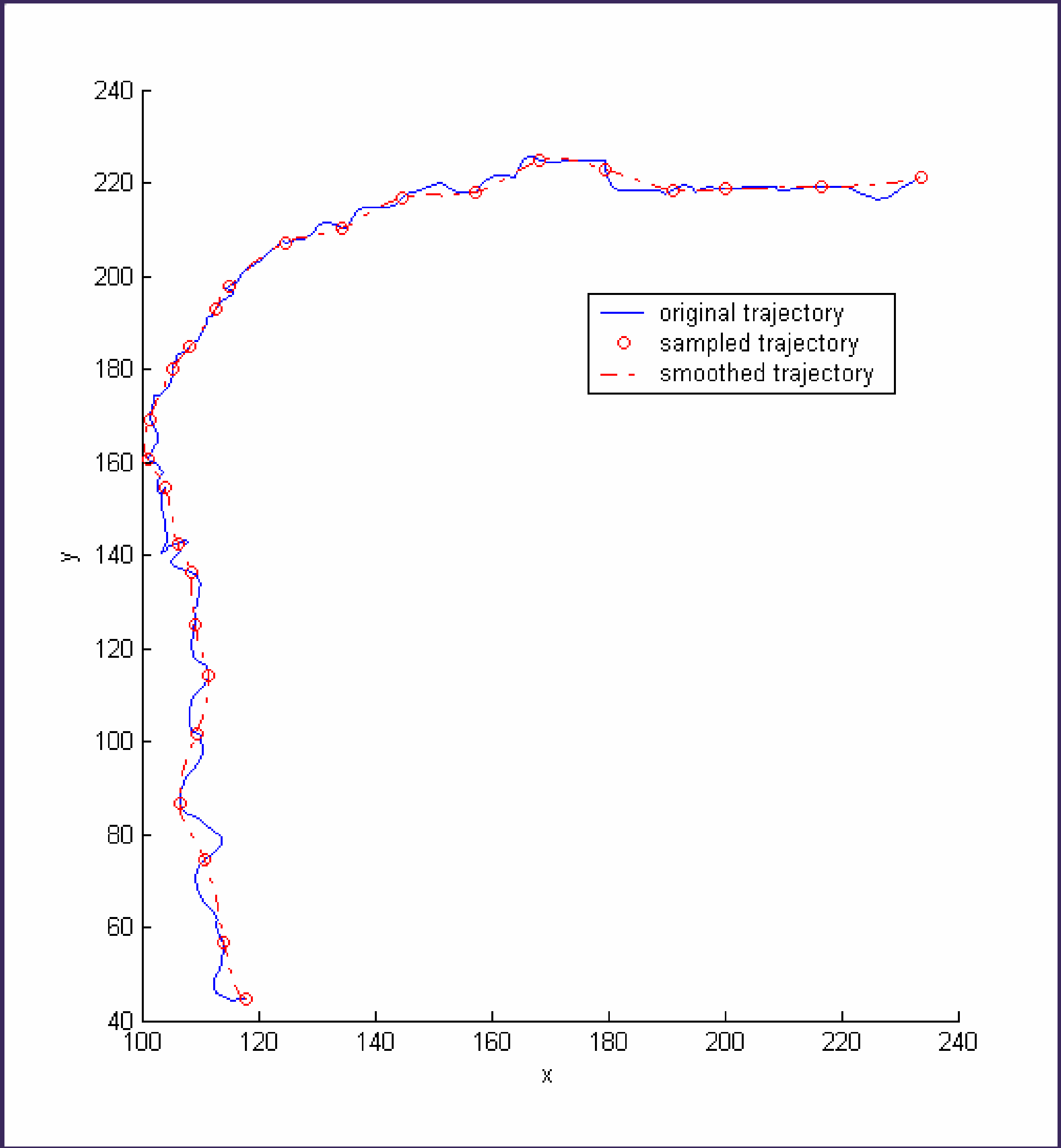
Class 4



Class 6

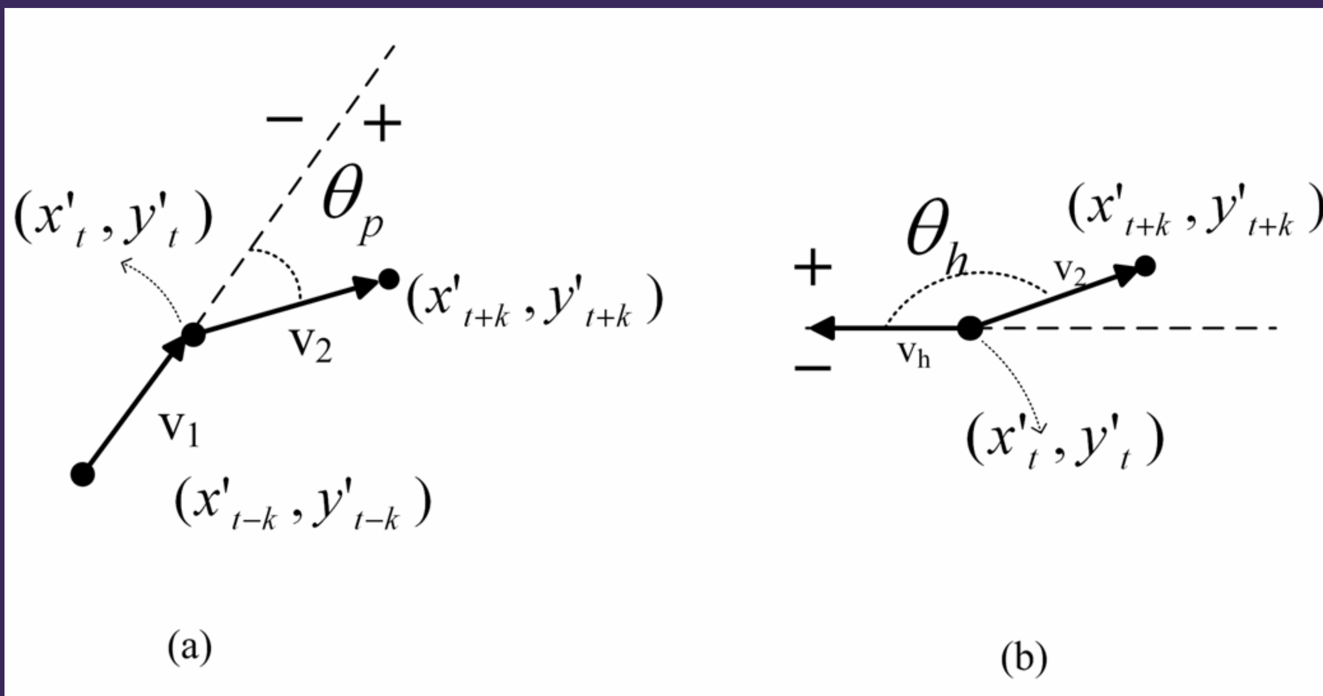
Trajectory Smoothing

- Sample the trajectory
- Perform cubic spline interpolation



Angle Feature Extraction

Relative Angle | **Absolute Angle**



$$\vec{v}_1 = (\tilde{x}_t - \tilde{x}_{t-k}, \tilde{y}_t - \tilde{y}_{t-k})$$

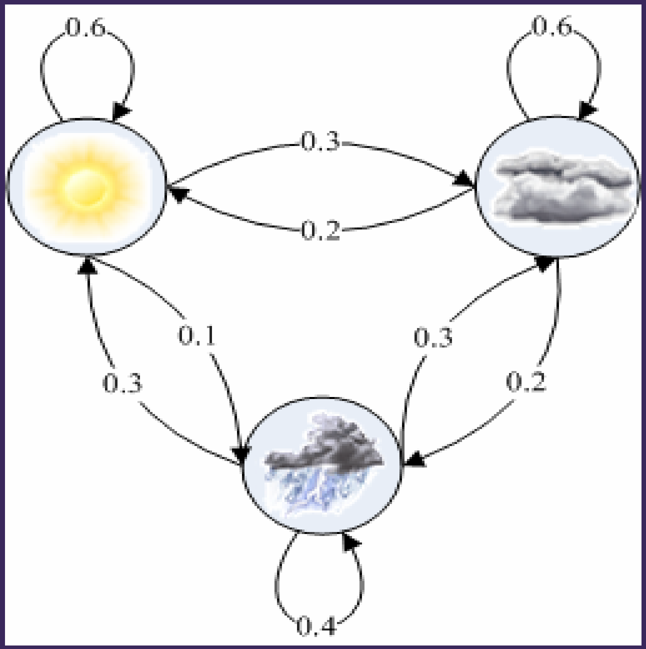
$$\vec{v}_2 = (\tilde{x}_{t+k} - \tilde{x}_t, \tilde{y}_{t+k} - \tilde{y}_t)$$

$$\theta_p = \cos^{-1}\left(\frac{\vec{v}_1 \cdot \vec{v}_2}{|\vec{v}_1||\vec{v}_2|}\right)$$

$$\theta_h = \cos^{-1}\left(\frac{\vec{v}_h \cdot \vec{v}_2}{|\vec{v}_h||\vec{v}_2|}\right)$$

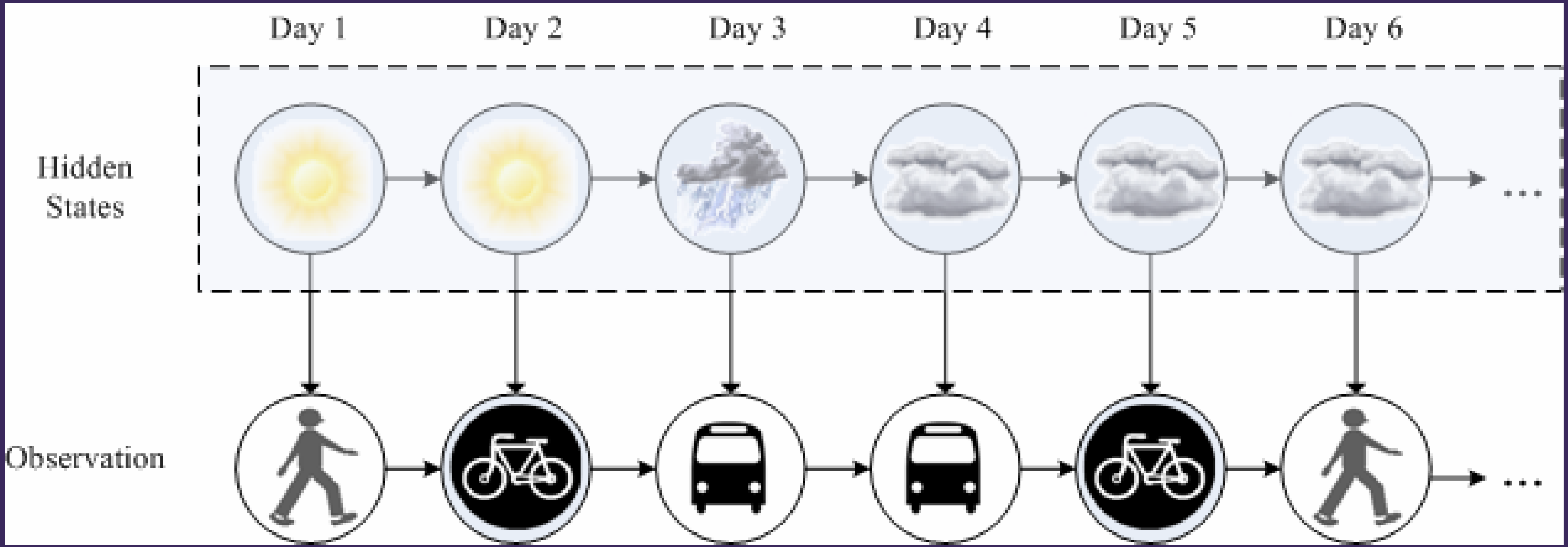
Hidden Markov Model

- N states S_i , i=1,…, N
- Transition probability a_{ij}
- Initial probability π_i
- Observation symbol probability $b_j(k)$
- A complete model λ=(A,B,Π)
  - A={a_{ij}}
  - B={b_{jk}}
  - Π={π_i}



Sunny	Cloudy	Rainy
P(walk) = 0.5 P(bike) = 0.4 P(bus) = 0.1	P(walk) = 0.4 P(bike) = 0.3 P(bus) = 0.3	P(walk) = 0.2 P(bike) = 0.1 P(bus) = 0.7

Example of HMM



Observation sequence $O = \{$ walk, bike, bus, bus, bike, walk, … $\}$

Three Problems in HMM

- Given λ, compute the probability that O is generated by this model

  How likely did *O* happen at this place?

  forward-backward algorithm

- Given λ, find the most likely sequence of hidden states that could have generated O

  How did the weather change day-by-day?

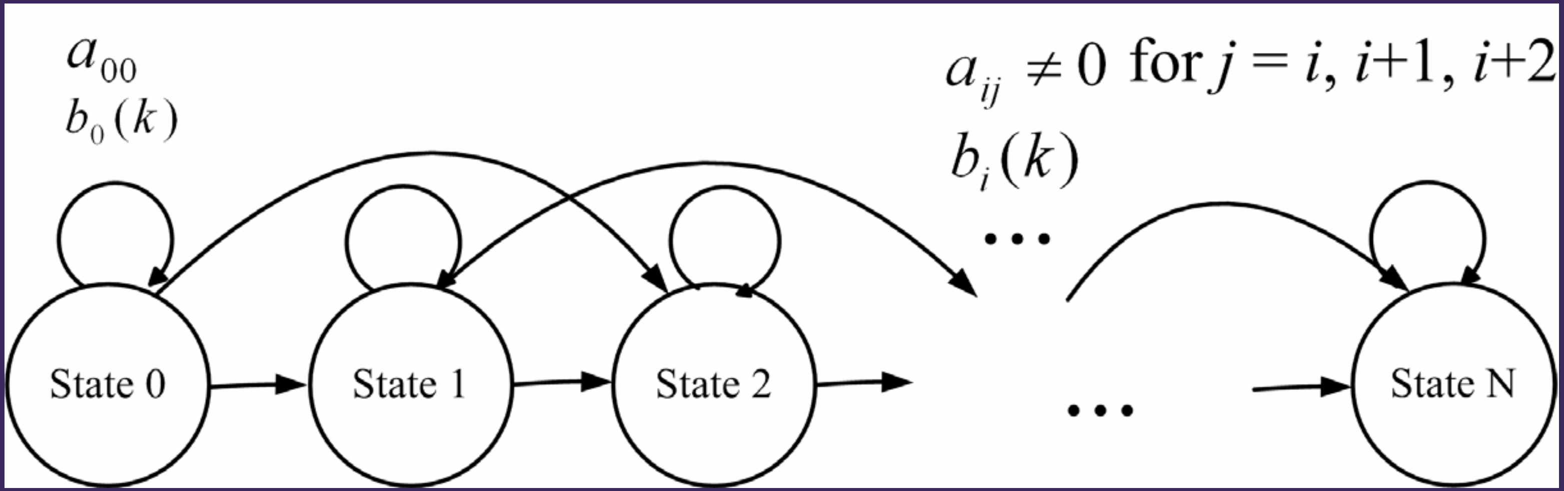
  Viterbi algorithm

- Given a set of O, learn the most likely λ

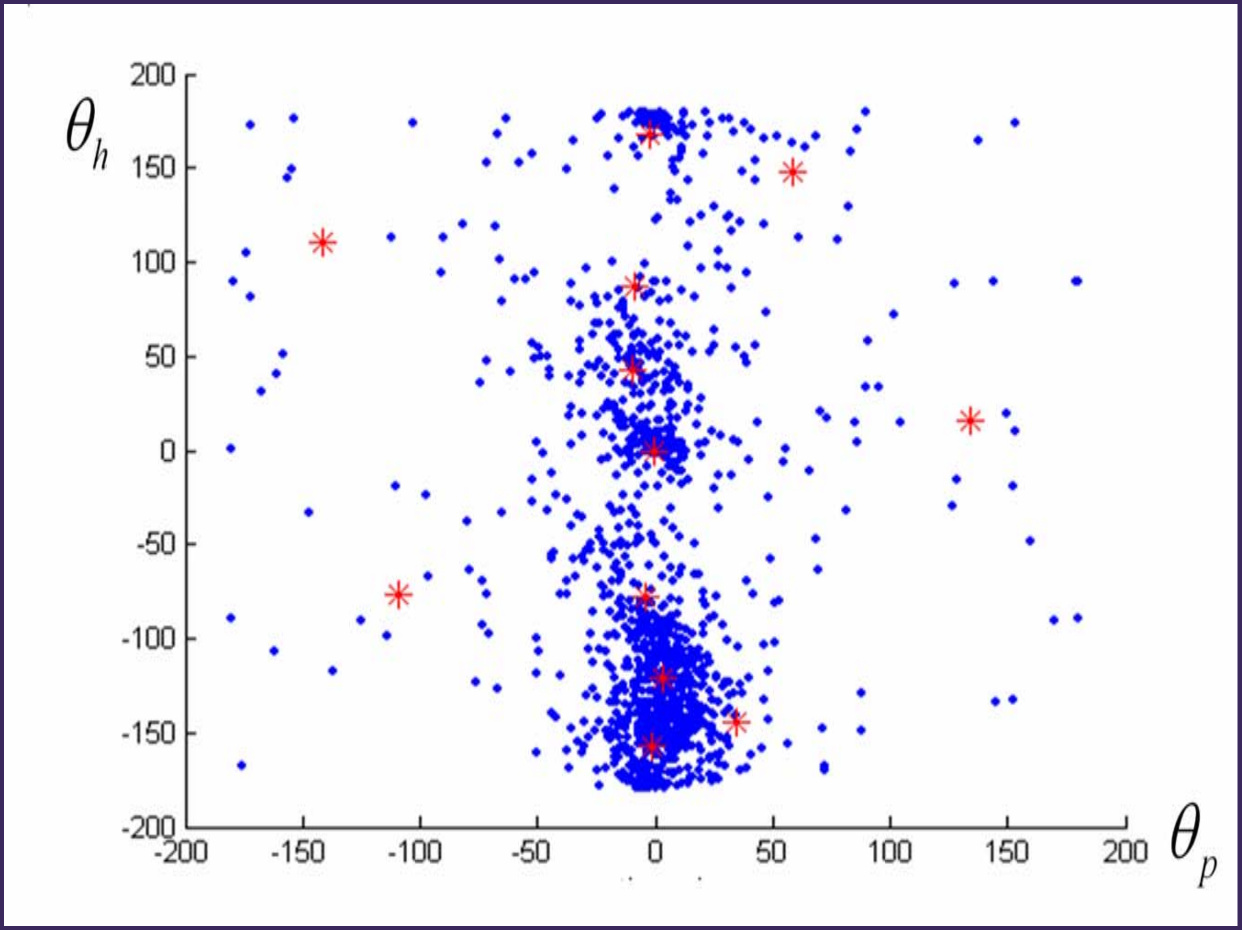
  Train the parameters of the HMM

  Baum-Welch algorithm

Left-to-right HMM for Trajectory Classification



K-means Clustering of Feature Points



Number of Training and Test sequences

Video for both training and testing

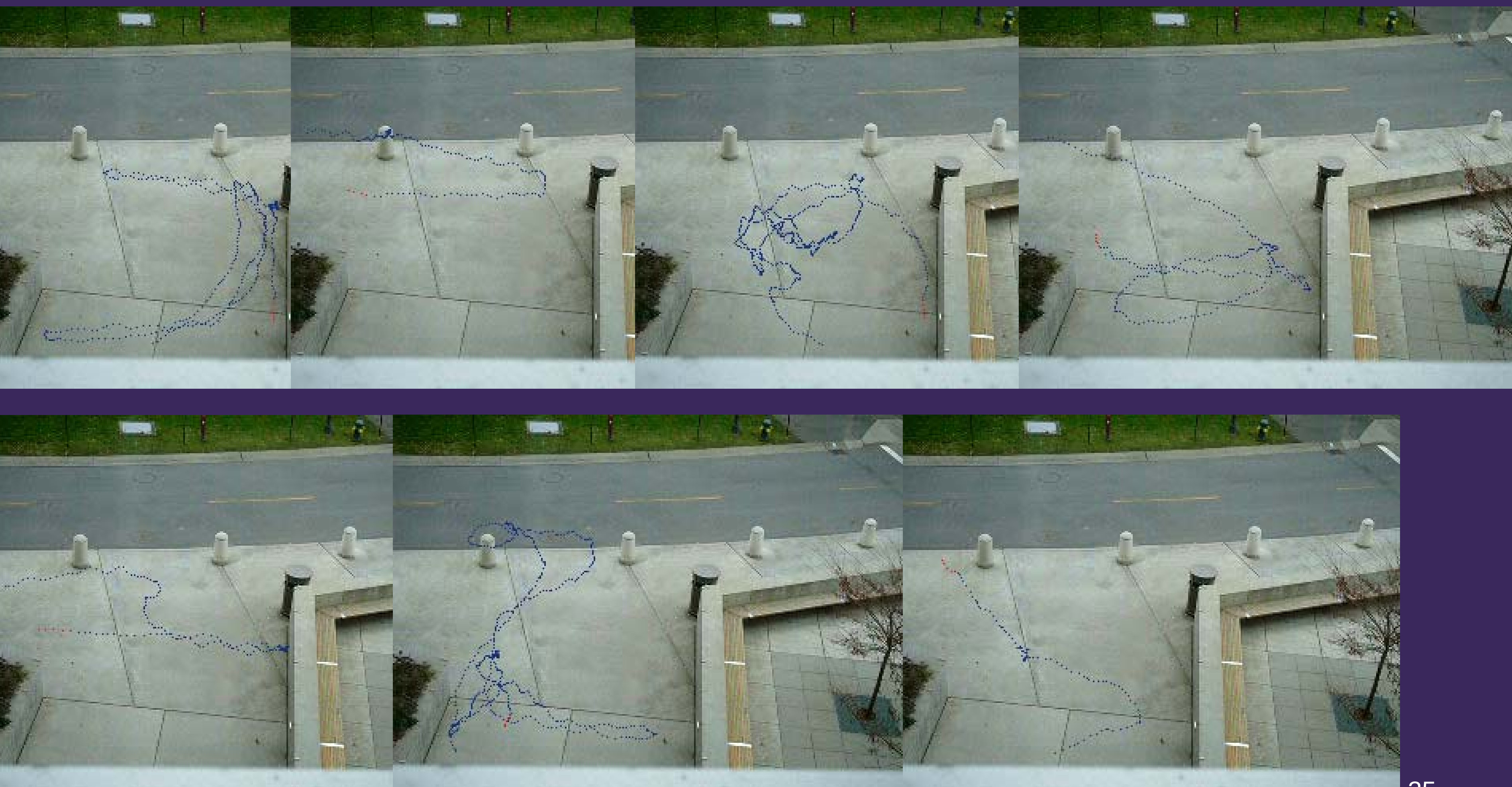
Video for testing only

Trajectory Class	Training Trajectories	Testing Objects	Testing Trajectories
Class 1	12	64	307
Class 2	11	18	66
Class 3	13	27	27
Class 4	5	20	20
Class 5	8	26	32
Class 6	8	29	45

Trajectory Classification Statistics

	C 1	C 2	C 3	C 4	C 5	C 6	Accuracy
Class 1	307	0	0	0	0	0	100%
Class 2	0	64	0	0	0	2	97.4%
Class 3	2	0	25	0	0	0	92.6%
Class 4	0	0	0	20	0	0	100%
Class 5	1	0	0	0	31	0	96.8%
Class 6	0	2	0	0	0	43	95.5%

Anomalous Trajectories



Event Detection

- Type I Events
  - Simple rule-based decision logic
  - Entering a dangerous region
  - Stopping in the scene
  - Driving on the road shoulder
- Type II Events
  - Based on trajectory classification results via HMM using angle features
  - Illegal U-turns or left turns
  - Anomalous trajectories
- Type III Events
  - Based on trajectory classification results via HMM using speed features
  - Speed change

Conclusions and Future Works

- Tracking
  - Kalman filtering for prediction
  - Modified PDA for data association
- Basic Events
  - Simple rule-based decision logic
  - HMM
- Higher Level Events
  - Combining basic events
  - More flexible models