Motion segmentation by model-based clustering of incomplete trajectories

V. Karavasilis, K. Blekas, C. Nikou

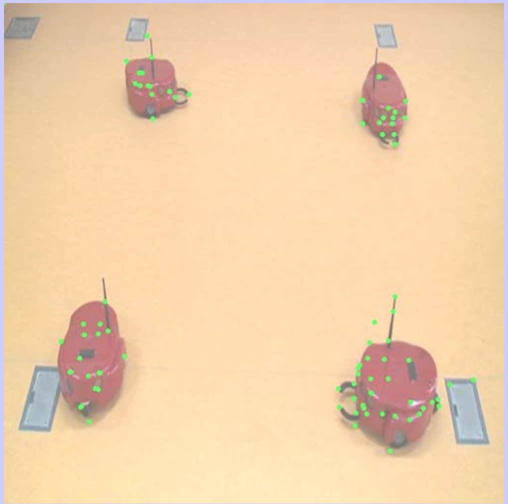
Department of Computer Science
University of Ioannina
Greece

Outline

- **Problem Explanation**
- Extracting Trajectories
- Clustering Trajectories
- Experimental Results
- Conclusions

Problem Explanation

- Image sequence with moving objects
- Tracking
  - Locating a moving object using a camera
- Motion segmentation
  - Separating objects based on their movement

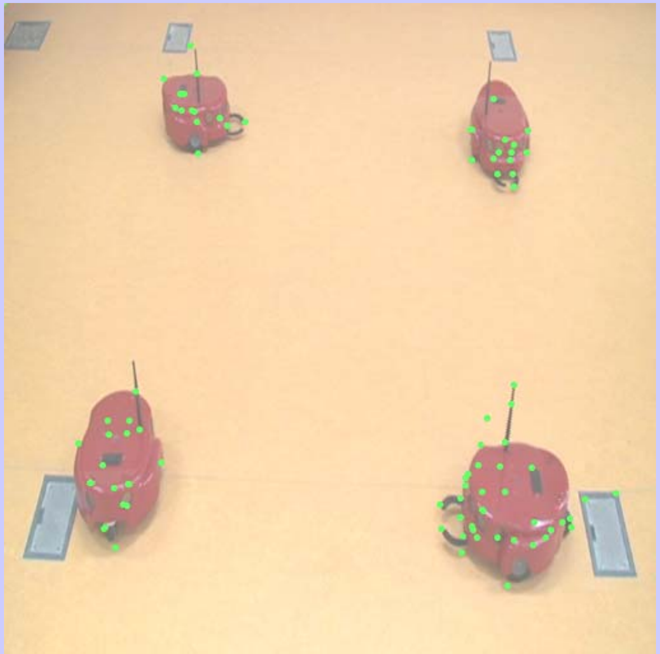


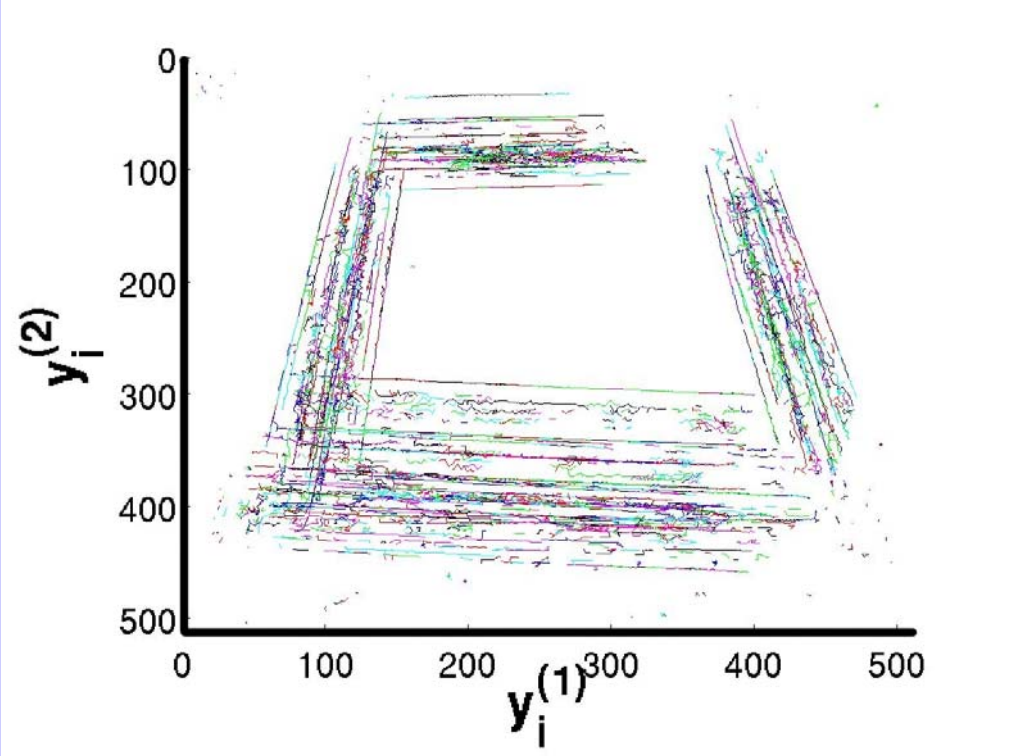
State of the art

- Tracking algorithms
  - Mean shift [Comaniciu *et al.* 2003]
  - Kalman filter [Cuevas *et al*. 2005]
  - Condensation [Isard and Blake 1998]
- Motion segmentation
  - LCSS [Buzan *et al.* 2004]
  - GPCA [Vidal *et al.* 2005]
  - SSC [Elhamifar and Vidal 2009]
  - HSC [Kim and Agapito 2009]
  - ELSA [Zappela *et al.* 2011]

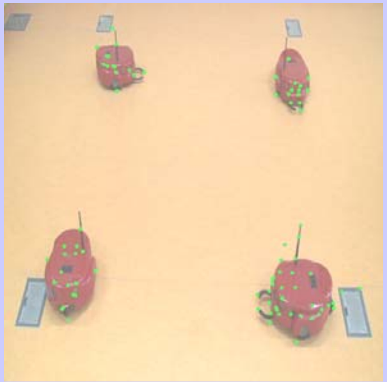
Method Overview

- Extracting trajectories with variable length
- Estimating the motion by clustering the trajectories

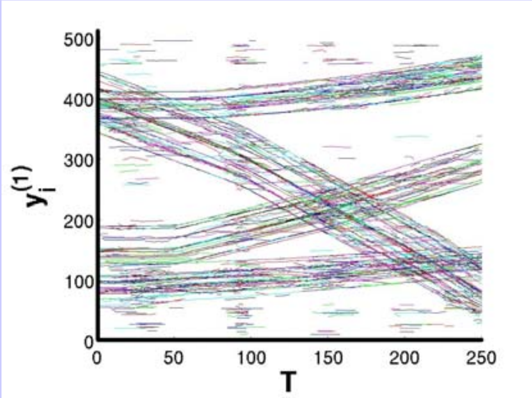


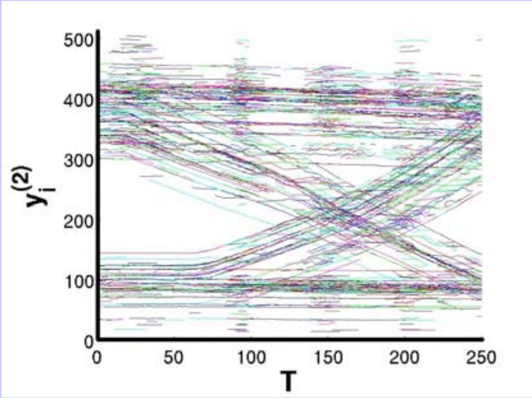


The Proposed Method

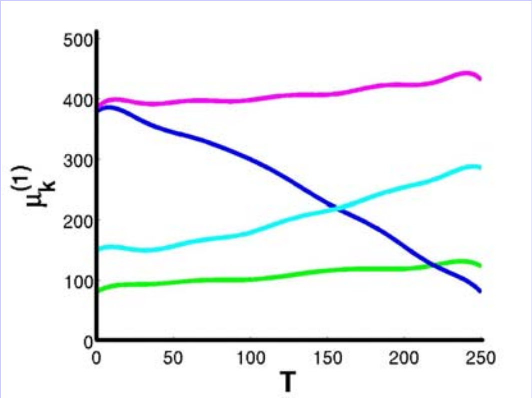


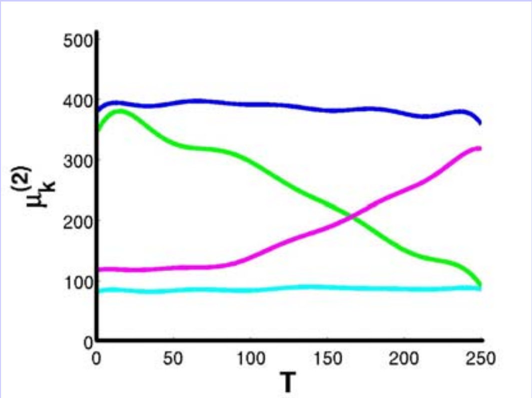
Trajectories Extractions



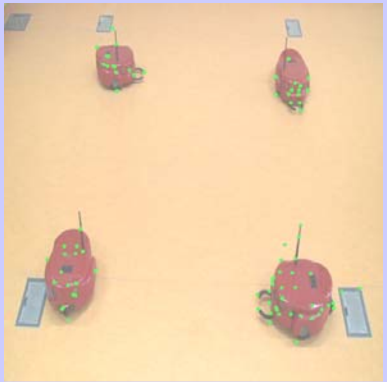


Trajectories Clustering

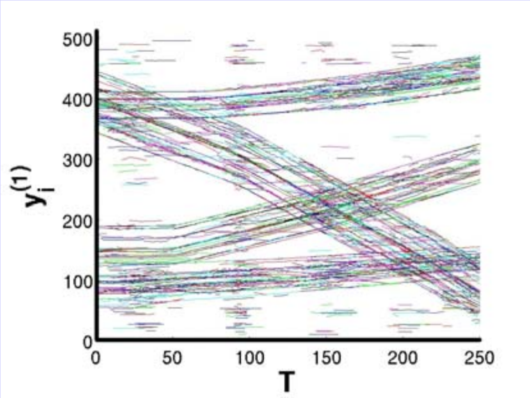


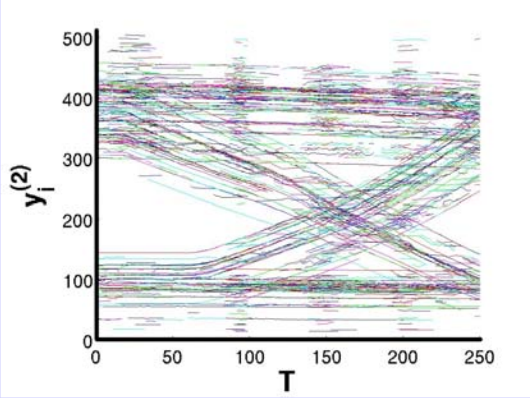


The Proposed Method

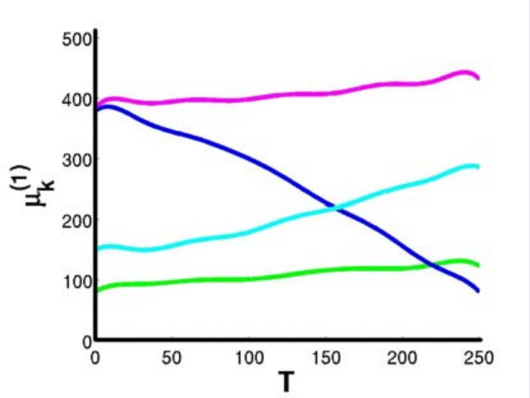


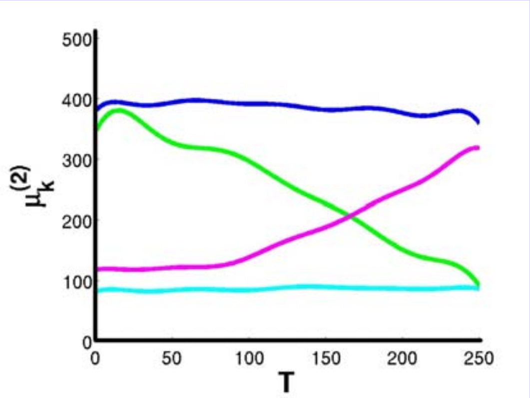
Trajectories Extractions





Trajectories Clustering



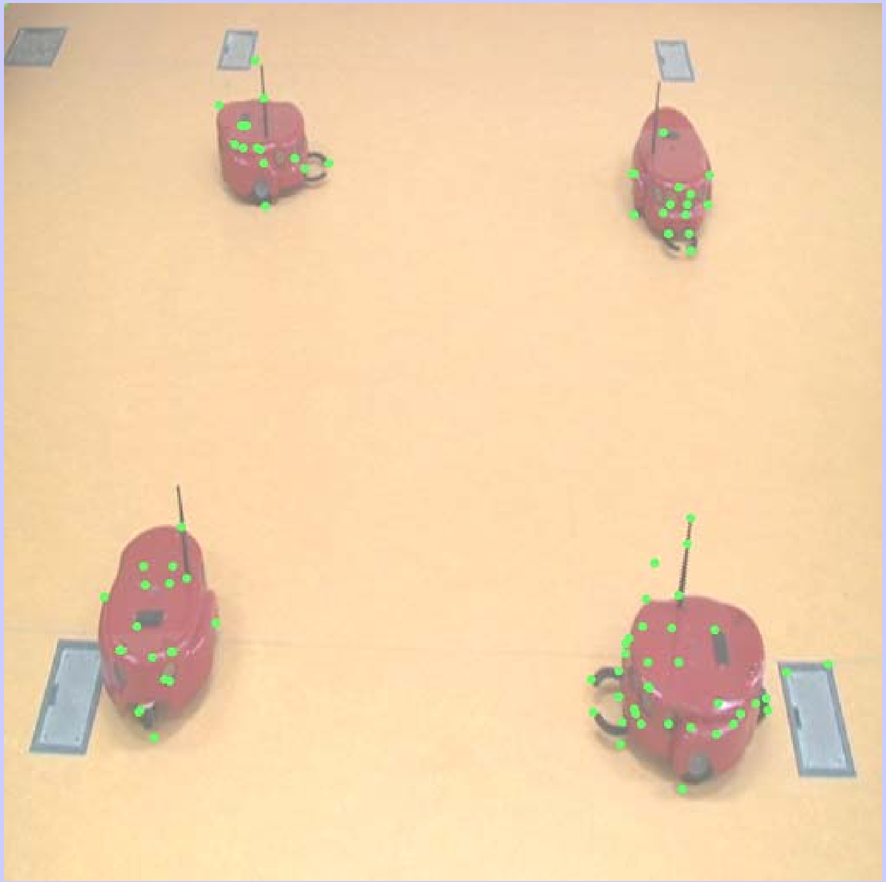


Outline

- Problem Explanation
- **Extracting Trajectories**
- Clustering Trajectories
- Experimental Results
- Conclusions

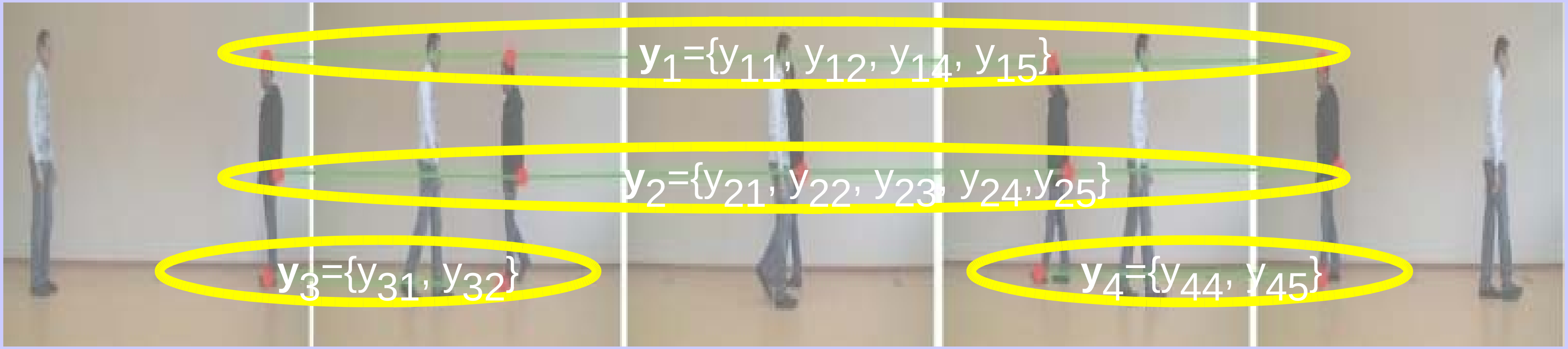
Key Points

- Extract features using **Harris corners** [Harris 1988]
  - Edges and corners



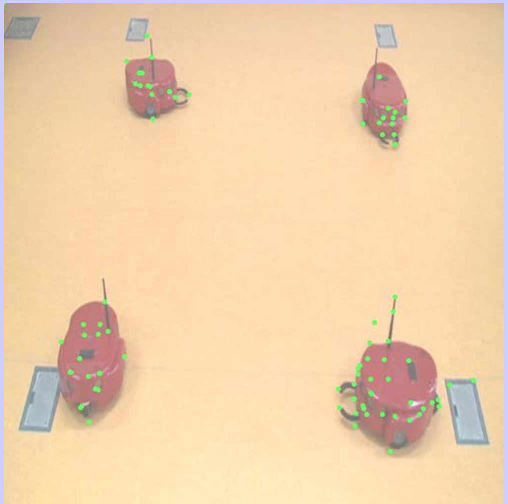
Trajectories Creation

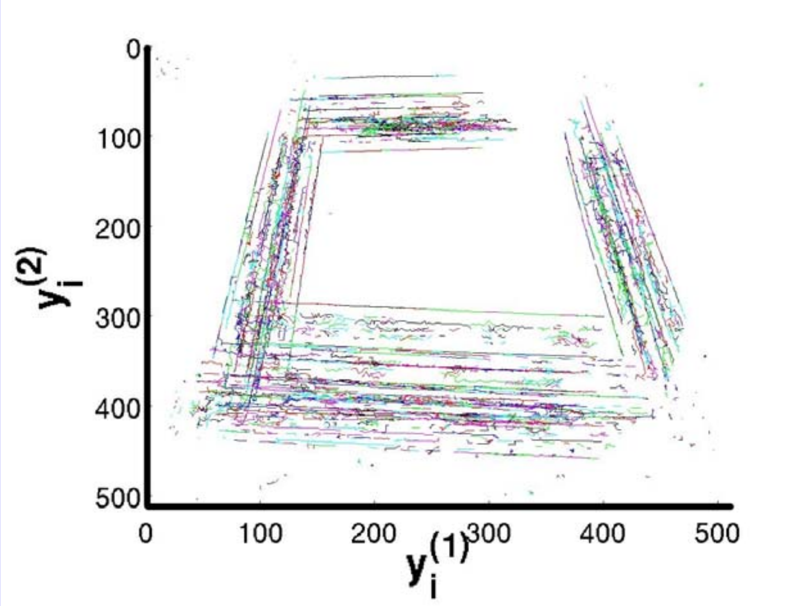
- Associate features using **optical flow** [Lucas *et al*. 1981]
  - Features may disappear and reappear
  - Connect trajectories based on the similarity of a window around the corner

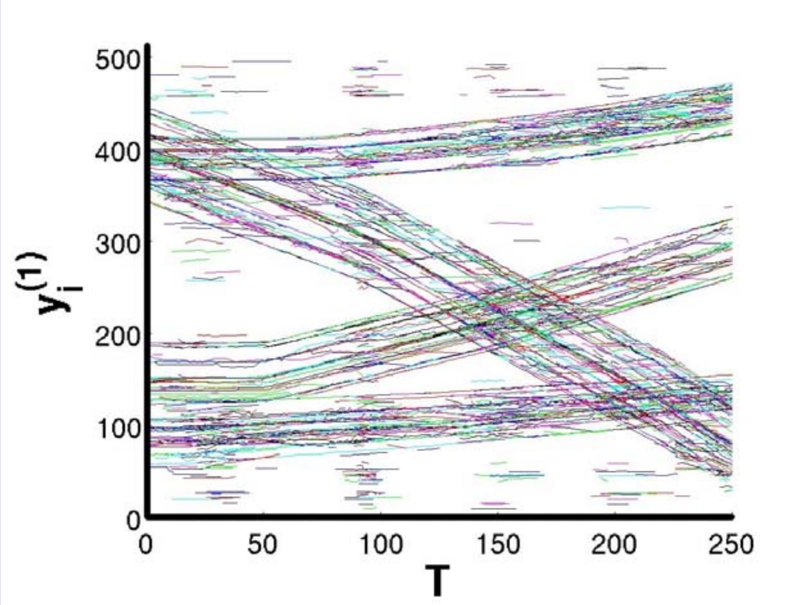


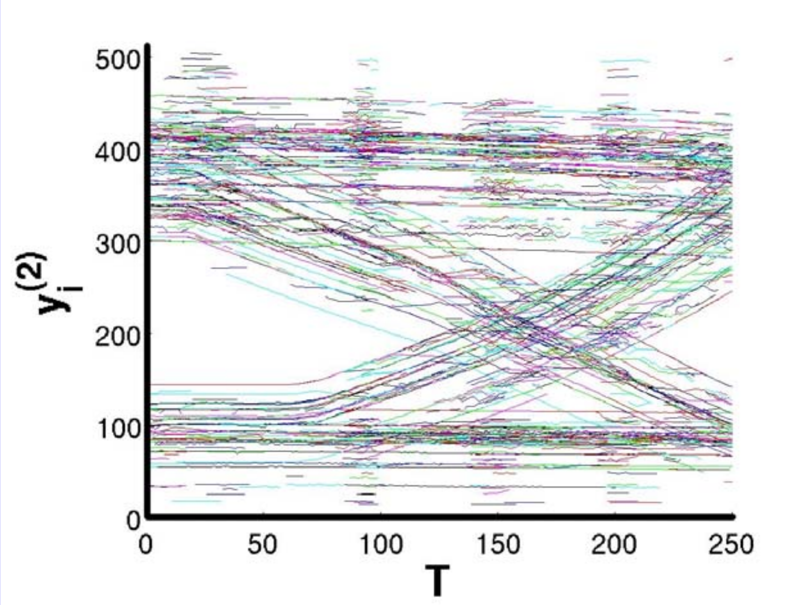
t=1 t=2 t=3 t=4 t=5

Extracting Trajectories







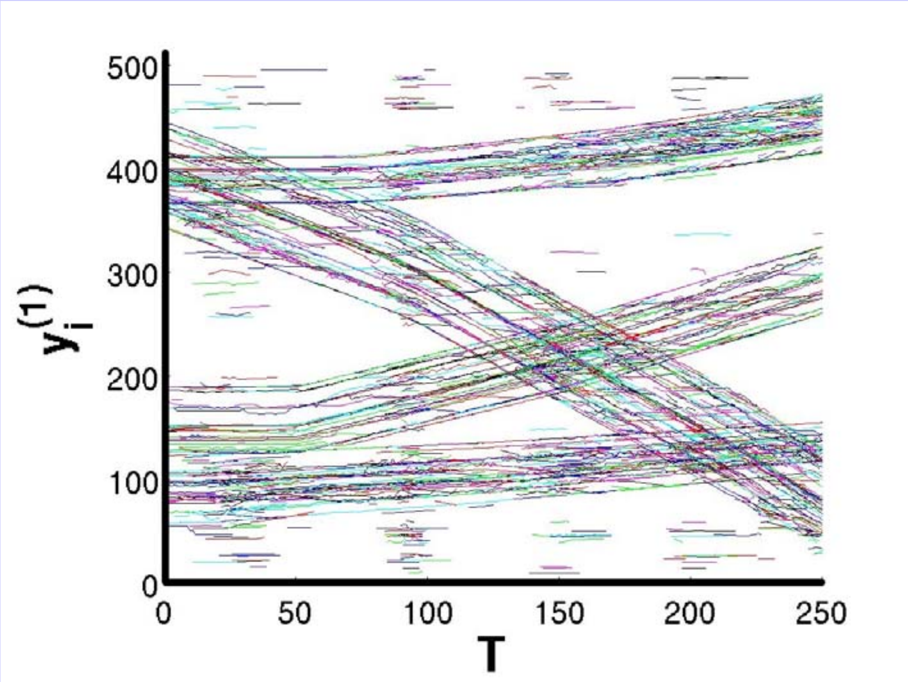


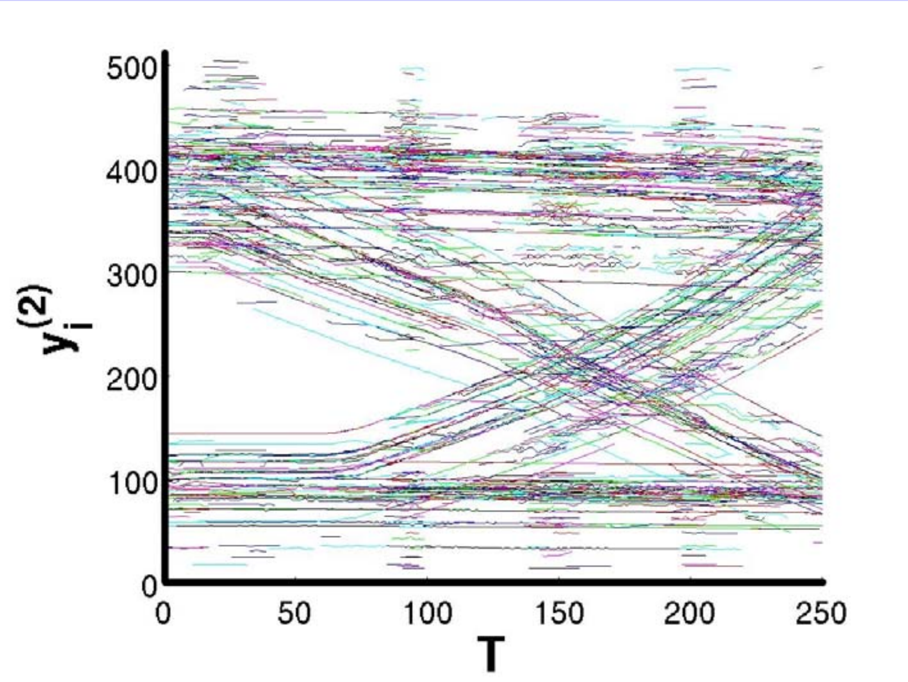
Outline

- Problem Explanation
- Extracting Trajectories
- **Clustering Trajectories**
- Experimental Results
- Conclusions

Input Trajectories

- Input set: $\boldsymbol{Y} = \left\{ \boldsymbol{y_i} = \left(\boldsymbol{y_i^{(1)}}, \boldsymbol{y_i^{(2)}} \right) \right\}_{i=1,\ldots,N}$
  - Horizontal and vertical coordinates
  - Trajectories of variable length T_i
  - Total number of frames: T





Linear Regression Model

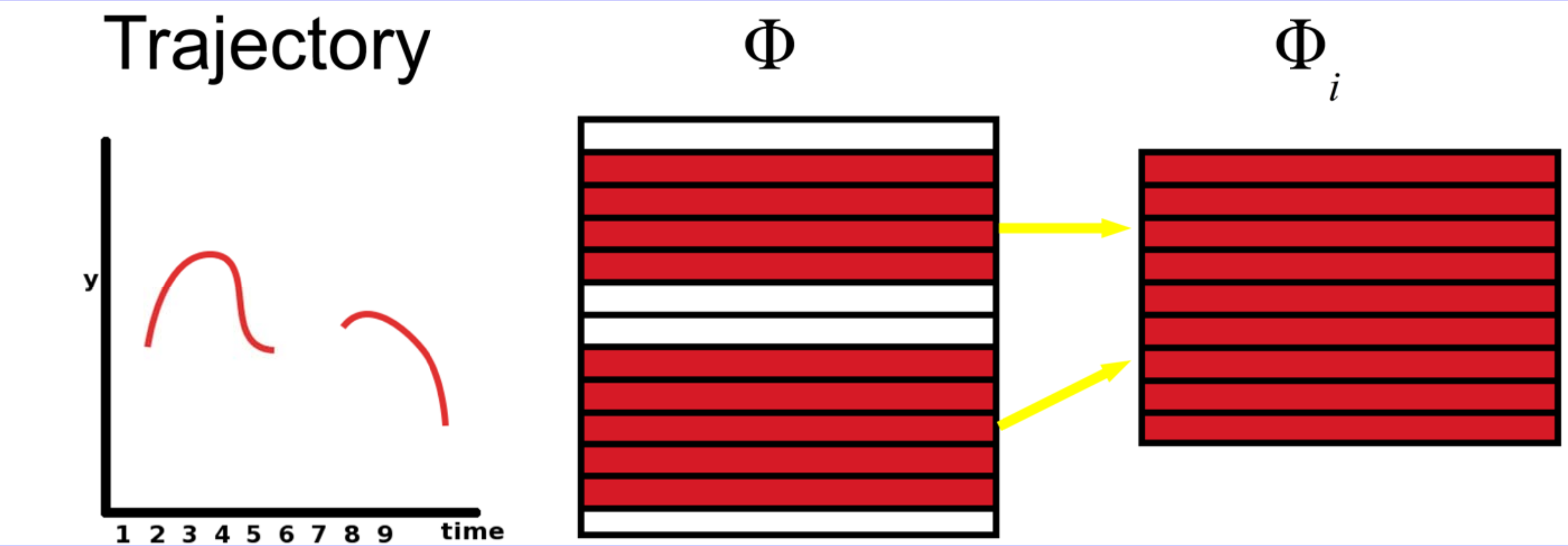
$$\boldsymbol{y}_{\boldsymbol{i}}^{(j)} = \Phi_i \boldsymbol{w}^{(j)} + d_i^{(j)} + \boldsymbol{e}_i^{(j)}$$

- Φ_i: design kernel matrix of size $T_i \times T$
- $\boldsymbol{w}^{(j)}$: regression coefficients (unknown)
- $d_i^{(j)}$: trajectory translation $d_i^{(j)} \sim N(0, u^{(j)})$
- $\boldsymbol{e}_i^{(j)}$: error term, $\boldsymbol{e}_i^{(j)} \sim N(\boldsymbol{0}, \Sigma_i^{(j)})$

Kernel matrix

- Each row of Φ corresponds to one frame
- Mexican hat wavelet function

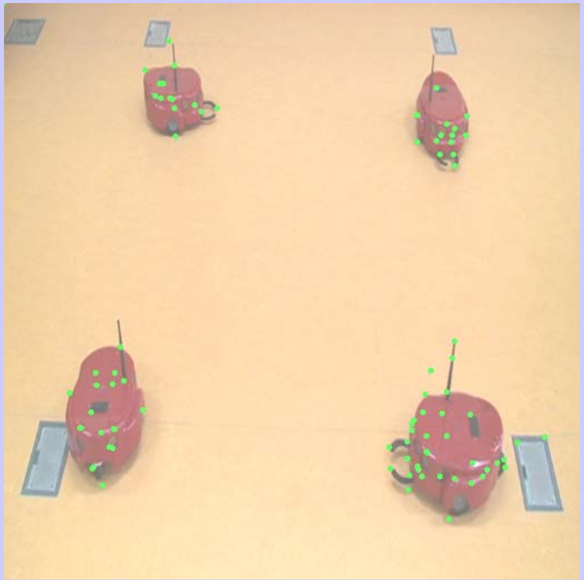
$$\Phi(n,m;\sigma)=\frac{2}{\sqrt{3\sigma}\,\pi^{1/4}}\left(1-\frac{(m-n)^2}{\sigma^2}\right)e^{\frac{-(m-n)^2}{2\sigma^2}}$$

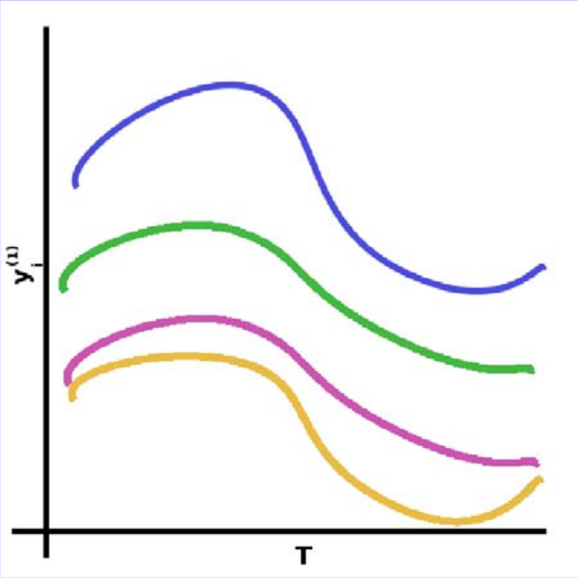


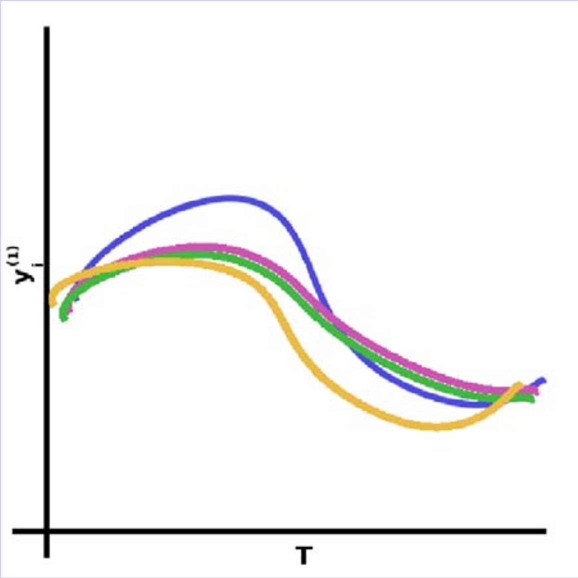
$T \times T$ $\qquad$ $T_i \times T$

Translation term $d_i^{(j)} \sim N(0,u)$

- Allows the entire trajectory of the key points to be translated globally
- The trajectories of the key points are aligned with the trajectory of the center of gravity of the object







Conditional density of observations

- Regression model: $\boldsymbol{y}_i^{(j)} = \Phi_i \boldsymbol{w}^{(j)} + d_i^{(j)} + \boldsymbol{e}_i^{(j)}$
- Conditional density of $\boldsymbol{y}_i$ is Gaussian:

$$p\left(\boldsymbol{y}_i^{(j)} | \boldsymbol{w}^{(j)}, \Sigma_i^{(j)}, d_i^{(j)}\right) = N\left(\Phi_i \boldsymbol{w}^{(j)} + d_i^{(j)}, \Sigma_i^{(j)}\right)$$

- Integrating out $d_i^{(j)}$:

$$\begin{aligned} p\left(\boldsymbol{y}_i^{(j)} | \boldsymbol{\theta}^{(j)}\right) &= \int p\left(\boldsymbol{y}_i^{(j)} | \boldsymbol{w}^{(j)}, \Sigma_i^{(j)}, d_i^{(j)}\right) p\left(d_i^{(j)}\right) dd_i^{(j)} \\ &= N\left(\Phi_i \boldsymbol{w}^{(j)}, \Sigma_i^{(j)} + u^{(j)} \mathbb{1}\right) \end{aligned}$$

  - where $\boldsymbol{\theta}^{(j)} = \{\boldsymbol{w}^{(j)}, u^{(j)}, \Sigma^{(j)}\}$

Regression Mixture Model

- K objects
- Each object has parameters $\theta_k^{(j)} = \left\{ \boldsymbol{w}_k^{(j)}, u_k^{(j)}, \Sigma_k^{(j)} \right\}$
- Regression model:

$$\boldsymbol{y}_i^{(j)} = \Phi_i \boldsymbol{w}_k^{(j)} + d_{ik}^{(j)} + \boldsymbol{e}_k^{(j)}$$

- Mixture Model:

$$p\left(\boldsymbol{y_i} \mid \boldsymbol{\Theta}\right) = \sum_{k=1}^{K} \pi_k \overbrace{p\left(\boldsymbol{y_i^{(1)}} \mid \theta_{\boldsymbol{k}}^{(\mathbf{1})}\right) p\left(\boldsymbol{y_i^{(2)}} \mid \theta_{\boldsymbol{k}}^{(\mathbf{2})}\right)}^{p\left(\boldsymbol{y_i} \mid \theta_{\boldsymbol{k}}\right)}$$

  - where

$$\boldsymbol{\Theta} = \left\{ \pi_k, \theta_k^{(1)}, \theta_k^{(2)} \right\}_{k=1,\ldots,K}$$

Sparse Modeling

- Smooth trajectories and avoid over-fitting
- Hierarchically:

1) $p\left(\boldsymbol{w}_k^{(j)} | \boldsymbol{\alpha}_k^{(j)}\right) = N\left(\boldsymbol{w}_k^{(j)} | \mathbf{0}, \mathrm{A}_k^{-1^{(j)}}\right) = \prod_{l=1}^{T} N\left(w_{kl}^{(j)} | 0, \alpha_{kl}^{-1^{(j)}}\right)$

$$A_k = diag\left(\alpha_{k1}^{(j)}, \ldots, \alpha_{kT}^{(j)}\right)^T$$

2) Student's *t* (Sparse prior)

$$p\left(\boldsymbol{\alpha}_k^{(j)}\right) = \prod_{l=1}^{T} Gamma\left(\alpha_{k1}^{(j)} | a, b\right) \propto \prod_{l=1}^{T} \alpha_{k1}^{a-1^{(j)}} e^{-b\alpha_{k1}^{(j)}}$$

- $p(\boldsymbol{w}_k^{(j)})$: Student's t (sparse)

MAP estimation problem

- Maximize the MAP log-likelihood with respect to $\mathbf{\Theta}$:

$$
\begin{aligned}
L(\Theta) \quad &= \quad \ln p(\boldsymbol{Y} \mid \mathbf{\Theta}) + \ln p(\mathbf{\Theta}) \\
&= \quad \sum_{i=1}^{N} \log \left\{ \sum_{k=1}^{K} \pi_k \, p\left(\boldsymbol{y}_i \mid \boldsymbol{\theta}_k^{(1)}\right) p\left(\boldsymbol{y}_i \mid \boldsymbol{\theta}_k^{(2)}\right) \right\} \\
&\quad + \sum_{k=1}^{K} \sum_{j=1}^{2} \log p\left(\boldsymbol{w}_k^{(j)} \mid \boldsymbol{\alpha}_k^{(j)}\right) \\
&\quad + \sum_{k=1}^{K} \sum_{j=1}^{2} \log p\left(\boldsymbol{\alpha}_k^{(j)}\right)
\end{aligned}
$$

$$
\mathbf{\Theta} = \left\{ \pi_k, \boldsymbol{\theta}_k^{(1)}, \boldsymbol{\theta}_k^{(2)} \right\}_{k=1,\ldots,K}
$$

$$
\boldsymbol{\theta}_k^{(j)} = \left\{ \boldsymbol{w}_k^{(j)}, \boldsymbol{u}_k^{(j)}, \Sigma_k^{(j)} \right\}
$$

EM Algorithm (1)

- E-step: expectation of hidden variables

$$z_{ik} = P\left(k \mid \boldsymbol{y}_i\right) = \frac{\pi_k \, p\left(\boldsymbol{y}_i^{(1)} \mid \boldsymbol{\theta}_k^{(1)}\right) p\left(\boldsymbol{y}_i^{(2)} \mid \boldsymbol{\theta}_k^{(2)}\right)}{\sum_{k'} \pi_{k'} \, p\left(\boldsymbol{y}_i^{(1)} \mid \boldsymbol{\theta}_{k'}^{(1)}\right) p\left(\boldsymbol{y}_i^{(2)} \mid \boldsymbol{\theta}_{k'}^{(2)}\right)}$$

$$\hat{d}_{ik}^{(j)} = V_{ik}^{(j)} \left(\boldsymbol{y}_i^{(j)} - \Phi_i \boldsymbol{w}_k^{(j)}\right)^T \Sigma_{ik}^{-1^{(j)}} \mathbf{1}_i$$

  - where $V_{ik}^{(j)} = \left(\mathbf{1}_i^T \Sigma_{ik}^{-1^{(j)}} \mathbf{1}_i + \frac{1}{u_k^{(j)}}\right)^{-1}$

EM Algorithm (2)

- M-step: maximize the complete data Q function

$$
\begin{aligned}
Q \quad \propto \quad & \sum_{i=1}^{N} \sum_{k=1}^{K} z_{ik} \Bigg\{ \log \pi_k \\
& + \sum_{j=1}^{2} -\frac{1}{2} \log \left| \Sigma_k^{(j)} \right| - \frac{\left(\boldsymbol{y}_i^{(j)} - \boldsymbol{\mu}_k^{(j)} \right)^T \Sigma_k^{(j)} \left(\boldsymbol{y}_i^{(j)} - \boldsymbol{\mu}_k^{(j)} \right)}{2} \Bigg\} \\
& + \sum_{k=1}^{K} \sum_{j=1}^{2} -\frac{1}{2} \log \left| \mathrm{A}_k^{(j)} \right| - \frac{\boldsymbol{w}_k^{(j)^T} \mathrm{A}_k^{(j)} \boldsymbol{w}_k^{(j)}}{2} \\
& + \sum_{k=1}^{K} \sum_{j=1}^{2} \sum_{l=1}^{T} \log \alpha_{kl}^{(j)^{a-1}} - b \, \alpha_{kl}^{(j)}
\end{aligned}
$$

EM Algorithm (3)

- M-step: update rules

$$\hat{\pi}_k = \frac{1}{N} \sum_{i=1}^{N} z_{ik}$$

$$\hat{\boldsymbol{w}}_k^{(j)} = \left[\sum_{i=1}^{N} z_{ik} \Phi_i^T \Sigma_{ik}^{-1^{(j)}} \Phi_i + \boldsymbol{A}_k^{(j)} \right]^{-1} \sum_{i=1}^{N} z_{ik} \Phi_i^T \Sigma_{ik}^{-1} \left(\boldsymbol{y_i} - \hat{d}_{ik} \right)$$

$$\hat{\alpha}_{kl}^{(j)} = \frac{1+2a}{\hat{w}_{kl}^{2^{(j)}} + 2b}$$

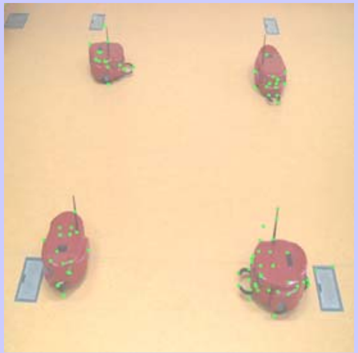
$$\hat{\sigma}_{kl}^{2^{(j)}} = \frac{1}{\sum_{i=1}^{N} z_{ik}} \sum_{i=1}^{N} z_{ik} \left\{ \left(\boldsymbol{y}_{il}^{(j)} - \left[\Phi_i \hat{\boldsymbol{w}}_k^{(j)} \right]_l - \hat{d}_{ik}^{(j)} \right)^2 + V_{ik}^{(j)} \right\}$$

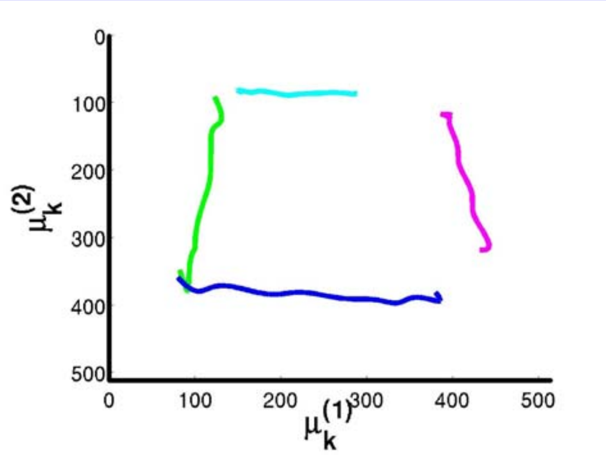
$$\hat{u}_k^{(j)} = \frac{1}{\sum_{i=1}^{N} z_{ik}} \sum_{i=1}^{N} z_{ik} \left(\hat{d}_{ik}^{2^{(j)}} + V_{ik}^{(j)} \right)$$

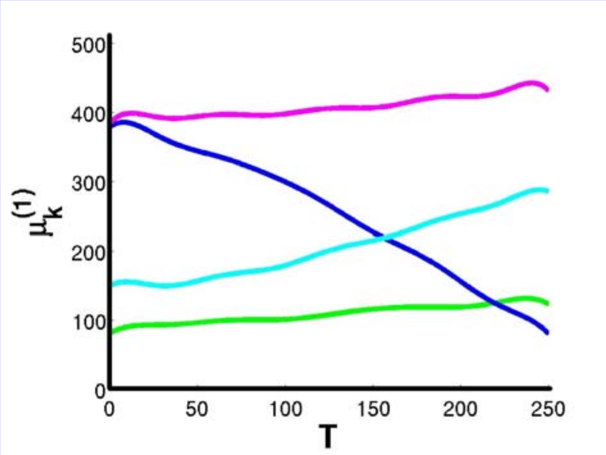
Extracted knowledge (output)

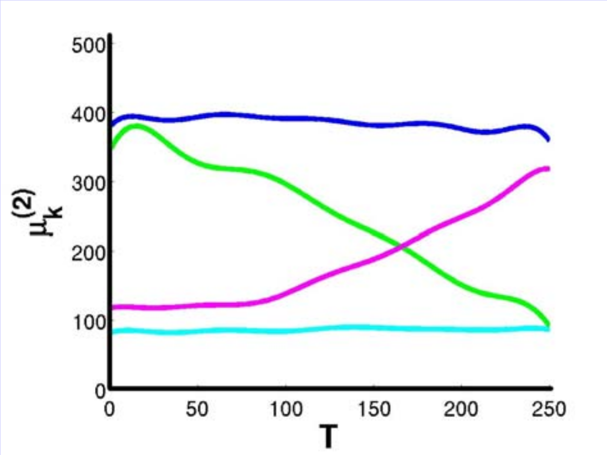
- Assign labels to the trajectories $max_k P\left(k \mid \boldsymbol{y}_i\right)$
- Mean trajectories per cluster

$$\boldsymbol{\mu}_k = \left(\boldsymbol{\mu}_k^{(1)}, \boldsymbol{\mu}_k^{(2)}\right) = \left(\Phi \boldsymbol{w}_k^{(1)}, \Phi \boldsymbol{w}_k^{(2)}\right)$$



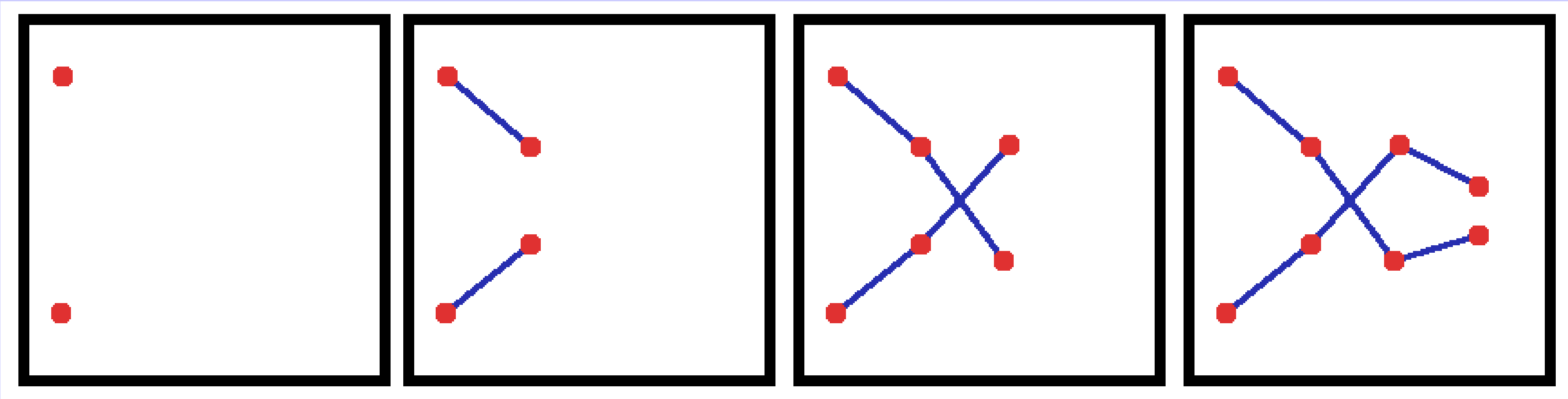






Initialization Strategy (1)

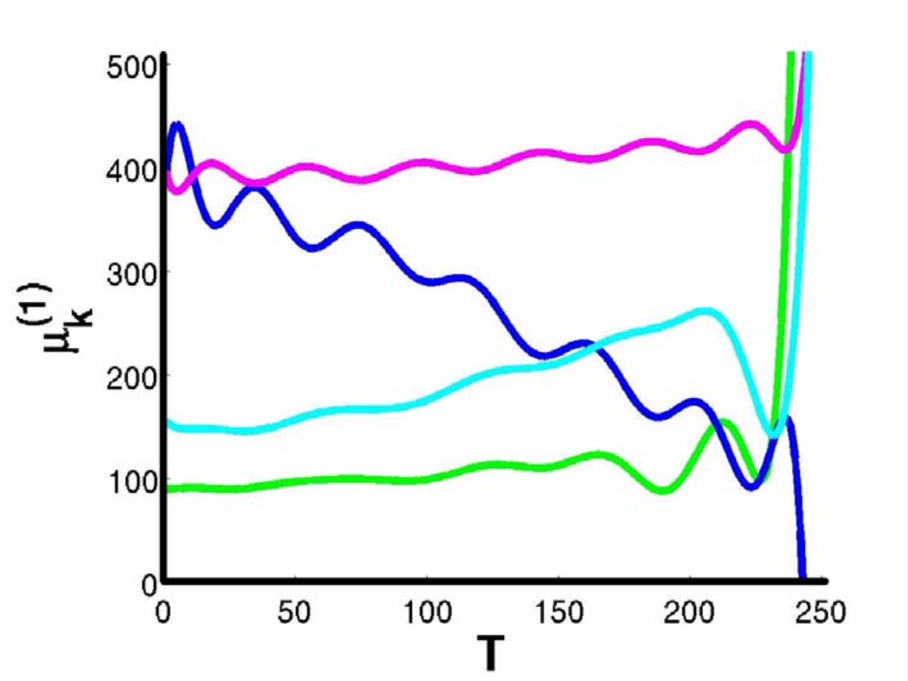
- Random selection from the samples
- Rough initialization
  - Apply k-means at certain time instances
  - Associate the resulting centers
  - Interpolation



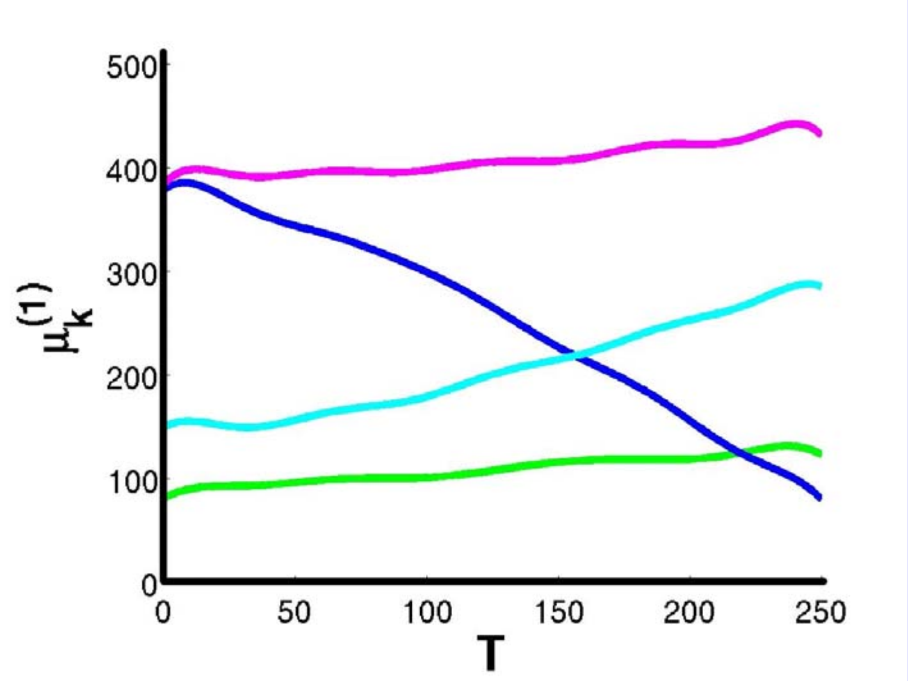
Initialization Strategy (2)

Example

Initialization



EM convergence



Outline

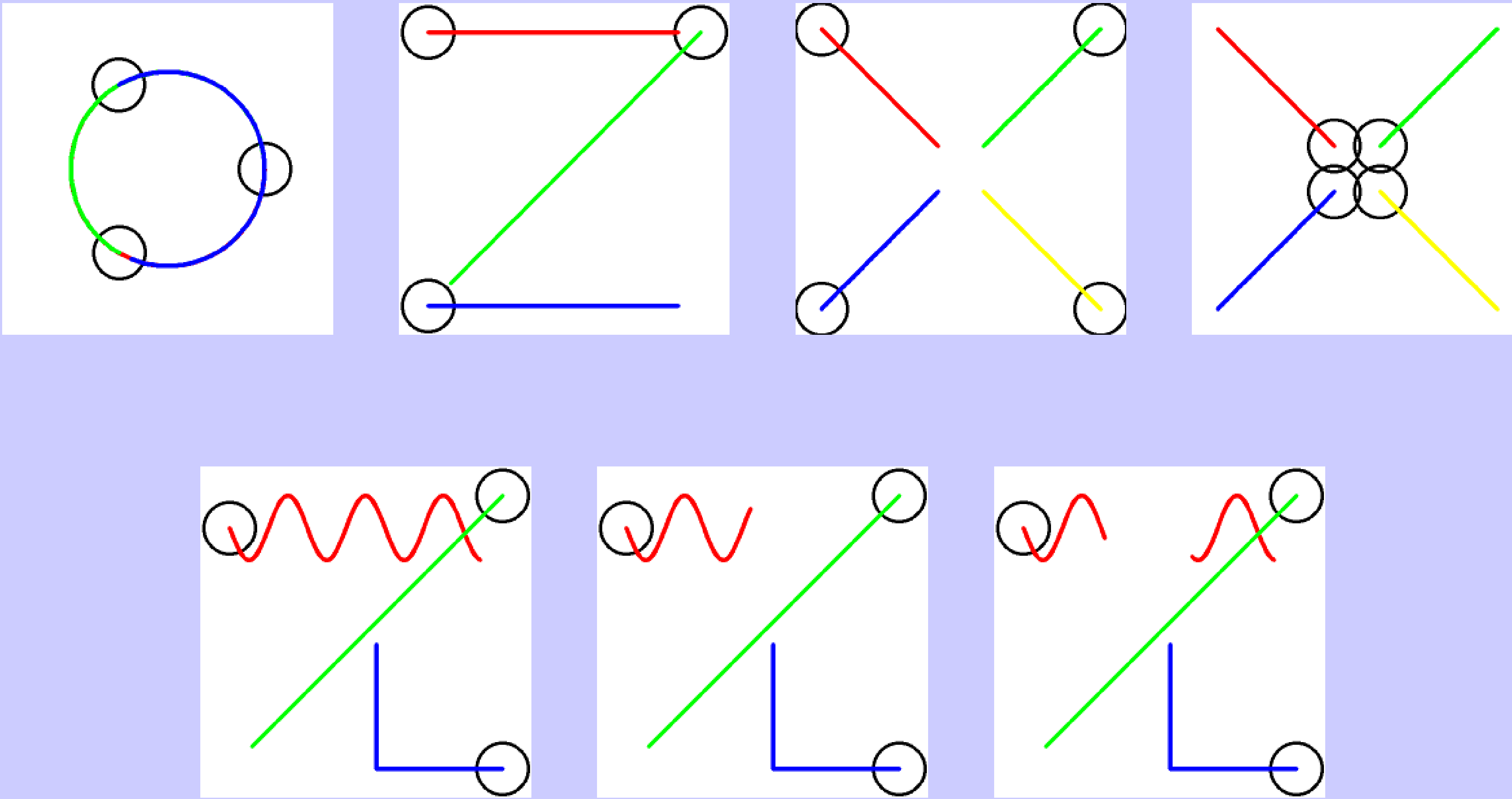
- Problem Explanation
- Extracting Trajectories
- Clustering Trajectories
- **Experimental Results**
- Conclusions

Experimental Results

- Comparison with the mean shift algorithm
  - One tracker per object
  - Manual initialization
- Experiments with
  - Simulated data sets
    - 7 image sequences
  - Real data sets
    - 5 image sequences

Experiments with simulated data sets (1)

T=130



Experiments with simulated data sets (2)

T=130






Experiments with simulated data sets (3)

- Ground truth is known
- Evaluation metrics:
  - Mean squared error (MSE)
    - Between the estimated mean trajectories and true mean trajectories
  - Accuracy (ACC)
    - Percentage of the correctly classified trajectories

Experiments with simulated data sets (4)

Problem	Our approach		Mean shift	
	MSE	ACC	MSE	ACC
Sim1	69	100%	121	100%
Sim2	10	99%	114	100%
Sim3	10	96%	114	99%
Sim4	15	97%	130	99%
Sim5	20	100%	118	100%
Sim6	29	100%	74	100%
Sim7	41	99%	lost	lost

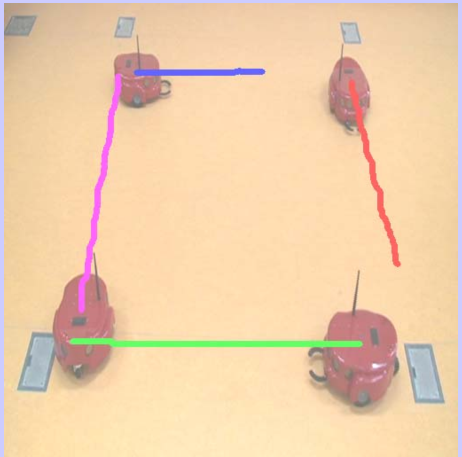
Experiments with simulated data sets (5)

: ground truth
: our approach
: mean shift

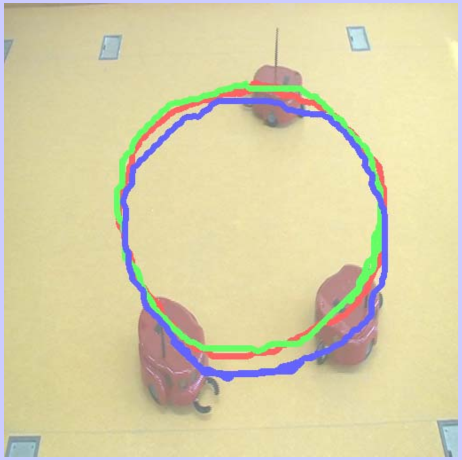
Experiments with real data sets (1)

amigobots

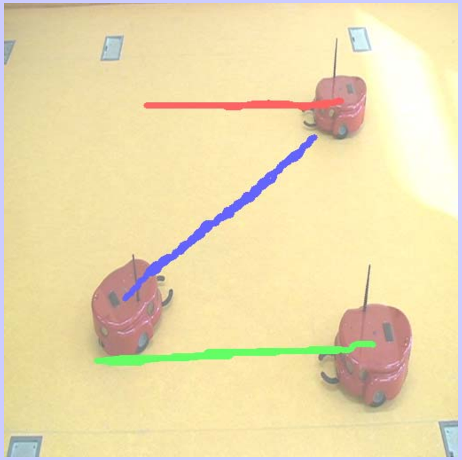
T=250



T=680

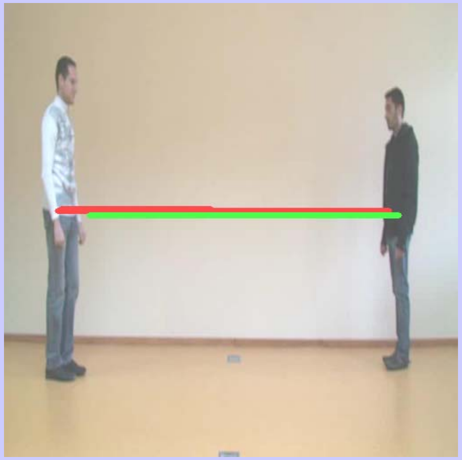


T=500



people

T=485



T=636

Experiments with real data sets (2)

amigobots

T=250

T=680

T=500

people

T=485

T=636

Experiments with real data sets (3)

Amigo bots

: **ground truth**
: **our approach**
: **mean shift**

Experiments with real data sets (4)

Experiments with real data sets (5)

: ground truth
: our approach
: mean shift

Experiments with real data sets (6)

Outline

- Problem Explanation
- Extracting Trajectories
- Clustering Trajectories
- Experimental Results
- **Conclusions**

Conclusions

- Complete framework for
  - Extracting trajectories
  - Tracking objects (clustering trajectories)
- Contribution:
  - **Simultaneous** motion segmentation and tracking
  - Address **occlusions**
  - Handle trajectories with **variable length**
- Future work
  - Initialization strategy
  - Other features (SIFT, HoG)
  - Experiments with other real datasets (surveillance)

Thank you