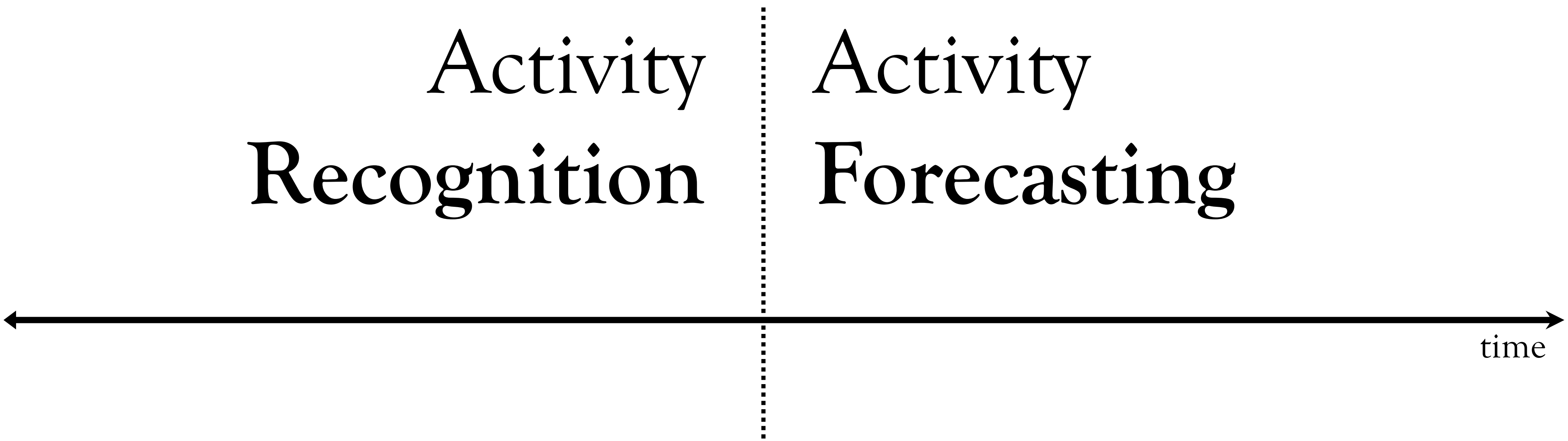
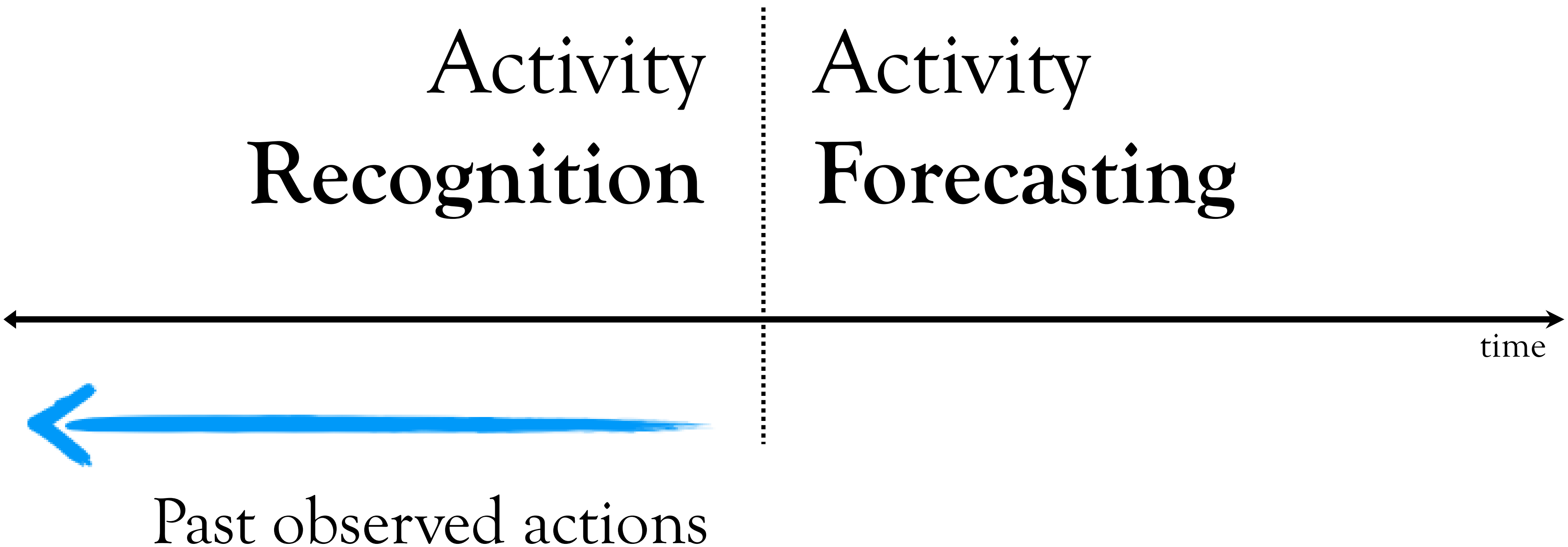
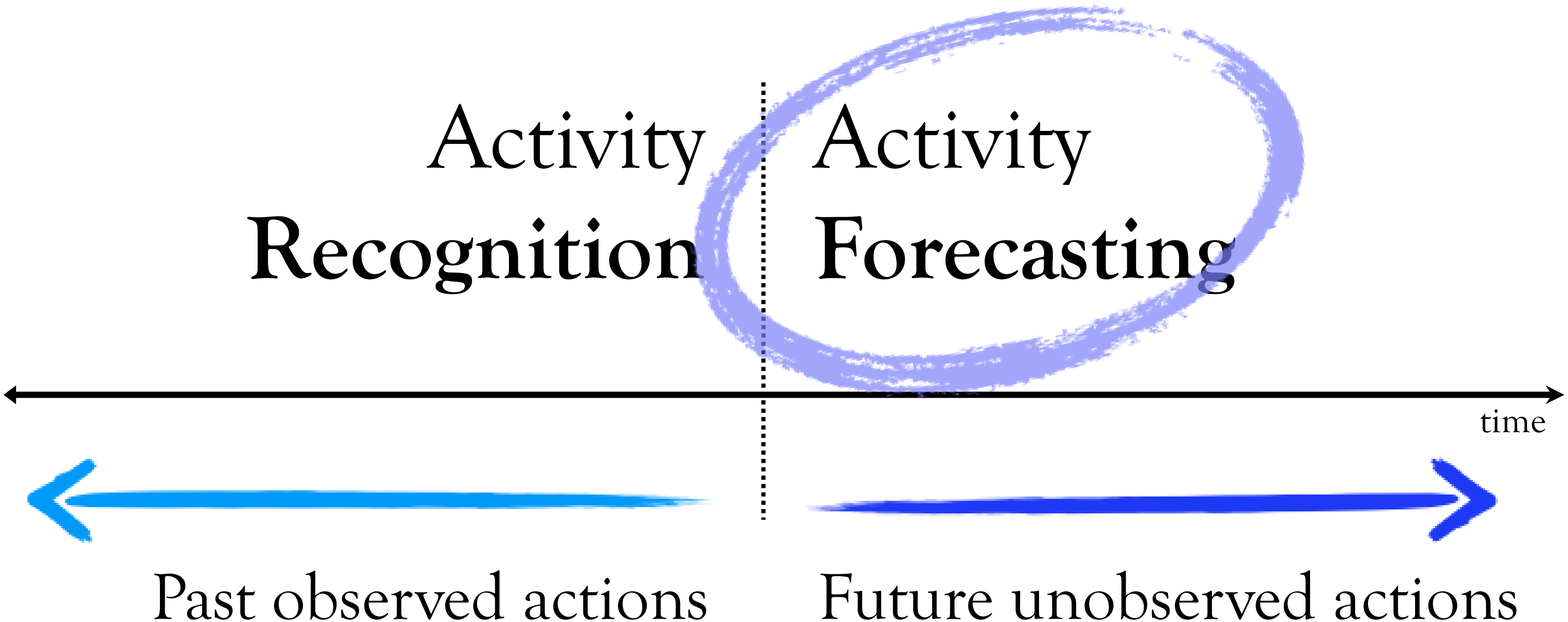
Activity Forecasting

Kris Kitani, Brian Ziebart*, Drew Bagnell, Martial Hebert
Carnegie Mellon University, *The University of Illinois at Chicago

Carnegie Mellon
THE ROBOTICS INSTITUTE


Activity
Recognition
Activity
Forecasting
time


Activity
Recognition
Activity
Forecasting
time
Past observed actions


Activity
Recognition
Activity
Forecasting
time
Past observed actions
Future unobserved actions

Activity Forecasting

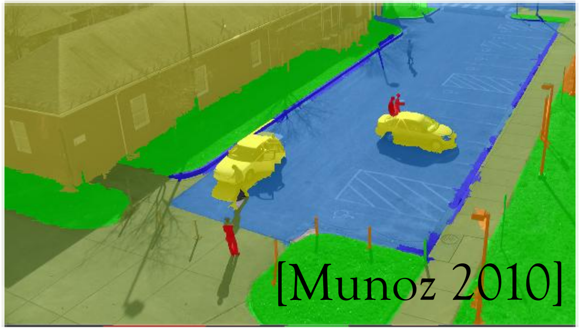
Activity Forecasting

Activity Forecasting

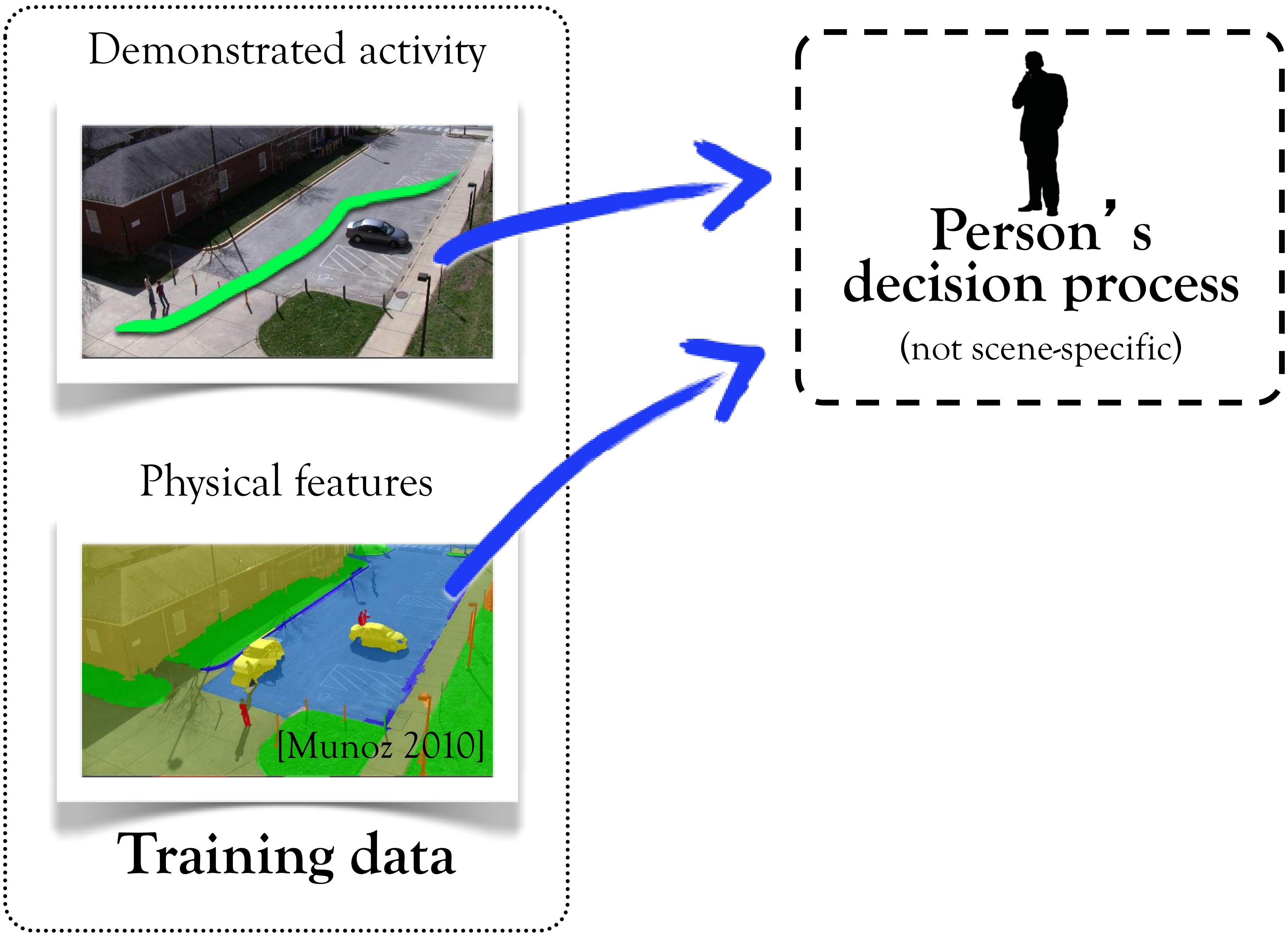
Discovery of subtle yet natural interactions

Demonstrated activity

Physical features

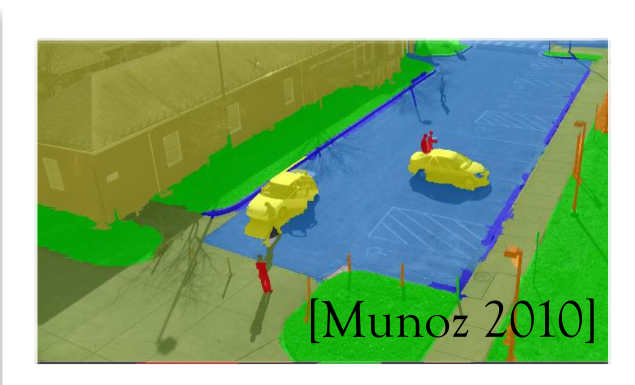


Training data


Demonstrated activity
Physical features
[Munoz 2010]
Training data
Person’s
decision process
(not scene-specific)

Demonstrated activity

Physical features



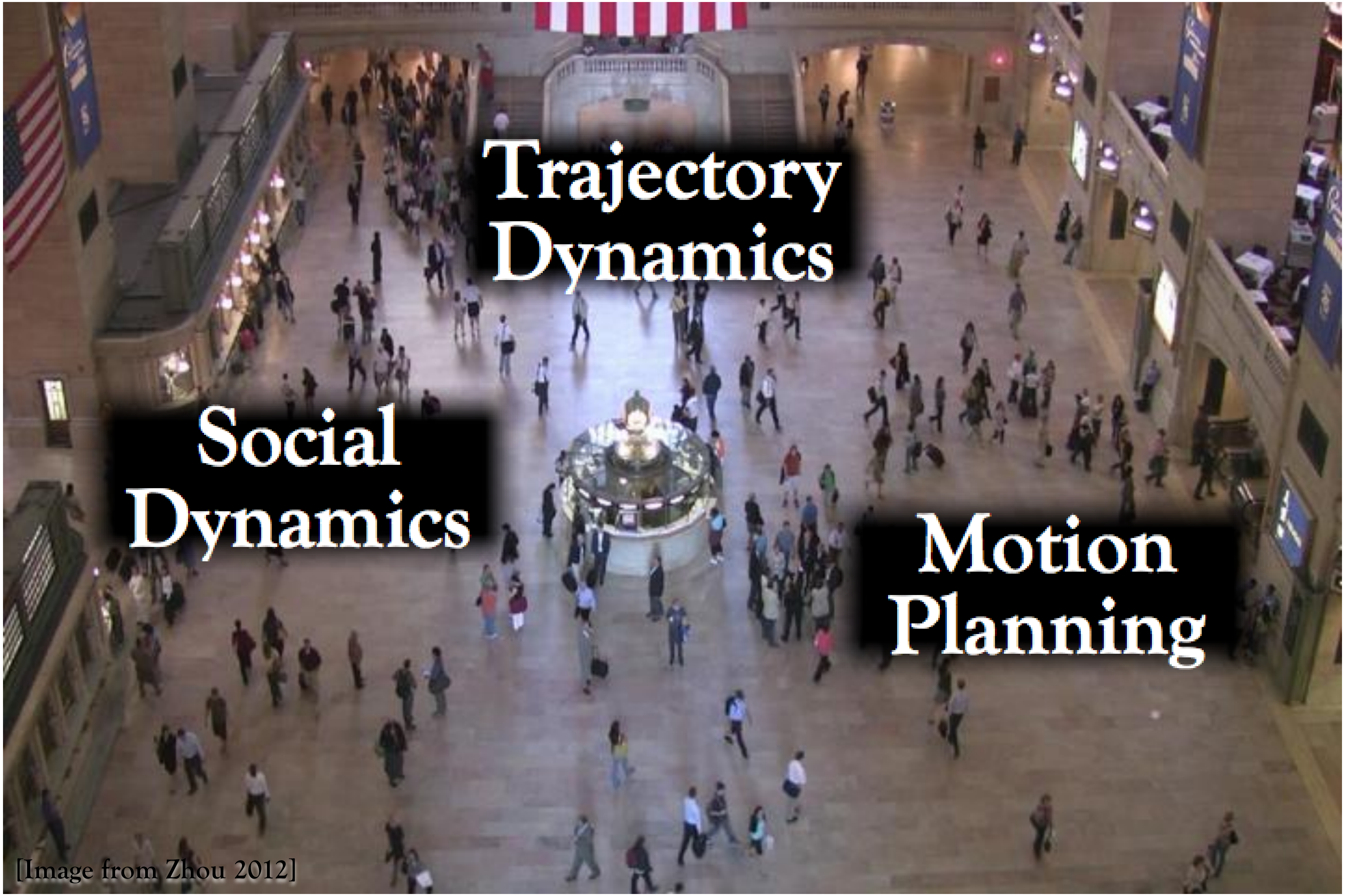
Training data

Person's decision process

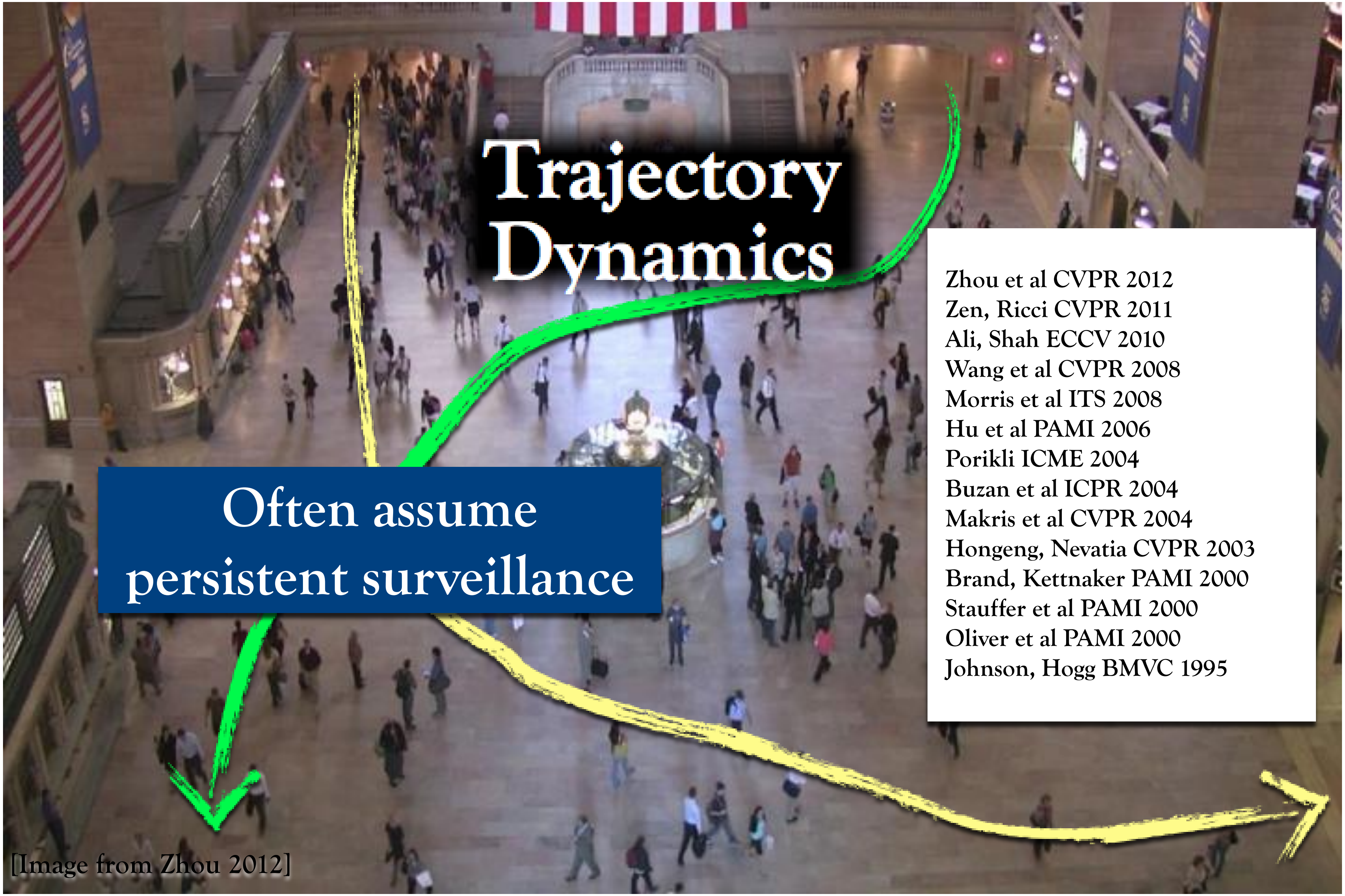
(not scene-specific)

Forecast activities in novel scene

Trajectory-based Activity Analysis

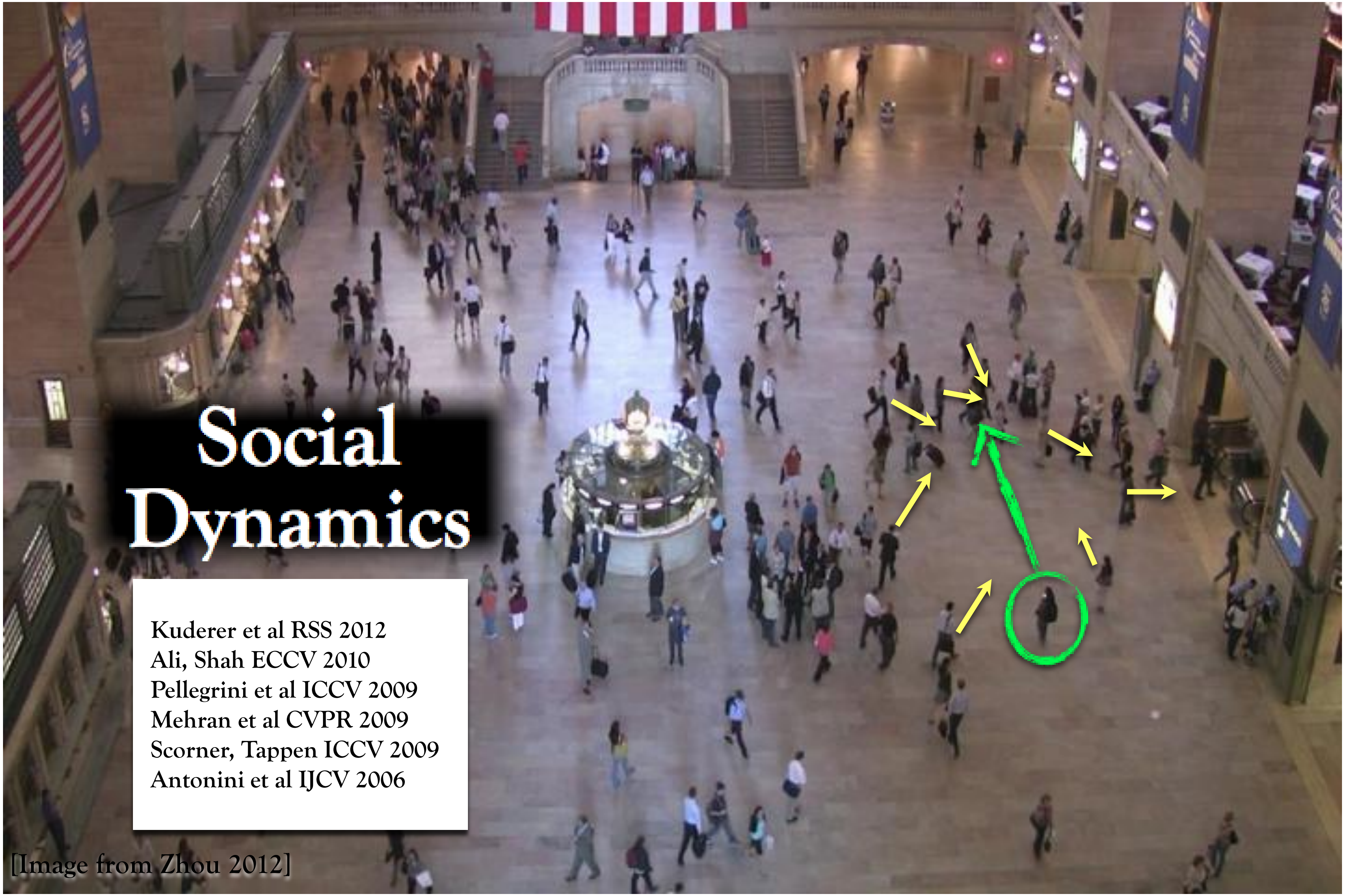


Trajectory-based Activity Analysis

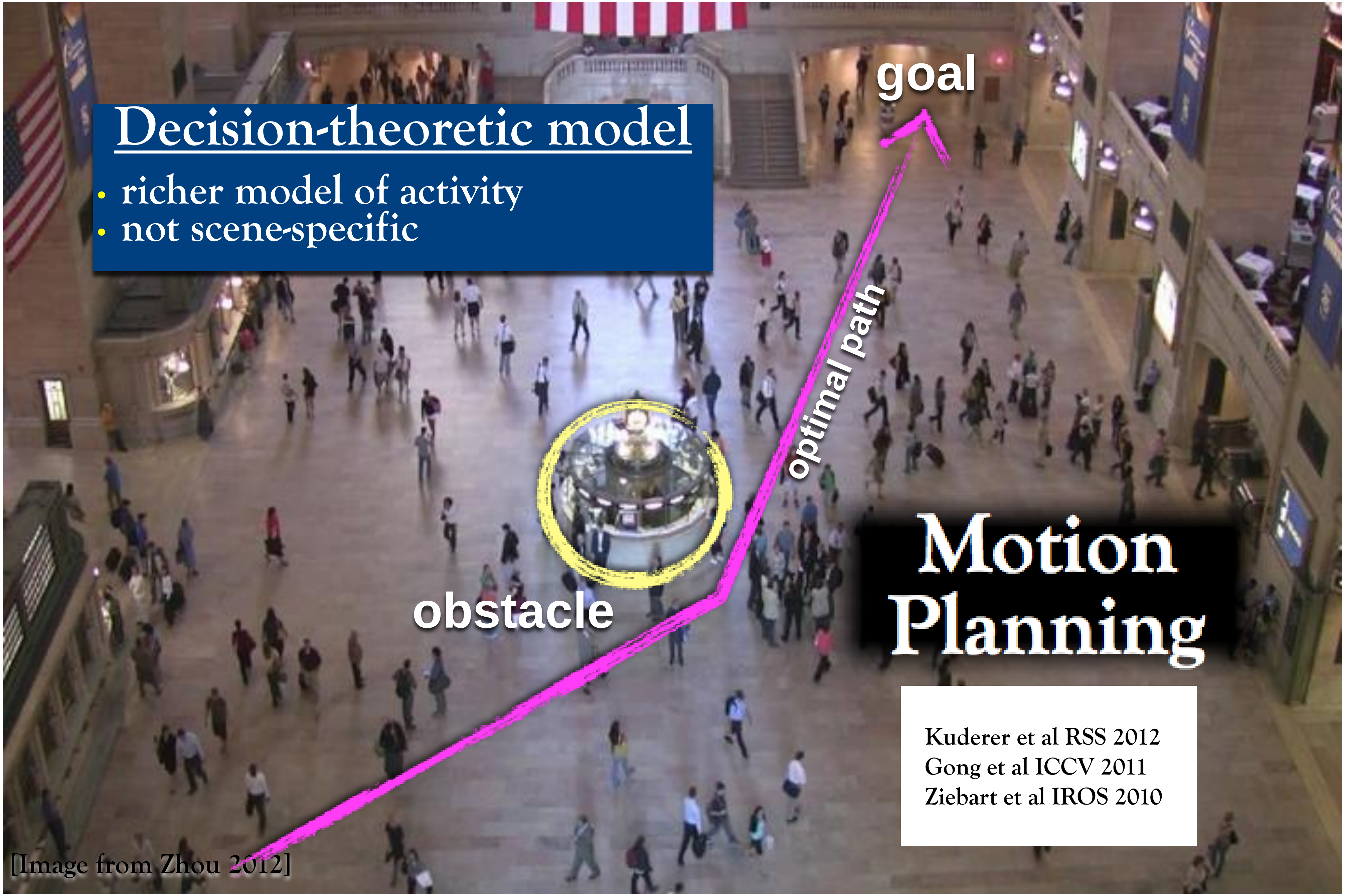


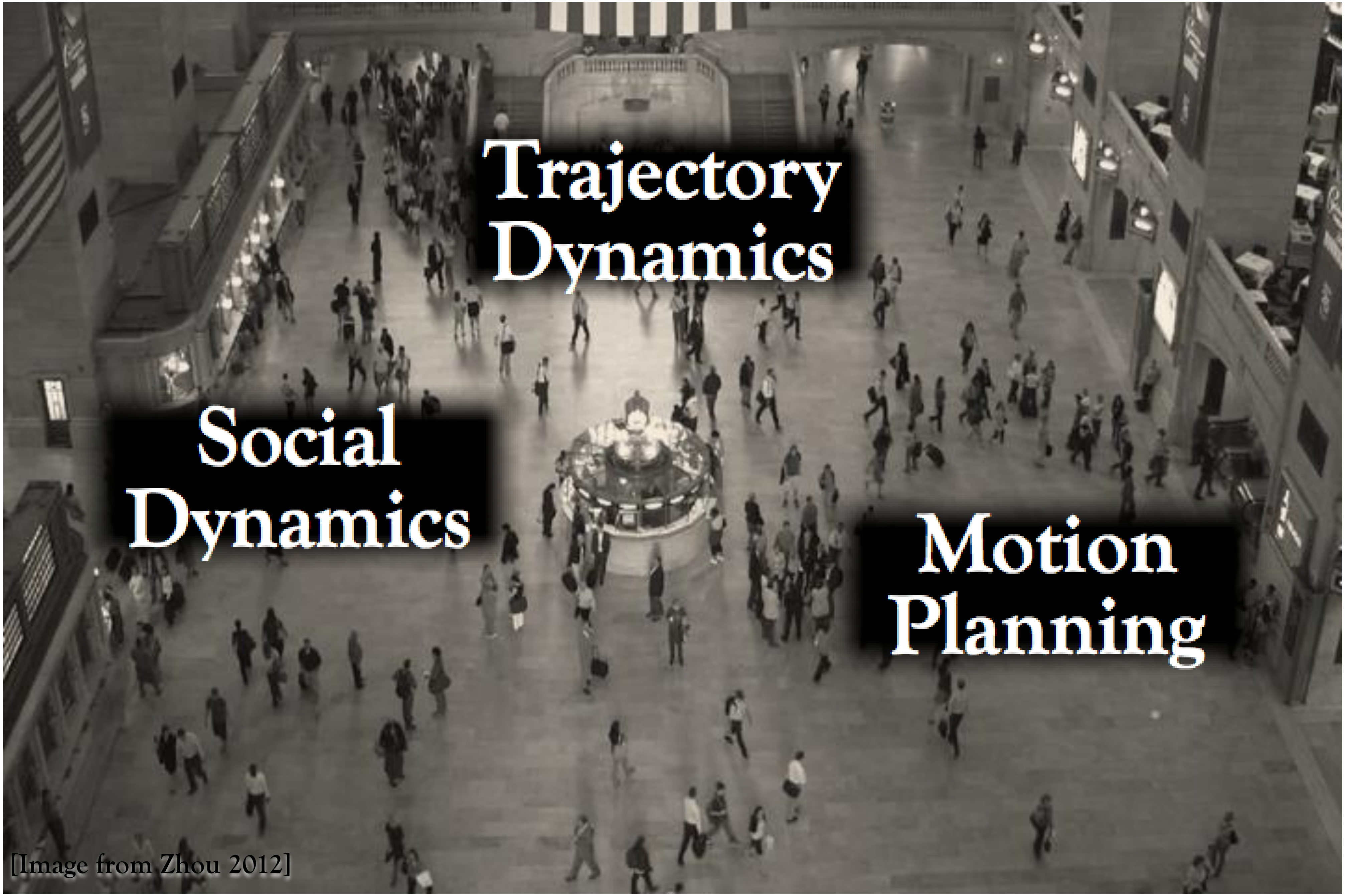
[Image from Zhou 2012]

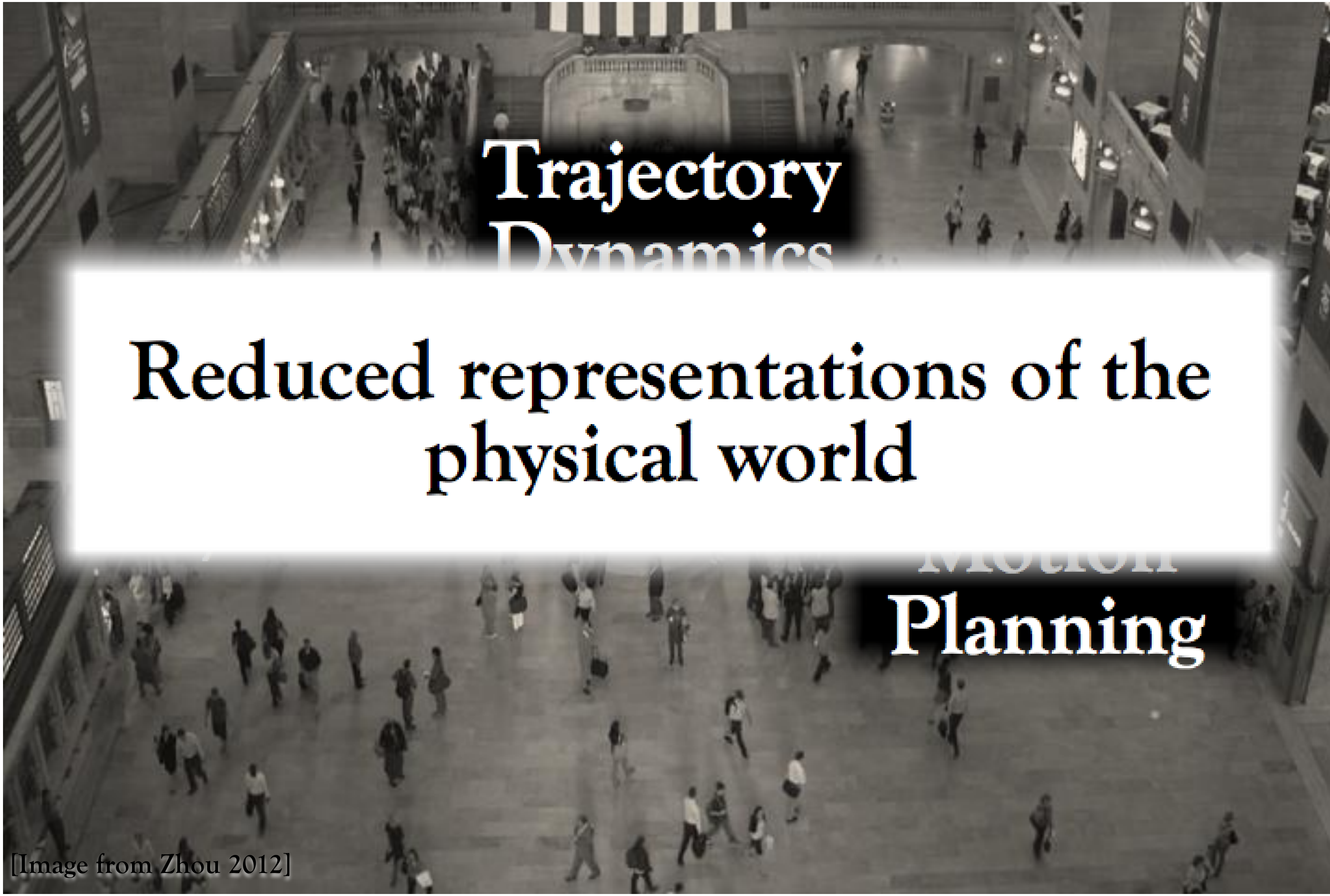
Trajectory-based Activity Analysis



Trajectory-based Activity Analysis



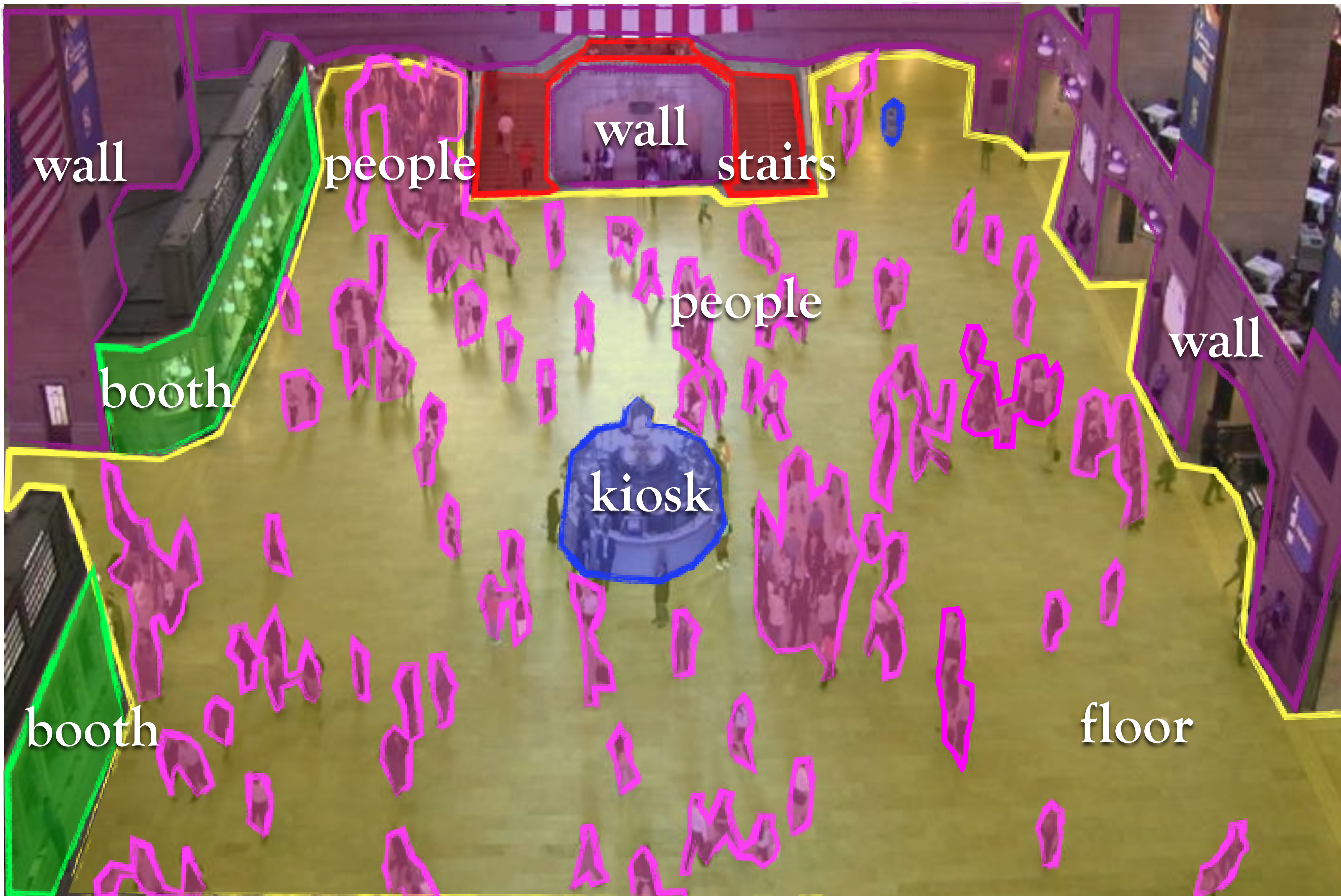

Trajectory
Dynamics
Social
Dynamics
Motion
Planning
[Image from Zhou 2012]



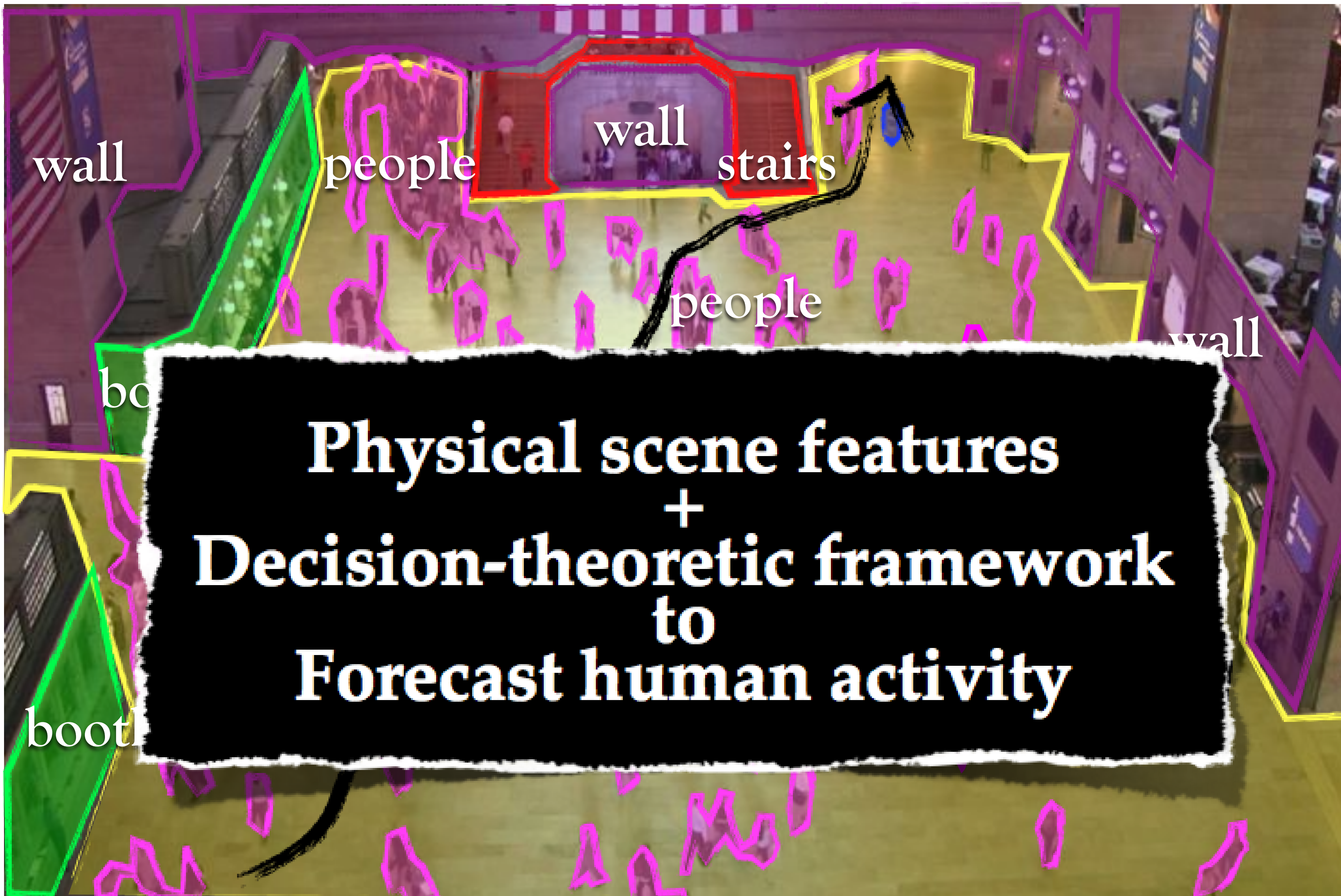
Reduced representations of the physical world

[Image from Zhou 2012]

[Image from Zhou 2012]


wall
people
wall
stairs
people
wall
booth
kiosk
booth
floor

Propose to model human activity…



Towards a decision-theoretic approach

Dynamics

$$p(x'|x)$$

'motion model'

[Kalman 1960]
[Baum 1966]

Towards a decision-theoretic approach

how?

Dynamics

$$p(x'|x)$$

'motion model'

[Kalman 1960]
[Baum 1966]

Towards a decision-theoretic approach

how? 

 why?

Dynamics

$p(x'|x)$

'motion model'

[Kalman 1960]
[Baum 1966]

Decisions

$p(a|x)$

'policy'

[Bellman 1957]
[Howard 1960]

Our Approach:

Activity sequence generated by an Markov Decision Process (MDP)

$$S = \{x_0, a_0, R(x_0),\ x_1, a_1, R(x_1), \ldots,\ x_g, a_g, R(x_g)\}$$

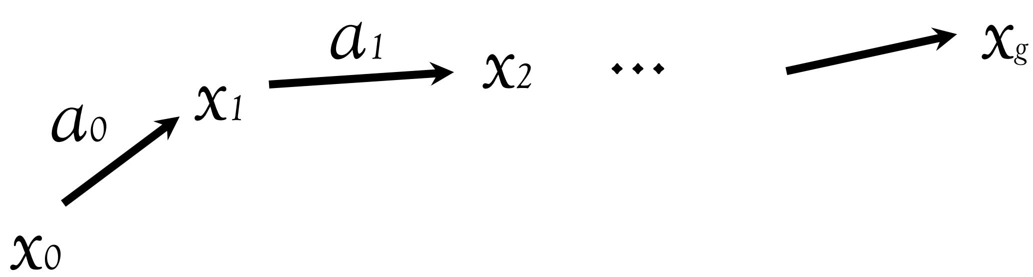
(state, action, reward)

Our Approach:

Activity sequence generated by an Markov Decision Process (MDP)

$$S = \{x_0, a_0, R(x_0),\ x_1, a_1, R(x_1), \ldots,\ x_g, a_g, R(x_g)\}$$

(state, action, reward)



Our Approach:

Activity sequence generated by an Markov Decision Process (MDP)

$$S = \{x_0, a_0, R(x_0),\ x_1, a_1, R(x_1), \ldots,\ x_g, a_g, R(x_g)\}$$

(state, action, reward)

Sequence determined by **Policy**

$$p(a|x)$$

Our Approach:

Activity sequence generated by an Markov Decision Process (MDP)

$$S = \{x_0, a_0, R(x_0),\ x_1, a_1, R(x_1), \ldots,\ x_g, a_g, R(x_g)\}$$

(state, action, reward)

Sequence determined by **Policy**

$$p(a|x) \propto \exp\left\{ \underbrace{R(x)}_{\text{Reward}} + \sum_{x'} \underbrace{p(x'|x,a)V(x')}_{\text{Expected future payoff}} \right\}$$

[Ziebart et al 2008]

Our Approach:

Activity sequence generated by an Markov Decision Process (MDP)

$$S = \{x_0, a_0, R(x_0),\ x_1, a_1, R(x_1), \ldots,\ x_g, a_g, R(x_g)\}$$

(state, action, reward)

Sequence determined by **Policy**

$$p(a|x) \propto \exp\left\{\underbrace{R(x)}_{\text{Reward}} + \sum_{x'} \underbrace{p(x'|x,a)V(x')}_{\text{Expected future payoff}}\right\}$$

[Ziebart et al 2008]

Policy determined by **Reward function**

$$V(x) = \operatorname{soft}\max_a Q(x,a)$$

$$Q(x,a) = R(x) + \sum_{x'} p(x'|x,a)V(x')$$

Our Approach:

Activity sequence generated by an Markov Decision Process (MDP)

$$S = \{x_0, a_0, R(x_0),\ x_1, a_1, R(x_1), \ldots,\ x_g, a_g, R(x_g)\}$$

(state, action, reward)

Sequence determined by **Policy**

$$p(a|x) \propto \exp\left\{ \underbrace{R(x)}_{\text{Reward}} + \sum_{x'} \underbrace{p(x'|x,a)V(x')}_{\text{Expected future payoff}} \right\}$$

[Ziebart et al 2008]

Policy determined by **Reward function**

$$V(x) = \text{soft}\max_a Q(x,a)$$

$$Q(x,a) = R(x) + \sum_{x'} p(x'|x,a)V(x')$$

Our Approach:

Activity sequence generated by an Markov Decision Process (MDP)

$$S = \{x_0, a_0, R(x_0),\ x_1, a_1, R(x_1), \ldots,\ x_g, a_g, R(x_g)\}$$

(state, action, reward)

Sequence determined by **Policy**

$$p(a|x) \propto \exp\left\{ \underbrace{R(x)}_{\text{Reward}} + \sum_{x'} \underbrace{p(x'|x,a)V(x')}_{\text{Expected future payoff}} \right\}$$

[Ziebart et al 2008]

Policy determined by **Reward function**

$$V(x) = \operatorname{soft\,max}_a Q(x,a)$$

$$Q(x,a) = R(x) + \sum_{x'} p(x'|x,a)V(x')$$

Infer the reward function from observed sequences

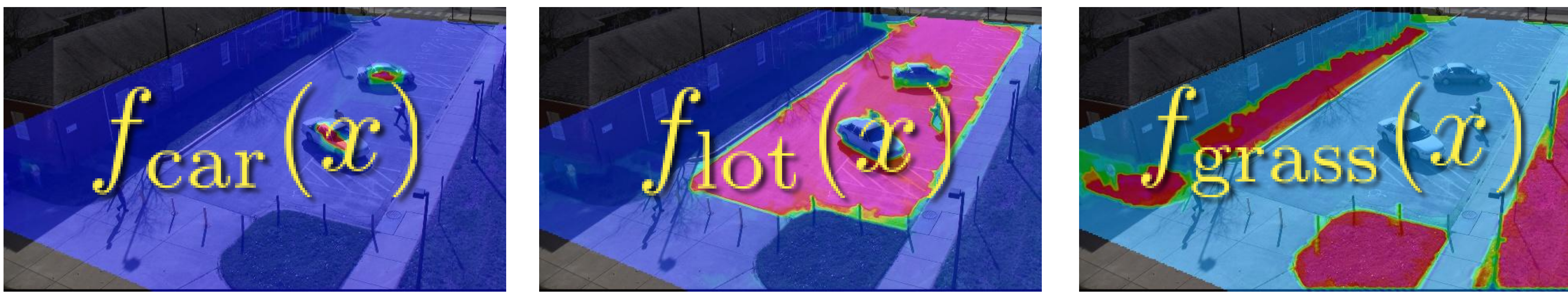
Reward function parameterization

$$R(x; \boldsymbol{\theta}) = \sum_n \theta_n f_n(x)$$

Reward function parameterization

$$R(x;\boldsymbol{\theta}) = \sum_n \theta_n f_n(x)$$

$$R(x;\boldsymbol{\theta}) = \theta_{\mathrm{car}} \cdot f_{\mathrm{car}}(x) + \theta_{\mathrm{lot}} \cdot f_{\mathrm{lot}}(x) + \theta_{\mathrm{grass}} \cdot f_{\mathrm{grass}}(x)$$

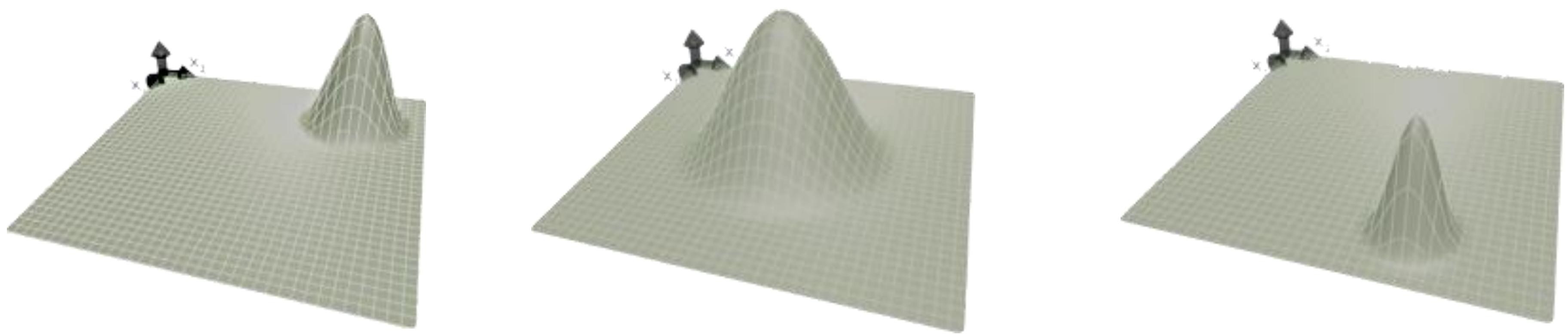


[Munoz et al 2010]

Reward function parameterization

$$R(x;\boldsymbol{\theta}) = \sum_n \theta_n f_n(x)$$

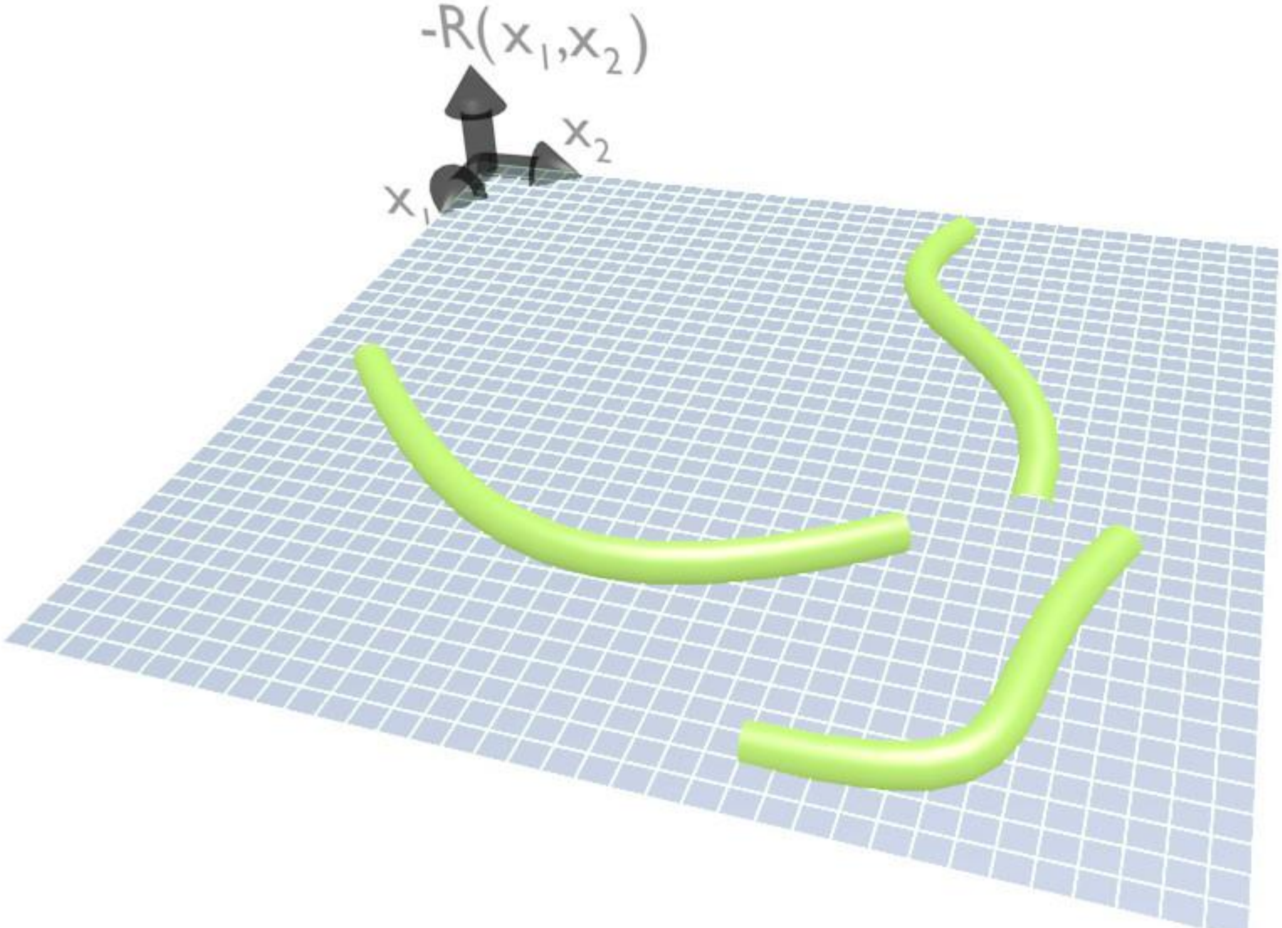
$$R(x;\boldsymbol{\theta}) = \theta_{\text{car}} \cdot f_{\text{car}}(x) + \theta_{\text{lot}} \cdot f_{\text{lot}}(x) + \theta_{\text{grass}} \cdot f_{\text{grass}}(x)$$

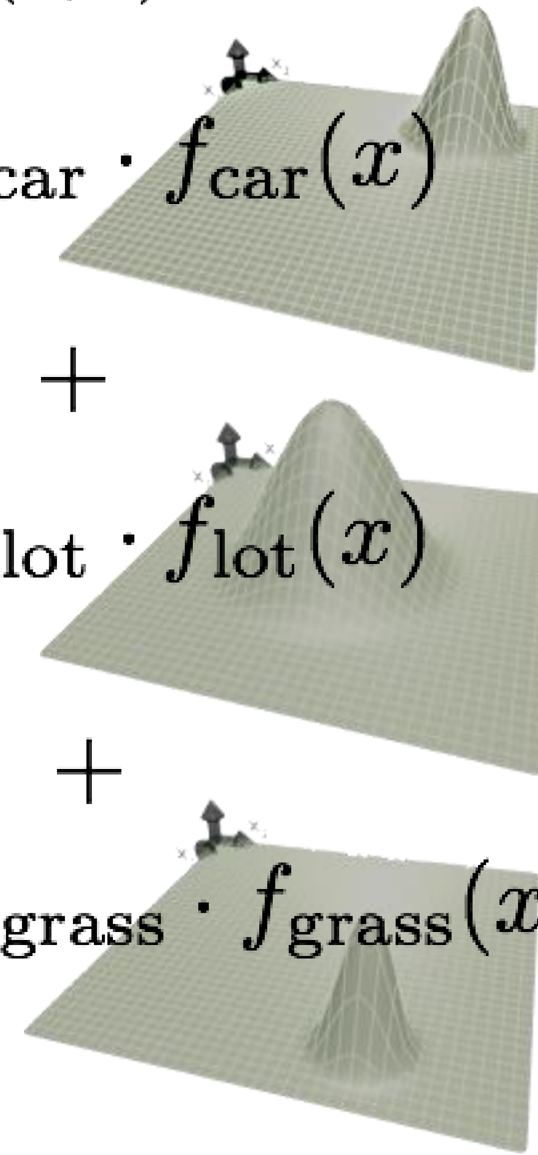


Learn the weights of the physical features

Input: Trajectories & feature responses

Output: Reward weights





$$R(x;\boldsymbol{\theta}) = \theta_{\text{car}} \cdot f_{\text{car}}(x) + \theta_{\text{lot}} \cdot f_{\text{lot}}(x) + \theta_{\text{grass}} \cdot f_{\text{grass}}(x)$$

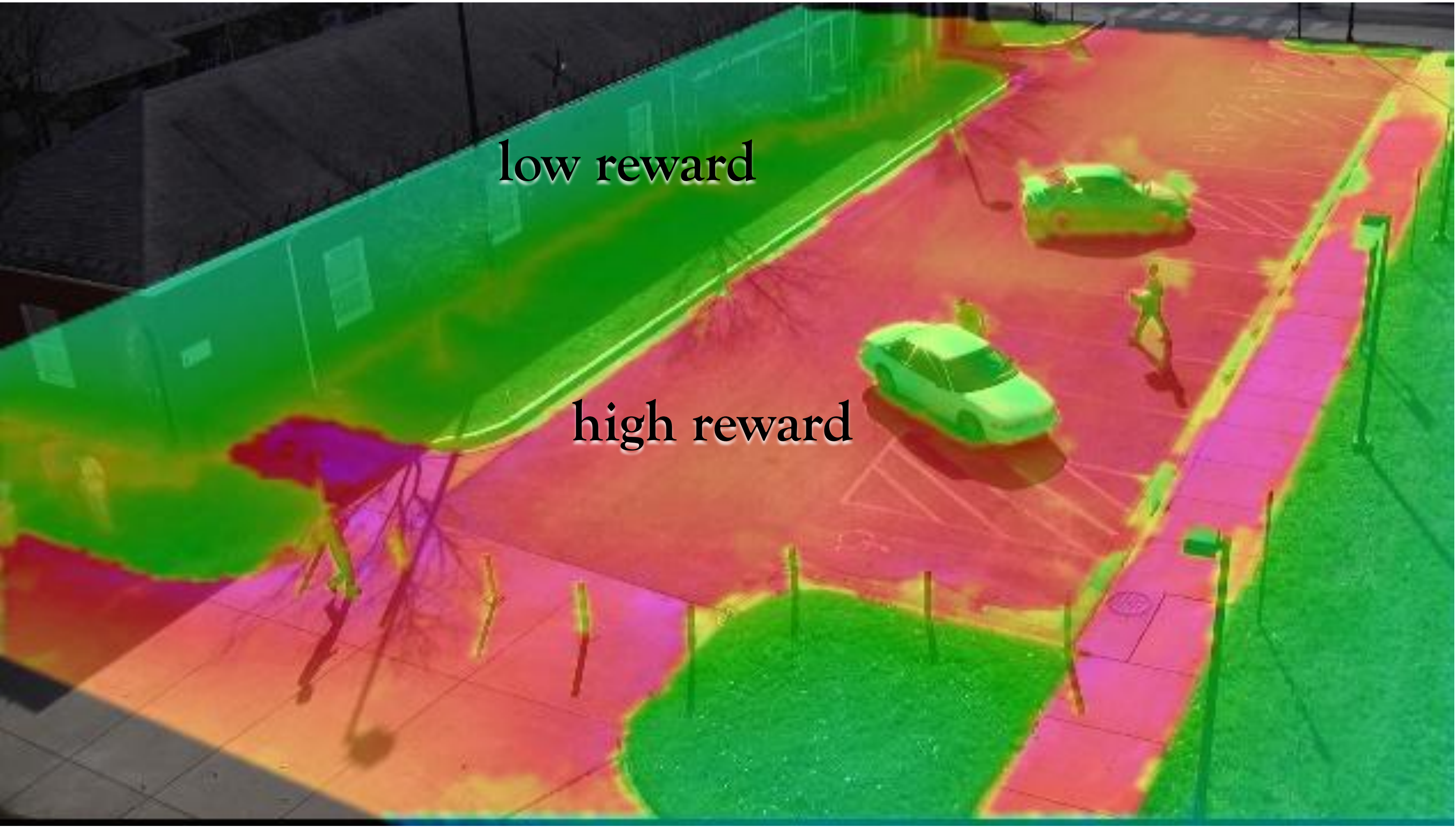
Learn the reward function via

Inverse Optimal Control

[Abbeel & Ng 2004, Ziebart et al 2008]

[Graphics by Paul Vernaza]

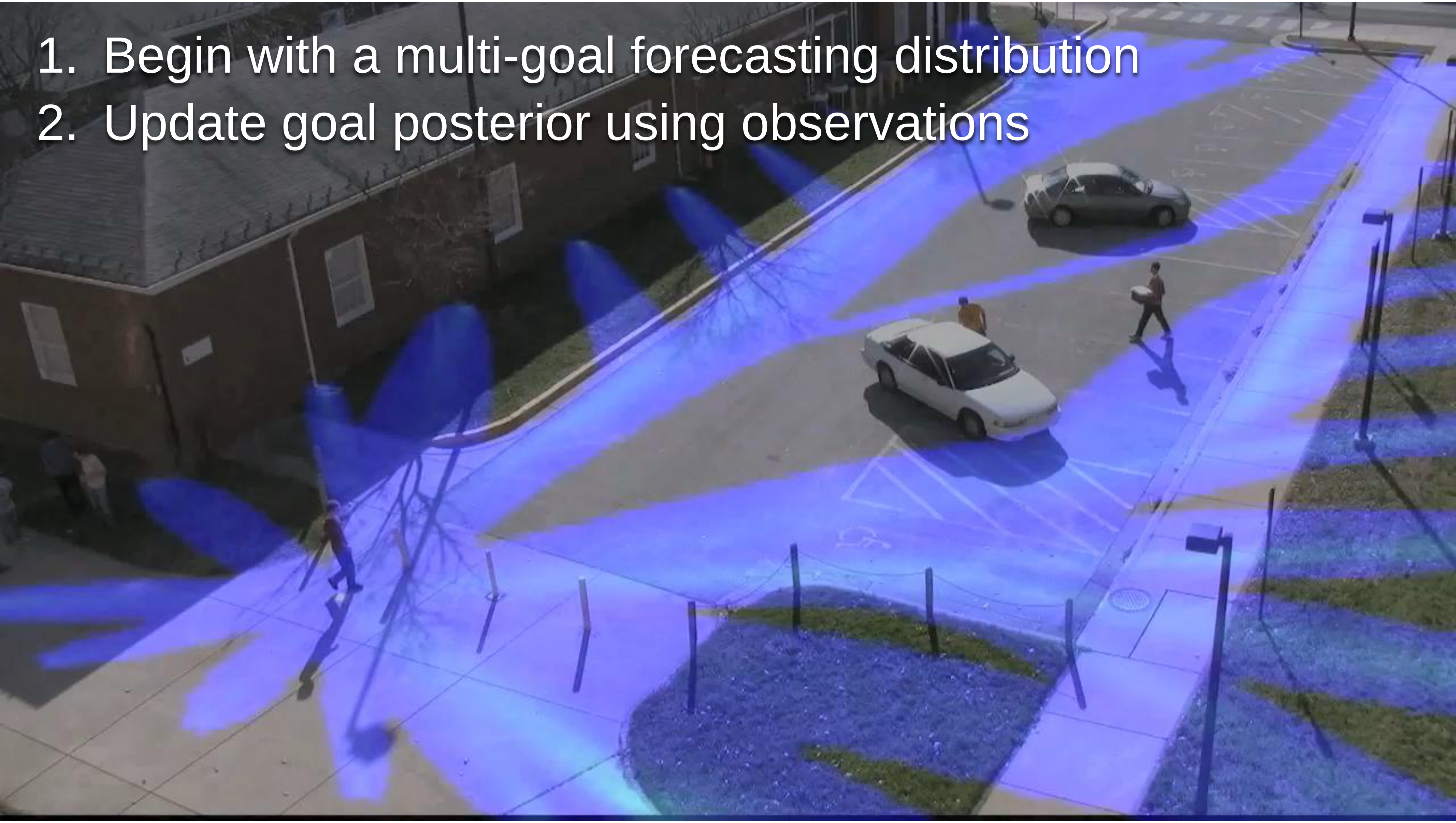
Reward Function



Activity Forecasting

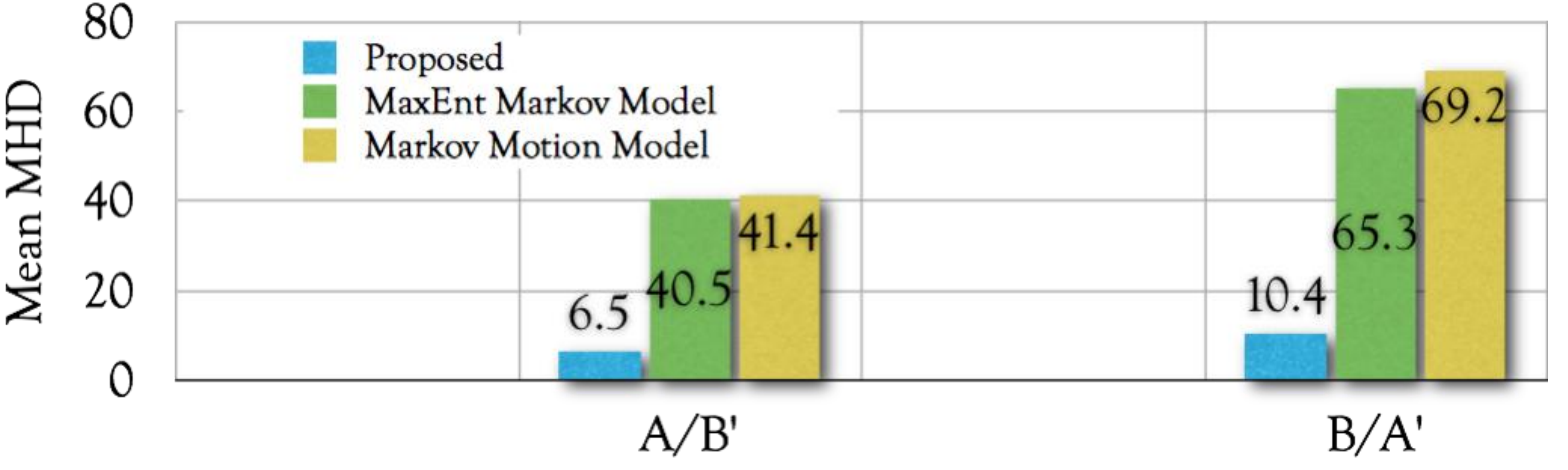
Destination Forecasting

1. Begin with a multi-goal forecasting distribution
2. Update goal posterior using observations



Dataset: 92 videos (A:56 / B:36)
Setup: 80% test, 20% train (3-fold cross validation)
Baselines: Maximum Entropy Markov Model [McCallum'00]
Markov Motion Model [Porikli' 04]
Metrics: Negative log loss,
Modified Hausdorff distance

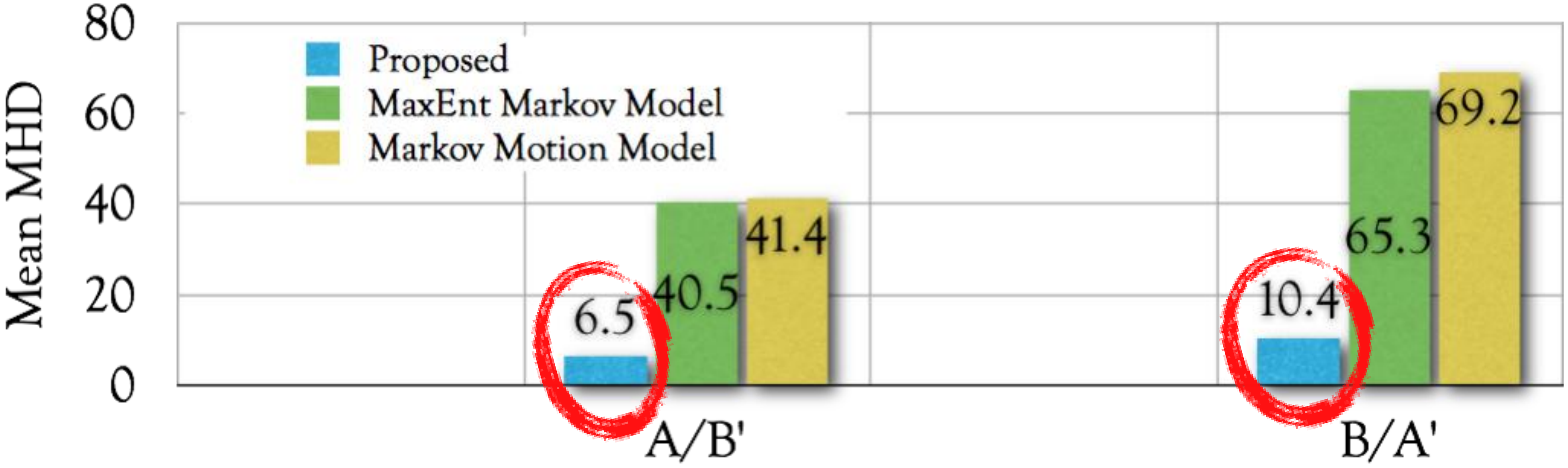
Forecasting performance



Modified Hausdorff Distance (MHD) [pixels]:
Euclidean distance between observed trajectory and sampled trajectories allowing for small temporal misalignment

Dataset: 92 videos (A:56 / B:36)
Setup: 80% test, 20% train (3-fold cross validation)
Baselines: Maximum Entropy Markov Model [McCallum'00]
Markov Motion Model [Porikli' 04]
Metrics: Negative log loss,
Modified Hausdorff distance

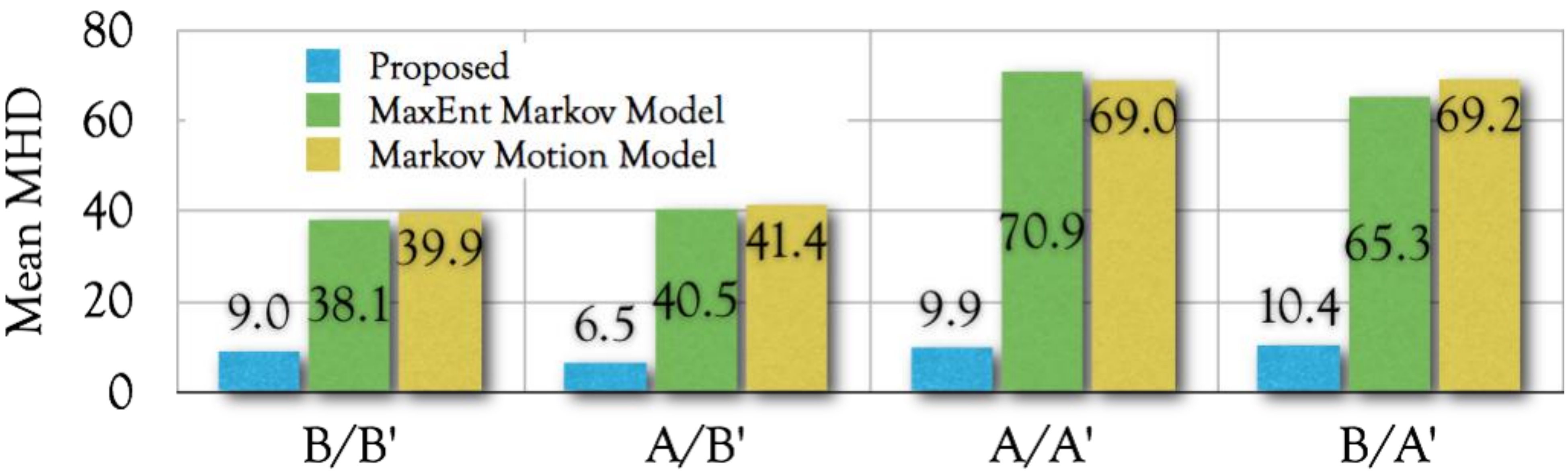
Forecasting performance



Modified Hausdorff Distance (MHD) [pixels]:
Euclidean distance between observed trajectory and sampled trajectories allowing for small temporal misalignment

Dataset: 92 videos (A:56 / B:36)
Setup: 80% test, 20% train (3-fold cross validation)
Baselines: Maximum Entropy Markov Model [McCallum'00]
Markov Motion Model [Porikli’ 04]
Metrics: Negative log loss,
Modified Hausdorff distance

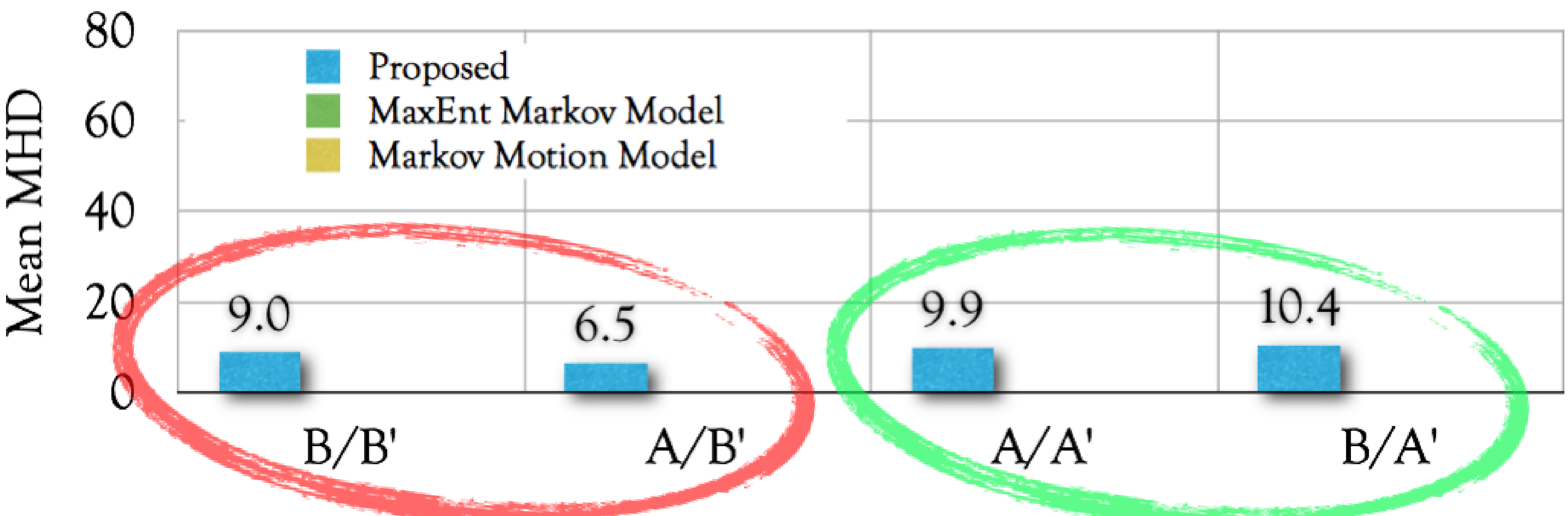
Forecasting performance



Modified Hausdorff Distance (MHD) [pixels]:
Euclidean distance between observed trajectory and sampled trajectories allowing for small temporal misalignment

Dataset: 92 videos (A:56 / B:36)
Setup: 80% test, 20% train (3-fold cross validation)
Baselines: Maximum Entropy Markov Model [McCallum'00]
Markov Motion Model [Porikli' 04]
Metrics: Negative log loss,
Modified Hausdorff distance

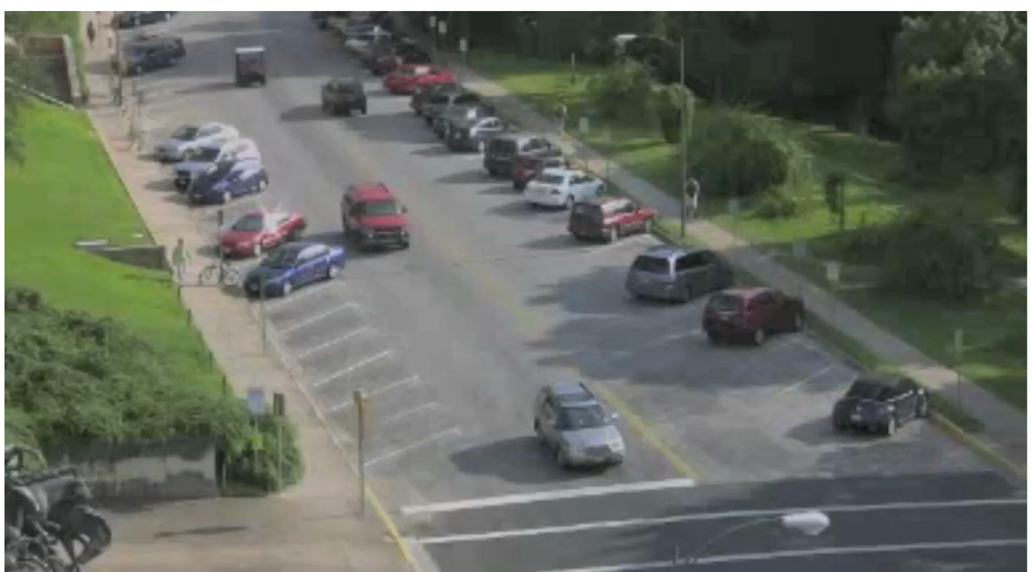
Forecasting performance



Modified Hausdorff Distance (MHD) [pixels]:
Euclidean distance between observed trajectory and sampled trajectories allowing for small temporal misalignment

Forecasting in new scenes






Activity Forecasting

Kris Kitani, Brian Ziebart*, Drew Bagnell, Martial Hebert
Carnegie Mellon University, *The University of Illinois at Chicago

Carnegie Mellon
THE ROBOTICS INSTITUTE