FastSLAM2.0

Adel Fakih

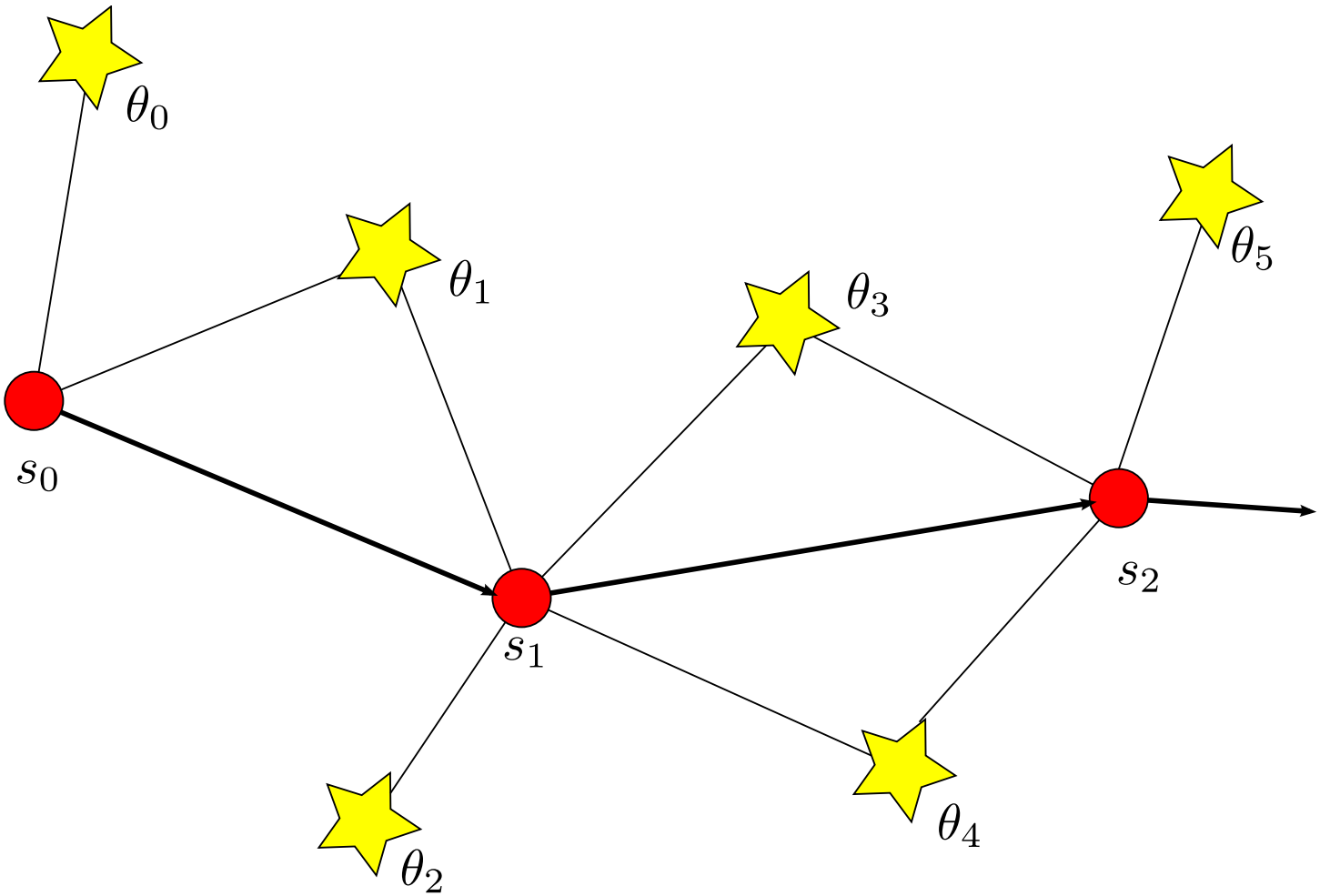
afakih@engmail.uwaterloo.ca

May 30, 2010

Outline

Outline

Simultaneous Localization and Mapping



University of Waterloo

Simultaneous Localization and Mapping

Determine the distribution $p(s_t, \theta^t | z^t, u_t, n^t)$

- s_t: Robot pose at time t
- $\theta^t = \theta_0, ..., \theta_t$: Landmarks
- $z^t = z_0, ..., z_t$: Measurements
- u_t: robot control
- $n^t = n_0, ..., n_t$ with $n_i = 1$ if the i^{th} feature is perceived at time t
- Measurement equation: $z_t = g(s_t, \theta_t, n_t) + \text{noise} \sim \mathcal{N}(0,\ R_t)$
- Transition equation: $s_t = h(s_{t-1}, u_t) + \text{noise} \sim \mathcal{N}(0,\ P_t)$

Outline

Rao-BlackWellized Particle Filtering

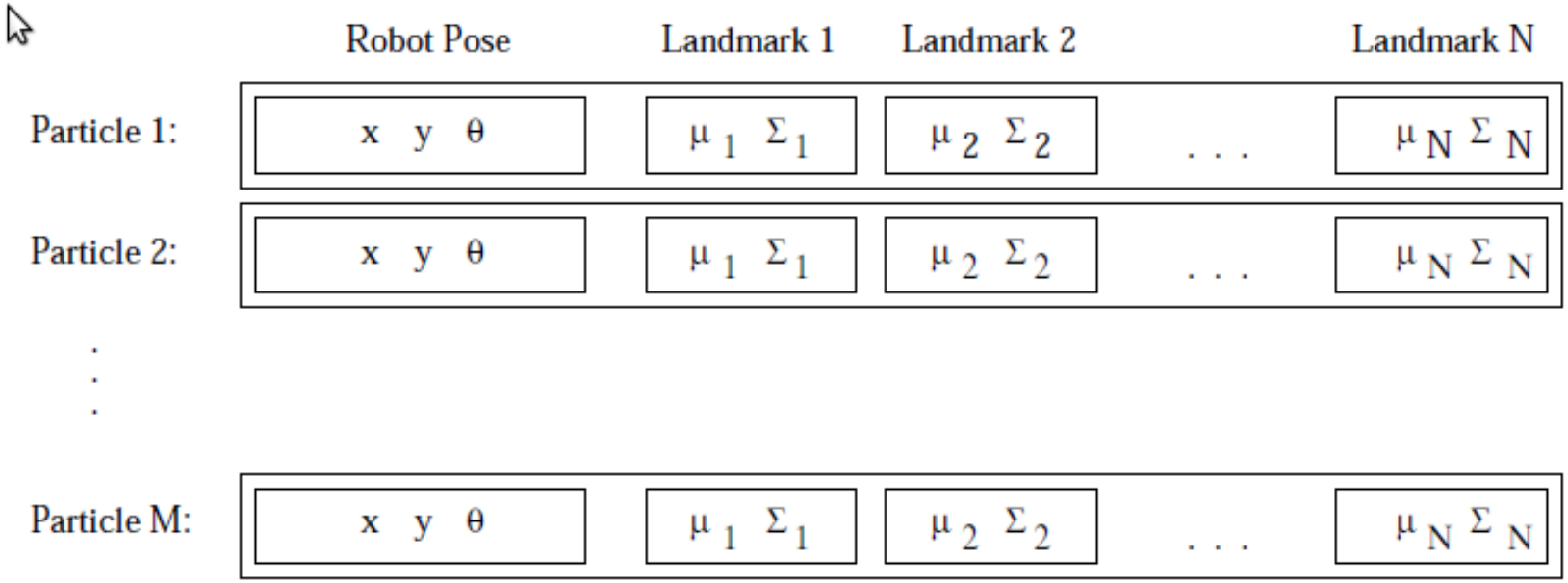
Factored Representation

- Landmarks are independent given the robot pose:

$$p(s_t, \theta | z^t, u_t, n^t) = \underbrace{p(s_t | z^t, u_t, n^t)}_{\text{path posterior}} \Pi \underbrace{p(\theta_k | s_t, z^t, u_t, n_t)}_{\text{landmark estimators}}$$

- The pose can be estimated using particle filtering
- Each landmark, in each particle, can be estimated using an Extended Kalman Filter conditioned on the robot pose of the particle

FastSLAM1.0: Particles



University of Waterloo

FastSLAM1.0: Pose Estimation

Sequential Importance Sampling

- Sample from $q(s_t|s_{t-1}, z^t, u_t, n^t)$ called importance function
- Assign weights to the samples as $w_t^{[m]} = \frac{p(s_t^{[m]}|z^t,u_t,n^t)}{q(s_t^{[m]}|s_{t-1}^{[m]},z^t,u_t,n^t)}$
- FastSLAM1.0 choice for q :

$$p(s_t^{[m]}|z^{t-1}, u_t, n^t) = p(s_t^{[m]}|s_{t-1}^{[m]}, u_t, n^t)$$

(which is the prior distribution)

- $w_t^{[m]} \propto p(z^t|s_t^{[m]}, z^{t-1}, u_t, n^t)$ (which is the Likelihood)

$$w_t^{[m]} = \sum_k \underbrace{p(z_t|\theta_k^{[m]}, s_t^{[m]}, n_t)p(\theta_k^{[m]})}_{\mathcal{N}(\hat{z}_t^{[m]}, G_\theta \Sigma_{\theta_k,t-1}^{[m]} G_\theta^T + R_t)}$$

FastSLAM1.0

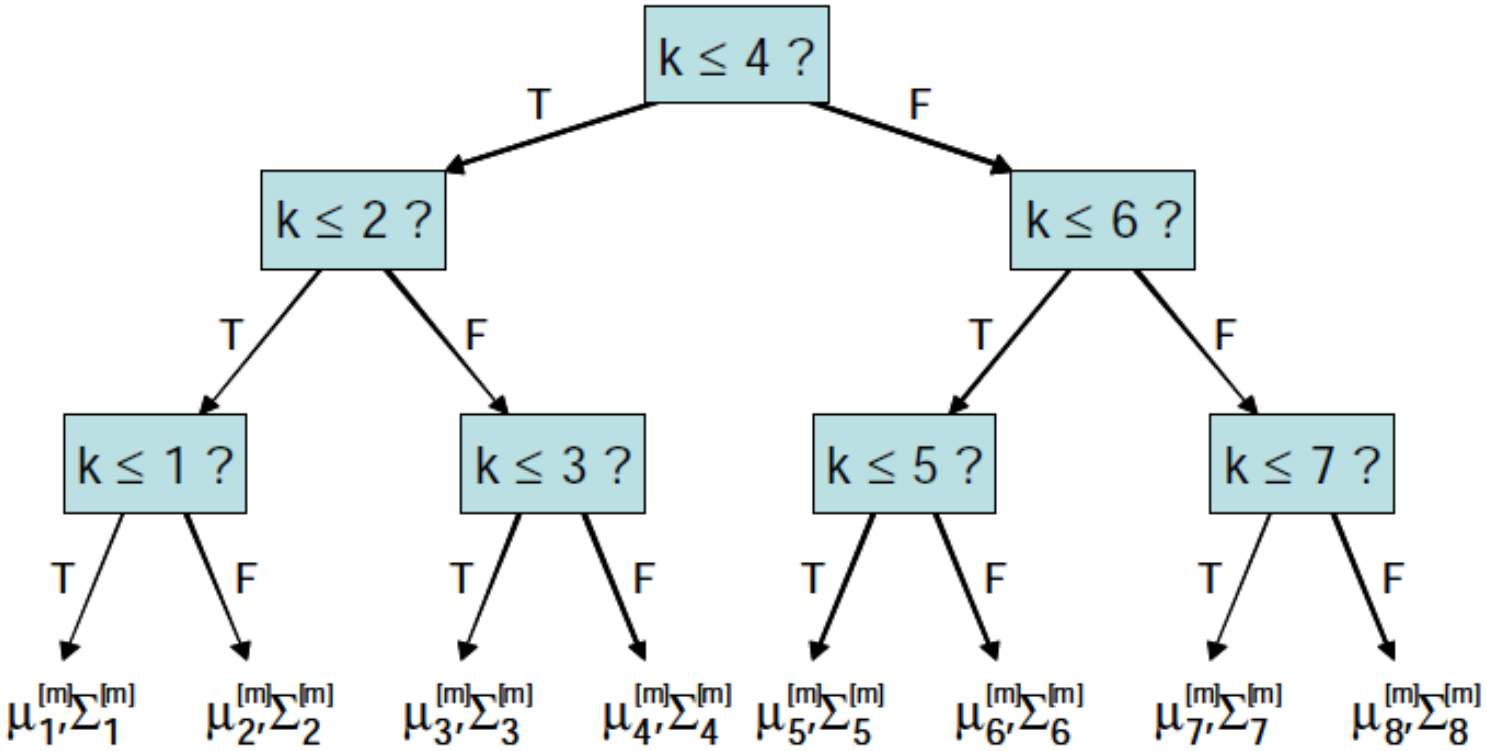
To recapitulate

- Sample new robot poses given the new control $s_t^{[m]} = h(s_{t-1}^{[m]}, u_t)$
- For each new robot pose:
  - $\longrightarrow$ Update the distribution of every landmark with new observation using Extended Kalman Filters
  - $\longrightarrow$ Assign a weight to the particle
- Resample the particles according to their weights

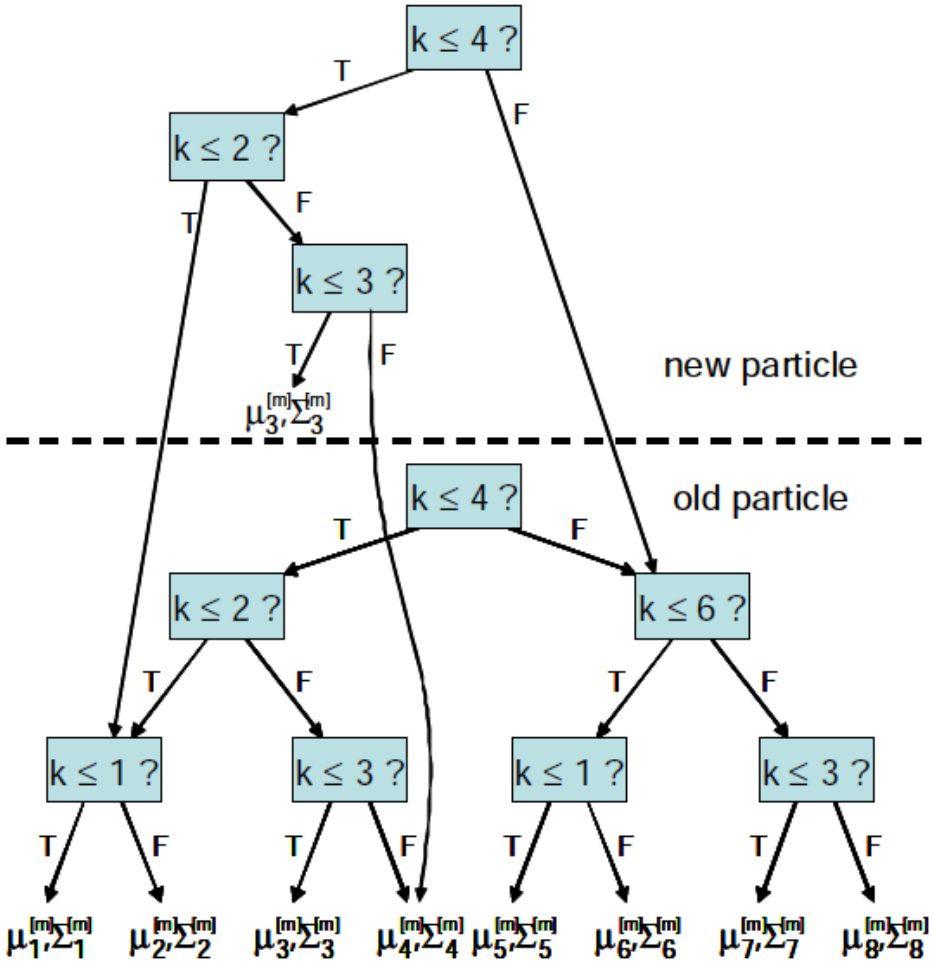
University of Waterloo

Outline

Binary Tree Data Structure



Binary Tree Data Structure



Outline

FastSLAM2.0: Optimal Importance Function

- FastSLAM1.0 samples only according to the prediction distribution and does not consider the new measurements z^t
  - $\longrightarrow$ The new measurement is only incorporated through resampling
  - $\longrightarrow$ Inefficient if the noise in the vehicle motion is large

FastSLAM2.0: Optimal Importance Function

- FastSLAM1.0 samples only according to the prediction distribution and does not consider the new measurements z^t
  - $\longrightarrow$ The new measurement is only incorporated through resampling
  - $\longrightarrow$ Inefficient if the noise in the vehicle motion is large
- FastSLAM2.0: Use the optimal importance function: $p(s_t^{[m]}|s_{t-1}^{[m]}, u_t, z^t, n^t)$ instead of $p(s_t^{[m]}|s_{t-1}^{[m]}, u_t, n^t)$ as in FastSLAM1.0

University of Waterloo

FastSLAM2.0: Optimal Importance Function

- FastSLAM1.0 samples only according to the prediction distribution and does not consider the new measurements z^t
  - $\longrightarrow$ The new measurement is only incorporated through resampling
  - $\longrightarrow$ Inefficient if the noise in the vehicle motion is large
- FastSLAM2.0: Use the optimal importance function: $p(s_t^{[m]}|s_{t-1}^{[m]}, u_t, z^t, n^t)$ instead of $p(s_t^{[m]}|s_{t-1}^{[m]}, u_t, n^t)$ as in FastSLAM1.0
  - $\longrightarrow$ How to sample from this importance function?
  - $\longrightarrow$ What happens to the weights?

University of Waterloo

FastSLAM2.0: Sampling New Poses

Use an EKF style approximation

- Robot motion: $\hat{s}_t^{[m]} = h(s_{t-1}^{[m]}, u_t) \rightarrow \mathcal{N}(\hat{s}_t^{[m]}, P_t)$
- Linearize the measurement model: $z_t = g(s_t, \theta_t)$:

$$\Sigma_{st}^{[m]} = [G_s^T (Q_t^{[m]})^{-1} G_s + P_t^{-1}]^{-1}$$

$$\mu_{st} = \Sigma_{st}^{[m]} G_s^T (Q_t^{[m]})^{-1} (z^t - \hat{z}^{t,[m]}) + \hat{s}_t^m$$

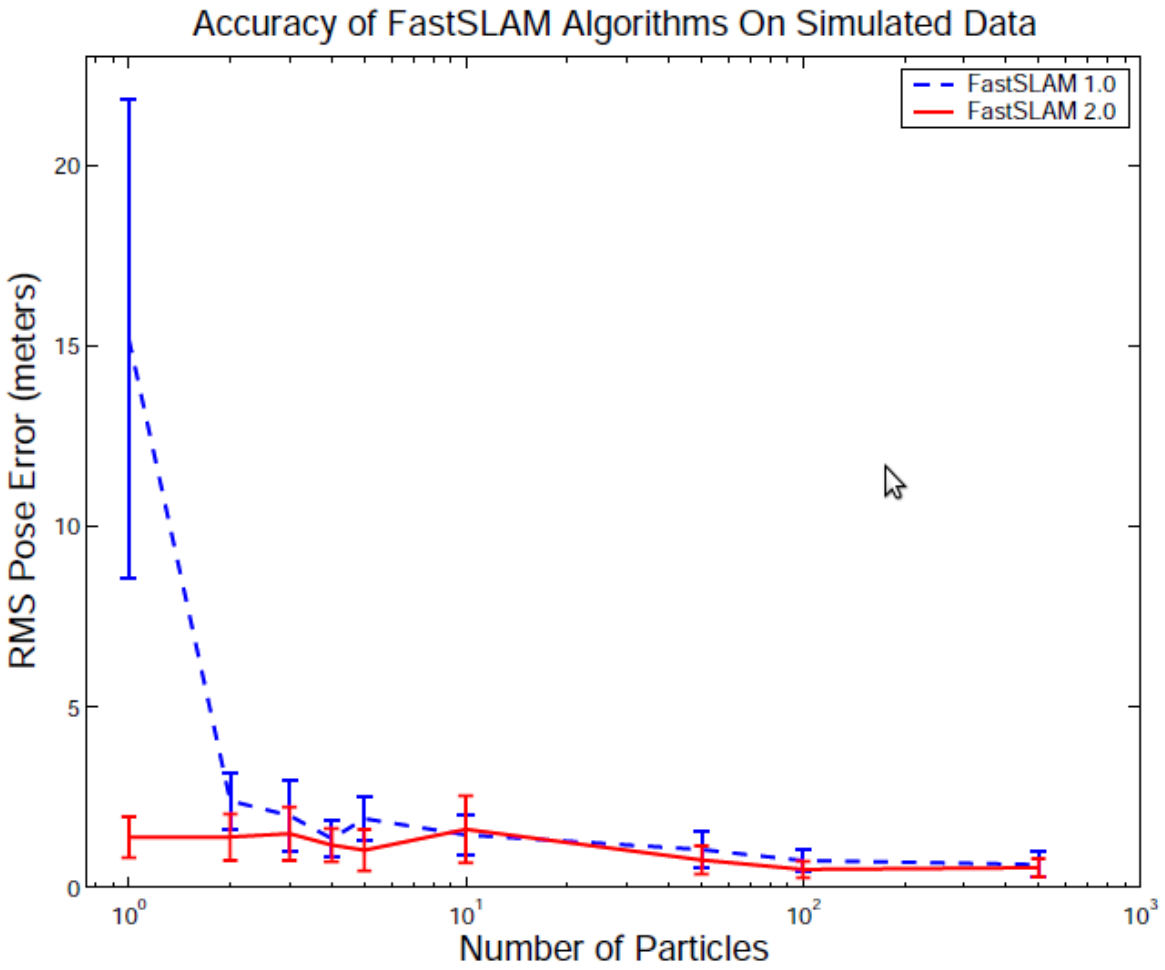
- $Q_t^{[m]} = R_t + G_\theta \Sigma_{\theta^t t-1} G_\theta^T$

FastSLAM2.0: Importance Weights

- The new weight is now equal to the likelihood of the predicted motion given the new measurement:
- $$w_t^{[m]} \propto p(z_t | s_{t-1}^{[m]}, z^{t-1}, u^t, n^t)$$
- $$\sum_k \underbrace{p(z_t | \theta_k^{[m]}, s_t^{[m]}, n_t) p(\theta_k^{[m]}) p(s_t^{[m]} | s_{t-1}^{[m]}, u_t)}_{\mathcal{N}(\hat{z}_{t,[m]}, G_s P_t G_s^T + G_\theta \Sigma_{\theta_k, t-1}^{[m]} G_\theta^T + R_t)}$$

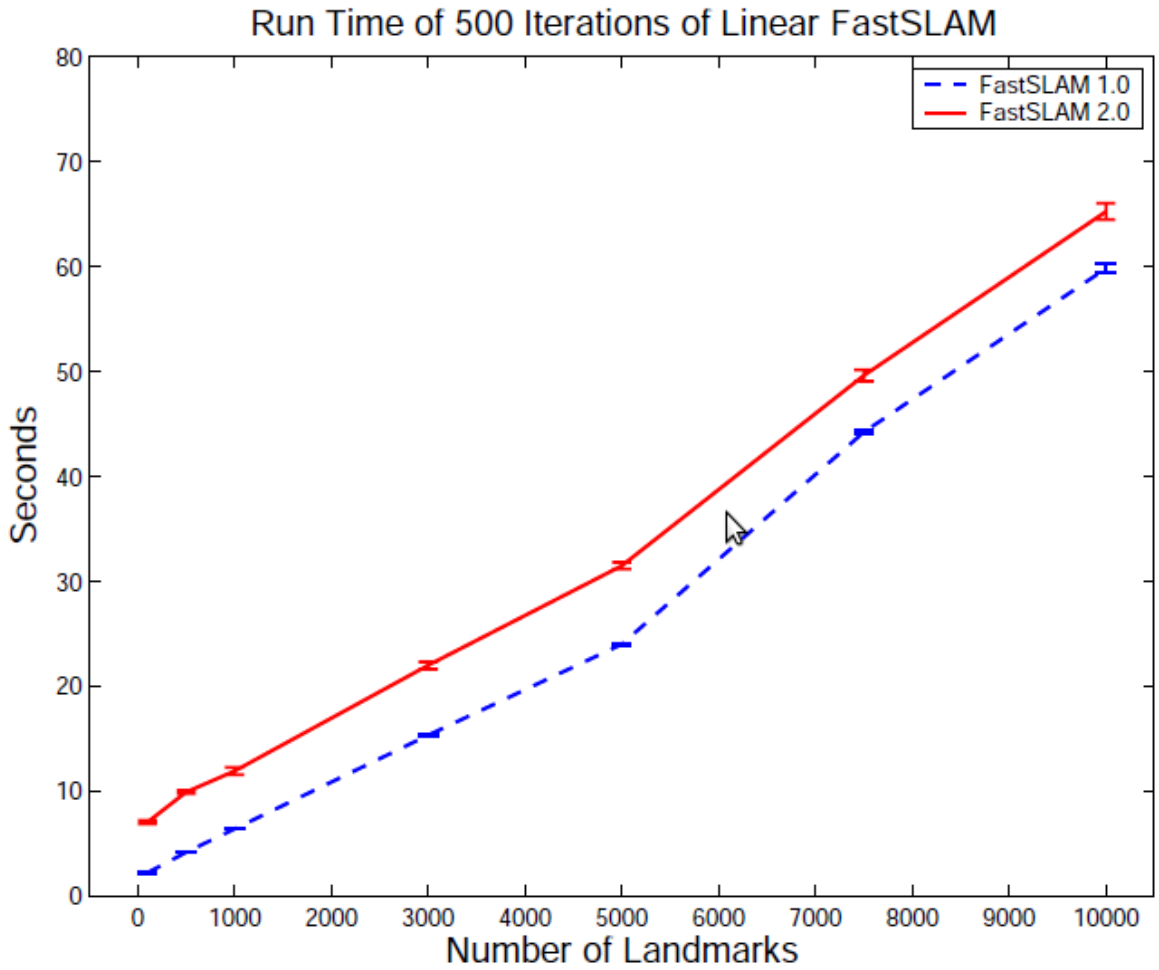
Outline

FastSLAM vs FastSLAM2.0: Accuracy



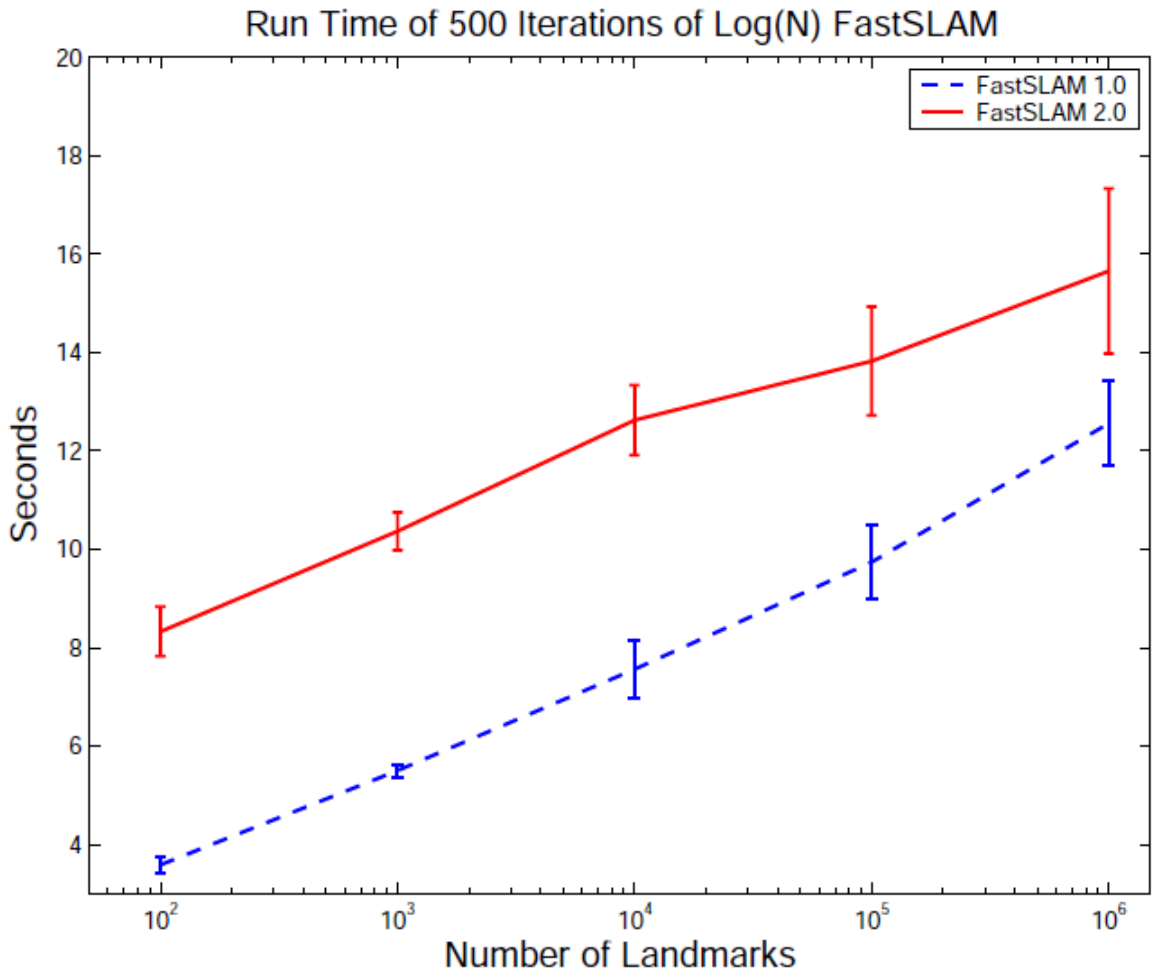
University of Waterloo

FastSLAM vs FastSLAM2.0: Linear time (No binary-tree)



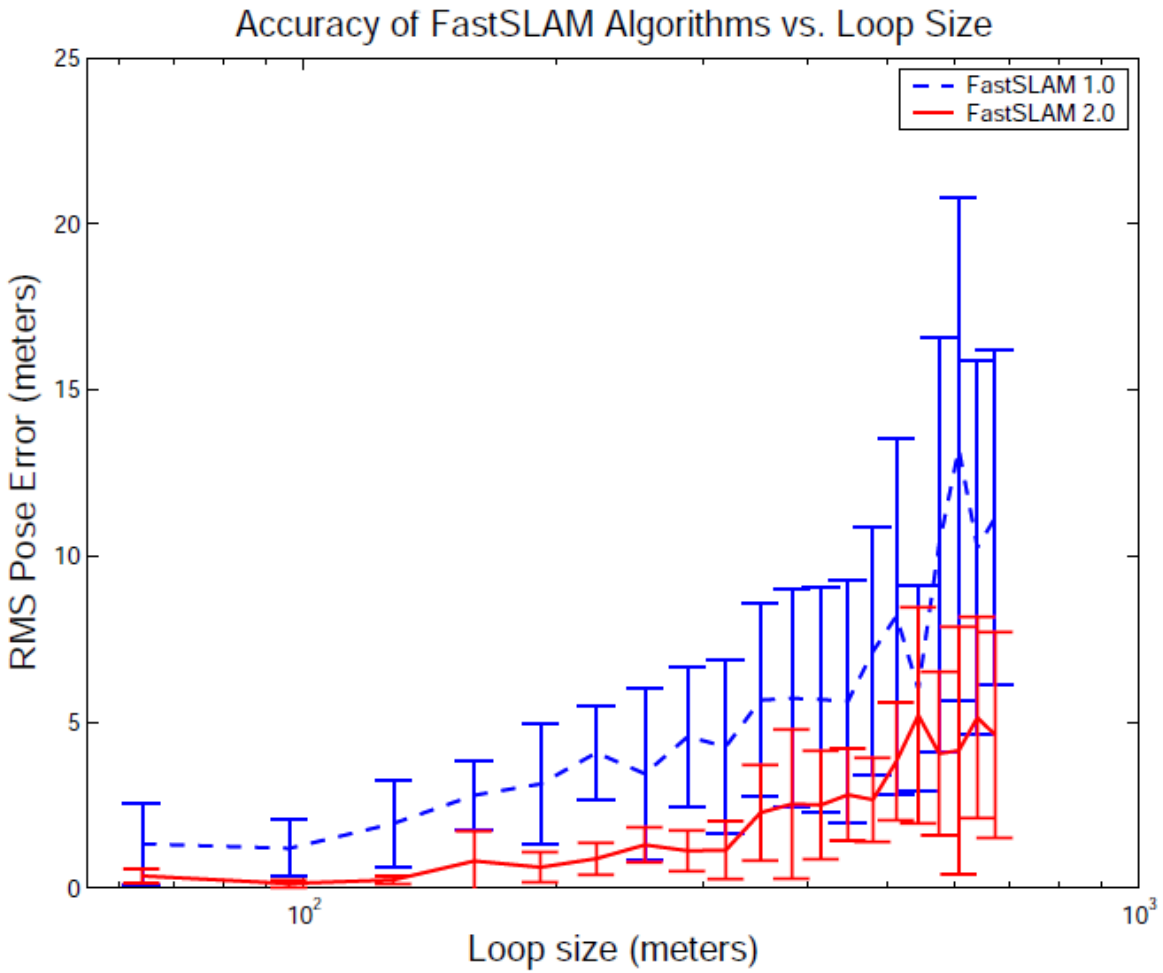
University of Waterloo

FastSLAM vs FastSLAM2.0: log(N) time (With binary-tree)



University of Waterloo

FastSLAM vs FastSLAM2.0: Loop closure



University of Waterloo

Thank you

University of Waterloo