# Robot Mapping

## FastSLAM – Feature-based SLAM with Particle Filters

**Cyrill Stachniss**



## Particle Filter in Brief

- Non-parametric, recursive Bayes filter
- Posterior is represented by a set of weighted samples
- Not limited to Gaussians
- Proposal to draw new samples
- Weight to account for the differences between the proposal and the target
- Works well in low-dimensional spaces



## Particle Filter Algorithm

1. Sample the next particle set using the proposal distribution

$$x_t^{[i]} \sim proposal(x_t \mid \dots)$$

2. Compute the importance weights

$$w_t^{[i]} = \frac{target(x_t^{[i]})}{proposal(x_t^{[i]})}$$

3. Resampling: "Replace unlikely samples by more likely ones"



## Particle Representation

- A set of weighted samples

$$\mathcal{X} = \left\{ \left\langle x^{[i]}, w^{[i]} \right\rangle \right\}_{i=1,\dots,N}$$

- Think of a sample as one hypothesis about the state
- For feature-based SLAM:

$$x = (x_{1:t}, l_{1,x}, l_{1,y}, \dots, l_{M,x}, l_{M,y})^T$$

**pose** **landmarks**

## Dimensionality Problem

Particle filters are effective in low dimensional spaces as the likely volumes of the state space need to be covered with samples.

$$x = (x_{1:t}, l_{1,x}, l_{1,y}, \ldots, l_{M,x}, l_{M,y})^T$$

**high-dimensional**



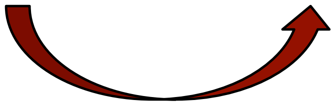
## Can We Exploit Dependencies Between the Different Dimensions of the State Space?

$$x_{1:t}, l_1, \ \ldots, l_M$$



## If We Know the Poses of the Robot, Mapping is Easy!

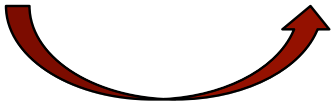
$$\underline{x_{1:t}}, \underline{l_1, \ \ldots, l_M}$$





## Key Idea

$$\underline{x_{1:t}}, \underline{l_1, \ \ldots, l_M}$$



If we use the particle set only to model the robot's path, each sample is a path hypothesis. For each sample, we can then compute an individual map of landmarks

## Rao-Blackwellization

- Factorization to exploit dependencies between variables:

$$p(a,b) = p(b \mid a)\, p(a)$$

- If $p(b \mid a)$ can be computed in closed form, represent only $p(a)$ with samples and compute $p(b \mid a)$ for every sample



## Rao-Blackwellization for SLAM

- Factorization of the SLAM posterior

poses map observations & movements

$$p(x_{0:t}, l_{1:M} \mid z_{1:t}, u_{1:t})$$

First introduced for SLAM by Murphy in 1999



## Rao-Blackwellization for SLAM

- Factorization of the SLAM posterior

poses map observations & movements

$$p(x_{0:t}, l_{1:M} \mid z_{1:t}, u_{1:t}) = p(x_{0:t} \mid z_{1:t}, u_{1:t})\; p(l_{1:M} \mid x_{0:t}, z_{1:t})$$

path posterior map posterior

First introduced for SLAM by Murphy in 1999



## Rao-Blackwellization for SLAM

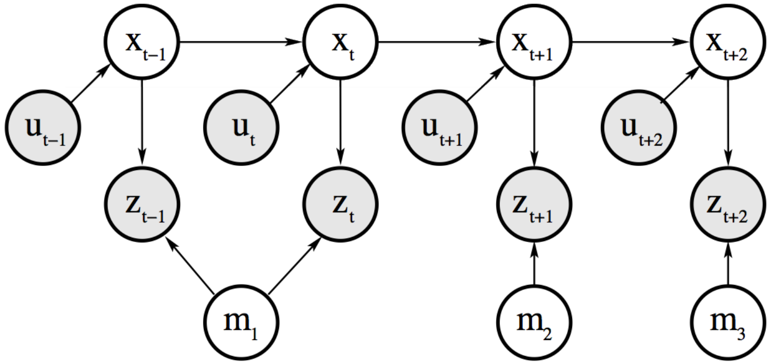
- Factorization of the SLAM posterior

$$p(x_{0:t}, l_{1:M} \mid z_{1:t}, u_{1:t}) = p(x_{0:t} \mid z_{1:t}, u_{1:t})\; p(l_{1:M} \mid x_{0:t}, z_{1:t})$$

**How to compute this term efficiently?**

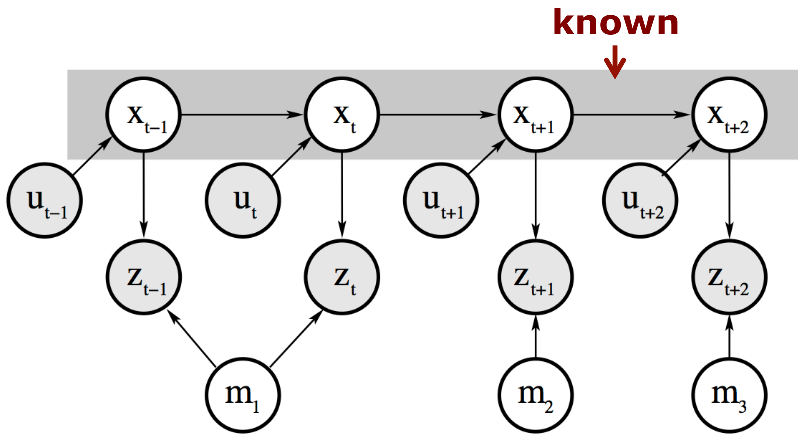
First introduced for SLAM by Murphy in 1999

## Revisit the Graphical Model



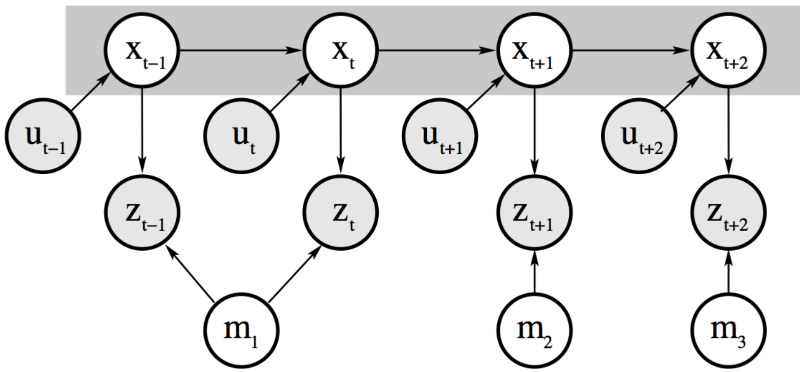



## Revisit the Graphical Model






## Landmarks are Conditionally Independent Given the Poses




**Landmark variables are all disconnected (i.e. independent) given the robot's path**



## Rao-Blackwellization for SLAM

- Factorization of the SLAM posterior

$$p(x_{0:t}, l_{1:M} \mid z_{1:t}, u_{1:t})$$
$$= p(x_{0:t} \mid z_{1:t}, u_{1:t}) \underline{p(l_{1:M} \mid x_{0:t}, z_{1:t})}$$

**Landmarks are conditionally independent given the poses**

First exploited in FastSLAM by Montemerlo et al., 2002

## Rao-Blackwellization for SLAM

- Factorization of the SLAM posterior

$$
\begin{aligned}
&p(x_{0:t}, l_{1:M} \mid z_{1:t}, u_{1:t}) \\
&\quad = p(x_{0:t} \mid z_{1:t}, u_{1:t})\, p(l_{1:M} \mid x_{0:t}, z_{1:t}) \\
&\quad = p(x_{0:t} \mid z_{1:t}, u_{1:t}) \prod_{i=1}^{M} p(l_i \mid x_{0:t}, z_{1:t})
\end{aligned}
$$

First exploited in FastSLAM by Montemerlo et al., 2002



## Rao-Blackwellization for SLAM

- Factorization of the SLAM posterior

$$
\begin{aligned}
&p(x_{0:t}, l_{1:M} \mid z_{1:t}, u_{1:t}) \\
&\quad = p(x_{0:t} \mid z_{1:t}, u_{1:t})\, p(l_{1:M} \mid x_{0:t}, z_{1:t}) \\
&\quad = p(x_{0:t} \mid z_{1:t}, u_{1:t}) \prod_{i=1}^{M} p(l_i \mid x_{0:t}, z_{1:t})
\end{aligned}
$$

**2-dimensional EKFs!**

First exploited in FastSLAM by Montemerlo et al., 2002



## Rao-Blackwellization for SLAM

- Factorization of the SLAM posterior

$$
\begin{aligned}
&p(x_{0:t}, l_{1:M} \mid z_{1:t}, u_{1:t}) \\
&\quad = p(x_{0:t} \mid z_{1:t}, u_{1:t})\, p(l_{1:M} \mid x_{0:t}, z_{1:t}) \\
&\quad = p(x_{0:t} \mid z_{1:t}, u_{1:t}) \prod_{i=1}^{M} p(l_i \mid x_{0:t}, z_{1:t})
\end{aligned}
$$

**particle filter similar to MCL**

**2-dimensional EKFs!**

First exploited in FastSLAM by Montemerlo et al., 2002



## Modeling the Robot's Path

- Sample-based representation for $p(x_{0:t} \mid z_{1:t}, u_{1:t})$
- Each sample is a path hypothesis

$x_0 \quad x_1 \quad x_2 \quad \ldots$

starting location, typically (0,0,0)

pose hypothesis at time t=1

- Past poses of a sample are not revised
- No need to maintain past poses in the sample set

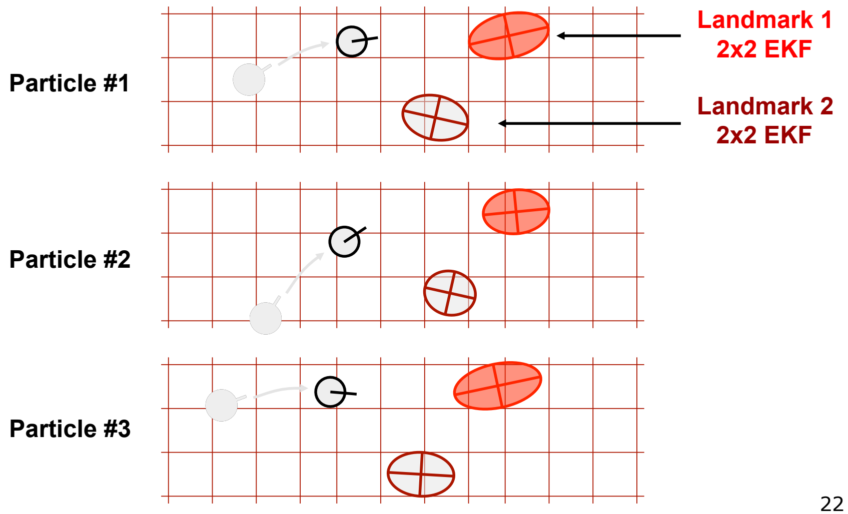
# FastSLAM

- Proposed by Montemerlo et al. in 2002
- Each landmark is represented by a 2x2 EKF
- Each particle therefore has to maintain M individual EKFs

| | | | | | |
|---|---|---|---|---|---|
| Particle 1 | $x, y, \theta$ | Landmark 1 | Landmark 2 | ... | Landmark M |
| Particle 2 | $x, y, \theta$ | Landmark 1 | Landmark 2 | ... | Landmark M |
| ⋮ | | | | | |
| Particle N | $x, y, \theta$ | Landmark 1 | Landmark 2 | ... | Landmark M |

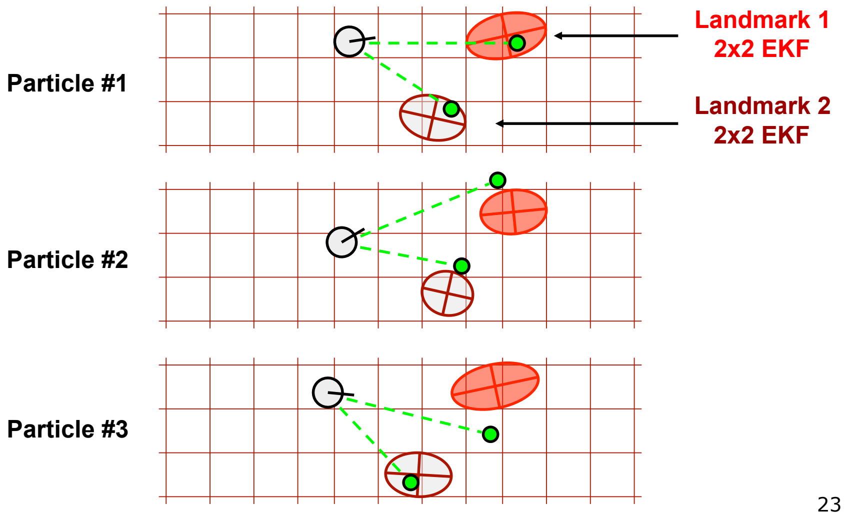


# FastSLAM – Action Update



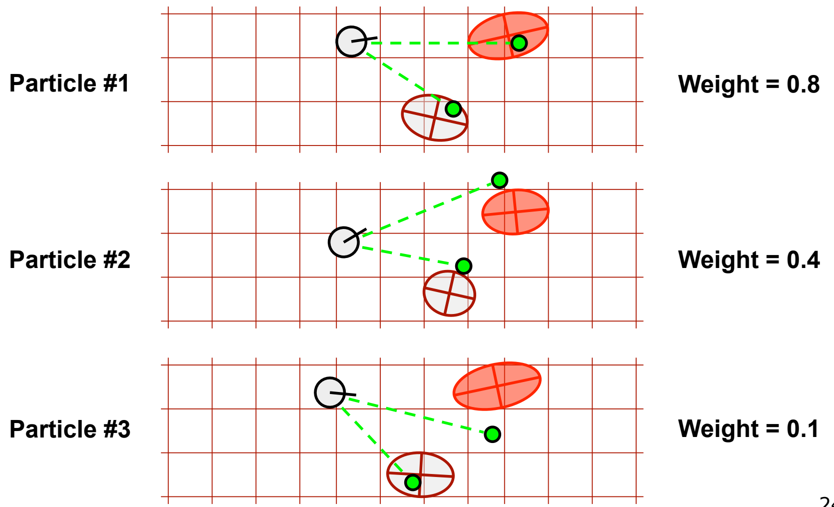



# FastSLAM – Sensor Update

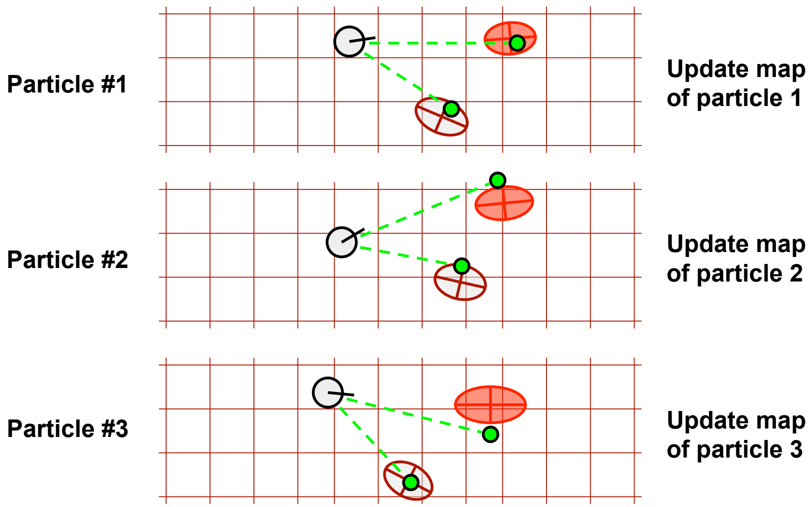





# FastSLAM – Sensor Update



# FastSLAM – Sensor Update






# Key Steps of FastSLAM 1.0

- Extend the path posterior by sampling a new pose for each sample

$$x_t^{[k]} \sim p(x_t \mid x_{t-1}^{[k]}, u_t)$$

- Compute particle weight

exp. observation

$$w^{[k]} = |2\pi Q|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(z_t - \hat{z}^{[k]})^T Q^{-1} (z_t - \hat{z}^{[k]})\right\}$$

measurement covariance

- Update belief of observed landmarks (EKF update rule)
- Resample



# FastSLAM 1.0 – Part 1

**FastSLAM1.0_known_correspondence**$(z_t, c_t, u_t, \mathcal{X}_{t-1})$:

for $k = 1$ to $N$ do // loop over all particles

Let $\left\langle x_{t-1}^{[k]}, \left\langle \mu_{1,t-1}^{[k]}, \Sigma_{1,t-1}^{[k]} \right\rangle, \ldots \right\rangle$ be particle $k$ in $\mathcal{X}_{t-1}$

$x_t^{[k]} \sim p(x_t \mid x_{t-1}^{[k]}, u_t)$ // sample pose



# FastSLAM 1.0 – Part 1

**FastSLAM1.0_known_correspondence**$(z_t, c_t, u_t, \mathcal{X}_{t-1})$:

for $k = 1$ to $N$ do // loop over all particles

Let $\left\langle x_{t-1}^{[k]}, \left\langle \mu_{1,t-1}^{[k]}, \Sigma_{1,t-1}^{[k]} \right\rangle, \ldots \right\rangle$ be particle $k$ in $\mathcal{X}_{t-1}$

$x_t^{[k]} \sim p(x_t \mid x_{t-1}^{[k]}, u_t)$ // sample pose

$j = c_t$ // observed feature

if feature $j$ never seen before

$\mu_{j,t}^{[k]} = h^{-1}(z_t, x_t^{[k]})$ // initialize mean

$H = h'(\mu_{j,t}^{[k]}, x_t^{[k]})$ // calculate Jacobian

$\Sigma_{j,t}^{[k]} = H^{-1}\, Q_t\, (H^{-1})^T$ // initialize covariance

$w^{[k]} = p_0$ // default importance weight

else

## FastSLAM 1.0 – Part 2

*else*
$\langle \mu_{j,t}^{[k]}, \Sigma_{j,t}^{[k]} \rangle = \text{EKF-Update}(\dots)$ // *update landmark*
$w^{[k]} = |2\pi Q|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(z_t - \hat{z}^{[k]})^T Q^{-1} (z_t - \hat{z}^{[k]})\right\}$

**measurement cov.** $Q = H\,\Sigma_{j,t-1}^{[k]}\,H^T + Q_t$ **exp. observation**

*endif*
*for all unobserved features* $j'$ *do*
$\langle \mu_{j',t}^{[k]}, \Sigma_{j',t}^{[k]} \rangle = \langle \mu_{j',t-1}^{[k]}, \Sigma_{j',t-1}^{[k]} \rangle$ // *leave unchanged*
*endfor*
*endfor*
$\mathcal{X}_t = \text{resample}\left(\left\langle x_t^{[k]}, \left\langle \mu_{1,t}^{[k]}, \Sigma_{1,t}^{[k]} \right\rangle, \dots, w^{[k]} \right\rangle_{k=1,\dots,N}\right)$
*return* $\mathcal{X}_t$

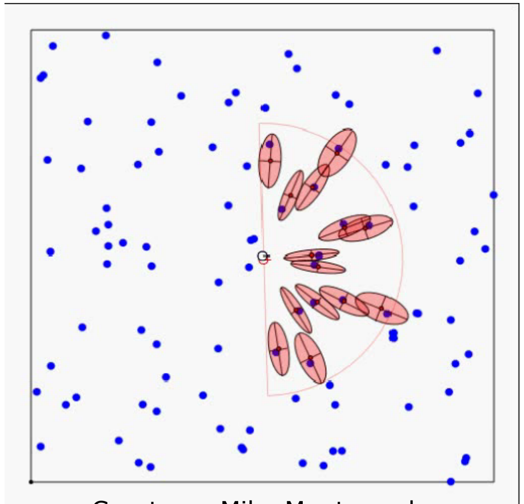


## FastSLAM 1.0 – Part 2 (long)

*else*
$\hat{z}^{[k]} = h(\mu_{j,t-1}^{[k]}, x_t^{[k]})$ // *measurement prediction*
$H = h'(\mu_{j,t-1}^{[k]}, x_t^{[k]})$ // *calculate Jacobian*
$Q = H\,\Sigma_{j,t-1}^{[k]}\,H^T + Q_t$ // *measurement covariance*
$K = \Sigma_{j,t-1}^{[k]}\,H^T\,Q^{-1}$ // *calculate Kalman gain*
$\mu_{j,t}^{[k]} = \mu_{j,t-1}^{[k]} + K(z_t - \hat{z}^{[k]})$ // *update mean*
$\Sigma_{j,t}^{[k]} = (I - K\,H)\Sigma_{j,t-1}^{[k]}$ // *update covariance*
$w^{[k]} = |2\pi Q|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(z_t - \hat{z}^{[k]})^T Q^{-1} (z_t - \hat{z}^{[k]})\right\}$ // *importance factor*
*endif*
*for all unobserved features* $j'$ *do*
$\langle \mu_{j',t}^{[k]}, \Sigma_{j',t}^{[k]} \rangle = \langle \mu_{j',t-1}^{[k]}, \Sigma_{j',t-1}^{[k]} \rangle$ // *leave unchanged*
*endfor*
*endfor*
$\mathcal{X}_t = \text{resample}\left(\left\langle x_t^{[k]}, \left\langle \mu_{1,t}^{[k]}, \Sigma_{1,t}^{[k]} \right\rangle, \dots, w^{[k]} \right\rangle_{k=1,\dots,N}\right)$
*return* $\mathcal{X}_t$



## FastSLAM  in Action



Courtesy: Mike Montemerlo



## The Importance Weight

- Derivation of the importance weight
- Based on the importance sampling principle

$$w^{[k]} = \frac{\text{target}(x^{[k]})}{\text{proposal}(x^{[k]})}$$

## The Importance Weight

- The target distribution is

$$p(x_{1:t} \mid z_{1:t}, u_{1:t})$$

- The proposal distribution is

$$p(x_{1:t} \mid z_{1:t-1}, u_{1:t})$$

- Proposal is used step-by-step

$$p(x_{1:t} \mid z_{1:t-1}, u_{1:t}) = \underbrace{p(x_t \mid x_{t-1}, u_t)}_{\text{from } \mathcal{X}_{t-1} \text{ to } \bar{\mathcal{X}}_t} \underbrace{p(x_{1:t-1} \mid z_{1:t-1}, u_{1:t-1})}_{\mathcal{X}_{t-1}}$$



## The Importance Weight

$$w^{[k]} = \frac{\text{target}(x^{[k]})}{\text{proposal}(x^{[k]})} = \frac{p(x_{1:t}^{[k]} \mid z_{1:t}, u_{1:t})}{p(x_t^{[k]} \mid x_{t-1}, u_t)\; p(x_{1:t-1}^{[k]} \mid z_{1:t-1}, u_{1:t-1})}$$



## The Importance Weight

$$w^{[k]} = \frac{\text{target}(x^{[k]})}{\text{proposal}(x^{[k]})} = \frac{p(x_{1:t}^{[k]} \mid z_{1:t}, u_{1:t})}{p(x_t^{[k]} \mid x_{t-1}, u_t)\; p(x_{1:t-1}^{[k]} \mid z_{1:t-1}, u_{1:t-1})}$$

**Bayes rule + factorization**



## The Importance Weight

$$\begin{aligned} w^{[k]} &= \frac{\text{target}(x^{[k]})}{\text{proposal}(x^{[k]})} \\ &= \frac{p(x_{1:t}^{[k]} \mid z_{1:t}, u_{1:t})}{p(x_t^{[k]} \mid x_{t-1}, u_t)\; p(x_{1:t-1}^{[k]} \mid z_{1:t-1}, u_{1:t-1})} \\ &= \frac{\eta\, p(z_t \mid x_{1:t}^{[k]}, z_{1:t-1})\; p(x_t \mid x_{t-1}^{[k]}, u_t)}{p(x_t^{[k]} \mid x_{t-1}^{[k]}, u_t)} \\ &\quad \frac{p(x_{1:t-1}^{[k]} \mid z_{1:t-1}, u_{1:t-1})}{p(x_{1:t-1}^{[k]} \mid z_{1:t-1}, u_{1:t-1})} \end{aligned}$$

## The Importance Weight

$$w^{[k]} = \frac{\text{target}(x^{[k]})}{\text{proposal}(x^{[k]})}$$

$$= \frac{p(x_{1:t}^{[k]} \mid z_{1:t}, u_{1:t})}{p(x_t^{[k]} \mid x_{t-1}, u_t)\; p(x_{1:t-1}^{[k]} \mid z_{1:t-1}, u_{1:t-1})}$$

$$= \frac{\eta\, p(z_t \mid x_{1:t}^{[k]}, z_{1:t-1})\; \cancel{p(x_t \mid x_{t-1}^{[k]}, u_t)}}{\cancel{p(x_t^{[k]} \mid x_{t-1}^{[k]}, u_t)}} \frac{\cancel{p(x_{1:t-1}^{[k]} \mid z_{1:t-1}, u_{1:t-1})}}{\cancel{p(x_{1:t-1}^{[k]} \mid z_{1:t-1}, u_{1:t-1})}}$$



## The Importance Weight

$$w^{[k]} = \frac{\text{target}(x^{[k]})}{\text{proposal}(x^{[k]})}$$

$$= \frac{p(x_{1:t}^{[k]} \mid z_{1:t}, u_{1:t})}{p(x_t^{[k]} \mid x_{t-1}, u_t)\; p(x_{1:t-1}^{[k]} \mid z_{1:t-1}, u_{1:t-1})}$$

$$= \frac{\eta\, p(z_t \mid x_{1:t}^{[k]}, z_{1:t-1})\; \cancel{p(x_t \mid x_{t-1}^{[k]}, u_t)}}{\cancel{p(x_t^{[k]} \mid x_{t-1}^{[k]}, u_t)}} \frac{\cancel{p(x_{1:t-1}^{[k]} \mid z_{1:t-1}, u_{1:t-1})}}{\cancel{p(x_{1:t-1}^{[k]} \mid z_{1:t-1}, u_{1:t-1})}}$$

$$= \eta\, p(z_t \mid x_{1:t}^{[k]}, z_{1:t-1})$$



## The Importance Weight

- Integrating over the pose of the observed landmark leads to

$$w^{[k]}$$
$$= \eta\, p(z_t \mid x_{1:t}^{[k]}, z_{1:t-1})$$
$$= \eta \int_{l_j} p(z_t \mid x_{1:t}^{[k]}, z_{1:t-1}, l_j)\; p(l_j \mid x_{1:t}^{[k]}, z_{1:t-1})\, dl_j$$



## The Importance Weight

- Integrating over the pose of the observed landmark leads to

$$w^{[k]}$$
$$= \eta\, p(z_t \mid x_{1:t}^{[k]}, z_{1:t-1})$$
$$= \eta \int_{l_j} p(z_t \mid x_{1:t}^{[k]}, z_{1:t-1}, l_j)\; p(l_j \mid x_{1:t}^{[k]}, z_{1:t-1})\, dl_j$$
$$= \eta \int_{l_j} p(z_t \mid x_t^{[k]}, l_j)\;\; p(l_j \mid x_{1:t-1}^{[k]}, z_{1:t-1})\, dl_j$$

# The Importance Weight

- Integrating over the pose of the observed landmark leads to

$$
\begin{aligned}
w^{[k]} &\\
&= \eta\, p(z_t \mid x_{1:t}^{[k]}, z_{1:t-1}) \\
&= \eta \int_{l_j} p(z_t \mid x_{1:t}^{[k]}, z_{1:t-1}, l_j)\, p(l_j \mid x_{1:t}^{[k]}, z_{1:t-1})\, dl_j \\
&= \eta \int_{l_j} \underbrace{p(z_t \mid x_t^{[k]}, l_j)}_{\mathcal{N}(z_t; \hat{z}^{[k]}, Q_t)}\ \underbrace{p(l_j \mid x_{1:t-1}^{[k]}, z_{1:t-1})}_{\mathcal{N}(l_j; \mu_{j,t-1}^{[k]}, \Sigma_{j,t-1}^{[k]})}\, dl_j
\end{aligned}
$$



# The Importance Weight

- This leads to

$$w^{[k]} = \eta \int_{l_j} \underbrace{p(l_j \mid x_{1:t-1}^{[k]}, z_{1:t-1})}_{\mathcal{N}(l_j; \mu_{j,t-1}^{[k]}, \Sigma_{j,t-1}^{[k]})}\ \underbrace{p(z_t \mid x_t^{[k]}, l_j)}_{\mathcal{N}(z_t; \hat{z}^{[k]}, Q_t)}\, dl_j$$

$$Q = H\, \Sigma_{j,t-1}^{[k]}\, H^T + Q_t$$

**measurement covariance**
**(pose uncertainty of l and measurement noise)**



# The Importance Weight

- This leads to

$$w^{[k]} = \eta \int_{l_j} \underbrace{p(l_j \mid x_{1:t-1}^{[k]}, z_{1:t-1})}_{\mathcal{N}(l_j; \mu_{j,t-1}^{[k]}, \Sigma_{j,t-1}^{[k]})}\ \underbrace{p(z_t \mid x_t^{[k]}, l_j)}_{\mathcal{N}(z_t; \hat{z}^{[k]}, Q_t)}\, dl_j$$

$$Q = H\, \Sigma_{j,t-1}^{[k]}\, H^T + Q_t$$

$$w^{[k]} \simeq |2\pi Q|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(z_t - \hat{z}^{[k]})^T Q^{-1} (z_t - \hat{z}^{[k]})\right\}$$

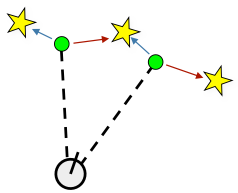


# FastSLAM 1.0 – Part 2

*else*
$\langle \mu_{j,t}^{[k]}, \Sigma_{j,t}^{[k]} \rangle = \text{EKF-Update}(\dots)$ *// update landmark*
$w^{[k]} = |2\pi Q|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(z_t - \hat{z}^{[k]})^T Q^{-1} (z_t - \hat{z}^{[k]})\right\}$
*endif*
*for all unobserved features $j'$ do*
$\langle \mu_{j',t}^{[k]}, \Sigma_{j',t}^{[k]} \rangle = \langle \mu_{j',t-1}^{[k]}, \Sigma_{j',t-1}^{[k]} \rangle$ *// leave unchanged*
*endfor*
*endfor*
$\mathcal{X}_t = \text{resample}\left(\left\langle x_t^{[k]}, \left\langle \mu_{1,t}^{[k]}, \Sigma_{1,t}^{[k]} \right\rangle, \dots, w^{[k]} \right\rangle_{k=1,\dots,N}\right)$
*return* $\mathcal{X}_t$

## Data Association Problem

- Which observation belongs to which landmark?

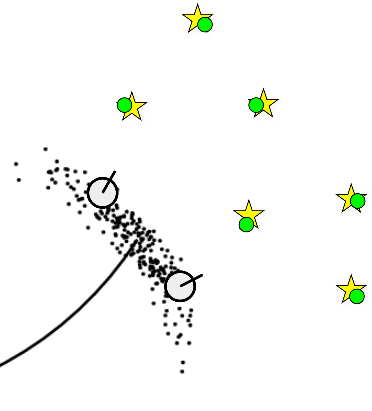


- More than one possible association
- **Potential data associations depend on the pose of the robot**



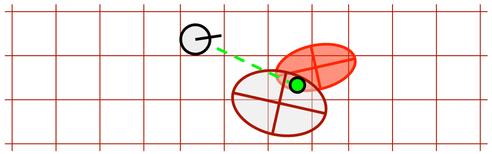
## Particles Support for Multi-Hypotheses Data Association

- Decisions on a per-particle basis
- Robot pose **error** is factored out of data association decisions





## Per-Particle Data Association

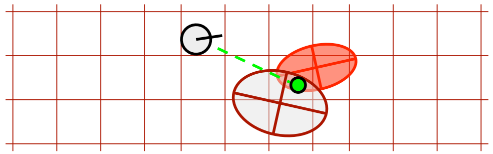


Was the observation generated by the **red** or by the **brown** landmark?

P(observation|red) = 0.3     P(observation|brown) = 0.7



## Per-Particle Data Association

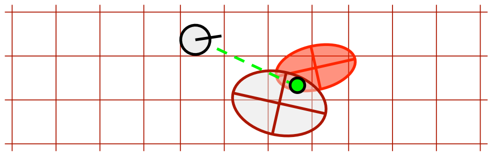


Was the observation generated by the **red** or by the **brown** landmark?

P(observation|red) = 0.3     P(observation|brown) = 0.7

- Two options for per-particle data association
  - Pick the most probable match
  - Pick an random association weighted by the observation likelihoods
- If the probability for an assignment is too low, generate a new landmark

## Per-Particle Data Association



Was the observation generated by the **red** or by the **brown** landmark?

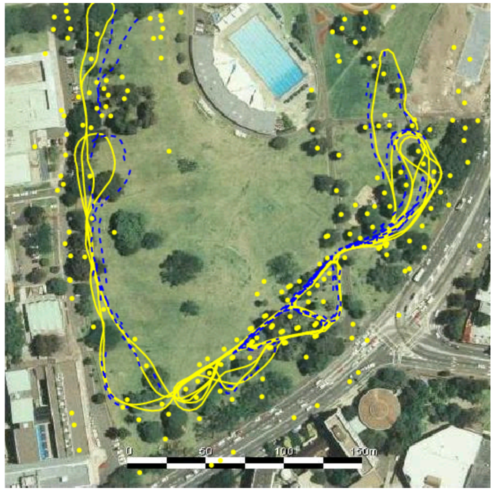
- Multi-modal belief
- Pose error is factored out of data association decisions
- **Simple but effective** data association
- Big **advantage of FastSLAM** over EKF



## Results – Victoria Park

- 4 km traverse
- < 2.5 m RMS position error
- 100 particles

**Blue** = GPS
**Yellow** = FastSLAM



Courtesy: Mike Montemerlo

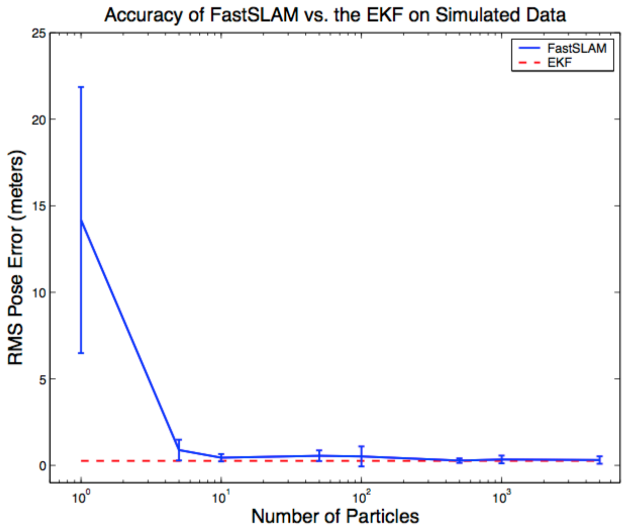


## Results – Victoria Park (Video)

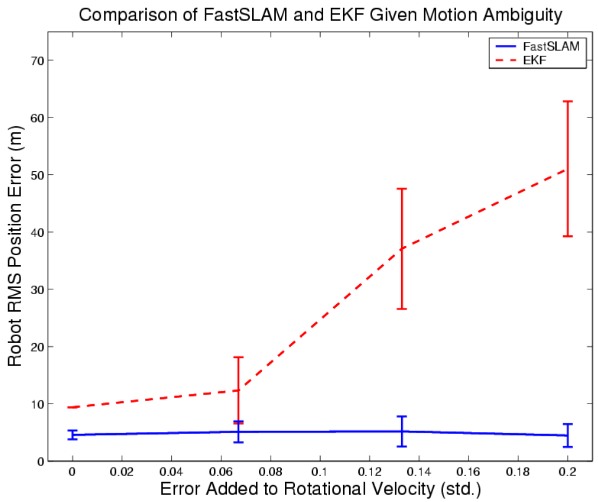
Courtesy: Mike Montemerlo



## Results (Sample Size)



## Results (Motion Uncertainty)






## FastSLAM 1.0 Summary

- Use a particle filter to model the belief
- Factors the SLAM posterior into low-dimensional estimation problems
- Model only the robot's path by sampling
- Compute the landmarks given the path
- Per-particle data association
- No robot pose uncertainty in the per-particle data association



## FastSLAM Complexity – Simple Implementation

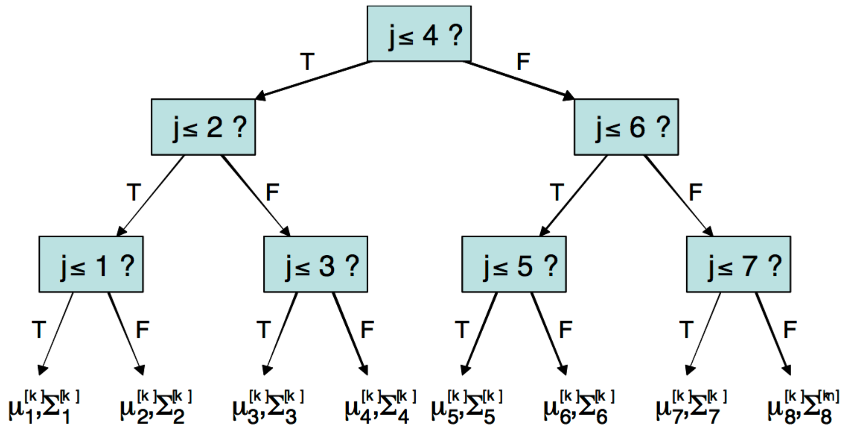
- Update robot particles based on the control $\mathcal{O}(N)$
- Incorporate an observation into the Kalman filters $\mathcal{O}(N)$
- Resample particle set $\mathcal{O}(NM)$

$$\mathcal{O}(NM)$$

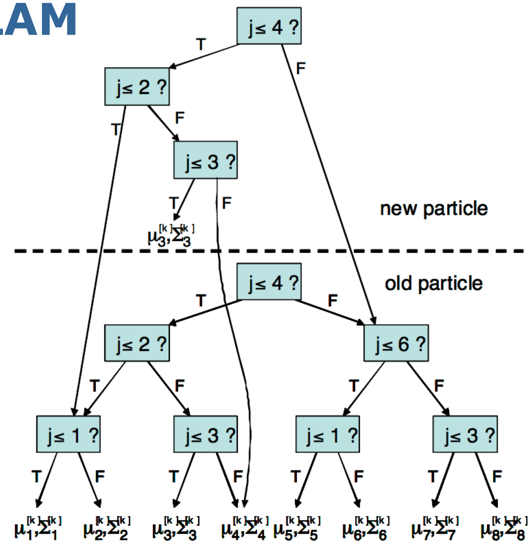
**N = Number of particles**
**M = Number of map features**



## A Better Data Structure for FastSLAM



# A Better Data Structure for FastSLAM






# FastSLAM Complexity

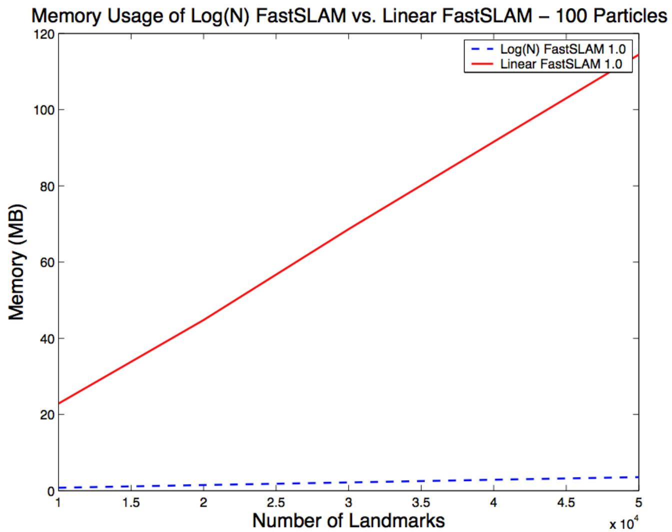
- Update robot particles based on the control $\mathcal{O}(N)$
- Incorporate an observation into the Kalman filters $\mathcal{O}(N \log M)$
- Resample particle set $\mathcal{O}(N \log M)$

**N = Number of particles**
**M = Number of map features**

$$\mathcal{O}(N \log M)$$



# Memory Complexity






# FastSLAM 1.0

- FastSLAM 1.0 uses the motion model as the proposal distribution

$$x_t^{[k]} \sim p(x_t \mid x_{t-1}^{[k]}, u_t)$$

- **Is there a better distribution to sample from?**

[Montemerlo et al., 2002]

## FastSLAM 1.0 to FastSLAM 2.0

- FastSLAM 1.0 uses the motion model as the proposal distribution

$$x_t^{[k]} \sim p(x_t \mid x_{t-1}^{[k]}, u_t)$$

- FastSLAM 2.0 **considers also the measurements during sampling**
- Especially useful if an accurate sensor is used (compared to the motion noise)

[Montemerlo et al., 2003]



## FastSLAM 2.0 (Informally)

- FastSLAM 2.0 samples from

$$x_t^{[k]} \sim p(x_t \mid x_{1:t-1}^{[k]}, u_{1:t}, z_{1:t})$$

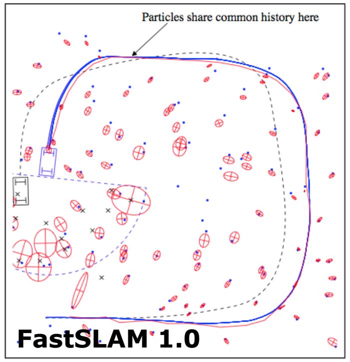
- Results in a more peaked proposal distribution
- Less particles are required
- More robust and accurate
- But more complex…

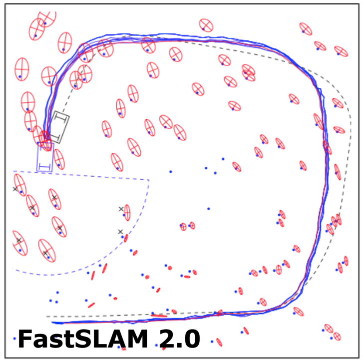
[Montemerlo et al., 2003]



## FastSLAM Problems

- How to determine the sample size?
- Particle deprivation, especially when closing (multiple) loops






## FastSLAM Summary

- Particle filter-based SLAM
- Rao-Blackwellization: model the robot’s path by sampling and compute the landmarks given the poses
- Allow for per-particle data association
- FastSLAM 1.0 and 2.0 differ in the proposal distribution
- Complexity $\mathcal{O}(N \log M)$

## FastSLAM Results

- Scales well (1 million+ features)
- Robust to ambiguities in the data association
- Advantages compared to the classical EKF approach (especially with non-linearities)



## Literature

**FastSLAM**


- Thrun et al.: "Probabilistic Robotics", Chapter 13.1-13.3 + 13.8 (see errata!)
- Montemerlo, Thrun, Kollar, Wegbreit: FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem, 2002
- Montemerlo and Thrun: Simultaneous Localization and Mapping with Unknown Data Association Using FastSLAM, 2003