Robot Mapping

FastSLAM – Feature-Based SLAM with Particle Filters

Cyrill Stachniss

PARTICLE FILTER STATE ESTIMATION IN THE CONTEXT OF NON PARAMETRIC DISTRIBUTIONS
→ SET of SAMPLES, sample = 1 possible state
→ KEY: very flexible in Terms of distribution to represent

?

How to USE PARTICLE FILTERS to solve The SLAM problem.

RIPASSO

Particle Filter

non abbiamo una descrizione parametrica, ma usiamo dei SAMPLE

- Non-parametric recursive Bayes filter
- Posterior is represented by a set of weighted samples → How MANY?
- Can model arbitrary distributions
- Works well in low-dimensional spaces
- 3-Step procedure
  - Sampling from proposal $x_t \rightarrow \bar{x}_t$ (MOTION MODEL)
  - Importance Weighting $\text{weights} = \frac{\text{target}(x_t^j)}{\text{proposal}(x_t^j)}$
  - Resampling CREA NUOVI SAMPLE (CON RIMPIAZZO) $\propto$ al peso dato in precedenza.

Particle Filter Algorithm

1. Sample the particles from the proposal distribution

$$x_t^{[j]} \sim \pi(x_t \mid \ldots)$$

2. Compute the importance weights

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

3. Resampling: Draw sample i with probability $w_t^{[i]}$ and repeat J times

Particle Representation

- A set of weighted samples

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

→ SAMPLE SET × LOCALIZZAZIONE
(x; y; θ) 3 dimension

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

$$x = (\underbrace{x_{1:t}}_{\text{poses}}, \underbrace{m_{1,x}, m_{1,y}, \ldots, m_{M,x}, m_{M,y}}_{\text{ALL landmarks LOCATIONS}})^T$$

STATE SPACE ?

3 DIM. IN LOCALIZATION TO ....

MUCH HIGHER DIMEN. SPACE.

PARTICLE FILTER E' OK QUANDO LO SPAZIO DEGLI STATI E' PICCOLO (3...4...10.. STOP!) MA SE ABBIAMO × ES 1 MILIONE DI LANDMARKS + LUNGA TRAIETTORIA = TROPPO GRANDE LA DIMENSIONE DEGLI STATI

Dimensionality Problem

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

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

high-dimensional

NO WAY TO HANDLE SUCH A PROBLEM WITH PARTICLE FILTER LIKE THIS

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

INDIVIDUAL STATE VARIABLES →

POSITION OF ROBOT

LANDMARK LOCATION

$$\underbrace{x_{1:t}}_{\text{POSITION OF ROBOT}}, \underbrace{m_1, \ldots, m_M}_{\text{LANDMARK LOCATION}}$$

POSSIAMO SFRUTTARE LA DIPENDENZA TRA LE SINGOLE VARIABILI ?

C'È DIPENDENZA TRA LE POSE ed OGNI SINGOLO LANDMARK?
- Y/N
- QUALI SONO LE DIPENDENZE CHE POSSIAMO SFRUTTARE?

IDEA

PARTICLE FILTER TO REPRESENT THE POSES OF THE ROBOT ← SE CONOSCIAMO LE POSIZIONI DEL ROBOT MAPPING IS EASY ← QUALI SONO LE DIPENDENZE CHE POSSIAMO SFRUTTARE?

PARTICLE

1 → PARTICELLA 1 ha la giusta traiettoria, quindi il MONDO sarà W_1
2 → " 2 " " " " " " " " W_2
n ⋮

(position of $x_{1:t}$ facendo mapping with KNOWN poses)
distribution of The POSSIBLE MAPS condizionate dai SAMPLE ovvero le PARTICLE

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

$$\underline{x_{1:t}}, \underline{m_1, \ldots, m_M}$$

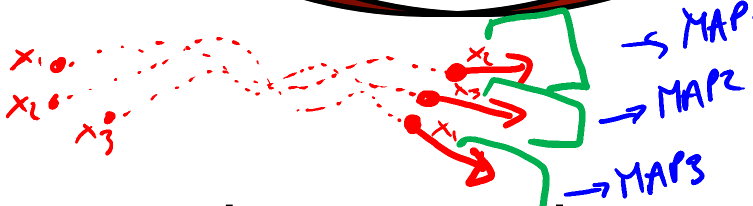
COSA C'E' IN PIÙ IN OGNI PARTICLE RISPETTO ALLA LOCALIZZAZIONE?

IN LOCALIZZAZIONE COSA C'ERA DENTRO UNA PARTICLE? $\{x, y, \theta\}$ → POSITION + ORIENTATION

Key Idea

+ LANDMARKS POSITIONS → The MAP

$$\underline{x_{1:t}}, \underline{m_1, \ldots, m_M}$$



mappa associata alla particella i-esima applicando le misure sensoriali

QUESTO SI CHIAMA RAO BLACKWELLIZATION

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

Rao-Blackwellization

SFRUTTA:

- Factorization to exploit dependencies between variables:

SE DUE ELEMENTI SONO CONDIZIONATI UNO DALL'ALTRO, ALLORA POSSIAMO DIRE CHE LA

PROBABILITÀ CONGIUNTA

• SE POSSIAMO FARLO VELOCEMENTE E FACILE, OVVERO SE "B" DATO "A" E' FACILE ALLORA

$$p(a, b) \quad = \quad \overbrace{p(b \mid a)}\, p(a)$$

$$p(\text{POSE}, \text{MAP}) = p(\text{MAP} \mid \text{POSE})\, p(\text{POSE})$$

RAPPRESENTIAMO P(A) CON PARTICELLE (ovvero le pose del Robot) E B è la mappa

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

Invece che computare p(pose AND map), prendiamo dei SAMPLE per rappresentare le pose e poi per ogni sample computiamo $p(b \mid a)$ [p(b) DATO a] ovvero la likelihood della MAPPA data la posizione. Le particelle dovranno coprire solamente lo spazio degli stati di POSE e non di POSE + MAP. Questo è OK perché POSE → LOW DIMENSIONAL w.r.t. MAP

Rao-Blackwellization for SLAM

- Factorization of the SLAM posterior

poses map observations & movements

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

First introduced for SLAM by Murphy in 1999

Rao-Blackwellization for SLAM

- Factorization of the SLAM posterior

poses map observations & movements

$$p(\underbrace{x_{0:t}}_{A}, \underbrace{m_{1:M}}_{B} \mid z_{1:t}, u_{1:t}) =$$

DIVIDIAMOLO UTILIZZANDO $P(A,B) = P(B|A) \cdot P(A)$

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

QUI DOVREMMO AVERE ANCHE $u_{1:t}$ ma se abbiamo le posizioni allora non ci interessano i controlli

path posterior map posterior

$$P(A, B) = P(A) \cdot P(B \mid A)$$

2 POSTERIOR = PATH stima quale "traiettoria" il robot ha preso, la nuova posizione
MAP, che DIPENDE da quale "traiettoria" ha preso il robot, ovvero la nuova pose

First introduced for SLAM by Murphy in 1999

Rao-Blackwellization for SLAM

- Factorization of the SLAM posterior

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

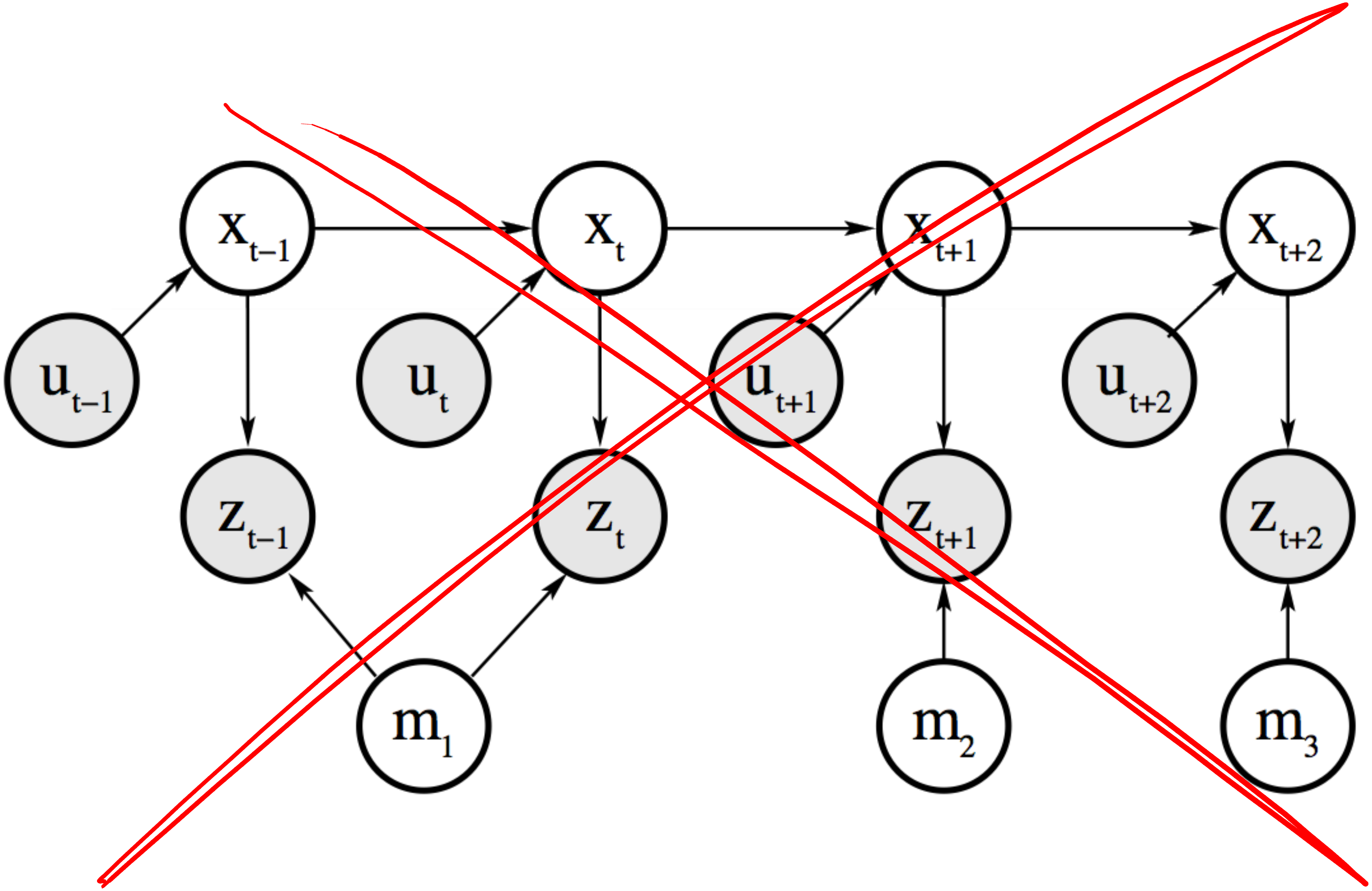
How to compute this term efficiently?

QUESTO E' CHIAMATO MAPPING WITH KNOWN POSES

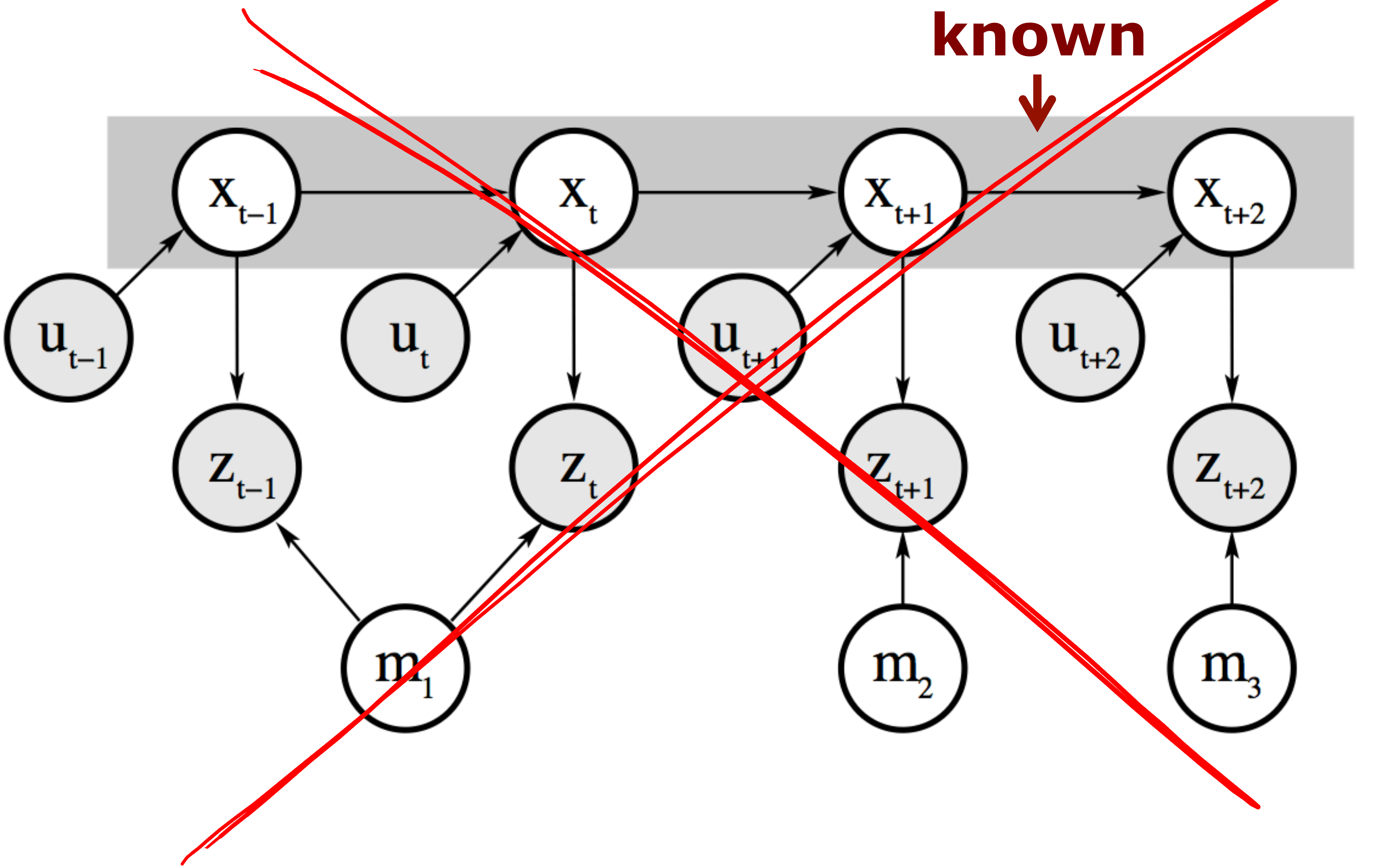
X VEDERE COME QUESTO SI PUÒ FARE FACILMENTE, GUARDIAMO IL GRAPHICAL MODEL ASSOCIATO AL PROBLEMA

First introduced for SLAM by Murphy in 1999

Revisit the Graphical Model



Revisit the Graphical Model



DALLA SLIDE PRECEDENTE .. $p(x_{0:t}, m \mid z_{1:t}, u_{1:t}) = p(\ldots) \cdot p(\ldots)$

$p(m_{1:t} \mid x_{0:t}, z_{1:t})$

e abbiamo detto che le conosciamo

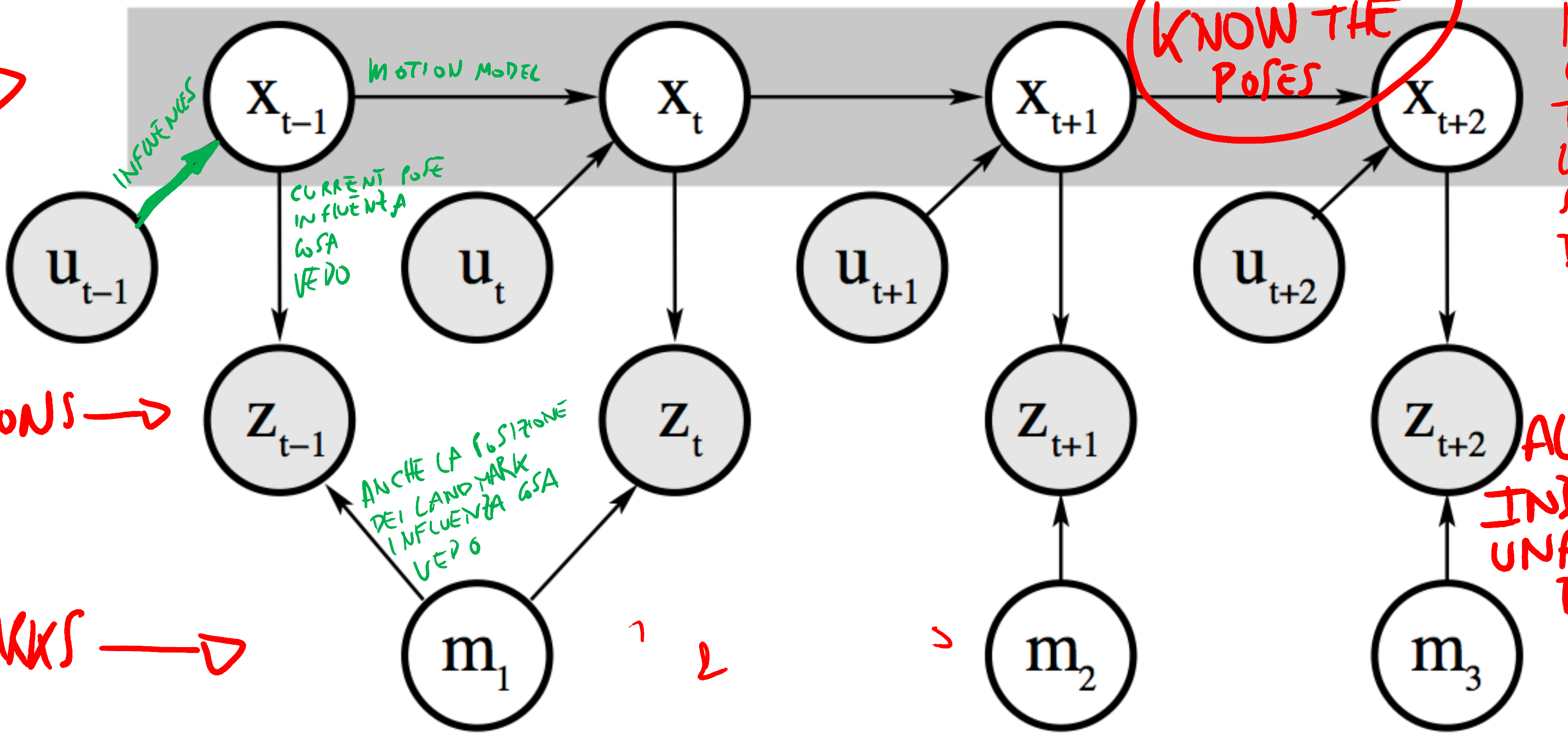
Landmarks are Conditionally Independent Given the Poses

POSES →

CONTROLS →

OBSERVATIONS →

LANDMARKS →



SE NON C'E' UN CAMMINO TRA DUE LANDMARK SENZA PASSARE DA x_t ALLORA SONO INDIPENDENTI UNA DALL'ALTRA DATO x_i

the poses

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

Rao-Blackwellization for SLAM

- Factorization of the SLAM posterior

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

QUINDI SE SAPPIAMO CHE TUTTI I LANDMARK SONO INDIPENDENTI UNO DALL'ALTRO....

Landmarks are conditionally independent given the poses

$$P(M_1 \wedge M_2 \wedge M_3 \dots M_M) = P(M_1) \cdot P(M_2) \cdot \dots \cdot P(M_M)$$

QUESTO PERCHÉ SONO INDIPENDENTI!

First exploited in FastSLAM by Montemerlo et al., 2002

Rao-Blackwellization for SLAM

- Factorization of the SLAM posterior

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

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

$$p(x_{0:t} \mid z_{1:t}, u_{1:t}) \prod_{i=1}^{M} p(m_i \mid x_{0:t}, z_{1:t})$$

E FACENDO IN QUESTO MODO C'È UNA GRANDE SEMPLIFICAZIONE:

per capire quale sia, dobbiamo pensare a cosa fa EKF-SLAM. Come sono i landmark in EKF SLAM, come sono rappresentati? IN EKF abbiamo una UNICA matrice per tutti i landmark e la complessità di EKF deriva proprio dall'avere matrici molto grandi. Ma qui abbiamo appena detto che ogni landmark può essere stimato INDIPENDENTEMENTE da tutto il resto DATE le poses

First exploited in FastSLAM by Montemerlo et al., 2002

Rao-Blackwellization for SLAM

- Factorization of the SLAM posterior

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

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

SIMILE A MCL (MONTE CARLO LOCALIZATION)

$$\boxed{p(x_{0:t} \mid z_{1:t}, u_{1:t})} \prod_{i=1}^{M} \underline{p(m_i \mid x_{0:t}, z_{1:t})}$$

QUINDI POSSIAMO PENSARE DI AVERE, PER OGNI POSIZIONE, UNA LISTA DI PICCOLI EKF, UNO PER OGNI LANDMARK, TENENDO TRACCIA DELLA POSIZIONE DI OGNI LANDMARK E QUINDI POSIZIONE = $\{x, y\}$ $\Rightarrow$

2-dimensional EKFs!

First exploited in FastSLAM by Montemerlo et al., 2002

Rao-Blackwellization for SLAM

- Factorization of the SLAM posterior

$$
\begin{aligned}
& p(x_{0:t}, m_{1:M} \mid z_{1:t}, u_{1:t}) = \\
& \qquad p(x_{0:t} \mid z_{1:t}, u_{1:t}) \; p(m_{1:M} \mid x_{0:t}, z_{1:t}) \\
& \qquad \underline{p(x_{0:t} \mid z_{1:t}, u_{1:t})} \; \prod_{i=1}^{M} \underline{p(m_i \mid x_{0:t}, z_{1:t})}
\end{aligned}
$$

particle filter similar to MCL

2-dimensional EKFs!

First exploited in FastSLAM by Montemerlo et al., 2002

VEDIAMO COME FUNZIONA IN DETTAGLIO:
ABBIAMO DETTO CHE USIAMO UN PARTICLE FILTER PER TRACCIARE LE ~~POSIZIONI~~ POSE

Modeling the Robot's Path

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

x_0 — starting location, typically (0,0,0)

x_1 — pose hypothesis at time t=1

x_2 — T=2

$\dots$

- Past poses of a sample are not revised

LE POSE AI TEMPI (T-x) "RECENTI" NON LE TOCCHIAMO, NON LE RI-AGGIORNIAMO
QUINDI

- No need to maintain past poses in the sample set

SEGUE CHE LE PARTICLE TRACCIANO "LE POSIZIONI" CORRENTI $\Rightarrow \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}$ + LANDM.

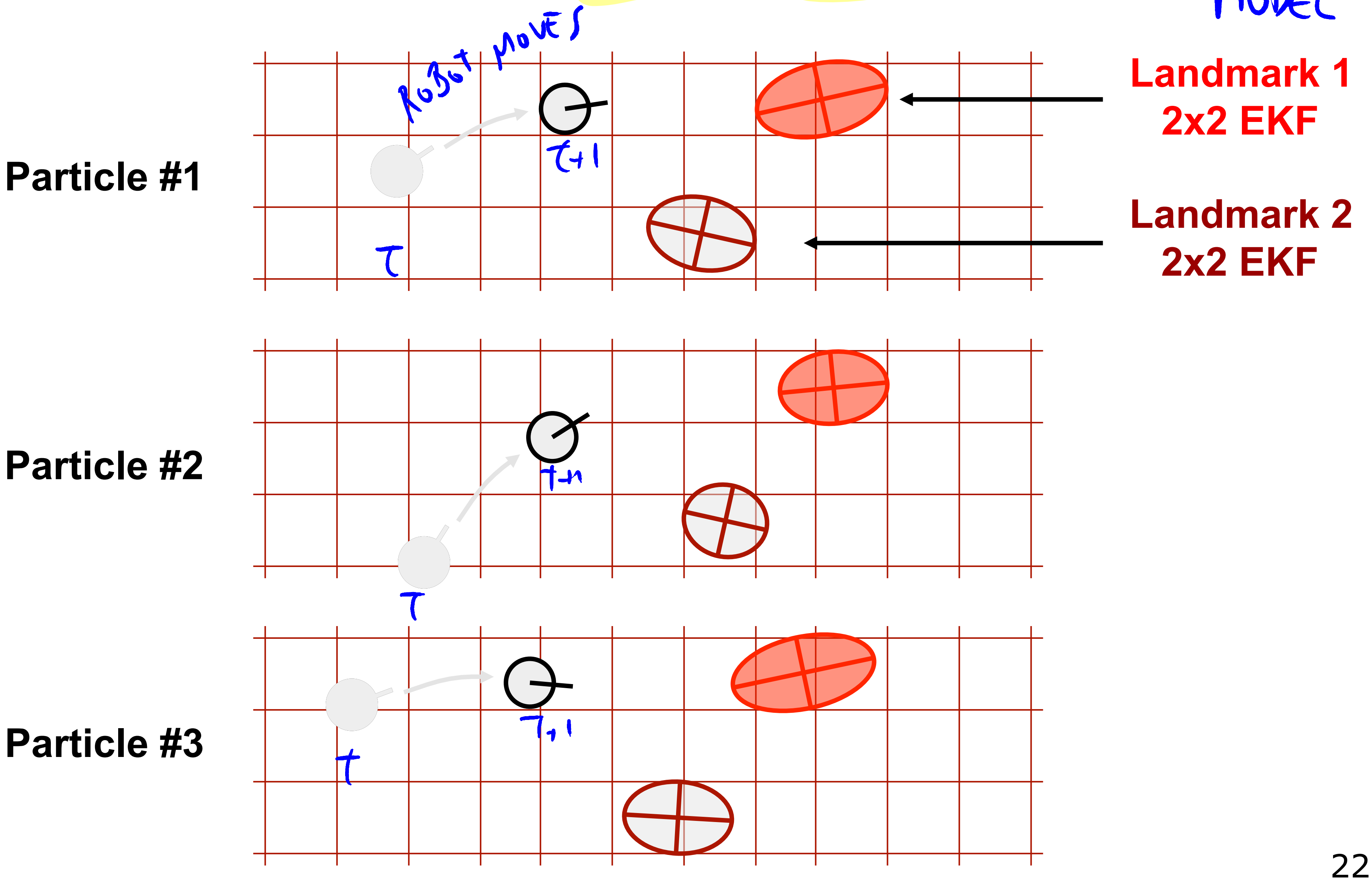
FastSLAM

- 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, θ	EKF1 **Landmark 1**	EKF2 **Landmark 2**	**...**	EKFM **Landmark M**
Particle 2	x, y, θ	**Landmark 1**	**Landmark 2**	**...**	**Landmark M**
⋮					
Particle N	x, y, θ	**Landmark 1**	**Landmark 2**	**...**	**Landmark M**

FastSLAM – Action Update

→ ODOMETRY MODEL



STEP 1/3

INTEGRATE

FastSLAM – Sensor Update

1 MISURA

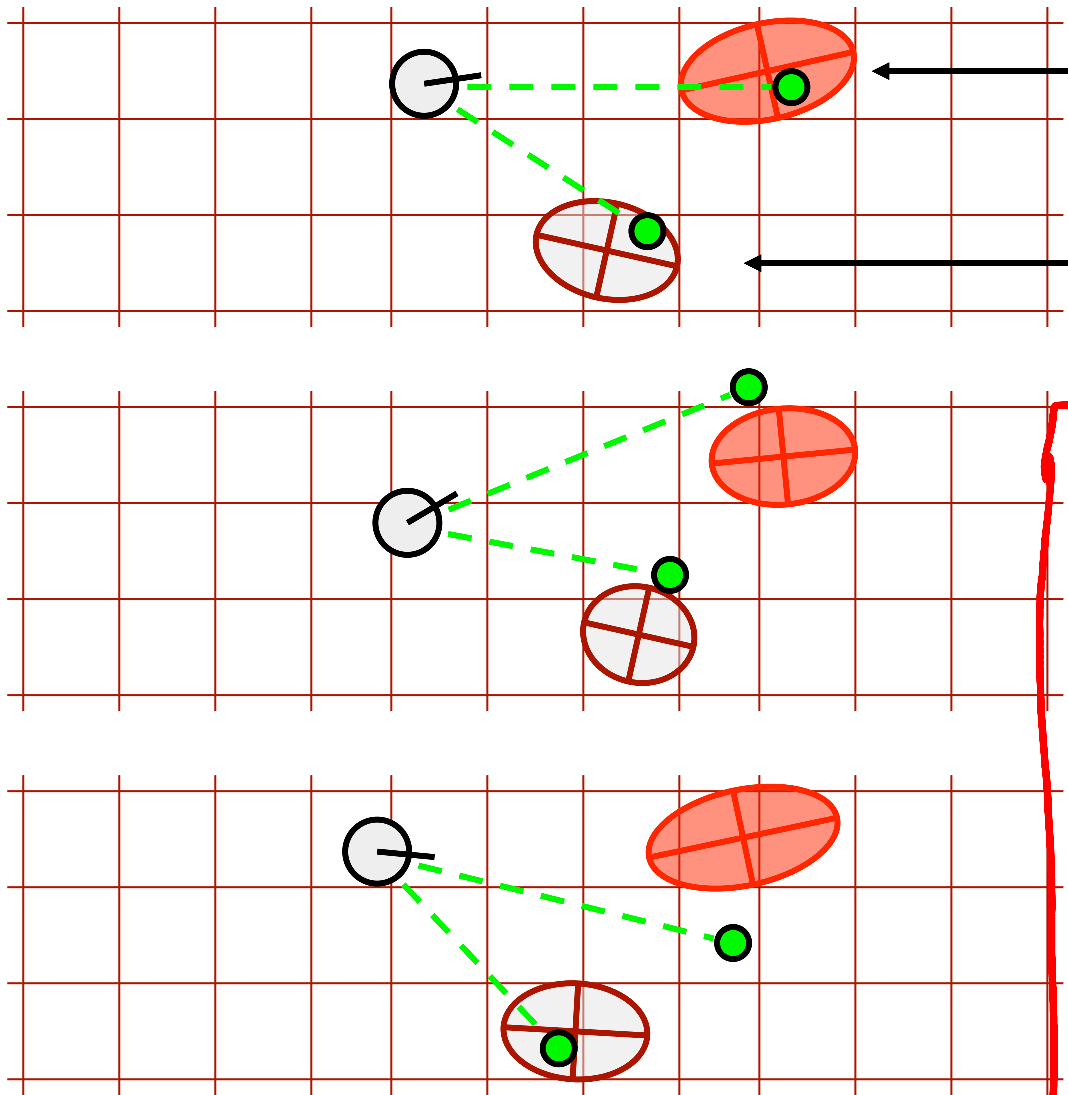
CASO

Particle #1

CASO

Particle #2

Particle #3



Landmark 1
2x2 EKF

Landmark 2
2x2 EKF

QUANTO BENE LA PARTICELLA APPROSSIMA LA MAPPA DATA L'INFORMAZIONE SENSORIALE ?

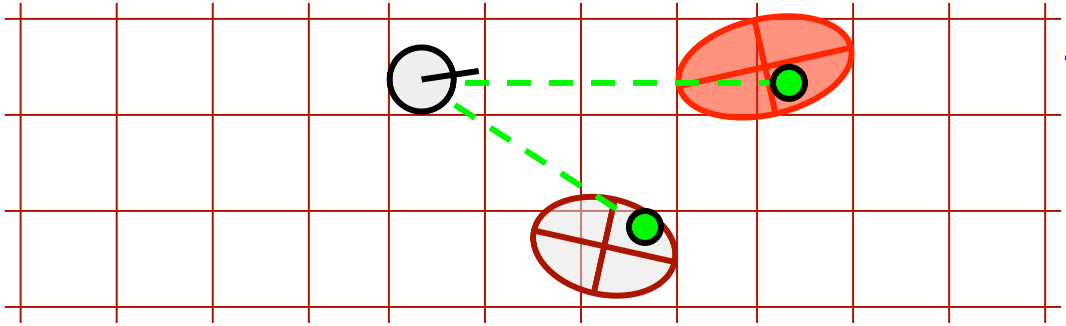
WEIGHT

OVVERO QUANTO SONO "ALLINEATE" LE MISURE w.r.t. la mappa della i-esima particella?

STEP 2/3

FastSLAM – Sensor Update

Particle #1

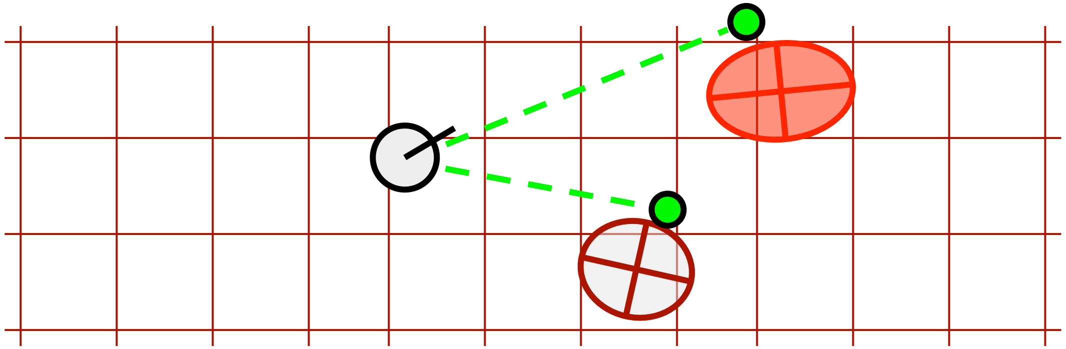


← QUI SEMBRA QUASI OK

Weight = 0.8

OK ✓

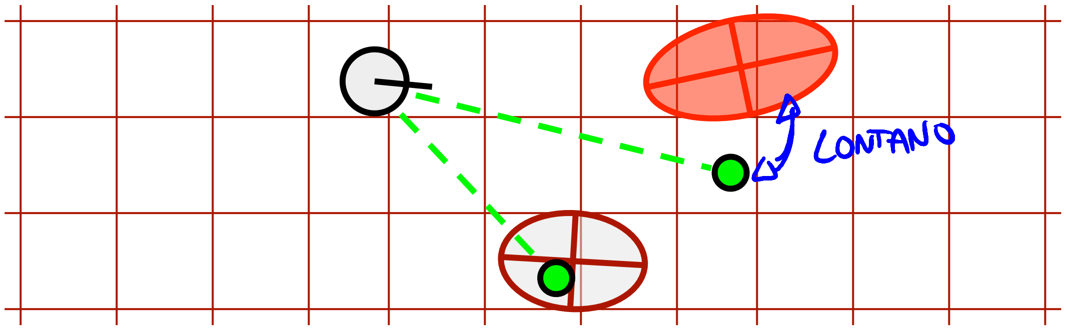
Particle #2



Weight = 0.4

MAH... ?

Particle #3



Weight = 0.1

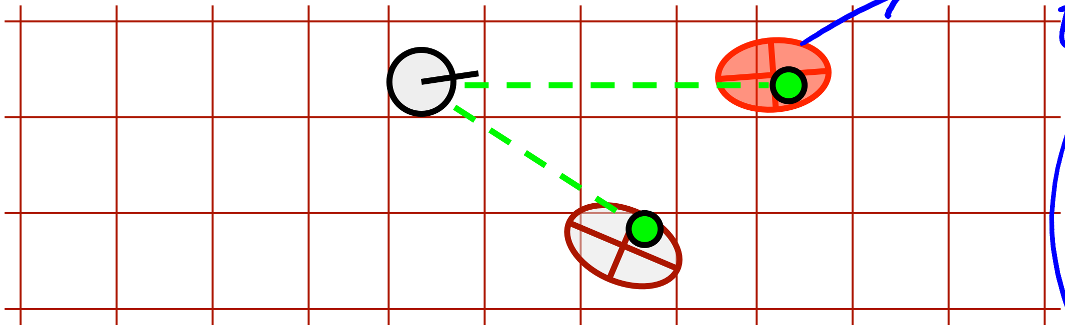
NO ✗

STEP 3/3

FastSLAM – Sensor Update

AGGIORNA I LANDMARK, SONO EKF!

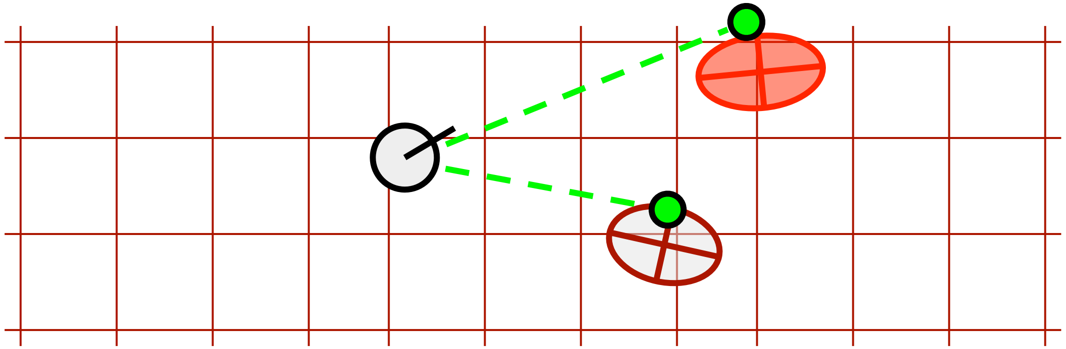
Particle #1



Update map (LIST OF LANDMARKS) of particle 1

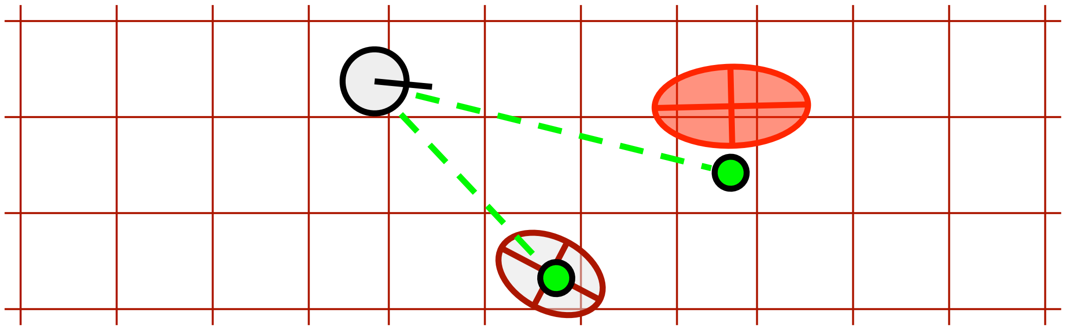
EKF 2x2, E' VELOCE RISPETTO AD EKF-SLAM

Particle #2



Update map of particle 2

Particle #3



Update map of particle 3

RECAP:

Key Steps of FastSLAM 1.0

- (1) PROPAGATE — Extend the path posterior by sampling a new pose for each sample

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

→ MOTION MODEL SAME AS MCL

MONTE CARLO LOCALIZATION

- (2) COMPUTE IMPORTANCE WEIGHTS — Compute particle weight

exp. observation

CALCOLATA PER OGNI PARTICLE

$w^{[k]} = \frac{\text{TARGET}}{\text{PROPOSAL}}$

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

OSSERVAZIONE CORRENTE, i.e. MISURA SENSORE

measurement covariance

QUI DENTRO CI SONO LE INCERTEZZE PROPRIE DEI LANDMARK + INCERTEZZE DELLE OSSERVAZIONI (SENSORI)

- (3) Update belief of observed landmarks (EKF update rule)
- (4) 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 EKF μ: mean Σ: covariance // *loop over all particles*

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

POSE + 1 LANDMARK + 2 ... 3 ... n

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

ODOMETRY MODEL

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*

      $\mu_{j,t}^{[k]} = h^{-1}(z_t, x_t^{[k]})$ INVERSE OBS. FUNCTION — *// initialize mean* non abbiamo mai visto la feature (landmark) quindi la inizializziamo con la sua misura

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

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

      $w^{[k]} = p_0$ — *// default importance weight* STANDARD VALUE

    else

NEW ONE

INCERTEZZA INTRINSECA DELLA MISURA (COVARIANZA)

FastSLAM 1.0 – Part 2

else SE INVECE NON E' UNA NUOVA FEATURE/LANDMARK ...

$\langle \mu_{j,t}^{[k]}, \Sigma_{j,t}^{[k]} \rangle =$ *EKF-Update(...)* // update landmark

EKF-PREDICTION? NO! QUELLA E' FATTA CON LE PARTICLE!

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

measurement cov. $Q = H \underbrace{\Sigma_{j,t-1}^{[k]}}_{\text{INCERTEZZA CHE AVEVAMO AL TEMPO PRECEDENTE}} H^T + Q_t$ **exp. observation** ∀ PARTICLE

INCERTEZZA CHE AVEVAMO AL TEMPO PRECEDENTE + INCERTEZZA SULLA MISURA

endif

QUELLE CHE NON VEDIAMO LASCIAMOLE COSÌ COME SONO

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

$\mathcal{X}_t = \text{resample}\left(\left\langle x_t^{[k]}, \left\langle \mu_{1,t}^{[k]}, \Sigma_{1,t}^{[k]} \right\rangle, \ldots, w^{[k]} \right\rangle_{k=1,\ldots,N}\right)$

return $\mathcal{X}_t$

ESPANDIAMO LA PARTE DI EKF-UPDATE

FastSLAM 1.0 – Part 2 (long)

else

EKF update

$$\hat{z}^{[k]} = h(\mu_{j,t-1}^{[k]}, x_t^{[k]}) \qquad \textit{// measurement prediction}$$

$$H = h'(\mu_{j,t-1}^{[k]}, x_t^{[k]}) \qquad \textit{// calculate Jacobian}$$

$$Q = \boxed{H\ \Sigma_{j,t-1}^{[k]}\ H^T + Q_t} \qquad \textit{// measurement covariance}$$

$$K = \Sigma_{j,t-1}^{[k]}\ H^T\ Q^{-1} \qquad \textit{// calculate Kalman gain}$$

$$\mu_{j,t}^{[k]} = \mu_{j,t-1}^{[k]} + K(z_t - \hat{z}^{[k]}) \qquad \textit{// update mean}$$

$$\Sigma_{j,t}^{[k]} = (I - K\ H)\Sigma_{j,t-1}^{[k]} \qquad \textit{// update covariance}$$

PERCHE' →

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

?

endif

for all unobserved features j' do

$$\langle \mu_{j',t}^{[k]}, \Sigma_{j',t}^{[k]} \rangle = \langle \mu_{j',t-1}^{[k]}, \Sigma_{j',t-1}^{[k]} \rangle \qquad \textit{// leave unchanged}$$

endfor

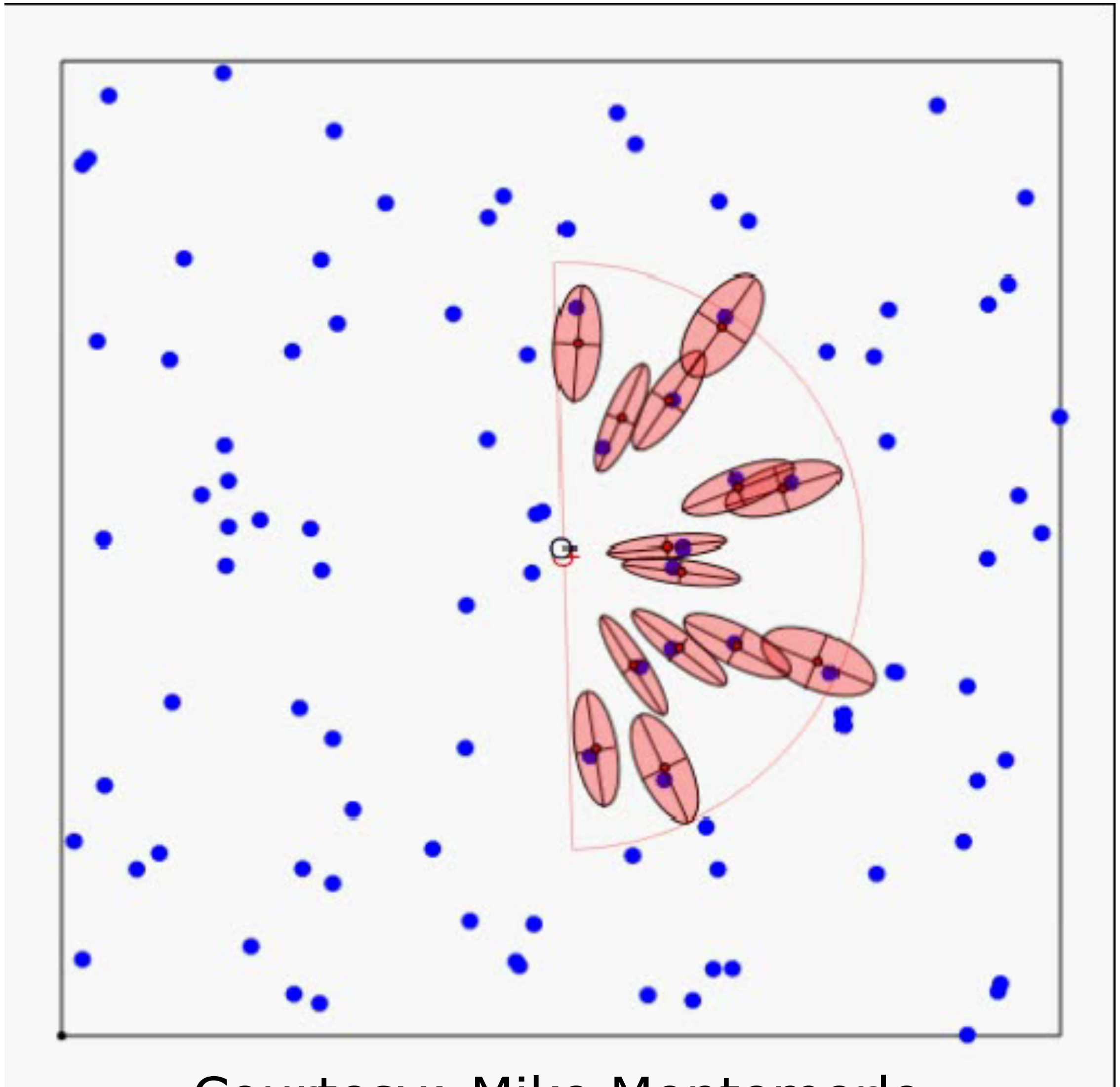
endfor

$$\mathcal{X}_t = \text{resample}\left(\left\langle x_t^{[k]}, \left\langle \mu_{1,t}^{[k]}, \Sigma_{1,t}^{[k]} \right\rangle, \ldots, w^{[k]} \right\rangle_{k=1,\ldots,N}\right)$$

return $\mathcal{X}_t$

SEARCH ON YOUTUBE: "FAST SLAM 1.0 with 100 particles and 150 obstacles"

FastSLAM in Action



Courtesy: Mike Montemerlo

WHY THE IMPORTANCE WEIGHT IS LIKE THIS?

The Weight is a Result From the Importance Sampling Principle

- Importance weight is given by the ratio of target and proposal in $x^{[k]}$
- See: importance sampling principle

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

THE DISTRIBUTION WE WANT TO APPROXIMATE

DISTRIBUZIONE DALLA QUALE ABBIAMO CREATO I "SAMPLE", LE IPOTESI

The Importance Weight

- The target distribution is

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

VOGLIAMO APPROSSIMARE LE POSIZIONI /"TRAIETTORIE"/, DATE LE OSSERVAZIONI ED I CONTROLLI

- The proposal distribution is

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

TUTTE LE INFORMAZIONI ECCETTO L'ULTIMA MISURA (OSSERVAZIONI)

- Proposal is used step-by-step, CON ASSUNZIONE MARKOVIANA...

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

PRENDIAMO LO STATO AL TEMPO PRECEDENTE ED ANDIAMO 1-PASSO IN AVANTI.

ODOMETRIA

CIO' CHE VIENE DAI BELIEF PRECEDENTI

The Importance Weight

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

The Importance Weight

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

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

Bayes rule + factorization

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

The Importance Weight

BAYES + FACTORIZATION

$p(z \mid x_{1:t}, z_{t-1}, \mu_{1:T})$ *

$p(x_t \mid x_{t-1}, \mu_t)$ *

CURRENT POSE GIVEN THE PREVIOUS

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

BELIEF of ALL PREVIOUS POSES

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

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

BAYES RULE:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

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

normaliz

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

CURRENT

$-$ x $-$ $+u$

The Importance Weight

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

FATTA LA DERIVAZIONE NOTIAMO CHE, SEMPLIFICANDO:

The Importance Weight

$$
\begin{aligned}
w^{[k]} &= \frac{\text{target}(x^{[k]})}{\text{proposal}(x^{[k]})} \\
&= \frac{p(x_{1:t}^{[k]} \mid z_{1:t}, u_{1:t})}{p(x_t^{[k]} \mid x_{t-1}, u_t)\; p(x_{1:t-1}^{[k]} \mid z_{1:t-1}, u_{1:t-1})} \\
&= \frac{\eta\, p(z_t \mid x_{1:t}^{[k]}, z_{1:t-1})\; \cancel{p(x_t \mid x_{t-1}^{[k]}, u_t)}}{\cancel{p(x_t^{[k]} \mid x_{t-1}^{[k]}, u_t)}} \\
&\quad \frac{\cancel{p(x_{1:t-1}^{[k]} \mid z_{1:t-1}, u_{1:t-1})}}{\cancel{p(x_{1:t-1}^{[k]} \mid z_{1:t-1}, u_{1:t-1})}} \\
W^{[k]} &= \eta\, p(z_t \mid \underbrace{x_{1:t}^{[k]}, z_{1:t-1}}_{\text{ALL THE POSES + PAST OBSERVATIONS}})
\end{aligned}
$$

normalization: per far sì che i pesi sommino ad 1

ALL THE POSES + PAST OBSERVATIONS ma:

The Importance Weight

- Integrating over the pose of the observed landmark leads to

$$
\begin{aligned}
w^{[k]} & \\
&= \eta\, p(z_t \mid x_{1:t}^{[k]}, z_{1:t-1}) \\
&= \eta \underbrace{\int p(z_t \mid x_{1:t}^{[k]}, z_{1:t-1}, m_j)\, p(m_j \mid x_{1:t}^{[k]}, z_{1:t-1})\, dm_j}_{\text{MARGINALIZZAZIONE}}
\end{aligned}
$$

SPECIFICO LANDMARK

m_j DATO IL RESTO

The Importance Weight

- Integrating over the pose of the observed landmark leads to

$$
\begin{aligned}
w^{[k]} & \\
&= \eta\, p(z_t \mid x_{1:t}^{[k]}, z_{1:t-1}) \\
&= \eta \int p(z_t \mid x_{1:t}^{[k]}, z_{1:t-1}, m_j)\, p(m_j \mid x_{1:t}^{[k]}, z_{1:t-1})\, dm_j \\
&= \eta \int p(z_t \mid x_t^{[k]}, m_j)\ p(m_j \mid x_{1:t-1}^{[k]}, z_{1:t-1})\ dm_j
\end{aligned}
$$

DATO X_t, DELLE PASSATE MISURE POSSIAMO FARNE A MENO

QUI: SE ABBIAMO LE MISURE z fino a $t-1$, di X_t NON CE NE FACCIAMO NIENTE

(se sappiamo la posizione nel FUTURO, questo non ci aiuta a conoscere la posizione del LANDMARK da)

The Importance Weight

- Integrating over the pose of the observed landmark leads to

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

SAME AS EKF →

LIKELIHOOD OF THE CURRENT z_t

GIVEN THE PREDICTED $\hat{z}$ AND

THE UNCERTAINTY OF THE OBSERVATION

LIKELIHOOD OF AN OBSERVATION GIVEN POSE AND THE LANDMARK LOCATION

CHE PROBLEMA È QUESTO: ? DATE LE POSIZIONI DEL ROBOT 1...t E LE MISURE 1...t

⇓

MAPPING WITH KNOWN POSES

The Importance Weight

- This leads to

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

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

measurement covariance (pose uncertainty of the landmark estimate plus measurement noise)

The Importance Weight

- This leads to

$$w^{[k]} = \eta \int \underbrace{p(m_j \mid x^{[k]}_{1:t-1}, z_{1:t-1})}_{\mathcal{N}(m_j;\mu^{[k]}_{j,t-1},\Sigma^{[k]}_{j,t-1})} \; \underbrace{p(z_t \mid x^{[k]}_t, m_j)}_{\mathcal{N}(z_t;\hat{z}^{[k]},Q_t)} \, dm_j$$

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

QUANTO SONO CERTO SULLA POSIZIONE DEL LANDMARK + UNCERTAINTY OF THE SENSOR

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

IS SIMILAR BUT NOT EXACT DUE TO LINEARIZATION INTRODUCED WITH EKF

FastSLAM 1.0 – Part 2

QUANTO E' "CONSISTENTE LA RAPPRESENTAZIONE DEL MONDO CHE I SAMPLE/PARTICLES HANNO GENERATO", CON QUANTO IL ROBOT "VEDE" IN QUESTO MOMENTO?

else

$\langle \mu_{j,t}^{[k]}, \Sigma_{j,t}^{[k]} \rangle = \textit{EKF-Update}(\dots)$ // *update landmark*

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

COSA VEDE IL ROBOT

"COSA STIMA LA PARTICELLA"

SENSOR NOISE + INCERTEZZA PARTICELLA

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

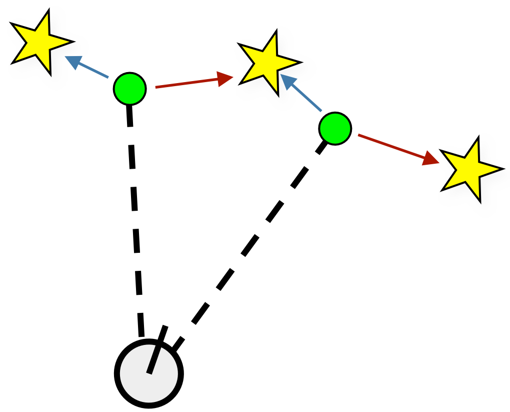
$$\boxed{\mathcal{X}_t = \text{resample}\left(\left\langle x_t^{[k]}, \left\langle \mu_{1,t}^{[k]}, \Sigma_{1,t}^{[k]} \right\rangle, \dots, w^{[k]} \right\rangle_{k=1,\dots,N} \right)}$$

FINAL STEP

return $\mathcal{X}_t$

Data Association Problem

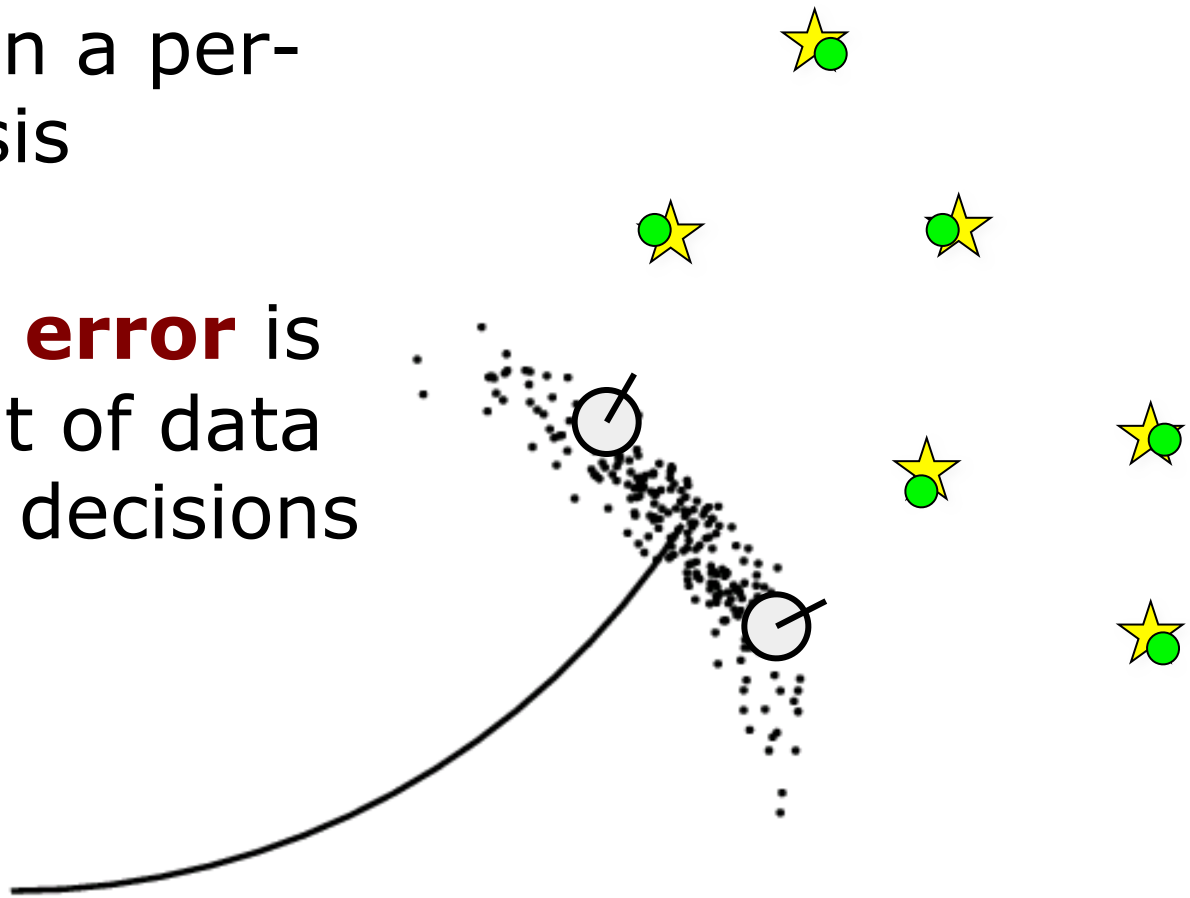
- Which observation belongs to which landmark?



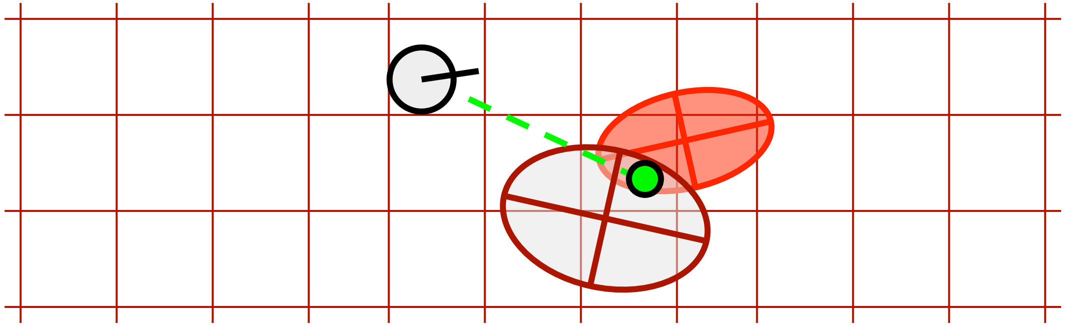
- More than one possible association
- **Potential data associations depend on the pose of the robot**

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



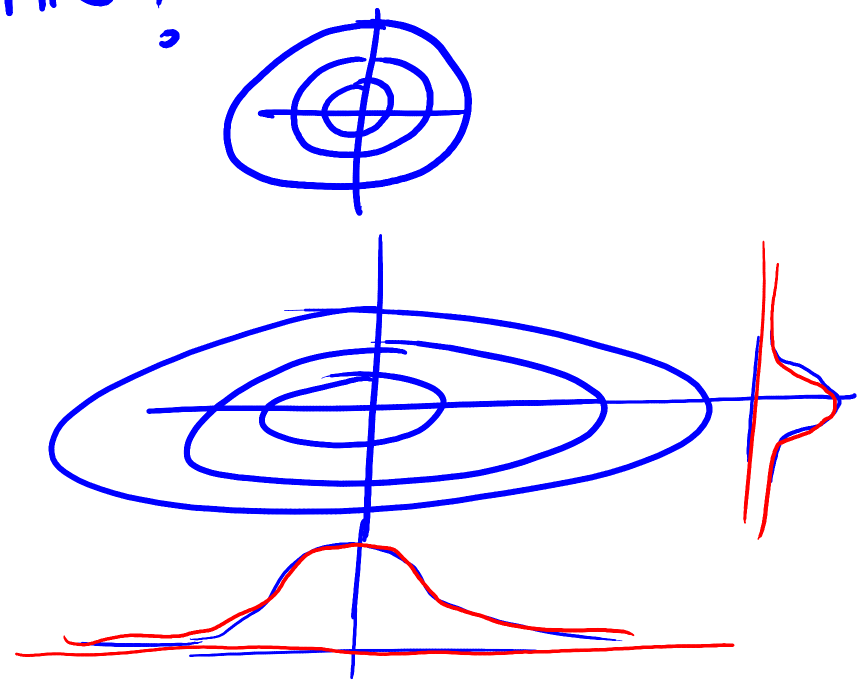
Was the observation generated by the **red** or by the **brown** landmark?

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

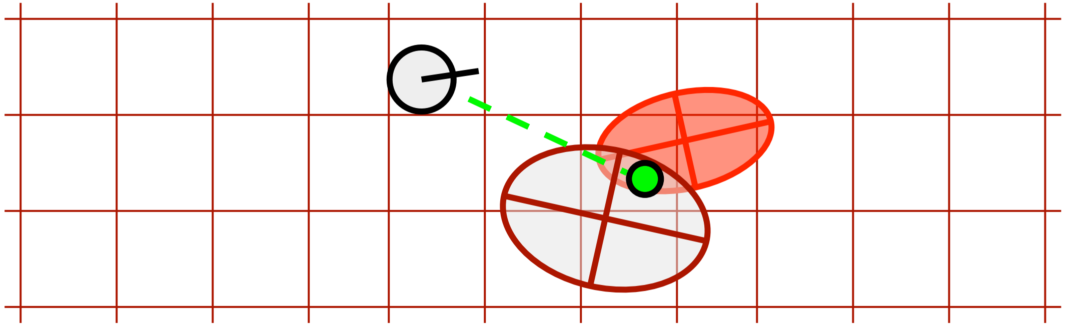
QUALE DISTANZA USARE?

EUCLIDEA

MAHALANOBIS



Per-Particle Data Association

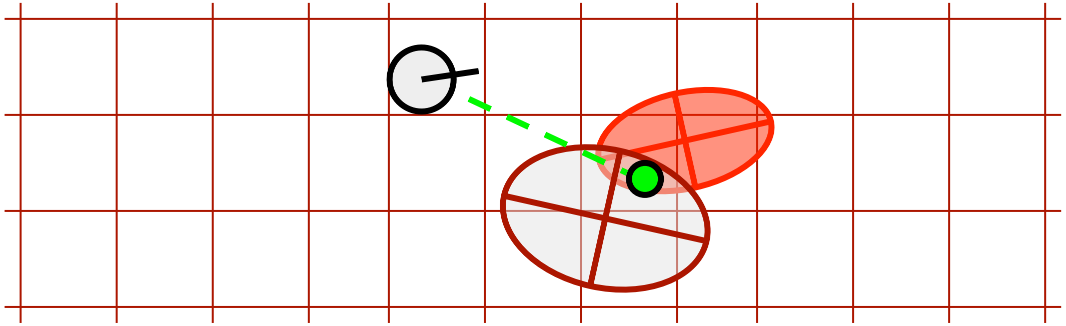


Was the observation generated by the **red** or by the **brown** landmark?

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

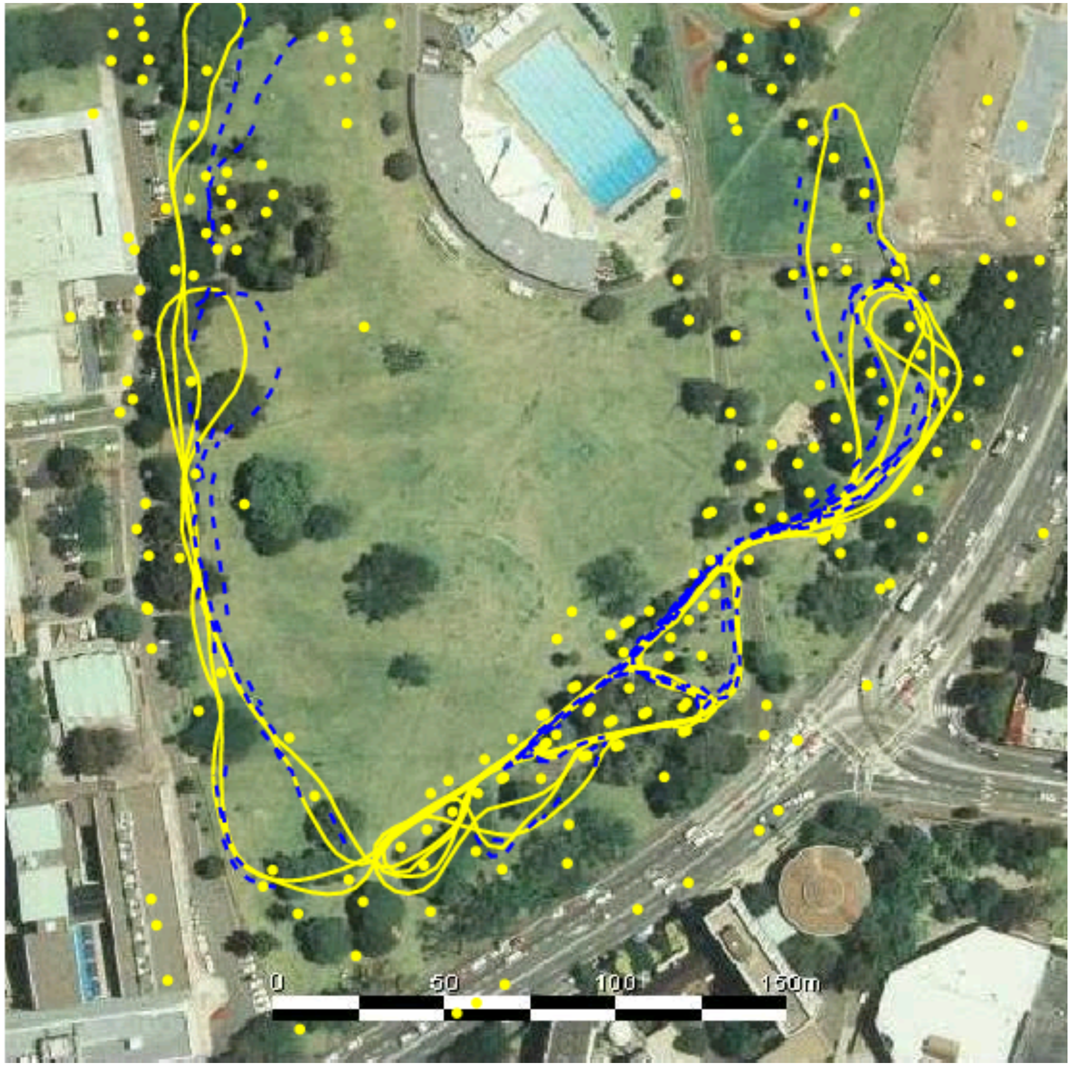


Was the observation generated by the **red** or by the **brown** landmark?

- 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



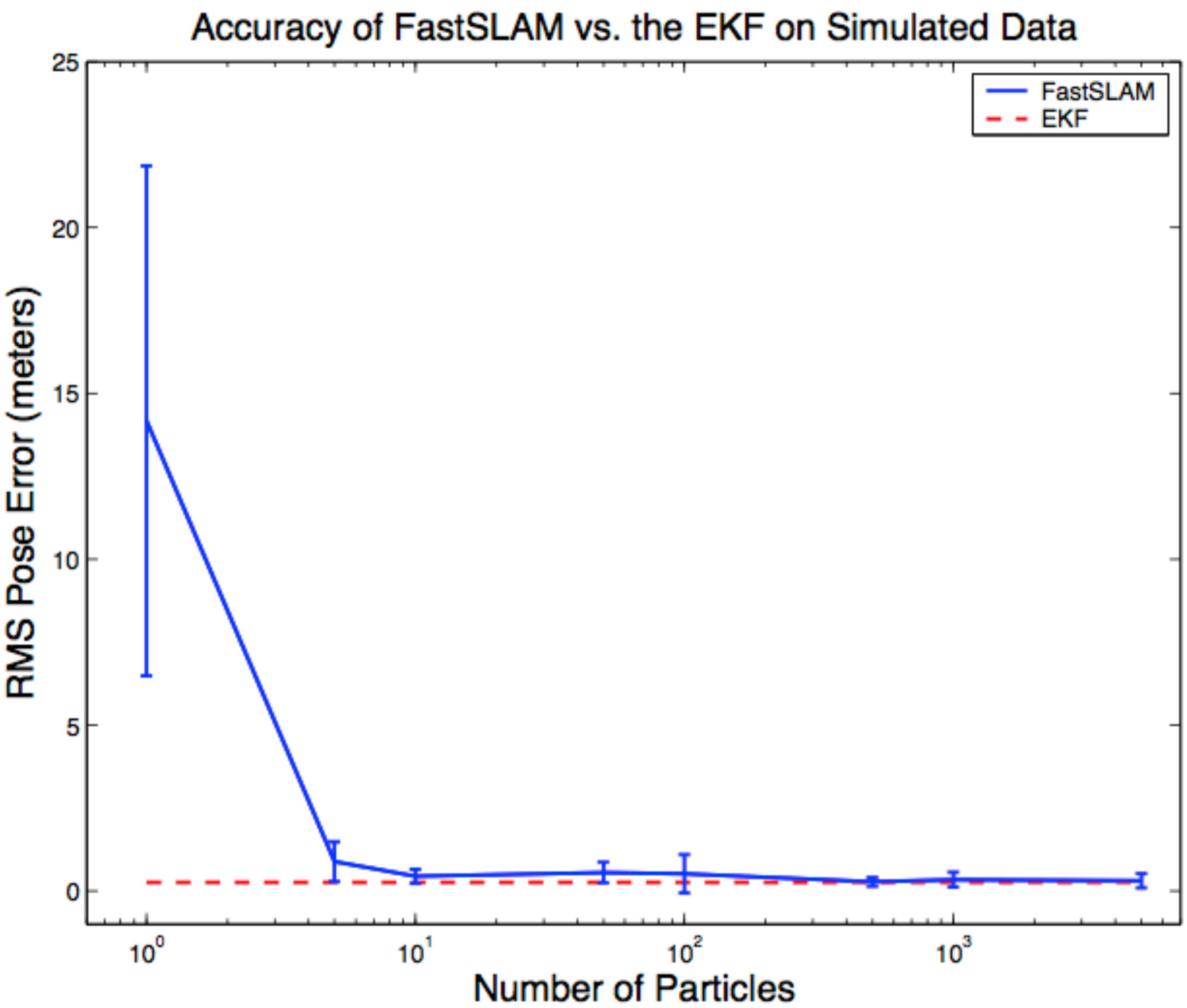
Blue = GPS
Yellow = FastSLAM

Courtesy: Mike Montemerlo

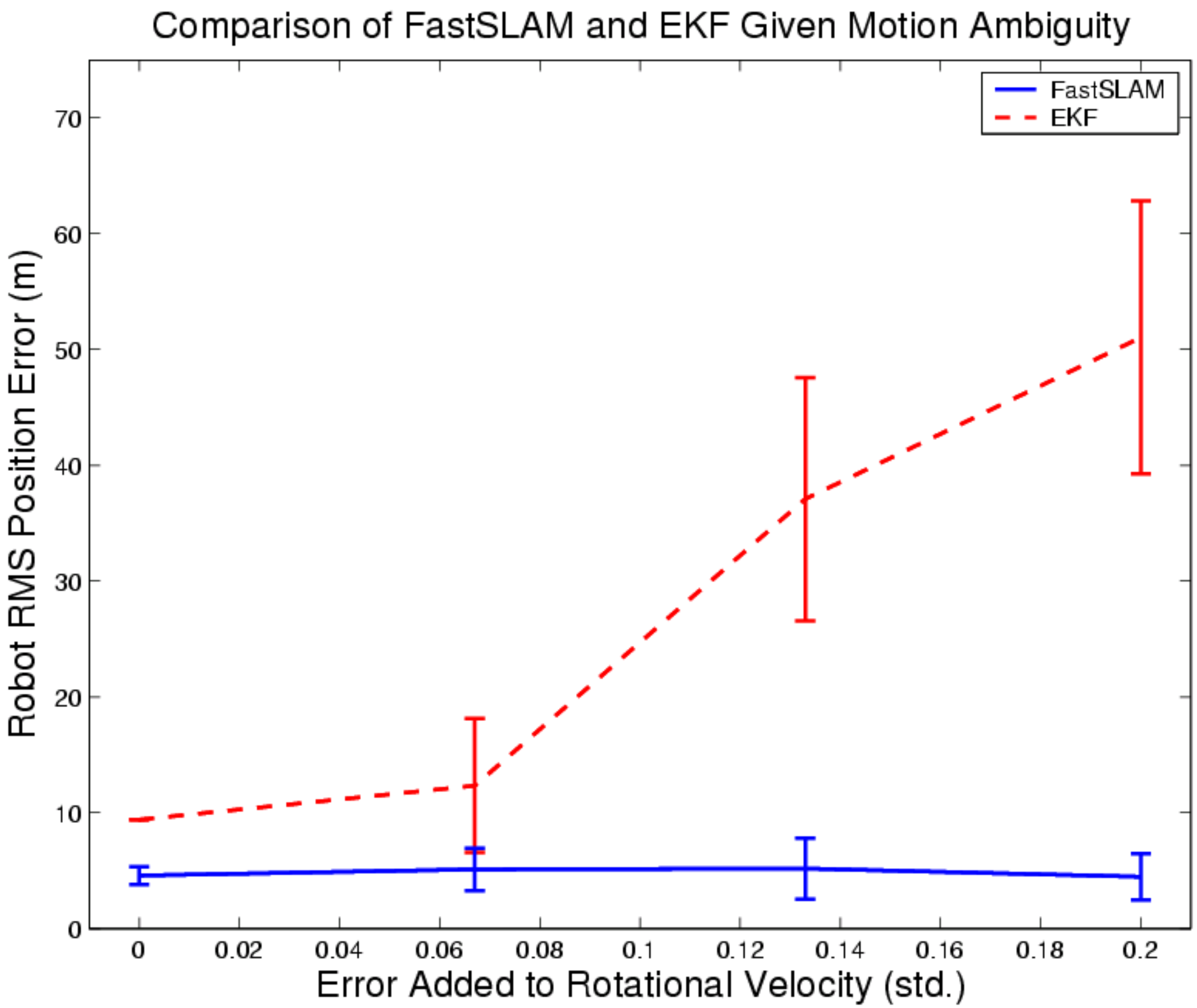
Results – Victoria Park (Video)

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

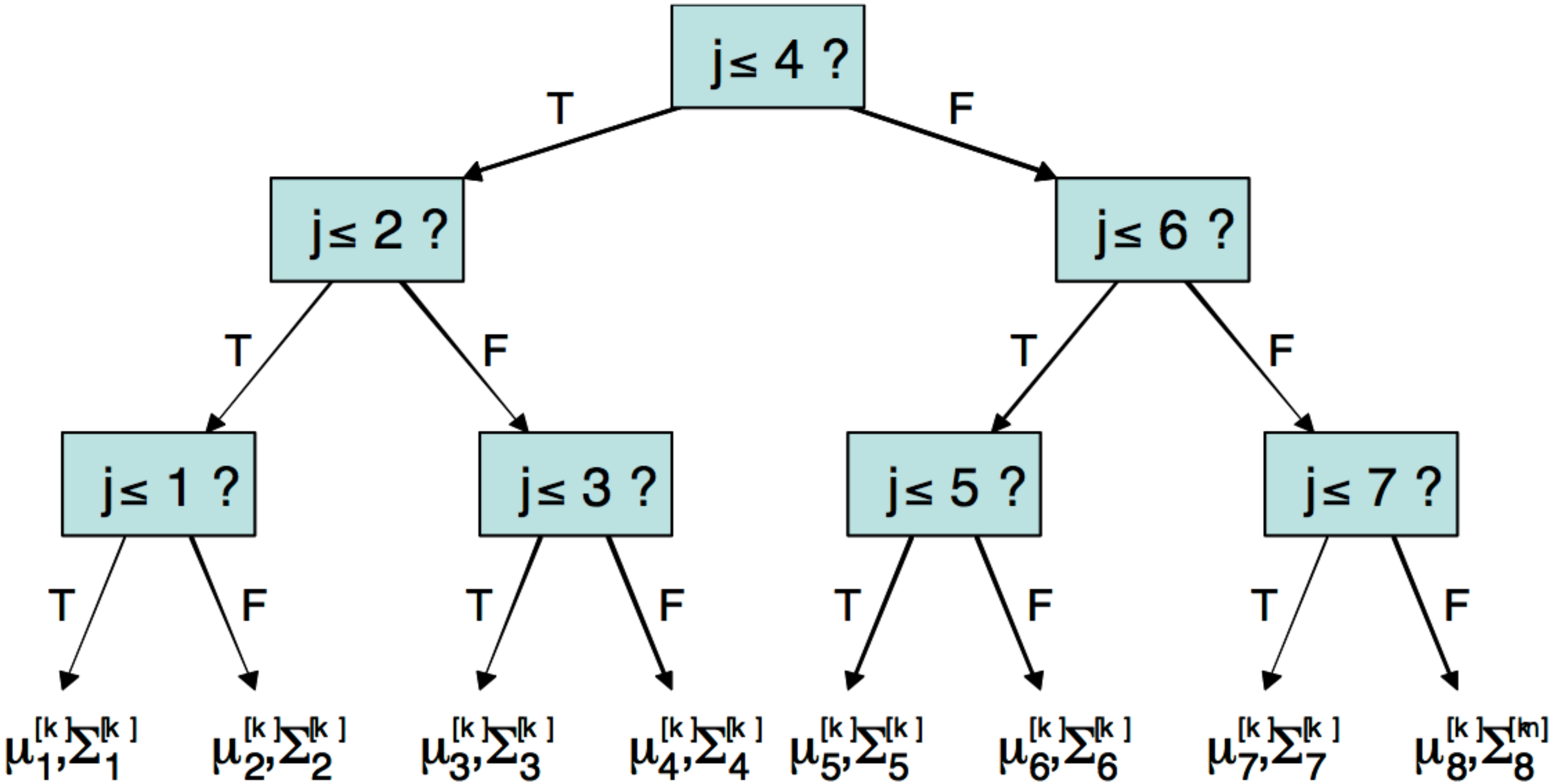
FastSLAM Complexity – Simple Implementation

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

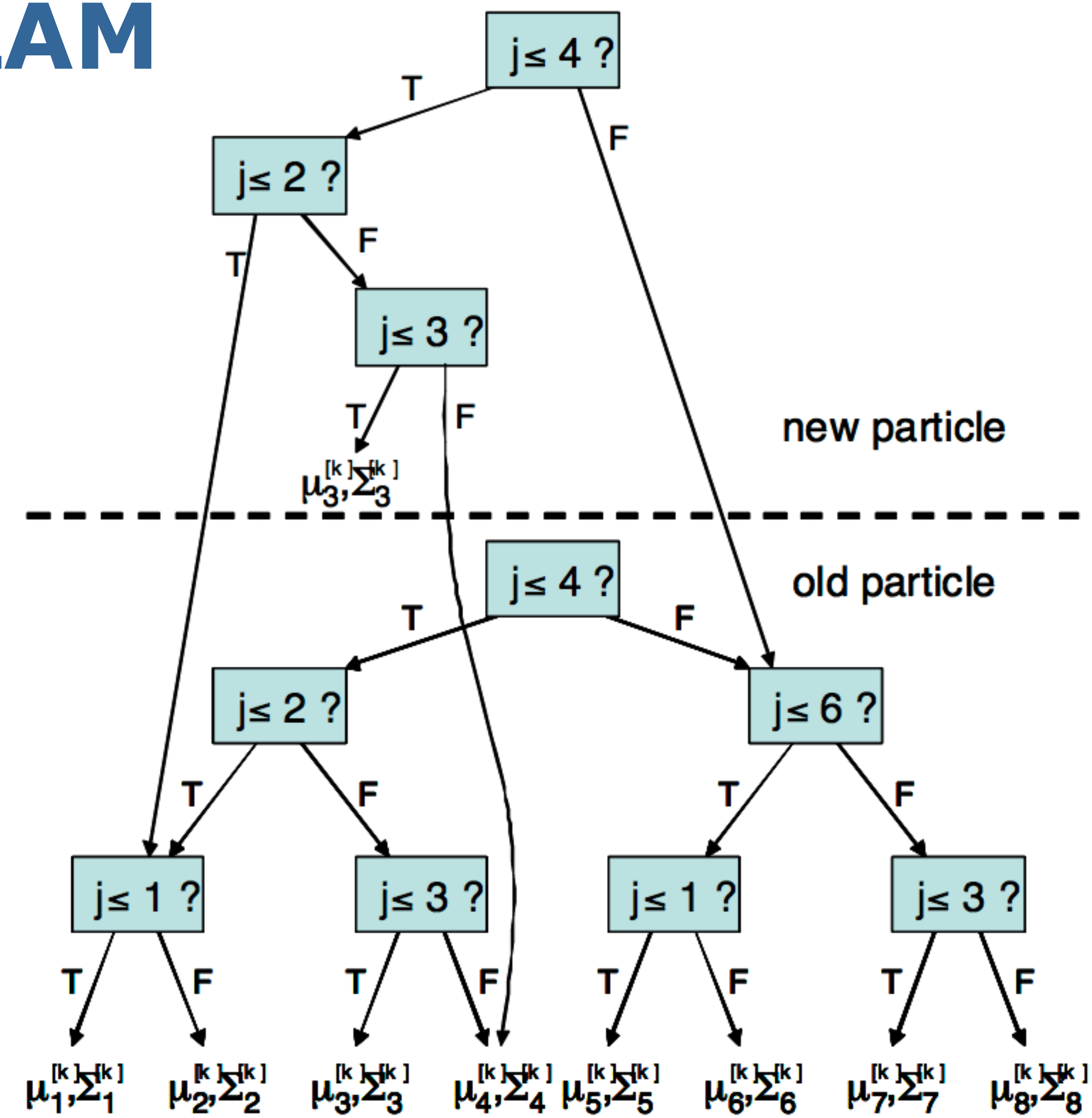
N = Number of particles
M = Number of map features

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

A Better Data Structure for FastSLAM



A Better Data Structure for FastSLAM



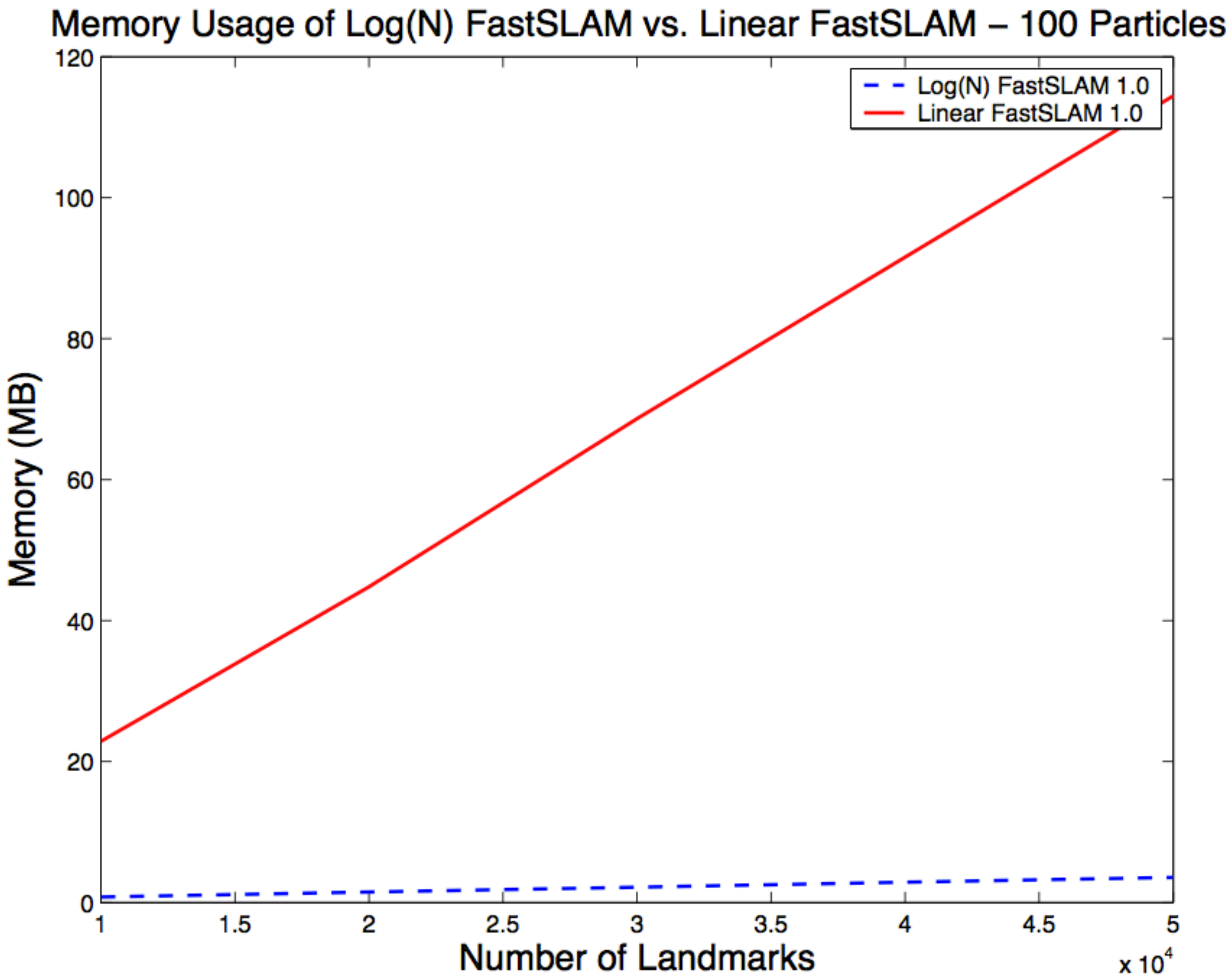
FastSLAM Complexity

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

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

N = Number of particles
M = Number of map features

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

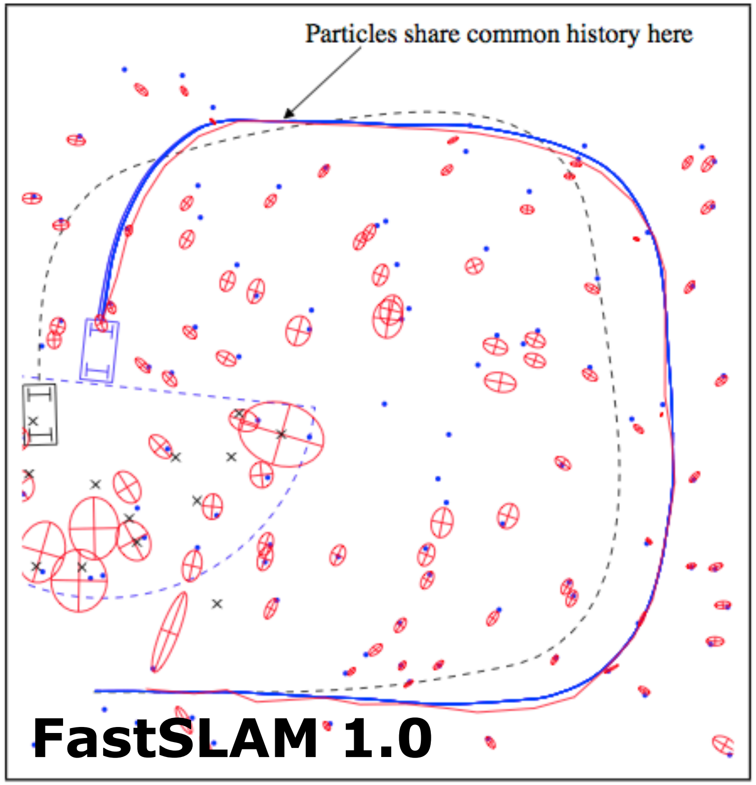
$$x_t^{[k]} \sim p(x_t \mid x_{1:t-1}^{[k]}, u_{1:t}, z_{1:t})$$

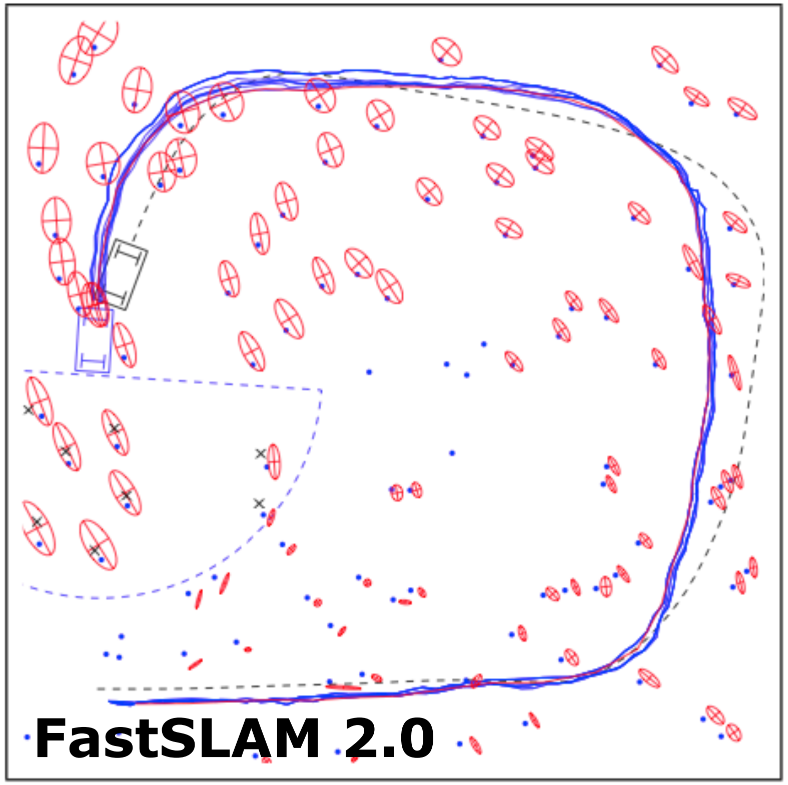
- Results in a more peaked proposal distribution
- Less particles are required
- More robust and accurate
- But more complex…

[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