Kapitel 6: Kartierung

1. Zum Einstieg: Worum geht es?
2. Sensorik
3. Sensordatenverarbeitung
4. Fortbewegung
5. Lokalisierung in [...]
6. **Kartierung**
7. Navigation
8. Umgebungsdate[...]
9. Roboterkontrollarchitekturen

   Ausblick

6.1 Überblick

6.2 Kartierung bei bekannten Posen

6.3 Inkrementelles SLAM

6.4 Vollständiges SLAM

SLAM (Simultaneous Localization & Mapping)
CML (Concurrent Mapping & Localization)

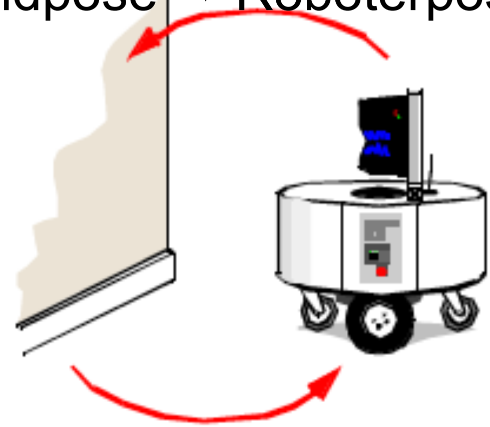
Definition *Lernen* (KI)

„… changes in the system that are adaptive in the sense that they enable the system to do the same task or tasks drawn from the same population more efficiently next time.“

Herbert Simon

Roboterkartierung ist ein Henne-Ei-Problem!

Wandpose ↪ Roboterpose



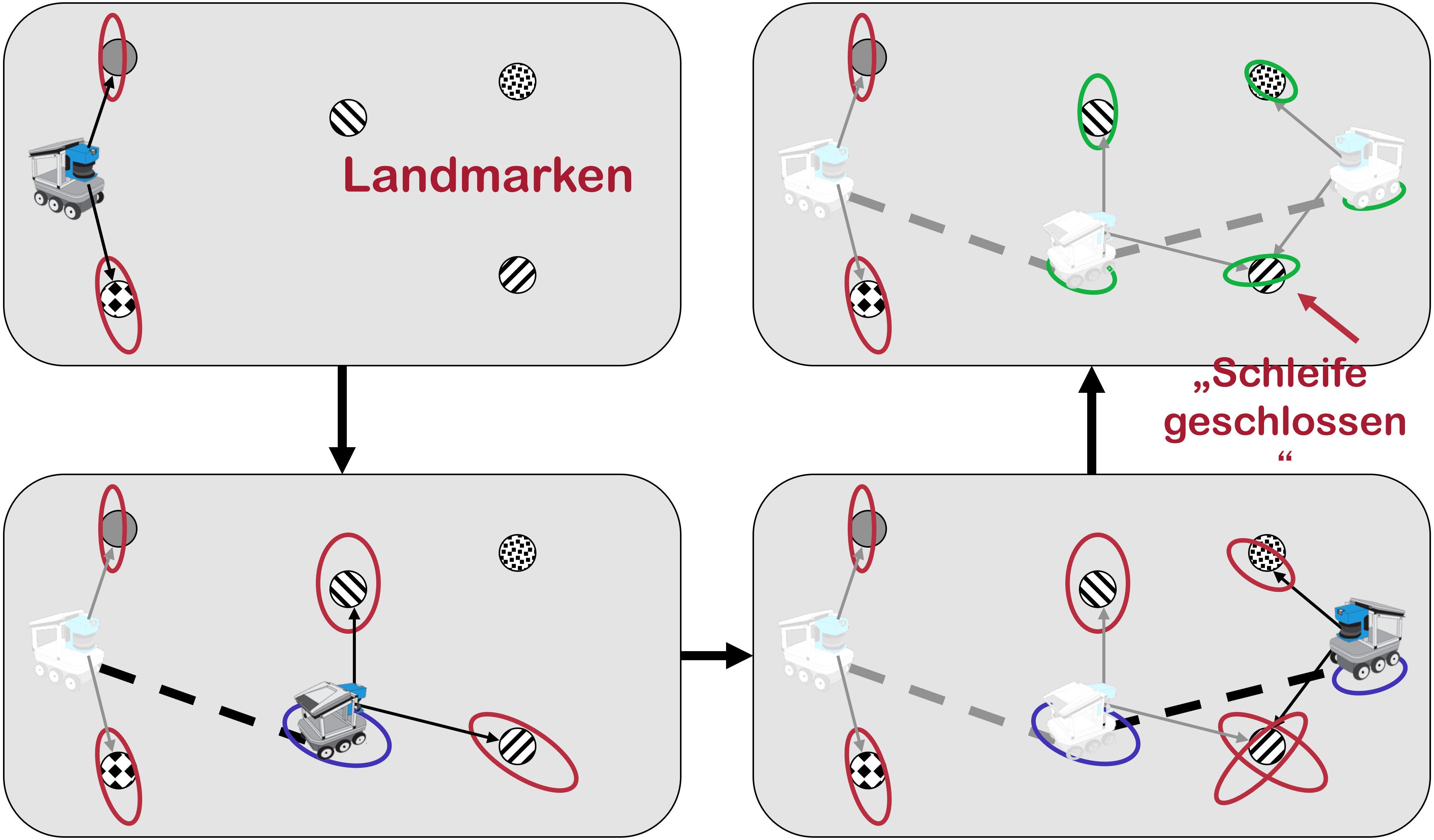
Roboterpose ↪ Wandpose

Ausgehend von gegebener Pose und evtl. vorliegender Karte:

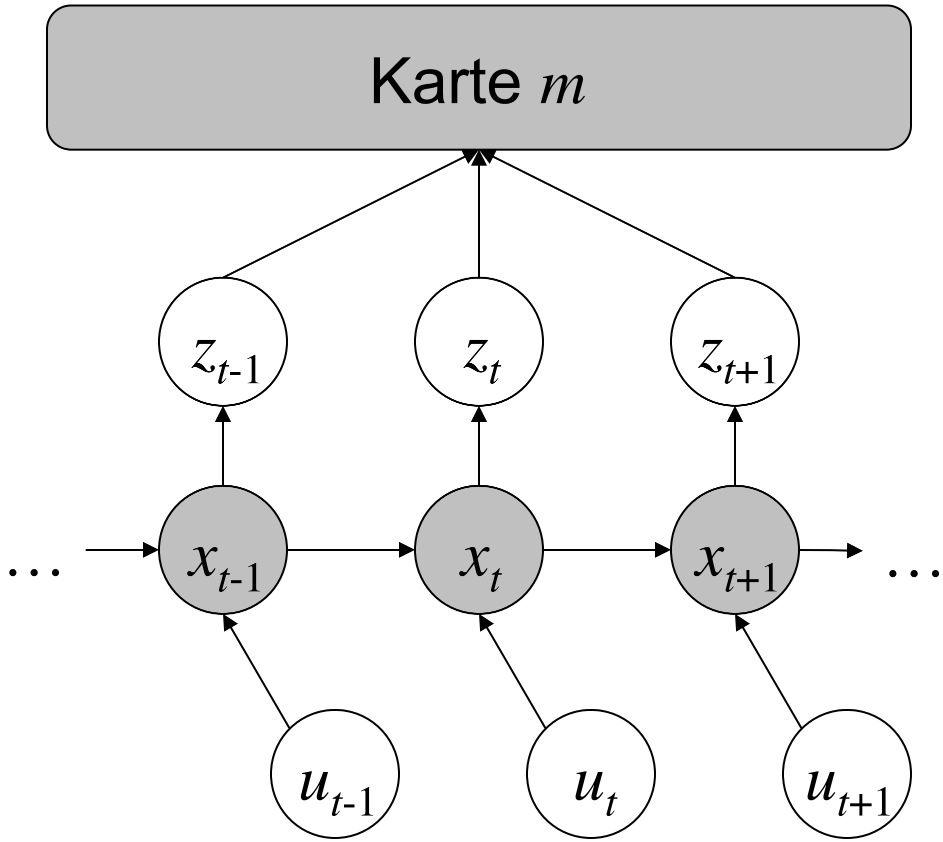
- Beim Bewegen durch die Umgebung,
- nur mit Hilfe der Sensorik an Bord,
- erstell eine Karte vorgegebener Art (metrisch, topologisch,…)

Poseplanung/Explorationsstrategie („Wohin als nächstes?“) noch ausgelassen!

Struktur Kartierungsproblem, intuitiv



Struktur Kartierungsproblem, mathematisch



Gegeben: Folge von Evidenzen (Beobachtungen z_i von Landmarken, Bewegungen u_i), i.a. mit Unsicherheit behaftet

Finde: Posen x_i der Beobachtungen und Karte m mit Positionen der Landmarken, die u_i und z_i optimal integrieren

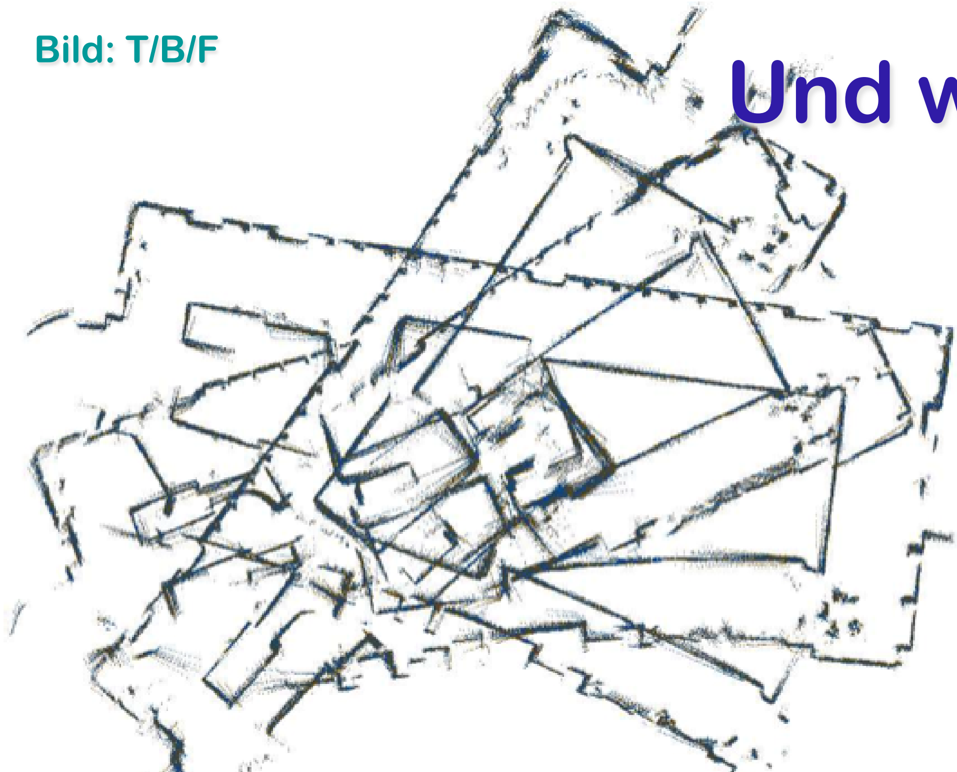
Probabilistische Formulierung

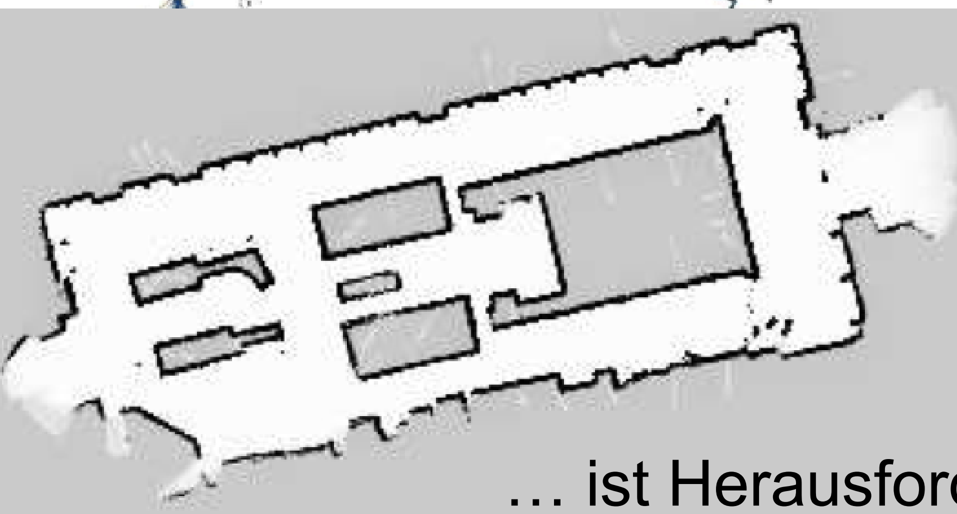
Berechne Verteilung

$$\mathbf{P}\left(\mathbf{m}, \mathbf{x}_{1:t} \middle| z_{1:t}, u_{1:t-1}\right)$$

Bild: T/B/F

Und warum ist das schwer?





- Messfehler sind z.T. unbeschränkt und statistisch nicht unabhängig (Odometrie!)
- Kartierung ist ein sehr hochdimensionales Problem (zB n Laserscans à 181 2D-Punkte)
- **Assoziation** von Sensordaten (mehrere Messdaten bezeichnen selben Punkt/selbes Umgebungsmerkmal) schwer zu ermitteln <u>ohne Karte</u>

„Schleifen schließen“

… ist Herausforderung und Hilfe beim Kartieren!

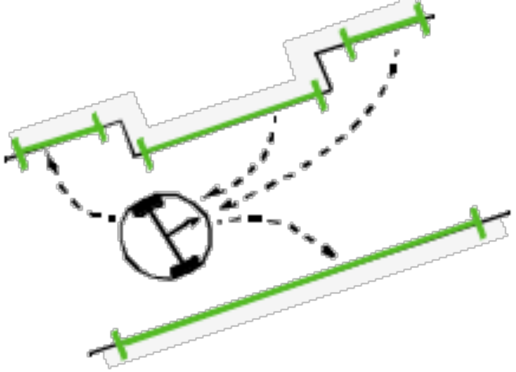
Varianten des Kartierungsproblems

- **Kartierung bei bekannten Posen**:
  „Basisfall" und fürs Problemverständnis

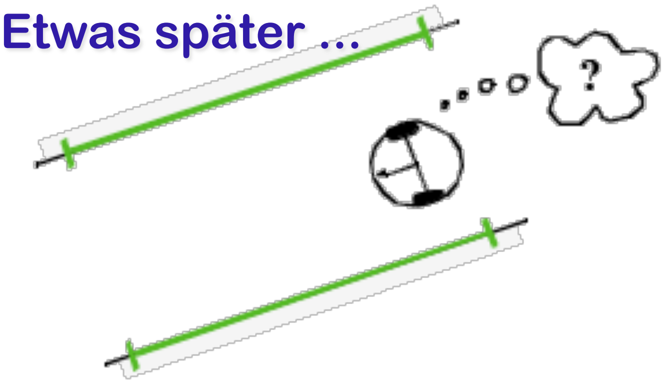
- **Inkrementelles SLAM**:
  SLAM unter Berücksichtigung nur der aktuellen Pose,
  d.h. Berechnung von $\mathbf{P}\left(m, x_t \middle| z_{1:t}, u_{1:t-1}\right)$

- **Volles SLAM**:
  Berechnung von $\mathbf{P}\left(m, x_{1:t} \middle| z_{1:t}, u_{1:t-1}\right)$

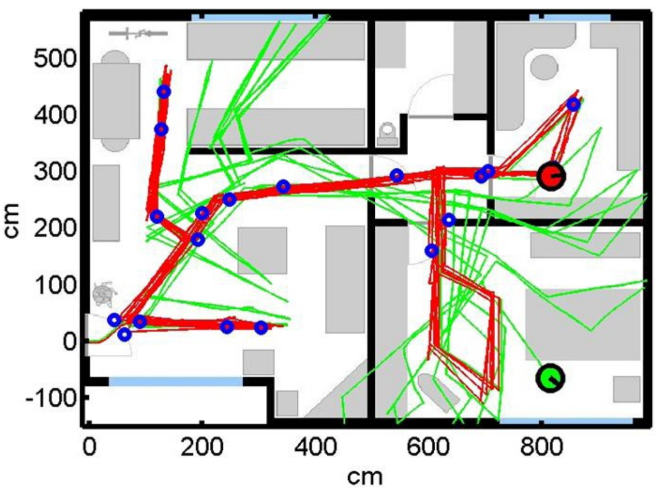
Themen, die wir auslassen



- SLAM in dynamischen Umgebungen
  - z.B. Hähnel, Triebel, Burgard & Thrun: *Map building with mobile robots in dynamic environments*. ICRA2003



- Kartierung basierend auf visuellen Umgebungsmerkmalen, z.B. SIFT-Merkmale
  - z.B. visual SLAM, Evolution Robotics vSLAM™



6.2 Kartierung bei bekannten Posen

… kommt vor z.B. im Freiland mit GPS und Kompass

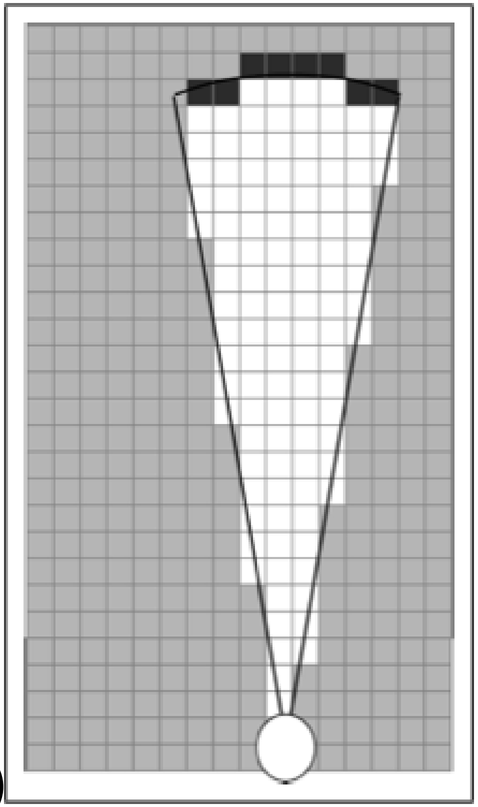
… ist einfach: $m_{\max,t} = \arg\max_m \mathbf{P}(\mathbf{m} | z_{1:t}, \boxed{x_{1:t}})$

Zum Beispiel **Rasterkarten**:

- Initialisiere Felder mit a-priori-Belegtheitsw'keit,
- fahre durch die Umgebung,
- miss in der Gegend herum,
- nach jeder Messung aktualisiere Belegtheitsw'keit der angemessenen Rasterzellen!
- Aktualisierung geht folgendermaßen …

„Inverses Sensormodell" in Rasterkarten

- Lokalisierung nutzt Sensorinformation, dass in gemessener Entfernung „etwas ist"
  ↪ vgl. erwartete Messung n. Sensormodell!
- Kartierung nutzt zusätzlich Information, dass vor gemessenem Wert „nichts ist"
  ↪ wie muss Welt aussehen, um Messung zu liefern?: **Inverses Sensormodell**



- Für alle Rasterfelder $m_{x,z}$ im Messbereich:
  - Felder in gemessener Entfernung erhöhen **Belegtheitschance** (*odds* (occupancy))
  - alle anderen überstrichenen Felder reduzieren *odds*

$$odds_{occ}(m_{x,z},t) = \frac{hits(m_{x,z},t)}{nonhits(m_{x,z},t)} \hat{=} \frac{P(m_{x,z}|o_{1:t})}{1-P(m_{x,z}|o_{1:t})} \in [0,\infty)$$

Belegtheit von $m_{x,z}$ ist boolesche ZV!

odds mit probabilistischem Sensormodell

- Bei probabilistischem inversem Sensormodell $P(m_{x,z}|o, p)$ für Zustand (Pose) p, Messung o, Kartenzelle $m_{x,z}$:

$$odds_{occ}(m_{x,z}, t+1) = odds_{occ}(m_{x,z}, t) \cdot \frac{P(m_{x,z}|o_{t+1}, p_{t+1})}{1 - P(m_{x,z}|o_{t+1}, p_{t+1})}$$

Gefahr von Multiplikationen $0 \cdot \infty$!
Daher verwende meist log odds

$$lodds(\ldots) = lodds(\ldots) + \log P(\ldots) - \log(1 - P(\ldots))$$

6.3 Inkrementelles SLAM

Grundidee: $\mathbf{P}\left(m, x_t \middle| z_{1:t}, u_{1:t-1}\right)$

Tracke die Schätzung der aktuellen Pose (und nur die), wobei folgende Abhängigkeiten bestehen:

- neue Karteneinträge hängen von der aktuellen Poseschätzung ab („Wo ich eine gerade gesehene Landmarke in der Karte eintrage, hängt von der Pose ab")
- die Poseschätzung hängt (unter anderem) ab von Lokalisierung an aktuell wahrgenommenen Landmarken, die bereits in der Karte sind

↪ Wiederverwendung von Lokalisierungs-Methoden

↪ hier: EKF und Partikelfilter/MCL

EKF-SLAM: Die Idee

- Geh vor wie bei EKF-Lokalisierung mit bekannter Karte, **aber**:
- Erweitere Zustandsraum von Pose ($\boldsymbol{x}=(t_x,t_z,\theta)$) auf Pose plus Koordinaten (hier: x, z) aller Landmarken
- Setze voraus, Landmarken sind objektiv eindeutig und unterscheidbar (kein „*aliasing*")
- Falls Zahl der Landmarken unbekannt, arbeite also mit Zustandsraum variabler Größe (wächst mit Zahl der wahrgenommenen Landmarken)
- Wende EKF an wie gehabt, wobei alle zustandsabhängigen Matrizen „mitwachsen" können
- Interessant dabei ist besonders die Kovarianzmatrix …

Kovarianzmatrix Σ_t für EKF-SLAM

- Seien n Landmarken in Karte eingetragen (je mit x- und z-Wert)
- Zustandsraumgröße aktuell also $3 + 2 \cdot n$

Kovarianzen sind Teil der Karte!

Unsicherheit der Roboterpose

$$\begin{pmatrix}
\sigma_x^2 & \sigma_{x,z} & \sigma_{x,\theta} & \sigma_{x,l_{1,x}} & \sigma_{x,l_{1,z}} & \cdots & \sigma_{x,l_{n,x}} & \sigma_{x,l_{n,z}} \\
\sigma_{x,z} & \sigma_z^2 & \sigma_{z,\theta} & \sigma_{z,l_{1,x}} & \sigma_{z,l_{1,z}} & \cdots & \sigma_{z,l_{n,x}} & \sigma_{z,l_{n,z}} \\
\sigma_{x,\theta} & \sigma_{x,\theta} & \sigma_\theta^2 & \sigma_{\theta,l_{1,x}} & \sigma_{\theta,l_{1,z}} & \cdots & \sigma_{\theta,l_{n,x}} & \sigma_{\theta,l_{n,z}} \\
\sigma_{l_{1,x},x} & \sigma_{l_{1,x},z} & \sigma_{l_{1,x},\theta} & \sigma_{l_{1,x}}^2 & \sigma_{l_{1,x},l_{1,z}} & \cdots & \sigma_{l_{1,x},l_{n,x}} & \sigma_{l_{1,x},l_{n,z}} \\
\sigma_{l_{1,z},x} & \sigma_{l_{1,z},z} & \sigma_{l_{1,z},\theta} & \sigma_{l_{1,z},l_{1,x}} & \sigma_{l_{1,z}}^2 & \cdots & \sigma_{l_{1,z},l_{n,x}} & \sigma_{l_{1,z},l_{n,z}} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\sigma_{l_{n,x},x} & \sigma_{l_{n,x},z} & \sigma_{l_{n,x},\theta} & \sigma_{l_{n,x},l_{1,x}} & \sigma_{l_{n,x},l_{1,z}} & \cdots & \sigma_{l_{n,x}}^2 & \sigma_{l_{n,x},l_{n,z}} \\
\sigma_{l_{n,z},x} & \sigma_{l_{n,z},z} & \sigma_{l_{n,z},\theta} & \sigma_{l_{n,z},l_{1,x}} & \sigma_{l_{n,z},l_{1,z}} & \cdots & \sigma_{l_{n,z},l_{n,x}} & \sigma_{l_{n,z}}^2
\end{pmatrix}$$

Korrelat. Roboterposen/ Landmarkenpositionen

Unsicherheiten der Landmarkenpositionen

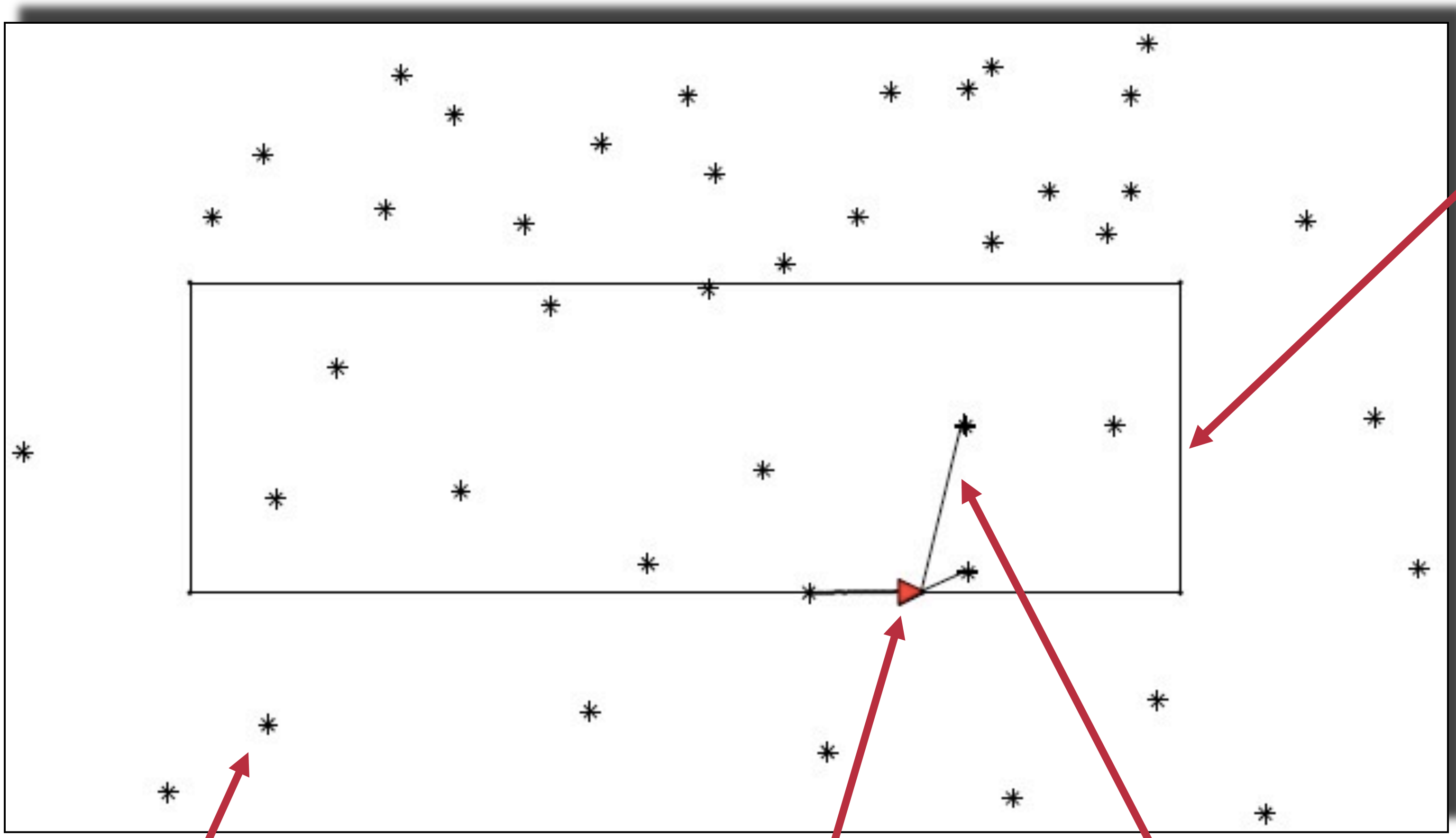
Prinzip des EKF-SLAM-Algorithmus

- Berechne a-priori-Roboterpose-Wert und -Kovarianz aus Bewegungsmodell (Landmarken zunächst unverändert) $\hookrightarrow$ $\underline{\mu}_t, \underline{\Sigma}_t$
- Für erstmalig gesehene Landmarken füge Zustandsraum-dimensionen hinzu; initialisiere ihre (x,z)-Werte über aktuelle Wahrnehmungen (umgerechnet in globale Koordinaten)
- Mittels aller sichtbarer Landmarken aktualisiere $\underline{\mu}_t, \underline{\Sigma}_t$ (berechne Kalman-Gewinnmatrix etc. dafür wie üblich)
- Das Ergebnis am Ende ist μ_t, Σ_t a posteriori: Neuer Wert + Kovarianzen für Roboterpose und Landmarkenpositionen

Bemerkungen

- Vollständige Formulierung des Alg.: Thrun/Fox/Burgard Kap.10.2
- Dort mit fest vorgegebener (Maximal-)Zahl von Landmarken und mit iterativer Berechnung von $\underline{\mu}_t, \underline{\Sigma}_t$ über die aktuell gesehenen Landmarken

Synthetische Beispiel, qualitativ (1)



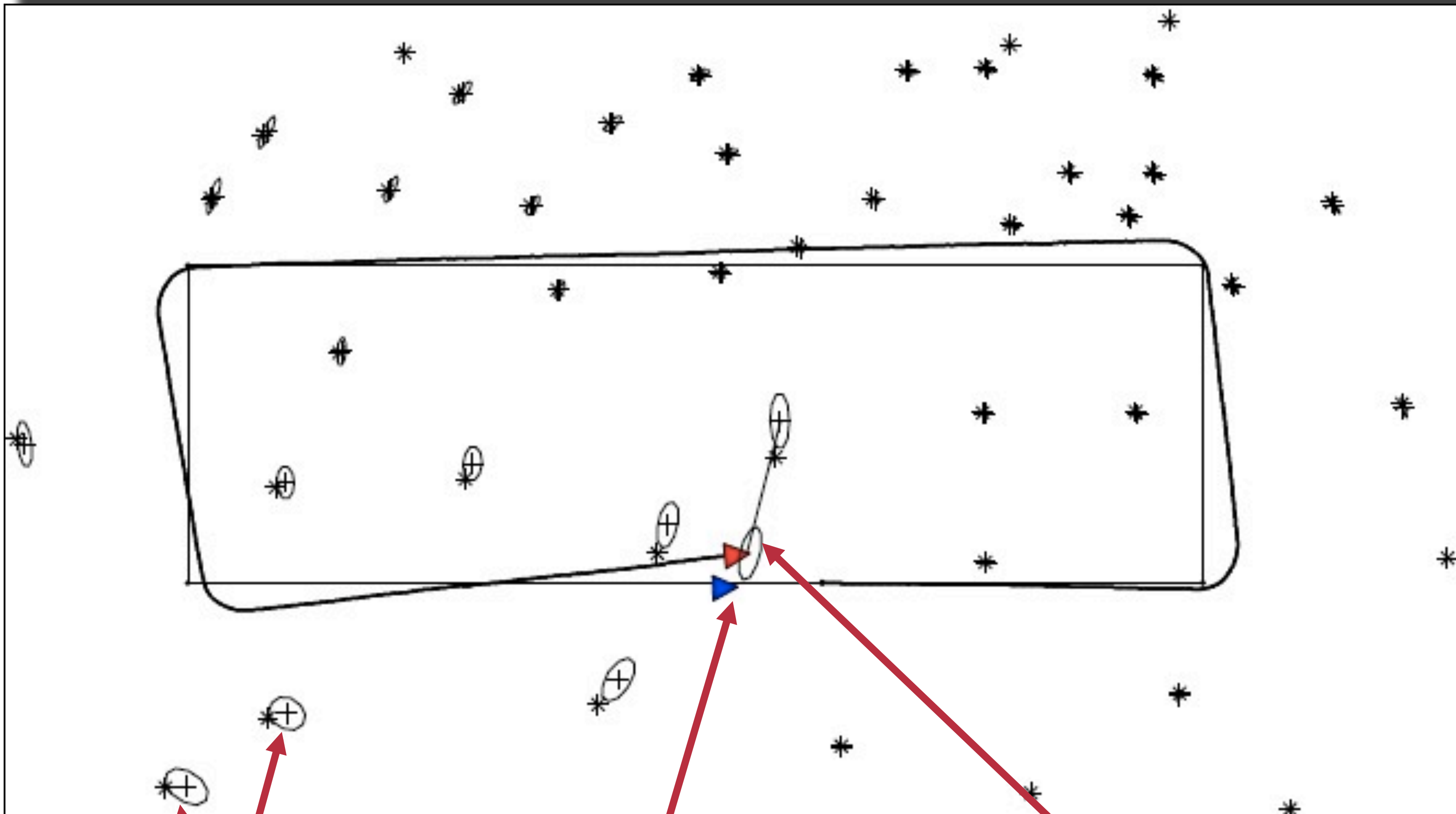
Fahrtroute (geplant bzw. real)

Landmarken

Roboterpose (geschätzt)

Landmarken-Wahrnehmungen

Synthetisches Beispiel, qualitativ (2)

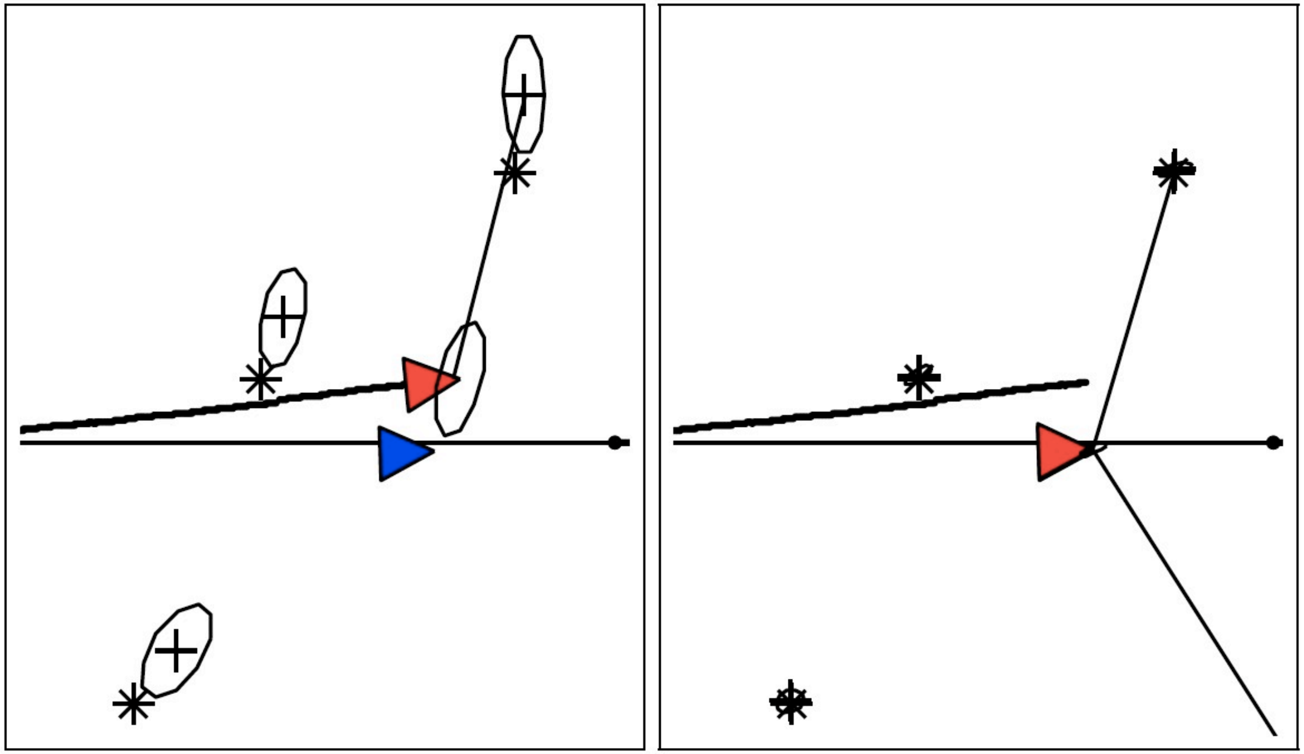


Unsicherheiten

Roboterpose (geschätzt vs. tatsächlich)

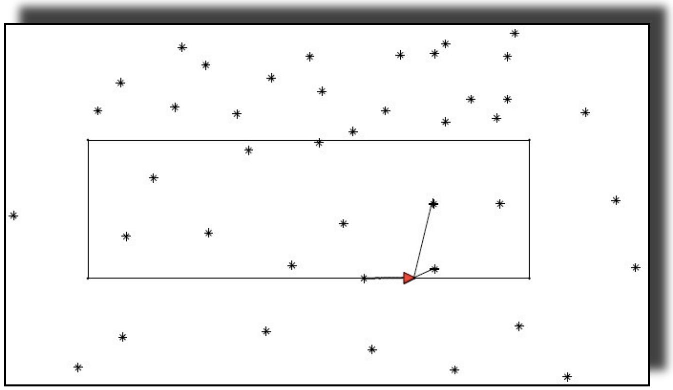
Pose-Kovarianz a priori

Synthetisches Beispiel, qualitativ (3)

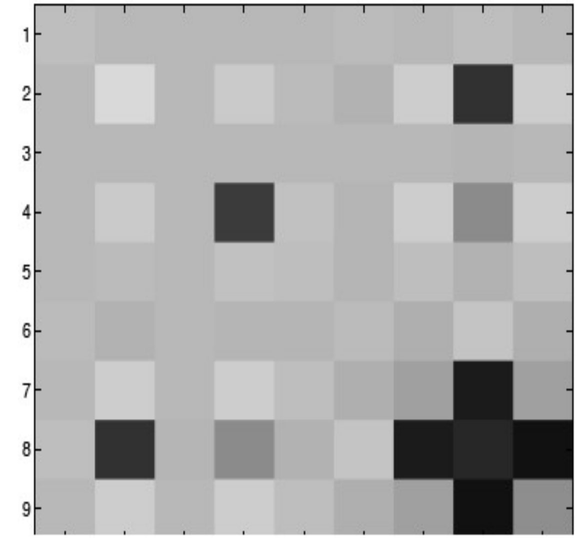


a-priori- vs. a posteriori-Poseschätzungen im Fall des Schleifenschlusses

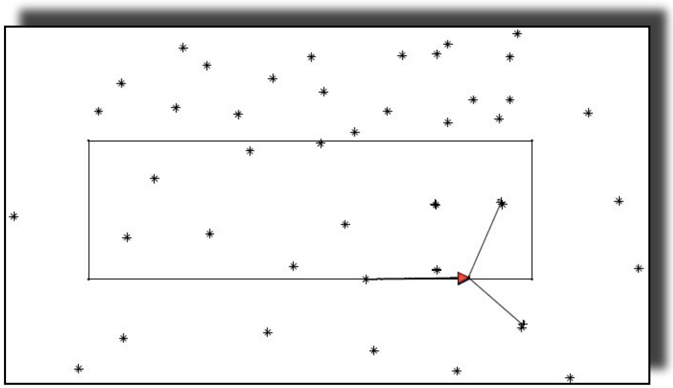
Entwicklung von Σ_t im Beispiel (1)

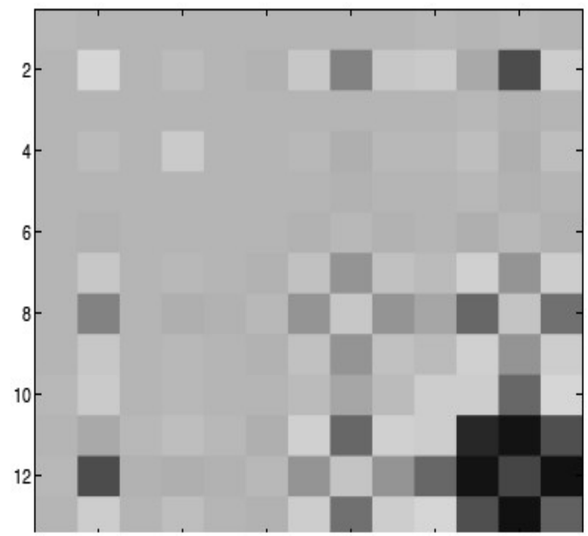


3 Landmarken gesehen



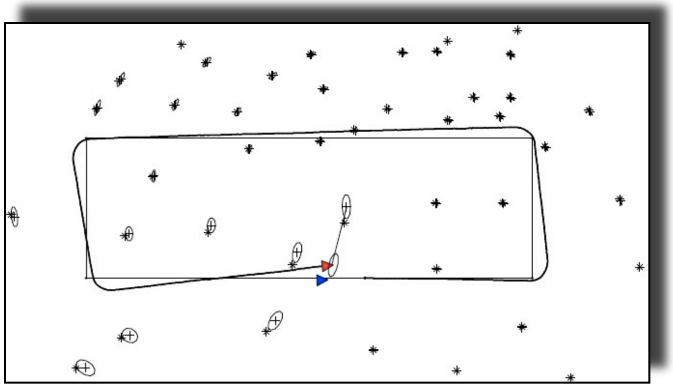
5 Landmarken gesehen

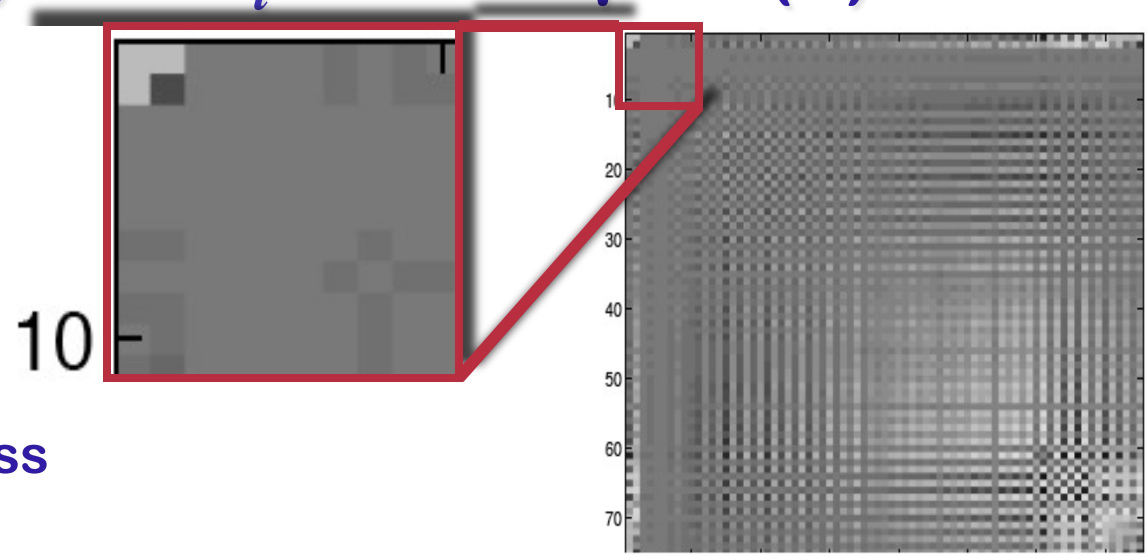




- 3x3 Zellen oben links sind Roboterpose
- heller Eintrag = niedriger Wert = kleine (Ko)Varianz
- Matrix wächst mit jeder neu gesichteten Landmarke

Entwicklung von Σ_t im Beispiel (2)

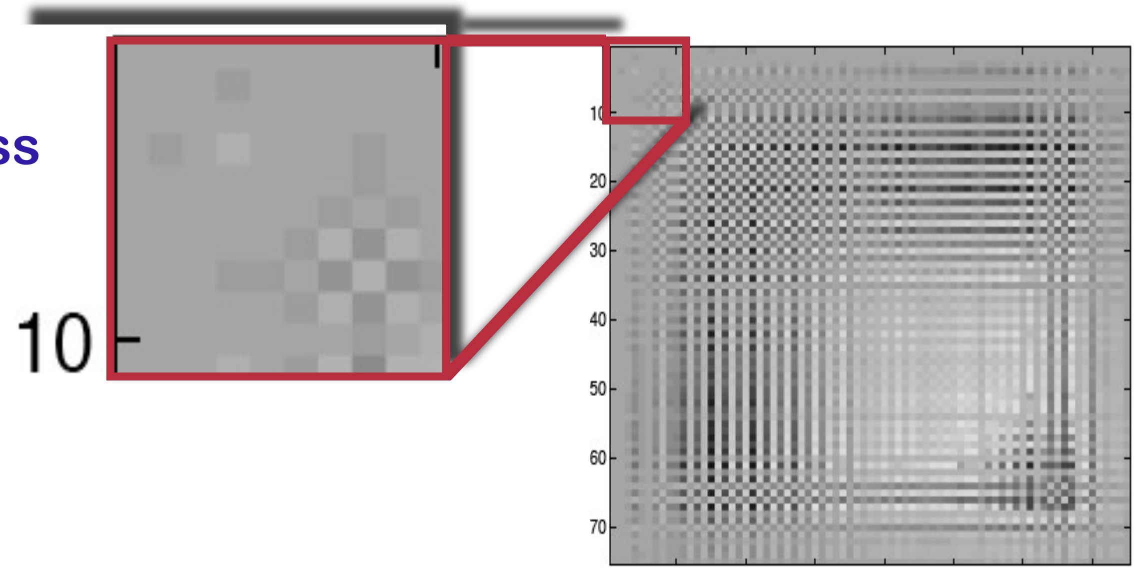


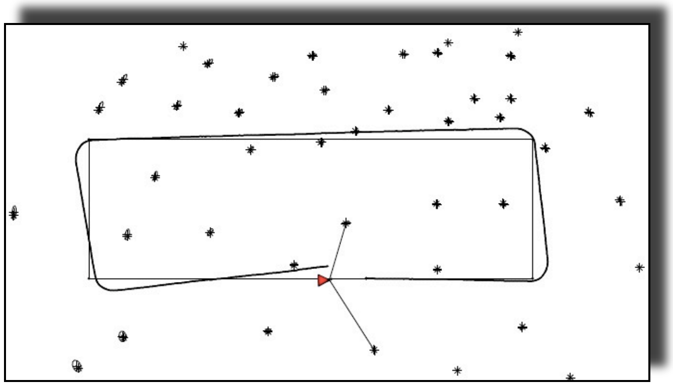


… vor Schleifenschluss

Alle Landmarken gesehen …

… nach Schleifenschluss





Konvergenz von EKF-SLAM

Dissanayake et al.

Satz:

Im Limes der Roboterfahrtlänge konvergiert die Teilmatrix, welche die Kovarianz der Landmarkenpositionen repräsentiert, gegen eine Diagonalmatrix (d.h., ihre Position ist dann genau bekannt).

- Landmarkenpositionen unabhängig voneinander
- Roboterpose bleibt dabei potenziell mit Unsicherheit behaftet!

Nachteil von EKF-SLAM:

Das Verfahren kann nicht die Information verarbeiten, dass an der aktuellen Pose <u>nichts</u> gesehen wird!