5.3.3 Lokalisierung an visuellen Merkmalen

Idee:

- Basiere Lokalisierung auf visuellen statt geometrischen Umgebungsmerkmalen („Kamera statt Laserscanner“)
- Rechne Bewegung im Raum zurück aus der Rückprojektion korrespondierender Bildpunkte (wie in Stereobildern)

Vorteil:

- Bilder sind „flächig“ ↪ Lokalisierung im 3D-Raum
- Für Menschen „natürlicher“ ↪ leichter interpretierbar

Problem:

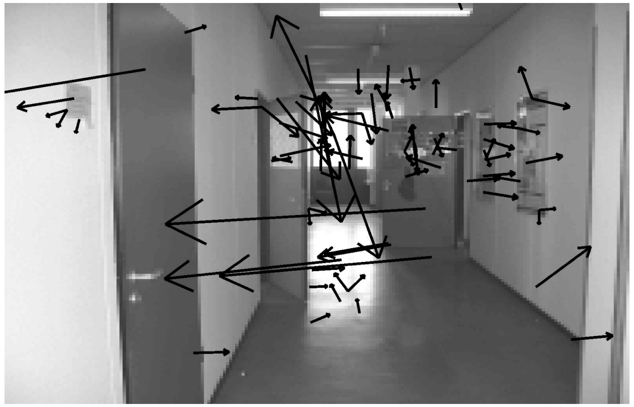
- Korrespondenz von Bildpunkten in Echtzeit ermitteln
- Varianz durch Beleuchtung und Bewegung (Größenänderung im Bild, optische Verzerrungen)

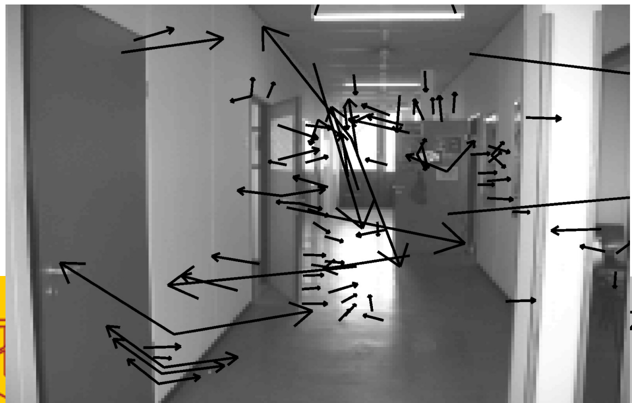
↪ SIFT-Merkmale (s. Kap. 3.2 – oder Harris, SURF, Censure, …)

SIFT-Merkmale zur Roboter-Lokalisierung

Absolute L.: Karte enthält SIFT-Merkmale an ihren „wahren“ Orten. Berechne Pose durch Triangulation (in x,y,z!) mit korrespondierenden SIFT-Merkmalen aus aktuellem Bild

Relative L.: Rechne Poseänderung zurück mit Triangulation (in x,y,z!) korrespondierender SIFT-Merkmale aus letztem Bild





SIFT-Merkmale und Korrespondierende nach Translation 50cm

5.3.4 Unimodale Probabilististische Lokalis.

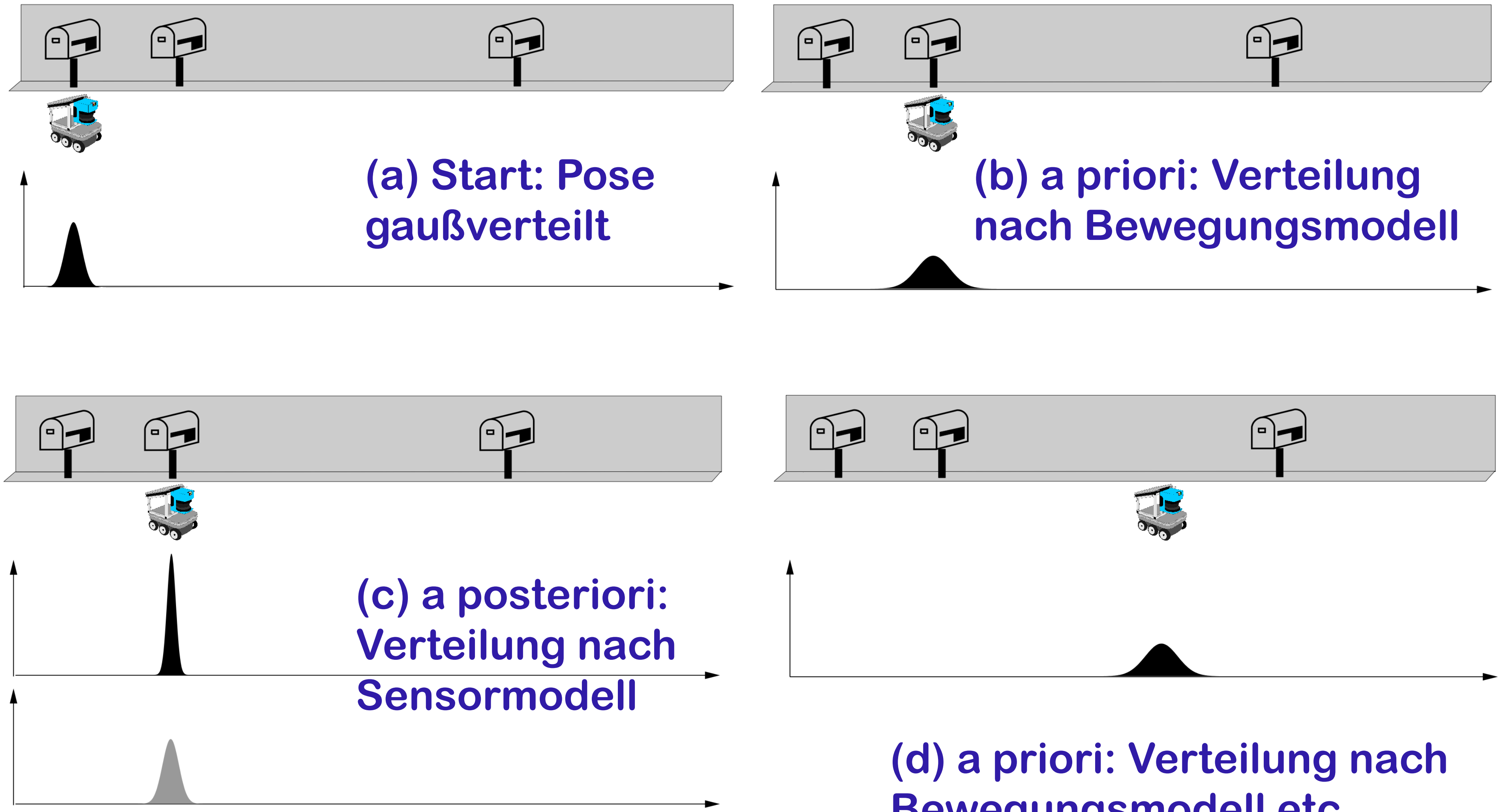
Ziel: Fusioniere 3D-Poseschätzung gemäß Vorwärtskinematik mit Korrekturwert aus Laserscan-Registrierung nach ICP

- Kartenlokalisierung verwendet echten Scan an aktueller Pose und simulierten Scan nach Karte an geschätzter Pose
- Derselbe Ansatz funktioniert ohne Karte mit zwei konsekutiven echten Scans

Ansatz: Modelliere mit EKF (s. Kalman-Folien):

- Zustandsvorhersage ($\underline{\mathbf{x}}_{t+1}$) ergibt sich aus Poseschätzung
- Messwert z ist Registrierungs-Korrekturvektor (umgerechnet ins Roboter-Bezugssystem)

Unimodale (Kalman-) Lokalisierung: Schema



(a) Start: Pose gaußverteilt

(b) a priori: Verteilung nach Bewegungsmodell

(c) a posteriori: Verteilung nach Sensormodell

(d) a priori: Verteilung nach Bewegungsmodell etc.

Erinnerung: KF, hier in EKF-Formulierung

s. Folie Kalman 19/20 (Algorithmus 4.3 im Buch):

$\mathbf{F}_t$ ersetzt $\mathbf{A}$ und $\mathbf{B}$; $\mathbf{H}$ wird zeitabhängig

Eingabe: Akt. Zustand $\mathbf{x}_t$, akt. Fehlermodell $\mathbf{\Sigma}_t$, $e \in \{u,z\}$ an $t+1$
Ausgabe: Aktualisierte $\mathbf{x}_{t+1}$, $\mathbf{\Sigma}_{t+1}$
globale Variablen: $\mathbf{K}_i$, $\underline{\mathbf{x}}_i$, $\underline{\Sigma}_i$

if e ist Aktion u **then**

$\underline{\mathbf{x}}_{t+1} := f(\mathbf{x}_t, u)$

$\underline{\Sigma}_{t+1} := \mathbf{F}_{t+1}\mathbf{\Sigma}_t\mathbf{F}_{t+1}{}^{\mathrm{T}} + \mathbf{\Sigma}_u$

else if e ist Messung z **then**

$\mathbf{K}_{t+1} := \underline{\mathbf{\Sigma}}_{t+1}\mathbf{H}_{t+1}{}^{\mathrm{T}}(\mathbf{H}_{t+1}\,\underline{\mathbf{\Sigma}}_{t+1}\mathbf{H}_{t+1}{}^{\mathrm{T}} + \mathbf{\Sigma}_z)^{-1}$

$\mathbf{x}_{t+1} := \underline{\mathbf{x}}_{t+1} + \mathbf{K}_{t+1}(z - \mathbf{H}_{t+1}\underline{\mathbf{x}}_{t+1})$

$\mathbf{\Sigma}_{t+1} := (\mathbf{1} - \mathbf{K}_{t+1}\mathbf{H}_{t+1})\underline{\mathbf{\Sigma}}_{t+1}$

return$(\mathbf{x}_{t+1}, \mathbf{\Sigma}_{t+1})$

end if

Transitionsmodell (n.-lin.)

$$\mathbf{F}_{t+1} = \left(F_{t+1}^{[i,j]}\right)_{i,j} = \frac{\partial f^{[i]}}{\partial u^{[j]}}(\mathbf{x}_t, \mathbf{u}_t)$$

Sensormodell (n.-lin.)

$$\mathbf{H}_{t+1} = \left(H_{t+1}^{[i,j]}\right)_{i,j} = \frac{\partial h^{[i]}}{\partial x^{[j]}}(\underline{\mathbf{x}}_{t+1})$$

EKF-Ansatz Lokalisierung (1/5)

Werde Fahr-Aktion $\boldsymbol{u}$ aus Pose $\mathbf{x}_t=(x_t,z_t,\theta_t)^\mathrm{T}$ ausgeführt.
Vorwärtskinematik ergebe $\boldsymbol{u} = (\Delta x_{\mathrm{odo}}, \Delta z_{\mathrm{odo}}, \Delta\theta_{\mathrm{odo}})^\mathrm{T}$.
Zustandsschätzung $\underline{\mathbf{x}}_{t+1}$ gemäß Vorwärtskinematik:

$$
\begin{aligned}
\bar{\boldsymbol{x}}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}) &= \boldsymbol{x}_t + \boldsymbol{R}_{\theta_t}\boldsymbol{u} \\
&= \begin{pmatrix} x_t \\ z_t \\ \theta_t \end{pmatrix} + \begin{pmatrix} \cos\theta_t & \sin\theta_t & 0 \\ -\sin\theta_t & \cos\theta_t & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \Delta x_{\mathrm{odo}} \\ \Delta z_{\mathrm{odo}} \\ \Delta\theta_{\mathrm{odo}} \end{pmatrix} \\
&= \begin{pmatrix} x_t + \Delta x_{\mathrm{odo}}\cos\theta_t + \Delta z_{\mathrm{odo}}\sin\theta_t \\ z_t - \Delta x_{\mathrm{odo}}\sin\theta_t + \Delta z_{\mathrm{odo}}\cos\theta_t \\ \theta_t + \Delta\theta_{\mathrm{odo}} \end{pmatrix} .
\end{aligned}
$$

EKF-Ansatz Lokalisierung (2/5)

Scanregistrierung ergebe Posedifferenz/Messung

$$z' = (\Delta x_{\mathrm{ICP}}, \Delta z_{\mathrm{ICP}}, \Delta\theta_{\mathrm{ICP}})^T .$$

Lokale Messung muss in Referenzsystem umgerechnet werden (analog Vorwärtskinematik-Wert):

$$\boldsymbol{z} = \boldsymbol{x}_t + \boldsymbol{R}_{\theta_t}\boldsymbol{z}'$$

$$= \begin{pmatrix} x_t \\ z_t \\ \theta_t \end{pmatrix} + \begin{pmatrix} \cos\theta_t & \sin\theta_t & 0 \\ -\sin\theta_t & \cos\theta_t & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \Delta x_{\mathrm{ICP}} \\ \Delta z_{\mathrm{ICP}} \\ \Delta\theta_{\mathrm{ICP}} \end{pmatrix}$$

$$= \begin{pmatrix} x_t + \Delta x_{\mathrm{ICP}}\cos\theta_t + \Delta z_{\mathrm{ICP}}\sin\theta_t \\ z_t - \Delta x_{\mathrm{ICP}}\sin\theta_t + \Delta z_{\mathrm{ICP}}\cos\theta_t \\ \theta_t + \Delta\theta_{\mathrm{ICP}} \end{pmatrix} .$$

Dies ist Vergleich alter/neuer Scan! Bei Korrektur nach simul. Scan aus Karte geh aus von Pose $\underline{\mathbf{x}}_{t+1}$ statt $\mathbf{x}_t$!

EKF-Ansatz Lokalisierung (3/5)

Funktion h dient beim EKF dazu, Messungen in Zustandsraum zu transformieren.

Mess-Raum ist nach Umrechnungen ins globale Referenzsystem identisch mit Zustandsraum, daher ist h Identitätsfunktion:

$$h(\bar{\boldsymbol{x}}_{t+1}) = \begin{pmatrix} \bar{x}_{t+1} \\ \bar{z}_{t+1} \\ \bar{\theta}_{t+1} \end{pmatrix}$$

EKF-Ansatz Lokalisierung (4/5)

$$f(\underline{\boldsymbol{x}}_t,\boldsymbol{u}) = \begin{pmatrix} x_t + \Delta x_{\text{odo}} \cos\theta_t + \Delta z_{\text{odo}} \sin\theta_t \\ z_t + \Delta x_{\text{odo}} \sin\theta_t - \Delta z_{\text{odo}} \cos\theta_t \\ \theta_t + \Delta\theta_{\text{odo}} \end{pmatrix} \qquad h(\underline{\boldsymbol{x}}_{t+1}) = \begin{pmatrix} \bar{x}_{t+1} \\ \bar{z}_{t+1} \\ \bar{\theta}_{t+1} \end{pmatrix}$$

$$\boldsymbol{F}_{t+1} = \left(\frac{\partial f^{[i]}}{\partial x^{[j]}}(\boldsymbol{x}_t, \boldsymbol{u}_t)\right)_{i,j} = \begin{pmatrix} 1 & 0 & -\Delta x_{\text{odo}} \sin\theta_t + \Delta z_{\text{odo}} \cos\theta_t \\ 0 & 1 & -\Delta x_{\text{odo}} \cos\theta_t - \Delta z_{\text{odo}} \sin\theta_t \\ 0 & 0 & 1 \end{pmatrix}$$

$$\boldsymbol{H}_{t+1} = \left(\frac{\partial h^{[i]}}{\partial \bar{x}^{[j]}}(\bar{\boldsymbol{x}}_{t+1})\right)_{i,j} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} .$$

EKF-Ansatz Lokalisierung (5/5)

Kovarianzmatrizen für Aktionen und Messungen:

$$\Sigma_u = \begin{pmatrix} \sigma_{u,1}^2 & 0 & 0 \\ 0 & \sigma_{u,2}^2 & 0 \\ 0 & 0 & \sigma_{u,3}^2 \end{pmatrix} \qquad \Sigma_z = \begin{pmatrix} \sigma_{z,1}^2 & 0 & 0 \\ 0 & \sigma_{z,2}^2 & 0 \\ 0 & 0 & \sigma_{z,3}^2 \end{pmatrix}$$

Die σ repräsentieren die Unsicherheit des Kinematik- und des Sensormodells ↪ empirisch ermitteln oder schätzen

- Kinematikmodell okay für Translation, schwach für Rotation; also z.B. $\sigma_{u,1}=0.1$, $\sigma_{u,2}=0.2$, $\sigma_{u,3}=0.8$
- ICP-Registrierung okay bei x,z; besser als Kinematik bei Rotation, also z.B. $\sigma_{z,1}=0.3$, $\sigma_{z,2}=0.3$, $\sigma_{z,3}=0.4$

EKF-Rechenbeispiel

… zum Nachvollziehen gemäß Kalmanfilter (EKF)-Algorithmus.
Hier nur 1 Bewegung, 1 Messung; mehr im Buch, Beispiel 5.3

Sei $\mathbf{x}_0=(0\text{cm}, 0\text{cm}, 0°)^\mathsf{T}$;
Odometrie gemäß Vorwärtskinematik ergebe
$\mathbf{u}=(0.25\text{cm}, 22.21\text{cm}, 0.65°)^\mathsf{T}$.

➥ $\underline{\mathbf{x}}_1 = f(\mathbf{x}_0,\mathbf{u}) = (0.25, 22.21, 0.65)^\mathsf{T}$ (ist gleich Δ_{Odo}, da $\mathbf{x}_0 = (0,0,0)^\mathsf{T}$)

➥ $\overline{\Sigma}_1 = \begin{pmatrix} 0.01 & 0 & 0 \\ 0 & 0.04 & 0 \\ 0 & 0 & 0.64 \end{pmatrix} \quad \boldsymbol{K}_1 = \begin{pmatrix} 0.1 & 0 & 0 \\ 0 & 0.31 & 0 \\ 0 & 0 & 0.8 \end{pmatrix}$

Aus ICP-Registrierung folge $z=(-1.98\text{cm}, 20.23\text{cm}, 1.2°)$

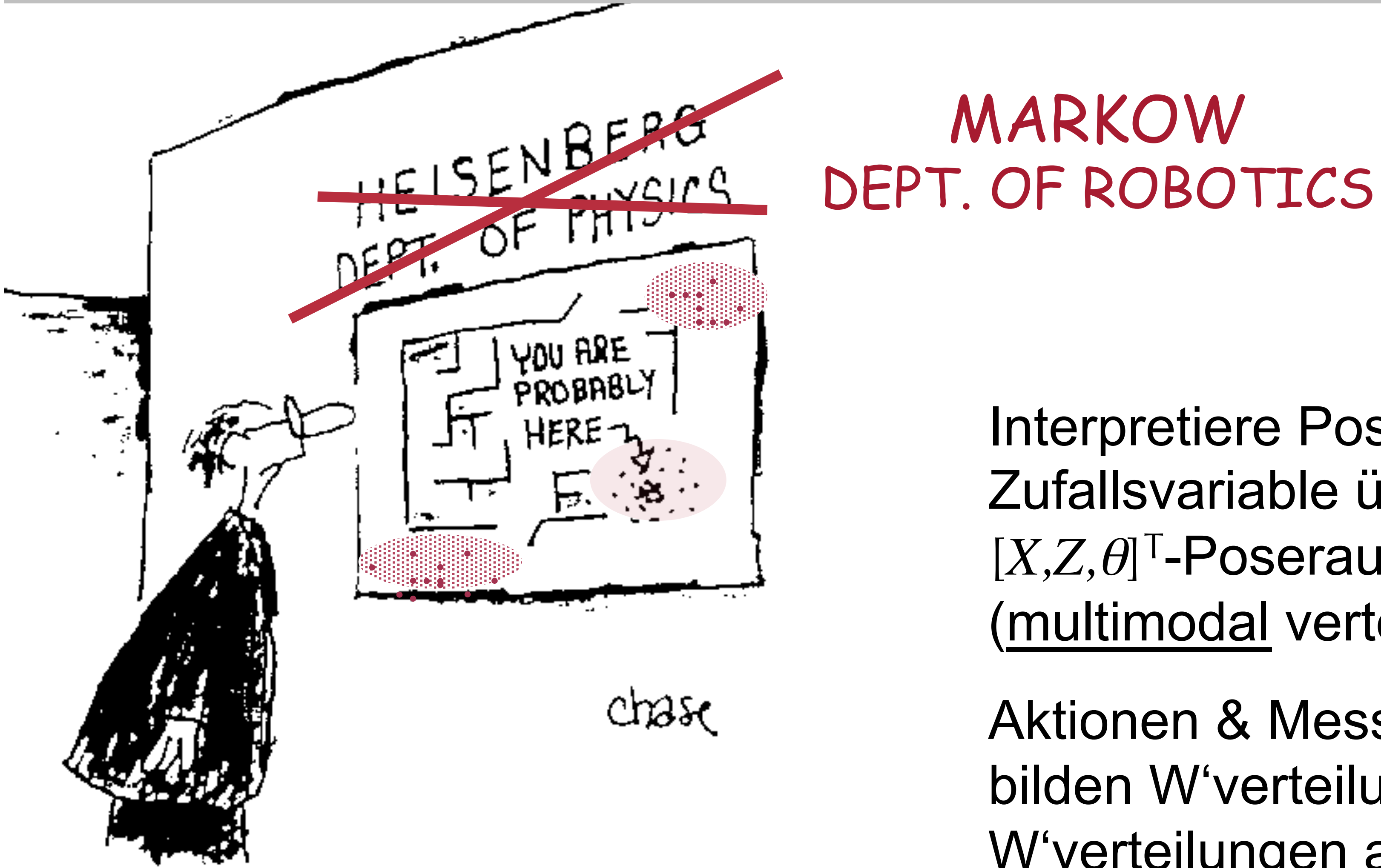
➥ Ergebnis $\mathbf{x}_1=(0.029, 21.61, 1.09)^\mathsf{T}$ mit Unsicherheit

$$\Sigma_1 = \begin{pmatrix} 0.009 & 0 & 0 \\ 0 & 0.028 & 0 \\ 0 & 0 & 0.128 \end{pmatrix}$$

Fazit Unimodale Lokalisierungsverfahren

- Bisher beschriebene Verfahren funktionieren unimodal, d.h. berechnen 1 neue a-posteriori-Pose (bzw. Hypothese)
- Grundlage ist praktisch immer die Poseschätzung gemäß Vorwärtskinematik (ggf. mit Verbesserungen, wie gehabt)
- Die meisten (Ausnahme: HAYAI) können als Modell-Datensatz Daten aus einer Roboterkarte oder aus einer früheren Sensormessung verwenden (absolute oder relative Lokalisierung)
- Unbekannte Startpose ist nicht behandelbar
- Geht die Pose einmal verloren, kann sie praktisch nicht mehr wiedergefunden werden: *Kidnapped Robot Problem*, aber auch (realistisch) Fahrt in merkmalsarmen Regionen

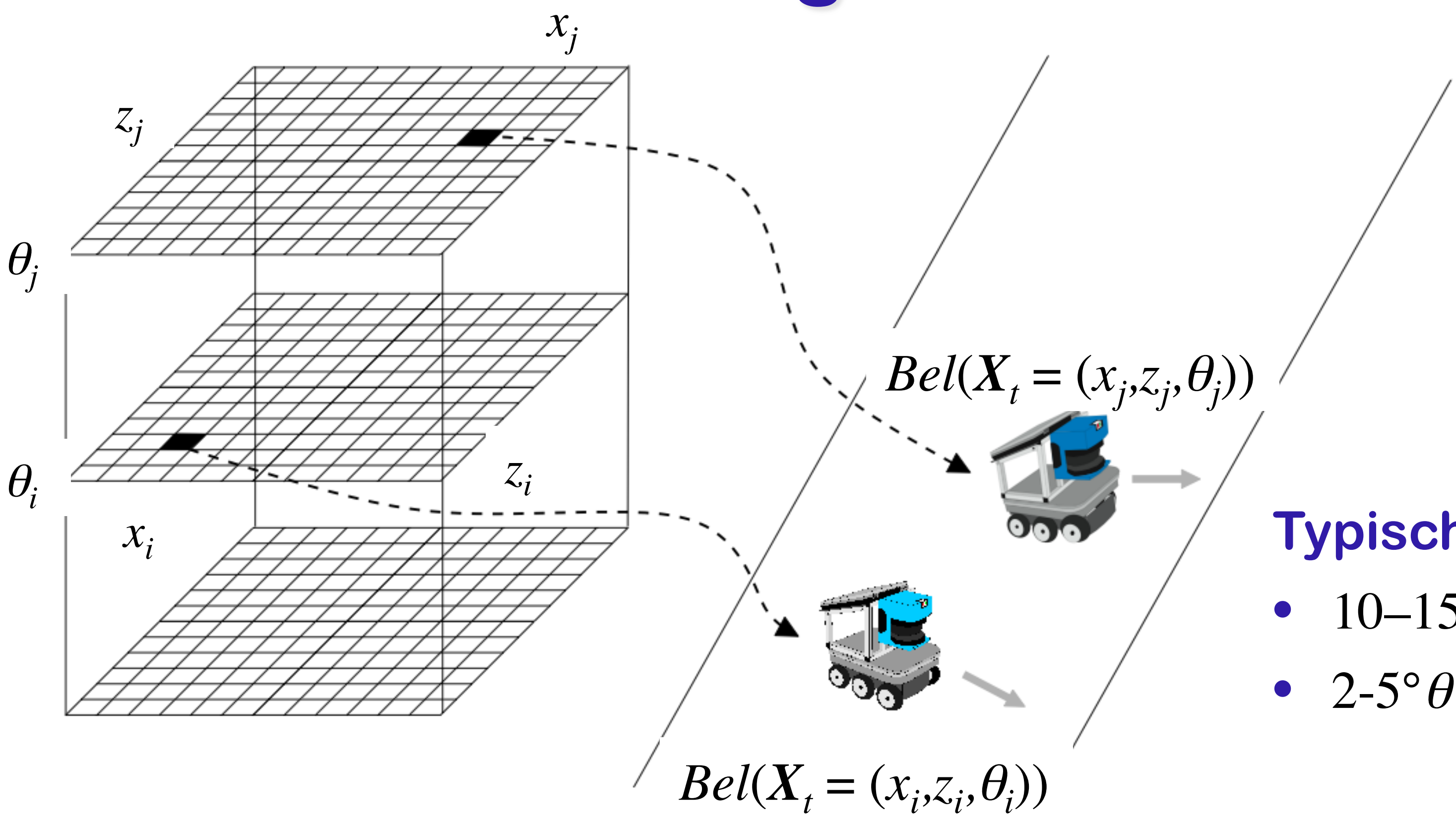
5.3.5 Markow-Lokalisierung



Interpretiere Pose als Zufallsvariable über $[X,Z,\theta]^\mathsf{T}$-Poseraum (multimodal verteilt);

Aktionen & Messungen bilden W'verteilungen in W'verteilungen ab

Diskretisierung der Poseverteilung



Typische Auflösungen

- 10–15cm x bzw. z
- 2-5° θ

- Geh aus von Raster-Belegungskarte (hier uniformes Raster)
- Jede Rasterzelle liegt in Kopien diskreter Orientierung vor
- Pose <u>ist</u> W'verteilung $\mathbf{P}(\mathbf{X} = (x, z, \theta))$

Bayes-Filter zur Lokalisierung

Markow-Lokalisierung ist die (naive) Anwendung des multimodalen Bayes-Filters aufs Lokalisierungsproblem

Erinnerung: Allgemeiner Bayes-Filter (Folie 162)

$$\mathbf{P}\left(\mathbf{X}_{t+1} \middle| \mathbf{e}_{1:t+1}\right) = \eta \cdot \mathbf{P}\left(\mathbf{e}_{t+1} \middle| \mathbf{X}_{t+1}\right) \cdot \int_{\mathbf{x}_t} \left[\mathbf{P}\left(\mathbf{X}_{t+1} \middle| \mathbf{x}_t\right) \cdot P\left(\mathbf{x}_t \middle| \mathbf{e}_{1:t}\right)\right]$$

Erinnerung: Spezieller Bayes-Filter (Handeln/Messen) (F. 164)

$$Bel\left(\mathbf{X}_{t+1}\right) = \eta \cdot \mathbf{P}\left(\mathbf{Z}_{t+1} \middle| \mathbf{X}_{t+1}\right) \sum_{x_t} \mathbf{P}\left(\mathbf{X}_{t+1} \middle| \mathbf{X}_t, \mathbf{u}_{t+1}\right) \cdot Bel\left(\mathbf{X}_t\right)$$

Sensormodell,
a-posteriori-Pose

Aktionsmodell,
a-priori-Pose

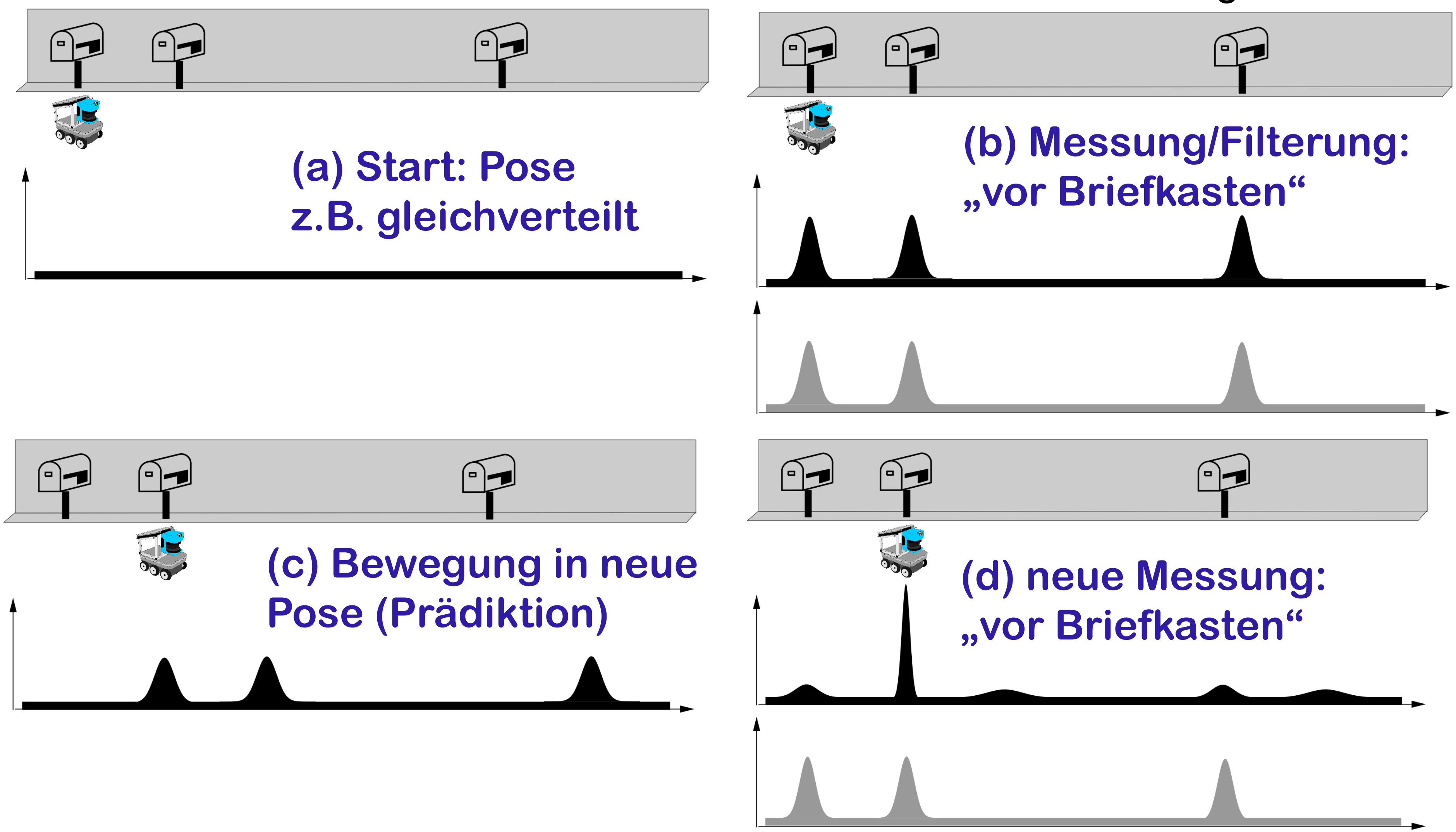
Spezieller Bayes-Filter: Idee

Voraussetzung: Roboterhandeln wird beschrieben als Folge von **abwechselnd** Aktion und Zustandsmessung. Gegeben Start-Zustandsschätzung (Poseverteilung)

1. Führe Aktion aus
2. **Prädiktion**: Schätze resultierenden Zustand für Aktion (**a-priori-Zustandsschätzung**)
$$\overline{Bel}(\mathbf{X}_{t+1}) = \sum_{\mathbf{X}_t} P(\mathbf{X}_{t+1} | \mathbf{u}_{t+1}, \mathbf{X}_t) \cdot Bel(\mathbf{X}_t)$$
3. Miss resultierenden Zustand
4. **Filterung**: berechne aktuelle Zustandsverteilung, die maximal gut zu allen Messungen und der Vorhersage passt (**a-posteriori-Zustandsschätzung**); weiter bei 1.
$$Bel(\mathbf{X}_{t+1}) = \eta P(\mathbf{Z}_{t+1} | \mathbf{X}_{t+1}) \cdot \overline{Bel}(\mathbf{X}_{t+1})$$

1D-Beispiel, quasi-kontinuierlich

vgl. Folie 247!



Naive Markow-Lokalisierung ist naiv!

- Umgebungen interessanter Größe ergeben Verteilungen von nicht handhabbarer Größe!
  ($20\mathrm{m} \times 50\mathrm{m}$ bei 10 cm und $3{,}6°$ Auflösung ➥ 10^8 Posen!)
- Auflösung verringern verwäscht die Lokalisierung

➥ So geht es nur für kleine Umgebungen!

Warum soll man eigentlich unsichere Information bis ins letzte Detail präzise durchrechnen?