

# Modellierung der Roboterwahrnehmung

Raffael Dzikowski

Institut für Informatik  
Humboldt-Universität zu Berlin  
dzikowsk@informatik.hu-berlin.de

13. November 2006

# Gliederung

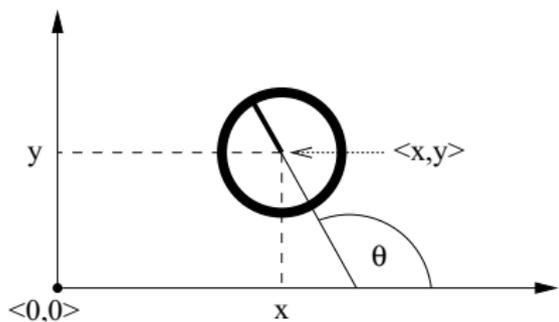
- 1 Einführung
- 2 Das Strahlenmodell für Entfernungssensoren
- 3 Wahrscheinlichkeitsfelder
- 4 Korrelationsbasierte Modelle
- 5 Quellenangaben

# Roboterwahrnehmung - Was ist das?

*Perception is the process by which the robot uses its **sensors** to obtain **information** about the state of its environment.*

*Probabilistic Robotics, p. 22*

# Notation: Pose des Roboters



Pose zum Zeitpunkt  $t$

$$x_t = (x, y, \theta)$$

**Achtung**

Normalerweise symbolisiert  $x_t$  den Zustand.

# Notation: Messungen

Messung zum Zeitpunkt  $t$

$$z_t = \{z_t^1, \dots, z_t^K\}$$

Merke

Eine Messung besteht aus mehreren Messwerten.

# Sensorentypen

- Kameras
- Tastsensoren
- Entfernungssensoren

# Entfernungssensoren

## Funktionsweise

Senden ein Signal aus und registrieren das Echo.

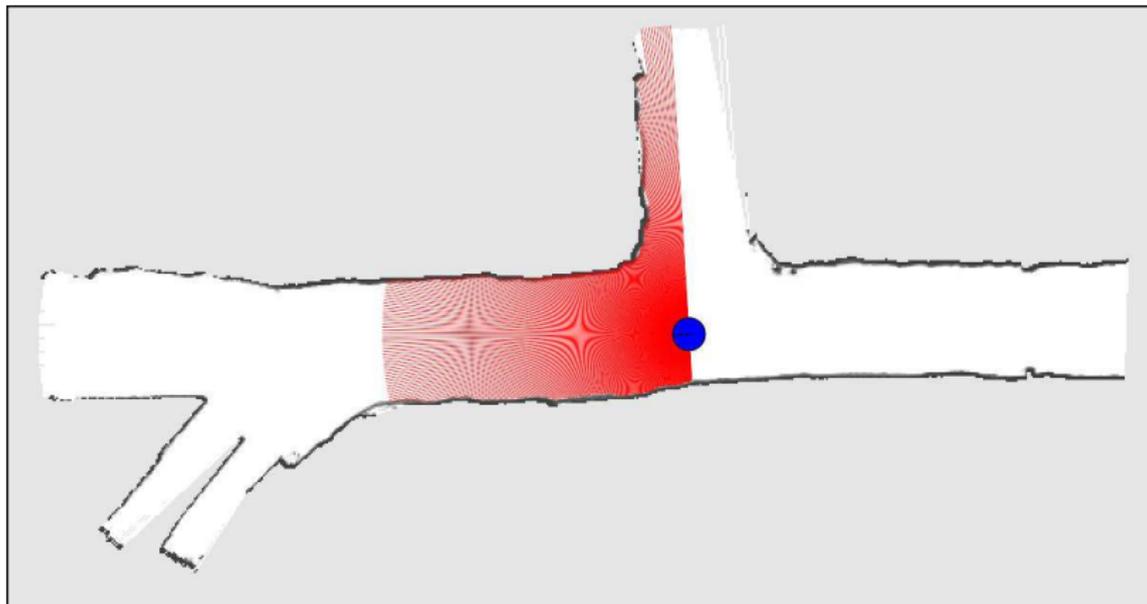
## Lasersensor

Signal: Gebündelter Lichtstrahl

## Ultraschallsensor

Signal: Schallwellen

# Ein typischer Laserscan



# Wahrnehmungsmodelle (allgemein)

Worum geht es?

Berechnung der **Wahrscheinlichkeitsverteilung**

$$p(z_t \mid x_t, m)$$

Erinnerung

Notation	Bedeutung
$z_t$	<b>Messung</b> zum Zeitpunkt $t$
$x_t$	<b>Pose</b> des Roboters zum Zeitpunkt $t$
$m$	<b>Karte</b> der Umgebung

# Berechnung der Messwahrscheinlichkeit aus den Messwerten

## Theoretische Berechnung

$$p(z_t | x_t, m) = \prod_{i=1}^K p(z_t^i | x_t, m)$$

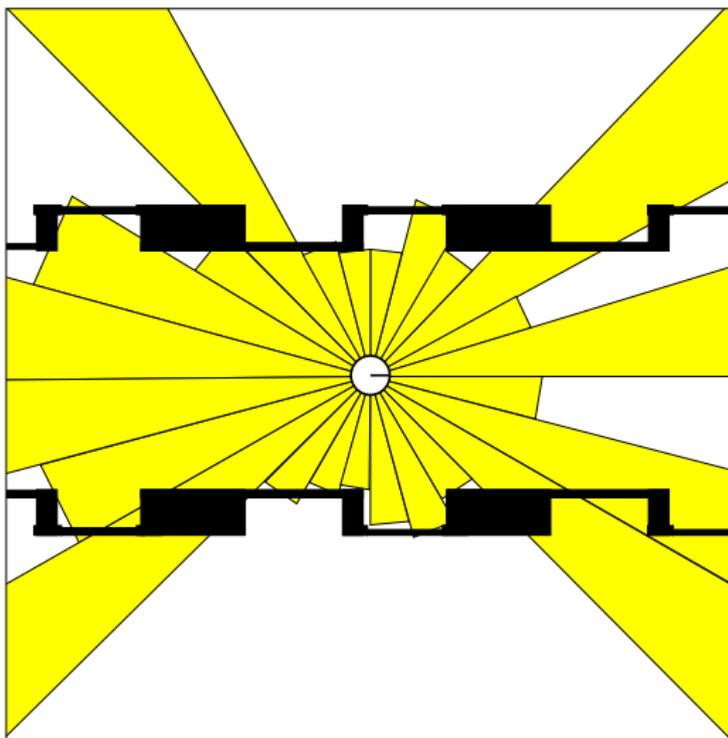
## Frage

Wie werden die  $p(z_t^i | x_t, m)$  berechnet?

## Antwort

Vorgehensweise vom **Modell** abhängig.

# Fehlerquellen bei Ultraschallsensoren



- 1 Einführung
- 2 Das Strahlenmodell für Entfernungssensoren**
- 3 Wahrscheinlichkeitsfelder
- 4 Korrelationsbasierte Modelle
- 5 Quellenangaben

# Motivation für das Strahlenmodell

## Problem

Sensormessungen sind **nie** störungsfrei.

## Konsequenz

Modell muss **Unsicherheit** einbeziehen.

# Überblick über das Strahlenmodell

## Kernpunkte

Sensortyp Entfernungssensoren

## Herangehensweise

- Ausnutzung der physikalischen Eigenschaften
- Modellierung verschiedener Fehlerquellen

# Einbezogene Fehlerquellen

## Fehlerquellen

- 1 Korrekte Entfernung mit kleineren Störungen
- 2 Fehler durch unvorhergesehene Hindernisse
- 3 Versagen bei der Erkennung von Hindernissen
- 4 Unerklärbare Fehler

## Vorgehen

Zuordnung von Wahrscheinlichkeitsverteilungen

# Korrekte Messung mit kleineren Störungen

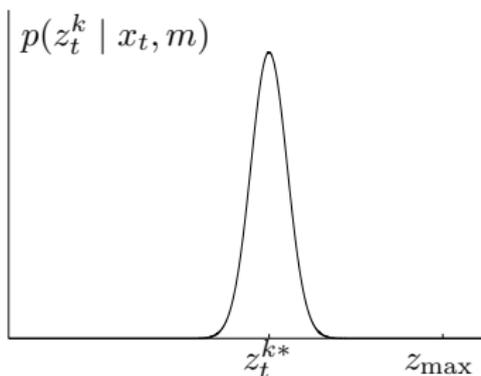
## Notation

- Wahre Entfernung zu einem Hindernis:  $z_t^{k*}$
- Maximale Reichweite:  $z_{\max}$

# Korrekte Messung mit kleineren Störungen

## Graph

(a) Gaussian distribution  $p_{\text{hit}}$



$$p_{\text{hit}}(z_t^k | x_t, m) = \begin{cases} \eta \mathcal{N}(z_t^k; t_t^{k*}, \sigma_{\text{hit}}^2) & \text{falls } 0 \leq z_t^k \leq z_{\text{max}} \\ 0 & \text{sonst} \end{cases}$$

# Fehler durch unvorhergesehene Hindernisse

intuitiv

## Problem

**statische** Karte vs. **dynamische** Roboterumgebung

## Zu klären

Behandlung unmodellierter Objekte

## Ansätze

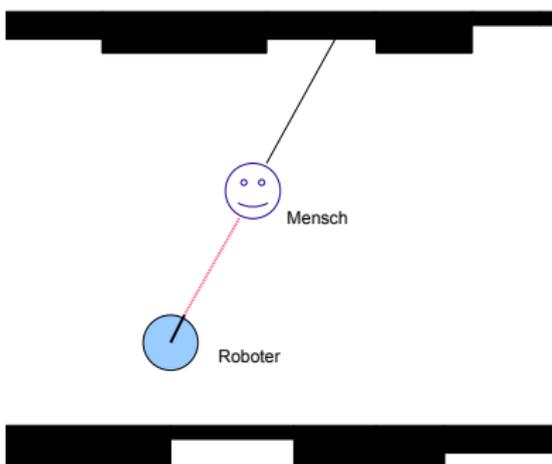
1. Möglichkeit: Teil des Zustandsvektors
2. Möglichkeit: Sensorrauschen

# Behandlung unvorhergesehener Hindernisse als Sensorrauschen

## Vorteile

- nicht rechenintensiv
- robust
- leichter zu implementieren

# Unvorhergesehene Hindernisse



## Beobachtungen

- stets **kürzere** Entfernungen ( $[0, z_t^{k^*}]$ )
- Wahrscheinlichkeit mit wachsender Entfernung abnehmend

# Modellierung unvorhergesehener Hindernisse

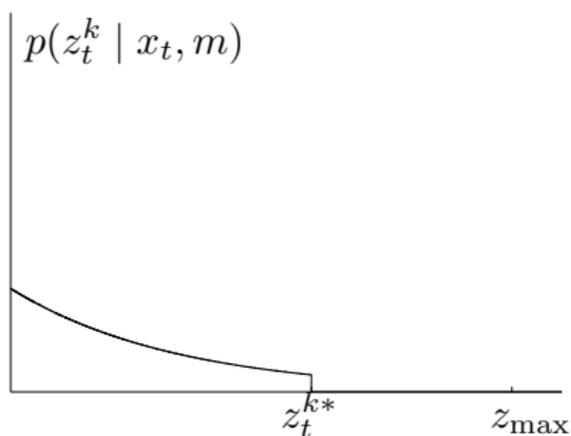
## Konsequenz für Modellierung

- Modellierung durch Exponentialverteilung
- Bezeichnung:  $p_{\text{short}}(z_t \mid x_t, m)$

# Fehler durch unvorhergesehene Hindernisse

## Graph

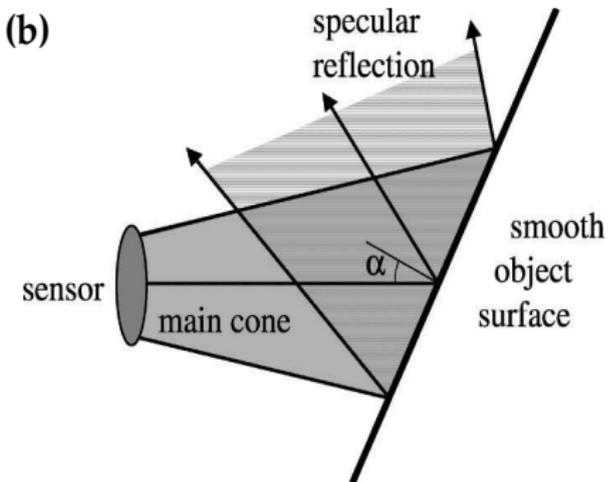
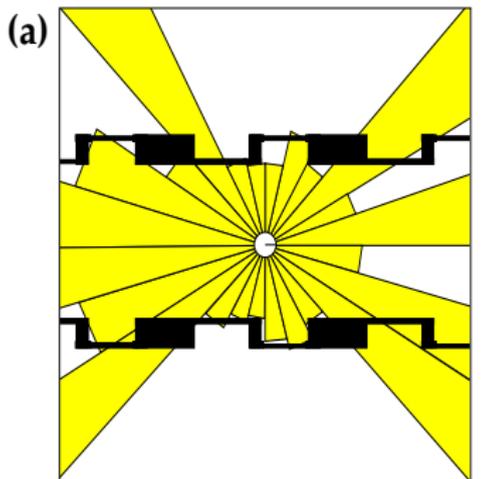
(b) Exponential distribution  $p_{\text{short}}$



$$p_{\text{short}}(z_t^k | x_t, m) = \begin{cases} \eta \lambda_{\text{short}} e^{-\lambda_{\text{short}} z_t^k} & \text{falls } 0 \leq z_t^k \leq z_t^{k*} \\ 0 & \text{sonst} \end{cases}$$

# Versagen bei der Erkennung von Hindernissen

Beispiel: Ultraschallsensoren



# Versagen bei der Erkennung von Hindernissen

## Modellierung

### Problem

Wie erkennen wir diese Fehlerart?

### Lösung

**Fakt:** Sensoren liefern in diesem Fall oft die **maximale Reichweite**

**Deshalb:** Explizite Modellierung dieser Werte als Fehler

# Versagen bei der Erkennung von Hindernissen

## Modell

Modellierung als diskrete Wahrscheinlichkeitsverteilung

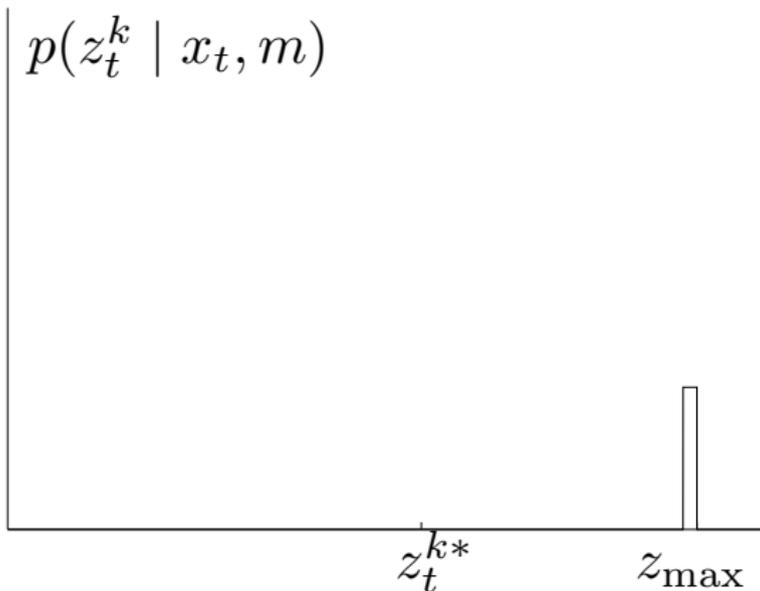
$$p_{\max}(z_t \mid x_t, m)$$

$$p_{\max}(z_t^{k^*} \mid x_t, m) = \begin{cases} 1 & \text{falls } z_t^k = z_{\max} \\ 0 & \text{sonst} \end{cases}$$

# Pseudo-Wahrscheinlichkeitsdichtefunktion

## Graph

(c) Uniform distribution  $p_{\max}$



# Unerklärbare Fehler

## Unerklärbare Fehler

- Fehler, die in keine der genannten Kategorien fallen
- Zufälliges Sensorversagen

# Unerklärbare Fehler Modell

## Problem

Wie erfassen wir diese Fehlerart?

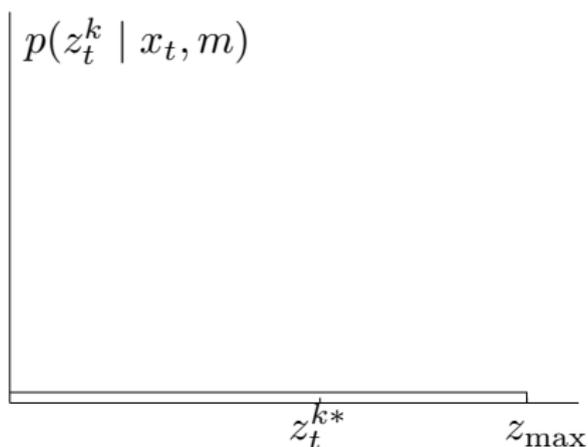
## Lösung

Durch eine **Gleichverteilung** über den gesamten Messbereich

# Unerklärbare Fehler

## Graph

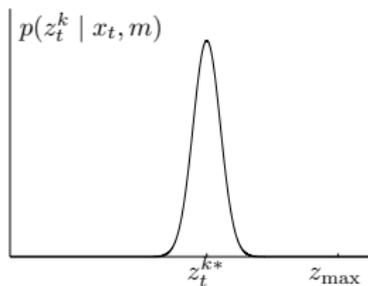
(d) Uniform distribution  $p_{\text{rand}}$



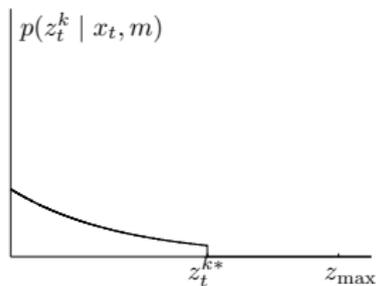
$$p_{\text{rand}}(z_t | x_t, m) = \begin{cases} \frac{1}{z_{\text{max}}} & \text{falls } 0 \leq z_t^k \leq z_{\text{max}} \\ 0 & \text{sonst} \end{cases}$$

# Zusammenstellung Fehlerarten grafisch

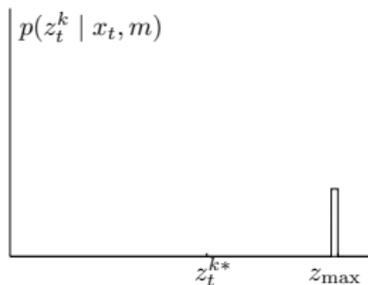
(a) Gaussian distribution  $p_{\text{hit}}$



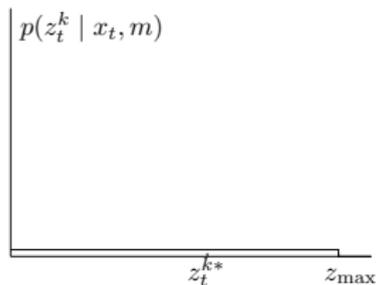
(b) Exponential distribution  $p_{\text{short}}$



(c) Uniform distribution  $p_{\text{max}}$



(d) Uniform distribution  $p_{\text{rand}}$



# Strahlenmodell (Umsetzung)

## Idee

### Grundidee

Messung als **Kombination** der vier  
Wahrscheinlichkeitsdichtefunktionen

### Gewichte

**Werte:**  $z_{\text{hit}}, z_{\text{short}}, z_{\text{max}}, z_{\text{rand}} \in [0, 1]$

**Bedingung:**  $z_{\text{hit}} + z_{\text{short}} + z_{\text{max}} + z_{\text{rand}} = 1$

# Strahlenmodell (Umsetzung)

## Berechnung

### Ziel

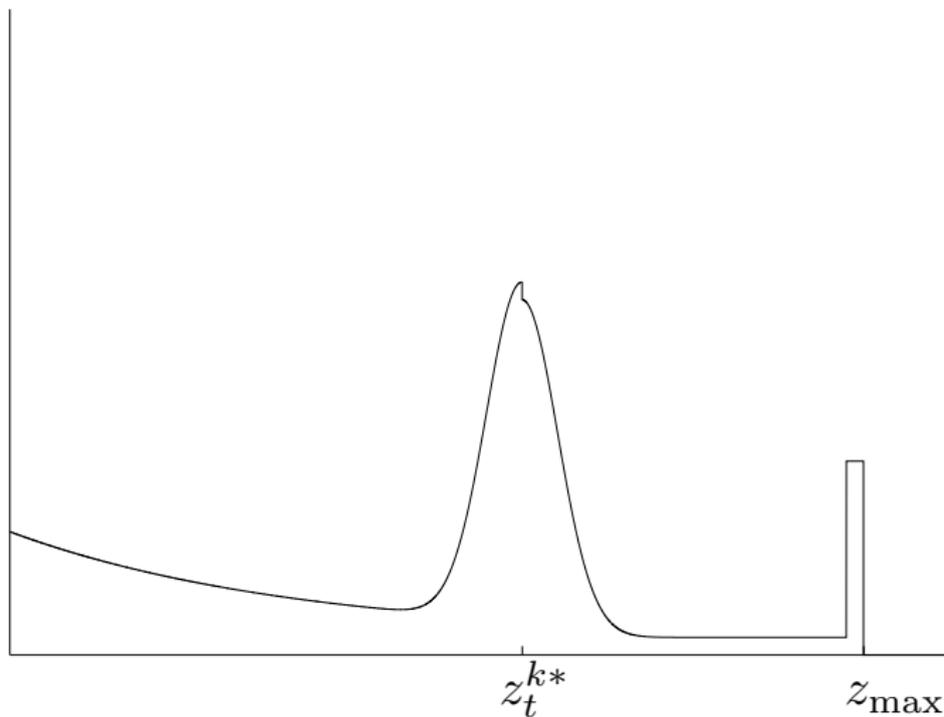
Berechnung der Wahrscheinlichkeitsverteilung  $p(z_t^k | x_t, m)$

### Berechnungsvorschrift

$$p(z_t^k | x_t, m) = \begin{pmatrix} z_{\text{hit}} \\ z_{\text{short}} \\ z_{\text{max}} \\ z_{\text{rand}} \end{pmatrix}^T \cdot \begin{pmatrix} p_{\text{hit}}(z_t^k | x_t, m) \\ p_{\text{short}}(z_t^k | x_t, m) \\ p_{\text{max}}(z_t^k | x_t, m) \\ p_{\text{rand}}(z_t^k | x_t, m) \end{pmatrix}$$

# Strahlenmodell

## Anwendungsbeispiel



# Algorithmus Strahlenmodell

## Messung zum Zeitpunkt $t$

$$z_t = \{z_t^1, \dots, z_t^K\}$$

## Unabhängigkeit

$$p(z_t | x_t, m) = \prod_{i=1}^K p(z_t^i | x_t, m)$$

## Algorithmus

BEAMRANGEFINDERMODEL( $z_t, x_t, m$ )

- 1  $q = 1$
- 2 **for**  $k = 1$  **to**  $K$
- 3     **do** berechne  $z_t^{k*}$  für den Messwert  $z_t^k$  durch *ray casting*
- 4          $p = z_{\text{hit}} \cdot p_{\text{hit}} + z_{\text{short}} \cdot p_{\text{short}} + z_{\text{max}} \cdot p_{\text{max}} + z_{\text{rand}} \cdot p_{\text{rand}}$
- 5          $q = q \cdot p$
- 6 **return**  $q$

# Interne Parameter

## Interne Parameter

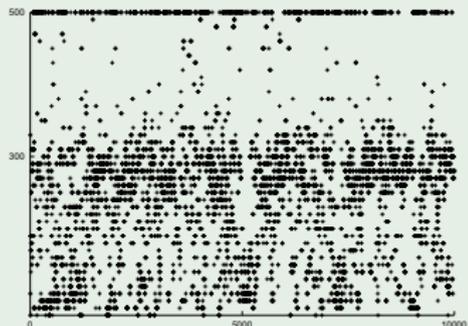
- $z_{\text{hit}}, z_{\text{short}}, z_{\text{max}}, z_{\text{rand}}$
- $\sigma_{\text{hit}}, \lambda_{\text{short}}$

# Interne Parameter

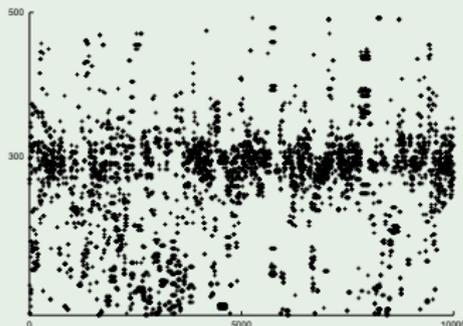
## Bestimmung nach Augenmaß

### Ein typischer Datensatz

(a) Sonar data



(b) Laser data



### Legende

x-Achse: 10.000 Messungen; y-Achse: gemessene Entfernung in *cm*  
( $z_t^{k*} = 300\text{cm}$ ,  $z_{\max} = 500\text{cm}$ )

# Einschätzung des Strahlenmodells

## Pro

- realistisches Modell

## Kontra

- rechenintensiv
- anfällig bei Positionsänderungen

- 1 Einführung
- 2 Das Strahlenmodell für Entfernungssensoren
- 3 Wahrscheinlichkeitsfelder**
- 4 Korrelationsbasierte Modelle
- 5 Quellenangaben

# Wahrscheinlichkeitsfelder

## Idee

**Projektion** des Endpunktes jedes Sensorstrahles in die Koordinatenebene der Karte  $m$ .

# Notation

Pose des Roboters

$$x_t = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}$$

Endpunkt des Sensorstrahls

$$\begin{pmatrix} x_{z_t^k} \\ y_{z_t^k} \end{pmatrix}$$

Anfangspunkt des Sensorstrahles (Sensorposition)

$$\begin{pmatrix} x_{k,sens} \\ y_{k,sens} \end{pmatrix}$$

Richtung des Sensorstrahles

$$\theta_{k,sens}$$

# Wahrscheinlichkeitsfelder (Modell)

zentrale Gleichung

## Projektion in das Koordinatensystem

$$\begin{pmatrix} x_{z_t^k} \\ y_{z_t^k} \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \cdot \begin{pmatrix} x_{k,sens} \\ y_{k,sens} \end{pmatrix} + \begin{pmatrix} \cos(\theta + \theta_{k,sens}) \\ \sin(\theta + \theta_{k,sens}) \end{pmatrix}$$

## Bemerkung

Nur sinnvoll, wenn ein Hindernis entdeckt wird

# Wahrscheinlichkeitsfelder (Modell)

## Modellierte Fehlerquellen

### Drei Fehlerquellen

- 1 Kleine Messstörungen (Neuer Modellierungsansatz)
- 2 Versagen bei der Erkennung von Hindernissen ( $p_{\max}$ )
- 3 Unerkärbare Fehlmessungen ( $p_{\text{rand}}$ )

### Analogie

Jede Messung ist eine Kombination aus diesen Fehlerquellen

# Kleinere Messstörungen

## Bemerkung

Wir arbeiten auf der Ebene der **Karte**

## Ziel

Die Entfernung zum **nächsten** Hindernis in der **Karte** finden

## Lösung

- 1 Berechnung des (euklidischen) Abstandes zwischen dem Endpunkt des Sensorstrahles und dem nächsten Hindernis (*dist*)
- 2 Einbeziehung der Fehlerwahrscheinlichkeit durch Standardnormalverteilung

# Kleine Messstörungen

## Berechnung der Fehlerwahrscheinlichkeit

### Berechnung

$$p_{\text{hit}}(z_t^k \mid x_t, m) = \epsilon_{\sigma_{\text{hit}}}(dist)$$

### Notation

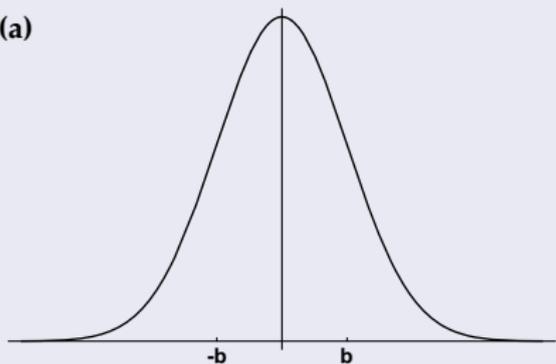
Notation	Bedeutung
$dist$	<b>Abstand</b> zum <b>nächsten</b> Hindernis
$\sigma_{\text{hit}}$	<b>Fehlerschwelle</b> des Sensors (Standardabweichung)
$\epsilon_{\sigma_{\text{hit}}}(dist)$	Standardnormalverteilung ( <b>PDF</b> für $dist$ )

# Kleinere Messstörungen

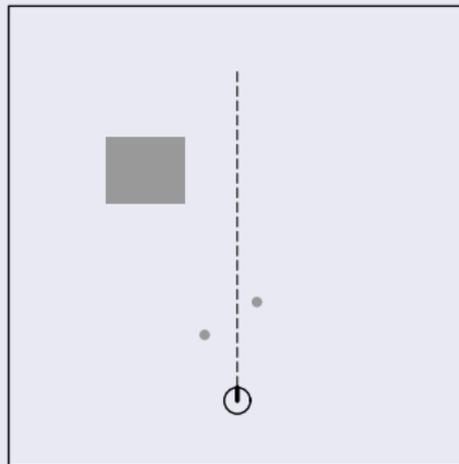
## Visualisierung

### Standardnormalverteilung

(a)



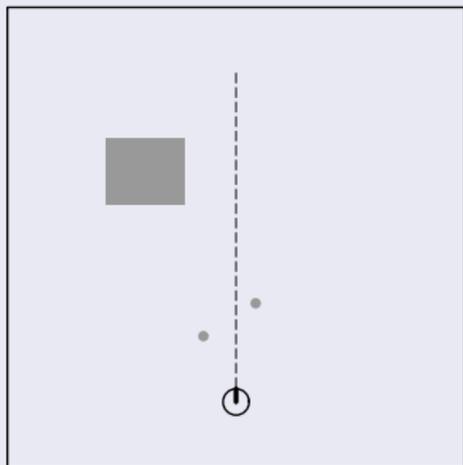
### Karte



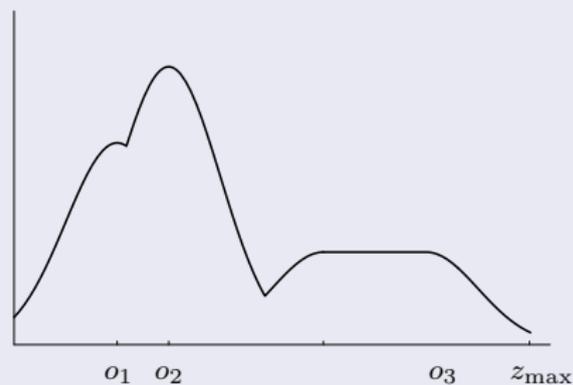
# Beispiel für die Wahrscheinlichkeitsverteilung $p_{\text{hit}}$

## Graph

Karte

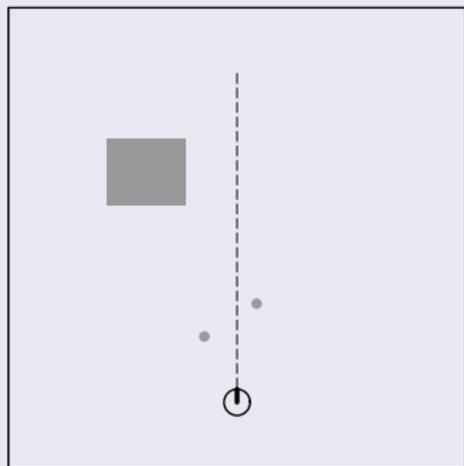
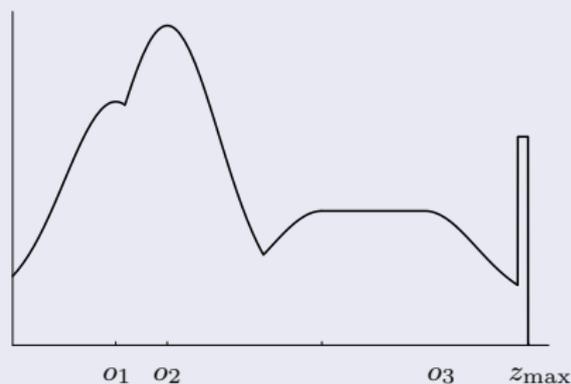
Wahrscheinlichkeitsverteilung  $p_{\text{hit}}$ 

(a)  $p_{\text{hit}}(z_t^k | x_t, m)$



# Einbeziehung aller drei Fehlerquellen

Karte


 $p(z_t^k | x_t, m)$ 
(b)  $p(z_t^k | x_t, m)$ 

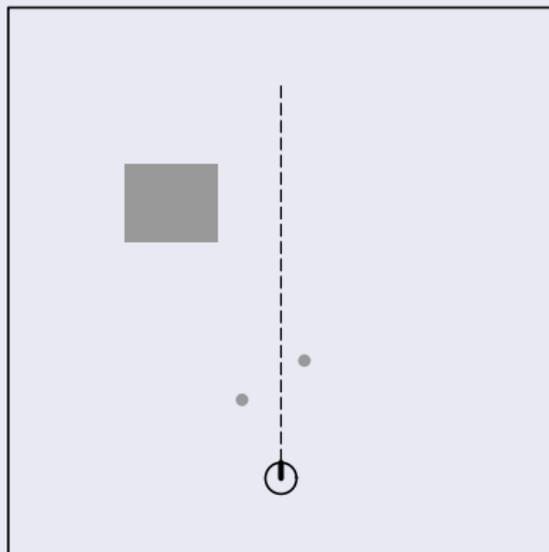
## Bemerkung

Einbeziehung von  $p_{\text{rand}}(z_t^k | x_t, m)$  und  $p_{\text{max}}(z_t^k | x_t, m)$

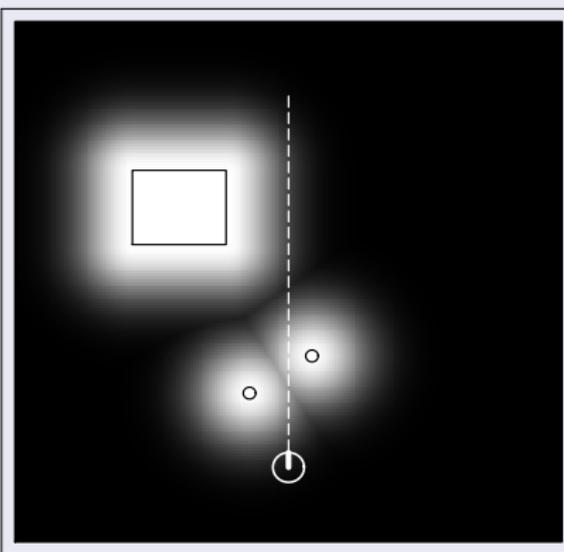
# Beispiel für ein Wahrscheinlichkeitsfeld

## Resultierendes Wahrscheinlichkeitsfeld

(a) example environment

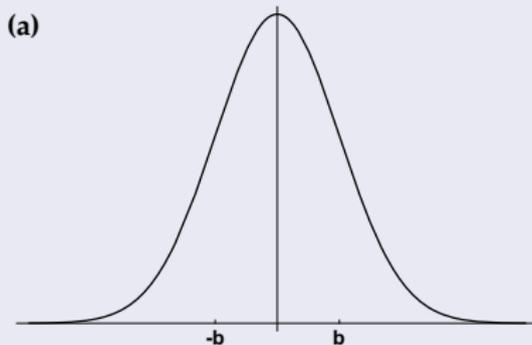


(b) likelihood field



# Die Funktion $\text{prob}(a, b)$

## Die Standardnormalverteilung



## Bedeutung

$$\text{prob}(a, b) = \epsilon_b(a) = \frac{1}{\sqrt{2\pi b^2}} e^{-\frac{1}{2} \frac{a^2}{b^2}}$$

# Algorithmus Wahrscheinlichkeitsfeld

## Algorithmus

LIKELIHOODFIELDMODEL( $z_t, x_t, m$ )

1  $q = 1$

2 **for all**  $k$

3     **do if**  $z_t^k \neq z_{\max}$

4         **then**  $x_{z_t^k} = x + x_{k,sens} \cdot \cos \theta - y_{k,sens} \cdot \sin \theta + z_t^k \cdot \cos(\theta + \theta_{k,sens})$

5          $y_{z_t^k} = y + y_{k,sens} \cdot \cos \theta - x_{k,sens} \cdot \sin \theta + z_t^k \cdot \cos(\theta + \theta_{k,sens})$

6          $dist = \min_{x',y'} \left\{ \sqrt{(x_{z_t^k} - x')^2 + (y_{z_t^k} - y')^2} \mid \langle x', y' \rangle \text{ in } m \text{ besetzt} \right\}$

7          $q = q \cdot \left( z_{\text{hit}} \cdot \text{prob} (dist, \sigma_{\text{hit}}) + \frac{z_{\text{random}}}{z_{\max}} \right)$

8 **return**  $q$

# Enschätzung Wahrscheinlichkeitsfelder

## Pro

- schnell berechnenbar
- auch in schwierigen Umgebungen robust

## Kontra

- keine Modellierung freier Fläche
- keine explizite Einbeziehung dynamischer Objekte

- 1 Einführung
- 2 Das Strahlenmodell für Entfernungssensoren
- 3 Wahrscheinlichkeitsfelder
- 4 Korrelationsbasierte Modelle**
- 5 Quellenangaben

# Map-Matching

Was ist das ?

## Ansatz

Untersuchung von **Übereinstimmungen** zwischen Messungen und der Karte

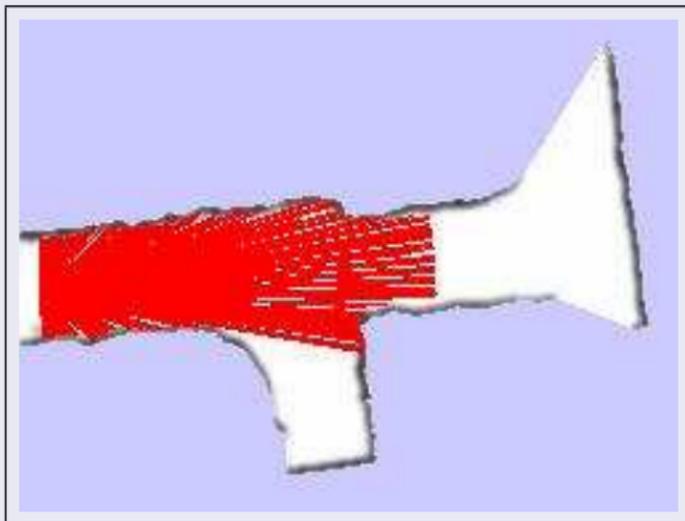
## Vorgehensweise

Zusammenfassung einer geringen Zahl von Messungen in sog. **lokale Karten**

# Map-Matching

Beispiel einer lokalen Karte

## Lokale Karte



# Zwei Arten von Karten

## Karte (allgemein)

$$m = \{m_1, m_2, \dots, m_n\}$$

## Wir unterscheiden

- 1 **Lokationsbasierte** Karten
  - $m_i \Leftrightarrow$  Punkt im Koordinatensystem
- 2 **Feature-**basierte Karten
  - $m_i \Leftrightarrow$  Objekt mit Eigenschaften

# Feature-basierte Karten: eine neue Abstraktionsebene

## Bemerkung

Im Folgenden **feature-basierte** Karten

## Bisher:

Modelle, die auf reinen Sensordaten aufbauten

## Jetzt:

Erkennung von Mustern in den Messdaten ( $\Rightarrow$  Objektbildung)

# Feature extraction

## Annahme

Funktion  $f(z_t)$  erkennt **Features** aus Sensordaten.

## Notation

$$f(z_t) = \{f_t^1, f_t^2, \dots, f_t^k\}$$

## Wichtig

Die Anzahl der erkannten Features **variiert** je nach Scan.

# Landmarken (Erinnerung)

## Definition Landmarke

Gut unterscheidbares, stationäres Objekt, das vom Roboter zur Navigation verwendet werden kann.

## Im Folgenden

Sensormodell für Landmarken

# Sensormodell für Landmarken

## Ziel

Das Sensormodell soll

- die Entfernung zur Landmarke  $r$
- den Beobachtungswinkel  $\phi$

berechnen können.

## Später

Wie wird  $\begin{pmatrix} r \\ \phi \end{pmatrix}$  berechnet ?

# Erkennung von Landmarken: Signaturen

## Signatur

Numerischer Wert zur Unterscheidung von Landmarken

## Festlegung

Feature extractor  $f$  liefert **Signatur**  $s$  der Landmarke

# Eigenschaften von Features

## Erinnerung

Wir benutzen **Feature-basierte** Karten

## Karte

$$m = \{m_1, m_2, \dots, m_n\}$$

## Eigenschaften der $m_i$

$$m_i = \left( \underbrace{m_{i,x}, m_{i,y}}_{\text{Koordinaten}}, \underbrace{s_i}_{\text{Signatur}} \right)$$

# Features: konkret

## Landmarkenerkennung

$$f(z_t) = \{f_t^1, f_t^2, \dots, f_t^k\} = \left\{ \begin{pmatrix} r_t^1 \\ \phi_t^1 \\ s_t^1 \end{pmatrix}, \begin{pmatrix} r_t^2 \\ \phi_t^2 \\ s_t^2 \end{pmatrix}, \dots, \begin{pmatrix} r_t^k \\ \phi_t^k \\ s_t^k \end{pmatrix} \right\}$$

# Annahme der bekannten Korrespondenz

## Zuordnungsproblem

Zu jedem Zeitpunkt  $t$  kann das Feature  $f_t^i$  **eindeutig** einer Landmarke  $m_j$  aus der Karte zugeordnet werden.

## Korrespondenzvariable

Wertebereich  $c_t^i \in \{1, 2, \dots, N\}$

**Bedeutung** Dem Feature  $f_t^i$  wird die Landmarke  $c_t^i = j$  zugeordnet

## Intuition

$c_t^i$  ist die wahre Identität des Features  $f_t^i$

# Sensormodell für Landmarken

Nun

Realisierung des Modells für Landmarken

Berechnung

$$\begin{pmatrix} r_t^i \\ \phi_t^i \\ s_t^i \end{pmatrix} = \begin{pmatrix} \sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2} \\ \arctan 2(m_{j,y} - y, m_{j,x} - x) - \theta \\ s_j \end{pmatrix} + \begin{pmatrix} \epsilon_{\sigma_r^2} \\ \epsilon_{\sigma_\phi^2} \\ \epsilon_{\sigma_s^2} \end{pmatrix}$$

# Algorithmus: Landmarkenmodell mit bekannter Korrespondenz

## Algorithmus

LANDMARKMODEL( $f_t^i, c_t^i, x_t, m$ )

1  $j = c_t^i$

2  $\hat{r} = \sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2}$

3  $\hat{\phi} = \arctan 2(m_{j,y} - y, m_{j,x} - x)$

4  $q = \mathbf{prob}(r_t^i - \hat{r}, \sigma_r) \cdot \mathbf{prob}(\phi_t^i - \hat{\phi}, \sigma_\phi) \cdot \mathbf{prob}(s_t^i - s_j, \sigma_s)$

5 **return**  $q$

# Navigation mit Landmarken: Schätzung von Positionen

## Problemstellung

Gegeben:

$$p(f_t^i \mid c_t^i, x_t, m)$$

Gesucht:

$$p(x_t \mid f_t^i, c_t^i, m)$$

# Navigation mit Landmarken: Schätzung von Positionen

## Lösung

### Annahme

**Gleichverteilung** von  $p(x_t | f_t^i, c_t^i, m)$

### Berechnung (nach dem Satz von Bayes)

$$\begin{aligned} p(x_t | f_t^i, c_t^i, m) &= \eta p(f_t^i | c_t^i, x_t, m) \cdot p(x_t | c_t^i, m) \\ &= \eta p(f_t^i | c_t^i, m) \end{aligned}$$

# Satz von Bayes

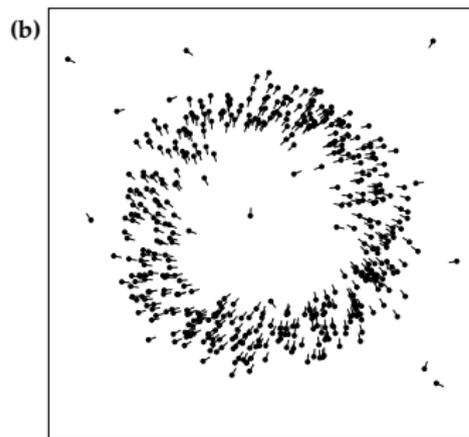
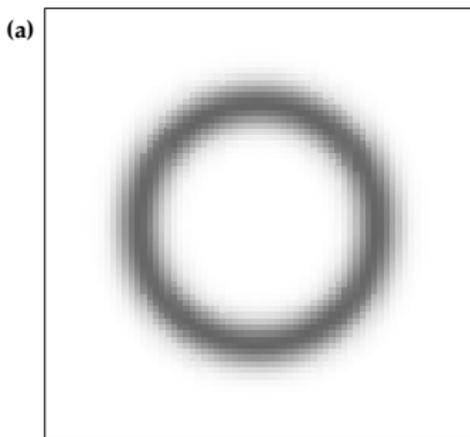
## Erinnerung

### Satz von Bayes

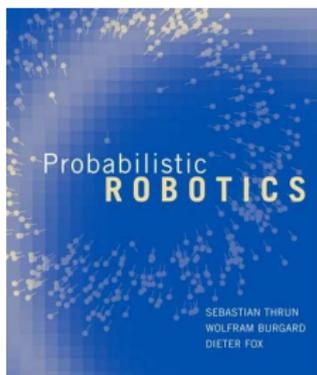
$$p(x | y) = \frac{p(y | x)p(x)}{p(y)} = \eta p(y | x)p(x)$$

# Navigation mit Landmarken: Schätzung von Positionen

## Grafisch



- 1 Einführung
- 2 Das Strahlenmodell für Entfernungssensoren
- 3 Wahrscheinlichkeitsfelder
- 4 Korrelationsbasierte Modelle
- 5 Quellenangaben**



Sebastian Thrun, Wolfram Burgard, Dieter Fox.

Probabilistic Robotics.

*Intelligent robotics and autonomous agents.*

MIT Press, 2005.

ISBN: 0-262-20162-3

URL: [www.probabilistic-robotics.org](http://www.probabilistic-robotics.org)