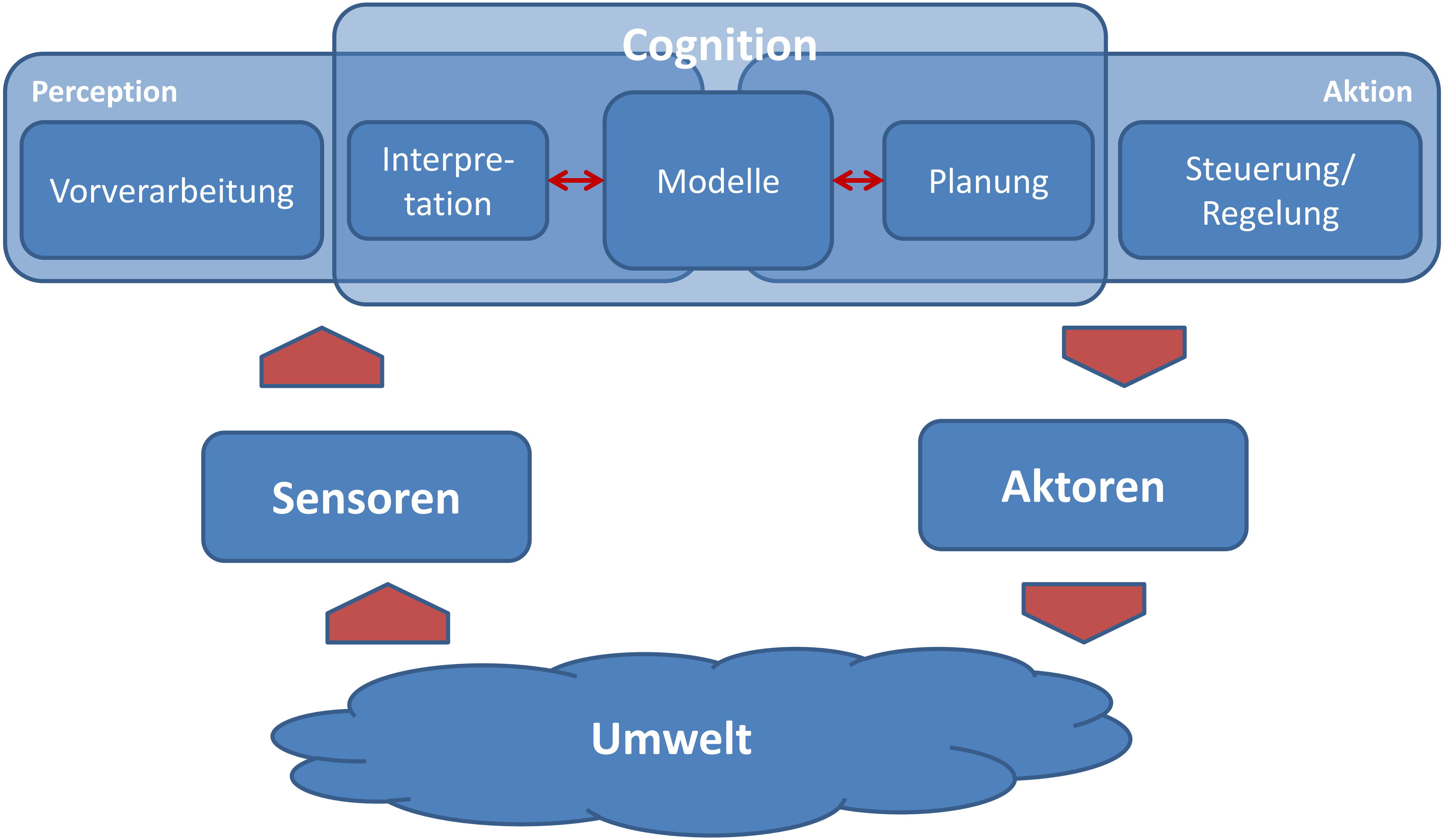
Advanced Robot Perception

Fortgeschrittene Konzepte der Wahrnehmung für Robotersysteme

Georg von Wichert, Siemens Corporate Technology

The Perception-Cognition-Action-Loop



Bayesfilterung ist fundamental!

$$Bel(x_t) = \eta \ \underbrace{P(z_t \mid x_t)}_{\text{Korrektur/ Mess-Update}} \ \underbrace{\int P(x_t \mid u_t, x_{t-1}) \ Bel(x_{t-1}) \ dx_{t-1}}_{\text{Prädiktion / Zeit-Update}}$$

Zustandsschätzung

- Kalman-Filter (Extended, **Unscented KF**)
- Partikelfilter
- Hidden Markov Modelle
- Dynamische Bayes'sche Netze
- Partially Observable Markov Decision Processes (POMDPs)

Spracherkennung
Gestenerkennung

Planung

Zur Erinnerung: Bayes Filter

- Prädiktion / Zeit-Update

$$\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1})\, bel(x_{t-1})\, dx_{t-1}$$

- Korrektur / Mess-Update

$$bel(x_t) = \eta\, p(z_t \mid x_t)\, \overline{bel}(x_t)$$

Diskretes Kalman-Filter

- Schätzt den Zustand x eines zeitdiskreten gesteuerten Prozesses, der durch die folgende lineare stochastische Differenzengleichung beschrieben wird

$$x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t$$

$$p(\mathbf{x}) \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}):$$

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2} |\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^t \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})}$$

wobei

$$z_t = C_t x_t + \delta_t$$

eine indirekte Messung des Zustands ist

Diskretes Kalman-Filter

- Prädiktion / Zeit-Update

$$\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t \qquad \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$$

- Korrektur / Mess-Update

$$\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t) \qquad \Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$$

$$K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$$

Nochmal zur Erinnerung: Bayes Filter

- Prädiktion / Zeit-Update

$$\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1})\, bel(x_{t-1})\, dx_{t-1}$$

- Korrektur / Mess-Update

$$bel(x_t) = \eta\, p(z_t \mid x_t)\, \overline{bel}(x_t)$$

Nichtlineare dynamische Systeme

- In der Realität sind die uns interessierenden Systeme meist nicht linear!

$$x_t = g(u_t, x_{t-1}) \qquad \cancel{x_t = A_t x_{t-1} + B_t u_t}$$

$$z_t = h(x_t) \qquad \cancel{z_t = C_t x_t}$$

→ Extended Kalman Filter

Abbildung einer Zufallsvariablen durch lineare Funktion

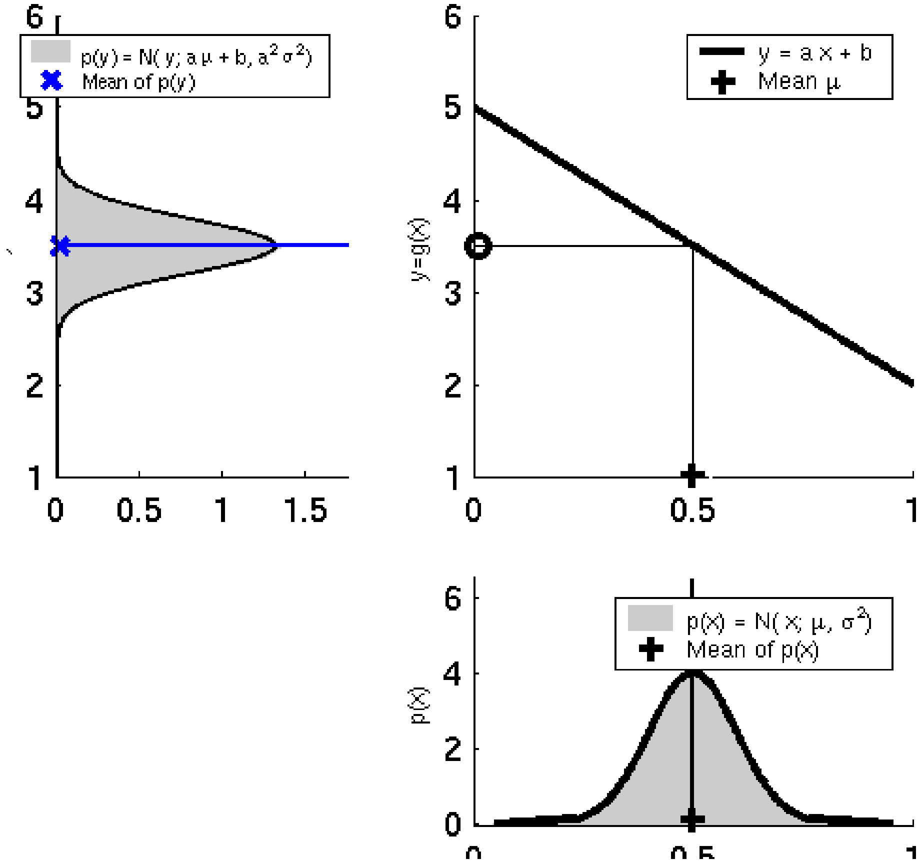
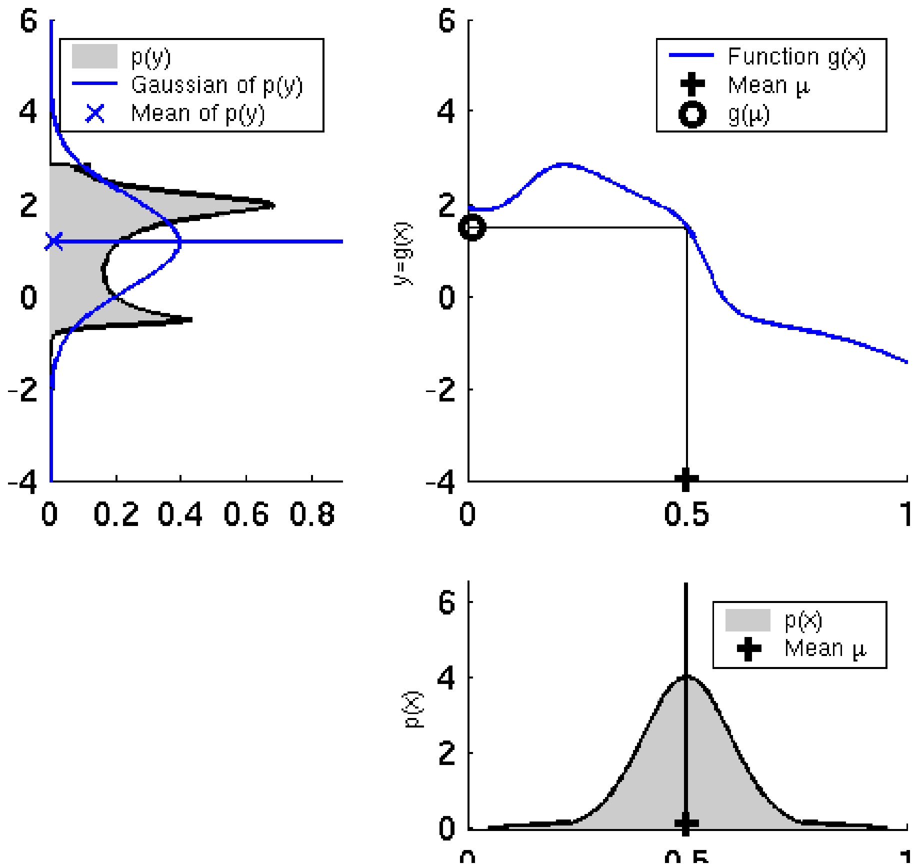
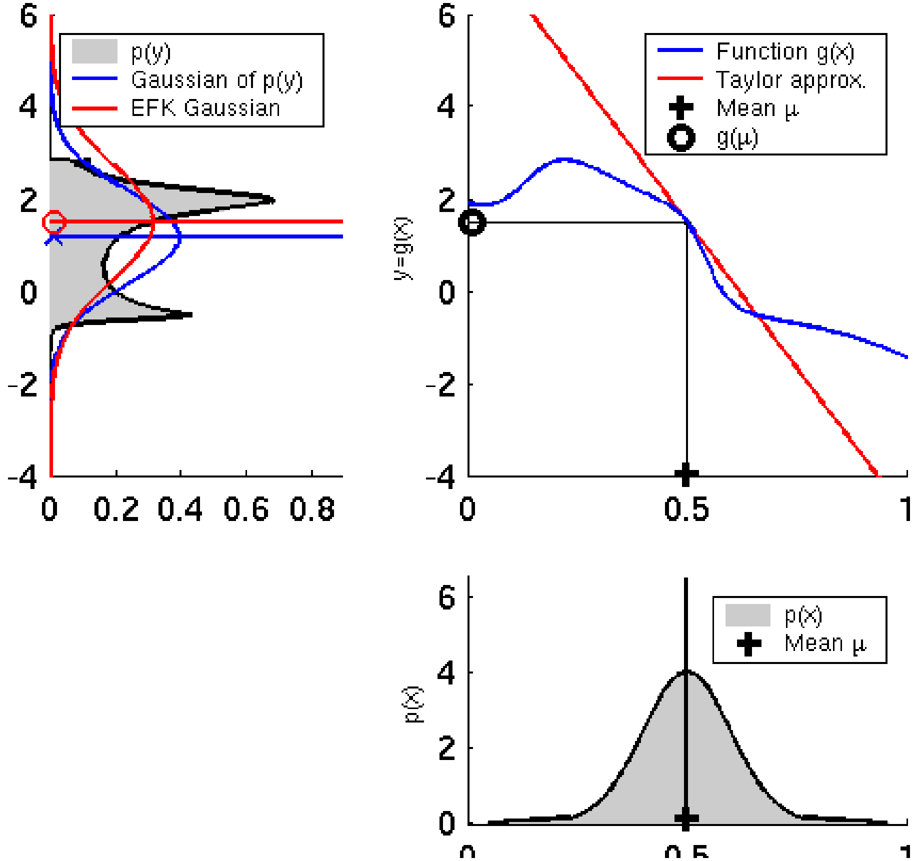


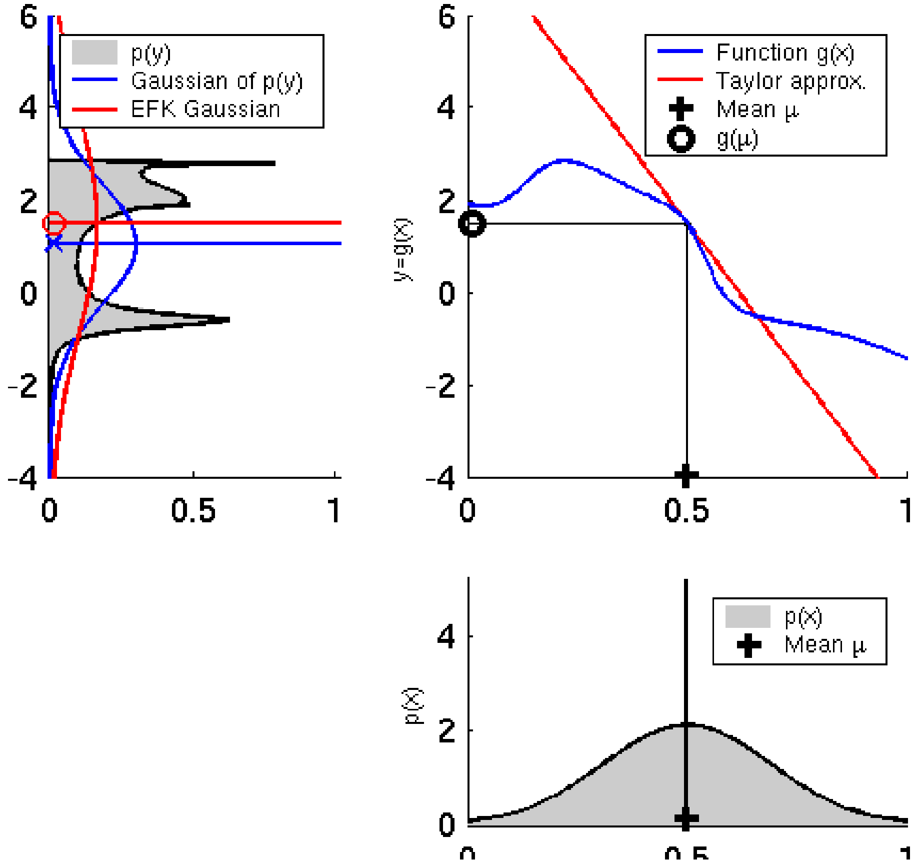
Abbildung einer Zufallsvariablen durch nichtlineare Funktion



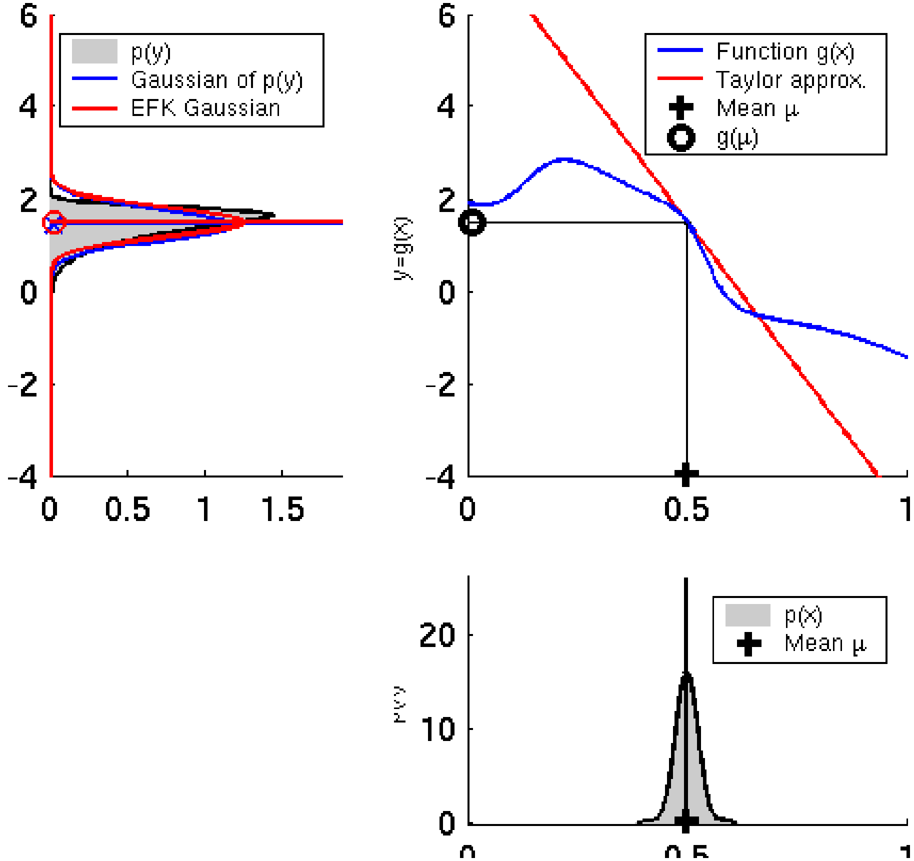
EKF Linearisierung (1)



EKF Linearisierung (2)



EKF Linearisierung (3)



EKF Linearisierung : Taylor Reihenentwicklung (1. Ordnung)

- Prädiktion:

$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}} (x_{t-1} - \mu_{t-1})$$

$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + G_t \, (x_{t-1} - \mu_{t-1})$$

Jakobimatrix von g(u,x)

- Korrektur:

Jakobimatrix von h(x)

$$h(x_t) \approx h(\overline{\mu}_t) + \frac{\partial h(\overline{\mu}_t)}{\partial x_t} (x_t - \overline{\mu}_t)$$

$$h(x_t) \approx h(\overline{\mu}_t) + H_t \, (x_t - \overline{\mu}_t)$$

EKF Algorithmus

1. **Extended_Kalman_filter**(μ_{t-1}, Σ_{t-1}, u_t, z_t):

2. Prädiktion / Zeit-Update:

3. $\bar{\mu}_t = g(u_t, \mu_{t-1})$ $\longleftarrow$ $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$

4. $\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t$ $\longleftarrow$ $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$

5. Korrektur / Meß-Update:

6. $K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$ $\longleftarrow$ $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$

7. $\mu_t = \bar{\mu}_t + K_t (z_t - h(\bar{\mu}_t))$ $\longleftarrow$ $\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$

8. $\Sigma_t = (I - K_t H_t) \bar{\Sigma}_t$ $\longleftarrow$ $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$

9. Return μ_t, Σ_t

$$H_t = \frac{\partial h(\bar{\mu}_t)}{\partial x_t} \qquad G_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}}$$

Zwei Sichten auf das Kalman-Filter

Optimaler Zustandsschätzer

- Beobachter für dynamische Systeme mit normalverteil-tem Rauschen auf Zustand und Messungen
- Ergebnis ist der geschätzte Zustandsvektor
  - Kovarianzmatrix eher ein Nebenprodukt

Bayes‘scher Belief-Tracker

- Bayes‘sche Interpretation des „belief“ als das gegenwärtig verfügbare Wissen
- Ergebnis ist die Dichte-funktion des Zustands:
  - Mittelwert und Kovarianzmatrix

Bayesfilterung ist fundamental!

$$Bel(x_t) = \eta \underbrace{P(z_t \mid x_t)}_{\text{Korrektur/ Mess-Update}} \underbrace{\int P(x_t \mid u_t, x_{t-1})\, Bel(x_{t-1})\, dx_{t-1}}_{\text{Prädiktion / Zeit-Update}}$$

- Kalman-Filter (Extended KF, **Unscented KF**)
- Partikelfilter
- Hidden Markov Modelle
- Dynamische Bayes'sche Netze
- Partially Observable Markov Decision Processes (POMDPs)

Zwischen-Fazit

- Bayes'sche Filter sind ein probabilistischer Ansatz zur Schätzung des Zustands dynamischer Systeme.
- …und zur **systematischen Datenfusion!**
  - Der Bayes'sche Satz erlaubt es uns in Kombination mit einigen gezielten Markov-Annahmen, anhand kausaler Sensormodellen durch **rekursives Bayes'sches Schließen** effizient Informationen zu akkumulieren und dabei nicht direkt beobachtbare Zustände zu schätzen

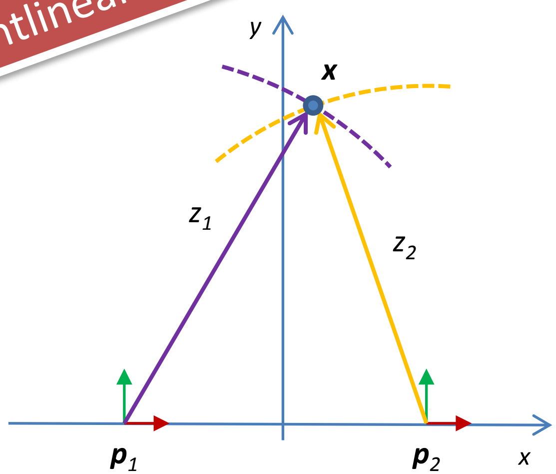
Datenfusion

- Kombination / Integration von Informationen aus unterschiedlichen Quellen, z.B.
  - Daten mehrerer Sensoren
  - Sensormessungen desselben Sensors, aber zu unterschiedlichen Zeitpunkten
- Erschließen nicht direkt und exakt messbarer Größen
  - Regelungstechnik: Beobachter
- Hauptproblem: Messfehler
  - Sensormessungen sind nicht widerspruchsfrei

Beispiel: Datenfusion mit dem Extended Kalman Filter

- Beobachtung eines beweglichen Objekts in der Ebene mit zwei ungerichteten Abstandsensoren

Nichtlinear $\longrightarrow$ EKF



$$\mathbf{p}_i = \begin{bmatrix} p_{i1} \\ p_{i2} \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

$$z_i = \|\mathbf{x} - \mathbf{p}_i\| = \sqrt{(x_1 - p_{i1})^2 + (x_2 - p_{i2})^2}$$

Sensormodell:

$$\mathbf{z}_t = \mathbf{h}(\mathbf{x}_t) + \nu_t = \begin{bmatrix} z_1(\mathbf{x}_t) \\ z_2(\mathbf{x}_t) \end{bmatrix} + \nu_t$$

Messrauschen: $\nu_t \sim \mathcal{N}(0, \mathrm{Q})$

Prozessmodell:

$$\mathbf{x}_t = \mathbf{g}(\mathbf{x}_{t-1}, \mathbf{u}_t) + \xi_t = \mathbf{x}_{t-1} + \epsilon(\mathbf{u}_t - \mathbf{x}_{t-1}) + \xi_t$$

Prozessrauschen: $\xi_t \sim \mathcal{N}(0, \mathrm{R})$

Beispiel: Datenfusion mit dem Kalman Filter

- H: Jakobimatrix des Messmodells

$$\mathbf{z}_t = \mathbf{h}(\mathbf{x}_t) + \nu_t = \begin{bmatrix} z_1(\mathbf{x}_t) \\ z_2(\mathbf{x}_t) \end{bmatrix} + \nu_t \qquad z_i = \|\mathbf{x} - \mathbf{p}_i\| = \sqrt{(x_1 - p_{i1})^2 + (x_2 - p_{i2})^2}$$

$$\mathbf{H}(\mathbf{x}) = \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial z_1}{\partial \mathbf{x}} \\ \frac{\partial z_2}{\partial \mathbf{x}} \end{bmatrix} \qquad \frac{\partial z_i}{\partial \mathbf{x}} = \begin{bmatrix} \frac{x_1 - p_{i1}}{\sqrt{(p_{i1}-x_1)^2+(p_{i2}-x_2)^2}} & \frac{x_2 - p_{i2}}{\sqrt{(p_{i1}-x_1)^2+(p_{i2}-x_2)^2}} \end{bmatrix}$$

- G: Jakobimatrix des Prozessmodells

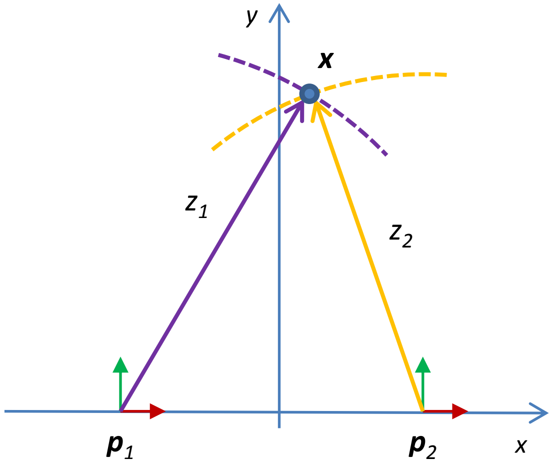
$$\mathbf{x}_t = \mathbf{g}(\mathbf{x}_{t-1}, \mathbf{u}_t) + \xi_t = \mathbf{x}_{t-1} + \epsilon(\mathbf{u}_t - \mathbf{x}_{t-1}) + \xi_t$$

$$\mathbf{G}(\mathbf{x}) = \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}} = (1 - \epsilon)\ \mathbf{I}$$

```matlab
classdef kalman2
    %KALMAN1 EKF-Demo für ARP-Vorlesung%...%

    properties
        %%% Parameter
        R % Kovarianmatrix des Systemrauschens
        Q % Kovarianzmatrix des Messrauschens

        %%% Zustandsvariablen
        Sigma      % korrigierte Zustandskovarianzmatrix
        x          % korrigierter Zustand
        Sigma_Praed % prädizierte Zustandskovarianzmatrix
        x_Praed     % prädizierter Zustand

        %%% Modelparameter
        statelen % Size of the state vector
        measlen % Size of the measruement vector
        sensor1Pos
        sensor2Pos
    end

    methods
        %%% Konstructor
        function obj = kalman2(R,Q,statelen,measlen, sensor1Pos, sensor2Pos)...

        %%% Initialisierung der Zustandsvariablen des Filters
        function obj = init(obj)...

        %%% EKF Algorithmus
        function obj = predict(obj,u)
            G = computeG(obj, obj.x); % Jakobimatrix der Systemfunktion g()
            obj.x_Praed = g(obj, obj.x,u);
            obj.Sigma_Praed = G * obj.Sigma * G' + obj.R;
        end
        function [obj K] = correct(obj,z)
            H = computeH(obj, obj.x_Praed); % Jakobimatrix der Messfunktion h()
            K = obj.Sigma_Praed * H' / (H * obj.Sigma_Praed * H' + obj.Q); % Kalmangain(Matrix)
            obj.x = obj.x_Praed + K * (z - h(obj,obj.x_Praed));
            obj.Sigma = (eye(obj.statelen) - K * H) * obj.Sigma_Praed;
        end

        %%% Prozessmodell und Messmodell
        function xn = g(obj, x, u)
            xn = x + 0.03 * (u - x);
        end
        function z = h(obj, x)
            dist1 = norm(x - obj.sensor1Pos);
            dist2 = norm(x - obj.sensor2Pos);
            z = [dist1 dist2]';
        end
        function G = computeG(obj,x)
            G = eye(obj.statelen) * ( 1 - 0.03 );
        end
        function H = computeH(obj,x)
            p = obj.sensor1Pos;
            H(1,1) = -(2*p(1) - 2*x(1))/(2*((p(1) - x(1))^2 + (p(2) - x(2))^2)^(1/2));
            H(1,2) = -(2*p(2) - 2*x(2))/(2*((p(1) - x(1))^2 + (p(2) - x(2))^2)^(1/2));

            p = obj.sensor2Pos;
            H(2,1) = -(2*p(1) - 2*x(1))/(2*((p(1) - x(1))^2 + (p(2) - x(2))^2)^(1/2));
            H(2,2) = -(2*p(2) - 2*x(2))/(2*((p(1) - x(1))^2 + (p(2) - x(2))^2)^(1/2));
        end

        %%% Simulation helpers
        function newx = simulateProcess(obj, x, u, processNoise)...
        function z = simulateMeasurement(obj, x,measurementNoise)...

    end
end
```

Ergebnis der „Versuche“

- Durch Fusion zweier ungerichteter Abstandsmessungen kann die 2D-Position eines Objekts bestimmt werden
  - Systemzustand ist mit dem einzelnen Sensor nicht beobachtbar
  - Aufstellung der Sensoren wesentlich
    - Bei dem Beispielszenario, sind ist die Beobachtbarkeit in Teilen des Arbeitsraums sehr eingechränkt!
  - Das Bayesfilter (hier EKF) schätzt den „Belief“ $bel(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{z}_1,\boldsymbol{z}_2,\ldots)$ als Dichtefunktion, die insbesondere die Genauigkeit des Wissens repräsentiert.



Datenfusion mit dem Kalman Filter

- Bisher nehmen wir implizit an:
  - Alle Messungen der Sensoren stammen von dem modellierten Objekt.
  - Es gibt genau ein Objekt!

$$\int p(\mathbf{x}) d\mathbf{x} = 1$$

  - Unsere Implementierung: Wir machen in jedem Schritt eine Messung, d.h. das Objekt wird immer gesehen.
    - Kein Problem: Nur Zeitupdate, falls keine Messungen vorliegen.

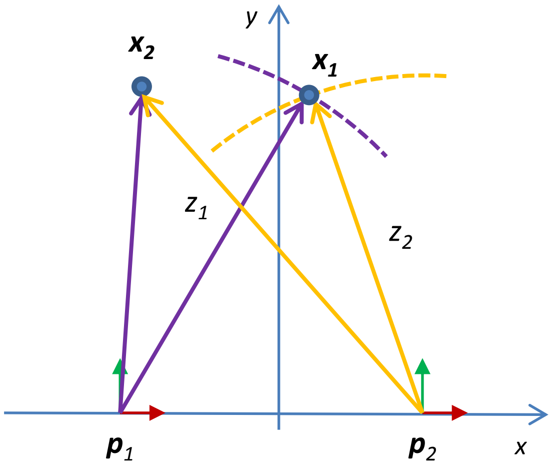
Komplexere Zustandsräume

- Was ändert sich, wenn wir mehrere Objekte (hier mit gleicher Dynamik) in der Welt haben?

$$\mathbf{x}^* = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \qquad \mathbf{u}^* = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix} \qquad \xi^* = \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}$$

$$\mathbf{x}_t^* = \mathbf{g}^*(\mathbf{x}_{t-1}^*, \mathbf{u}_t^*) + \xi_t^* = \begin{bmatrix} \mathbf{g}(\mathbf{x_1}_{t-1}, \mathbf{u_1}_t) \\ \mathbf{g}(\mathbf{x_2}_{t-1}, \mathbf{u_2}_t) \end{bmatrix} + \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}$$

$$\mathbf{z}_t^* = \mathbf{h}^*(\mathbf{x}_t^*) + \nu_t^* = \begin{bmatrix} \mathbf{h}(\mathbf{x_1}_t) \\ \mathbf{h}(\mathbf{x_2}_t) \end{bmatrix} + \begin{bmatrix} \nu_1 \\ \nu_2 \end{bmatrix}$$



Bei statistischer Unabhängigkeit der Störungen:

$$\left.\begin{matrix} \nu_i \sim \mathcal{N}(0, \mathrm{Q_i}) \\ \xi_i \sim \mathcal{N}(0, \mathrm{R_\rangle}) \end{matrix}\right\} \qquad bel(\mathbf{x}^*) = p(\mathbf{x}_1 | \mathbf{z}_t, \ldots, \mathbf{z}_0)\ p(\mathbf{x}_2 | \mathbf{z}_t, \ldots, \mathbf{z}_0)$$

Faktorisierung des Beliefs $\longrightarrow$ zwei getrennte Filter

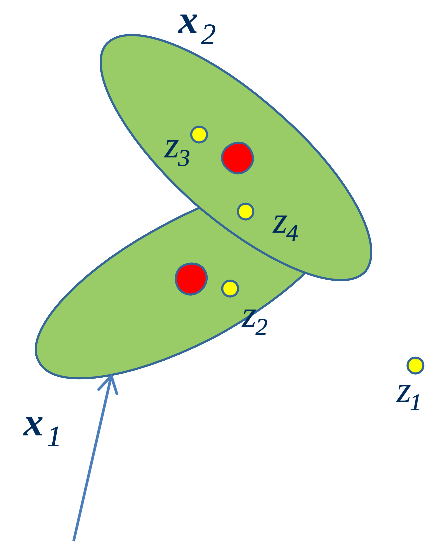
Datenassoziation bei mehreren Objekten

- Frage: Welche Messungen nehmen wir für die Korrekturen/Messupdates der jeweiligen Bayes-Filter?
  - Einfachste Lösung: „Expectation Gate“

$$\mu_t = \bar{\mu}_t + K_t \underbrace{(z_t - h(\bar{\mu}_t))}$$

Innovation $\Delta_t = z_t - h(\bar{\mu}_t)$

Kovarianzmatrix der Innovation $S_t = H_t \bar{\Sigma}_t H_t^T + Q_t$



  - Wir nehmen eine Messung $\boldsymbol{z}_t$ an, wenn $\Delta_t^T S_t^{-1} \Delta_t \leq \lambda(P_e)$
  - $\lambda(P_e)$ ist eine Schwelle, sodass die Innovation mit Wahrscheinlichkeit P_e in das ellipsoide „Gate“ fällt.

Datenassoziation bei mehreren Objekten

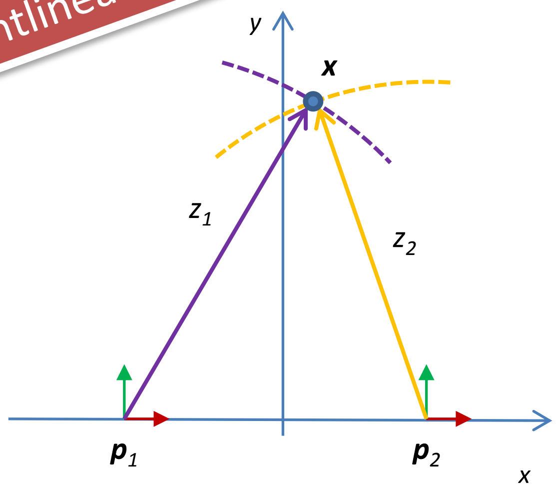
- **Datenassoziation** entspricht dem früher in der Vorlesung betrachteten Problem der **Segmentierung**
  - Welche sensorielle Beobachtung wird von welchem der modellierten Weltzustände beeinflusst.
  - Nur relevant, wenn wir den Belief $p(\boldsymbol{x})$ partitionieren bzw. wenn es nichtmodellierte Umweltphänomene oder nichtmodellierte Zustände gibt, welche dennoch sensorisch wahrgenommen werden
    - Beides ist immer der Fall!

- Zu komplexeren Datenassoziationsansätzen beachten Sie auch die Zusatzunterlage zur Vorlesung (PDF in moodle)

Beispiel: Datenfusion mit dem Extended Kalman Filter

- Beobachtung eines beweglichen Objekts in der Ebene mit zwei ungerichteten Abstandsensoren

Nichtlinear → EKF



$$\mathbf{p}_i = \begin{bmatrix} p_{i1} \\ p_{i2} \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad \mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

$$z_i = \|\mathbf{x} - \mathbf{p}_i\| = \sqrt{(x_1 - p_{i1})^2 + (x_2 - p_{i2})^2}$$

Sensormodell:

$$\mathbf{z}_t = \mathbf{h}(\mathbf{x}_t) + \nu_t = \begin{bmatrix} z_1(\mathbf{x}_t) \\ z_2(\mathbf{x}_t) \end{bmatrix} + \nu_t$$

Messrauschen: $\nu_t \sim \mathcal{N}(0, \mathrm{Q})$

Prozessmodell:

$$\mathbf{x}_t = \mathbf{g}(\mathbf{x}_{t-1}, \mathbf{u}_t) + \xi_t = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{t-1} + \xi_t$$

Prozessrauschen: $\xi_t \sim \mathcal{N}(0, \mathrm{R})$

Beispiel: Datenfusion mit dem Kalman Filter

- H: Jakobimatrix des Messmodells

$$\mathbf{z}_t = \mathbf{h}(\mathbf{x}_t) + \nu_t = \begin{bmatrix} z_1(\mathbf{x}_t) \\ z_2(\mathbf{x}_t) \end{bmatrix} + \nu_t \qquad z_i = \|\mathbf{x} - \mathbf{p}_i\| = \sqrt{(x_1 - p_{i1})^2 + (x_2 - p_{i2})^2}$$

$$\mathbf{H}(\mathbf{x}) = \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial z_1}{\partial \mathbf{x}} \\ \frac{\partial z_2}{\partial \mathbf{x}} \end{bmatrix} \qquad \frac{\partial z_i}{\partial \mathbf{x}} = \begin{bmatrix} \frac{x_1 - p_{i1}}{\sqrt{(p_{i1}-x_1)^2 + (p_{i2}-x_2)^2}} & \frac{x_2 - p_{i2}}{\sqrt{(p_{i1}-x_1)^2 + (p_{i2}-x_2)^2}} \end{bmatrix}$$

- G: Jakobimatrix des Prozessmodells

$$\mathbf{x}_t = \mathbf{g}(\mathbf{x}_{t-1}, \mathbf{u}_t) + \xi_t = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{t-1} + \xi_t$$

$$\mathbf{G}(\mathbf{x}) = \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Maschinelle Wahrnehmung

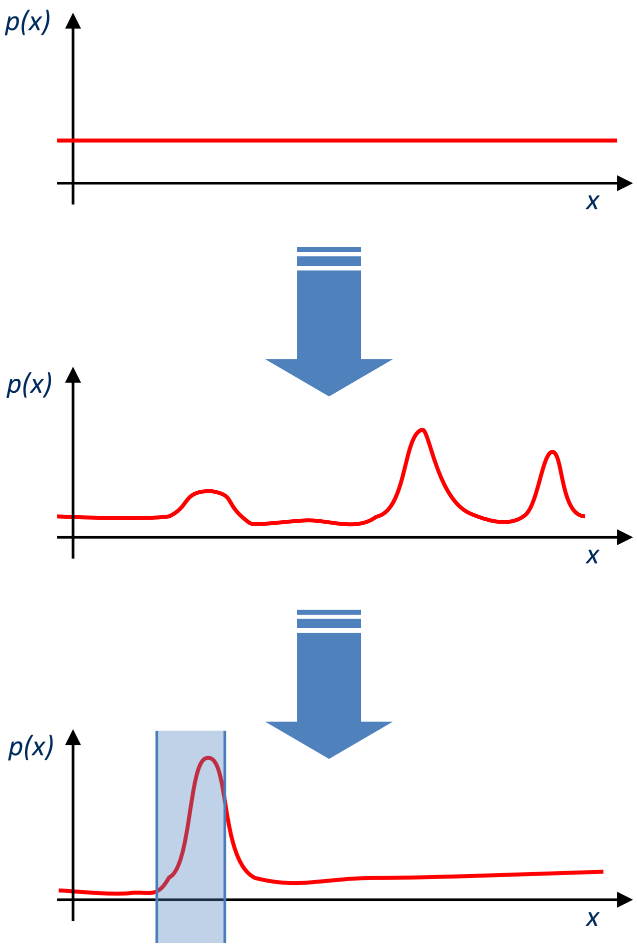
- Wahrnehmung = Zustandsschätzung = Datenfusion
  - Die interessierenden Variablen sind fast nie direkt beobachtbar (vgl. Konzept der Beobachtbarkeit in der Regelungstechnik)
  - Alle Messungen sind fehlerbehaftet

- Bayes-Filter, wie z.B. das Kalman-Filter, bilden einen Wahrnehmungsprozeß unter Unsicherheit ab
  - Der „Belief" *bel*(***x***) = p(***x***|$\boldsymbol{z}_1$,$\boldsymbol{z}_2$,…) repräsentiert unser aktuelles Wissen über den Zustand ***x*** der relevanten Welt

- Wahrnehmung (unter Unsicherheit) erfordert immer Modelle
  - Dynamik $$x_t = g(u_t, x_{t-1})$$
  - Beobachtungsvorgänge $$z_t = h(x_t)$$

Wahrnehmung als probabilistische Datenfusion

- Alle relevanten Fakten werden als Zufallsvariablen modelliert
  - Der aktuelle Wissenstand (belief) bezgl. einer Variablen wird durch die Wahrscheinlichkeitsverteilung über den jeweiligen Wertebereich repräsentiert
  - Alle Informationsquellen (z.B. die Sensoren) werden probabilistisch modelliert
- Integration neuer Messungen ist möglich **ohne irreführende Schwellwert-Entscheidungen zu fällen**
- Nutzen der statistischen Unabhängigkeit um das Gesamtproblem in kleiner teilprobleme zu zerlegen

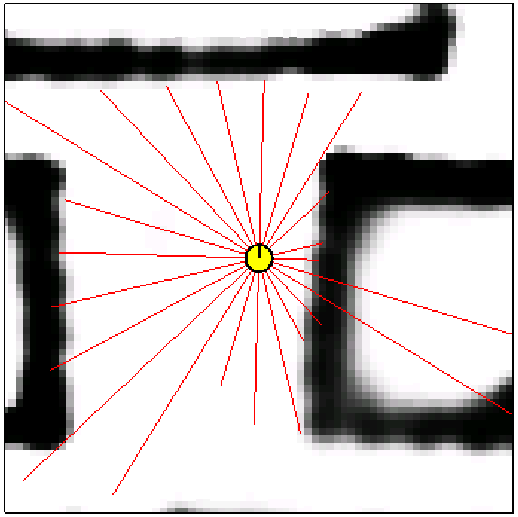
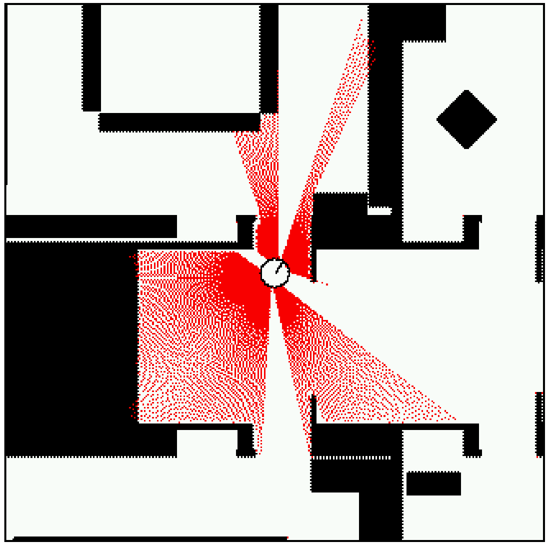
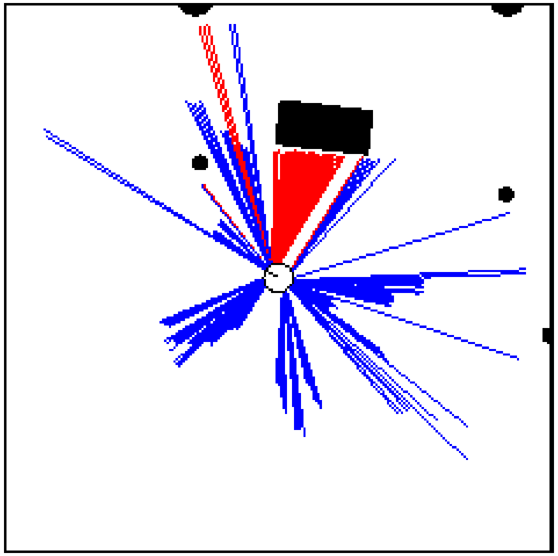


Wahrnehmungsprozess

$$Bel(x_t) = \underbrace{\eta\ P(z_t \mid x_t)}_{\text{Korrektur/ Mess-Update}} \underbrace{\int P(x_t \mid u_t, x_{t-1})\ Bel(x_{t-1})\ dx_{t-1}}_{\text{Prädiktion / Zeit-Update}}$$

- Bayes-Filter als Modell für Wahrnehmungsprozess unter Unsicherheit
  - Dynamischer Zustand der Welt
  - Sensoren
- Modelle erforderlich
  - Prozessmodell: $P(\boldsymbol{x}_t / \boldsymbol{u}_t, \boldsymbol{x}_{t-1})$
  - $\boxed{\text{Sensormodell: } P(\boldsymbol{z}_t / \boldsymbol{x}_t)}$

1. Beispiel: Abstandssensoren

- Aufgabe: Bestimme *P(**z**|**x**)*, d.h., Die Wahrscheinlichkeit der Messung ***z***, wenn der Roboter an Position ***x*** steht.
- **Frage:** Woher bekommen wir P(***z***|***x***)?
- **Ansatz:** Versuchen wir mal die Messung zu erklären.

Die folgenden Folien sind Adaptionen von Folien zum Buch „Probabilistic Robotics“
http://www.probabilistic-robotics.org/

Ein Probabilistisches Model für Abstandssensoren

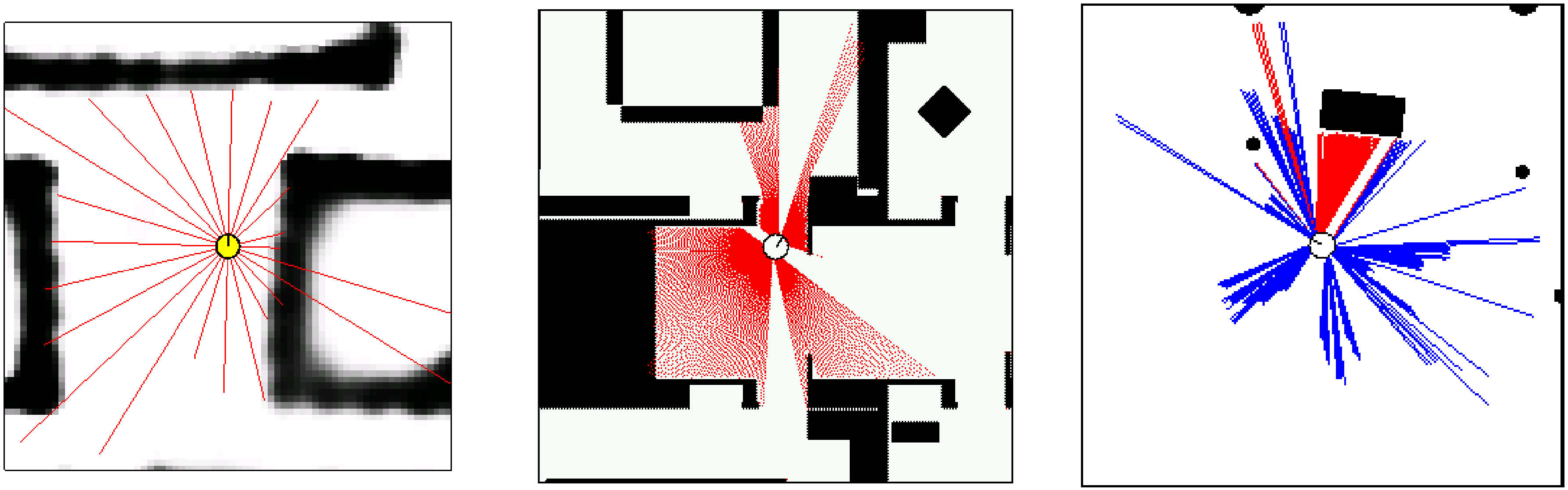
- Ein Laserscan ***z*** besteht aus *K* Einzelmessungen.

$$z = \{z_1, z_2, \ldots, z_K\}$$

- **Annahme:** Gegeben die Position *x*, sind die einzelne Messungen bedingt unabhängig.

$$P(z \mid x, m) = \prod_{k=1}^{K} P(z_k \mid x, m)$$

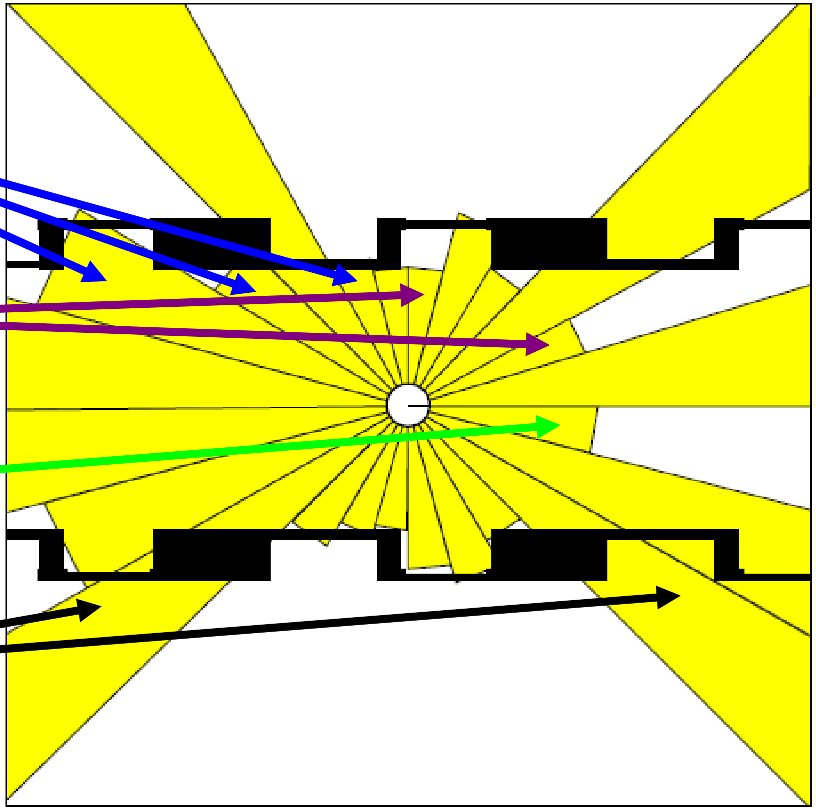
Beam-based Sensor Model



$$P(z \mid x, m) = \prod_{k=1}^{K} P(z_k \mid x, m)$$

Typische Quellen für Messfehler

1. Strahlen werden von Hindernissen reflektiert
2. Strahlen werden von Personen reflektiert /bzw. sind Crosstalk
3. Tatsächlich zufällige Messungen
4. Maximalabstand (keine Reflexion)

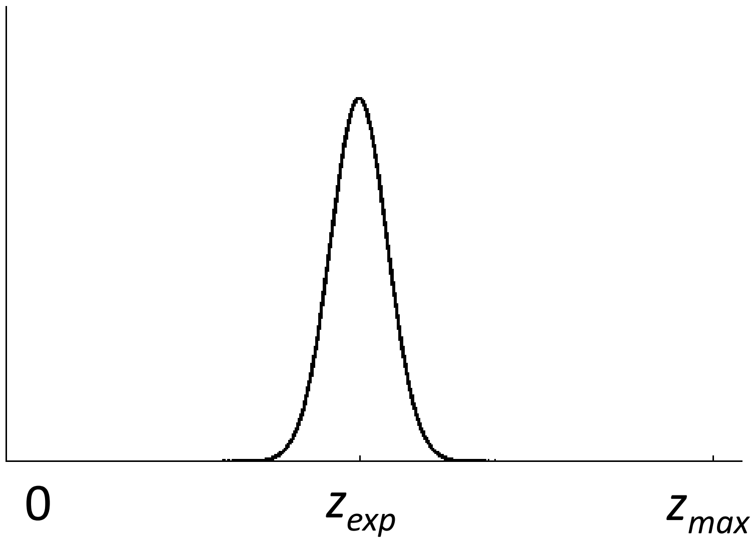


Typische Quellen für Messfehler

- Messung wird verursacht durch…
  - Bekannte Objekte (Wände in der Karte).
  - Cross-talk.
  - Unerwartete / nicht modellierte Objekte in der Umgebung (Menschen, Möbel, …).
  - Gar keine Objekte (Totalreflexion, Glas, Abstände außerhalb des Messbereichs …).
- Unsicherheit resultiert aus Ungenauigkeiten…
  - bei der eigentlichen Abstandsmessung.
  - bei der Position bekannter Objekte (Umgebungsmodell).
  - hinsichtlich der Existenz und Position unbekannter Objekte.
  - bei der Beantwortung der Frage, ob ein Objekt tatsächlich von Strahl getroffen wurde.

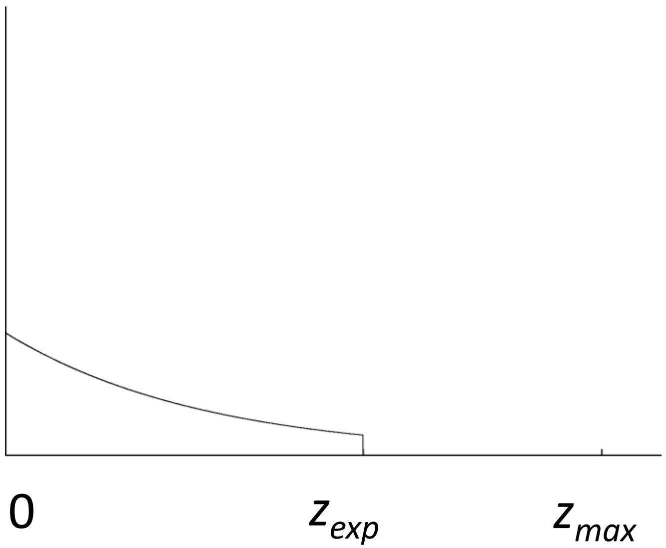
Strahlbasiertes Modell

Tatsächliche Meßfehler



$$P_{hit}(z \mid x, m) = \eta \frac{1}{\sqrt{2\pi b}} e^{-\frac{1}{2}\frac{(z - z_{\exp})^2}{b}}$$

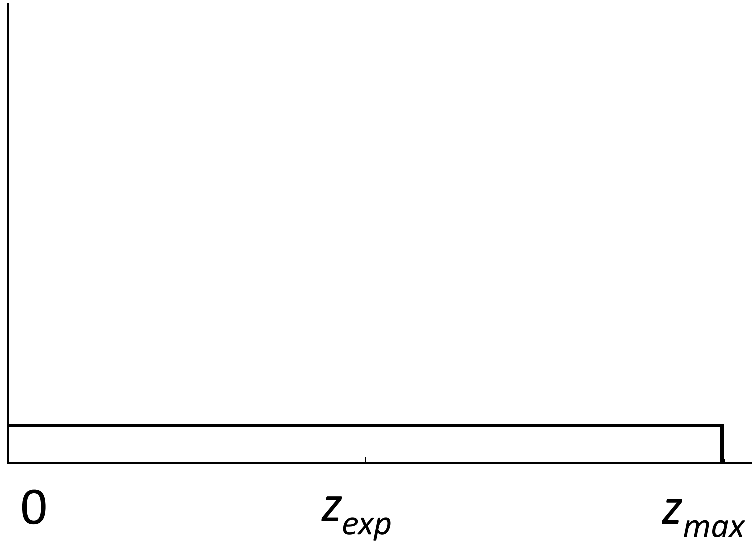
Unerwartete Objekte



$$P_{\text{unexp}}(z \mid x, m) = \begin{cases} \eta\, \lambda\, e^{-\lambda z} & z < z_{\exp} \\ 0 & sonst \end{cases}$$

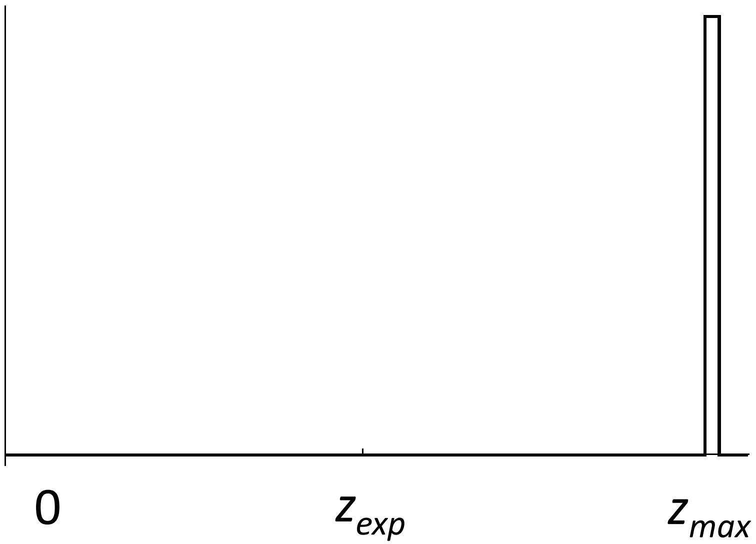
Strahlbasiertes Modell

Zufällige Messung



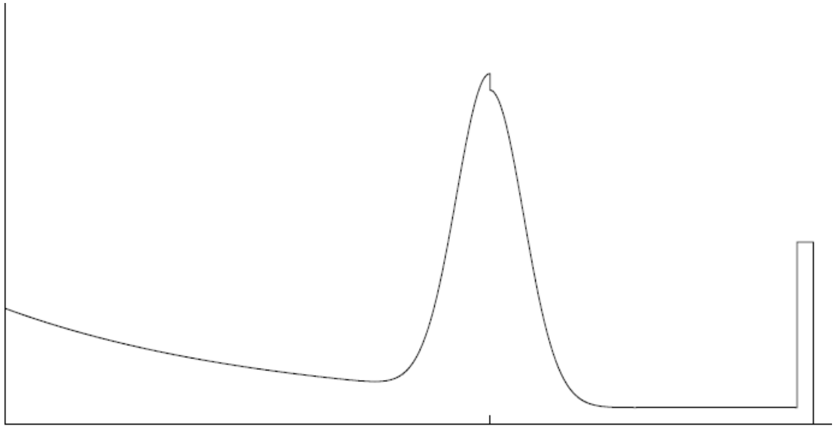
$$P_{rand}(z \mid x, m) = \eta \frac{1}{z_{\max}}$$

Maximalabstand



$$P_{\max}(z \mid x, m) = \eta \frac{1}{z_{small}}$$

Resultierende Mischverteilung



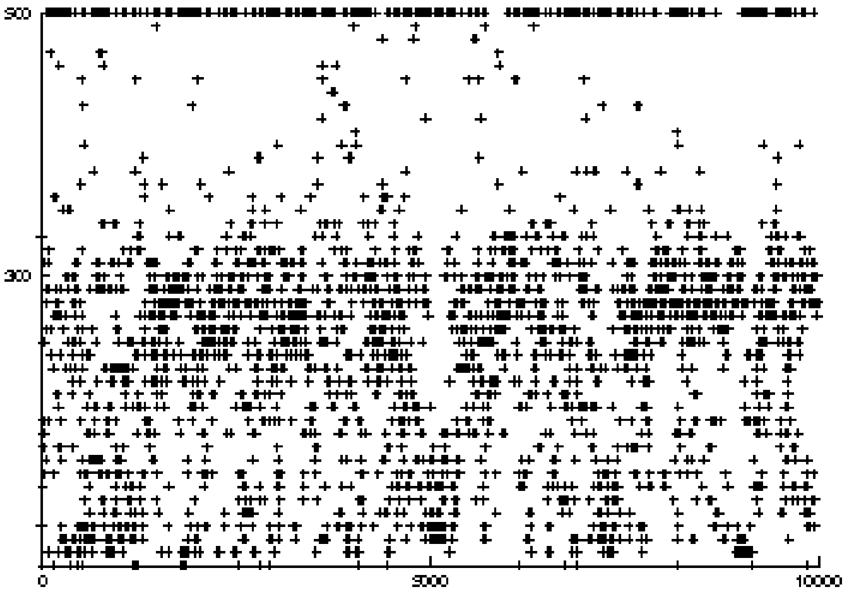
$$P(z \mid x,m) = \begin{pmatrix} \alpha_{\text{hit}} \\ \alpha_{\text{unexp}} \\ \alpha_{\max} \\ \alpha_{\text{rand}} \end{pmatrix}^T \cdot \begin{pmatrix} P_{\text{hit}}(z \mid x,m) \\ P_{\text{unexp}}(z \mid x,m) \\ P_{\max}(z \mid x,m) \\ P_{\text{rand}}(z \mid x,m) \end{pmatrix}$$

Wie ermitteln wir die Modelparameter?

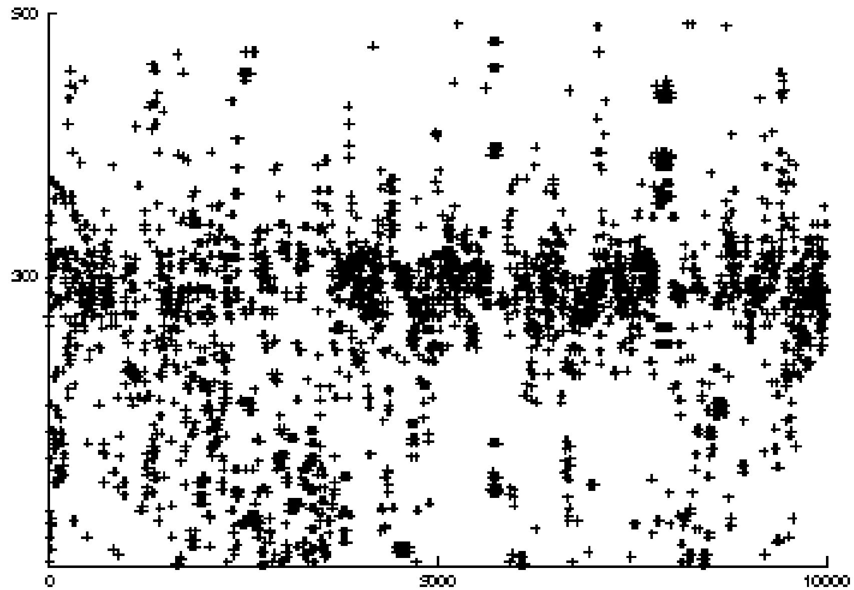
– Engineering von Hand
– Parameterschätzung anhand von Experimenten

Sensorrohdaten

Gemessene Abstände bei einem tatsächlichen Abstand von 3 m.



Sonar / Ultraschall



Laser

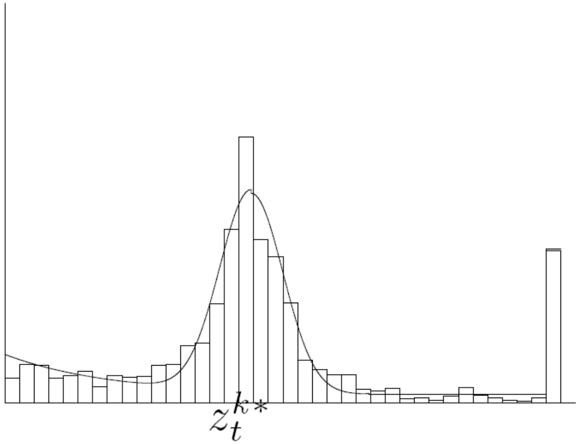
Parameterschätzung

- Maximieren der Data-Likelihood für eine Referenzdatensatz
  - z.B. Roboterpositionen $\boldsymbol{X} = \{x_i\}$
  - korrespondierende Messungen $\boldsymbol{Z}=\{z_i\}$
  - Parametersatz $\boldsymbol{\theta}=\{\alpha_i, \ldots\}$
  - Umweltmodell/Karte $\boldsymbol{m}$

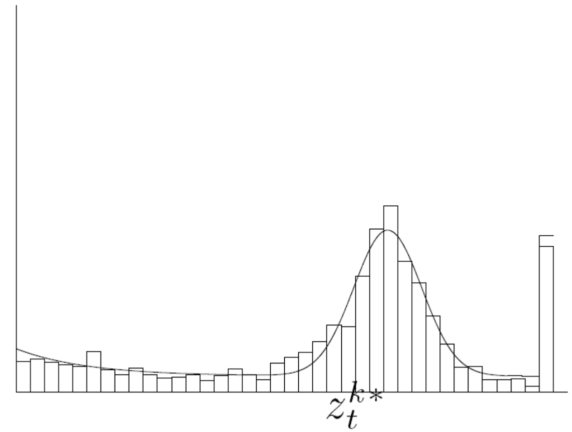
$$\theta^* = \operatorname*{argmax}_{\theta} P(Z|X, m, \theta)$$

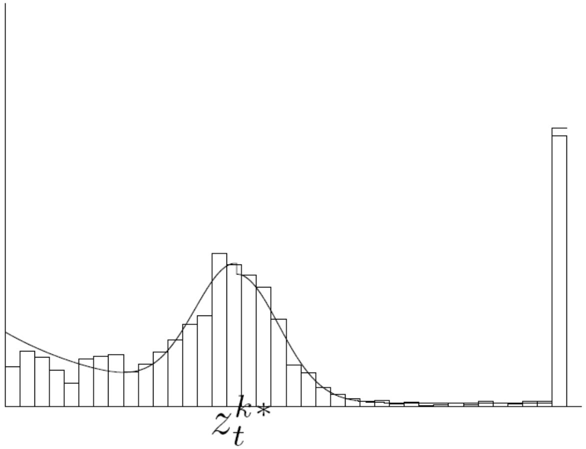
- Optimierung über **θ,** verschiedene Möglichkeiten.
  - Übliche Optimierungsverfahren
  - Wichtig: $\int P(z \mid x, m)\, dz = 1$
  - Möglichkeit: über N-1Parameter optimieren und dann den Nten so berechnet dass das Integral 1 wird.

Beispielhafte Ergebnisse

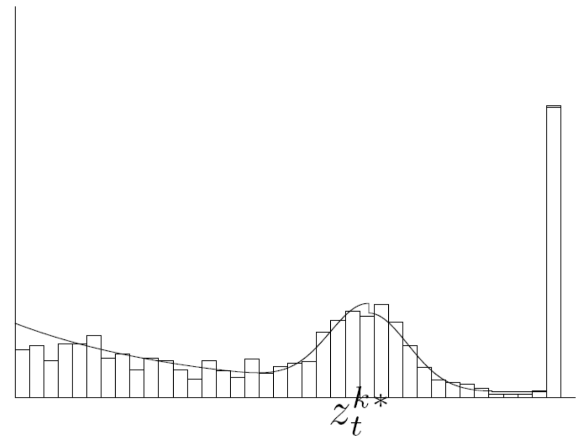


Laser





Sonar

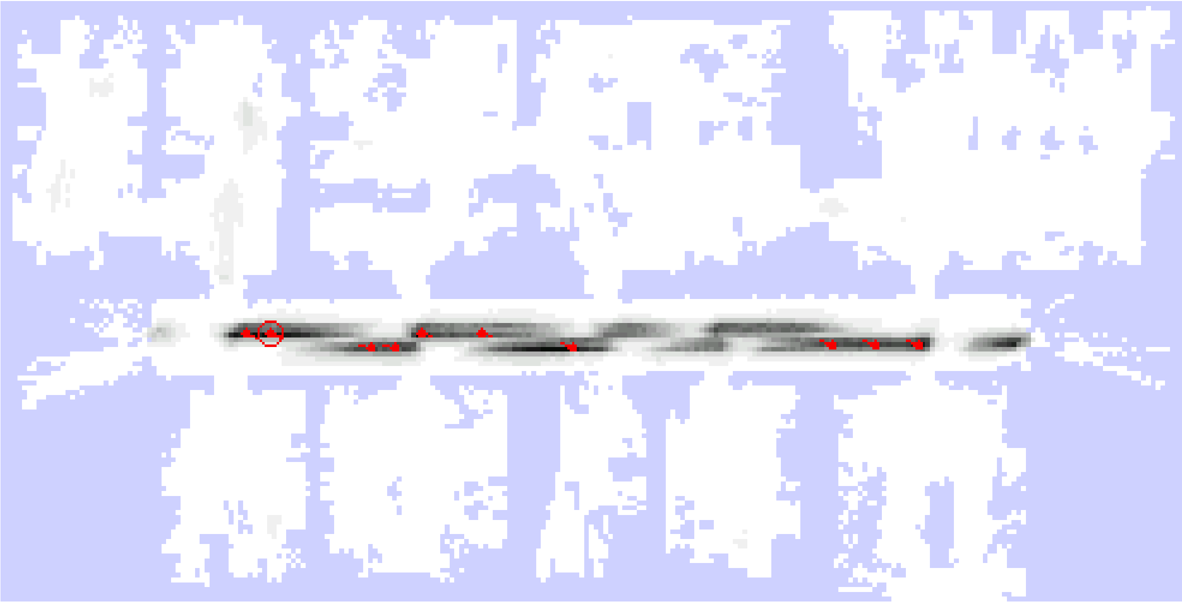


300cm

400cm

Example

z



P(z|x,m)

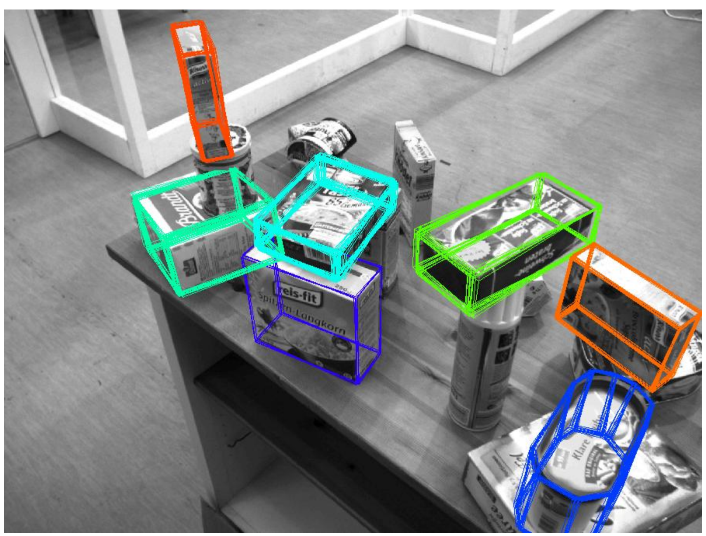
Zwischenfazit

- **Explizite Modelle der Sensorfehler** sind wichtige Voraussetzung für die Robustheit des Gesamtsystems

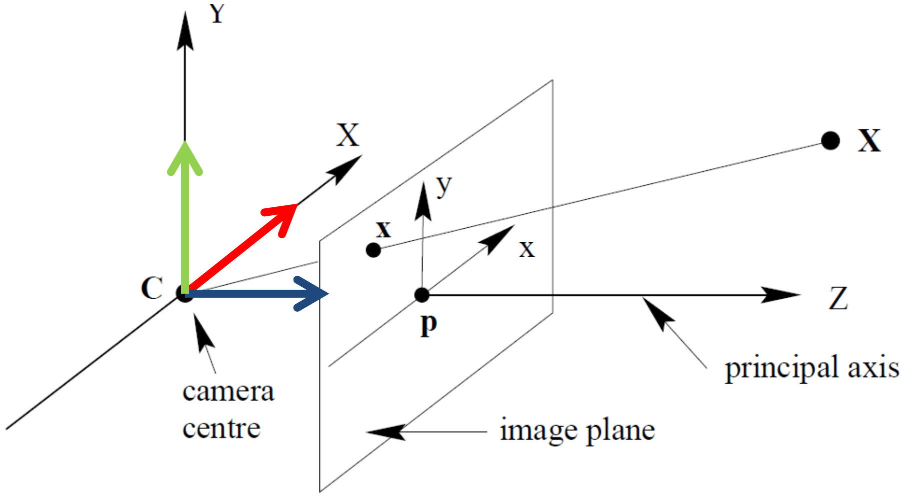
- In vielen Fällen liefert das folgende Vorgehen gute Modelle:
  - Bestimme ein parametrisches Modell der ungestörten Messung
  - Analysiere mögliche Fehlerquellen / mögliches "Rauschen"
  - Hinzufügen angemessener Störungsparameter
  - Bestimmen und verifizieren der Parameter durch Anpassen des Modells an experimentelle Messdaten

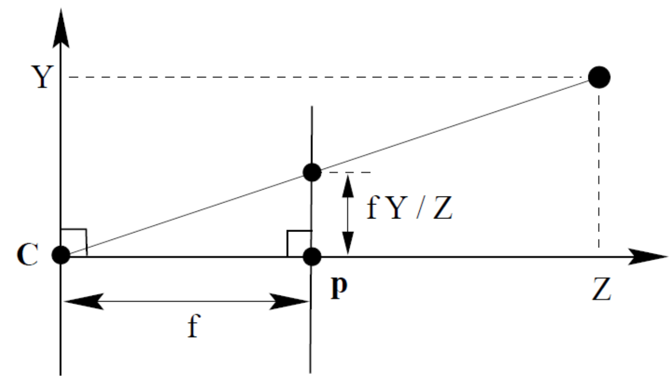
- Es ist extrem wichtig, sich die dem Modell **zugrundeliegenden Annahmen** bewusst zu machen
  - z.B. Unabhängigkeitsannahmen hinsichtlich der Messungen der einzelnen Stahlen. Falls diese nicht gerechtfertigt sind, wird das System zu „selbstsicher"!

Sensormodell für SIFT-basierte Objekt-posenschätzung mit (Stereo-)Kameras



Kameramodell (Lochkamera)





Intrinsische Kameraparameter

- Pixelkoordinaten durch Skalierung und Verschiebung

$$\tilde{\mathbf{x}} = \underbrace{\begin{pmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}}_{\text{intrinsics } K} \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\text{projection}} \tilde{\mathbf{p}}$$

- Brennweite f_x, f_y
- Bildhauptpunkt c_x, c_y
- Schiefe s

Extrinsische Kameraparameter

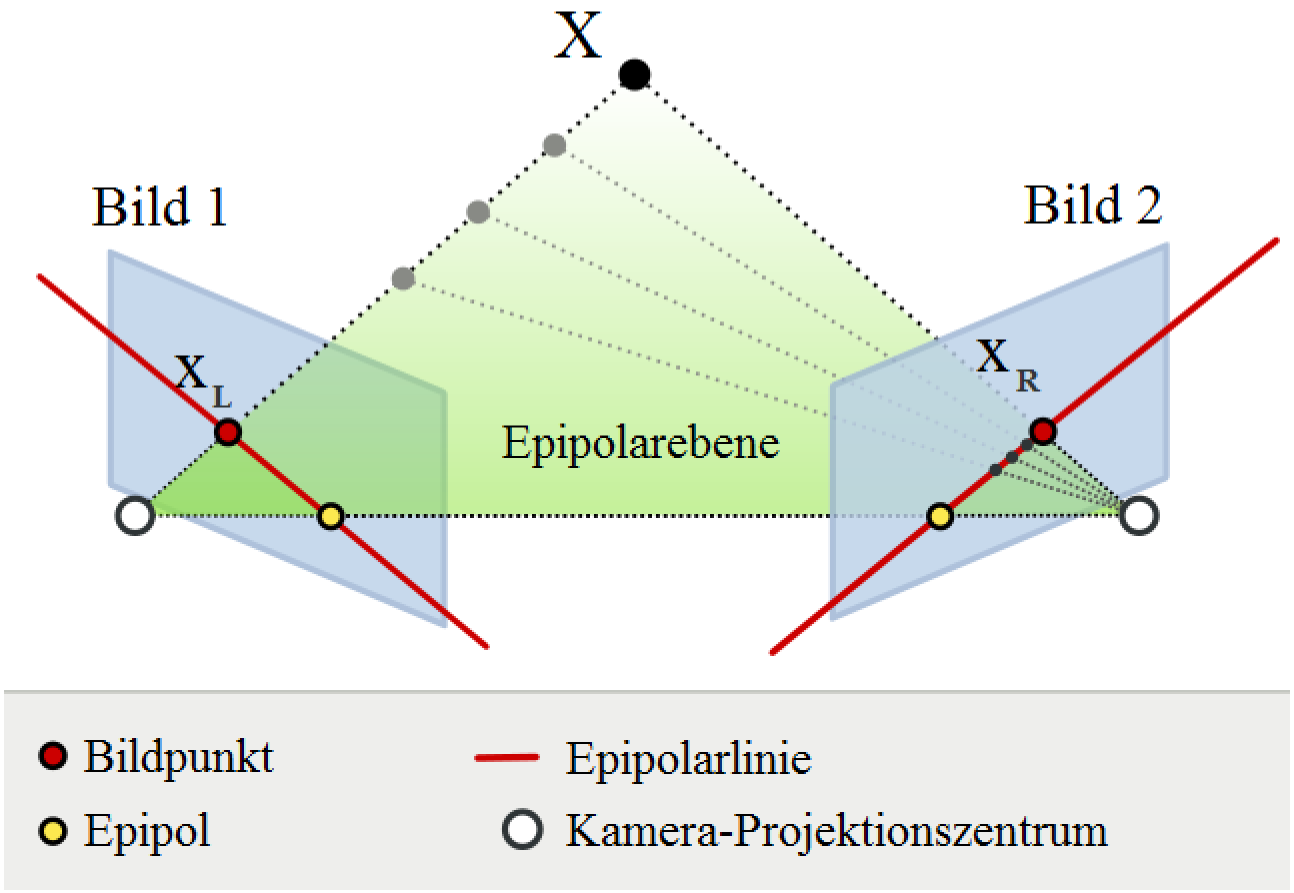
- Für einen Punkt $\tilde{\mathbf{p}}_w$ gegeben in Weltkoordinaten
- Transformation von Welt in Kamerakoordinaten (wird auch als Kamera Extrinsik bezeichnet)

$$\tilde{\mathbf{p}} = \begin{pmatrix} R & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{pmatrix} \tilde{\mathbf{p}}_w$$

- Gesamttransformation inklusive Intrinsik und Projektion

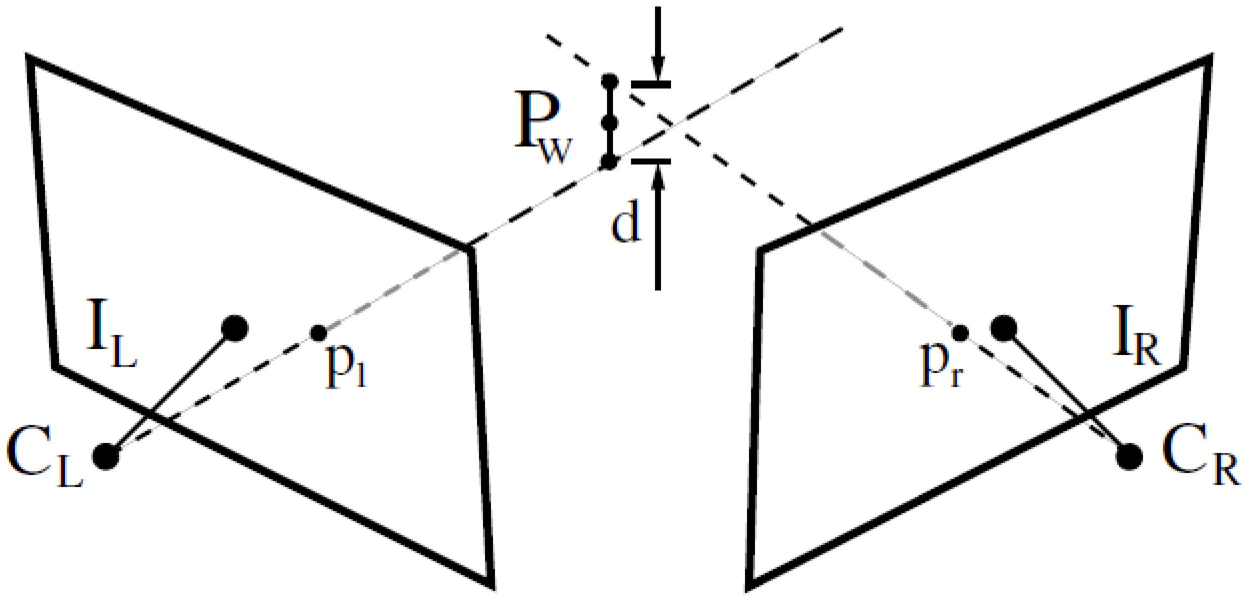
$$\tilde{\mathbf{x}} = \begin{pmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} R & \mathbf{t} \end{pmatrix} \tilde{\mathbf{p}}_w$$

Epipolargeometrie



http://de.wikipedia.org/wiki/Epipolargeometrie

Stereorekonstruktion



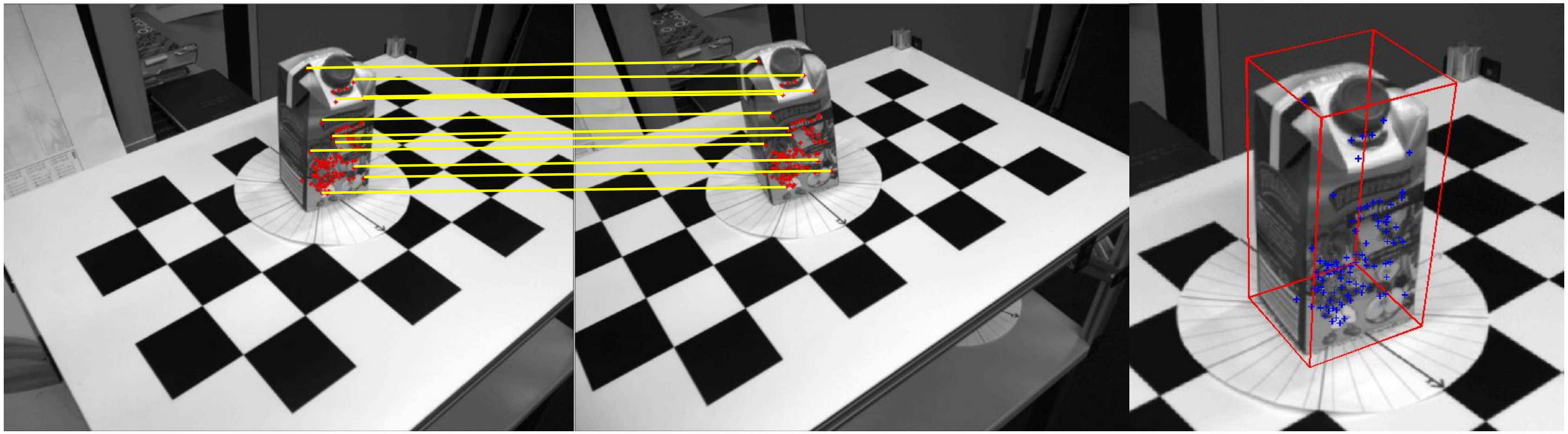
- Weltpunkt $\boldsymbol{P}_w$ wird auf $\boldsymbol{p}_l$ und $\boldsymbol{p}_r$ abgebildet
- Mit den Kamerazentren $\boldsymbol{C}_l$ und $\boldsymbol{C}_r$ ergibt sich jeweils ein Strahl
  - In der Realität werden sich die Strahlen nicht schneiden
- Einfachster Ansatz: $\boldsymbol{P}_w$ ist der Mittelpunkt der gemeinsamen Senkrechten der beiden Stahlen
  - Details und andere (bessere) Ansätze: *Richard I. Hartley and Peter Sturm* „Triangulation“ http://users.cecs.anu.edu.au/~hartley/Papers/triangulation/triangulation.pdf

Beispiel: SIFT-basierte Objektposen-schätzung mit (Stereo-)Kameras

3D-Modell der lokalen (SIFT-)Features
auf dem Objekt

T. Grundmann, R. Eidenberger, M. Schneider, M. Fiegert und G. v. Wichert: *Robust high precision 6D pose determination in complex environments for robotic manipulation*. In: IEEE International Conference on Robotics and Automation, 2010.

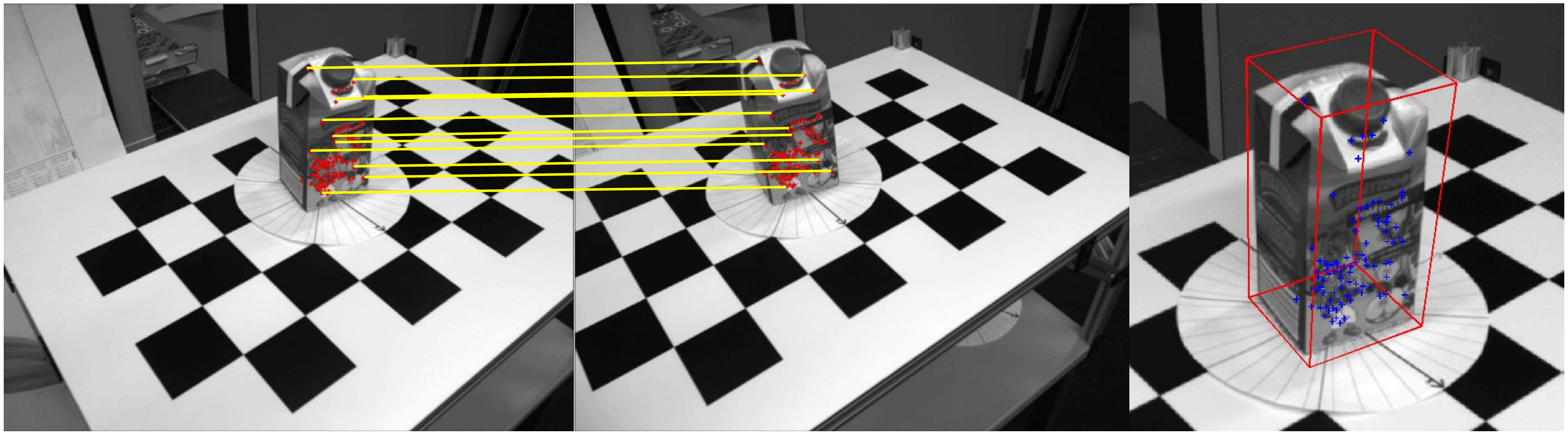
Beispiel: : SIFT-basierte Objektposen-schätzung mit (Stereo-)Kameras



- Extraktion der SIFT Merkmale in beiden Bildern einer kalibrierten Stereokamera
- Zuordnung anhand der Merkmalsvektoren im rechten und linken Bild sowie im Objektmodell
- Rekonstruktion der 3D-Lage: RANSAC, Least Squares Fit
  - Für Sensormodell <u>nicht</u> erforderlich!

T. Grundmann, W. Feiten and G. v. Wichert: *A Gaussian Measurement Model for Local Interest Point Based 6 DOF Pose Estimation*. In: IEEE International Conference on Robotics and Automation, Shanghai, China, 2011.

Beispiel: : SIFT-basierte Objektposen-schätzung mit (Stereo-)Kameras



- Sensormodell: $p(\mathbf{z}|\mathbf{x})$ zur Bewertung von Lagehypothesen

  z: Lage der SIFT-Punkte in beiden Bildern,

  x: Lage des Objekts im Raum

T. Grundmann, W. Feiten and G. v. Wichert: *A Gaussian Measurement Model for Local Interest Point Based 6 DOF Pose Estimation*. In: IEEE International Conference on Robotics and Automation, Shanghai, China, 2011.

Kalibrierung

Exkurs

- Modelle (Sensoren und Aktoren!) enthalten immer Parameter, deren Wert von physikalischen Gegebenheiten abhängt
- In der Regel sind diese Parameter nicht mit ausreichender Genauigkeit bekannt
  - z.B. Variationen im Produktionsprozess
- Beispiel: Kameramodell

$$\tilde{\mathbf{x}} = \begin{pmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} R & \mathbf{t} \end{pmatrix} \tilde{\mathbf{p}}_w$$

Grundsätzliche Vorgehensweise

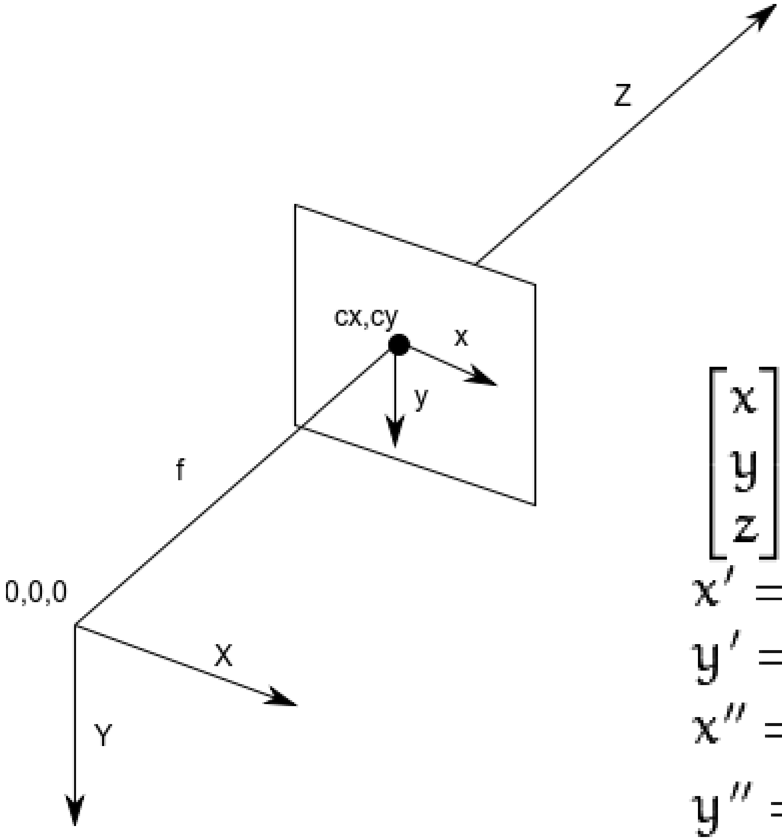
Exkurs

Anpassung der Modellparameter anhand von Referenz-messungen

1. Definition eines geeigneten Meßaufbaus mit bekannten Eigenschaften, der eine „Beobachtung“ der Modellparameter erlaubt
2. Durchführen einer ausreichenden Menge von Messungen
3. Vorhersage der erwarteten Messungen unter Verwendung des zu kalibierenden Modells
4. Optimierung der Modellparameter, z.B.
   - Minimierung Differenz zwischen erwarteten und tatsächlichen Messungen
   - Maximierung der Data Likelihood $P(z \mid x, m)$

Kamerakalibrierung mit OpenCV

Exkurs



$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t$$

$$x' = x/z$$

$$y' = y/z$$

$$x'' = x' \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + 2 p_1 x' y' + p_2 (r^2 + 2 x'^2)$$

$$y'' = y' \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + p_1 (r^2 + 2 y'^2) + 2 p_2 x' y'$$

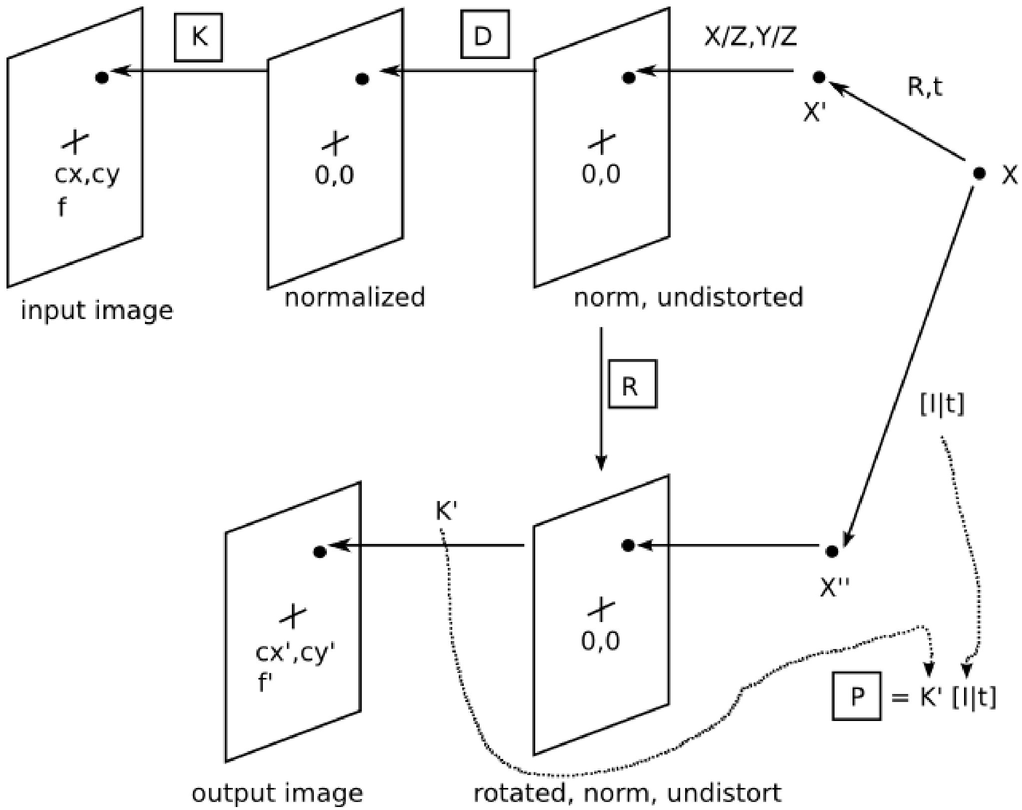
$$\text{where} \quad r^2 = x'^2 + y'^2$$

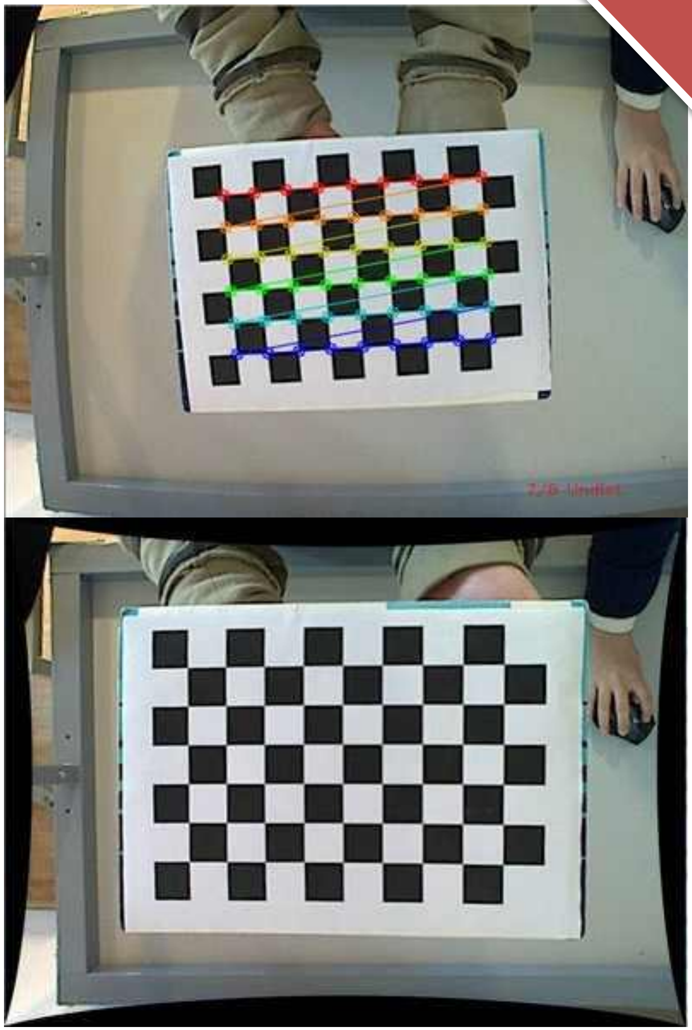
$$u = f_x * x'' + c_x$$

$$v = f_y * y'' + c_y$$

Kamerakalibrierung mit OpenCV

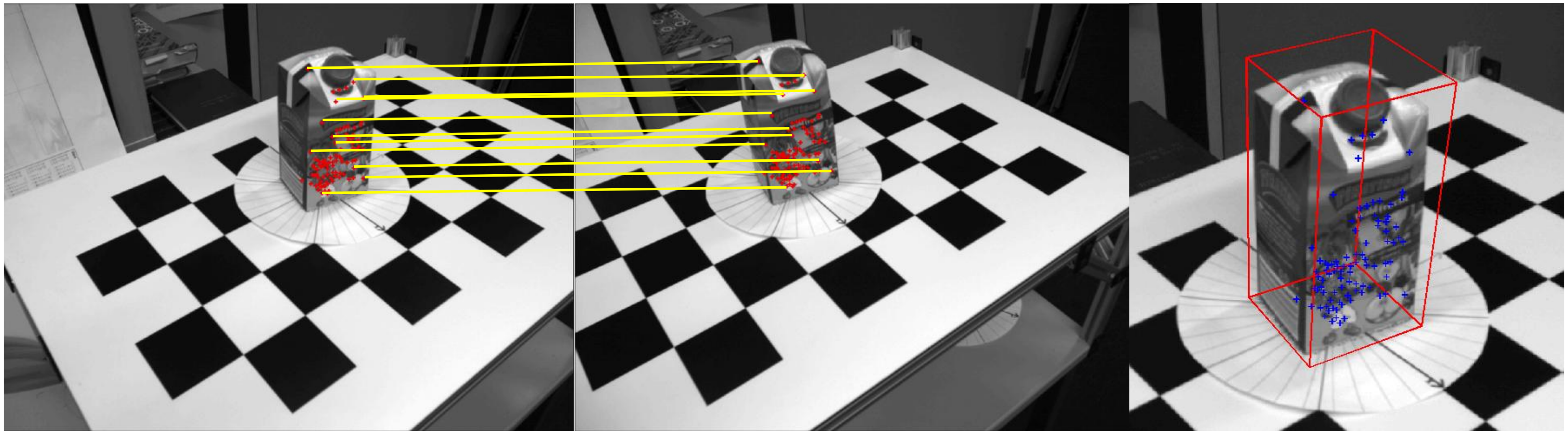
Exkurs





http://opencv.org

Beispiel: : SIFT-basierte Objektposen-schätzung mit (Stereo-)Kameras



- Sensormodell: $p(\mathbf{z}|\mathbf{x})$ zur Bewertung von Lagehypothesen

  z: Lage der SIFT-Punkte in beiden Bildern,

  x: Lage des Objekts im Raum

T. Grundmann, W. Feiten and G. v. Wichert: *A Gaussian Measurement Model for Local Interest Point Based 6 DOF Pose Estimation*. In: IEEE International Conference on Robotics and Automation, Shanghai, China, 2011.

Probabilistisches Sensormodell

Physikalisches Modell des Messvorgangs

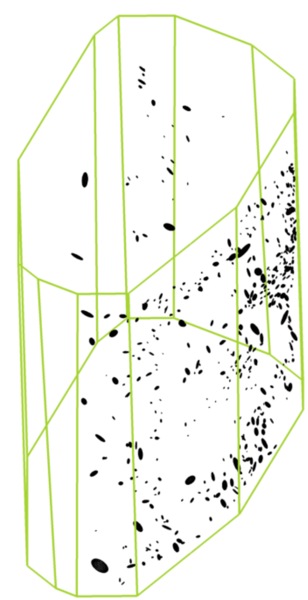
- Punktweise Projektion der SIFT-Merkmalspunkte in die Bilder der Beiden Kameras

$$\tilde{\mathbf{x}} = \begin{pmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} R & \mathbf{t} \end{pmatrix} \tilde{\mathbf{p}}_w$$

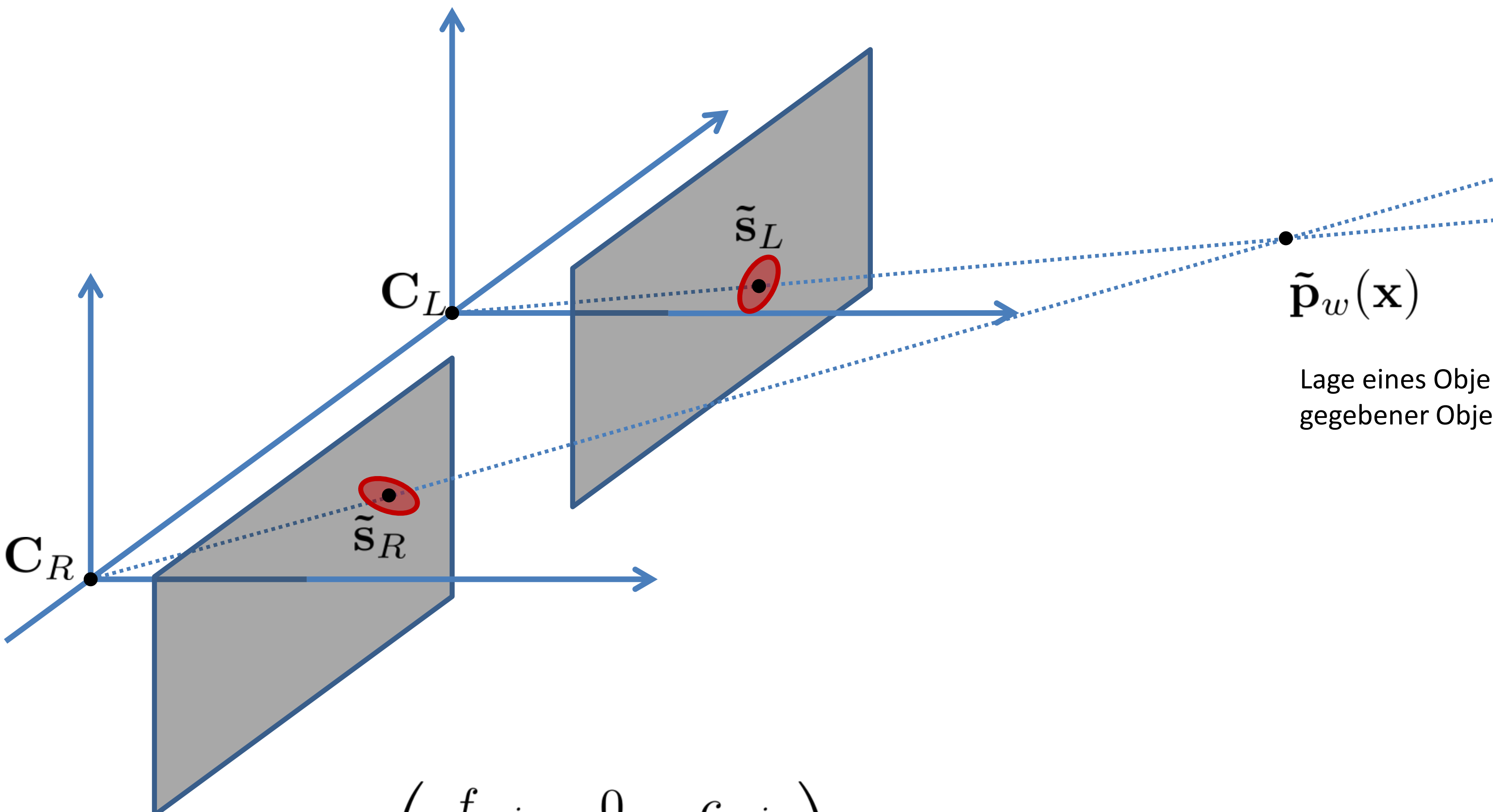
Wesentliche Fehlerquellen

- Fehler bei der Detektion der SIFT-Merkmalspunkte
- (Modellfehler im 3D Objektmodell)

→ müssen im Sensormodell berücksichtigt werden



Zur Herleitung des Messmodells



Lage eines Objektpunktes bei gegebener Objektpose **x**

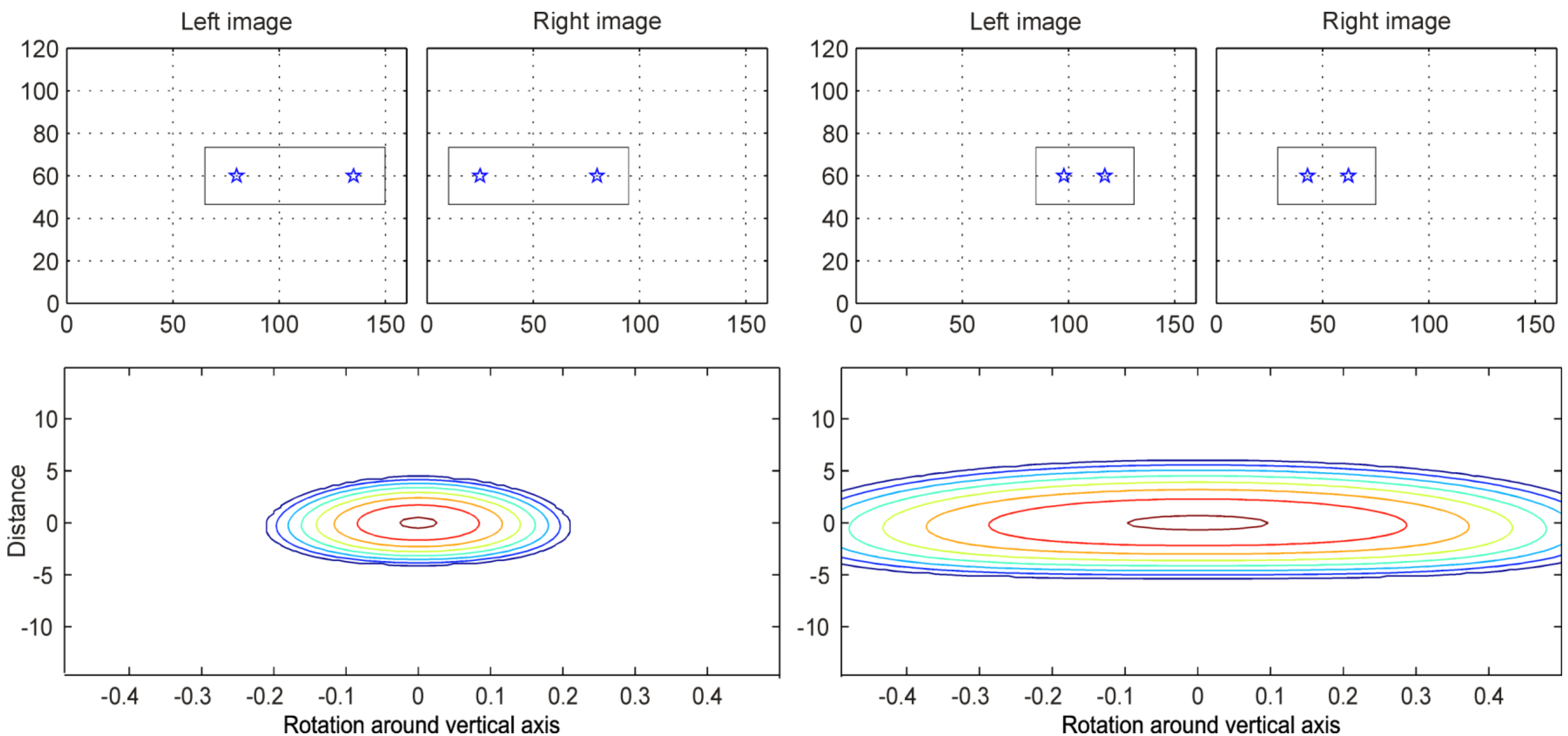
$$\tilde{\mathbf{s}}_\mathbf{i} = \begin{pmatrix} f_{x,i} & 0 & c_{x,i} \\ 0 & f_{y,i} & c_{y,i} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} R_i & \mathbf{t}_\mathbf{i} \end{pmatrix} \tilde{\mathbf{p}}_w + \nu_i \quad \nu_i \sim N(0, Q_i)$$

$$i \in \{L, R\}$$

Normalverteiltes Rauschen mit Kovarianzmatrix Q_i

Beispiel: : SIFT-basierte Objektposen-schätzung mit (Stereo-)Kameras

- Positionsunsicherheit hängt von der Anordnung der Features auf dem Objekt ab!!



Probabilistisches Sensormodell

Observation

Sample from object pose distribution

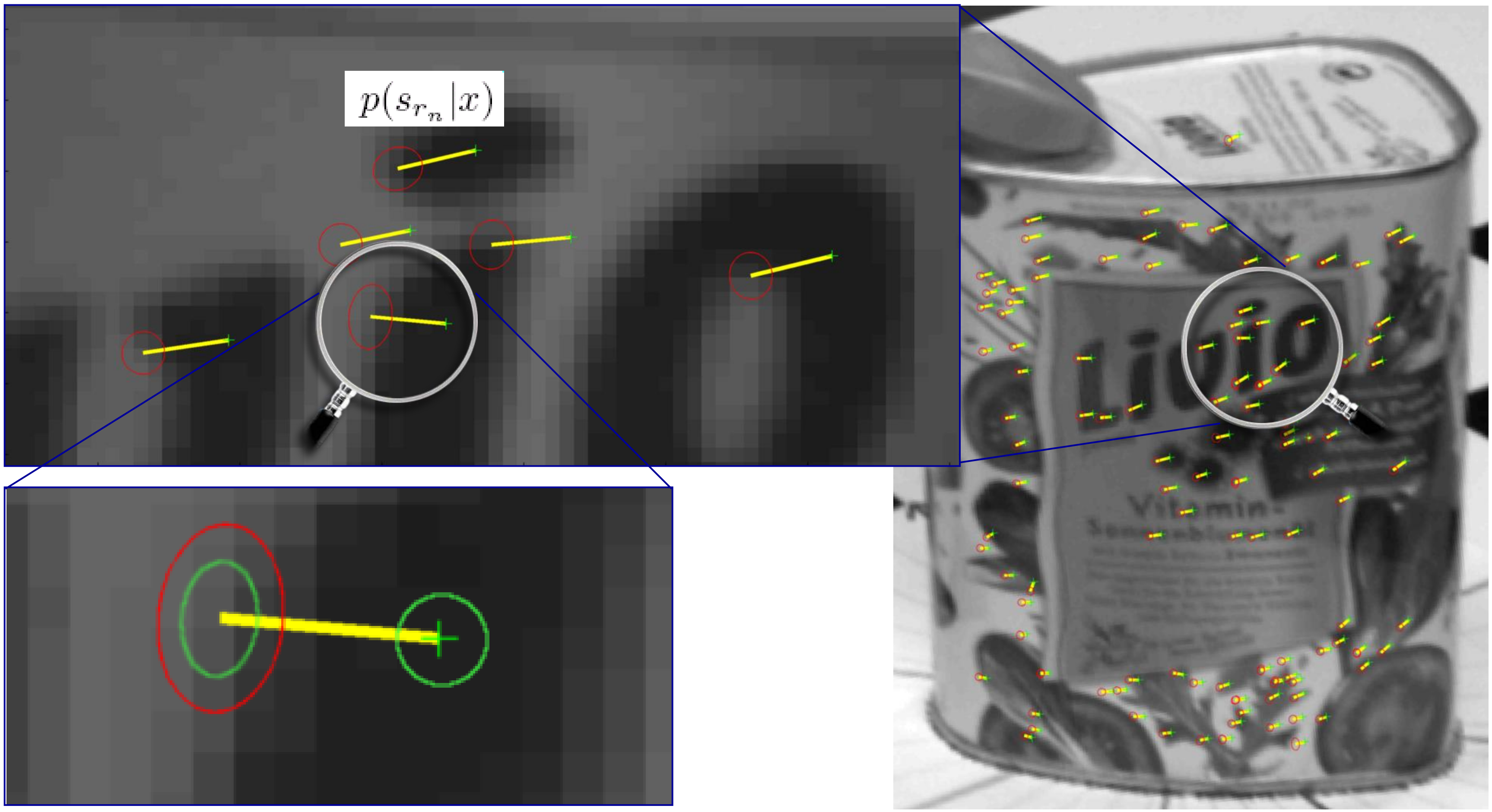
$$p(z|x) = \prod_{1}^{N} p(s_{l_n}|x)p(s_{r_n}|x)$$

Interest points in left and right image

- Projektion der Modellfehler in die jeweilige Bildebene
- Gauß’sche Modellierung des Detektionsfehlers
- Alle Features werden als statistisch unabhängig angenommen

Probabilistisches Sensormodell

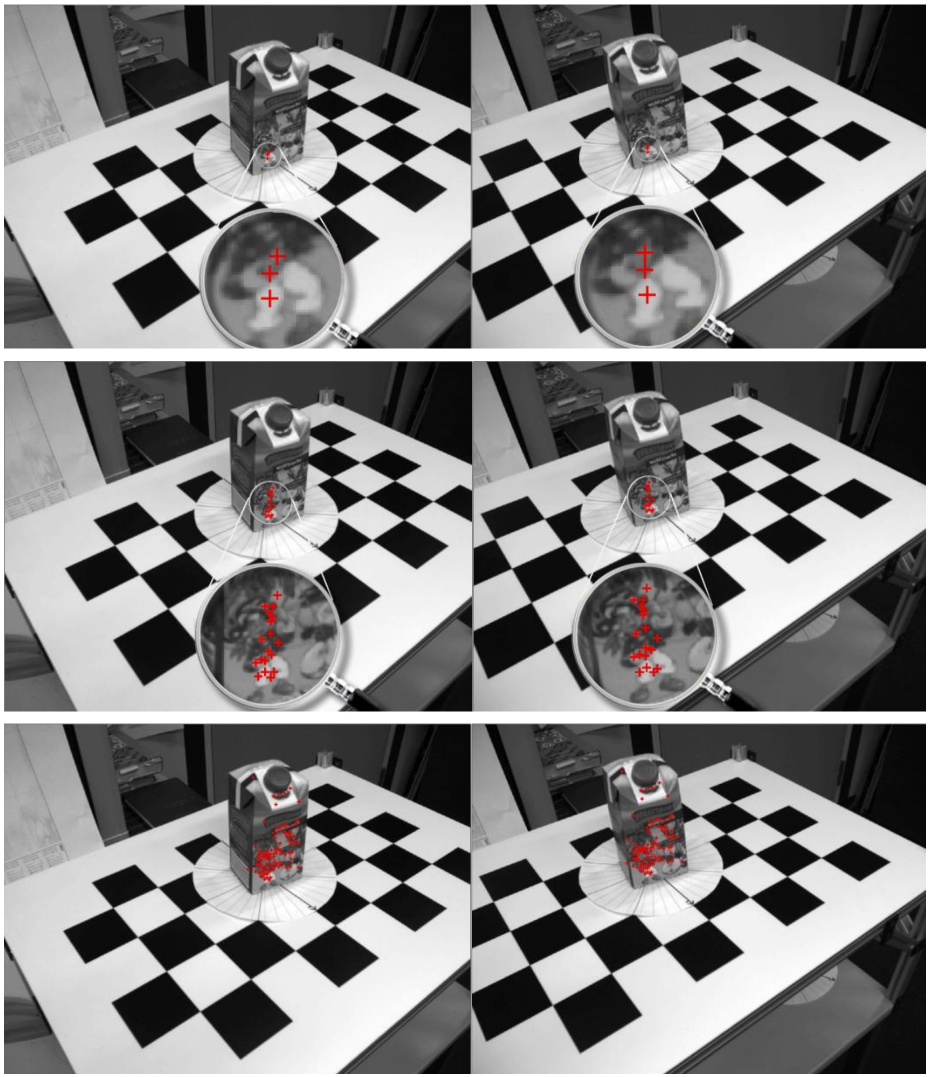
- **Beispiel für eine Teilmessung**

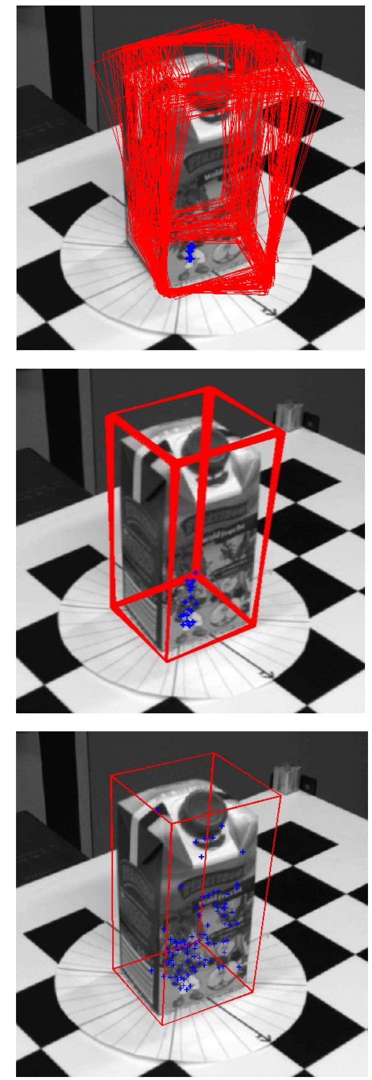


Probabilistisches Sensormodell

- **Vergleich zweier Lagehypothesen**

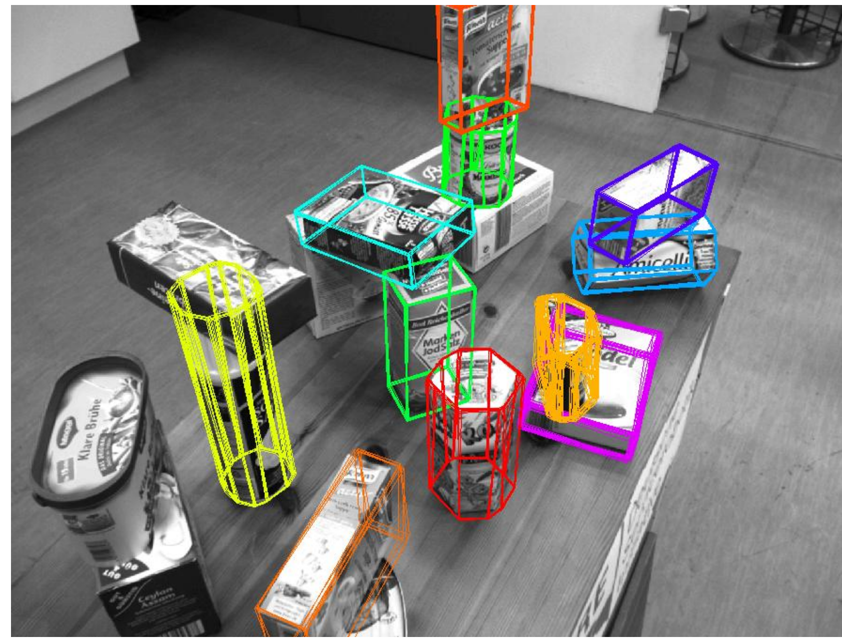
Probabilistisches Sensormodell





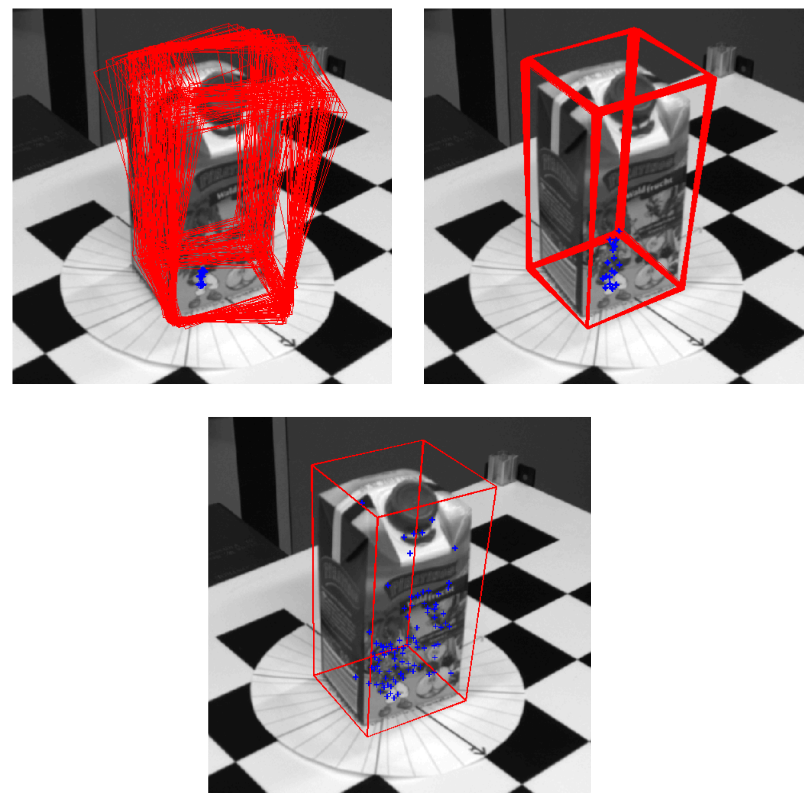
Probabilistisches Sensormodell

- **Ergebnisse für ein ganze Szene**



Probabilistische Szenen-Repräsentation

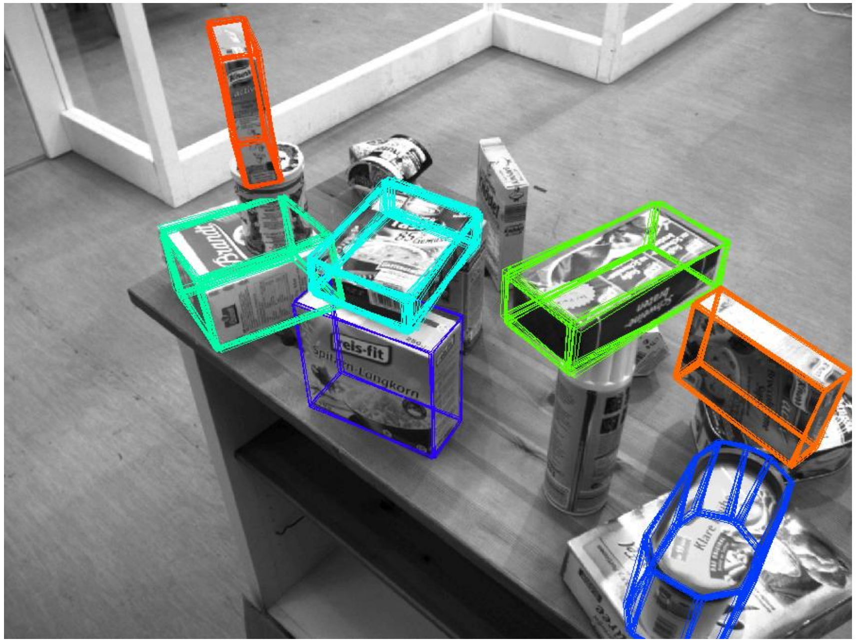
- Explizite Repräsentation der Unsicherheit!



Zusammenfassung

- Probabilistische Modellierung des Sensors
  - Hier Stereokamera in Kombination mit SIFT-Merkmalsextraktion und 3D-Modell der lokalen Merkmale auf der Objektoberfläche
  - Welche Annahmen, die wir dafür gemacht haben, könnten unzutreffend sein?

- Resultat: Unsicherheitsbehaftetes Modell der Szene
  - Hier: 6D-Lage (und Klassifikation) der Objekte



Nächstes Mal

- Aktive Wahrnehmung: Was kann ich als Maschine tun, um meine Wahrnehmungsergebnisse gezielt zu verbessern?