

03-MB-
709.03

Echtzeitbildverarbeitung (13)

Prof. Dr. Udo Frese

Initialisierung

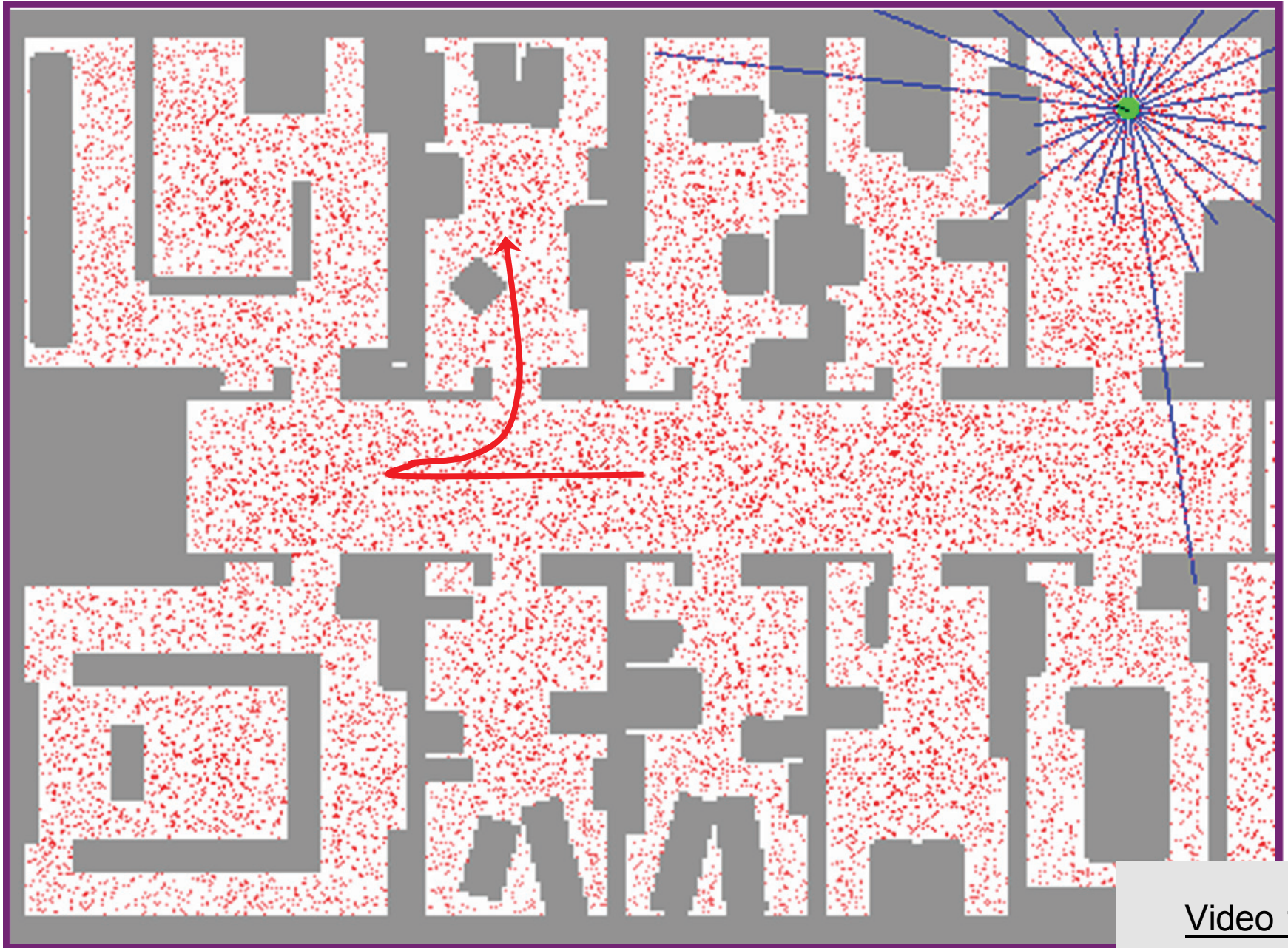
Resampling

Herleitung Partikelfilter

Was bisher geschah

- ▶ **Situation: Dynamisches System mit sich änderndem Zustand X_t und gegebener Messung z_t und Dynamikmessung u_t .**
 - ▶ alle Messungen tragen zur Information über X_t bei, nicht nur z_t .
 - ▶ Markov Annahme: z_t nur abhängig von X_t , u_t von X_t und X_{t+1}
 - ▶ \Rightarrow Verteilung von X_t enthält alle Information von $z_{1..t}$, $u_{1..t-1}$
- ▶ **rekursiver Schätzer berechnet aus Verteilung von X_{t-1} in die Verteilung von X_t .**
- ▶ **Partikelfilter repräsentiert Verteilung durch gewichtete zufällige Stichproben (Partikel) im Zustandsraum.**
- ▶ **Messschritt: Gewichte der Partikel mit $p(Z_t=z_t|X_t=x_t)$**
 - ▶ praktisch für Bildverarbeitung: Gausscher Messfehler
- ▶ **Dynamikschritt: Ziehe aus $p(X_t|X_{t-1}, u_t)$ wobei X_{t-1} aus Partikel**
 - ▶ praktisch: nominales Bewegungsmodell plus Rauschen
- ▶ **Literatur: S. Thrun, W. Burgard, D. Fox, Probabilistic Robotics, (2,4)**

Monte Carlo Lokalisierung mit Ultraschall Sensoren



Initialisierung

Initialisierung

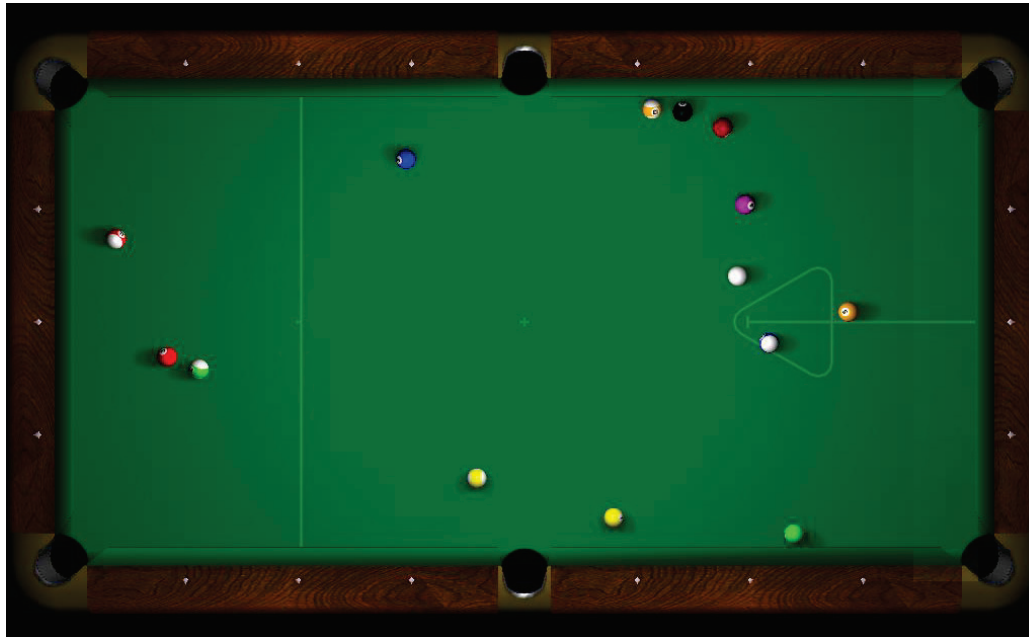
Einfachster Ansatz

- ▶ ziehen aus der a-priori Verteilung
- ▶ gleichverteilte zufällige Partikel
- ▶ Frage an das Auditorium: Wie viele Partikel braucht man im Billard Beispiel um den Zustandsraum halbwegs abzudecken? (ganz grob)



Initialisierung

- ▶ Frage an das Auditorium: Wie viele Partikel braucht man im Billard Beispiel um den Zustandsraum halbwegs abzudecken? (ganz grob)
- ▶ sagen wir: Alle 1cm, bzw. alle 5cm/s
- ▶ also $(3\text{m}/1\text{cm}) * (1.5\text{m}/1\text{cm}) * (5\text{m/s}/1\text{cm/s})^2 = 450 \text{ Mio.}$



Initialisierung

Ziehen aus der a-priori Verteilung

- ▶ **Was passiert?**
 - ▶ ziehen aus $P(X_1)$
 - ▶ gewichten mit $P(z_1|x_1)$
 - ▶ resampling
- ▶ **wenn $P(X_1)$ als a-priori Wissen gleich verteilt, haben die meisten Partikel fast Gewicht 0**
 - ▶ \Rightarrow ineffizient

Initialisierung

Ziehen aus der a-posteriori Verteilung

- ▶ **besser: Partikel mit erster Messung initialisieren.**
 - ▶ ziehen aus $P(X_1|z_1)$
 - ▶ $P(X_1|z_1) \propto P(z_1|X_1)P(X_1)$.
- ▶ **für $P(X_1)$ konstant**
 - ▶ $P(X_1|z_1) \propto P(z_1|X_1)$
 - ▶ $P(z_1|X_1)$ als Verteilung von X_1 für festes z_1 sehen
 - ▶ daraus ziehen
- ▶ **für $P(X_1)$ nicht konstant**
 - ▶ mit $P(X_1)$ gewichten
 - ▶ besonders unmögliche ($P(X_1)=0$) verwerfen

Initialisierung

Ziehen aus der a-posteriori Verteilung

- ▶ **Zustand und Messung gleiche Dimension und Nominalfunktion plus Rauschen**
 - ▶ $z=f(x)+n$ mit invertierbarem f
 - ▶ Rauschwerte n ziehen und von Messung z abziehen.
 - ▶ Zustand als $f^{-1}(z-n)$ ausrechnen.
- ▶ **Zustand höhere Dimension als Messung**
 - ▶ mehrere Messungen für Initialisierung nötig
 - ▶ so viele, wie nötig nehmen,
 - ▶ verrauschen
 - ▶ daraus Zustand ausrechnen

Initialisierung

Frage an das Auditorium: Wie initialisiert man den Partikelfilter für das Billard Problem?

- ▶ **Zustand: $\mathbf{x} = (p_x, p_y, v_x, v_y)$**
 - ▶ Position [m] (p_x, p_y),
 - ▶ Geschwindigkeit (v_x, v_y). [m/s]
- ▶ **Messung: $\mathbf{z} = (i_x, i_y)$ [Pixel] Position der Kugel im Kamerabild.**

$$\begin{pmatrix} i_x \\ i_y \end{pmatrix} = f(p_x, p_y, v_x, v_y) = p_{center} + \frac{\alpha}{Z} \begin{pmatrix} p_x \\ p_y \end{pmatrix}$$

Initialisierung

Frage an das Auditorium: Wie initialisiert man den Partikelfilter für das Billard Problem?

- ▶ **Zustand: $\mathbf{x} = (p_x, p_y, v_x, v_y)$**
 - ▶ Position [m] (p_x, p_y),
 - ▶ Geschwindigkeit (v_x, v_y). [m/s]
- ▶ **Messung: $\mathbf{z} = (i_x, i_y)$ [Pixel] Position der Kugel im Kamerabild.**
- ▶ **2 Messungen definieren den Zustand:**
 - ▶ 2. Messung definiert die Position (p_x, p_y)
 - ▶ 1.&2. definieren Geschwindigkeit (v_x, v_y)
- ▶ **Vorgehen:**
 - ▶ ersten zwei Messungen speichern
 - ▶ Rauschen addieren
 - ▶ Zustand berechnen (siehe rechts)
 - ▶ nicht gewichten!
 - ▶ danach wie gewohnt weiter

$$p_x = \frac{Z}{\alpha} (i_{x2} - p_{centerx})$$

$$p_y = \frac{Z}{\alpha} (i_{y2} - p_{centery})$$

$$v_x = \frac{1}{\Delta t} \left(p_x - \frac{Z}{\alpha} (i_{x1} - p_{centerx}) \right)$$

$$v_y = \frac{1}{\Delta t} \left(p_y - \frac{Z}{\alpha} (i_{y1} - p_{centery}) \right)$$

Resampling

**Quelle einiger der folgenden Folien (modifiziert) mit freundlicher Genehmigung von Wolfram Burgard, Introduction to Mobile Robotics, Lecture 9, Universität Freiburg, 2005
<http://ais.informatik.uni-freiburg.de/lehre/ss05/robotics/>**

Resampling

Aufgabenstellung

- ▶ **gegeben: $S = \{(x_i, w_i) | i = 1..n\}$ Partikel x_i mit Gewicht w_i .**
- ▶ **gesucht : n zufällige Stichproben, wobei die Wahrscheinlichkeit x_i zu ziehen proportional w_i ist.**
- ▶ **naïve, aber nicht gute Lösung: n mal unabhängig zufällig aus der Partikelmenge ziehen mit einer Wahrscheinlichkeit die den Gewichten entspricht.**

Resampling

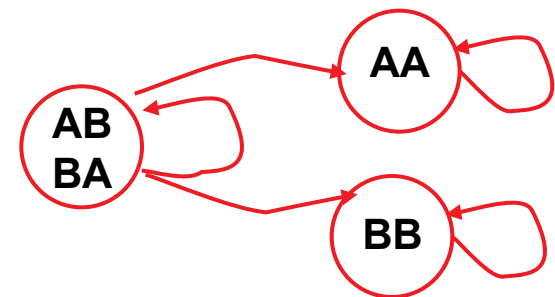
Naïve Lösung

- ▶ **n mal unabhängig zufällig aus der Partikelmenge ziehen**
- ▶ **Frage an das Auditorium: Wo liegt das Problem?**
 - ▶ zwei Zustände A, B
 - ▶ zwei Partikel ($w_1=w_2$)
 - ▶ 1000 mal resampling
 - ▶ was passiert?

Resampling

Naïve Lösung

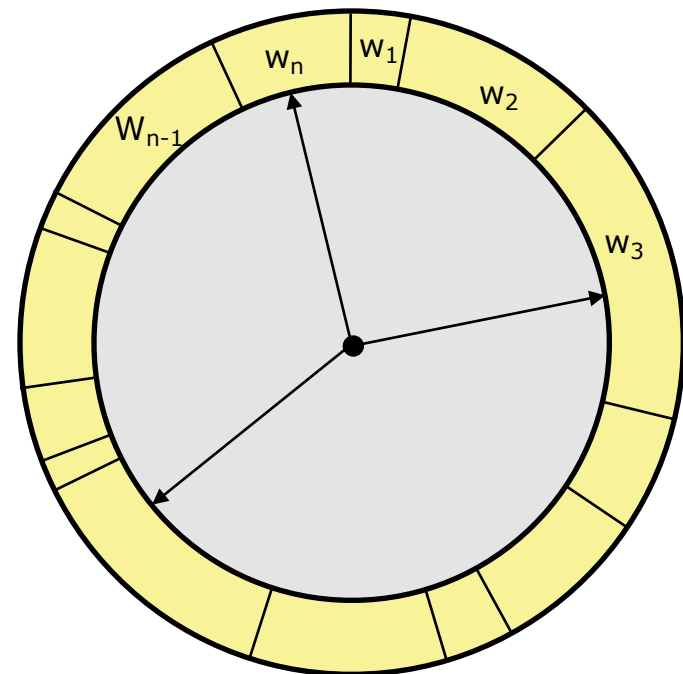
- ▶ **n mal unabhängig zufällig aus der Partikelmenge ziehen**
- ▶ **Frage an das Auditorium: Wo liegt das Problem?**
 - ▶ zwei Zustände A, B
 - ▶ zwei Partikel ($w_1=w_2$)
 - ▶ 1000 mal resampling
 - ▶ was passiert?
- ▶ **einer der Zustände geht verloren**
- ▶ **sehr wahrscheinlich (50% je Resampling)**
- ▶ **Partikelfilter glaubt danach fälschlicherweise, er kenne den Zustand**



Resampling

Naïve Lösung

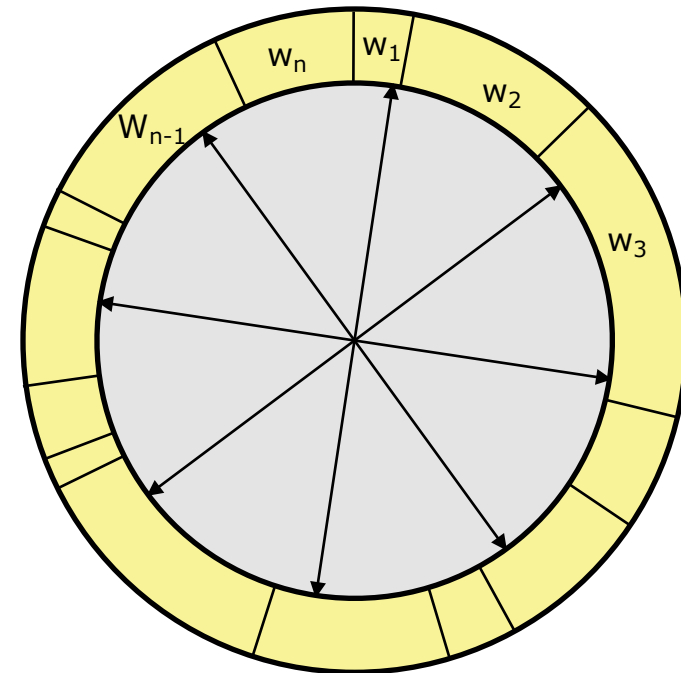
- ▶ **unabhängiges Resampling**
 - ▶ wie Glücksrad
 - ▶ mit Feldgrößen entsprechend Partikelgewicht w_i
 - ▶ n -mal drehen



Resampling

Systematisches Resampling

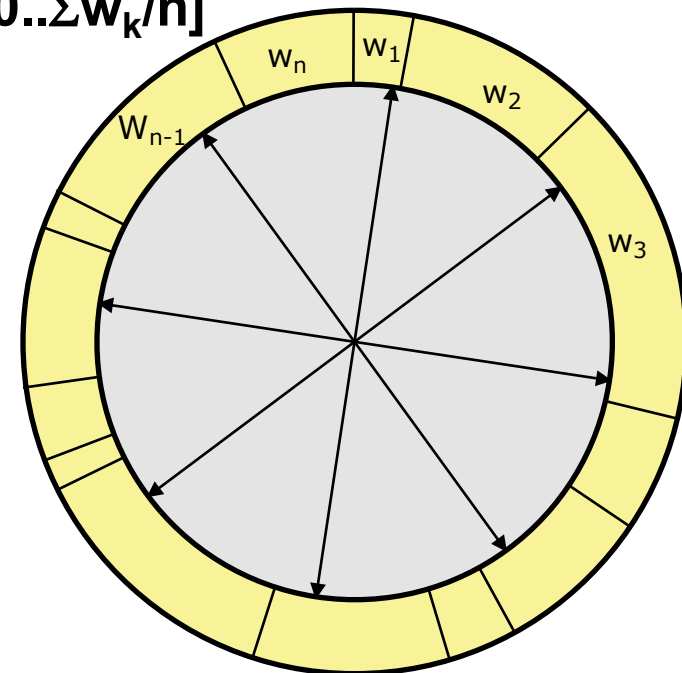
- ▶ **unabhängiges Resampling**
 - ▶ wie Glücksrad
 - ▶ mit Feldgrößen entsprechend Partikelgewicht w_i
 - ▶ n-mal drehen
- ▶ **systematisches Resampling**
 - ▶ einmal Drehen mit n Zeigern



Resampling

Systematisches Resampling (Algorithmus)

- ▶ Winkel des Glücksrades darstellen als Zahl $0.. \sum w_k$.
- ▶ Glückradszeiger im Abstand $\sum w_k / n$
- ▶ erster Glücksradzeiger ist Zufallszahl $\alpha_0 \in [0.. \sum w_k / n]$
- ▶ nächsten Zeiger als $\alpha_i = \alpha_0 + i \sum w_k / n$
- ▶ Partikel (Feld) $j(i)$ zu Zeiger α_i bestimmen
 - ▶ erstes mit Gewichtssumme $> \alpha_i$
 - ▶ $j(i) = \min \{j \mid \alpha_i < \sum_{k=0..j} w_k\}$
- ▶ wiederholen für $i=0..n-1$



Resampling

```
resample (vector<Particle>& p, int n) { // Systematic Resampling of n particles
    totalWeight=0;
    for (int i=0; i<p.size(); i++) totalWeight += p[i].weight;
    vector<Particle> pNew; weightUpToJ = 0; j = -1;
    weightChosen = draw from [0..totalWeight/n); // „1. Glücksradzeiger“
    for (int i=0; i<n; i++) { // Für jeden neuen Partikel i
        while (weightChosen>=weightUpToJ) {
            j++;
            weightUpToJ += p[j].weight;
        }
        pNew.push_back (p[j]); // ist der neue Partikel
        pNew.back().weight = 1.0/n;
        weightChosen += totalWeight/n; // „Nächster Glücksradzeiger“
    }
    p = pNew;
}
```

Herleitung Partikelfilter

Herleitung Partikelfilter

- ▶ **Bayes Formel:**

$$p(X = x|Y = y)p(Y = y) = p(X = x, Y = y) = p(Y = y|X = x)p(X = x)$$

$$p(X = x|Y = y) = \frac{p(Y = y|X = x)p(X = x)}{p(Y = y)} \stackrel{y \text{ fest}}{\propto} p(Y = y|X = x)p(X = x)$$

- ▶ **Bayes mit Vorbedingung z:**

$$p(X = x|Y = y, Z = z) = \frac{p(Y = y|X = x, Z = z)p(X = x|Z = z)}{p(Y = y|Z = z)}$$

$$\stackrel{y \text{ fest}}{\propto} p(Y = y|X = x, Z = z)p(X = x|Z = z)$$

Herleitung Partikelfilter

- ▶ **Gesetz der Gesamtwahrscheinlichkeit**

$$\int p(Z = z)dz = 1,$$

$$p(X = x) = \int p(X = x, Z = z)dz = \int p(X = x|Z = z)p(Z = z)dz$$

$$p(X = x|Y = y) = \int p(X = x|Y = y, Z = z)p(Z = z|Y = y)dz$$

Herleitung Partikelfilter

Bayes Filter

▶ Notation:

- ▶ Grossbuchstaben Zufallsvariablen, Kleinbuchstaben Zahlen (bekannt)
- ▶ Zustand zur Zeit t : X_t
- ▶ Messung zur Zeit t : $Z_t=z_t$
- ▶ Dynamikmessung $t \rightarrow t+1$: $U_t=u_t$

▶ repräsentiert und aktualisiert die Wahrscheinlichkeitsverteilung des aktuellen Zustandes gegeben alle bisherigen Messungen.

$$p(X_t=x_t | Z_t=z_t, U_{t-1}=u_{t-1}, Z_{t-1}=z_{t-1}, \dots, Z_1=z_1, U_0=u_0, Z_0=z_0)$$

▶ Verteilung heißt Belief, weil sie im System gespeichert das repräsentiert was das System über die Umwelt glaubt.

▶ Markov Annahme:

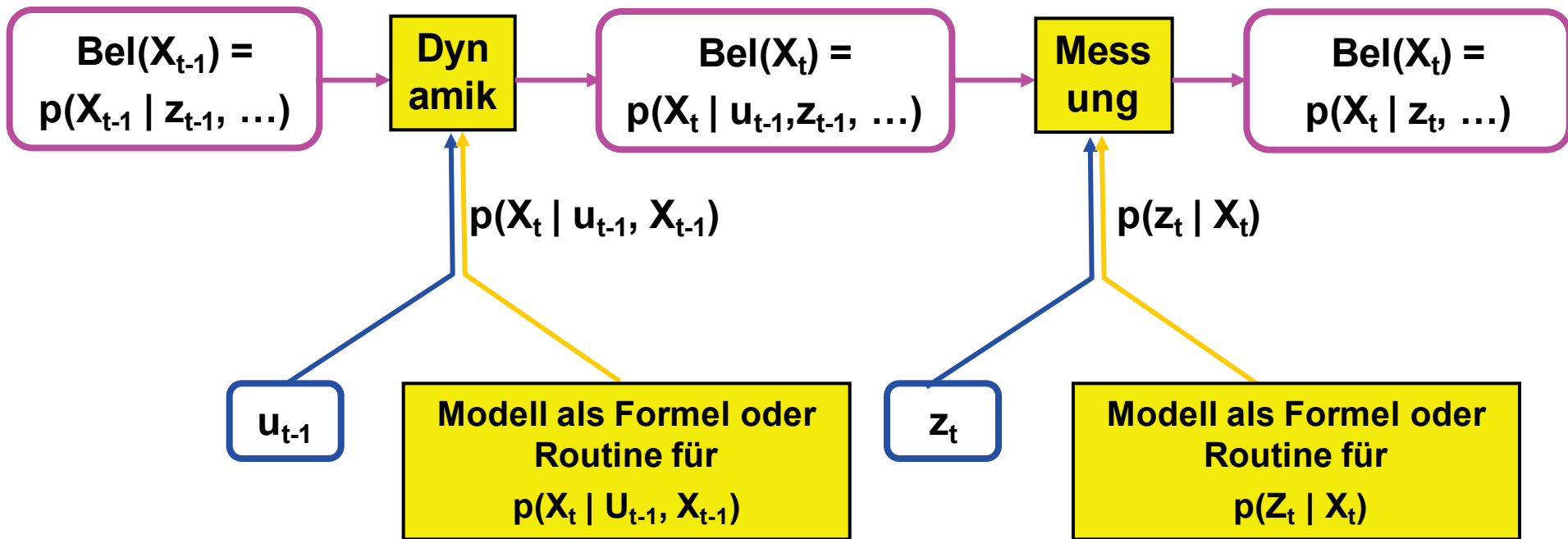
- ▶ (1) Messung beruht nur auf Zustand: $p(Z_t=z_t | X_t=x_t, \dots) = p(Z_t=z_t | X_t=x_t)$
- ▶ (2) Zustand beruht nur auf Dynamikmessung und Vorgänger:
 $p(X_t=x_t | U_{t-1}=u_{t-1}, X_{t-1}=x_{t-1}, \dots) = p(X_t=x_t | U_{t-1}=u_{t-1}, X_{t-1}=x_{t-1})$

Herleitung Partikelfilter

- ▶ **abgekürzte Notation $p(x) = p(X=x)$, $p(x|z) = p(X=x|Z=z)$, etc.**
 - ▶ jeweils Wahrscheinlichkeits(-dichte) eines konkreten Wertes x , z , ..
 - ▶ als Wahrscheinlichkeits(-dichte) dass dazugehörige Zufallsvariable diesen Wert hat.

Herleitung Partikelfilter

Bayes Filter: Abstrakter Rahmen für Partikel (u.a. Kalman) Filter



→ Wahrscheinlichkeitsverteilung im Rechner
 → Daten
 → Modelle, d.h. Programm

Herleitung Partikelfilter

Bayes Filter

$$p(x_t | z_t, u_{t-1}, z_{t-1}, \dots)$$

Bayes \rightarrow

$$\stackrel{=}{=} \frac{p(z_t | x_t, u_{t-1}, z_{t-1}, \dots) p(x_t | u_{t-1}, z_{t-1}, \dots)}{p(z_t | u_{t-1}, z_{t-1}, \dots)}$$

z_t konstant \rightarrow

$$\propto p(z_t | x_t, u_{t-1}, z_{t-1}, \dots) p(x_t | u_{t-1}, z_{t-1}, \dots)$$

Markov (1) \rightarrow

$$= p(z_t | x_t) p(x_t | u_{t-1}, z_{t-1}, \dots)$$

Gesamtwahrscheinlichkeit \rightarrow

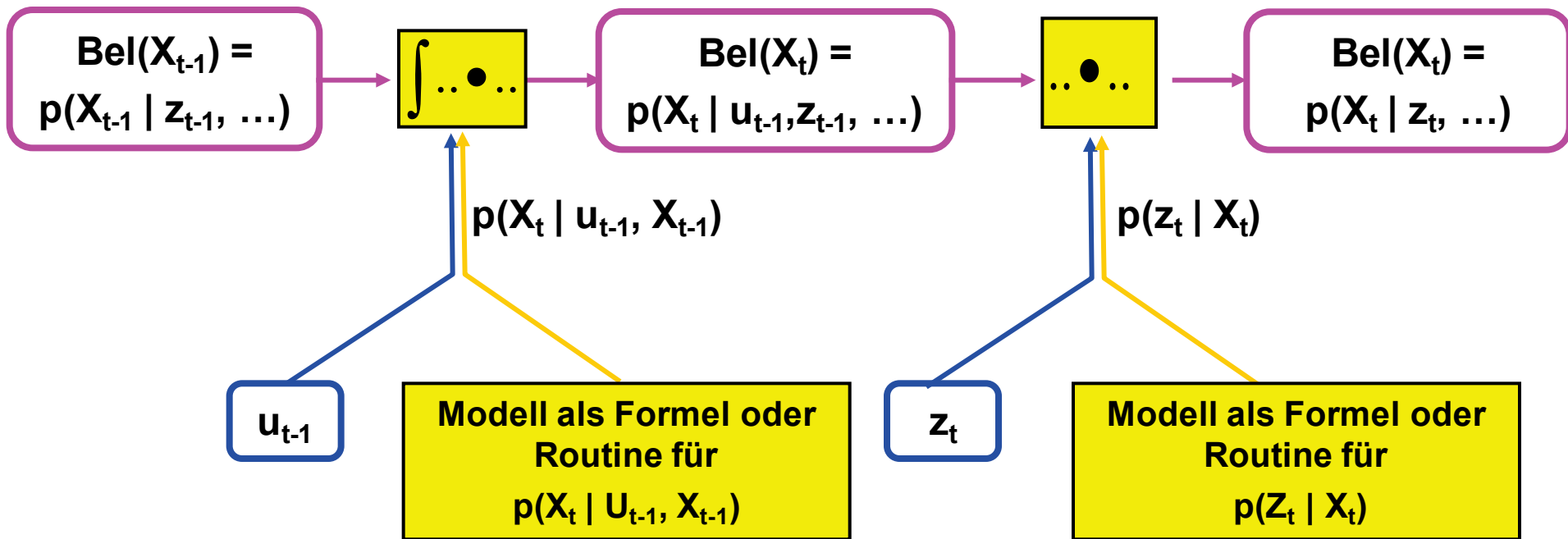
$$= p(z_t | x_t) \int p(x_t | x_{t-1}, u_{t-1}, z_{t-1}, \dots) p(x_{t-1} | u_{t-1}, z_{t-1}, \dots) dx_{t-1}$$

Markov (2) \rightarrow

$$= p(z_t | x_t) \int p(x_t | u_{t-1}, x_{t-1}) p(x_{t-1} | z_{t-1}, \dots) dx_{t-1}$$

Herleitung Partikelfilter

Bayes Filter: Abstrakter Rahmen für Particle (u.a. Kalman) Filter



Herleitung Partikelfilter

Warum „Partikel“?

- ▶ **Verschiedene Bayes Filter unterscheiden sich in der Darstellung der Wahrscheinlichkeitsverteilung $Bel(X_t) = p(X_t | z_t, u_{t-1}, z_{t-1}, \dots, z_1, u_0, z_0)$**
 - ▶ Partikel Filter: *gewichteter* Menge von *Stichproben* (Particles, Samples)
 - ▶ Kalman Filter: Mittelwert, Kovarianz (siehe Theorie der Sensorfusion)
 - ▶ Hidden Markov Models: Tabellierte diskrete Verteilung
 - ▶ Dynamische Bayes Netzwerke
- ▶ **Idee beim Partikel Filter:**
 - ▶ Wahrscheinlichkeitsverteilung durch *gewichtete* Menge von *Stichproben* (Particles, Samples) darstellen.
 - ▶ Einsatz von Zufallszahlen beim Ziehen der Stichproben (Monte Carlo Methode)
- ▶ **Notation nimmt an, dass Messungen und Zustandsübergänge abwechselnd erfolgen. Das ist aber nicht nötig.**

Herleitung Partikelfilter

Was heißt aus $p(X_t | z_t, u_{t-1}, z_{t-1}, \dots, z_1, u_0, z_0)$ ziehen?

- ▶ **größter Teil von X_t hat sehr kleine Wahrscheinlichkeit.**
 - ▶ daher finden, wo $p(X_t | \dots)$ groß ist.
- ▶ **nicht wie ein Zufallszahlen-generator.**
- ▶ **wie Funktionsmaximierung mit zusätzlichem Zufall**
- ▶ **verwandte Operationen:**
 - ▶ Funktionsmaximierung
 - ▶ Gleichung lösen
 - ▶ aus Verteilung ziehen

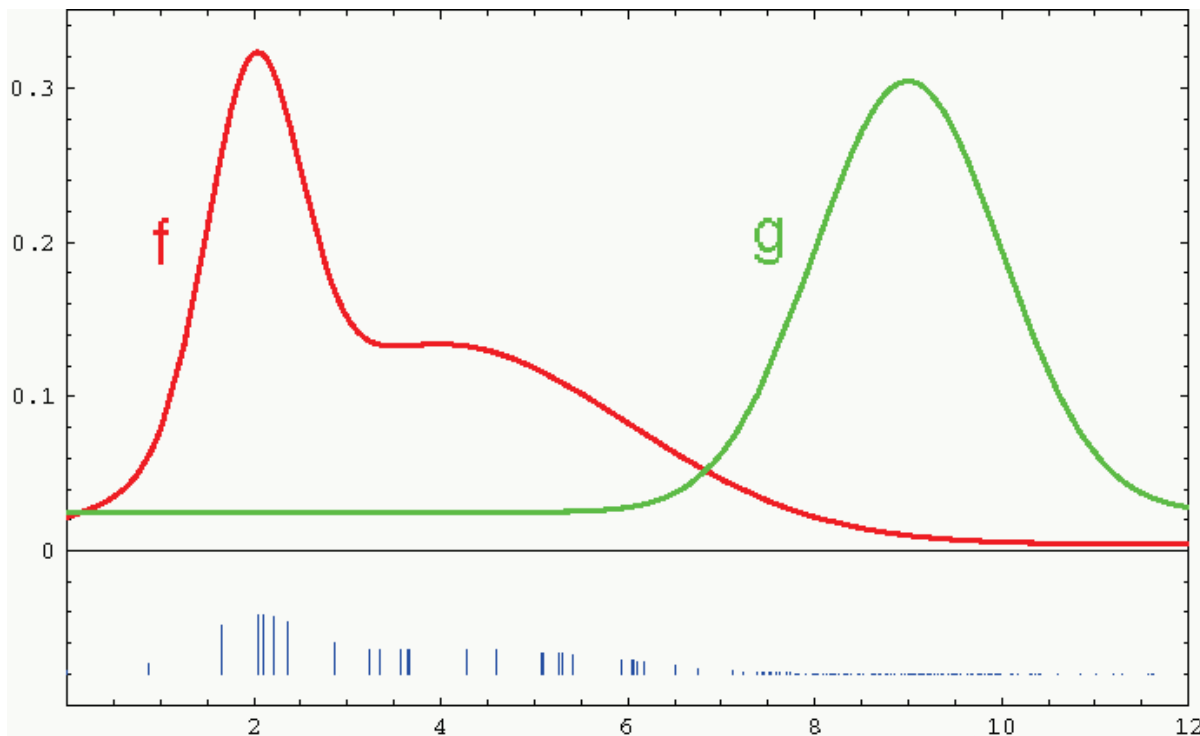


Herleitung Partikelfilter

Bedeutung der Gewichte

- ▶ **Partikel „zählt“ proportional zu seinem Gewicht.**
- ▶ **hohe Wahrscheinlichkeit dargestellt durch**
 - ▶ viele Partikel mit niedrigem Gewicht
 - ▶ wenige Partikel mit hohem Gewicht
- ▶ **Gewicht ermöglicht „importance sampling“**
 - ▶ aus einer ähnlichen aber einfacheren Verteilung ziehen.
 - ▶ Verhältnis der Wahrscheinlichkeiten ins Gewicht ziehen
 - ▶ wenn nötig danach resampeln
 - ▶ \Rightarrow entweder ziehen oder gewichten
- ▶ **Partikelfilter realisiert „importance sampling“**
 - ▶ Dynamik ziehen
 - ▶ Messung gewichten
 - ▶ Dynamik * Messung repräsentiert

Herleitung Partikelfilter



hohe Gewichte

viele Partikel

Importance Sampling

- ▶ aus einer anderen Verteilung ziehen, als man repräsentieren will.
 - ▶ ziehen nach g
 - ▶ Gewicht $w = f/g$
 - ▶ repräsentiert $g f/g = f$
- ▶ praktisch, wenn f schwieriger als g ist, aber beide ähnlich.
- ▶ Hier unsinnig, weil f und g zu verschieden

Herleitung Partikelfilter

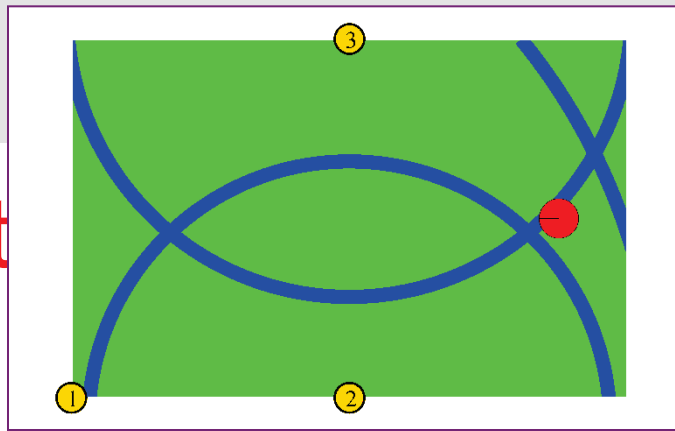
Importance Sampling Beispiel:

Lokalisation auf einem RoboCup Feld per Bildverarbeitung

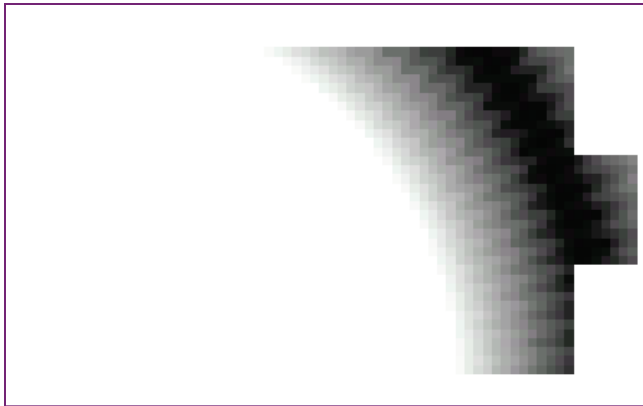
Zu farbigen Eck- / Mittellandmarken wird die Entfernung bestimmt.



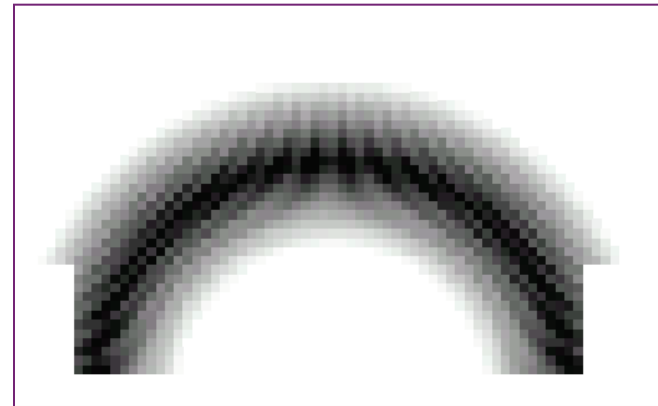
Herleitung Part



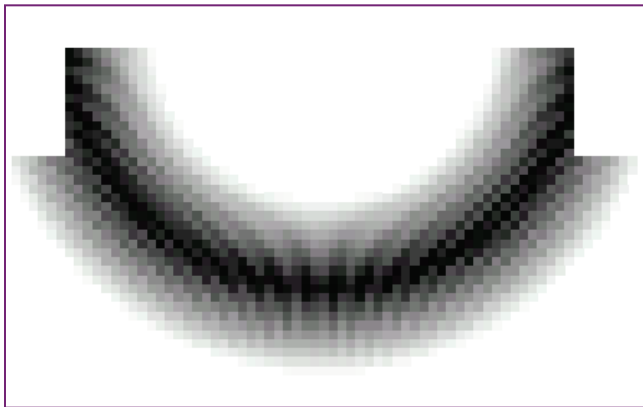
$p(x|z_1)$



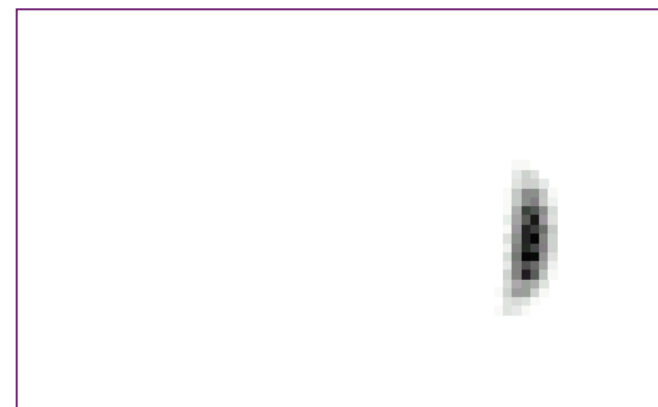
$p(x|z_2)$



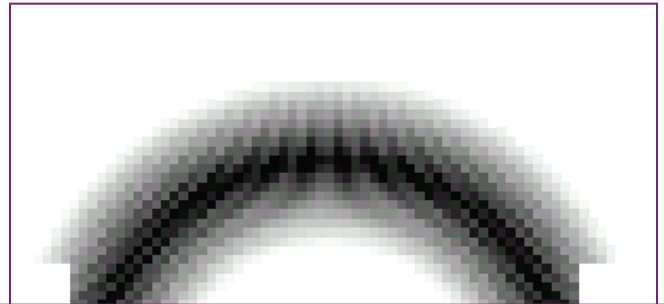
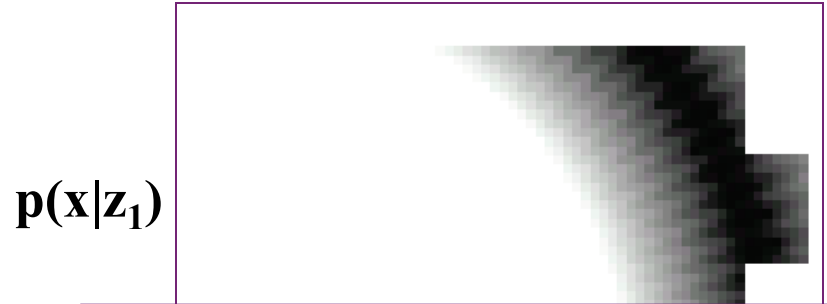
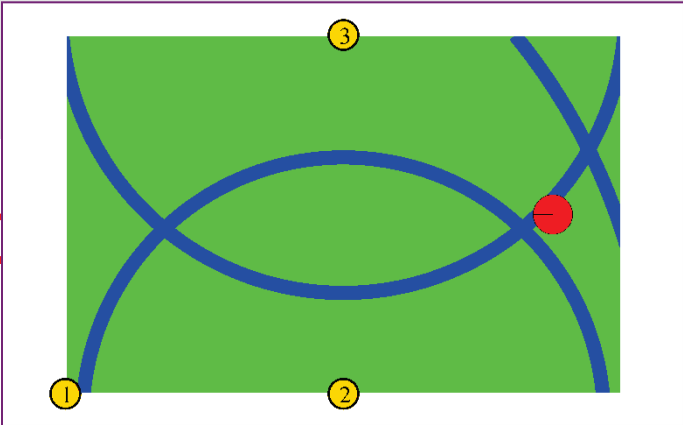
$p(x|z_3)$



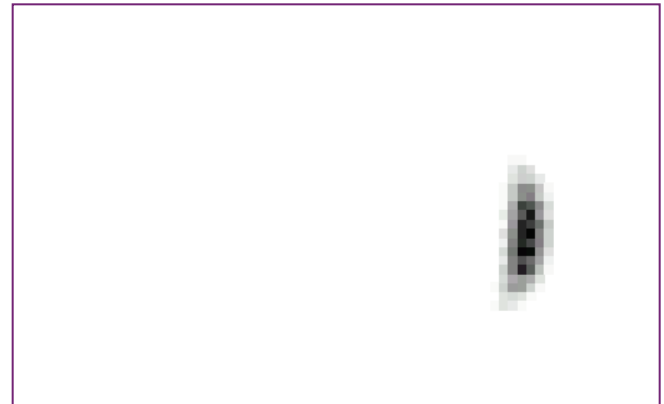
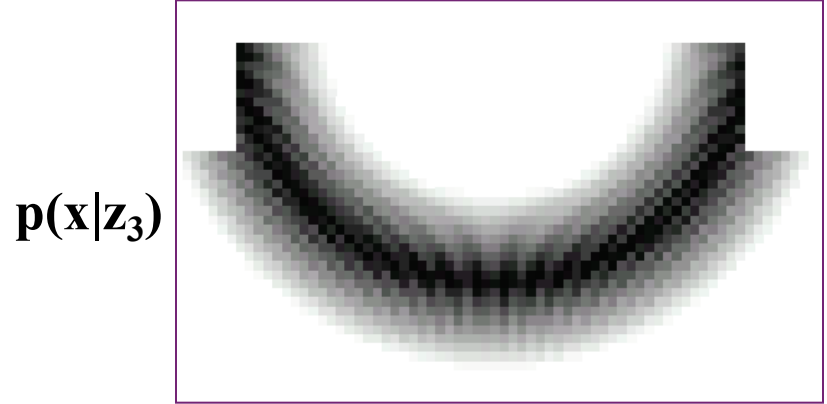
$p(x|z_1, z_2, z_3)$



Herleitung Part



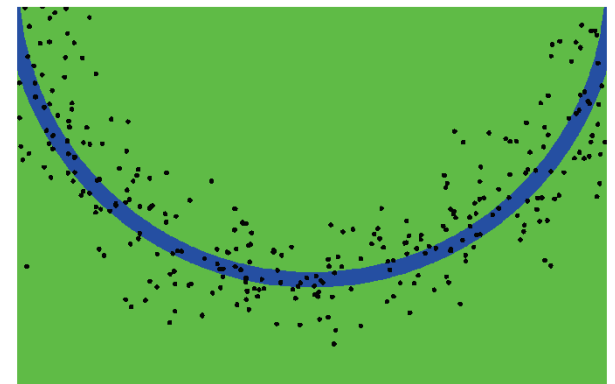
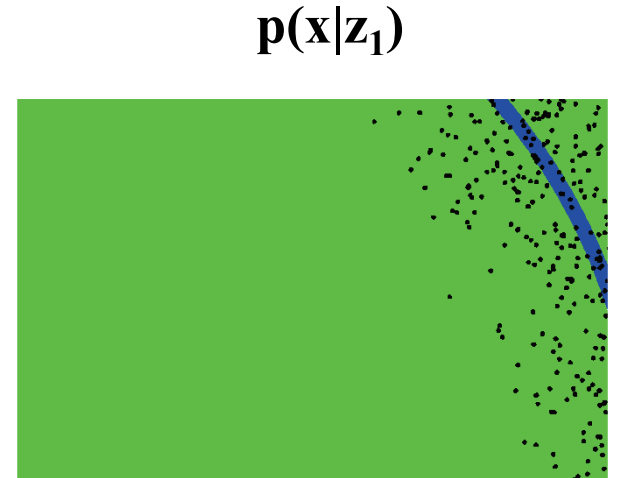
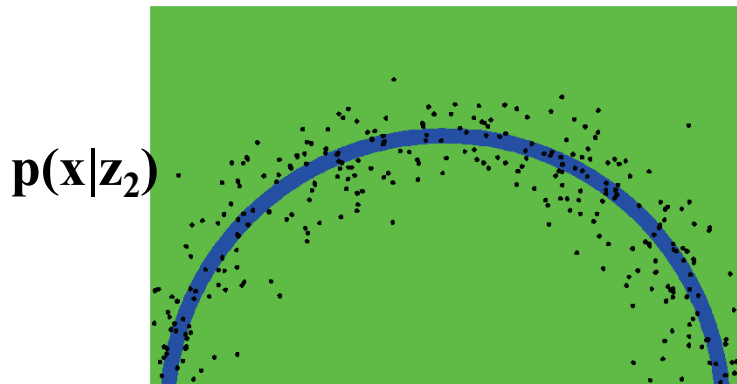
Gesucht: Stichproben aus $P(X | z_1, z_2, z_3)$



Herleitung Partikelfilter

Importance Sampling Beispiel:

- ▶ ziehen aus $P(X | z_i)$:
 - ▶ Rauschen auf z_i addieren
 - ▶ Winkel würfeln.



$p(x|z_3)$

Herleitung Partikelfilter

Importance Sampling Beispiel:

- ▶ ziehen aus $p(x|z_1)$
- ▶ gewichten nach $p(z_2|x)$
- ▶ gewichten nach $p(z_3|x)$
- ▶ resampling

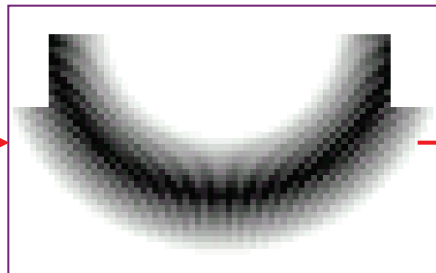
Entweder ziehen oder gewichten, nicht beides.



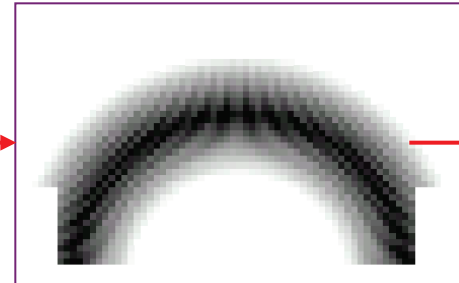
ziehen aus
 $p(x|z_1)$



gewichten
 $p(z_2|x)$



gewichten
 $p(z_3|x)$



resampling



Herleitung Partikelfilter

Importance Sampling Beispiel:

- ▶ ziehen aus $p(x|z_1)$
- ▶ gewichten nach $p(z_2|x)$
- ▶ gewichten nach $p(z_3|x)$
- ▶ resampling

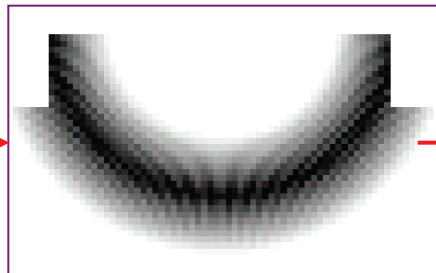
Frage an das Auditorium:

Welche Vorgehensweise wäre noch besser?

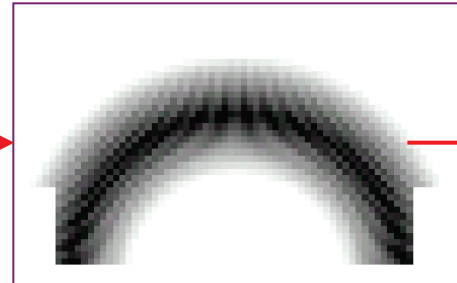
ziehen aus
 $p(x|z_1)$



gewichten
 $p(z_2|x)$



gewichten
 $p(z_3|x)$



resampling



Herleitung Partikelfilter

Importance Sampling Beispiel:

- ▶ ziehen aus $p(x|z_1)$
- ▶ gewichten nach $p(z_2|x)$
- ▶ gewichten nach $p(z_3|x)$
- ▶ resampling

Frage an das Auditorium:

Welche Vorgehensweise wäre noch besser?

- ▶ ziehen aus $p(x|z_1, z_2)$
- ▶ durch verrauschen und Kreise schneiden
- ▶ gewichten nach $p(z_3|x)$
- ▶ resampling

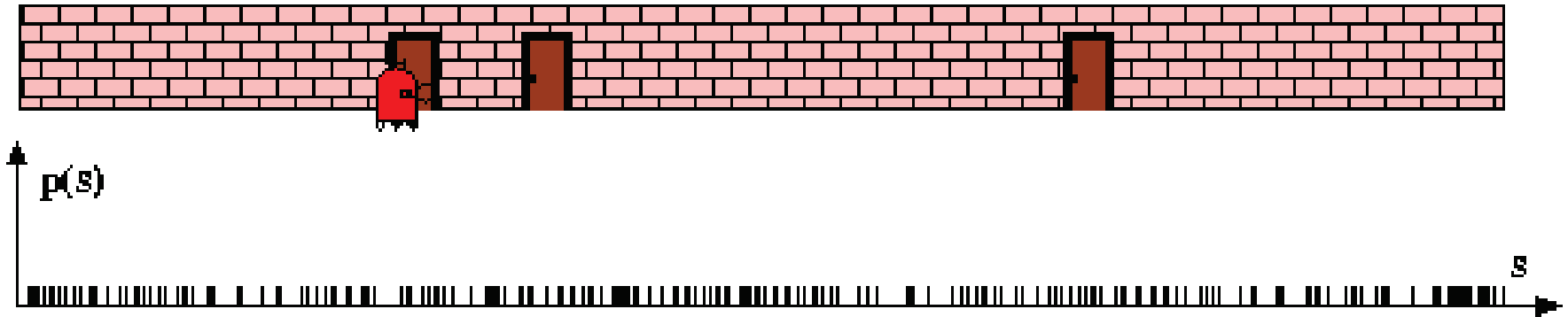
Herleitung Partikelfilter

Beispiel für vollen Particle Filter (aus Importance-Sampling Sicht):
Lokalisation eines Roboters mit Odometrie und “Türsensor”.

Quelle: Thrun et al., Probabilistic Robotics

Anfangsverteilung über den Zustand x_0

$$\text{Bel}(x_0) = P(x_0)$$

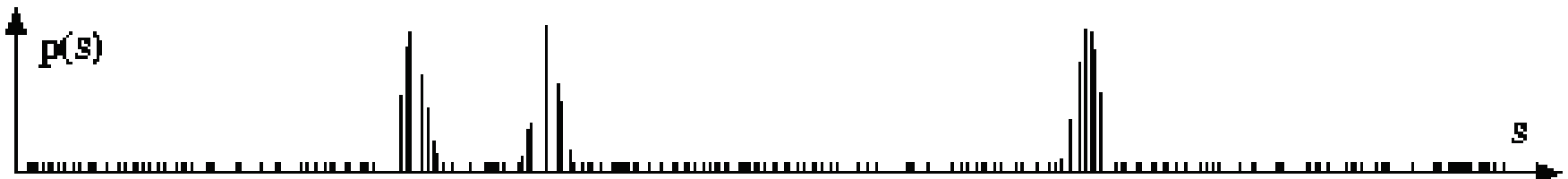
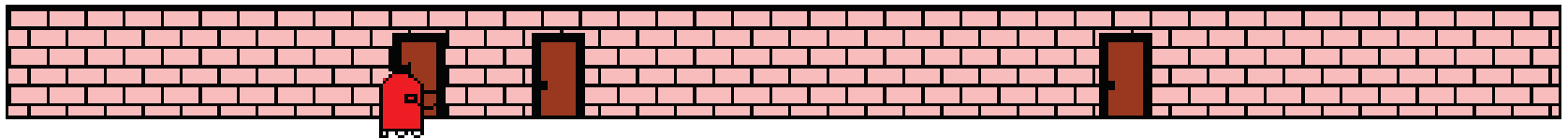
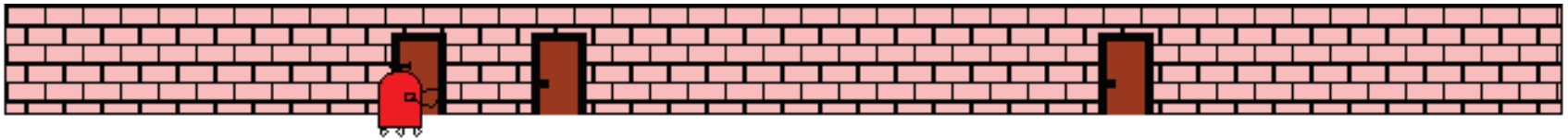


Messung z_0

$$Bel(x_0) \leftarrow \alpha p(z | x_0) Bel(x_0)$$

$$w \leftarrow \frac{\alpha p(z | x_0) Bel(x_0)}{Bel(x_0)} = \alpha p(z | x_0)$$

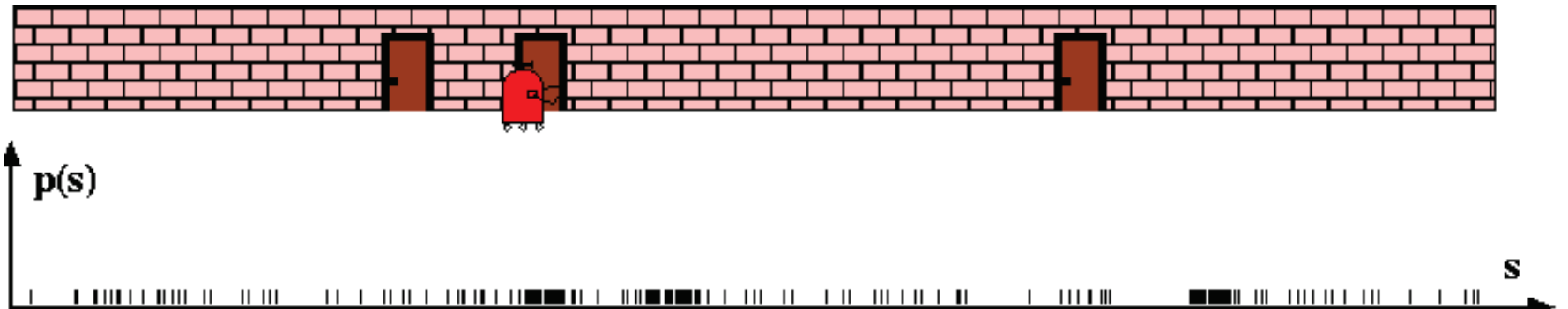
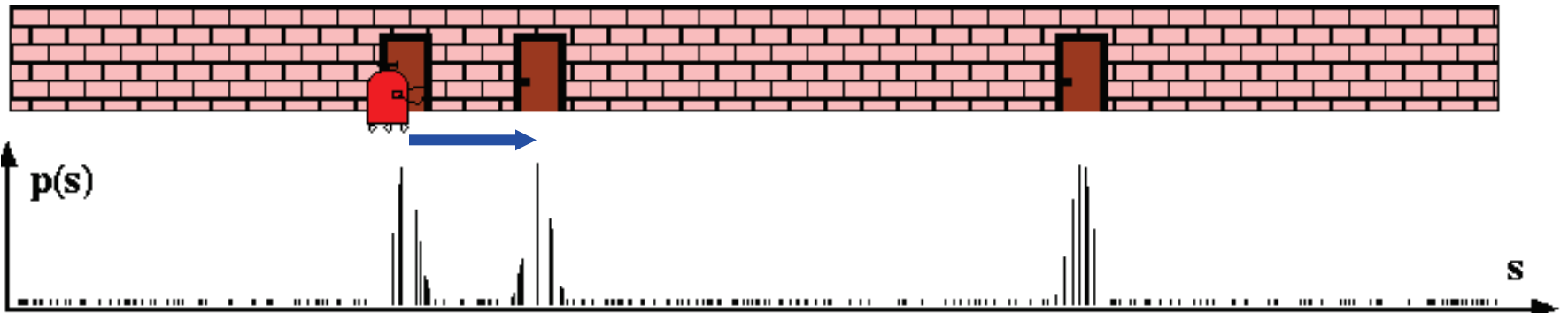
Multipliziere
Gewicht mit
 $p(z|x_0)$



Dynamik u_0 von x_0 nach x_1

$$Bel(x_1) \leftarrow \int p(x_1 | u_0, x_0) Bel(x_0) dx_0$$

Ziehe aus Partikeln
gemäß Gewicht.
Ziehe dann x_t aus
 $p(X_t | x_0, u_0)$
Integration implizit
durch weglassen
von x_0

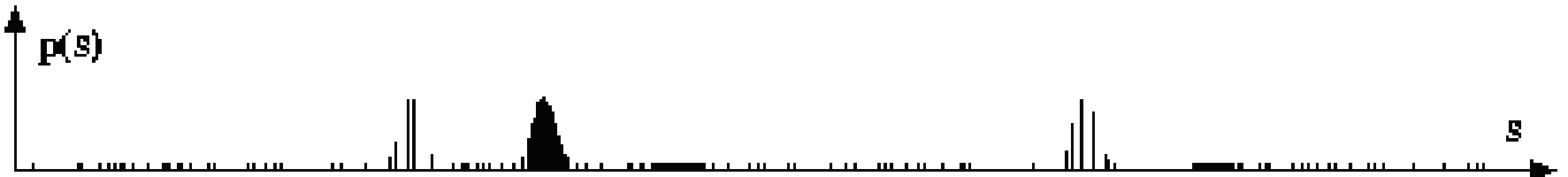
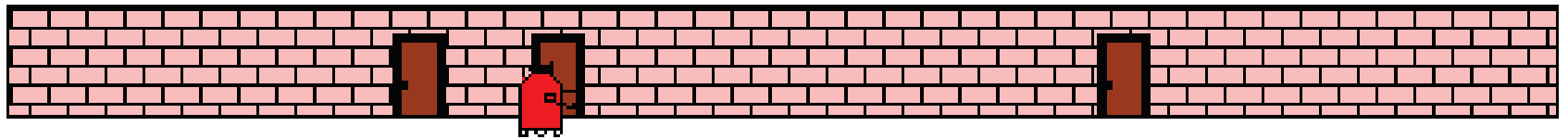
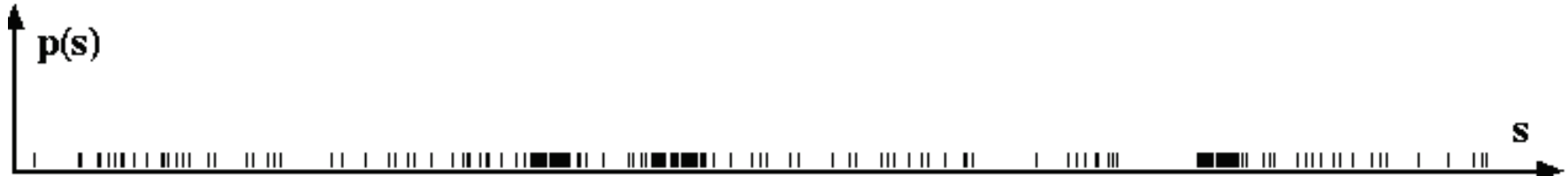
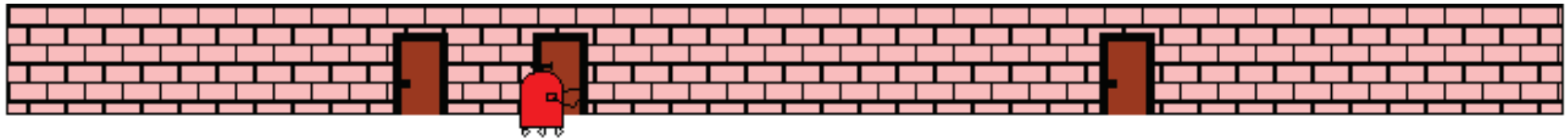


Messung z_1

$$Bel(x_1) \leftarrow \alpha p(z | x_1) Bel(x_1)$$

$$w \leftarrow \frac{\alpha p(z | x_1) Bel(x_1)}{Bel(x_1)} = \alpha p(z | x_1)$$

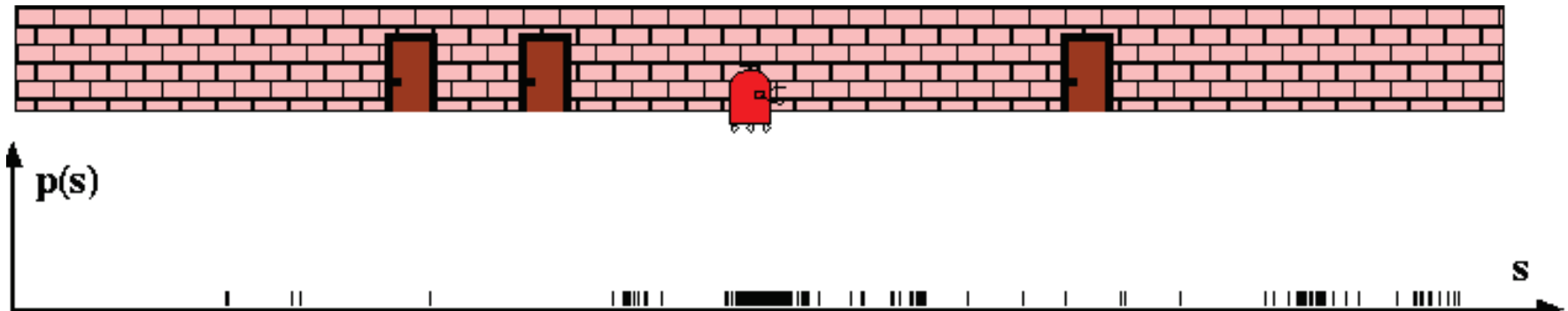
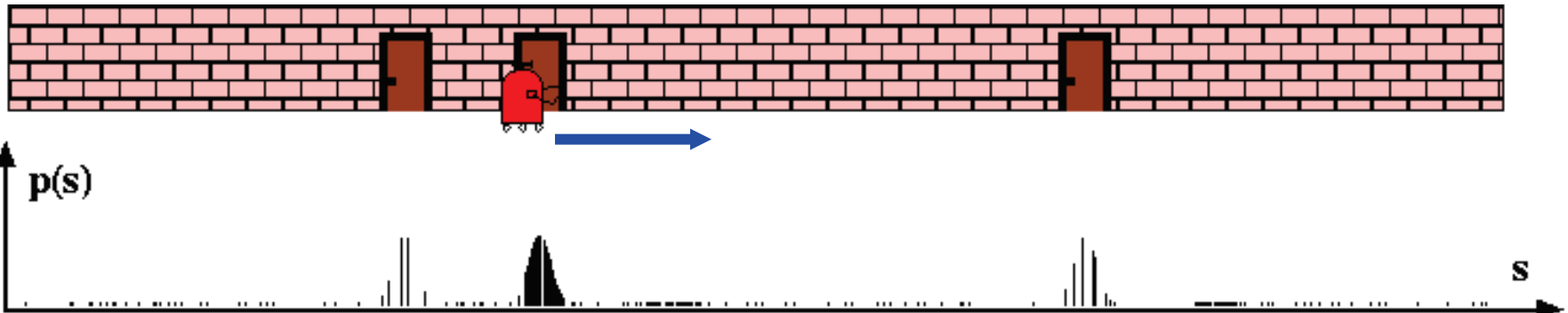
**Multipliziere
Gewicht mit
 $p(z|x_1)$**



Dynamik u_1 von x_1 nach x_2

$$Bel(x_2) \leftarrow \int p(x_2 | u_1, x_1) Bel(x_1) dx_1$$

Ziehe aus Partikeln
gemäß Gewicht.
Ziehe dann x_t aus
 $p(X_2|x_1, u_1)$
Integration implizit
durch weglassen
von x_1



Herleitung Partikelfilter

Partikelfilter aus Importance-Sampling Sicht

$$Bel(x_t) = \eta p(z_t | x_t) \int p(x_t | x_{t-1}, u_{t-1}) Bel(x_{t-1}) dx_{t-1}$$

**Gewicht-
summe=1**

Gewicht

Ziehen

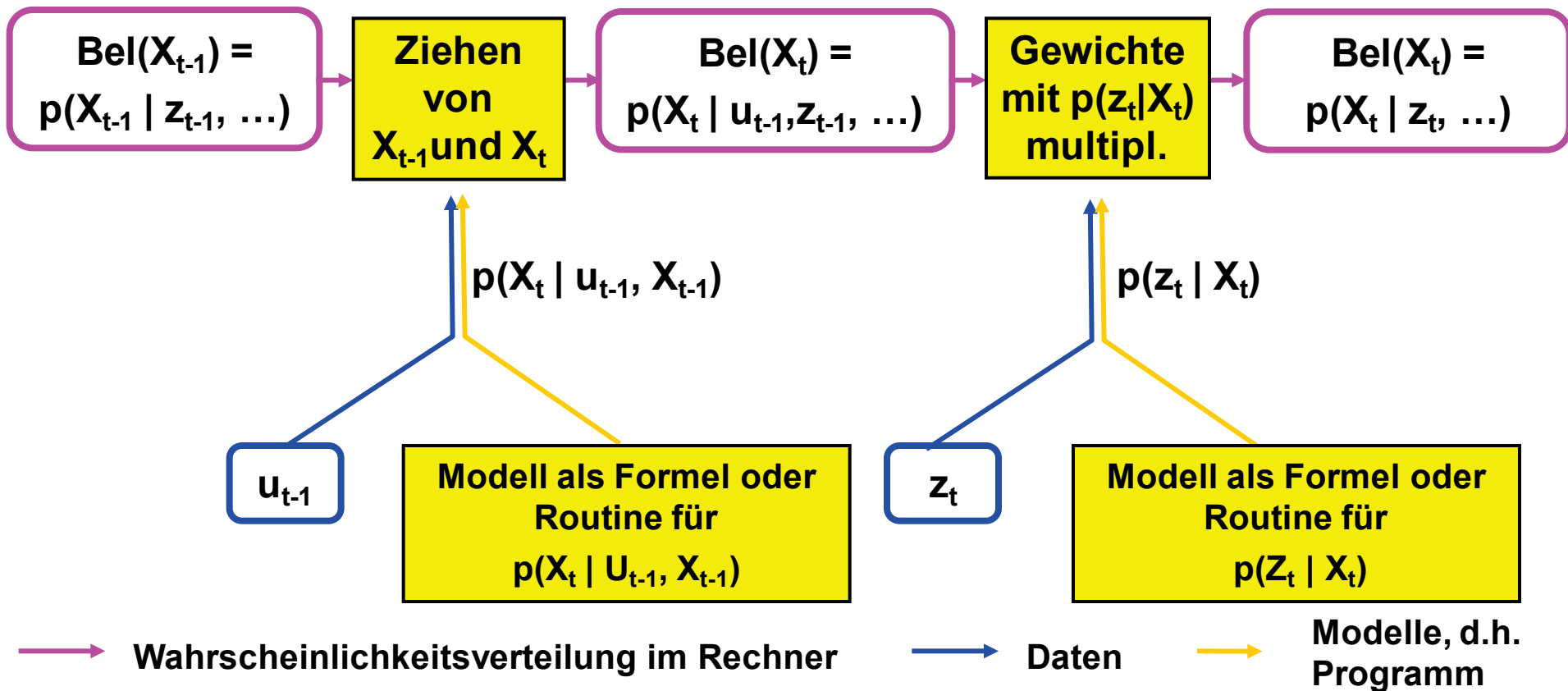
ziehe x_{t-1}^i aus $Bel(x_{t-1})$

ziehe x_t^i aus $p(x_t | x_{t-1}^i, u_{t-1})$

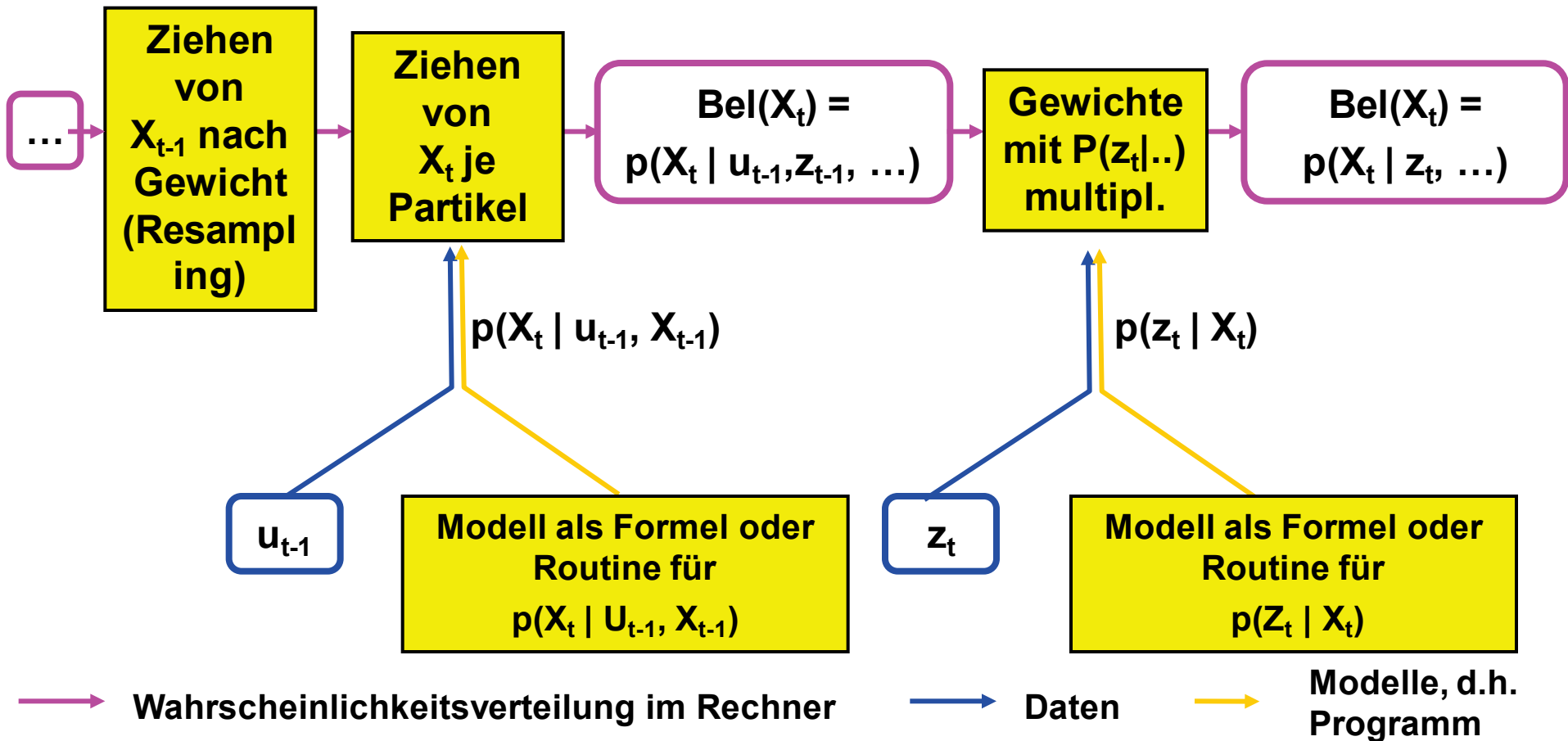
Gewicht für x_t^i :

$$\begin{aligned} w_t^i &= \frac{\text{Ziel Verteilung}}{\text{gezogene Verteilung}} \\ &= \frac{\eta p(z_t | x_t) p(x_t | x_{t-1}^i, u_{t-1}) Bel(x_{t-1}^i)}{p(x_t | x_{t-1}^i, u_{t-1}) Bel(x_{t-1}^i)} \\ &\propto p(z_t | x_t) \end{aligned}$$

Herleitung Partikelfilter



Herleitung Partikelfilter



Zusammenfassung

- ▶ **Partikel Filter sind Bayesfilter**
- ▶ **repräsentieren a-posteriori Verteilung durch gewichtete Stichproben**
- ▶ **Dynamik: ziehen aus Zustandsübergangsverteilung**
 - ▶ praktisch: ideale Bewegung + gezogenes Rauschen
- ▶ **Messung: gewichten**
 - ▶ praktisch oft: Differenz zu idealer Messung in Fehlerverteilung
- ▶ **re-sampling: ziehen aus Partikeln mit Wahrscheinlichkeit proportional zu Gewicht**
 - ▶ systematisch, nicht unabhängig ziehen
- ▶ **initialisieren auf Grund der ersten (paar) Messungen**
- ▶ **jede Unsicherheit *einmal* einbringen, d.h. *entweder* zufällig ziehen oder gewichten.**