

Kognitive Robotik – Ablauf

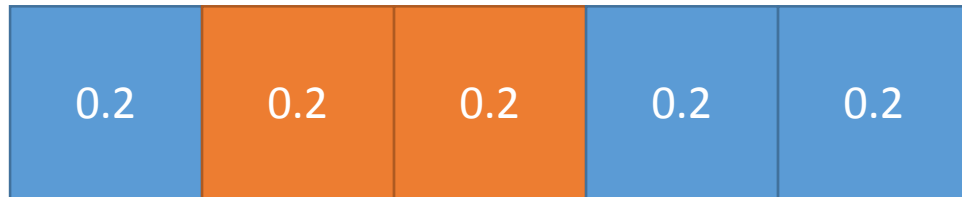
- 25.04 – Einführung Vision Aufgabe
- 02.05 – nach bedarf
- 09.05 – Vorstellung der Ergebnisse
- 16.05 – Einführung Lokalisierung
- 23.05 - Lokalisierung
- 30.05 - Lokalisierung
- 13.06 - Lokalisierung
- 20.06 – Lokalisierung
- 27.06 - Lokalisierung
- 04.07 – Parallelkinematik
- 11.07 – Parallelkinematik
- 18.07 – Prüfungsvorbereitung
oder Parallelkinematik

Recap Pylonenaufgabe

- Bitte keine Binaries abgeben!
- Bitte Format einhalten!
- Bitte Anleitung schreiben wie man das Programm starten soll
- Bitte alle Namen vollständig in der Email schreiben
- Betreff in der Email nicht vergessen (sonst wird es als Spam erkannt)

Beispiel: 1D Histogramfilter

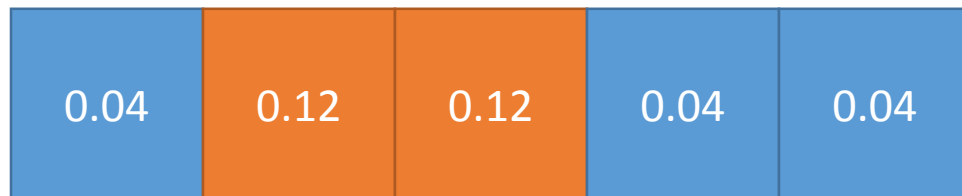
Wahrscheinlichkeitsverteilung vor Sensorupdate:



Wahrscheinlichkeitsverteilung nach dem eine orange Zelle wahrgenommen wurde.

Orange Zellen werden in dem Fall mit 0.6 multipliziert, blaue mit 0.2

D.h. bedeutet das mit wenn der Roboter Orange wahrnimmt mit hoher Wahrscheinlichkeit auch Orange da ist.

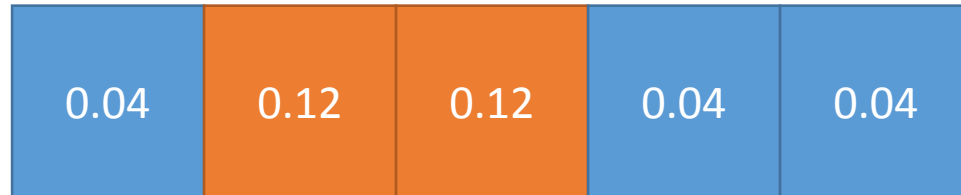


$$\sum X_i = 0.04 + 0.12 + 0.12 + 0.04 + 0.04 = 0.36$$

$$X_i = \frac{X_i}{\sum X_i}$$

Beispiel: 1D Histogramfilter

Verteilung vor Normalisierung



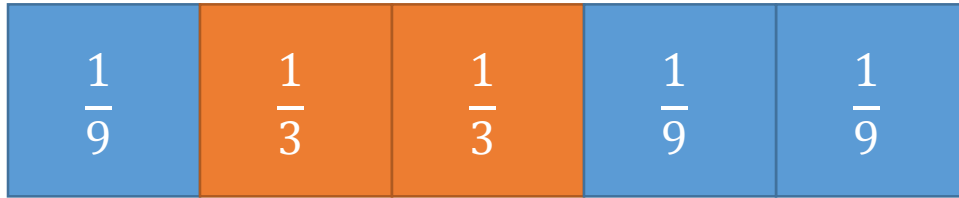
Verteilung nach Normalisierung (Posterior Distribution)



$$X_i = \frac{X_i}{\sum X_i}$$

Beispiel: 1D Histogramfilter

Verteilung nach Normalisierung (Posterior Distribution) – vor Bewegung

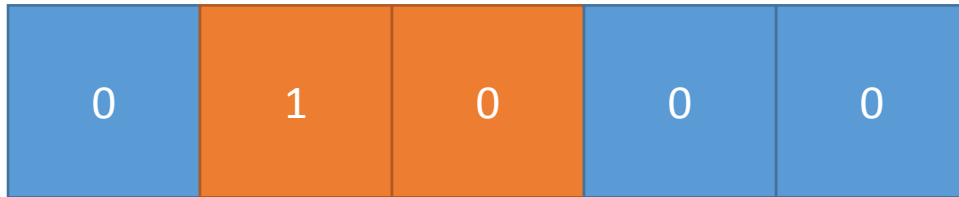


Bewegung nach rechts um 1 Feld. Die Welt ist dabei zyklisch

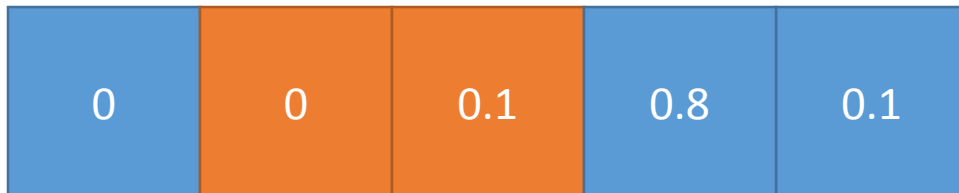


Beispiel: 1D Histogramfilter

Neues Beispiel: Wahrscheinlichkeitsverteilung vor der Bewegung



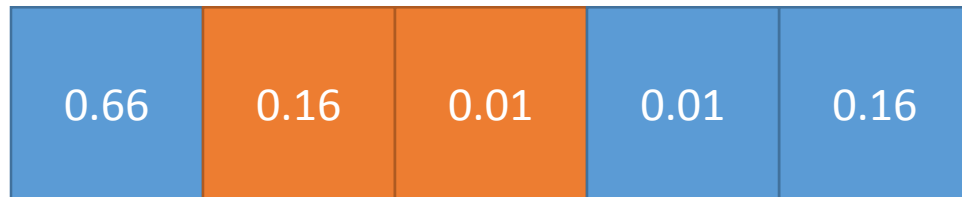
Der Roboter ist mit Wahrscheinlichkeit von 1 in der 2. Zelle. Eine perfekte Bewegung des Roboters entspricht zwei Schritte nach rechts in einer zyklischen Welt. Wir nehmen jetzt an der Roboter kann keine perfekten Bewegungen machen. Mit einer Wahrscheinlichkeit von 0.8 führt er tatsächlich die 2 Schritte aus und mit einer Wahrscheinlichkeit von 0.1 geht er nur ein Schritt bzw. 3 Schritte.



Beispiel: 1D Histogramfilter



Eine wiederholte Anwendung der Motion Update Regel ergibt:



Achtung: Die Update Regel muss für jede Zelle berechnet werden.
Danach kommt wieder ein Sensorupdate so wie auf Folie 3 beschrieben

Erste Aufgabe: 2D Histogram Filter

- 1DLocalization.py als Ausgangspunkt nehmen und eine 2D Version daraus erstellen
- Das Script 1DLocalization.py implementiert den ersten Teil der Übung
- motions Array kann in 2D folgende Werte enthalten:
 - [0,0] – stehen bleiben
 - [0,1] – nach rechts bewegen
 - [0,-1] – nach links bewegen
 - [1,0] – nach unten
 - [-1,0] nach oben bewegen
 - Diagonale Bewegungen könnt ihr weglassen
- World Variable muss zu einem 2D Array werden
- Fragen bitte vor Dienstag an schlottb@informatik.hu-berlin.de per Email stellen

Abgabe Modalitäten

- README.txt mit folgenden Informationen:
 - Namen aller Gruppenmitglieder
 - Beschreibung des Kommandozeilenaufrufs (ggf. Parameter erklären)
 - Python Version
- Modifizierte Python Datei(en)
- In der Email müssen die vollen Namen aller Gruppenmitglieder stehen
- Bitte keine Links auf Github Repositories/Dropbox oder ähnliches
- Dateien sollen als tar.gz verpackt im Anhang der Email stehen.
 - Email an schlottb@informatik.hu-berlin.de
- Deadline: 23.05.2017 23:59