technische universität
dortmund

Institut für Roboterforschung
Abteilung Informationstechnik

Modellierung

Probabilistische Lokalisierung (Part 1) und kooperative Weltmodellierung (Part 2)

Stefan Tasse

Part 1 – Hands on Kalman Filter

- Theorie
  - Kalman Filter
  - Extended Kalman Filter
  - Unscented Kalman Filter
    - [noch mehr Formeln...]

$$\boldsymbol{F}_{k-1} = \frac{\partial f}{\partial \boldsymbol{x}}$$

$$\boldsymbol{H}_k = \left.\frac{\partial h}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}}$$

$$\hat{\boldsymbol{x}}_{k|k-1} = f(\hat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{u}_{k-1})$$

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{F}_{k-1}\boldsymbol{P}_{k-1|k-1}\boldsymbol{F}_{k-1}^\top + \boldsymbol{Q}_{k-1}$$

$$\tilde{\boldsymbol{y}}_k = \boldsymbol{z}_k - h(\hat{\boldsymbol{x}}_{k|k-1})$$

$$\boldsymbol{S}_k = \boldsymbol{H}_k\boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^\top + \boldsymbol{R}_k$$

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^\top\boldsymbol{S}_k^{-1}$$

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k\tilde{\boldsymbol{y}}_k$$

$$\boldsymbol{P}_{k|k} = (I - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{P}_{k|k-1}$$

- Und jetzt?

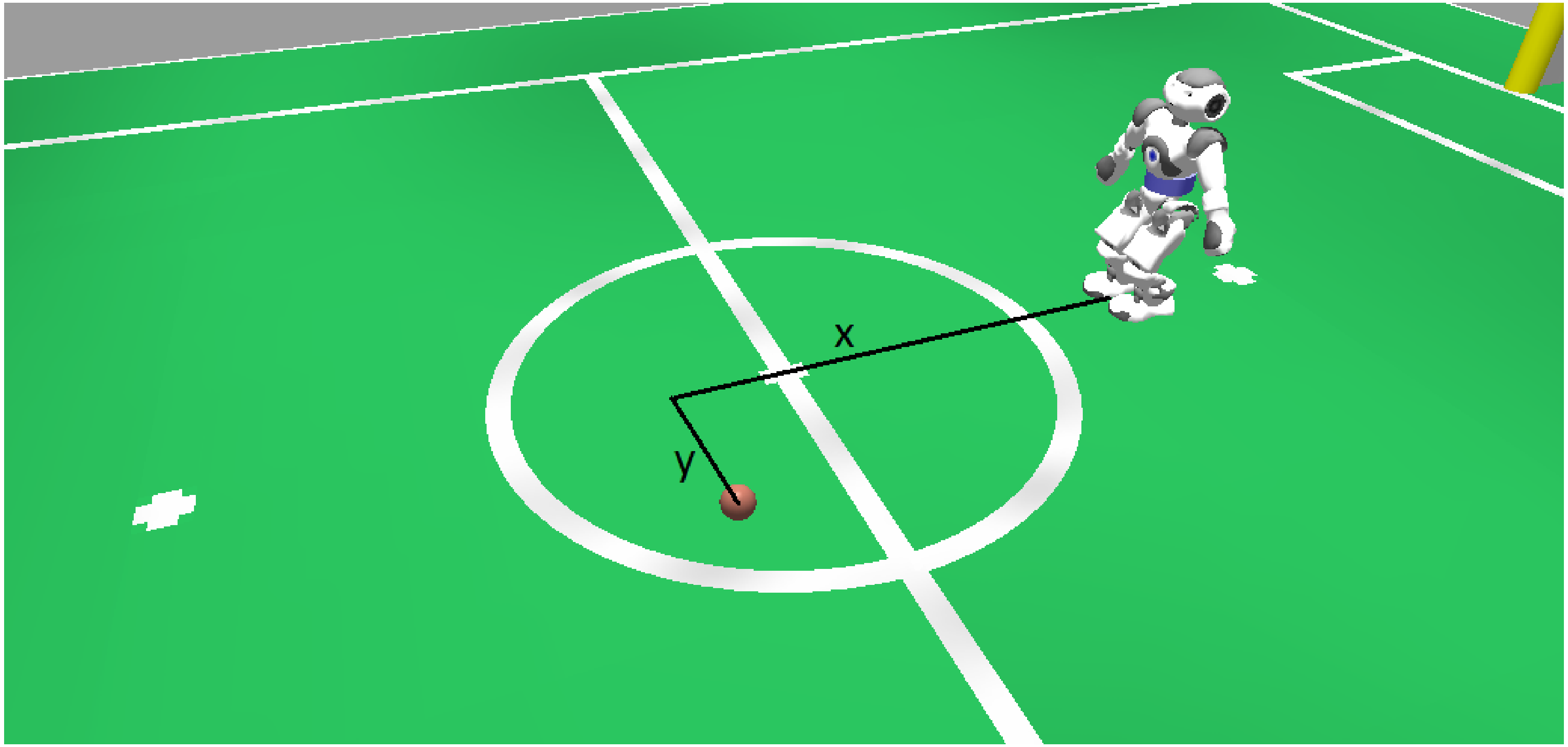
- Anwendung
  - Prozessmodel f(x,u) festlegen: trivial für Lokalisierung
  - Sensormodel h(x) festlegen: einfach, aber verschiedene Umsetzungen möglich...

Sensormodelle: Koordinatensysteme

- Messung z_t
  - Roboter beobachtet ein Feature zum Zeitpunkt t
  - Verschiedene Koordinatensysteme als Interface zwischen Bildverarbeitung und Lokalisierung möglich
  - Umrechnung ohne Informationsverlust möglich
  - Koordinatensystem also egal, oder?

- Zur Erinnerung: Kalman nur „toll" für lineare Systeme!
  - $\hat{z} = h(x,l)$ (erwartete Beobachtung)
  - Jacobi-Matrix H (Zusammenhänge bei Abweichung von Erwartung)

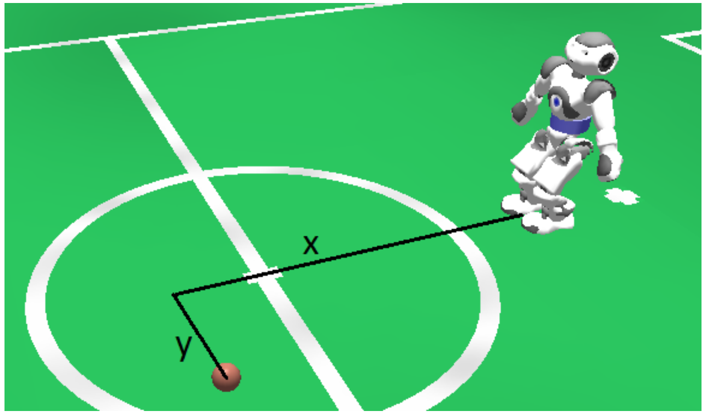
Sensormodel in euklidischen Koordinaten



Sensormodel in euklidischen Koordinaten

- Sensormodel

$$\overline{z} = \begin{pmatrix} z_x \\ z_y \end{pmatrix} = h(x, l) = \Omega(-p_\theta) \cdot \left[\begin{pmatrix} l_x \\ l_y \end{pmatrix} - \begin{pmatrix} p_x \\ p_y \end{pmatrix} \right]$$

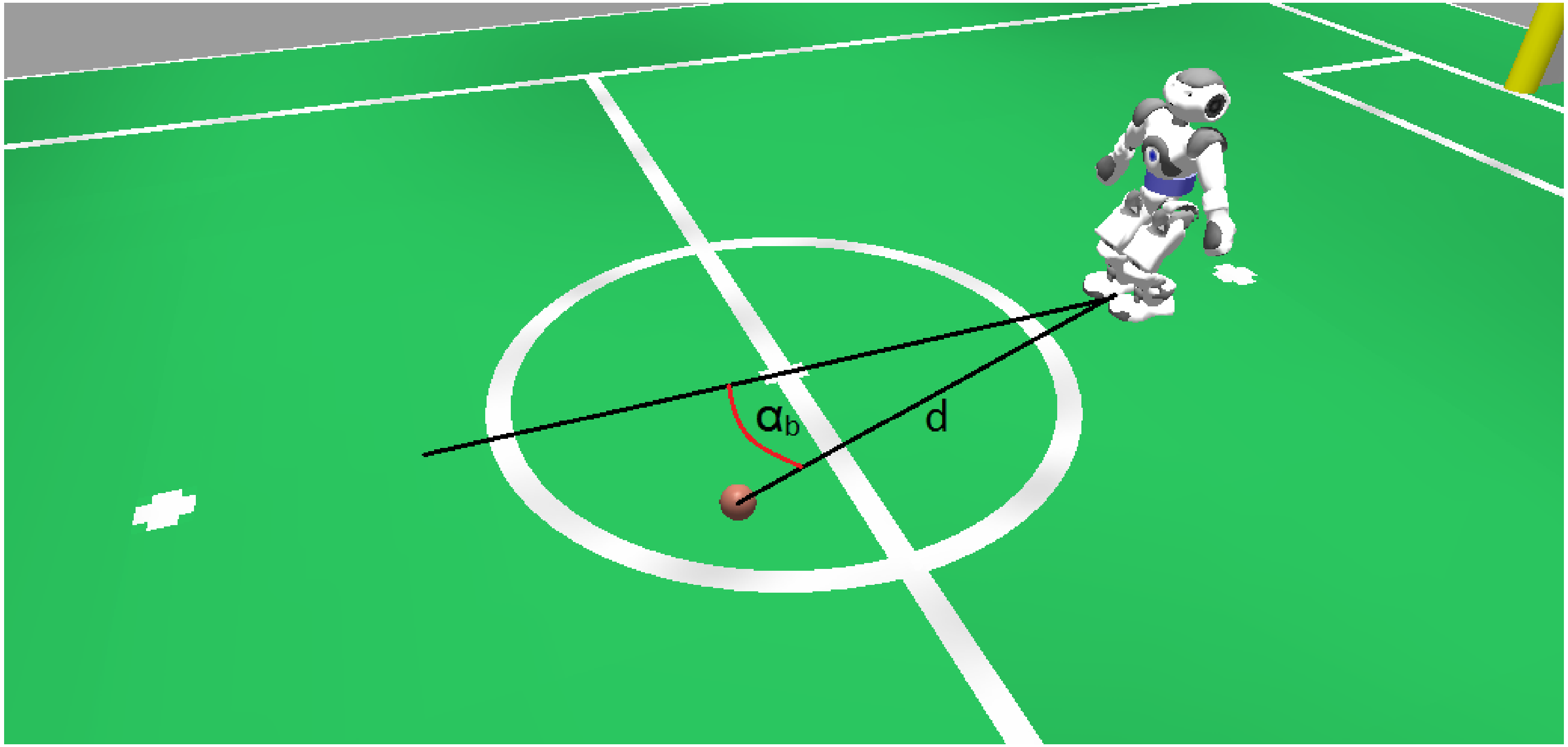


- Jacobi-Matrix H

$$H = \frac{\partial h(x,l)}{\partial x} = \begin{pmatrix} \frac{\partial z_x}{\partial p_x} & \frac{\partial z_x}{\partial p_y} & \frac{\partial z_x}{\partial p_\theta} \\ \frac{\partial z_y}{\partial p_x} & \frac{\partial z_y}{\partial p_y} & \frac{\partial z_y}{\partial p_\theta} \end{pmatrix}$$

$$= \begin{pmatrix} -\cos p_\theta & -\sin p_\theta & -(l_x - p_x)\sin p_\theta + (l_y - p_y)\cos p_\theta \\ sin p_\theta & -\cos p_\theta & -(l_x - p_x)\cos p_\theta - (l_y - p_y)\sin p_\theta \end{pmatrix}$$

Sensormodel in Range/Bearing

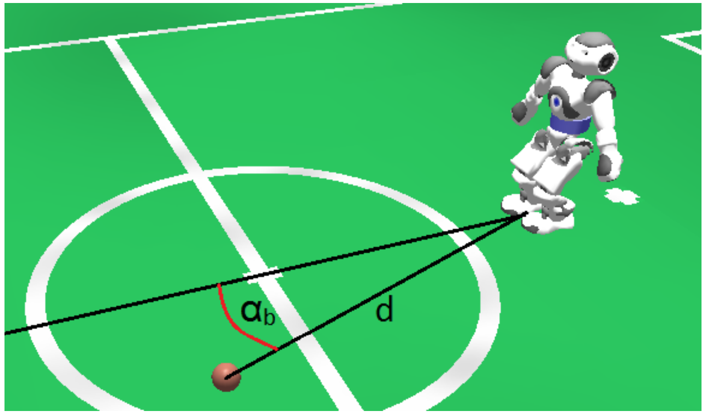


Sensormodel in Range/Bearing

- Sensormodel

$$\bar{z} = \begin{pmatrix} z_r \\ z_b \end{pmatrix} = h(x,l) = \begin{pmatrix} \sqrt{(l_x - p_x)^2 + (l_y - p - y)^2} \\ \operatorname{atan2}(l_y - p_y, l_x, p_x) - p_\theta \end{pmatrix}$$

$$d^2 = (l_x - p_x)^2 + (l_y - p_y)^2.$$

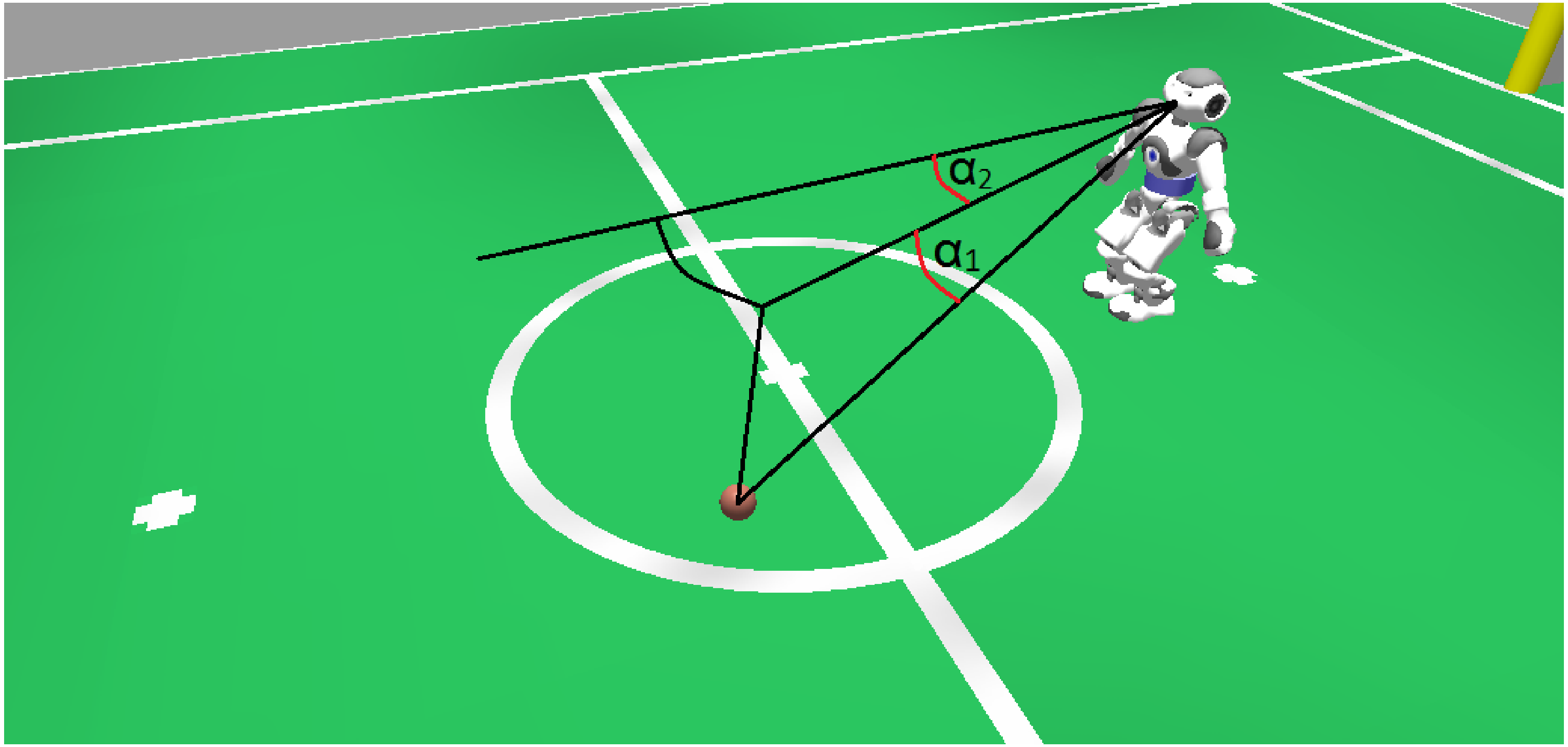


- Jacobi-Matrix H

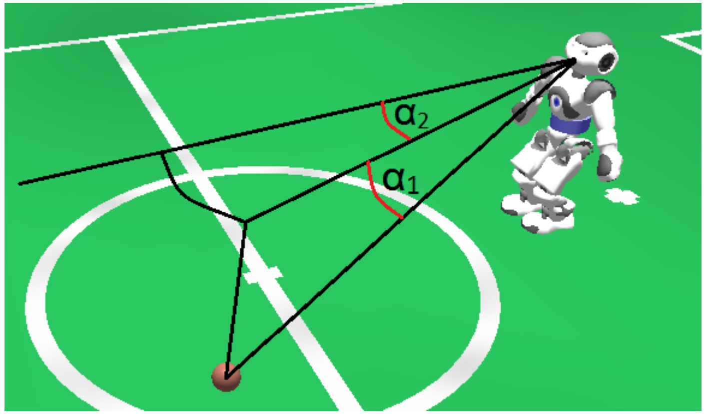
$$H = \frac{\partial h(x,l)}{\partial x} = \begin{pmatrix} \frac{\partial z_r}{\partial p_x} & \frac{\partial z_r}{\partial p_y} & \frac{\partial z_r}{\partial p_\theta} \\ \frac{\partial z_b}{\partial p_x} & \frac{\partial z_b}{\partial p_y} & \frac{\partial z_b}{\partial p_\theta} \end{pmatrix}$$

$$= \begin{pmatrix} (-l_x + p_x)d^{-1} & (-l_y + p_y)d^{-1} & 0 \\ (-l_y + p_y)d^{-2} & (l_x - p_x)d^{-2} & -1 \end{pmatrix}$$

Sensormodel in Winkel-Koordinaten



Sensormodel in Winkel-Koordinaten



- Sensormodel

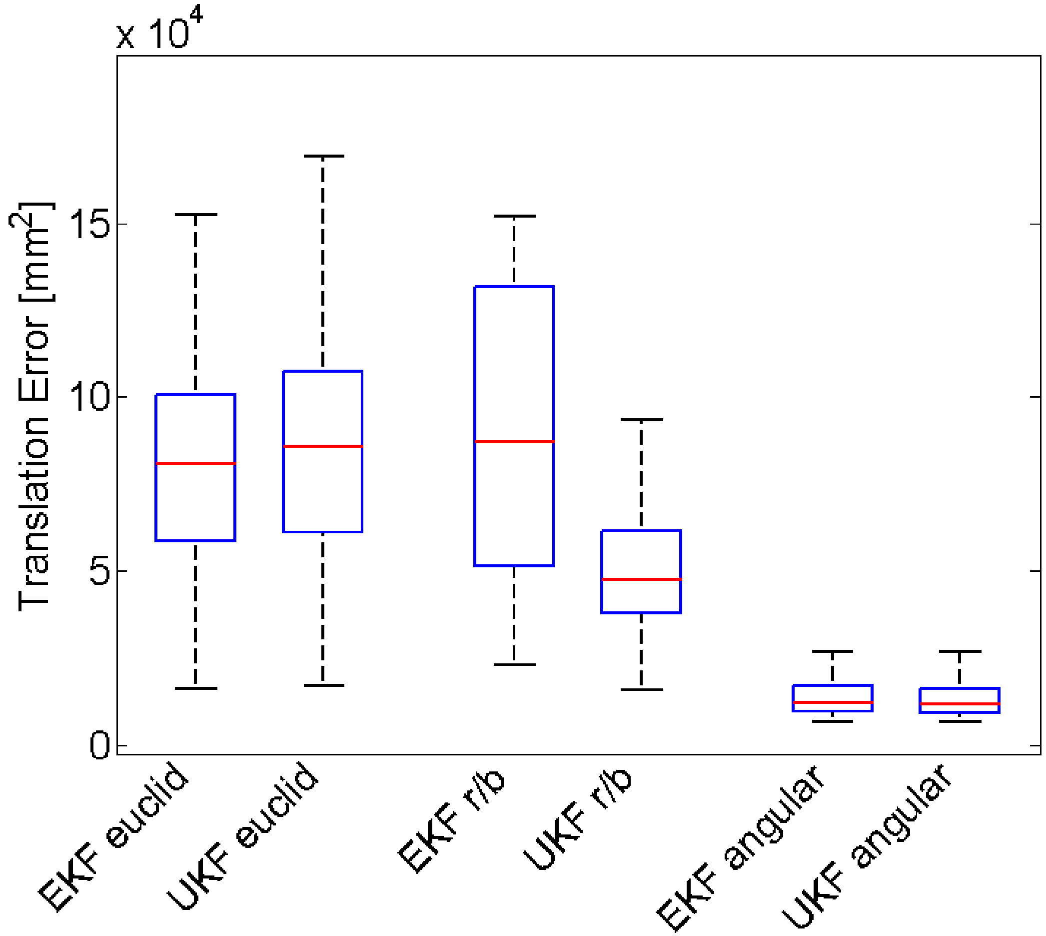
$$\bar{z} = \begin{pmatrix} z_{\alpha_1} \\ z_{\alpha_2} \end{pmatrix} = h(x, l, h_{camera}) = \begin{pmatrix} \operatorname{atan2}(h_{camera}, \sqrt{(l_x - p_x)^2 + (l_y - p - y)^2}) \\ \operatorname{atan2}(l_y - p_y, l_x, p_x) - p_\theta \end{pmatrix}$$

- Jacobi-Matrix H

$$H = \frac{\partial h(x,l)}{\partial x} = \begin{pmatrix} \frac{\partial z_{\alpha_1}}{\partial p_x} & \frac{\partial z_{\alpha_1}}{\partial p_y} & \frac{\partial z_{\alpha_1}}{\partial p_\theta} \\ \frac{\partial z_{\alpha_2}}{\partial p_x} & \frac{\partial z_{\alpha_2}}{\partial p_y} & \frac{\partial z_{\alpha_2}}{\partial p_\theta} \end{pmatrix}$$

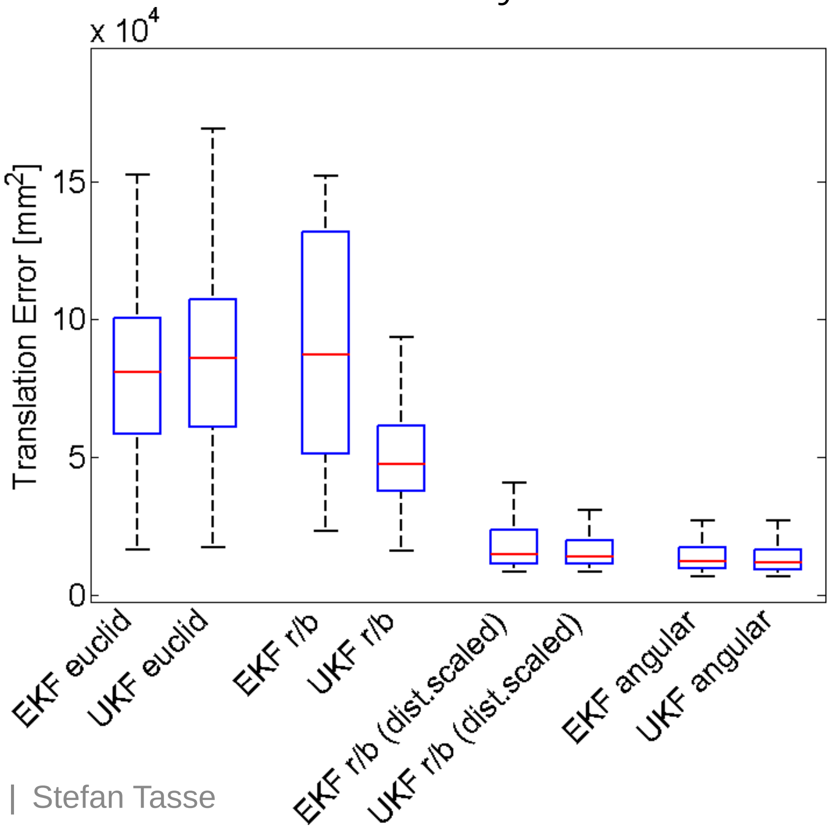
$$= \begin{pmatrix} h_{camera} d (h_{camera}^2 + d^2)^{-1} (l_x - r_x) & h_{camera} d (h_{camera}^2 + d^2)^{-1} (l_y - r_y) & 0 \\ (-l_y + p_y) d^{-2} & (l_x - p_x) d^{-2} & -1 \end{pmatrix}$$

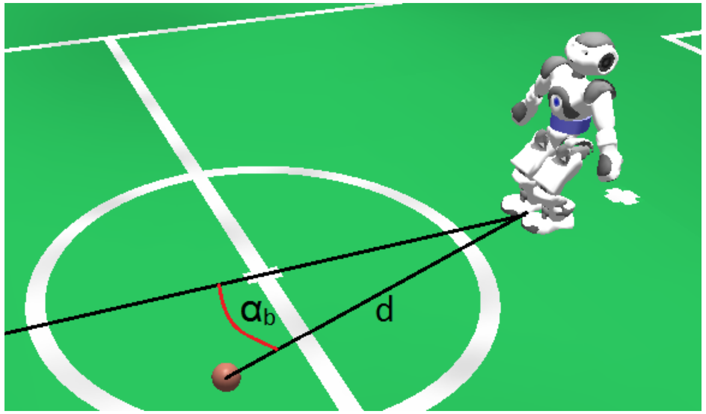
Experimenteller Vergleich



Sensormodel in Range/Bearing (angepasst)

- Range-Variance skalieren mit Wissen (partielle Ableitungen usw.) aus Winkelkoordinatensystem





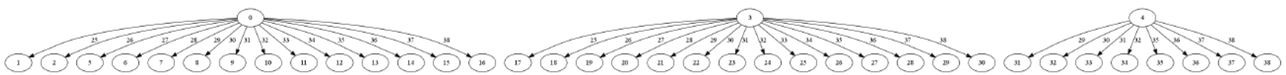
Sensormodelle: Mehrere Beobachtungen gleichzeitig

- Messung z_t
  - Roboter beobachtet mehr als ein Feature zum Zeitpunkt t
- 2 Alternativen:
  - n mal 2-dim Sensorupdates (einfach, weil schon implementiert)
  - Ein (n*2)-dim Sensorupdate

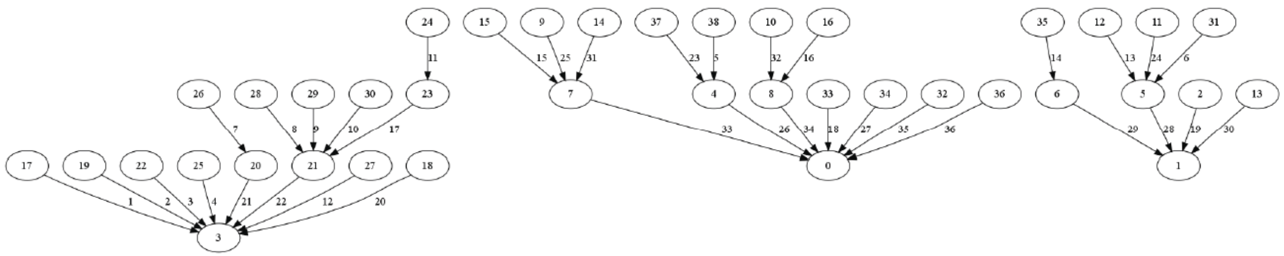
Multihypothesen-Systeme (Kursfassung)

- Partikelfilter

- Multi-Model Kalman / Gaussian Mixture Filter
  - Quinlan, M., Middleton, R.: Multiple Model Kalman Filters: A Localization Technique for RoboCup Soccer. In: RoboCup 2009, Springer



(a) Split when models are uncertain



(b) Merge after uncertain split

Multihypothesen-Systeme (Kursfassung)

- Partikelfilter

- Multi-Model Kalman / Gaussian Mixture Filter
  - Quinlan, M., Middleton, R.: Multiple Model Kalman Filters: A Localization Technique for RoboCup Soccer. In: RoboCup 2009, Springer

- Multi-Model Kalman (Alternative Version)
  - Jochmann, G., Kerner, S., Tasse, S., Urbann, O.: Efficient Multi-Hypotheses Unscented Kalman Filtering for Robust Localization. In: RoboCup 2011, Springer
  - Nebenläufige Modelle mit Maximum-Likelyhood Updates
  - Neue Modelle per Sensor Resetting Techniken

Part 1 – Hands on Kalman Filter

- Fragen soweit? :-)

Part 2 – Kooperative Weltmodellierung

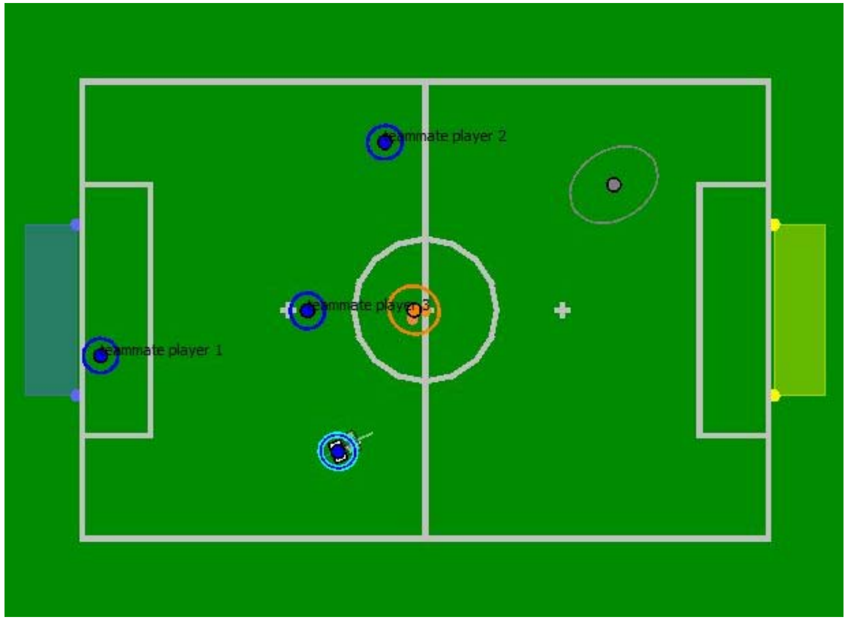
- „Wissen, wo alles ist, auch wenn man selbst grad nicht hinguckt."
  - Eigene Wahrnehmung oft unsicher:
    Ball noch ok, Roboter sehr unzuverlässig
  - Viele potentielle Verdeckungen bei 4 vs 4
- Weltmodellierung als SLAM-Problem

Weltmodellierung als SLAM-Problem

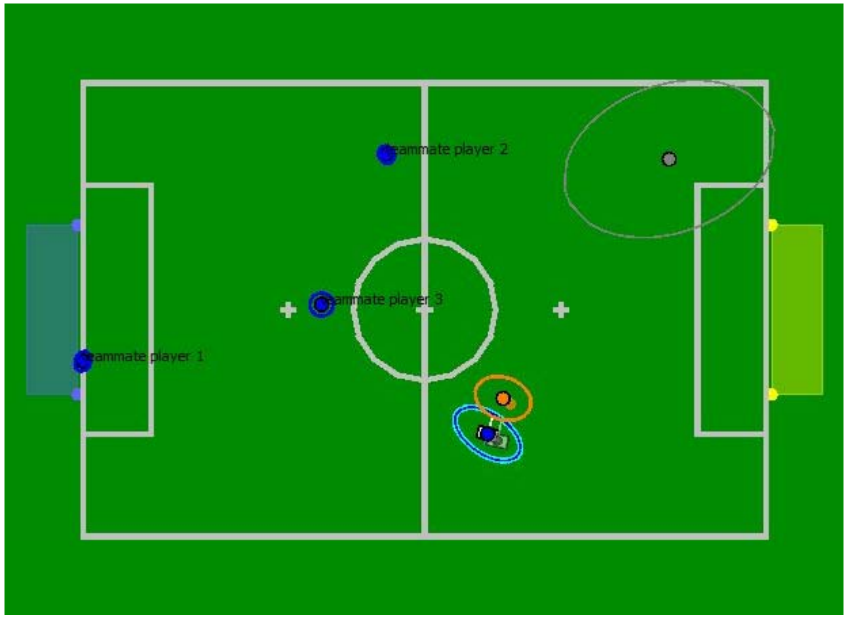
- Warum so kompliziert?!?
  - Lokalisierungsproblem? Gelöst.
  - Tracking üblicherweise einfach in lokalen Koordinaten
    - Ist effizienter
    - Ist „auserforscht“: Stationärer Beobachter trackt dynamisches Objekt

- Modellierung des vollständigen Zustandes hat Vorteile
  - Dynamische Elemente potentiell nützlich für Lokalisation
  - Geteilte Information kann mehrdeutige Lokalisierung auflösen
  - Odometriefehler ist in Positionsschätzung schon kompensiert und summiert sich nicht auf

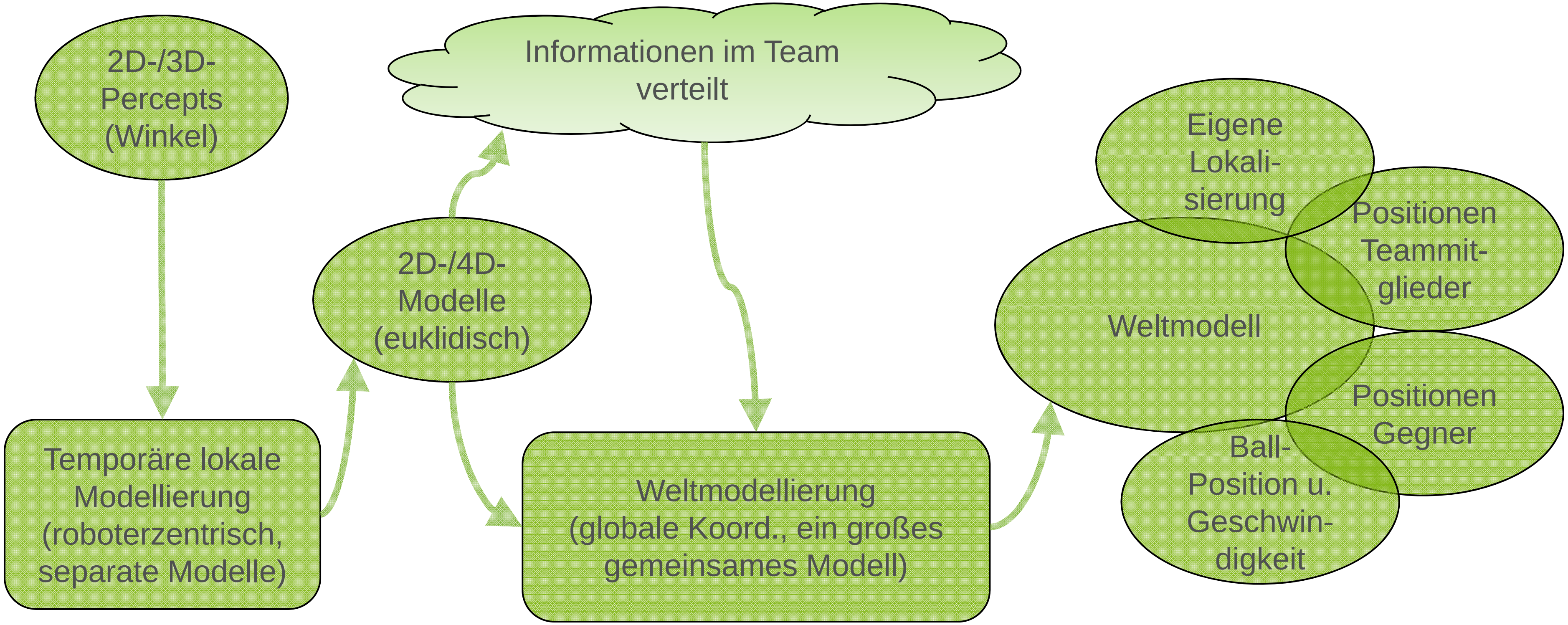
Weltmodellierung als SLAM-Problem



Weltmodellierung als SLAM-Problem



Kooperative Weltmodellierung



Was wir 2011 benutzt haben

- Basiert auf MultiUKF-Lokalisierung, ohne „S“ in SLAM
- Ist umgesetzt und wird seit den GermanOpen2011 benutzt
  - Funktioniert (aber noch keine quantitative Analyse)
  - Schnell genug
  - In manchen Situationen trotzdem noch lokales Modell nötig
    - Ballannäherung auf mittlere bis kurze Distanz
    - Ausweichen:
      eigentlich nicht gewünscht, da die Robotermodellierung am meisten von gemeinsamer Modellierung profitiert
  - Taktische Entscheidungen auf globalem Gegnermodell möglich

Part 2 – Kooperative Weltmodellierung

- Noch Fragen?   :-)