

# Handgefertigte „Objekt“verankerung

... oder: Ein Beispiel funktionierender Verankerung bei Robotern, das aber leider nicht auf Anderes übertragbar ist

## Das zeit-variable „Objekt“ $\text{Pose}(x,z,\beta)$ bzw. $\text{Pose}(x,y,z,\alpha,\beta,\gamma)$

- Eigentlich nicht Objektverankerung, sondern *Symbol Grounding* (abstraktes Konzept)
- Das gesamte Kapitel zur Lokalisierung behandelt Spezial-Verankерungsverfahren für das Pose-Prädikat!
- Man sieht daran, wie schwierig robuste Verankerung im Allgemeinen ist!  
(Perzeptionsfehler, Prädiktionsfehler, Aktionsfehler)

# Zusammenfassung Verankerung

- Objekte (Klassen, Instanzen) müssen in Sensordaten wurzeln, um wahrnehmbar zu sein
- („Affordanzen“ auch)
- (Abstrakta, Relationen, ... funktionieren vermutlich „irgendwie“ anders)
- Verankerung ist ein aktiver Prozess
- Es gibt bislang nur wenige Arbeiten und wenig Literatur aus der Robotik dazu

## 8.3 Semantische Kartierung

- Beim klassischen Bildverstehen will man genau die Daten verwerten, die man „sowieso“ da hat
- Weiter kommt man (möglicherweise), wenn man Daten aktiv sucht („Hier muss doch irgendwo ein Stuhl sein!“) und/oder vervollständigt („Die Wand geht hinterm Regal bestimmt weiter!“)
- Das ist ein Bootstrap-Prozess: Wenn ich die Szene schon ein wenig verstehe, könnte ich sie noch besser verstehen
- A priori-Verstehen muss durch Kontextinformation vorgegeben sein und/oder durch Objekterkennung „aus dem Stand“ erzeugt werden
- Eine Semantische Karte aggregiert konsistent die vorhandenen Daten und symbolischen Informationen

*Perception is controlled hallucination.* Max Clowes, 1971

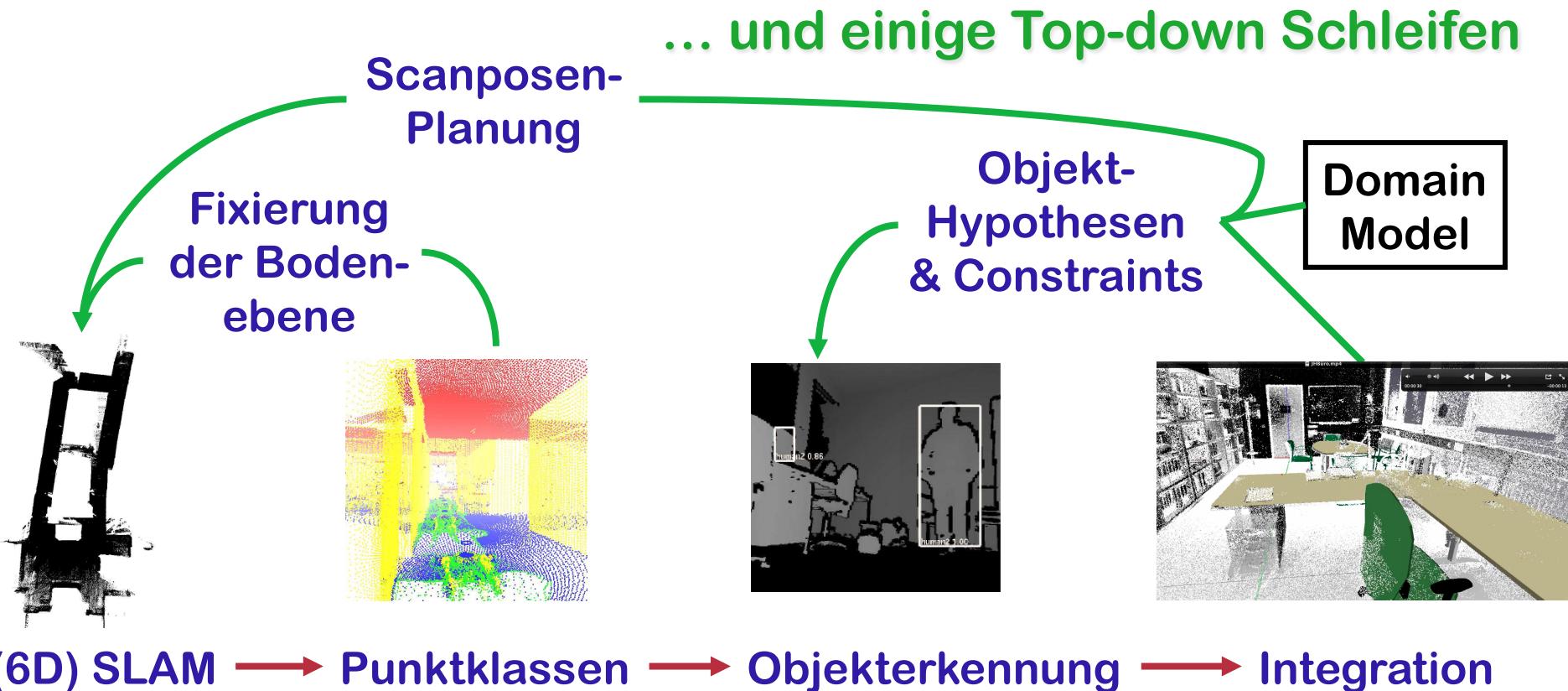
# Beispiel: Wieviele Stühle sehen Sie?

B. Neumann, R. Möller: On Scene Interpretation with Description Logics.  
Image Vision Comput. 26:82-101 (2008); frühere Fassung 2006



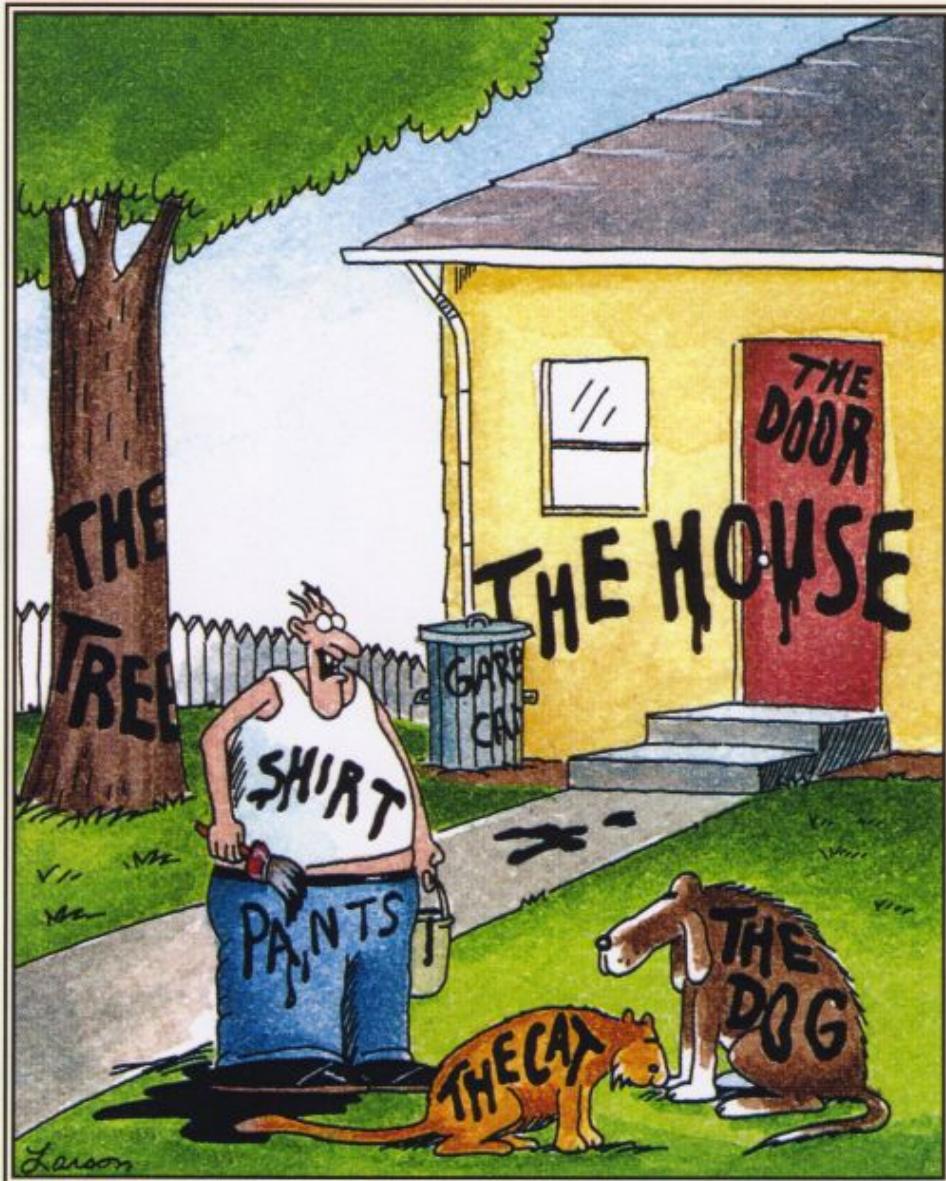
- *Cognitive Vision*
- „Abduziere“ potenzielle Aggregate aus erkannten Objekten und einer „Hintergrundtheorie“ in (Beschreibungs-)Logik
- Reinterpretiere Objekte
- Fülle Sensordaten über Schlussfolgerungen auf (Okklusionen auflösen)
- Generiere Erwartungen

# Objektklassen vs Semantische Kartierung



Die Bottom-up Kaskade ...

[Nüchter&Hertzberg, J.RAS 2008]



"Now! ... That should clear up a few things around here!"

## Merke ...

- ... Namensschilder zu vergeben (*bottom-up* Objektklassen, Objektindividuen) ist noch keine Semantische Kartierung!
- ... Es fehlt dafür eine (i.w.S.) logische Hintergrundtheorie ...
- ... und wenn man die hat, ändert sich der Prozess der Kartierung!

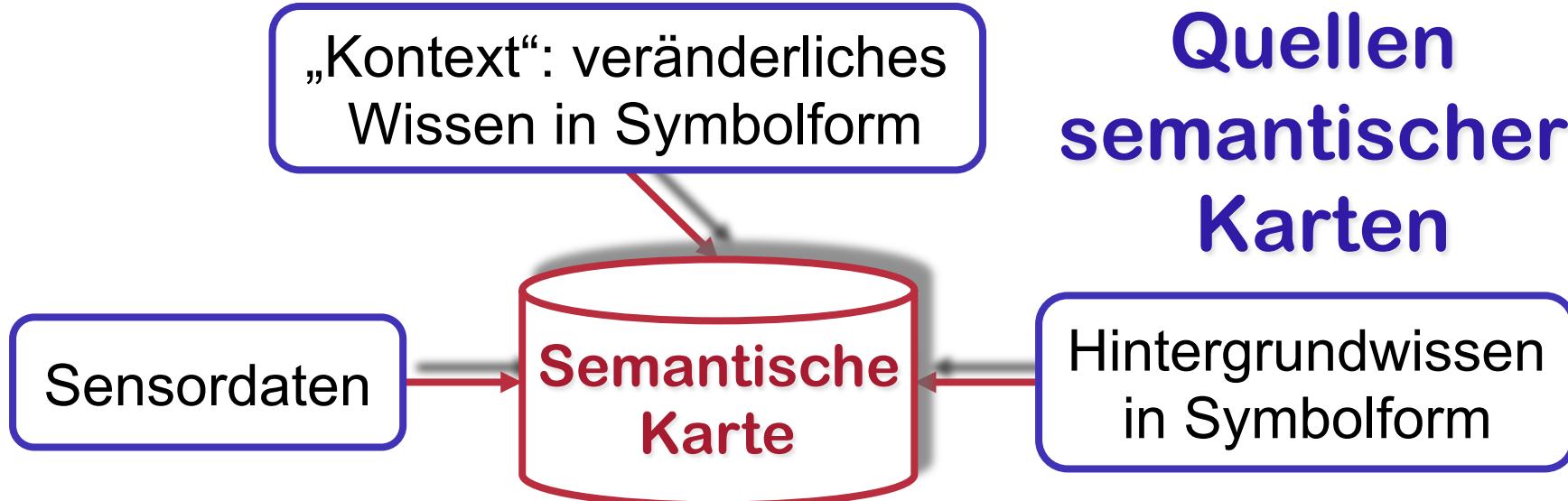
# Arbeitsdefinition

Eine **semantische Karte** ist eine Roboterkarte, die zusätzlich zu metrischen und geometrischen Informationen Zuordnungen von kartierten Strukturen und Objekten zu bekannten Konzepten und Relationen enthält.

Weiterhin umfasst die semantische Karte statisches Wissen über diese Konzepte und Relationen, das in einer Wissensbasis abgelegt ist.

Inferenzen sind möglich, welche die raumbezogene Information und das statische Wissen über Objekte in der semantischen Karte kombinieren.

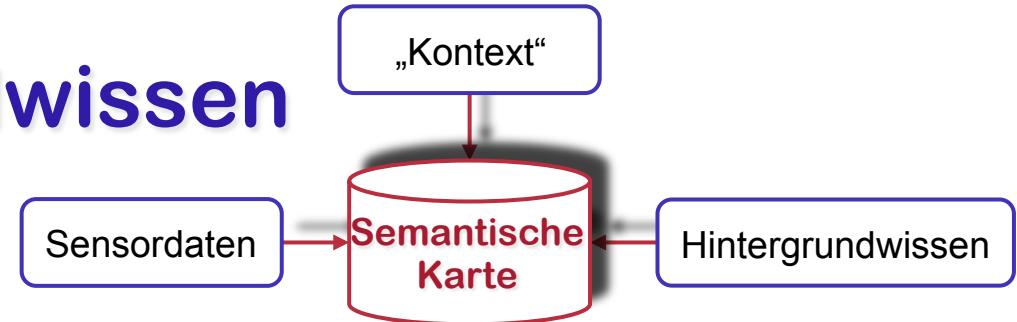
# Quellen semantischer Karten



Beim **Aufbau semantischer Karten, Semantischer Kartierung** werden sensorische + semantische Information genutzt, um

- vorhandene Sensorinformation auszuwerten (z.B. zu disambiguieren)
- weitere semantische Information zu „erschließen“ (die über die Sensorinformation hinausgeht, ihr ggf. teils widerspricht)
- gezielt neue Sensorinformation zu erwerben (z.B. Aufmerksamkeitssteuerung, „aktive“ Wahrnehmung)

# Kontext, Hintergrundwissen



## Kontext

Information, egal auf welcher Abstraktionsstufe, die nicht an sich in den Sensordaten enthalten ist, zusätzlich gegeben ist und sich räumlich/zeitlich auf die aktuelle Szene bezieht

**Beispiel:** Es ist Sonntag; Sie sind in einem Bürogebäude

## Hintergrundwissen

Wissensbasis in Symbolform, z.B. **Beschreibungslogik**, die Klassen elementarer Objekte/Ereignisse, zusammengesetzter Obj./Ereignisse, ihre Eigenschaften und Relationen beschreibt

**Beispiel:** Klassen: Decke, Boden; Aggregate: Zimmer; Ereignis: Frühstück (die Mahlzeit!); Raumrelationen: über, unter; andere Relationen: größer; Zeitrelationen: nach, vor,

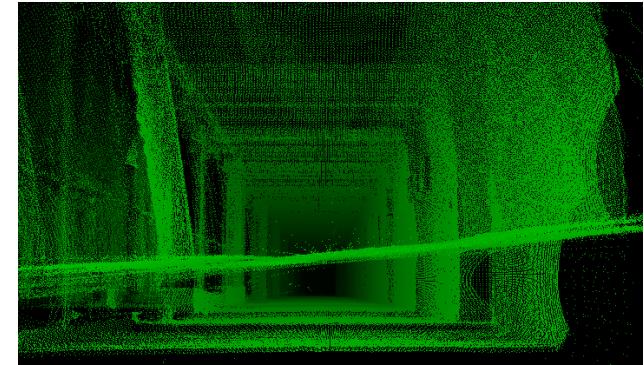
# Beispiel: Boden & Decke aus 3D-Punkten

- Ein Beispiel für **Bottom-up Dateninterpretation!**
- Ein autonom fahrender Roboter muss jederzeit wissen, wo der befahrbare Boden weitergeht!
- In Gebäuden oft vernachlässigt („befahrbar ist, wo kein Hindernis ist“ – Vorsicht bei Treppen!); im Freiland überlebenswichtig
- Ansatz hier:  
Extrahiere Bodenfläche aus 3D-Laserscans

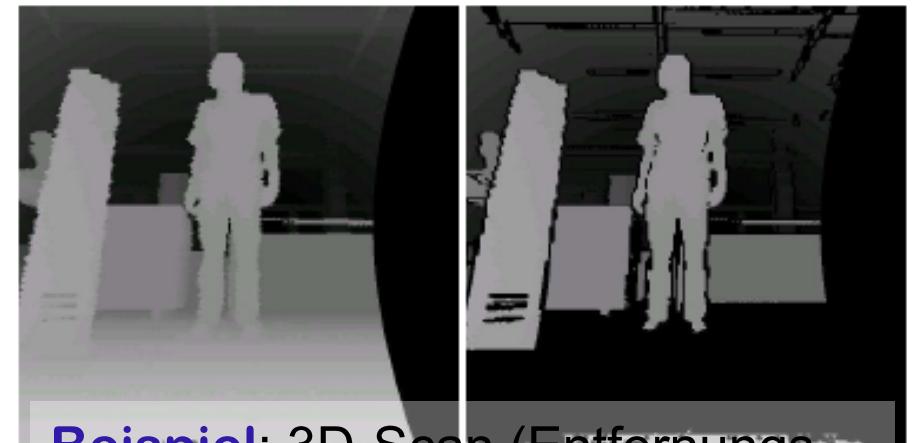


# Bodenerkennung hilft (top-down!) bei ...

- ... der online-Fahrplanung („Steure nur dahin, wo Bodenfläche an die aktuelle Position grenzt!“)
- ... der Registrierung von sequenziellen Scans in einer Szene („Boden im neuen Scan sollte Boden vom vorigen Scan fortsetzen“)
- ... der Komprimierung der Sensordaten („In Gebäudekarten speichere 1 Polygon statt x Mio 3D-Punkten!“)
- ... der Segmentierung von Objekten einer Szene („Wo endet ein Mensch nach unten?“)
- ... der Interpretation/  
Klassifizierung von Objekten („Was nicht auf dem Boden steht, ist vermutlich kein Stuhl“)



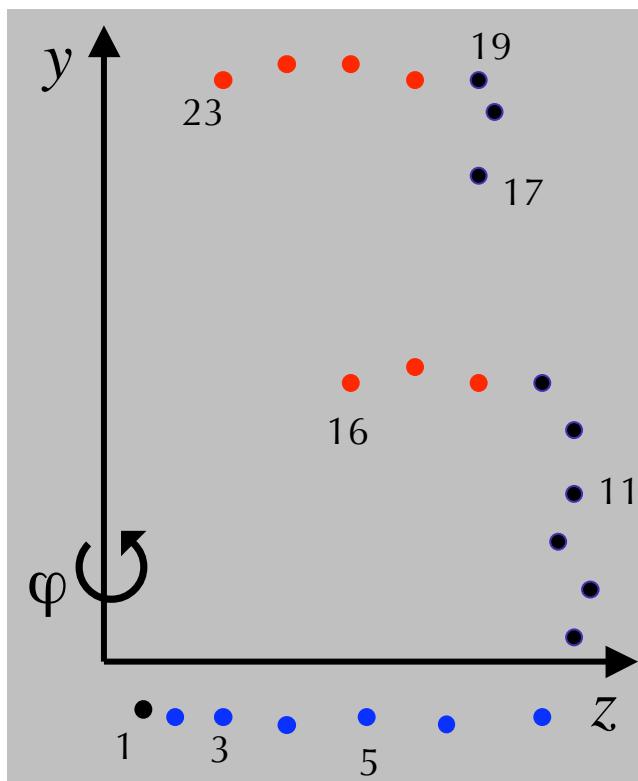
Falsch registriert!



**Beispiel:** 3D-Scan (Entfernungs-Grauwert) roh und mit Boden-Extraktion (+ Konturverstärkung)



## Vertikalscan



# Bsp. (Boden): Punktklassen

Wulf & al., ICRA-2004

Wenn der Boden keine Ebene ist ...

$p_{i,j} = (\varphi_i, z_{i,j}, y_{i,j})$  ist im  $i$ -ten Vertikalscan  
der  $j$ -te Punkt (beginne unten zu zählen)

Winkel zwischen  
Punkt ( $j-1$ ) und  $j$

$$\alpha_{i,j} = \text{atan} 2 \left( \begin{array}{c} z_{i,j} - z_{i,j-1} \\ y_{i,j} - y_{i,j-1} \end{array} \right)$$

„Fußbodenpunkte“  
flacher Winkel in  
Scanreihenfolge

$$|\alpha_{i,j}| < \tau$$

z.B.:  $\tau \approx 20^\circ \approx 0,111$  rad

„Deckenpunkte“  
flacher Winkel gegen  
Scanreihenfolge

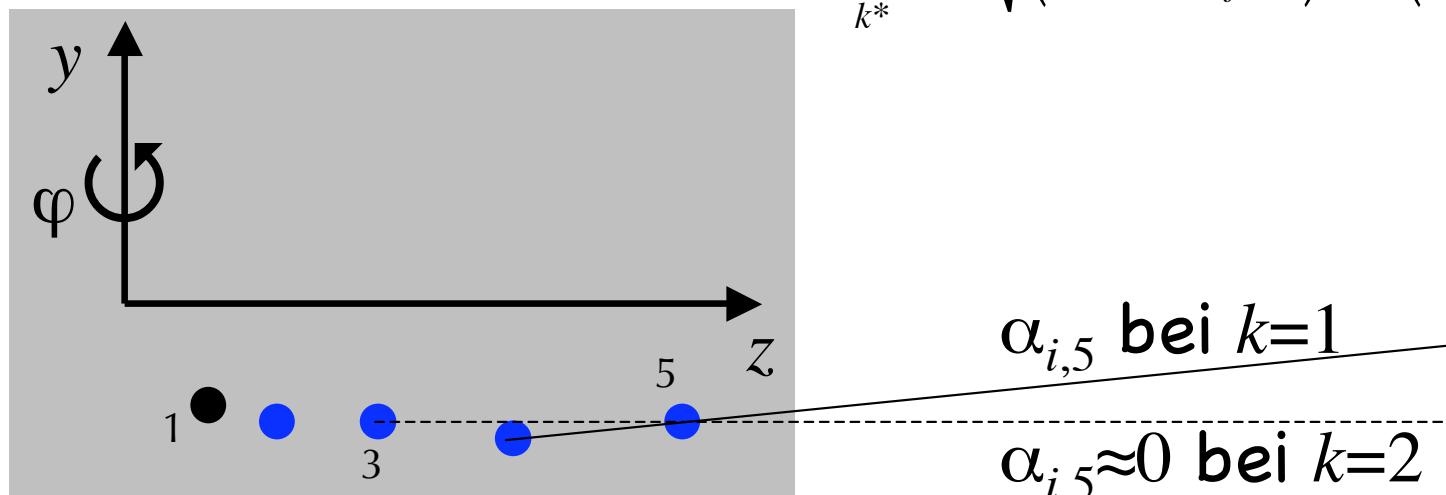
$$|\alpha_{i,j}| > \pi - \tau$$

# Bsp. (Boden): Rauschunterdrückung

Messfehler machen den Gradienten zwischen Nachbarpunkten fehlerhaft steil. Ausbügeln durch entfernteren Nachbarn  $k$ :

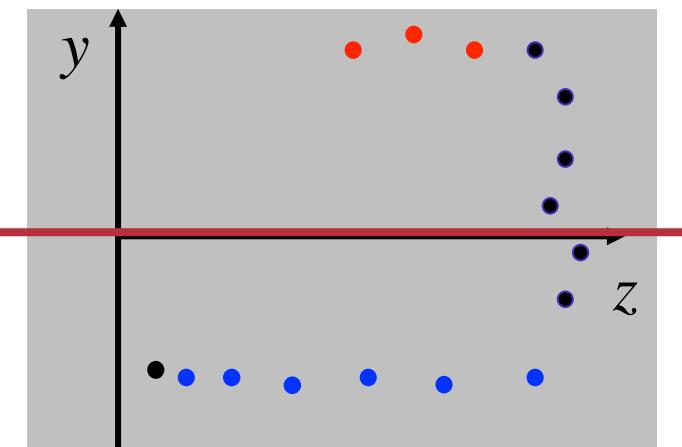
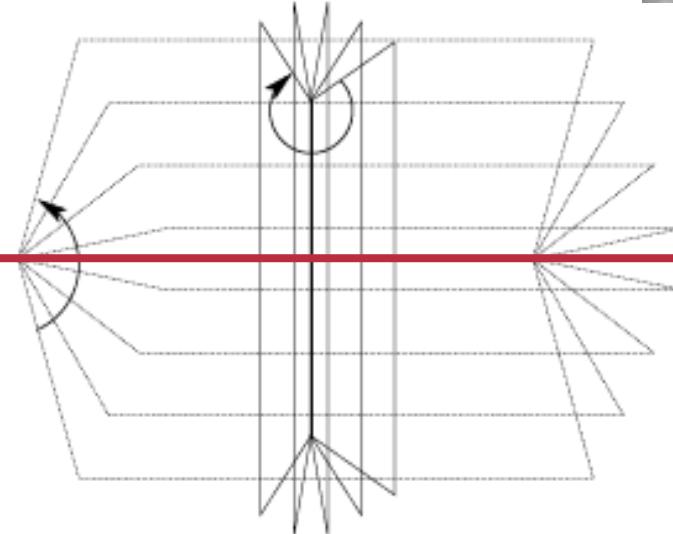
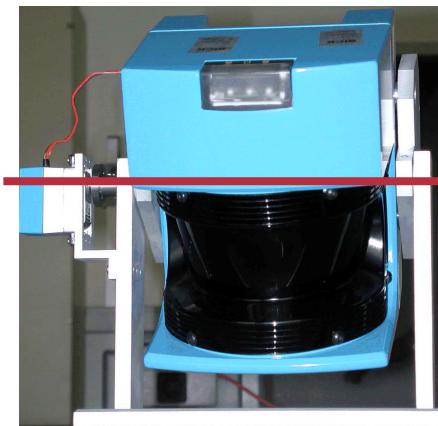
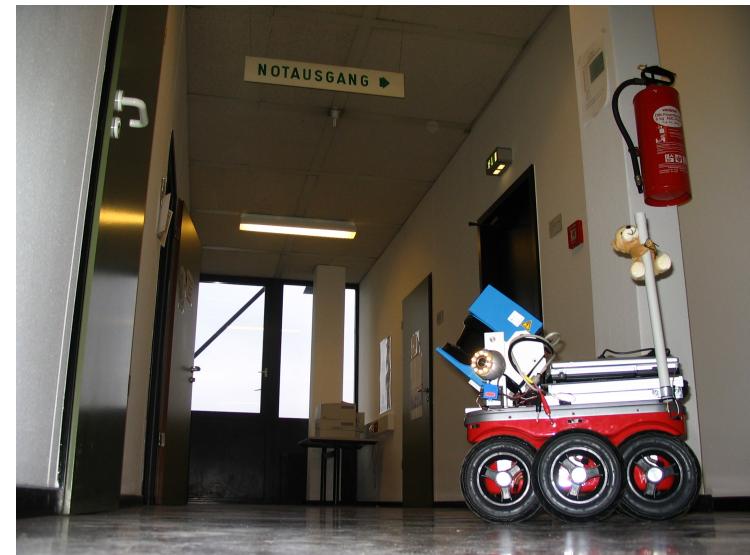
$$\alpha_{i,j} = \text{atan} 2 \begin{pmatrix} z_{i,j} - z_{i,j-k} \\ y_{i,j} - y_{i,j-k} \end{pmatrix} \quad \text{wobei}$$

$$k = \min_{k^*} : \sqrt{\left( z_{i,j} - z_{i,j-k^*} \right)^2 + \left( y_{i,j} - y_{i,j-k^*} \right)^2} > d_{\min}$$

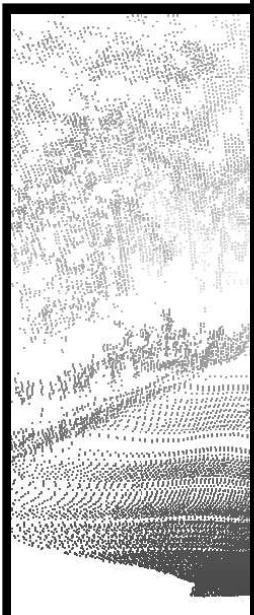


# Bsp. (Boden): Umformung in Vertikalscans

Wenn wir real einen nickenden 3D-Scanner haben: Transformiere horizontal-zylindrische in virtuelle vertikal-zylindrische Scans



Einzel-

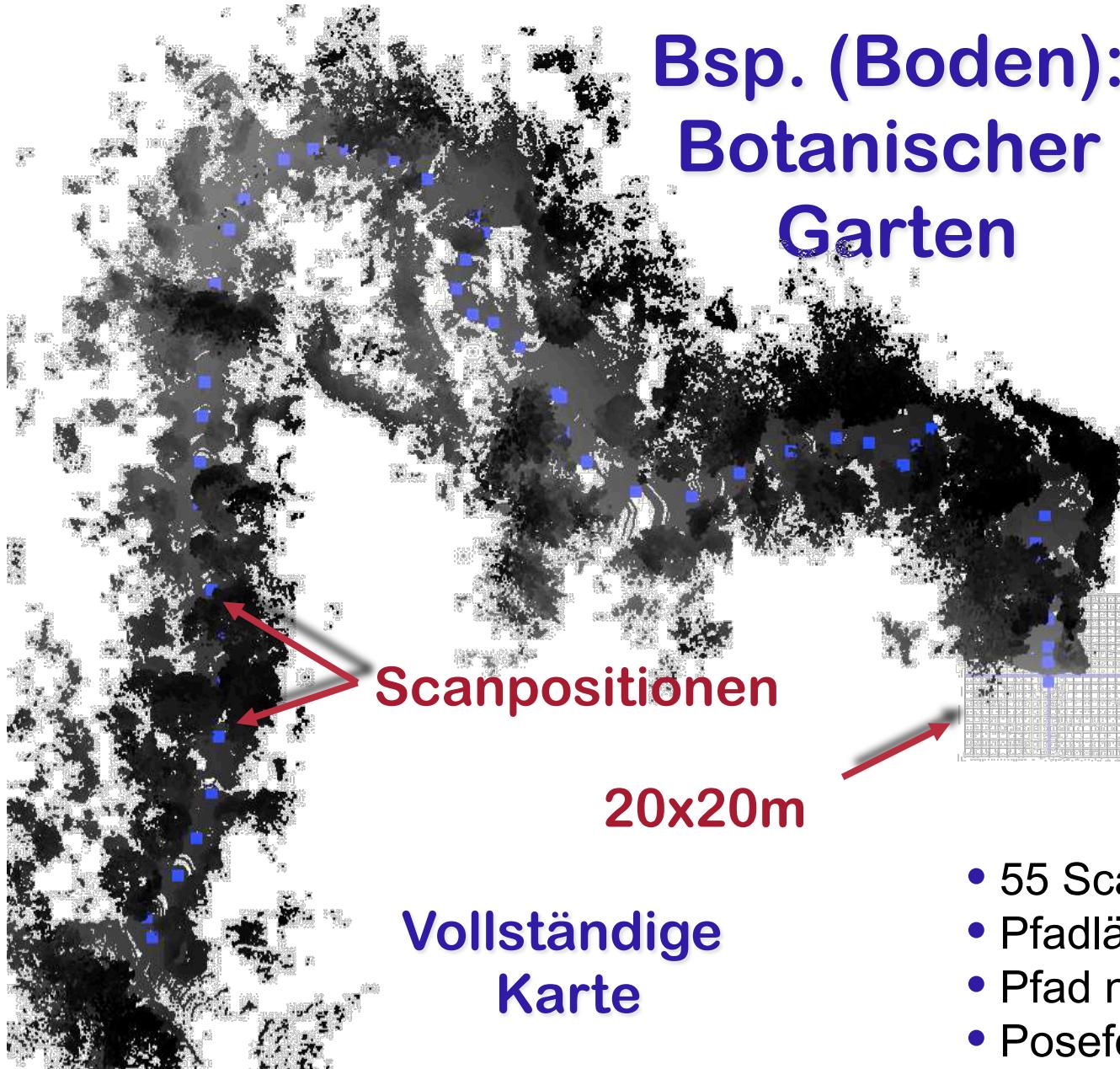


Roboter  
**Botanisc**  
gesteuert

Scans,  
Areale

blau

blaues  
e  
chte



- 55 Scans, je 85,000 Pkt.
- Pfadlänge ca. 150m
- Pfad nicht geschlossen
- Posefehler bzgl. Höhe