

Robotik I: Einführung in die Robotik

Umweltmodellierung

Tamim Asfour, Rüdiger Dillmann

KIT-Fakultät für Informatik, Institut für Anthropomatik und Robotik (IAR)
Hochperformante Humanoide Technologien (H²T)



■ Motivation: Adaptive Roboteraufgaben

■ Objektmodelle

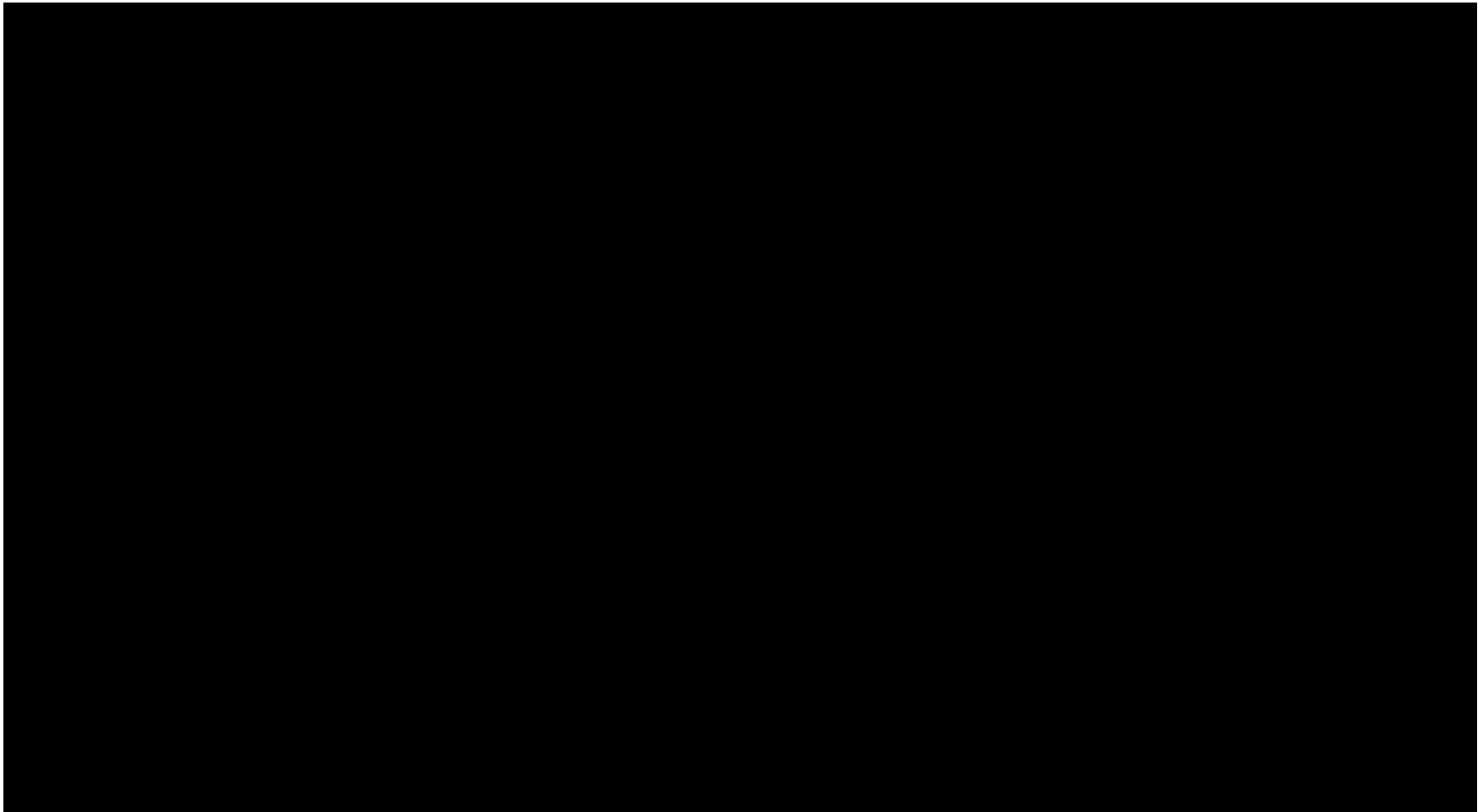
- Geometrische Beschreibung
- Zusätzliche Eigenschaften

■ Szenenmodelle

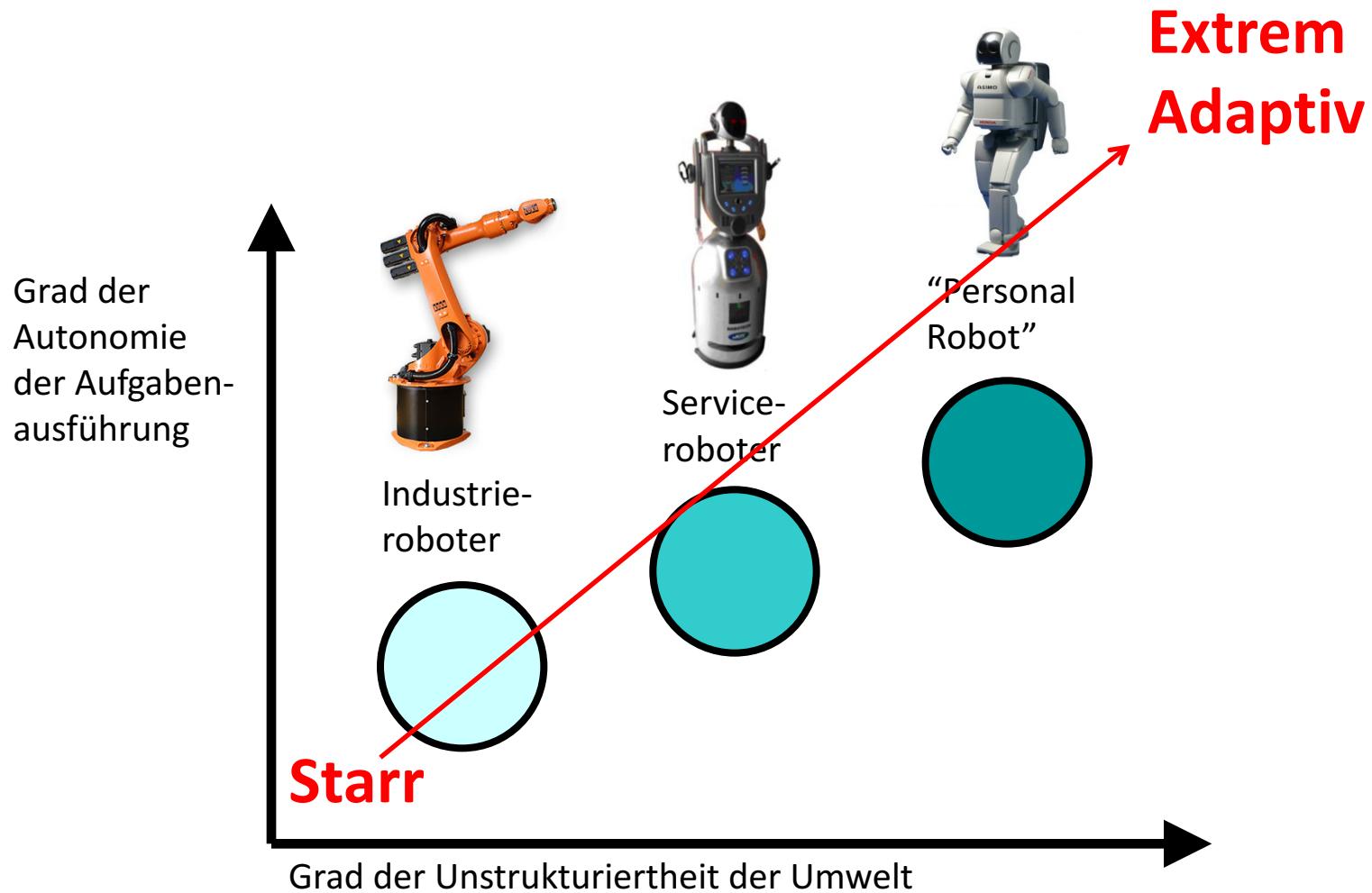
Bisher: Starre Roboter Aufgaben

- **Bisher in VL: un- oder marginal flexible Roboter Aufgaben**
 - Feste Trajektorie programmiert:
 - Positionsregelung
 - Eventuell lokale Kraftregelung
 - Benötigte Modelle für Programmierung und Regelung:
 - Kinematisches Modell
 - Dynamisches Modell
 - Benötigte Sensoren:
 - Gelenkencoder
 - (eventuell) Kraft-Momenten-Sensor
- Bahn absolut fest oder nur lokal-adaptiv, repetitiv ausgeführt

Starre Roboteraufgaben



Adaptionsanforderungen



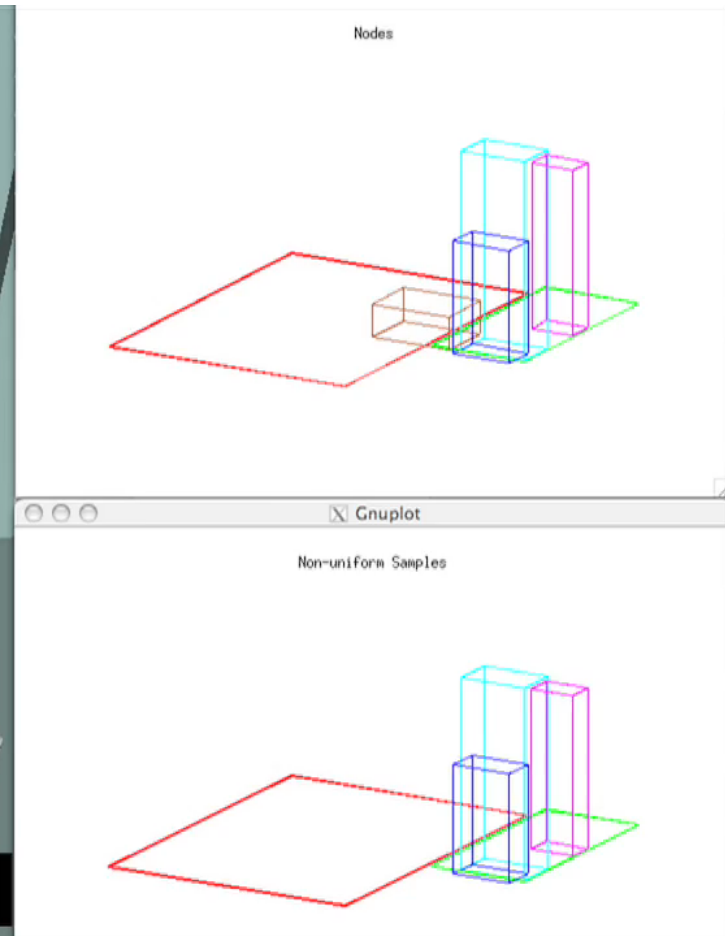
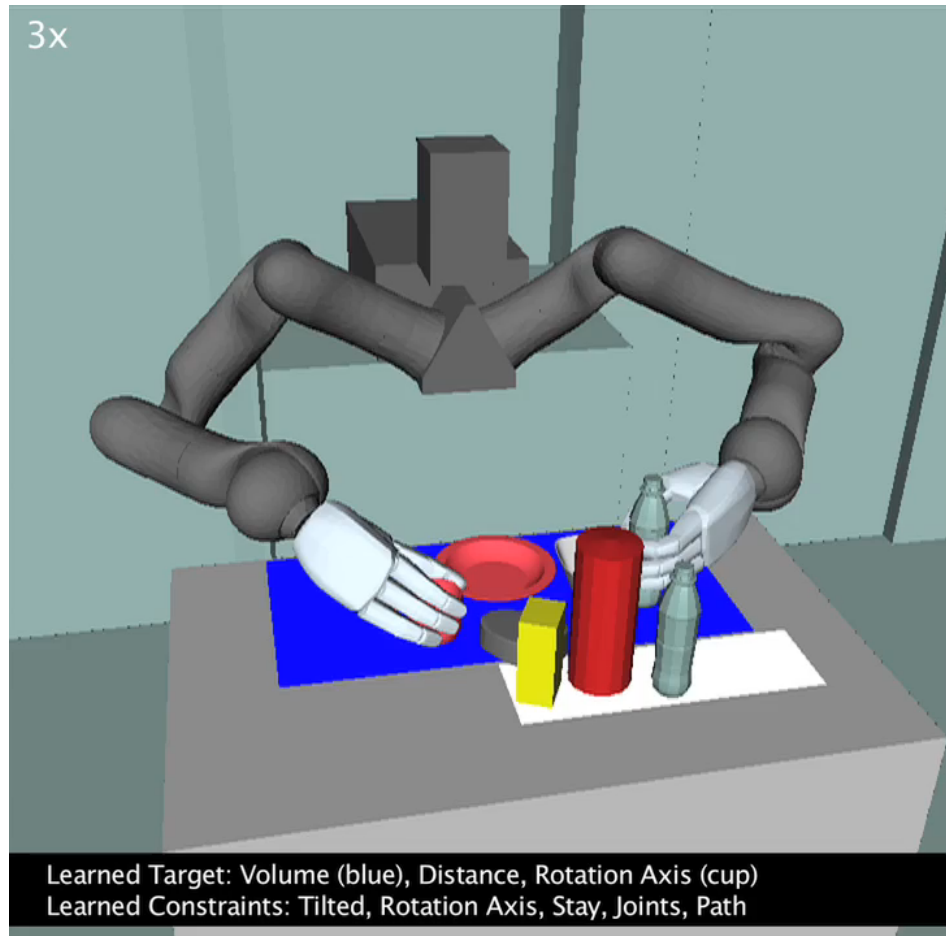
Adaptive Roboteraufgaben

Bei adaptiven Roboteraufgaben andere Voraussetzungen!

- Keine feste Trajektorie programmierbar
 - Trajektorie muss flexibel vom Roboter bestimmt werden
 - Benötigte Modelle für Programmierung und Regelung:
 - Kinematisches Modell
 - Dynamisches Modell
 - **+ Umweltmodelle**
 - **+ Aufgaben/Planungsmodelle**
 - Benötigte Sensoren:
 - Gelenkencoder
 - (eventuell) Kraft-Momenten-Sensor
 - **+ visuelle und eventuell taktile Sensoren**
- Bahn völlig frei, flexibel, nicht-repetitiv ausgeführt

Adaptive Roboteraufgaben

Bei adaptiven Roboteraufgaben andere Voraussetzungen!



Anforderungen für Adaption

Zusätzliche Anforderungen gegenüber starren Aufgaben

1. Visuelle und (eventuell) taktile Sensorik zur Umwelterfassung:

- *Zentrales Thema der Vorlesung Robotik 3 im SoSe*

2. Datenrepräsentation der Umgebung/Umwelt:

- *Diese Vorlesung*

3. Planungsmethoden zur Aufgaben/Bewegungsberechnung

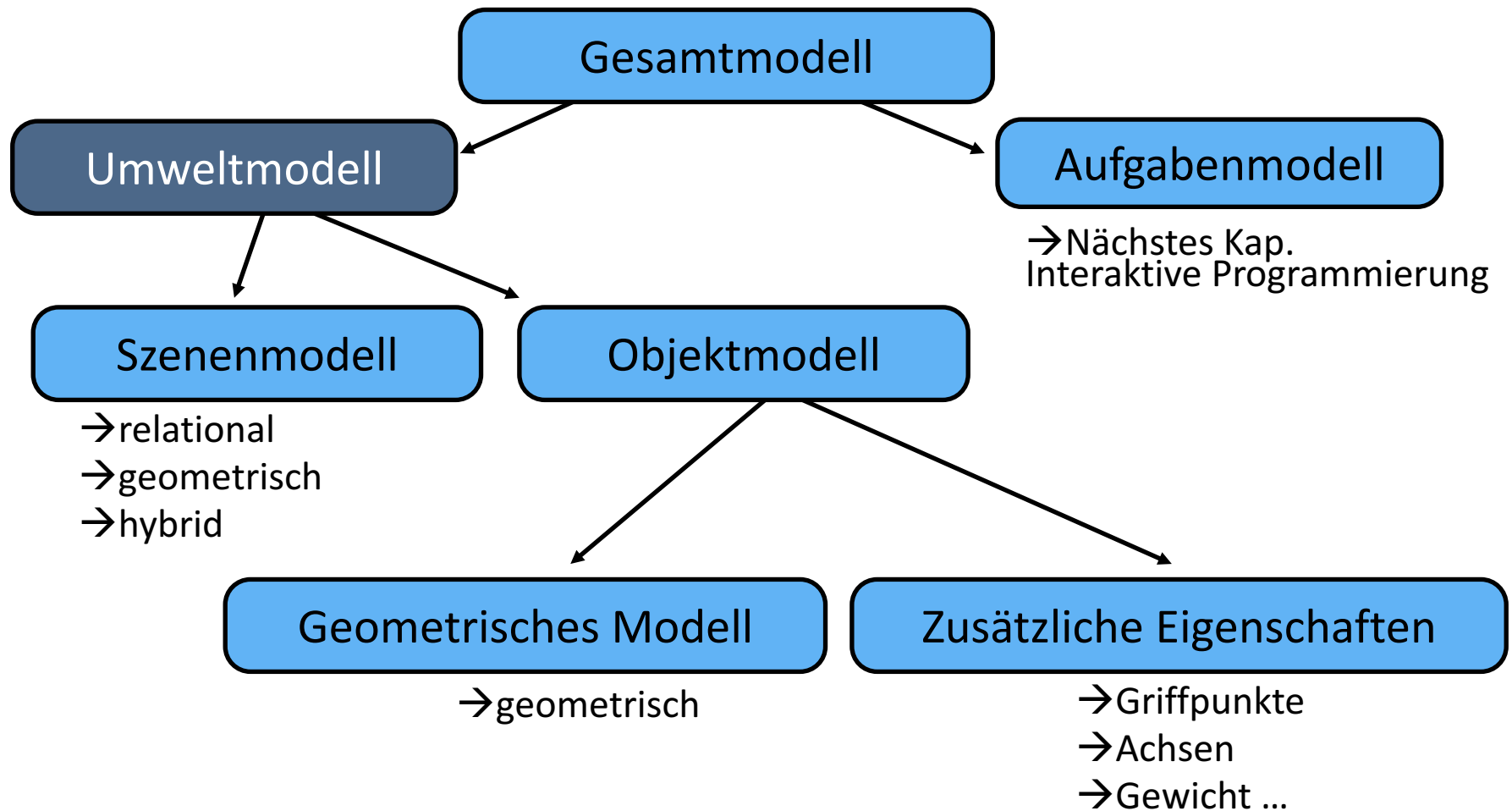
- *Robotik 1*
 - *VL9: Bahnplanung*
 - *VL10: Greifplanung*
 - *VL12: Interaktive Programmierung*

Übersicht Modelle

Verschiedene Arten von Modellen in der Robotik

1. Modell der Kinematik (Manipulator, mobile Plattform)
2. Modell der Dynamik (Manipulator, mobile Plattform)
3. Modelle der Sensoren
4. Umweltmodell (Umgebung: Einzelobjekte und Szenen)
5. Aufgabenmodell (Bewegungen und abstrakte Aufgaben)

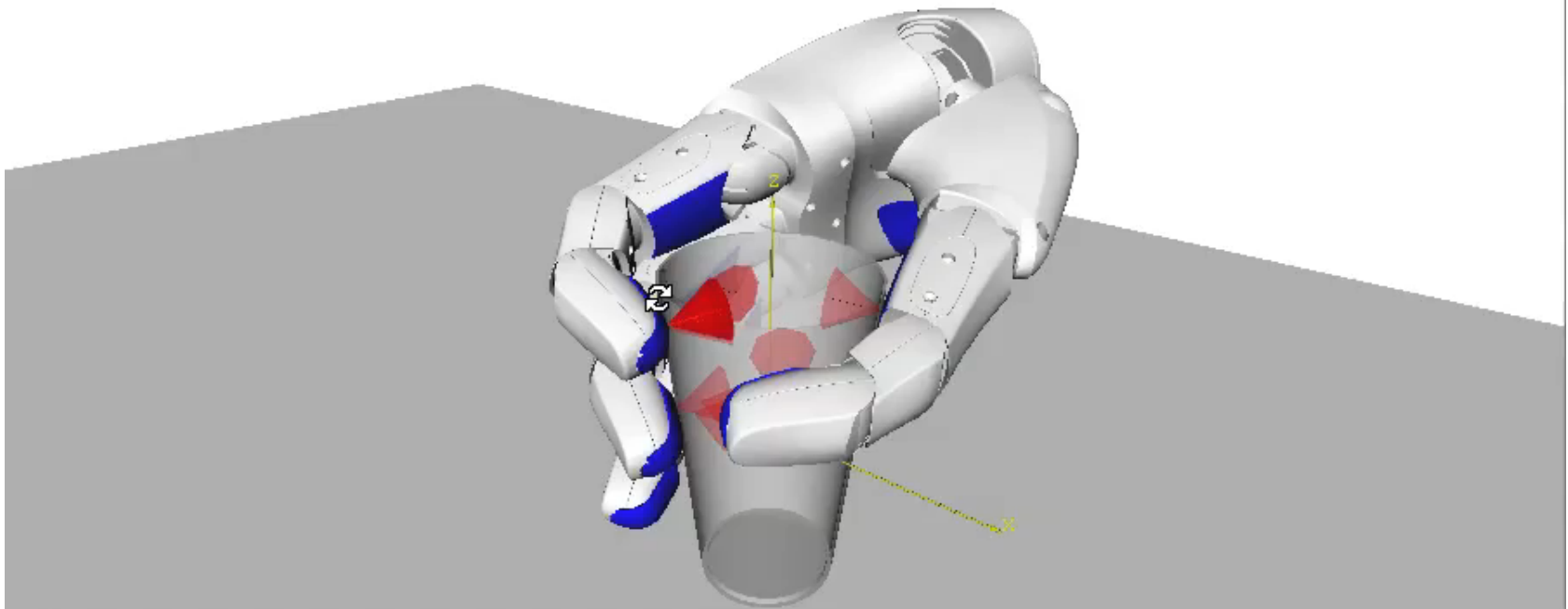
Übersicht Umweltmodellierung



Inhalt

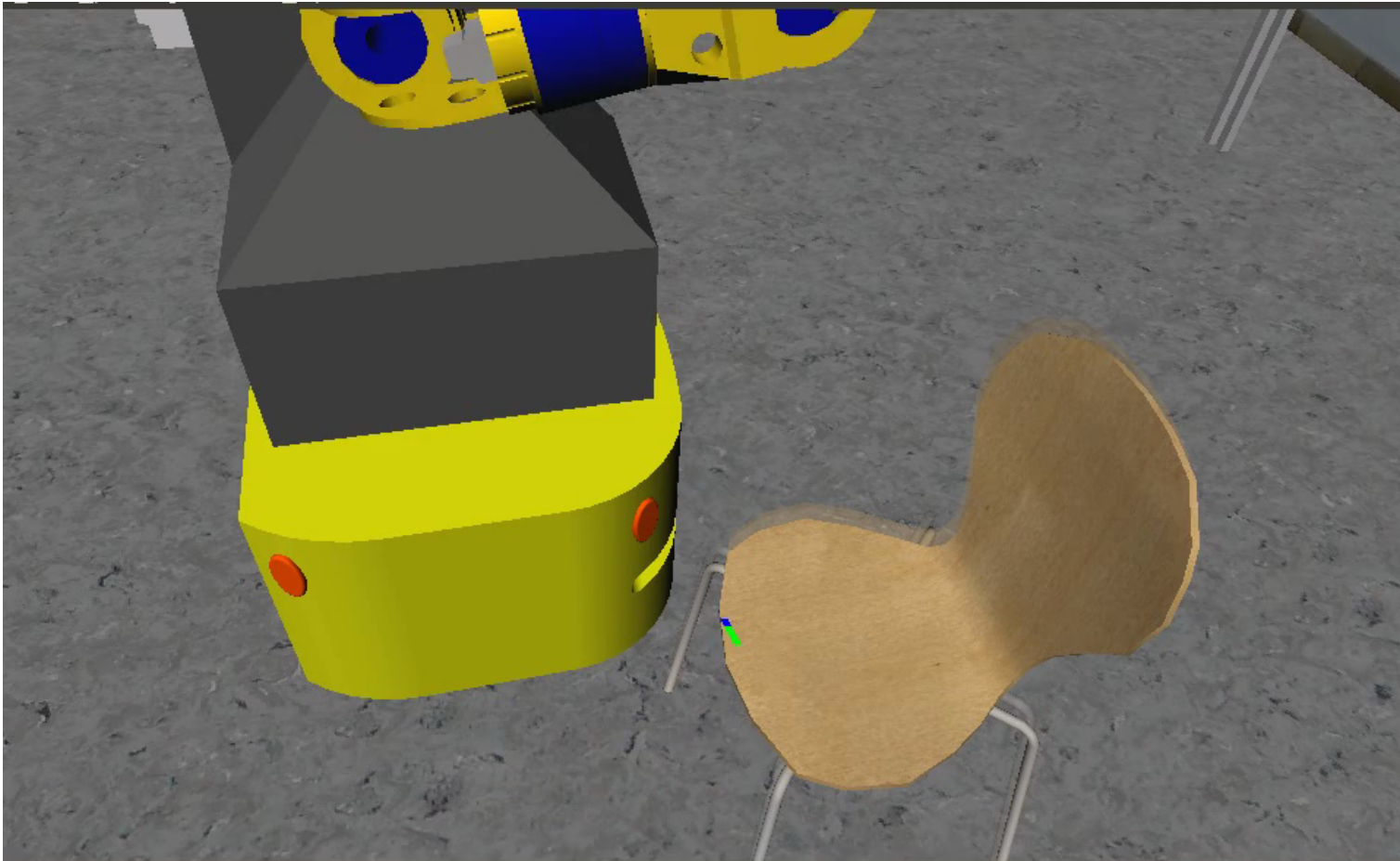
- Motivation: Adaptive Roboteraufgaben
- **Objektmodelle**
 - **Geometrische Beschreibung**
 - Zusätzliche Eigenschaften
- Szenenmodelle

Geometrisches Modell: Motivation



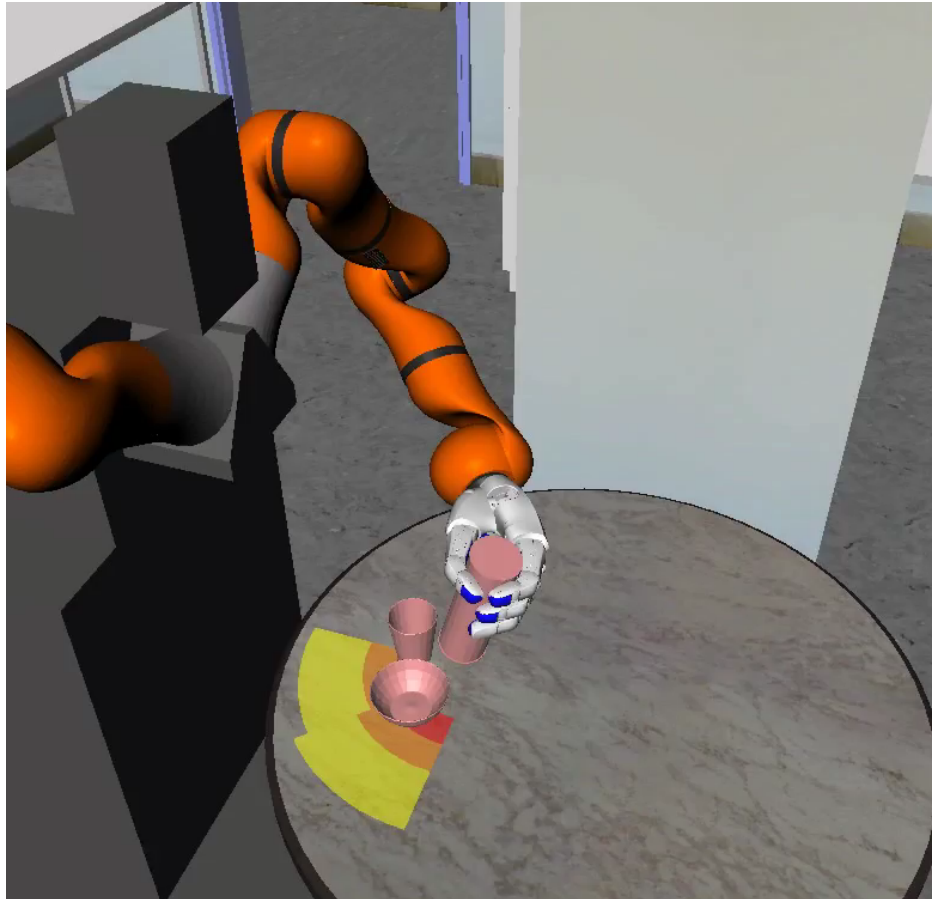
Geometrisches Modell: Motivation

Simulation der Effekte von Handlungen auf die Umwelt



Geometrisches Modell: Motivation

Simulation der Effekte von Handlungen auf die Umwelt



Geometrisches Modell: Motivation

Anwendungsgebiete für das geometrische Modell

- Umweltwahrnehmung (Live)
 - Objektklassifizierung
 - Objektlokalisierung
 - Bewegungserkennung/klassifikation des Menschen
- Bewegungsplanung (Live, offline)
 - Pfadplanung (mobile Systeme)
 - Bewegungsplanung (Manipulatoren)
 - Greifplanung und Planung von Inhandmanipulationsaufgaben
- Abstrakte Aufgabenplanung (Live, offline)
- Dynamische Effektsimulation und –prädiktion (Live, offline)
- Arbeitsraum/Anlagenplanung (Offline)

Übersicht Volumenmodelle

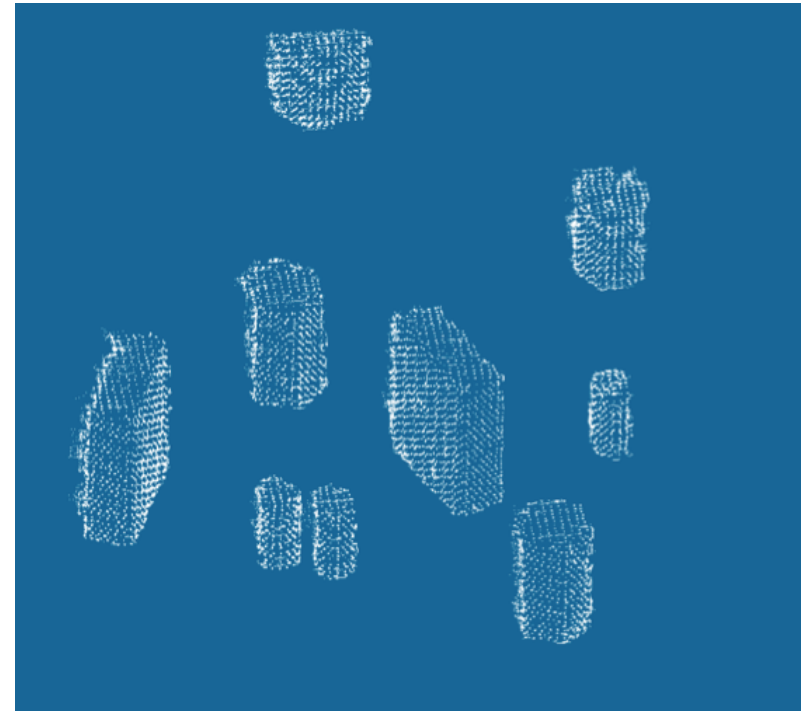
- Punktwolken
- Approximative Oberflächenmodelle (B-Rep)
 - Dreiecksflächen (Meshes)
 - Vierecksflächen
 - Bezierflächen
- Approximative Zellenbelegung
 - Voxel
 - Octree
- Analytisch-parametrische Modelle
 - Constructive Solid Geometry (CSG)

Punktwolken

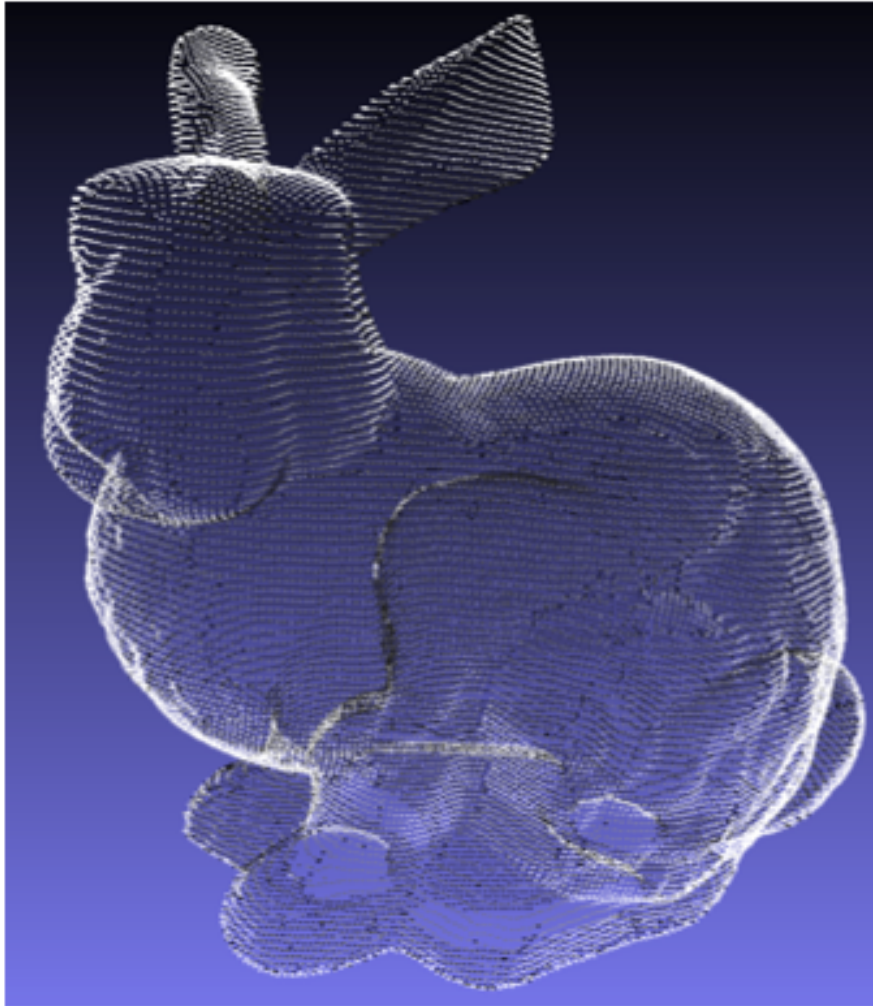
Punktwolken (Point Clouds) sind das einfachste Objektmodell

- Modell ist eine beliebige Menge aus Raumpunkten
- Daten direkt aus aktuellen Tiefensensoren (3D-Sensoren)
- Objekte nur implizit, indirekt modelliert
- Mehrdeutigkeiten
- Hochgradig approximativ
- Datenlücken durch Modellform
- Grundlage für weitergehende Objektmodelle
 - z.B. Erkennung/Lokalisierung von Objekten im Raum

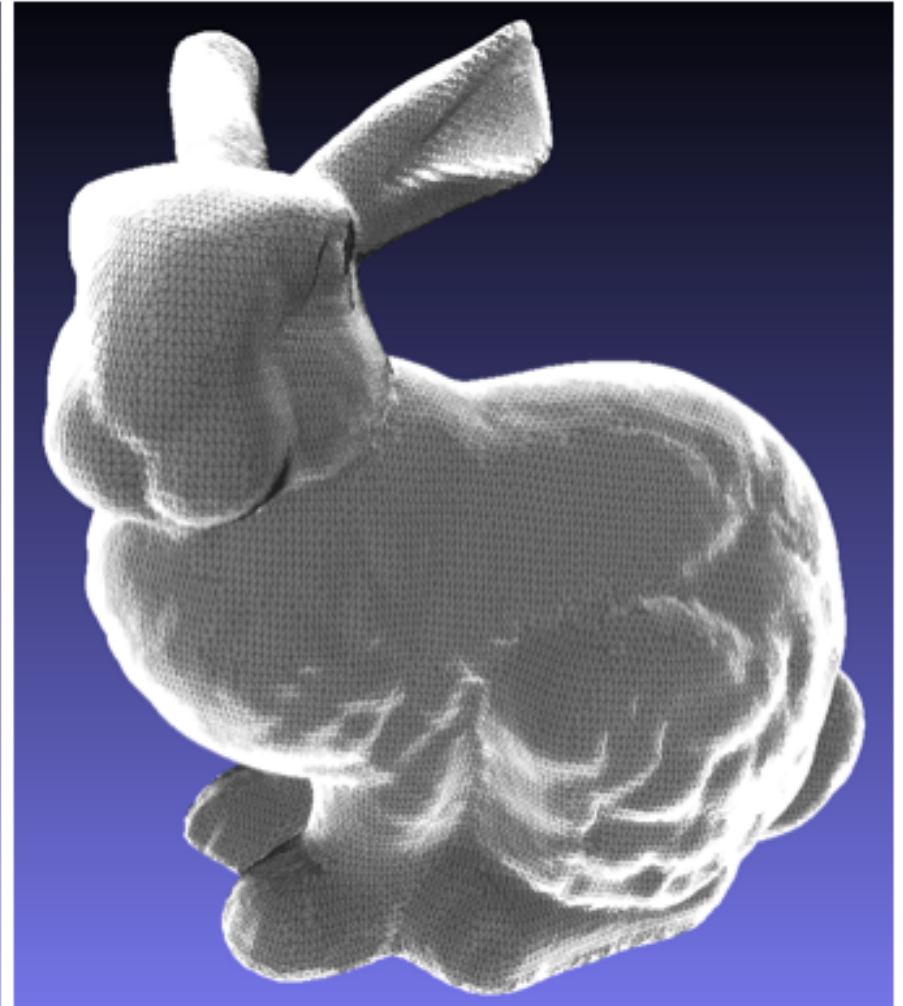
Punktwolken (Beispiele)



Punktwolken (Beispiele)



(a) Stanford Bunny dargestellt mit Punkten



(b) Stanford Bunny dargestellt mit Flächen

Approximative Oberflächen

- Bildung einer großen Fläche aus einem Netz („Mesh“) von einfachen Einzelflächen, z.B. Dreiecke, Vierecke

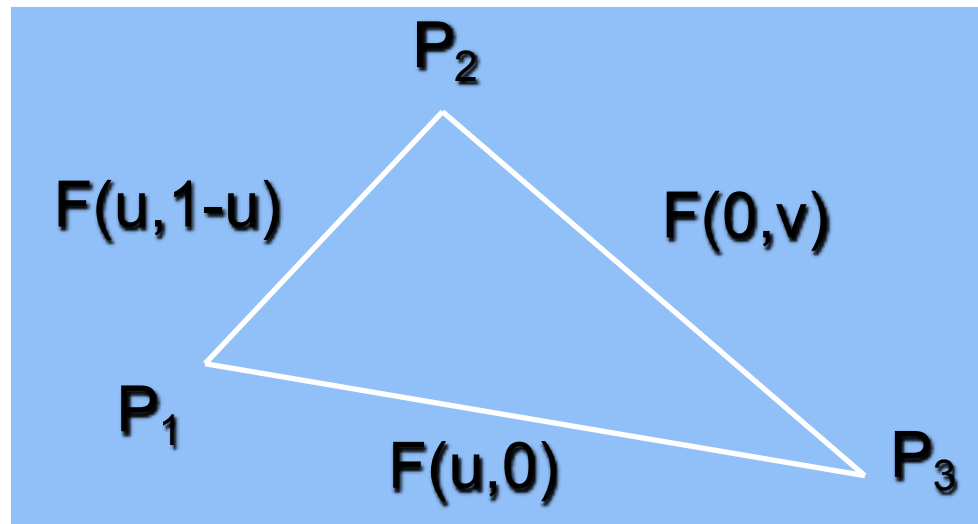
- **Vorteile:**
 - + Definition sehr einfach
 - + einfache Algorithmen

- **Nachteile:**
 - hoher Speicherbedarf
 - hoher Rechenaufwand

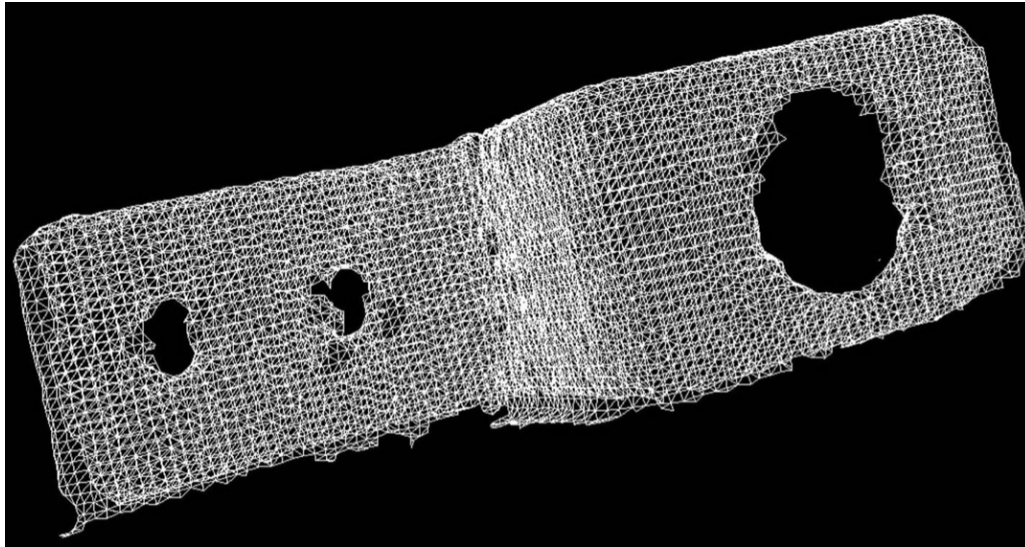
Dreiecksflächen

- Approximation von Freiformflächen: Die einfachste Fläche ist die Dreiecksfläche.
- Definition: Gegeben seien 3 Punkte im Raum P_1, P_2, P_3 . Damit hat die Fläche folgende Gleichung:

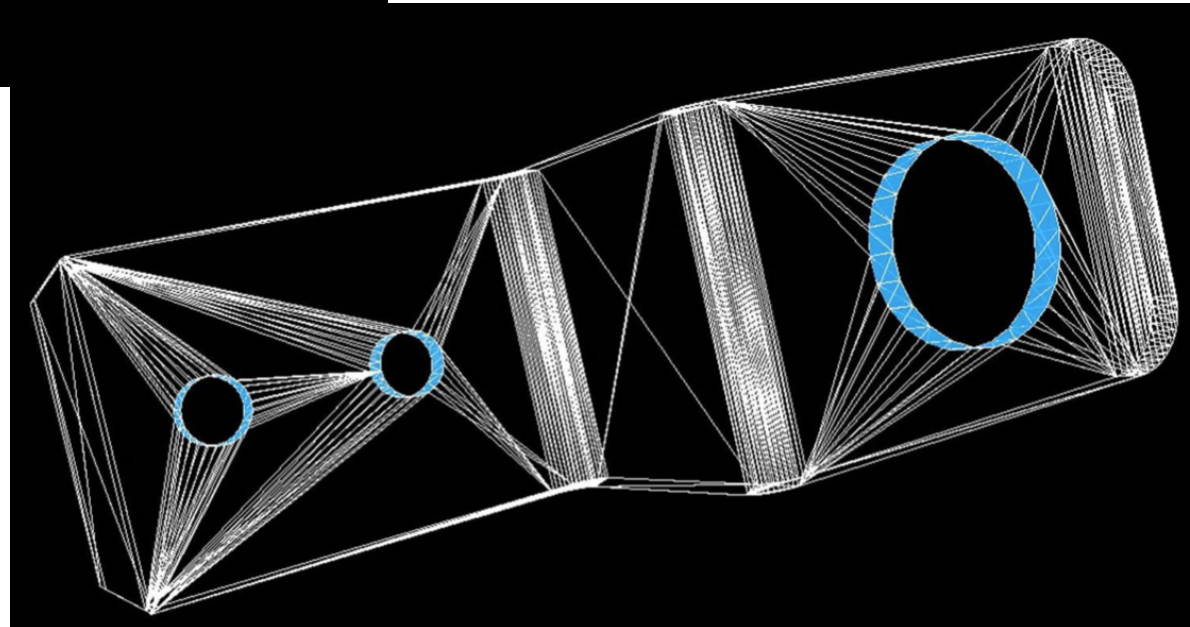
$$F(u, v) = u \cdot P_1 + v \cdot P_2 + (1 - u - v) \cdot P_3 \text{ mit } 0 \leq u, v, u + v \leq 1.$$



Erzeugung von Dreiecksflächen: Beispiel

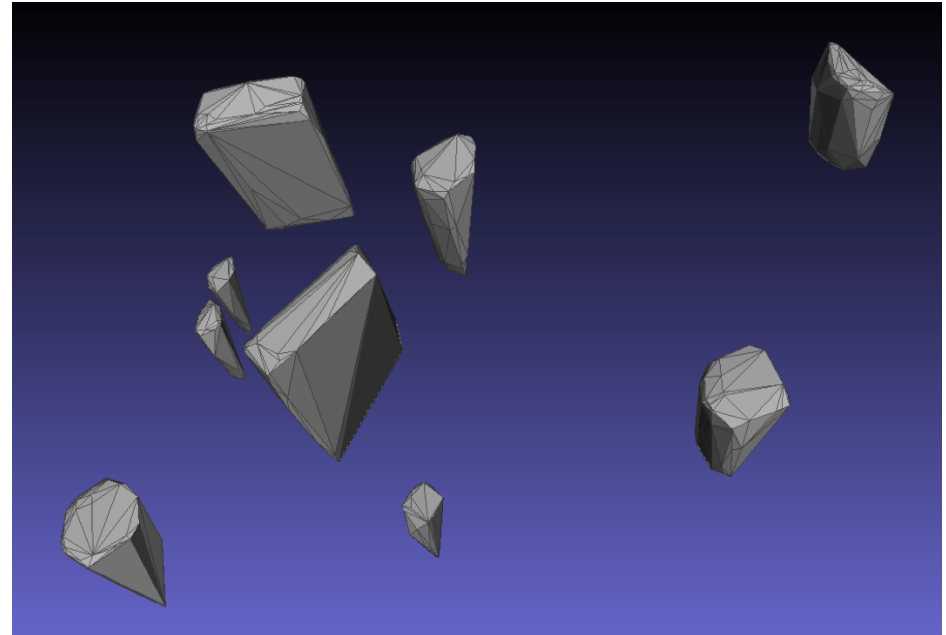


Mesh aus
Laserdaten
(Punktwolke)
rekonstruiert



Mesh aus
CAD-Modell
(analytisches Modell)
generiert

Erzeugung von Dreiecksflächen: Beispiel



Rekonstruktion aus Punktwolken

Bilineare Vierecksflächen

- Definition: Gegeben sind 4 Punkte im Raum P_1, P_2, P_3, P_4 . Damit wird die Fläche definiert durch:

$$F(u, v) = (1 - u)(1 - v) \cdot P_1 + (1 - u)v \cdot P_2 + u(1 - v) \cdot P_3 + uv \cdot P_4$$

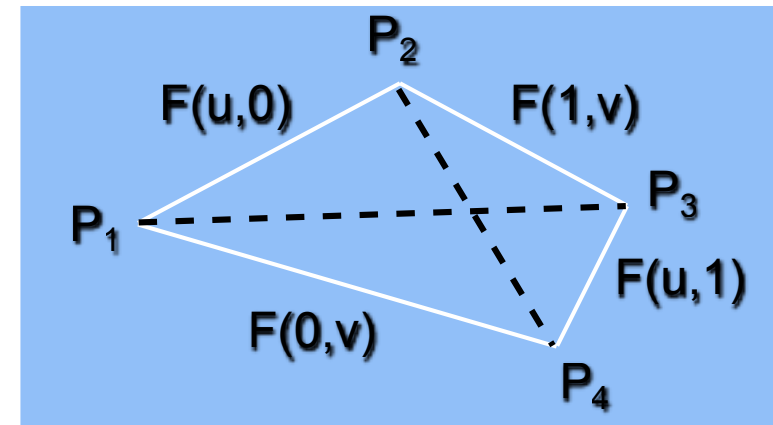
mit $0 \leq u \leq 1, 0 \leq v \leq 1$.

- **Vorteil:**

- + Flächenelemente können gekrümmt sein
- ⇒ weniger Gitterpunkte bei gleich guter Approximation

- **Nachteil:**

- Rechnen mit gekrümmten Flächen ist aufwendig



Bezierflächen

Erweiterung des Ansatzes der Bezierkurven

- Definition:
Gegeben ist ein Gitter
von Führungspunkten P_{ij}
 $0 \leq i \leq N$ und $0 \leq j \leq M$

- Damit ist die Fläche
beschrieben durch

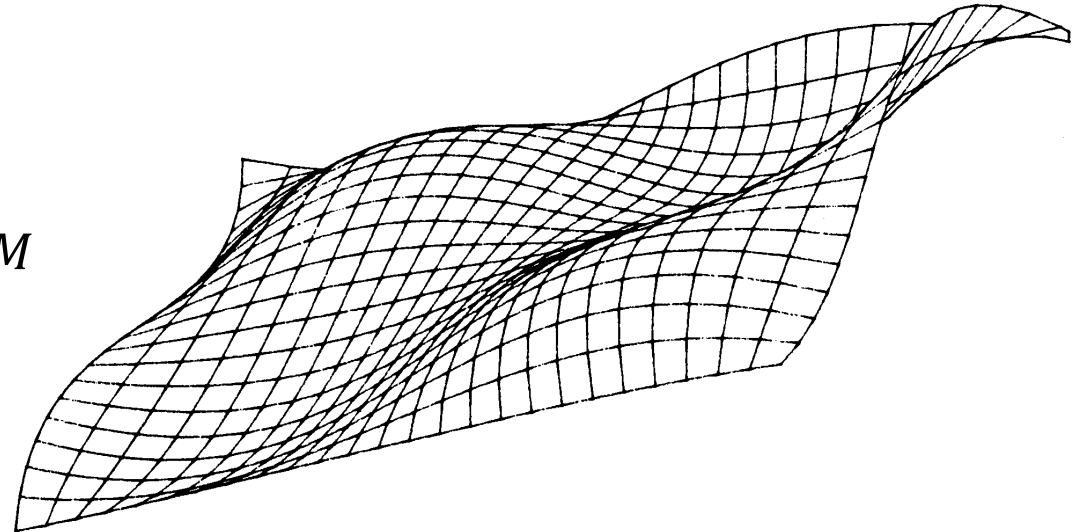
$$F(u, v) = \sum_{i=0}^N \sum_{j=0}^M P_{ij} \cdot B_{i,N}(u) \cdot B_{j,M}(v)$$

mit

$$B_{i,N}(u) = (1 - u)B_{i,N-1}(u) + u \cdot B_{i-1,N-1}(u)$$

$$B_{j,M}(v) = (1 - v)B_{j,M-1}(v) + v \cdot B_{j-1,M-1}(v)$$

Die $B_{i,N}$ bzw. $B_{j,M}$ heißen auch Bernsteinpolynome.



Inhalt

■ Motivation: Adaptive Roboteraufgaben

■ Objektmodelle

■ Geometrische Beschreibung

■ Kantenmodell

■ Flächenmodell

■ Volumenmodell

■ Zusätzliche Eigenschaften

■ Szenenmodelle

Kantenmodell

Reines Kantenmodell ohne weitere Informationen

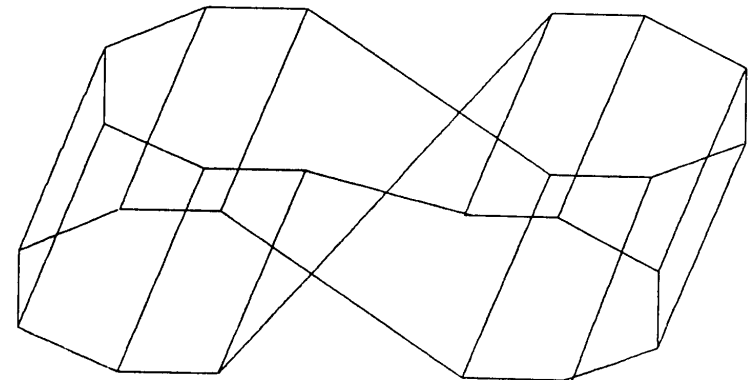
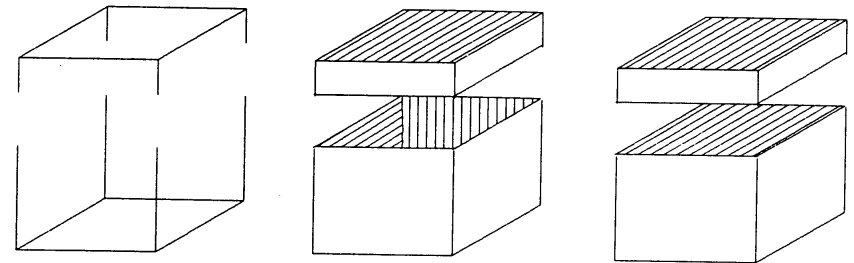
- Nur die Kanten werden gespeichert, d.h. Punkte und Verbindungen (Gerade, Polygonzug, Bezierkurve, ...)

- **Vorteile:**

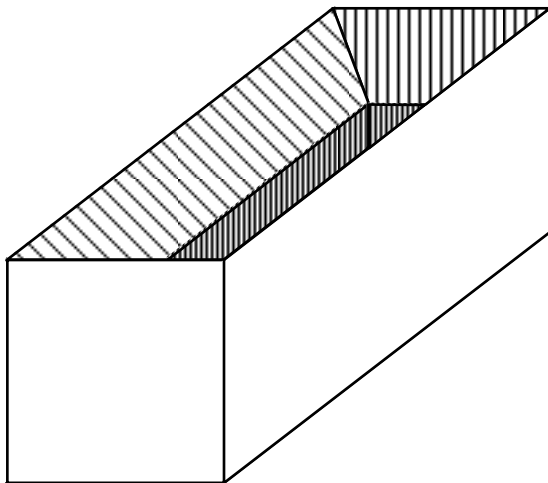
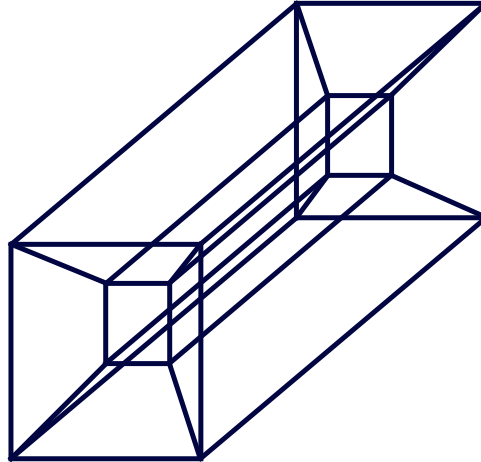
- + einfache Daten
- + wenige Daten

- **Nachteile:**

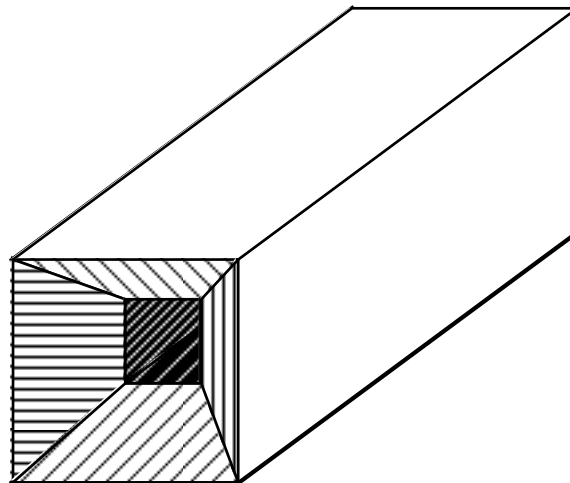
- Mehrdeutigkeiten
- hoher Eingabeaufwand
- keine Kollisionsberechnung
- kein Schnitt



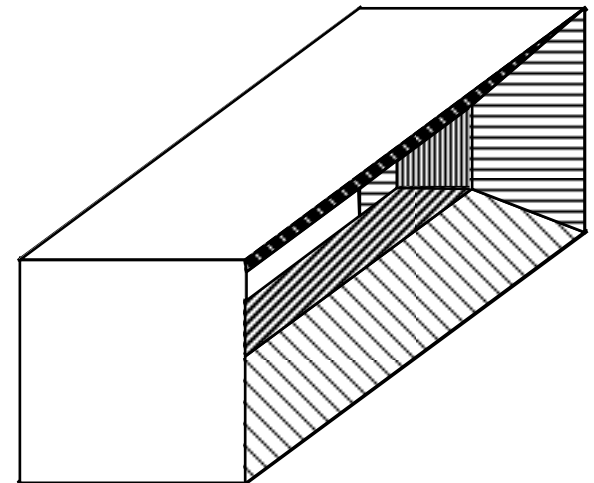
Kantenmodell



oben - unten

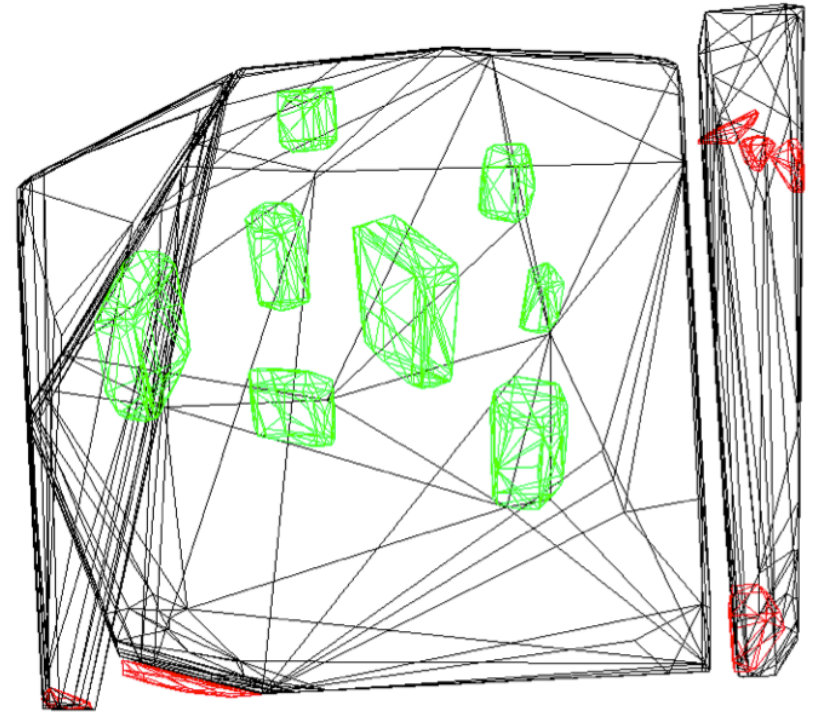


vorne - hinten



links - rechts

Kantenmodell: Beispiel



Flächenmodell

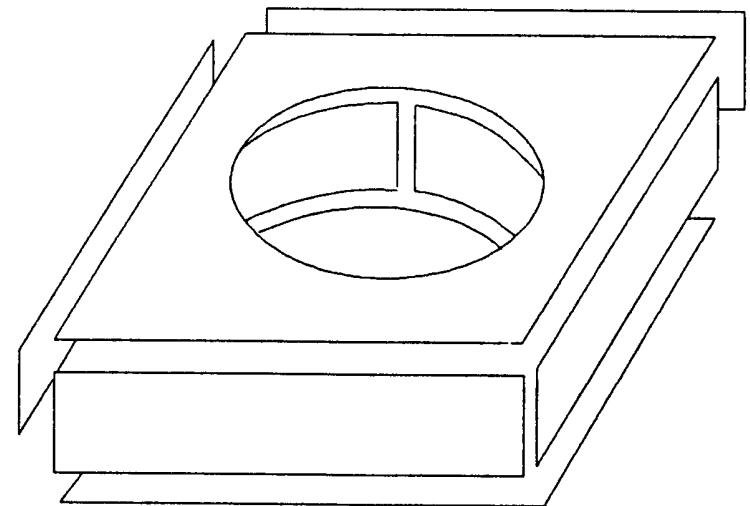
Speicherung von Kanten & Oberflächen (Dreiecke, Bezier)

■ Vorteile:

- + effiziente Verfahren
- + entspricht dem Vorgehen während der Modellierung
- + schnelle Kollisions- und Abstandsberechnung

■ Nachteile:

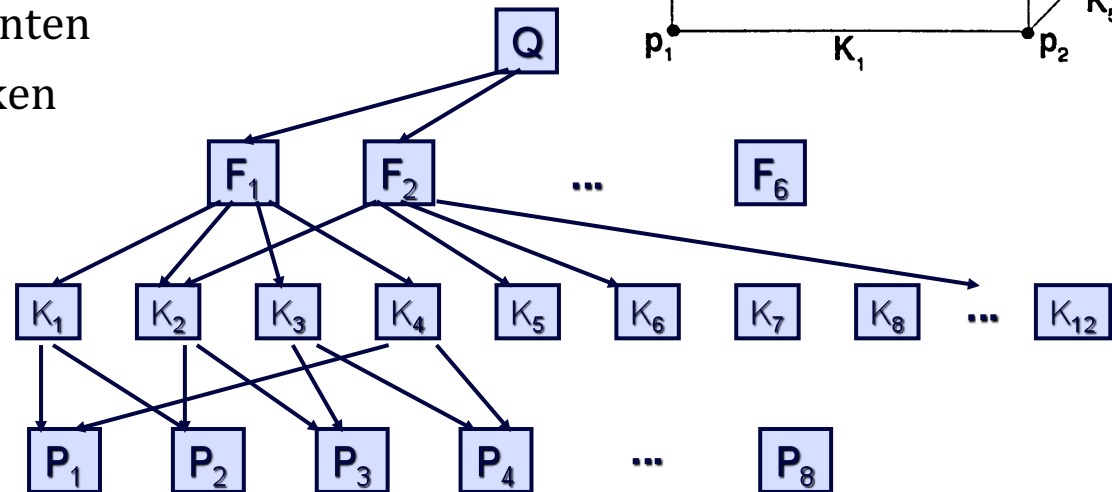
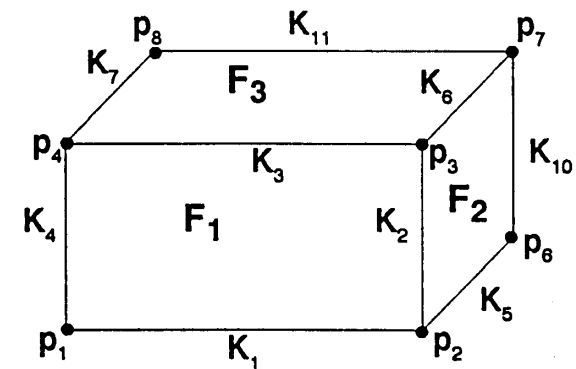
- hoher Eingabeaufwand
- Darstellung aufwendig
- Problem bei Schnittoperationen
- Inkonsistenzen möglich



Boundary Representation

Organisationsform der geometrischen Flächenmodelle

- Hierarchische Darstellung eines Objektes durch begrenzende Elemente, i.d.R. Kanten oder Flächen.
- Elemente eines Quaders im Flächenmodell
- Elemente:
 - Q : Quader
 - $F_i: i \in \{1, \dots, 6\}$: Flächen
 - $K_i: i \in \{1, \dots, 12\}$: Kanten
 - $P_i: i \in \{1, \dots, 8\}$: Ecken



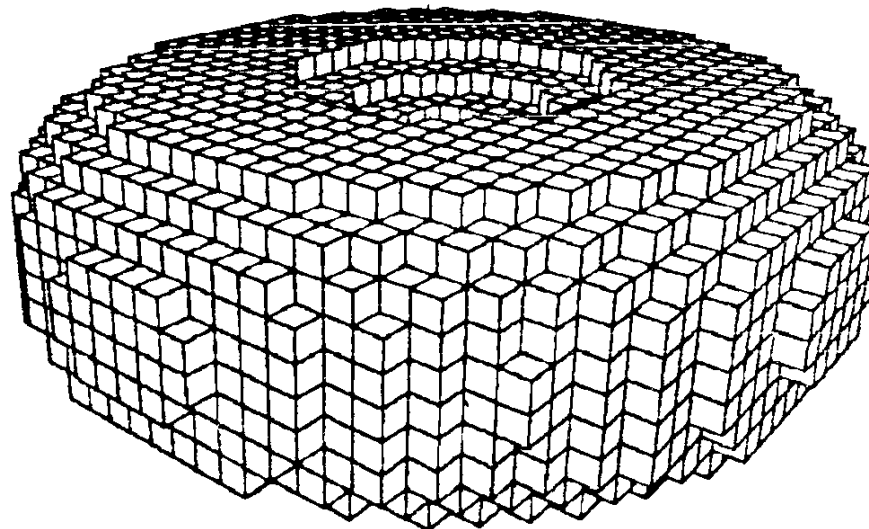
Boundary Representation: Vorteile

Aus topologischer Struktur Informationen über

- Welche Flächen gehören zum Objekt?
- Welche Kanten gehören zur Fläche?
- Zu welchem Objekt gehört eine Fläche?
- Zu welchem Objekt gehört eine Kante?
- Welche Flächen stoßen aneinander?
→ *kantenbasierte Objekterkennung*

Approximative Zellenbelegung

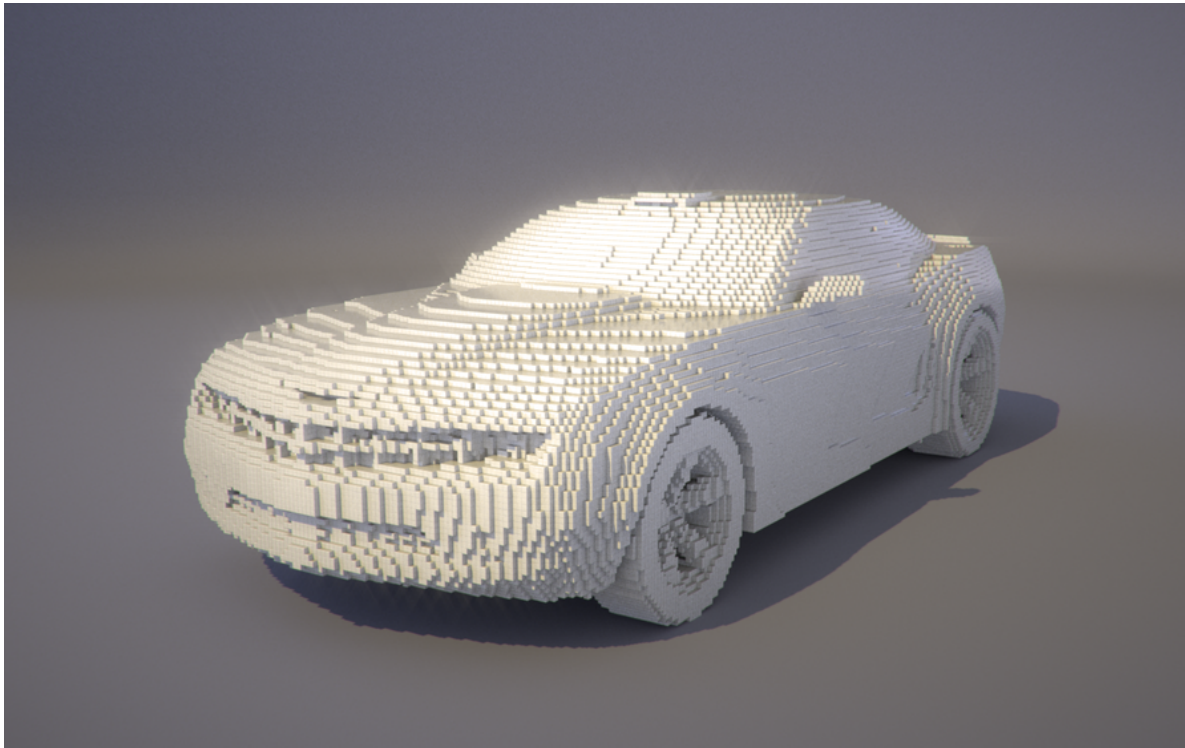
- Objekte werden aus disjunkten Elementarzellen aufgebaut. Verwendung finden einfache geometrische Objekte z.B. Tetraeder, Quader, ...
- Benutzt in der Strukturanalyse mit Finite-Elemente-Methoden (FEM).



Voxeldarstellung

Äquidistante Raumunterteilung in 3D

- Speicherungsmöglichkeiten
 - 0: nicht-belegt / 1: belegt / 2: weiß-nicht (unbekannt)
 - Wahrscheinlichkeit belegt 0.0 bis 1.0



Voxeldarstellung

Äquidistante Raumunterteilung in 3D

■ Vorteile:

- + Einfache Darstellung
- + Berechnungen homogen, parallel auf GPU
- + Raycasting u.ä. einfach parallelisierbar

■ Nachteile:

- Festes Gitter, gleicher Detaillierungsgrad sowohl bei großflächigen als auch bei feiner aufgelösten Strukturen
- Kartengröße/Präzision durch Speicher begrenzt

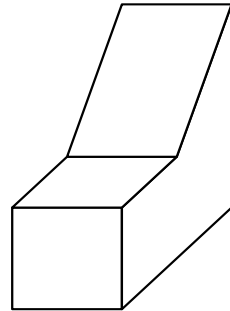
Octree

Octree: effizientere Topologie als Voxel

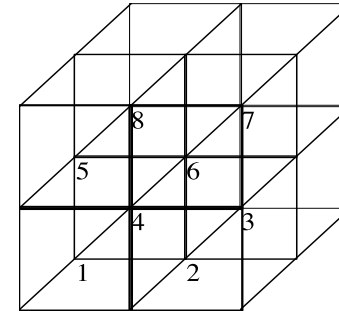
- Der Raum wird in mehrere Zellen unterteilt (i.d.R. 8 Zellen: „oct-tree“).
- Zelle komplett vom Objekt belegt → als „belegt“ markieren
- Wenn die Zelle nur teilweise belegt ist, dann wird auf diese Zelle das Verfahren rekursiv angewendet. Ansonsten ist die Zelle leer.
- Die Rekursion terminiert bei einer vorbestimmten minimalen Zellgröße.
- Teilbelegte kleinste Zellen werden als belegt markiert.

Octree (Beispiel)

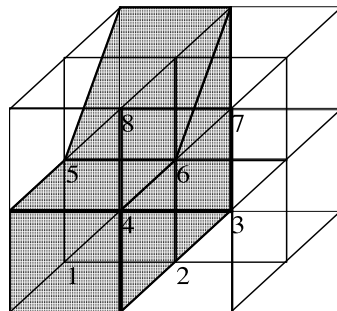
Körper



Zerlegung



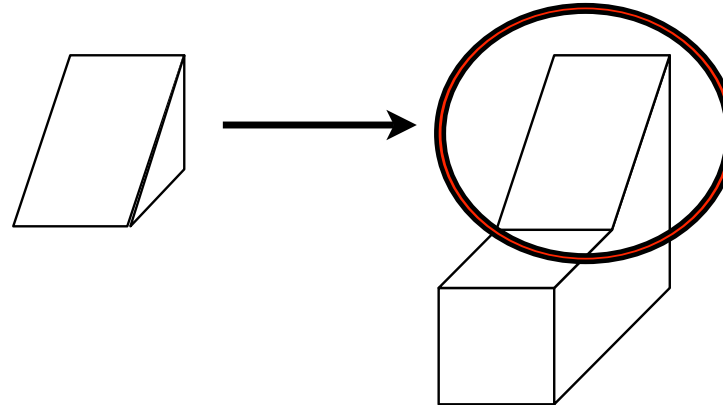
1. Zerlegung



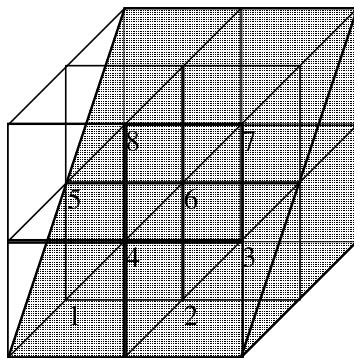
Zelle 1	belegt
Zelle 2	frei
Zelle 3	frei
Zelle 4	belegt
Zelle 5	frei
Zelle 6	frei
Zelle 7	frei
Zelle 8	teilbelegt

Octree (Beispiel)

Restkörper

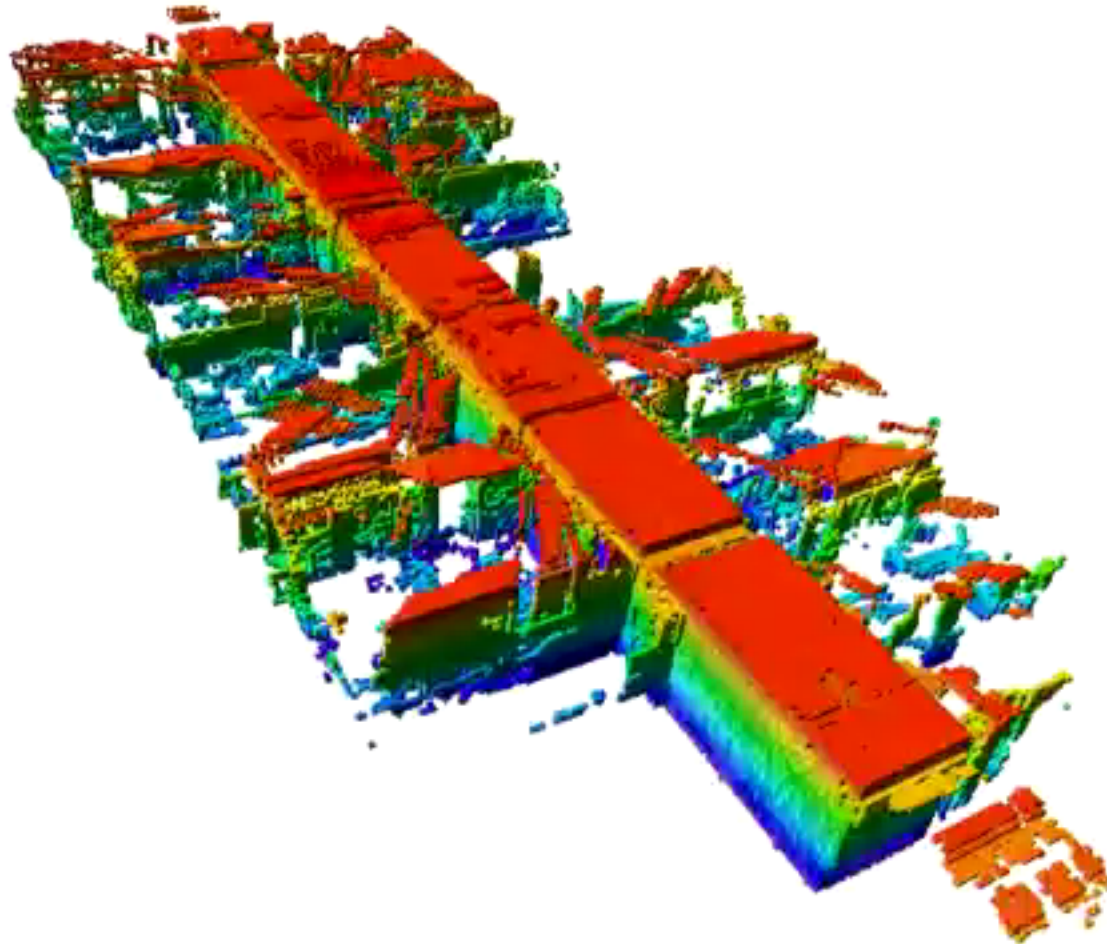


2. Zerlegung



Zelle 1	teilbelegt
Zelle 2	teilbelegt
Zelle 3	belegt
Zelle 4	belegt
Zelle 5	frei
Zelle 6	frei
Zelle 7	teilbelegt
Zelle 8	teilbelegt

Octree: Octomap



<http://octomap.github.io>

Ottree: Octomap

Octomap (BSD Lizenz) für Manipulation:

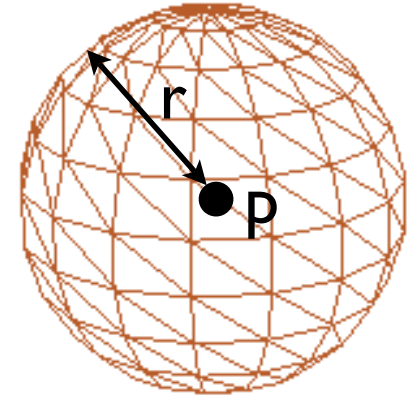


<http://octomap.github.io>

Analytisch-parametrische Modelle

Beispiel analytisch gegebene Fläche

- Beispiel: 3D-Kugel
- $r = ||x - p||$
- Exaktes Modellierungsverfahren
- **Vorteile:**
 - + Geschlossene Darstellung (wenig Speicherbedarf)
 - + Analytische Darstellung erlaubt einfache Rechenverfahren (z.B. Schnitt von Ebenen / Kugeln → schnelle Kollisionsberechnung)
- **Nachteil:**
 - Wenige Flächen sind analytisch darstellbar

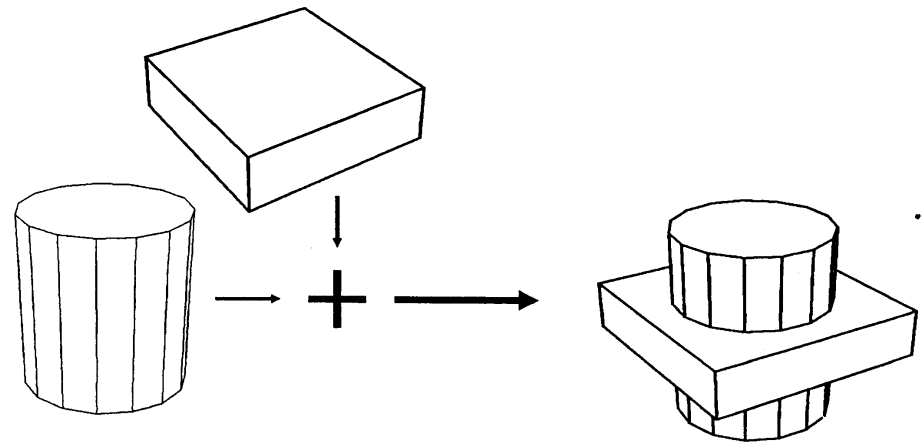


Parametrische Volumenmodelle

- Grundkörper und topologische Operationen auf diesen (Schnitt, Vereinigung, ...) werden abgespeichert.

- **Vorteile:**

- + eindeutige Objektbeschreibung
- + geringer Eingabeaufwand
- + Ergebnis von Operationen sind korrekte Objekte

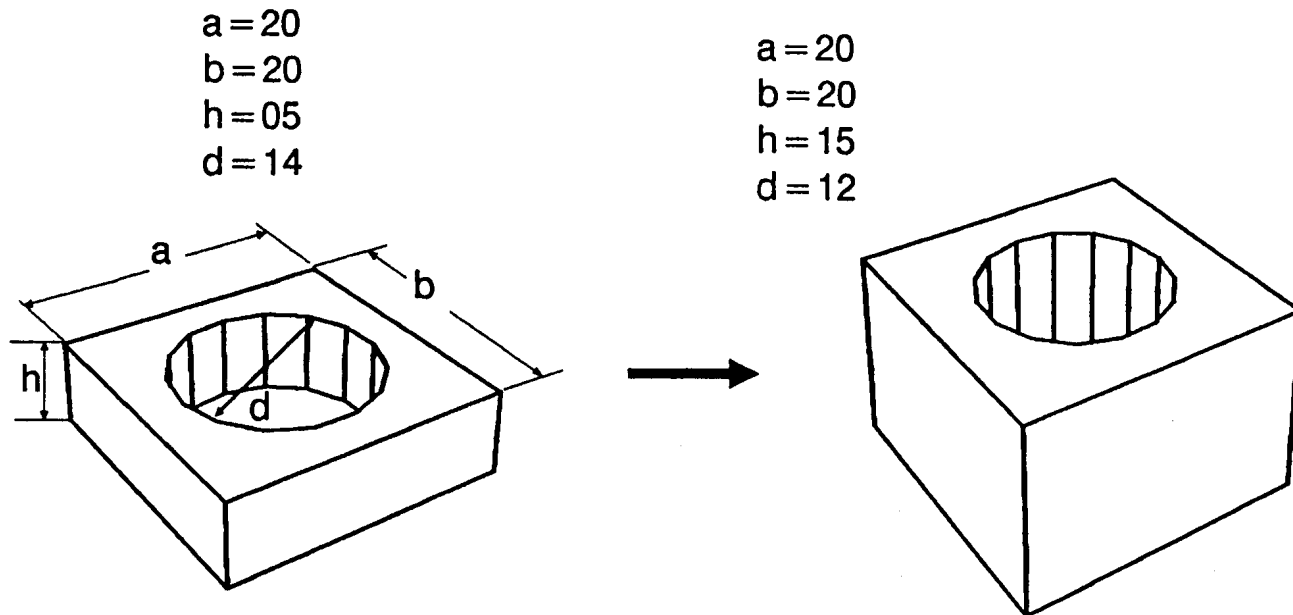


- **Nachteile:**

- hoher Implementierungsaufwand
- Einbindung von Freiformflächen schwierig

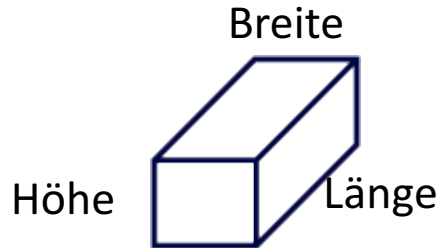
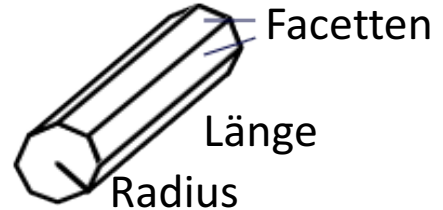
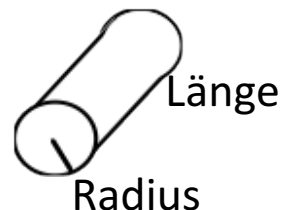
Parametrische Volumenmodelle

- Objekte sind bereits vorhanden und können durch Angabe von Parametern angepasst werden (Varianten).
- Konsistenzprüfungen sind notwendig ! ($d < a$)

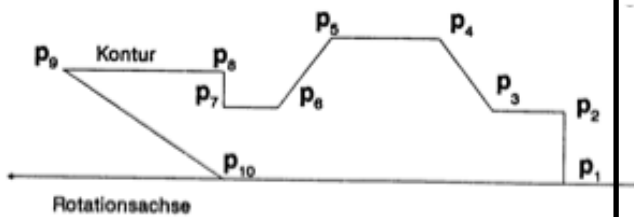


Constructive Solid Geometry (CSG)

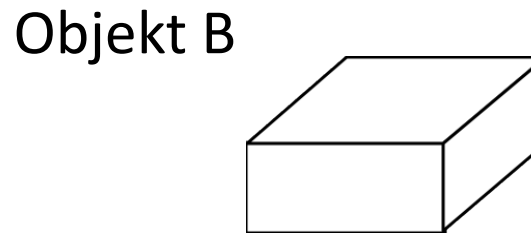
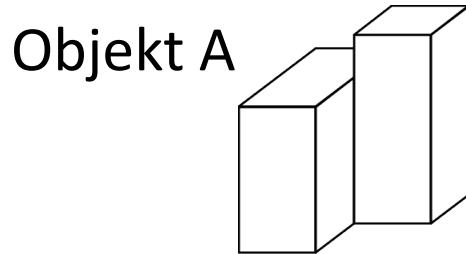
- Es gibt eine Menge von einfachen Grundkörpern, die parametrisiert werden können und auf denen verschiedene Operationen definiert sind.

Grundkörper	Parameter	Skizze
Quader	Länge, Breite, Höhe	 Breite Höhe Länge
Prisma	Länge, Radius, Facetten	 Facetten Länge Radius
Zylinder	Länge, Radius	 Länge Radius

Constructive Solid Geometry (CSG)

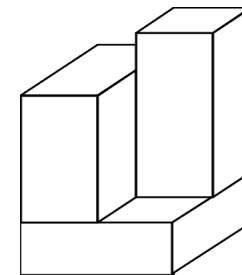
Grundkörper	Parameter	Skizze
Kegel	Länge, Radius a, Radius b	 Radius a Länge Radius b
Ellipsoid	Radius a, Radius b, Radius c	 Radius b Radius c Radius a
Rotationskörper	Achse, Kontur	

CSG: Operatoren

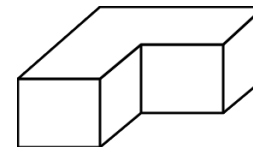


Operatoren

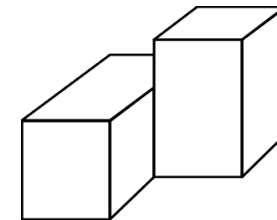
■ Vereinigung (Summe) $A \cup B$



■ Schnitt $A \cap B$



■ Differenz A / B

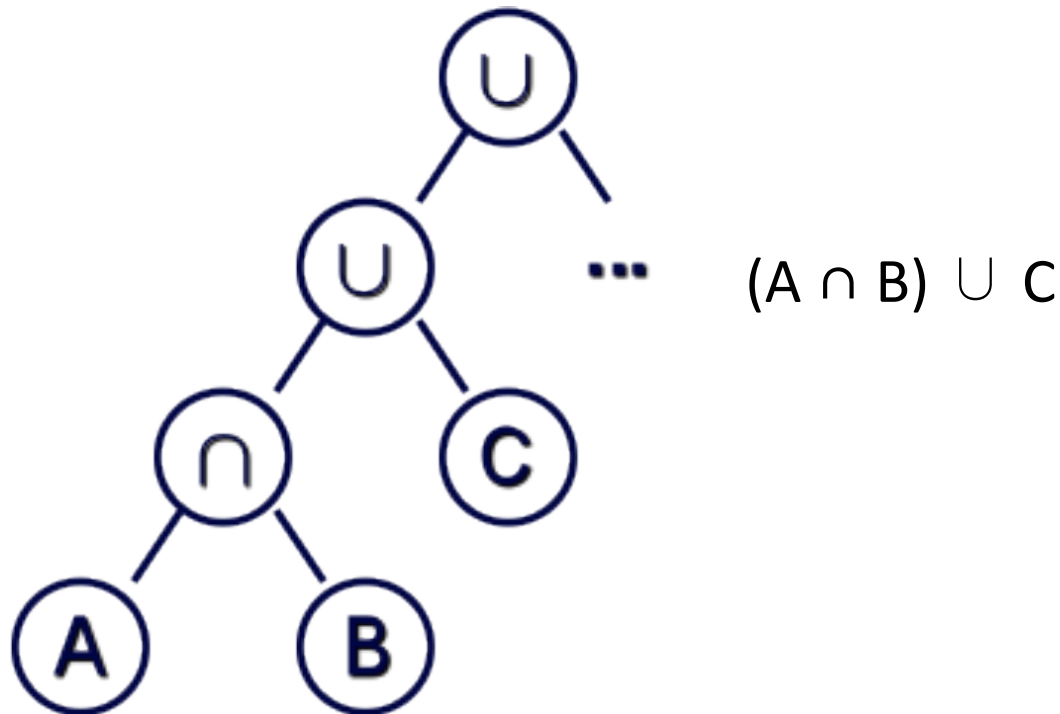


■ Sweep:

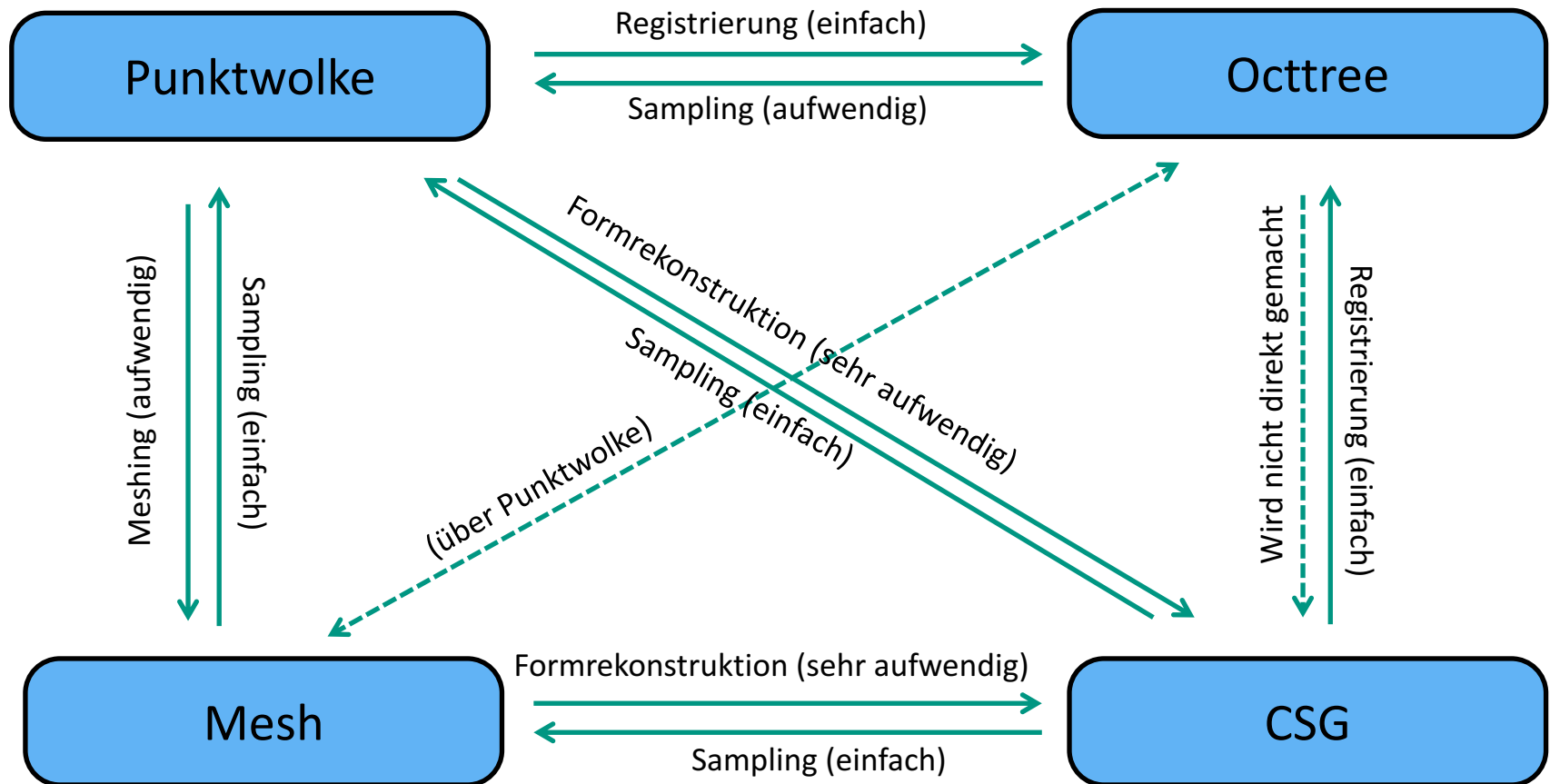
Ein Grundelement (z.B. eine Fläche) wird entlang einer Raumkurve verschoben. Der durchdrungene Raum stellt das neue Objekt dar.

CSG: Operatoren

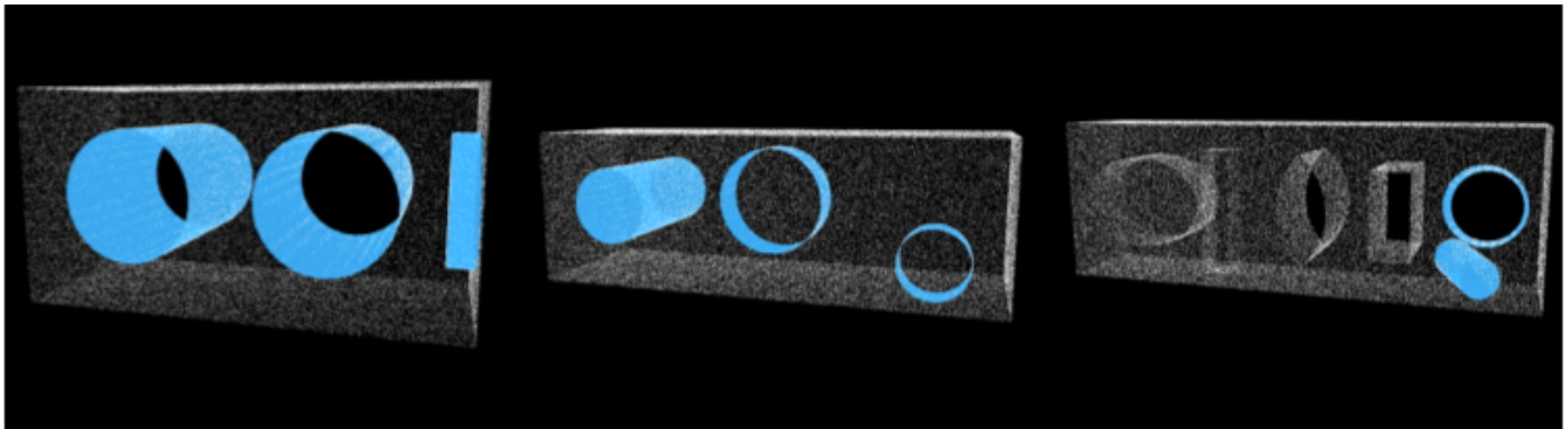
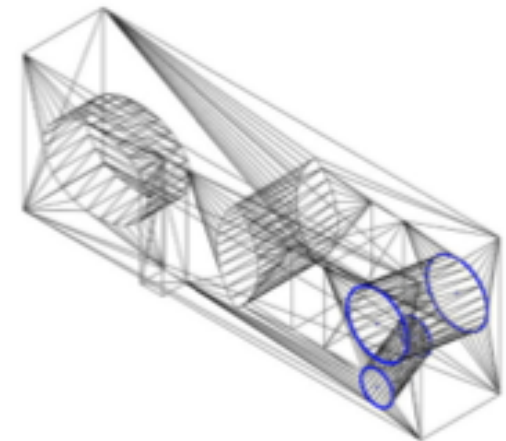
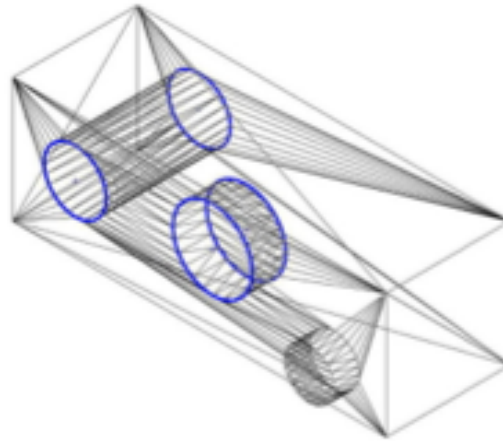
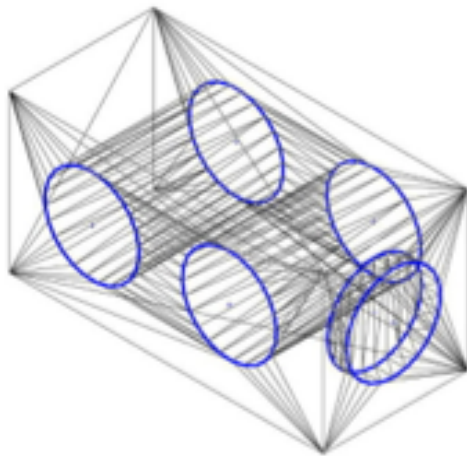
- Speicherung erfolgt als Binär-Baum mit
 - Knoten als Operation
 - linker bzw. rechter Teilbaum als Teil-CSG oder Grundkörper



Umwandlungen zwischen Modelltypen



Beispiele Umwandlungen



Anwendungsgebiete Modelltypen

Anwendungsgebiete für das geometrische Modell

- Punktwolke:
 - Lokalisation, Klassifikation
 - Kartierung (mobile Systeme)
- Mesh:
 - Bewegungsplanung (v.a. Manipulatoren)
 - Dynamische Simulation (komplexe Starrkörper)
- Voxel/Octtree:
 - Bewegungsplanung (v.a. mobile Systeme)
 - Dynamische Simulation elastische Materialien (FEM)
- CSG:
 - CAD/CAE/CAM
 - Dynamische Simulation (einfache Starrkörper)

Inhalt

- Motivation: Adaptive Roboter Aufgaben
- **Objektmodelle**
 - Geometrische Beschreibung
 - **Zusätzliche Eigenschaften**
- Szenenmodelle

Zusätzliche Eigenschaften

Eigenschaften des Objektes mit Geometriebezug

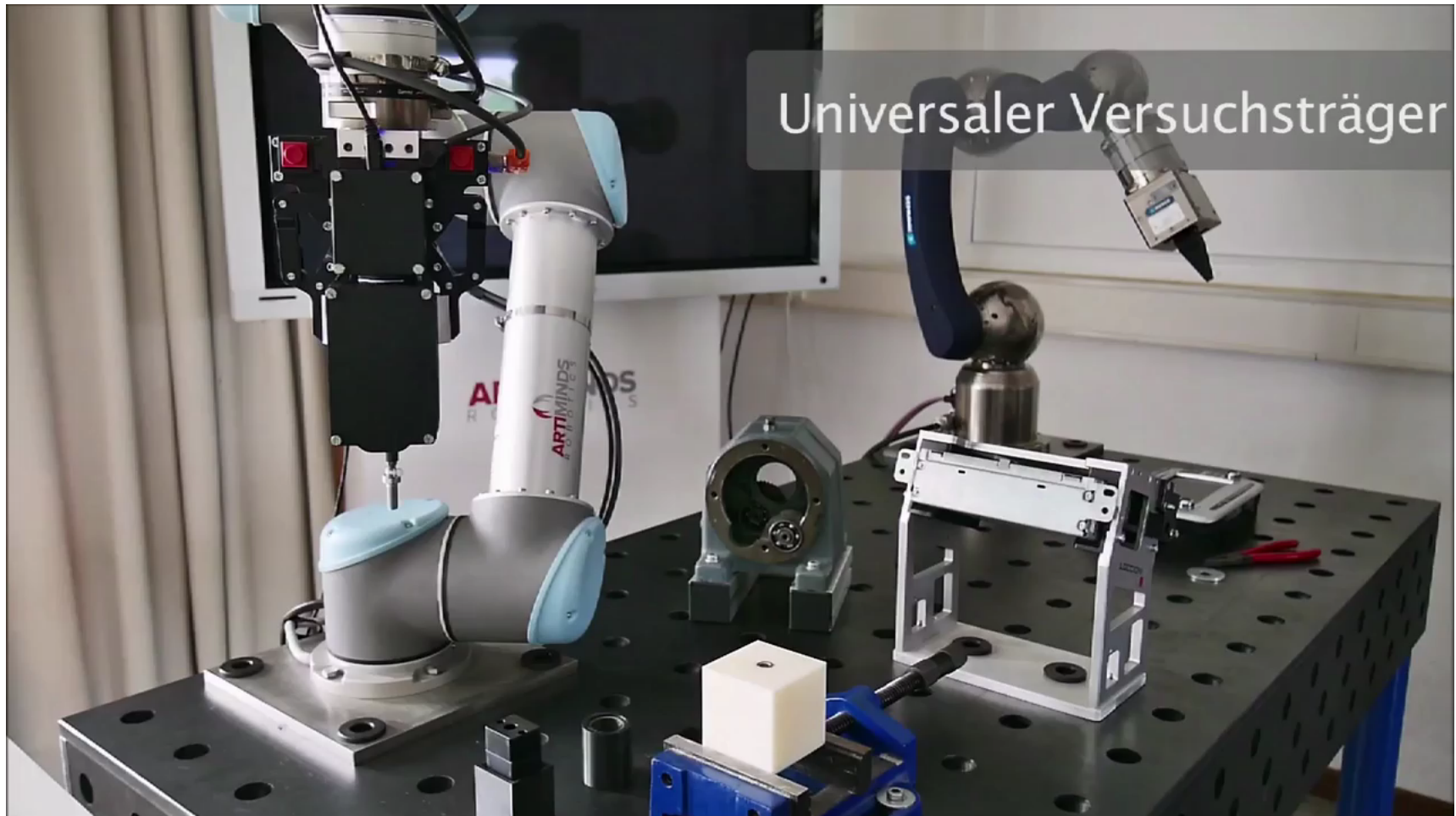
- Masse
- Oberflächeneigenschaften (z.B. Reibung)
- Temperatur
- Steifigkeit

- Greifpunkte
- Verbindungspunkte zur Montage
- Ablagepunkte bei Paletten
- Füllmenge bei Paletten
- Stellung bezüglich eines Referenzobjektes

Stark aufgabenabhängig!

Zusätzliche Eigenschaften

Beispiele zusätzliche Eigenschaften



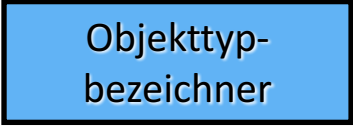
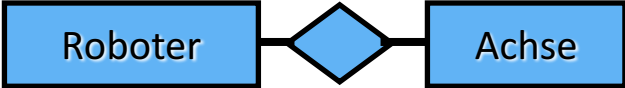
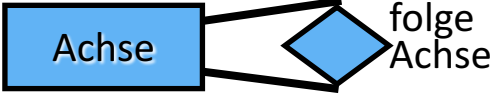
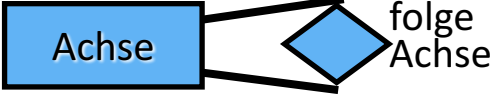
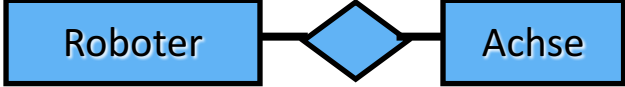
Inhalt

- Motivation: Adaptive Roboter Aufgaben
- Objektmodelle
 - Geometrische Beschreibung
 - Zusätzliche Eigenschaften
- Szenenmodelle

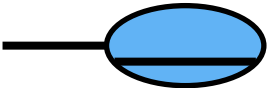
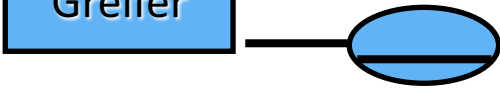
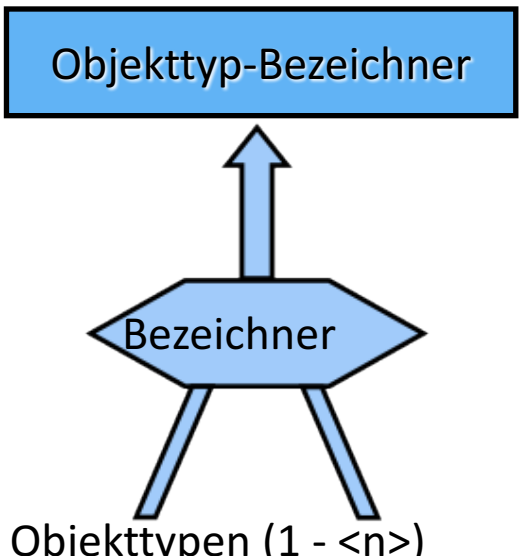
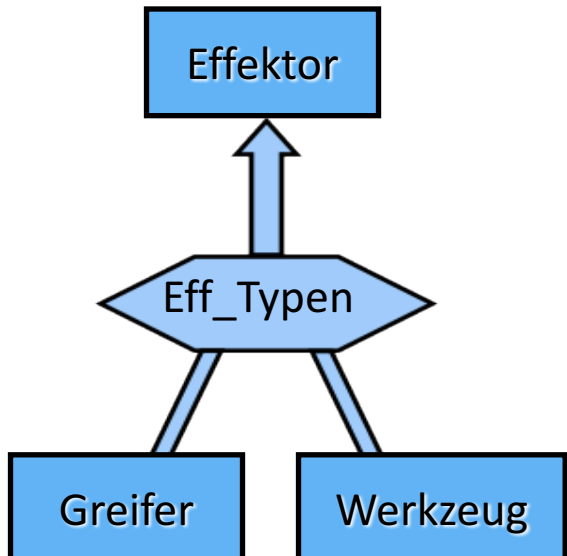
Übersicht Szenenmodelle

- Entity-Relationship-Modelle
- Semantische Netze
- Frame Modelle nach Minsky
- Implicit Shape Models (ISMs)
- Probabilistische Object Constellation Models (OCMs)

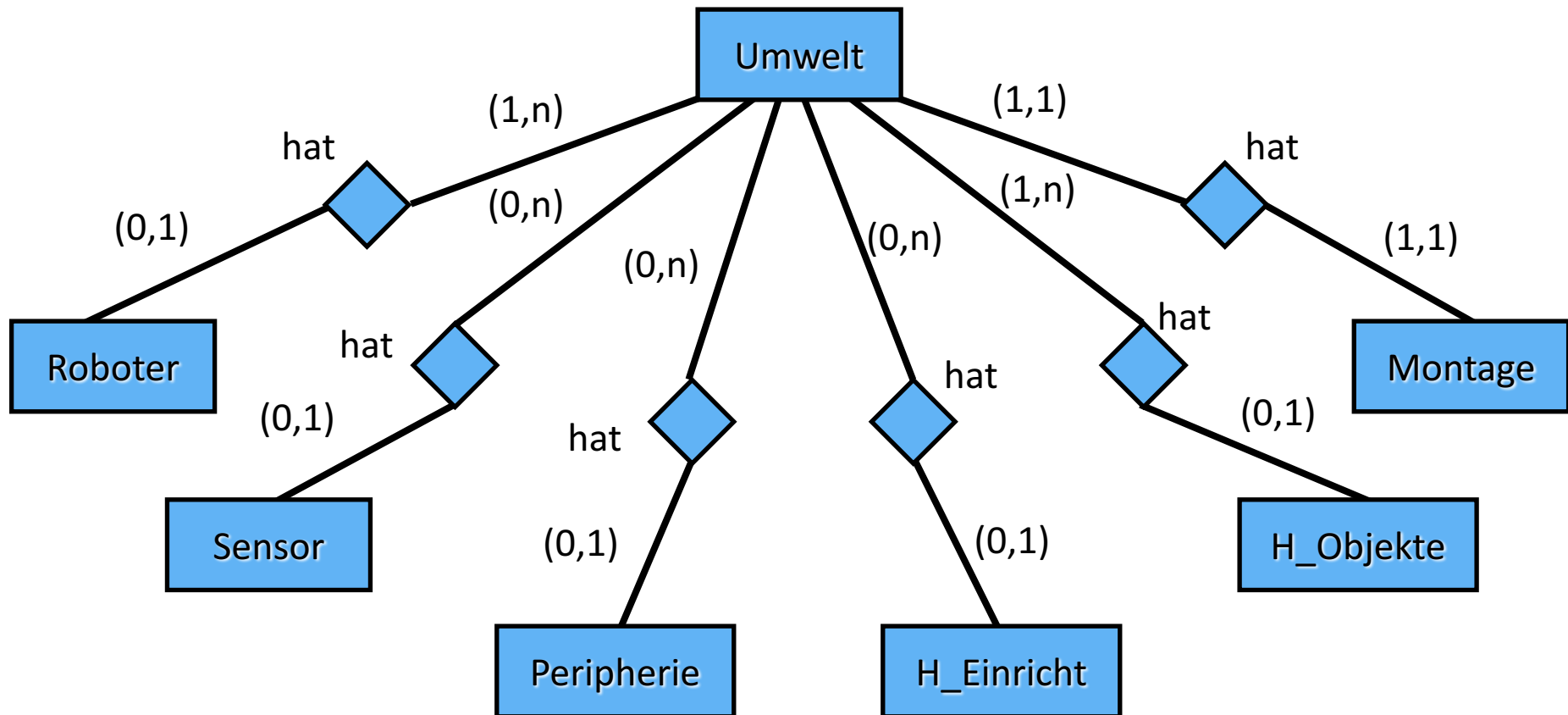
Entity-Relationship-Modelle (ER)

Modellierkonstrukt	Graphisches Symbol	Beispiel
Objekttyp (Entity set)		
Beziehungstyp (Relationship type) ○ zweiseitig ○ rekursiv	Beziehungstyp- bezeichner 	hat_achse  Achse 
Rollen	Rolle A Rolle B 	Vorgänger  Nachfolger
Kardinalität	(min, max) (min, max)  (0, 1) (0, 1) (0, n) (0, 1) (0, n) (0, m)	hat_achse  (1,n) (1,1)

Entity-Relationship-Modelle (ER)

Modellierkonstrukt	Graphisches Symbol	Beispiel
Attribute ○ beschreibende ○ identifizierende	Attributbezeichner  Attributbezeichner 	Anz_Finger  Gr_Bezeichnung 
Generalisierungshierarchie	 Objektypen (1 - <n>)	

Beispiel eines ER-Modells



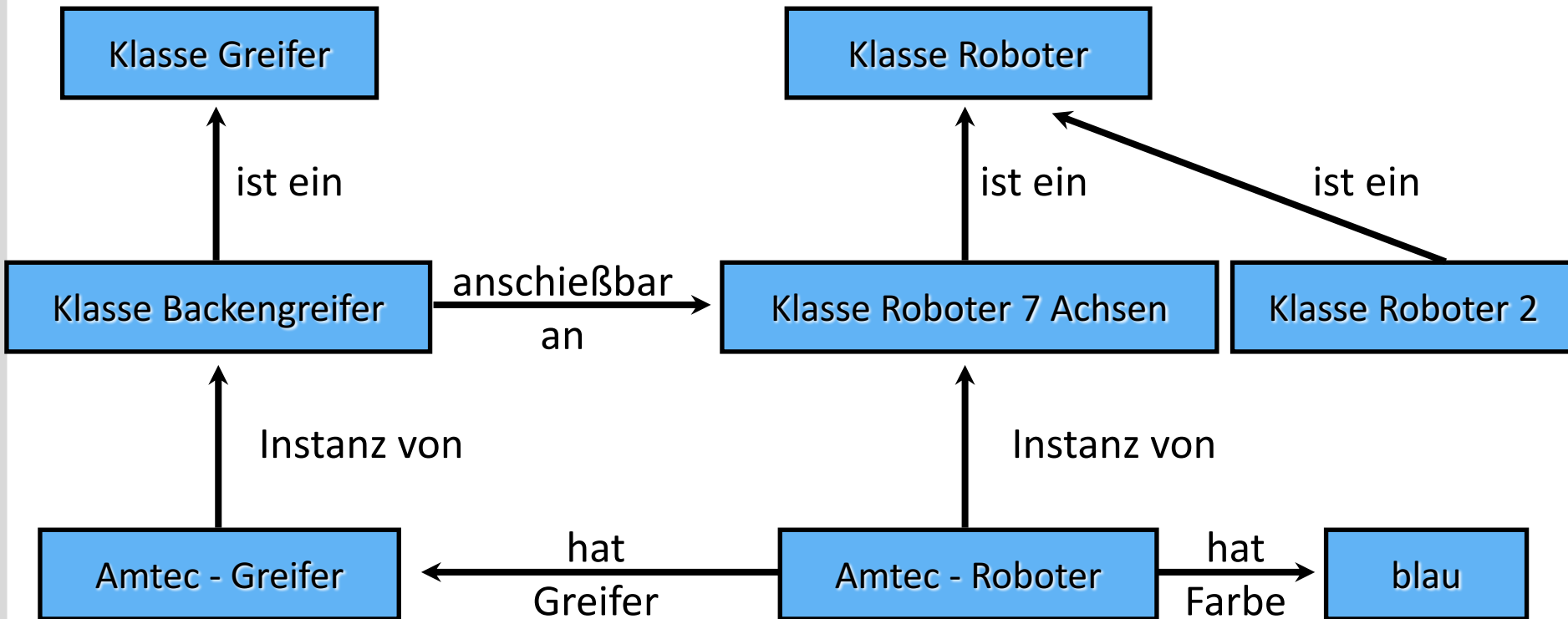
Semantische Netze

Konzept: Objekte und Beziehungen

- Semantisches Netz = gerichteter Graph
- Knoten = Objektklasse oder Einzelobjekt
- gerichtete und benannte Kanten = Beziehungen

- nur zweistellige Beziehungen
- mehrstellige Beziehungen => als eigenes Objekt
- keine Attribute => Attribute als „Wertobjekt“

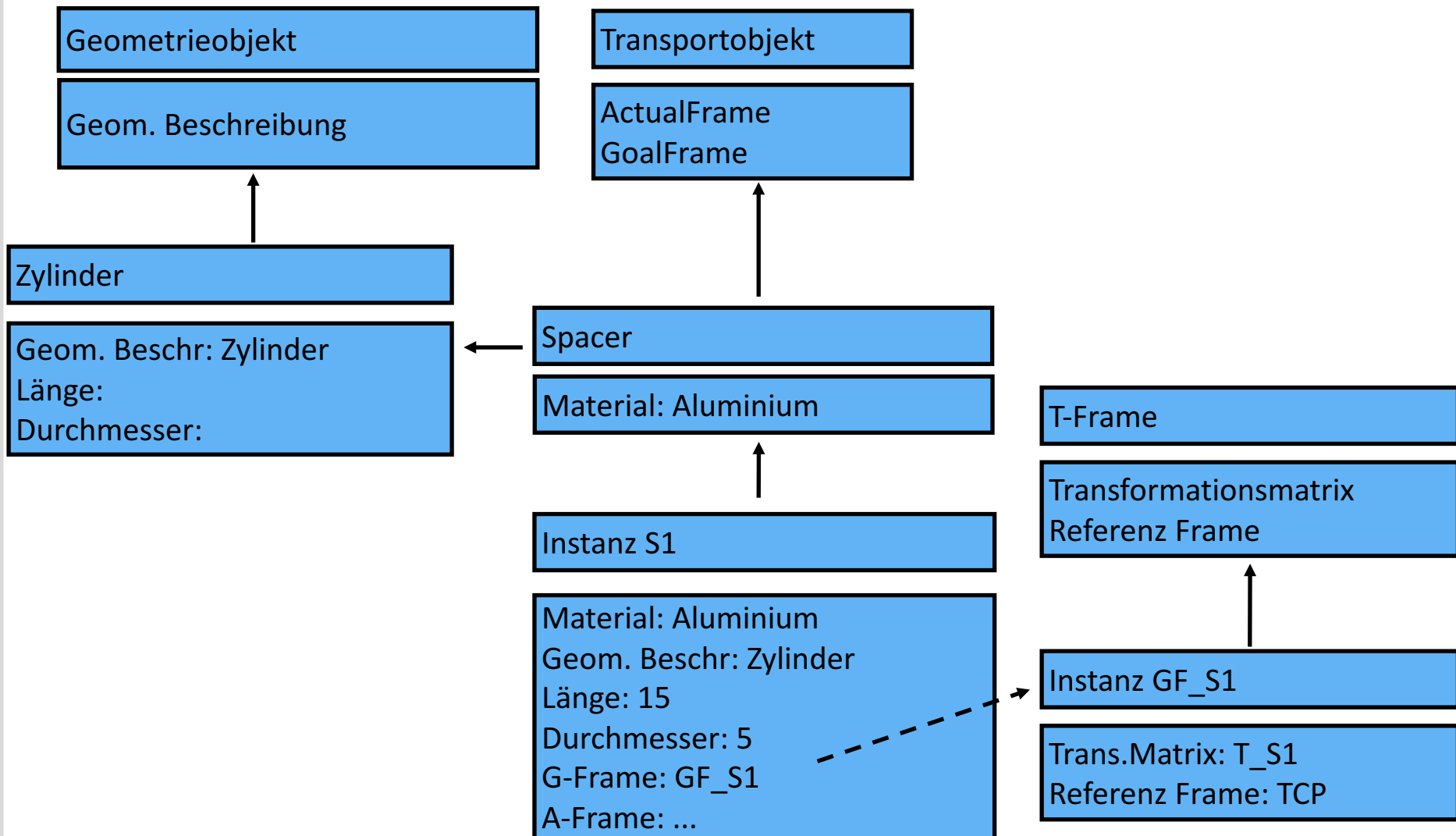
Beispiel eines Semantischen Netzes



Frame Modell nach Minsky

- Frame = Schablone (nicht mit Koordinaten-Frames verwechseln)
 - Erfassung der:
 - Eigenschaften von Objekten
 - Einordnung in Hierarchie von Objektklassen
 - Leichte Implementierung mit objektorientierten Sprachen
 - Frame:
 - beschreibt ein Objekt / Objektklasse
 - enthält eine Menge von Slots
 - Slots enthalten:
 - Attribute
 - Verweise auf andere Frames
- Instanzen, Vererbung

Beispiel eines Frame Modells



Implicit Shape Modell (ISM)

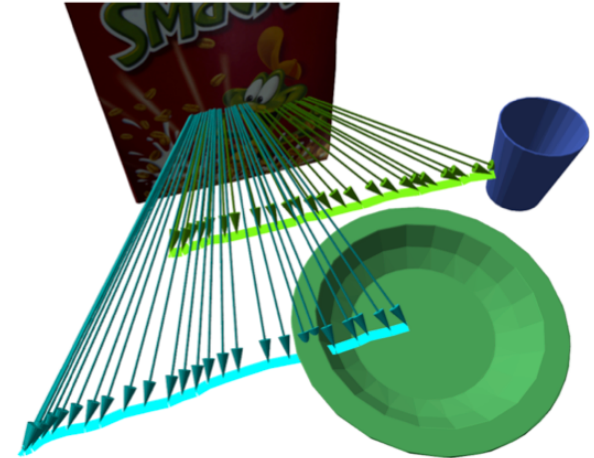
Moderne Szenenbeschreibung, robust bei Varianzen

- Szene besteht aus relativen Transformationsrelationen zwischen Objekten
- ISM ist Variation der Generalisierten Hough Transformation (Maschinensehen)
- Daraus folgt, dass Instanzen der Relationen in *buckets* abstimmen (*voten*)
- Baumartige Topologie der relativen Relationen

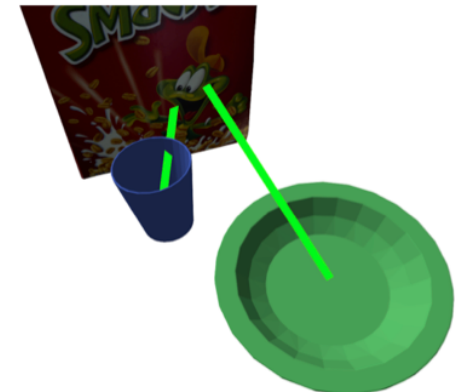
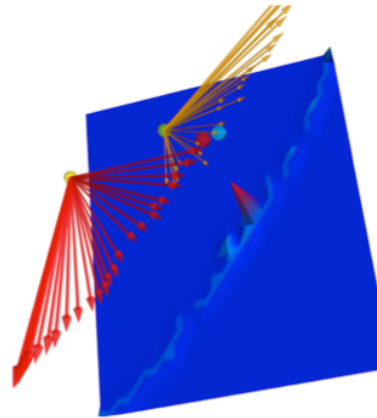
Implicit Shape Modell (ISM)

Hough voting findet im 3D-Raum statt

- Votes werden in einem Voxel-Gitter akkumuliert



- Weitere Informationen:
Meißner et al.



Active Scene Recognition for Programming by Demonstration using Next-Best-View Estimates from Hierarchical Implicit Shape Models

Pascal Meißner, Reno Reckling, Valerij Wittenbeck, Sven R. Schmidt-Rohr and Rüdiger Dillmann

Objekt Constellation Modell (OCM)

Kurzvorstellung OCM: probabilistisches Modell

- Probabilistische Szenenrelationen
- Basiert auf Gauss-Mixturen (GMMs) und probabilistischer Graph-Inferenz

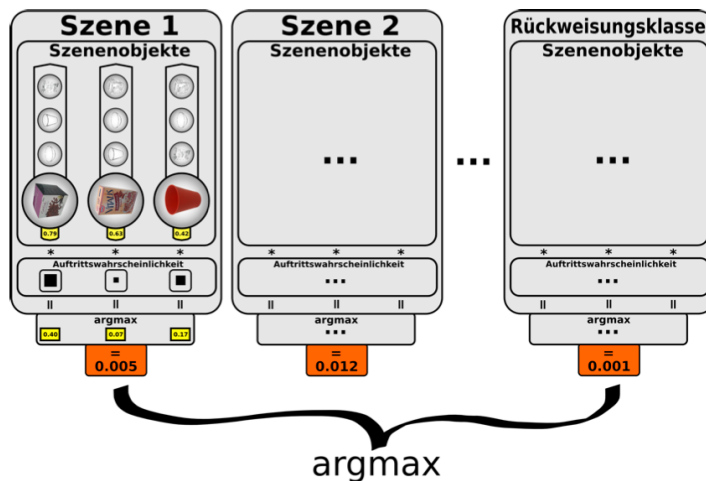
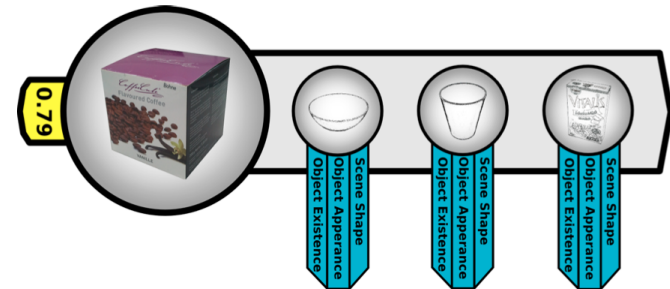


Abbildung 5.4: Eine grafische Darstellung des Szenenmodells. Es wird die Wahrscheinlichkeit einer Reihe von Szenen inklusive der Rückweisungsklasse berechnet. Jede Szene basiert auf einer Reihe von Szenenobjekten, wobei das mit der höchsten gewichteten Wahrscheinlichkeit die Wahrscheinlichkeit der Szene ergibt.



5.6: Das Object Constellation Modell beschreibt ein einzelnes Szenenobjekt, also ein Objekt (hier die Kaffeebox) im Kontext einer Szene. Die Slots werden durch die kleinen Kreise rechts symbolisiert, jeder Slot beschreibt ein Objekt der Szene mit den in blau dargestellten Parametern. Der gelbe Kasten gibt an, mit welcher Wahrscheinlichkeit das Szenenobjekt vorliegt.

Weitere Informationen:
Gehring, Meißner et al.