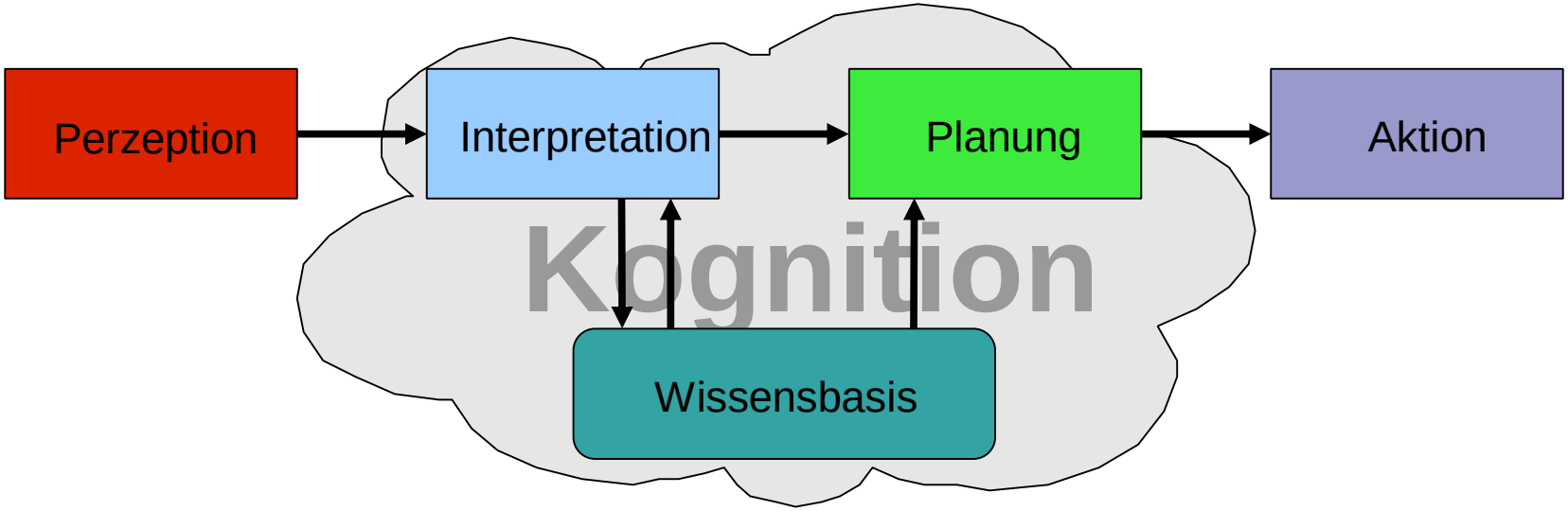
Kognitive Systeme

Robotik

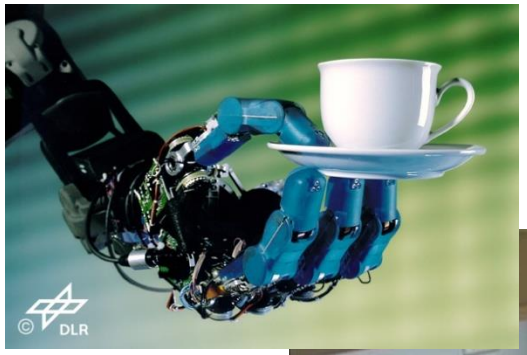
Montag, 17.Juli 2017

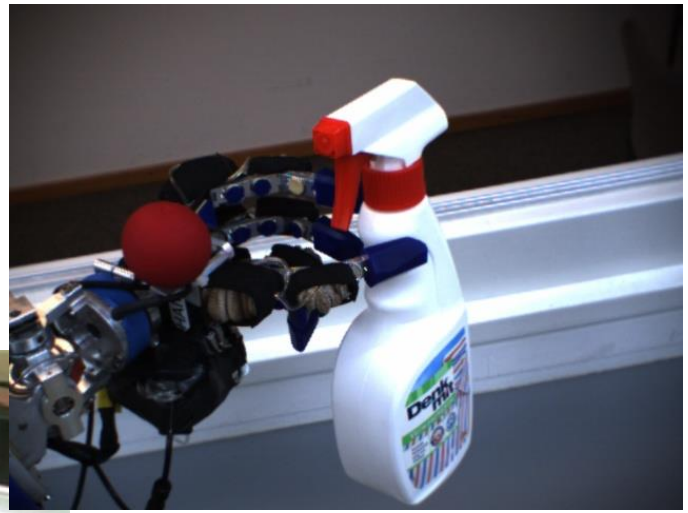
Institut für Anthropomatik und Robotik
Prof. Dr.-Ing. R. Dillmann
Kognitive Systeme

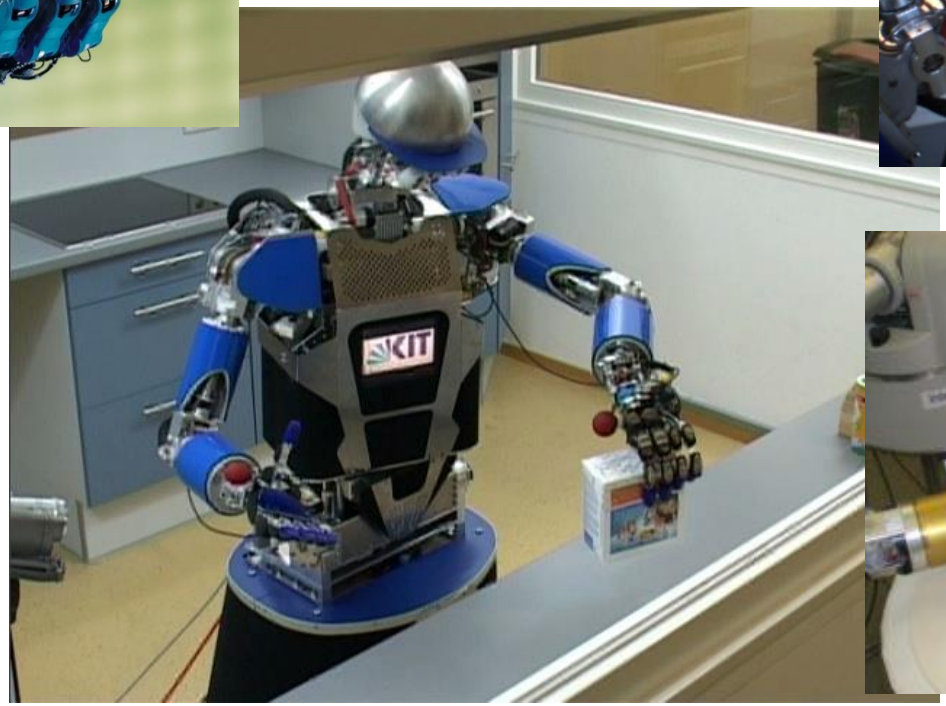
Überblick


Perzeption
Interpretation
Planung
Aktion
Kognition
Wissensbasis

Greifen von Alltagsobjekten

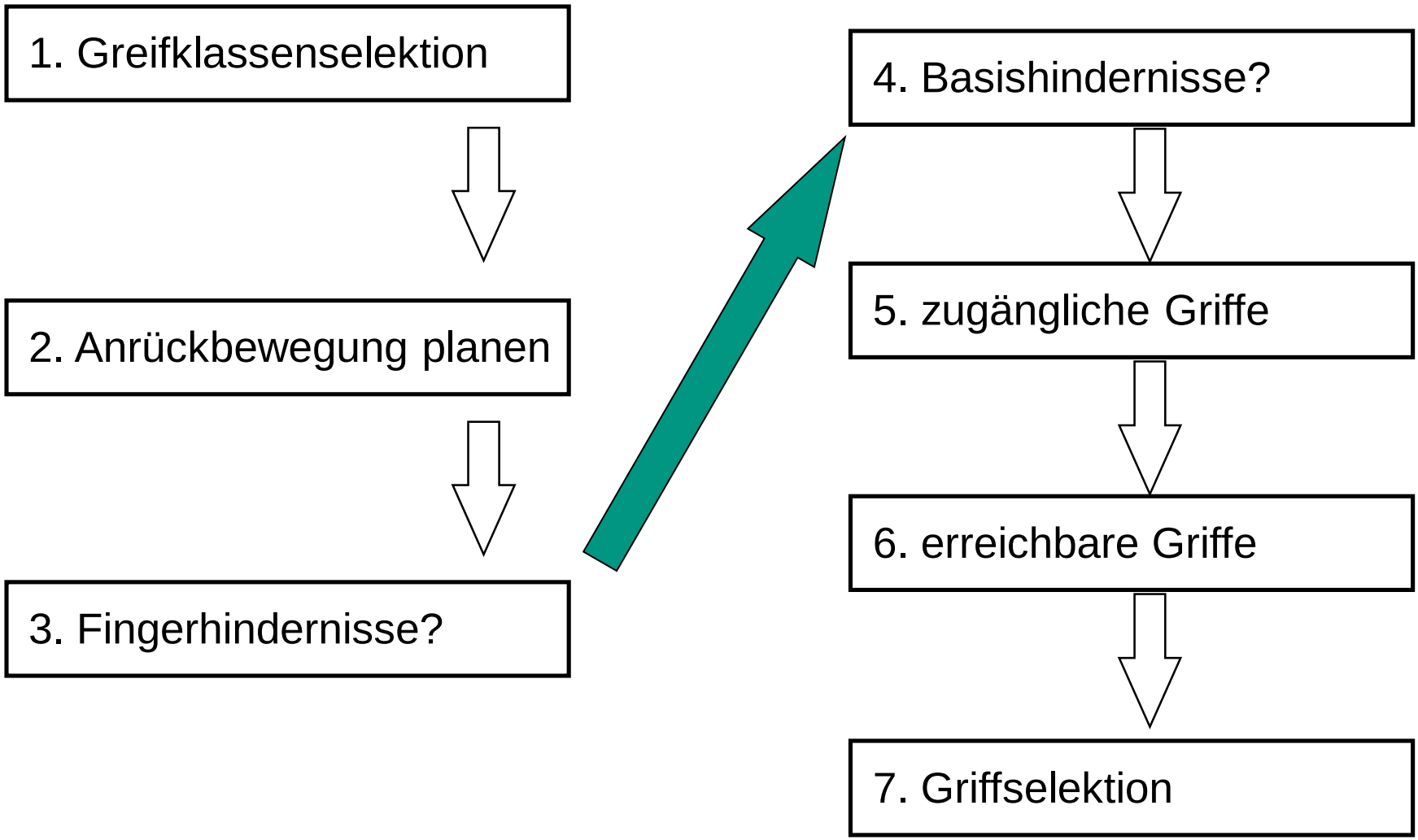






Institut für Anthropomatik und Robotik
Prof. Dr.-Ing. R. Dillmann
Kognitive Systeme

Planung von Greifoperationen



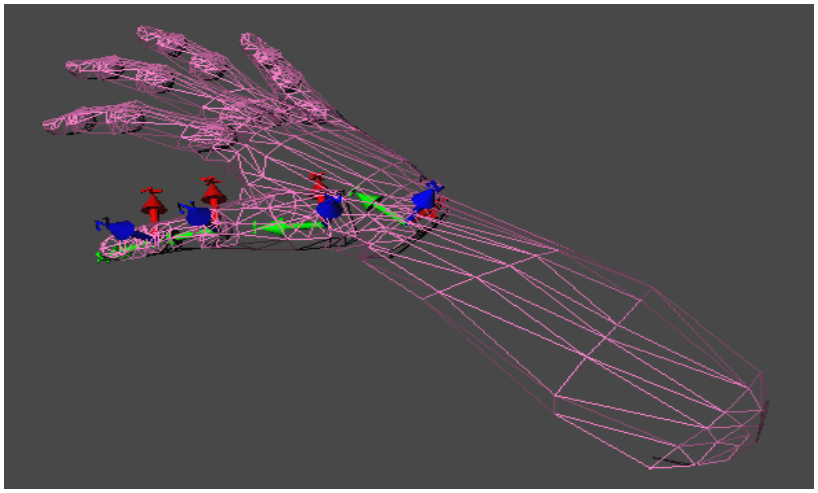
Institut für Anthropomatik und Robotik
Prof. Dr.-Ing. R. Dillmann
Kognitive Systeme

Griffklassen

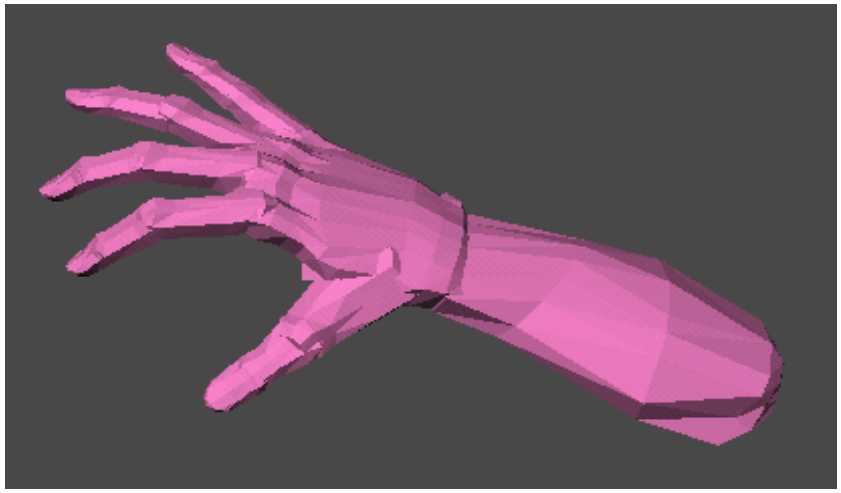
Die menschliche Hand

- 16 Gelenke
- 22 Freiheitsgrade

Zur Modellierung:

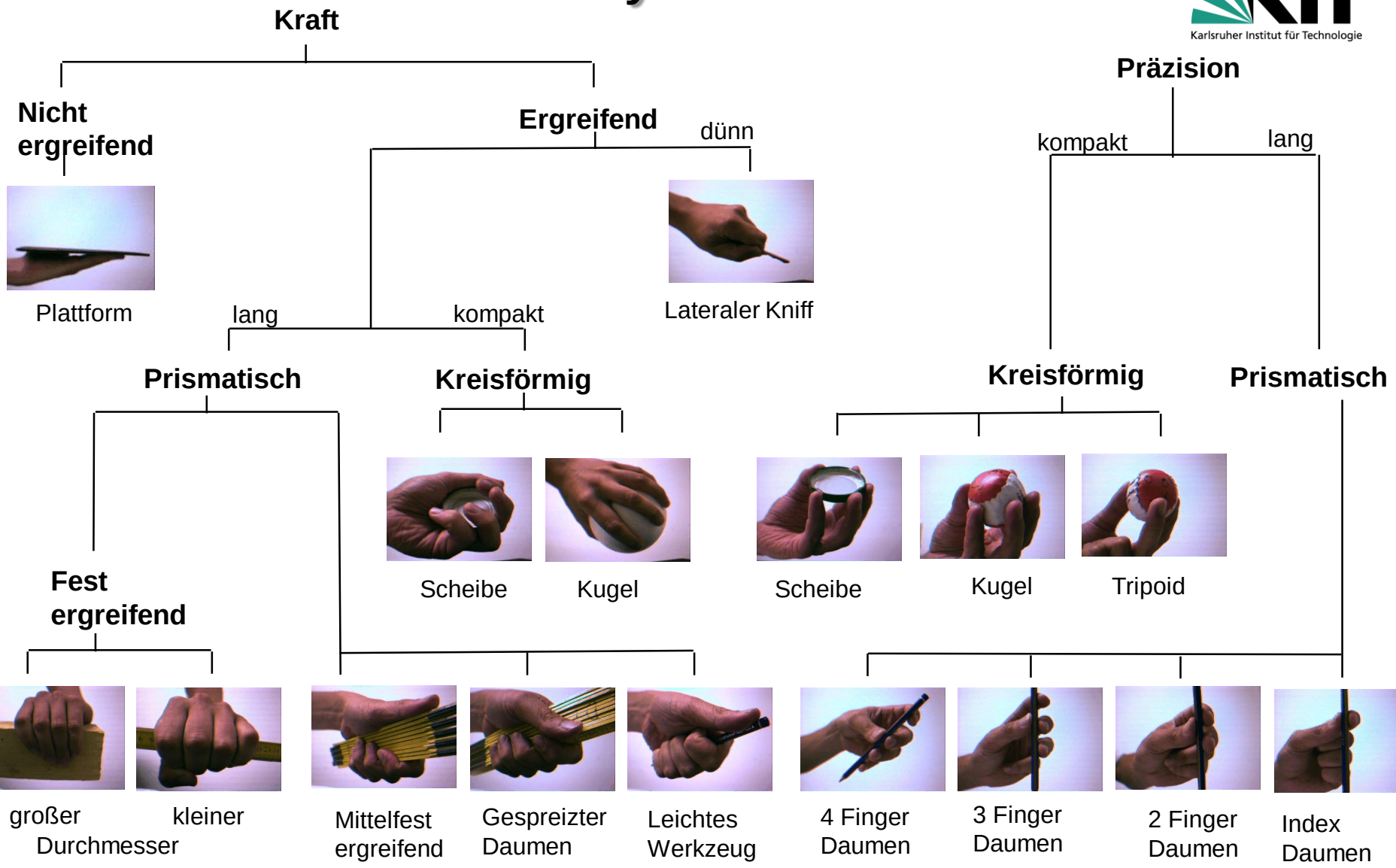


kinematisches Modell



Oberflächenmodell

Cutkosky-Griffhierarchie



Institut für Anthropomatik und Robotik
Prof. Dr.-Ing. R. Dillmann
Kognitive Systeme

Bewegungstypen

Im Hinblick auf Greifen sind folgende Bewegungstypen unterscheidbar:

1. **Greifen / Loslassen eines Objektes** mit montiertem Greifer
   - Auswahl eines sicheren Griffes
   - Kollisionen zw. Greifer, Objekt, Umwelt vermeiden

2. **An- / Abrückbewegung des Greifers**
   - Bewegungsplanung
   - Kollisionen zw. Greifer, Objekt, Roboterarm, Umwelt vermeiden

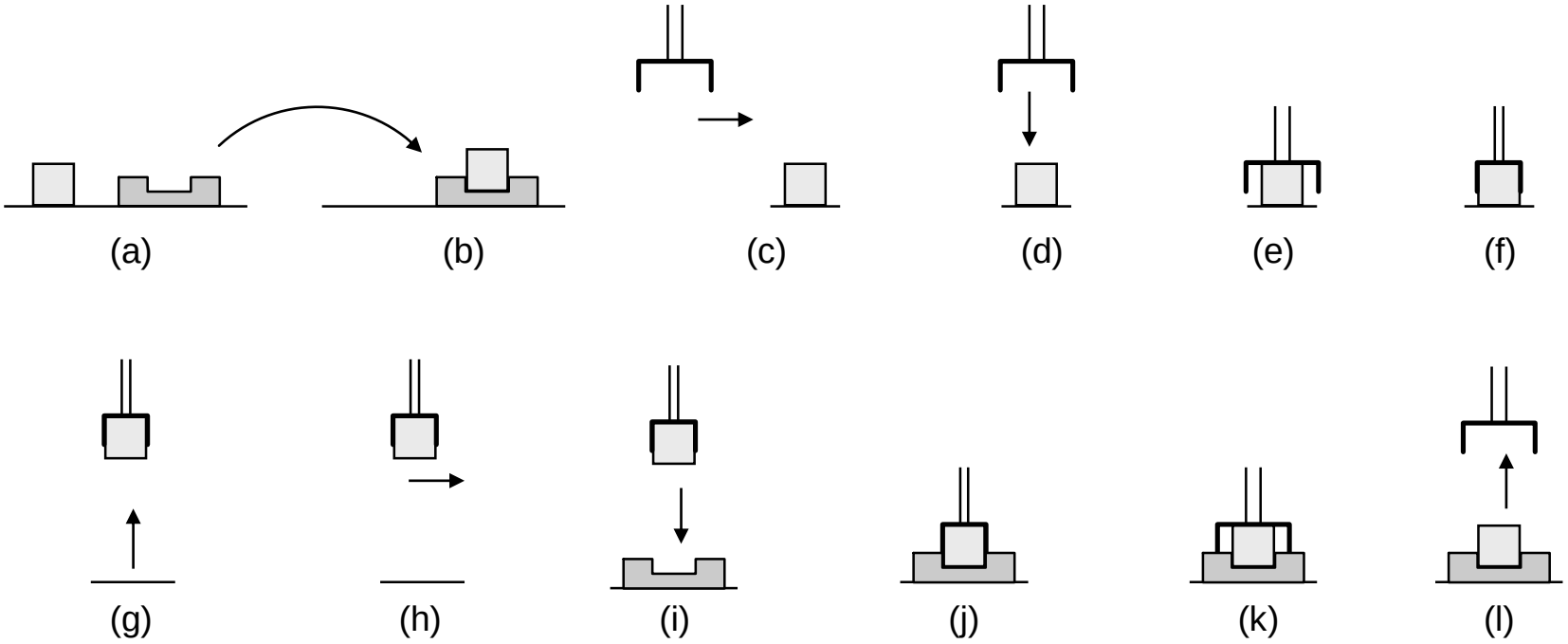
Bewegungstypen

3. **An- / Abrückbewegung mit gegriffenem Objekt**
   - Bewegungsplanung des Greifers
   - Kollisionen zw. Greifer, Objekt, Roboterarm, Umwelt vermeiden

4. **Verbinden des gegriffenen Objektes mit anderen Objekten**
   - sensorüberwachte und/oder sensorgeführte Bewegungen

5. **Transferbewegung des Greifers ohne/mit Objekt**
   - höhere Ausführgeschwindigkeiten
   - geringere Genauigkeitsanforderungen

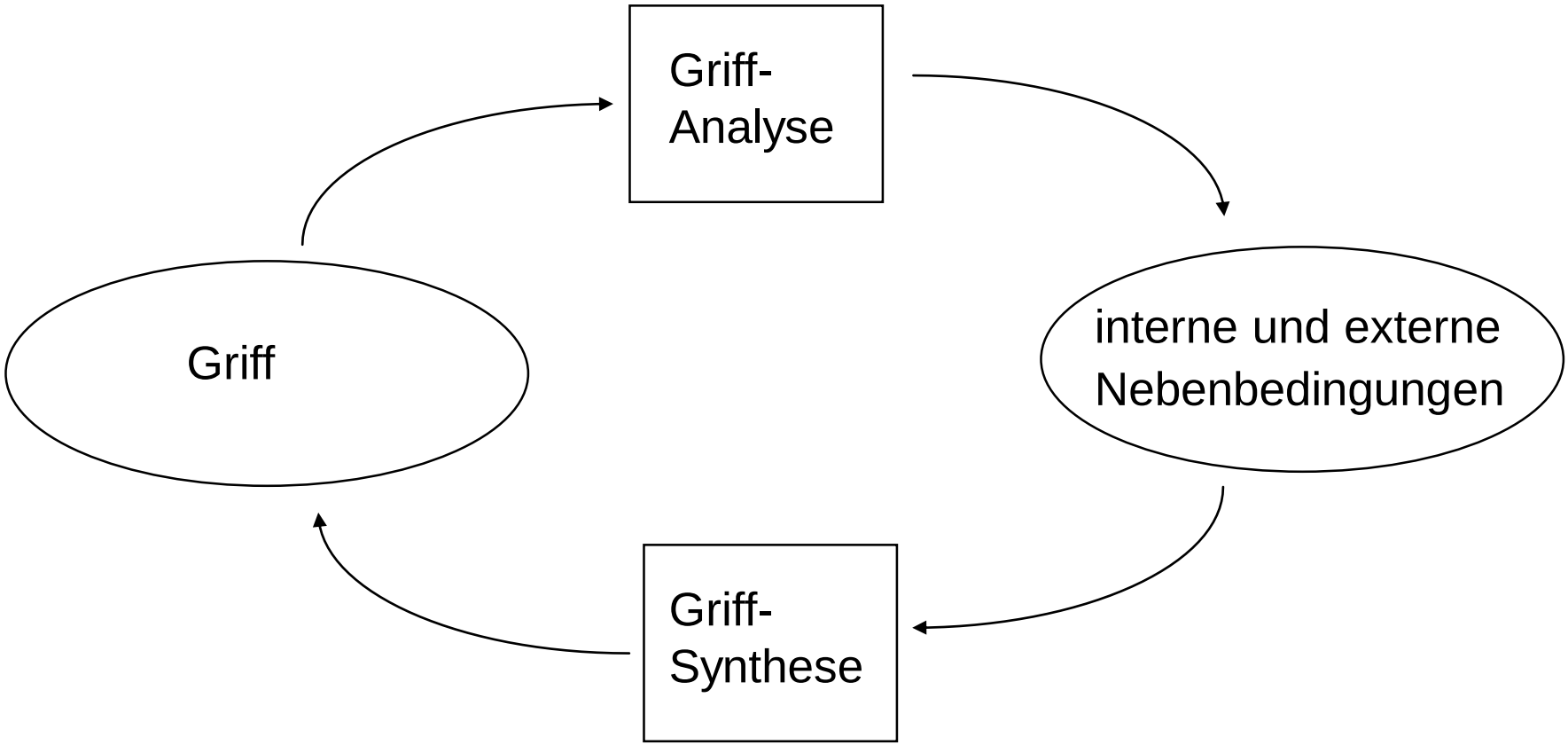
Bewegungstypen



(a) Aufnahmekonfiguration, (b) Ablagekonfiguration, (c) Transferbewegung, (d) Anrückbewegung, (e) Erreichen der Aufnahmekonfiguration,

(f) Greifen des Objektes, (g) Abrückbewegung mit Objekt,
(h) Transferbewegung mit Objekt, (i) Anrückbewegung mit Objekt,
(j) Erreichen der Ablagekonfiguration, (k) Loslassen des Objektes,
(l) Abrückbewegung.

Griffgenerierung: Planungsschritte



Griffgenerierung: Nebenbedingungen

Interne Nebenbedingungen eines Griffs:

1. **Gültigkeit:** Überlappung zwischen Greifmerkmalen des Objektes und der Finger
2. **Kollisionsfreiheit:** keine Kollisionen zwischen Greifer und gegriffenem Objekt
3. **Zugänglichkeit:** kollisionsfrei erreichbar

Externe Nebenbedingungen:

1. **Kollisionsfreie Anrückbewegung des Greifers:** Keine Kollisionen zwischen Roboterarm, Greifer, benachbarten Objekten und der Arbeitsebene
2. **Kollisionsfreie Abrückbewegung mit gegriffenem Objekt**
3. **Berücksichtigung der Roboterkinematik:** Selektierter Griff liegt im Arbeitsraum des Roboters, An-/Abrücktrajektorien können abgefahren werden.

Griffgenerierung: Nebenbedingungen

4. **Stabilität eines Griffes:** relative Lage / Orientierung des Objektes zum Greifer verändert sich nicht
5. **Stabilität der Szene:** Abrückbewegung des Greifers mit Objekt sollte die Stabilität der Szene nicht beeinflussen
6. **Aufgabenabhängigkeit:** zur Aufnahme- und Ablagekonfiguration des Objektes kompatiblen Griff wählen
   - Wenn ein Griff bestimmt werden kann
     --> bestimme Umgreifsequenz
   - Ausübung von Kräften / Momenten erforderlich
     --> wähle Griff dementsprechend

Wie generiert man „gute“ Griffe?

Problem der Dimensionalität:

- Bsp.: Menschliche Hand: 21 Freiheitsgrade (Gelenke)
- Pose der Hand relativ zum Objekt: 6 Freiheitsgrade
- Insgesamt: Problem der Dimension 27
- Zusätzlich zu berücksichtigen:
  - Stabilität: Objekt sicher halten, nicht fallenlassen
  - Ausregelung eines Griffes: Wieviel Kraft anwenden?
    Rohes Ei mit gleicher Kraft greifen wie Kaffeetasse?
  - Task-Spezifität: Welcher Griff für welche Aufgabe?
  - Griff kinematisch erreichbar? (Inverse Kinematik)
  - Hindernisse im Weg?

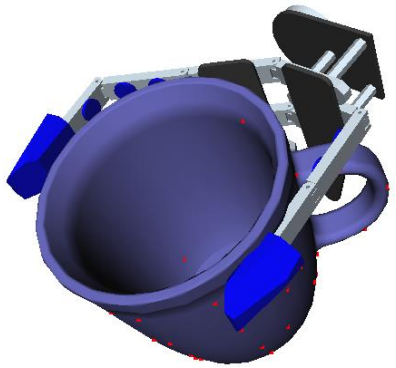
Fazit:

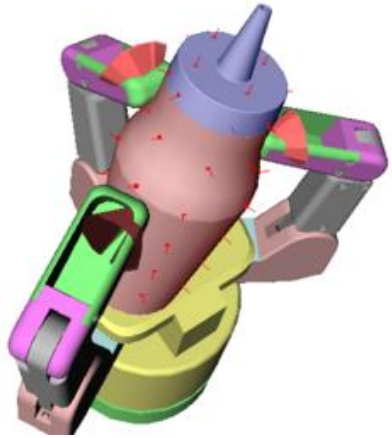
- Greifen als hochdimensionales Problem
- Analytisch kaum zu lösen

Ausweg: Simulation (GraspIt!, OpenRAVE, OpenGRASP)

Ausweg: Griffe im Greifsimulator generieren

- **Genereller Ansatz: „Generiere-und-Teste“**
- **Algorithmus:**
  1. Handmodell und Objektmodell in Simulationsumgebung laden
  2. Griffkandidaten generieren:
     - Anrückvektor der Hand an das Objekt
     - Zielpunkt auf der Objektoberfläche
     - Rollwinkel der Hand um den Anrückvektor
     - „Preshape“ der Hand: Konfiguration (Gelenkwinkelvektor) der geöffneten Hand z.B.:
       - Sphärisch, parallel, …
     - Verschiedene Heuristiken für die Kandidatenerzeugung, um Suchraum zu verkleinern
  3. Für jeden Griffkandidaten:
     - Bewege Hand (entlang des Anrückvektors) auf Zielpunkt auf dem Objekt zu, bis Kollision
     - Schließe Finger, bis Kontakt
     - Bestimme Kontaktpunkte
     - Werte Gütekriterium für Stabilität aus (meist: Force closure, d.h. Kraftschluss)

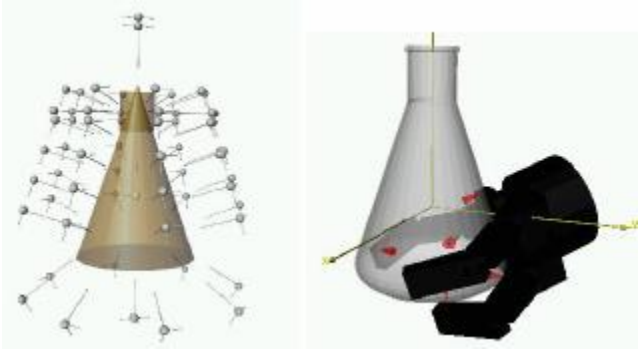




Heuristiken für Griffkandidatenerzeugung I

- **Ansatz:** Approximation des Objektes durch Formprimitiven (Boxen, Kugeln, Zylinder, Kegel)

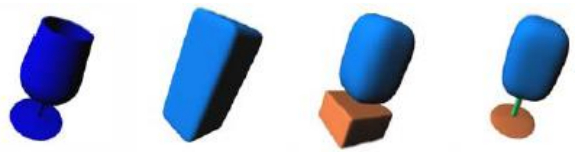
- **Vorgehensweise:**
  - Nachbau eines vereinfachten Objektmodells aus Formprimitiven
  - Generiere Griffkandidaten für die einzelnen Teile des Primitiven-Modells nach bestimmten Regeln, z.B.:
    - Anrückrichtungen senkrecht zu Zylindermantel / Kegelmantel, senkrecht zu Flächen einer Box
    - alle vordefinierten Preshapes durchprobieren
- **Nachteil:**
  - Zerlegung von Objekten in Formprimitiven nur von Hand möglich



Heuristiken für Griffkandidatenerzeugung II

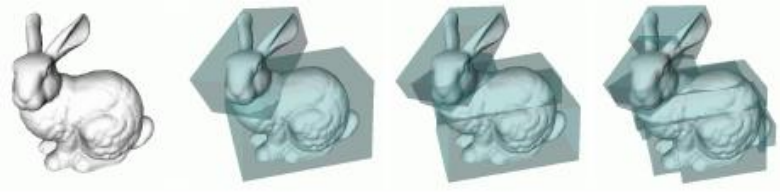
- **Ansatz:** Approximation des Objektes durch Superquadriken

- **Vorgehensweise:**
  - Zerlegung des Modells in einen Baum aus Superquadriken (Decomposition Tree)
  - Generiere Griffkandidaten auf den einzelnen Superquadriken
- **Vorteil:**
  - Automatische Zerlegung möglich

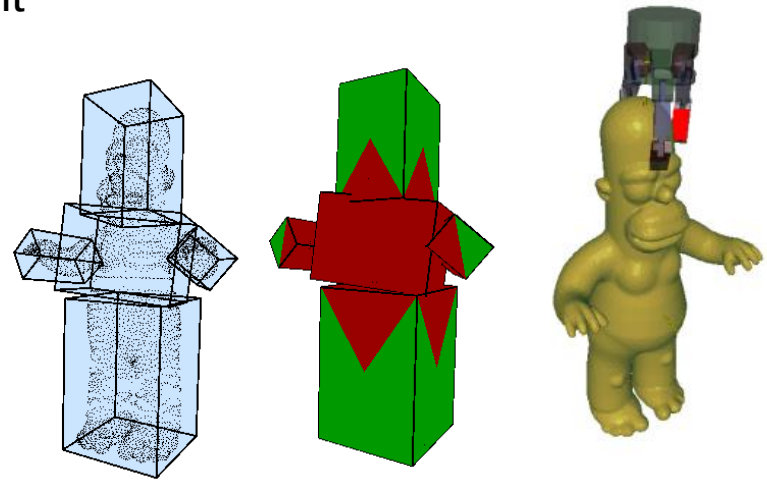


Heuristiken für Griffkandidatenerzeugung III

- **Ansatz:** Box Decomposition



- **Vorgehensweise:**
  - Stelle Objekt dar durch eine Menge von Minimum Volume Bounding Boxes
  - Generiere Griffkandidaten auf den einzelnen Boxen (Anrückrichtungen immer senkrecht zu Flächen der Boxen)
- **Vorteile:**
  - Automatische Zerlegung möglich
  - Auch auf unbekannten Objekten möglich (Daten aus Stereokamera oder Laserscanner)
- **Nachteile:**
  - Sehr grobe Approximation der Geometrie
  - Liefert daher wenige Griffkandidaten

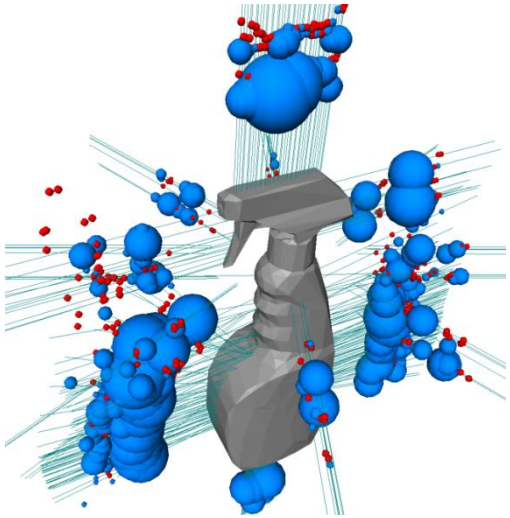


Heuristiken für Griffkandidatenerzeugung IV

- **Ansatz:** Mediale-Achse-Transformation

- **Idee:** Schreibe Kugeln maximalen Durchmessers in das Objekt ein. Die Vereinigung all dieser Kugelmittelpunkte ist die sog. Mediale Achse (im Bild: grün)

- **Vorgehensweise:**
  - Werte Schnittebenen der Medialen Achse aus (im Bild: ganz rechts)
  - Erzeuge mittels Hauptkomponentenanalyse der Kugelmittelpunkte Griffkandidaten

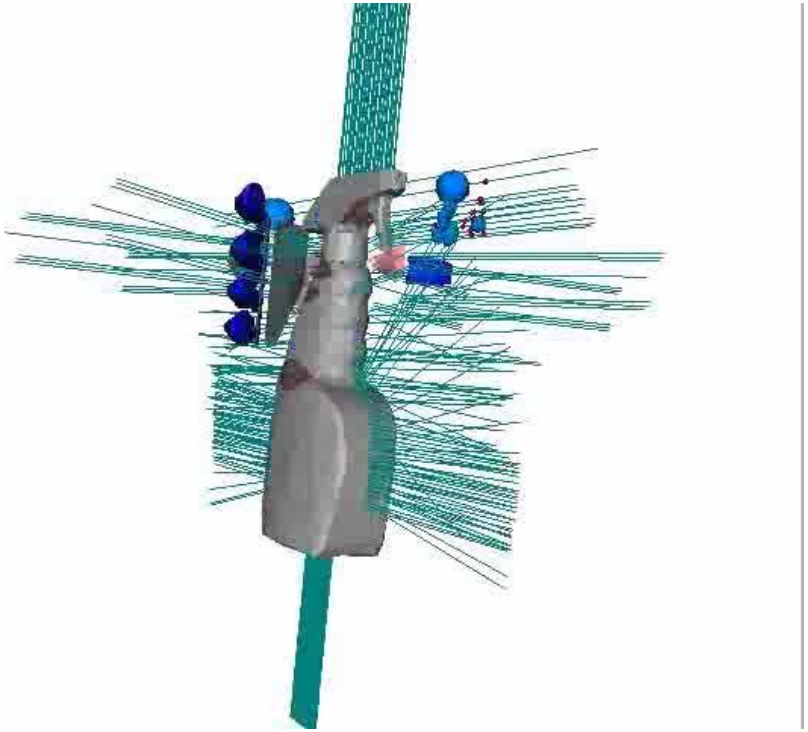


- **Vorteil:**
  - Sehr genaue Approximation der Geometrie
  - Kugelradien liefern Information über lokale Dicke (Greifbarkeit des Objektes!)

Greifplanung im Simulator

Visualisierung

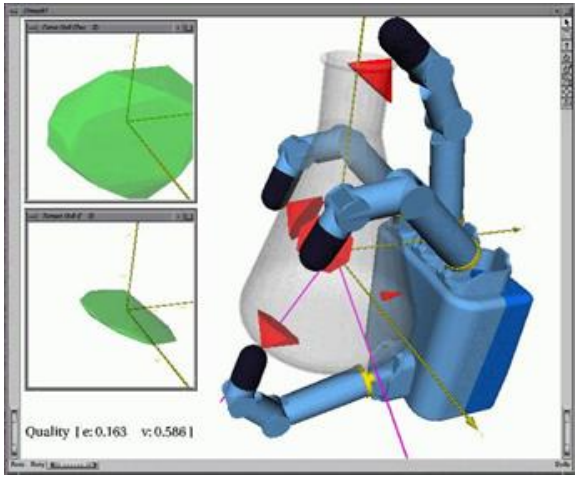
- Grüne Linien: Anrückrichtungen der Hand an das Objekt
- Kugeln: Position des Handgelenks während eines Griffes
- Blaue Kugeln: Stabile Griffe
- Rote Kugeln: Instabile Griffe
- Kugeldurchmesser: proportional zur Force Closure Bewertung (größere Kugeln entsprechen stabileren Griffen)



Gütemaß für Stabilität: Kraftschluss

- **Anschaulich:** „Wie gut widersteht ein Griff extern einwirkenden Kräften?“

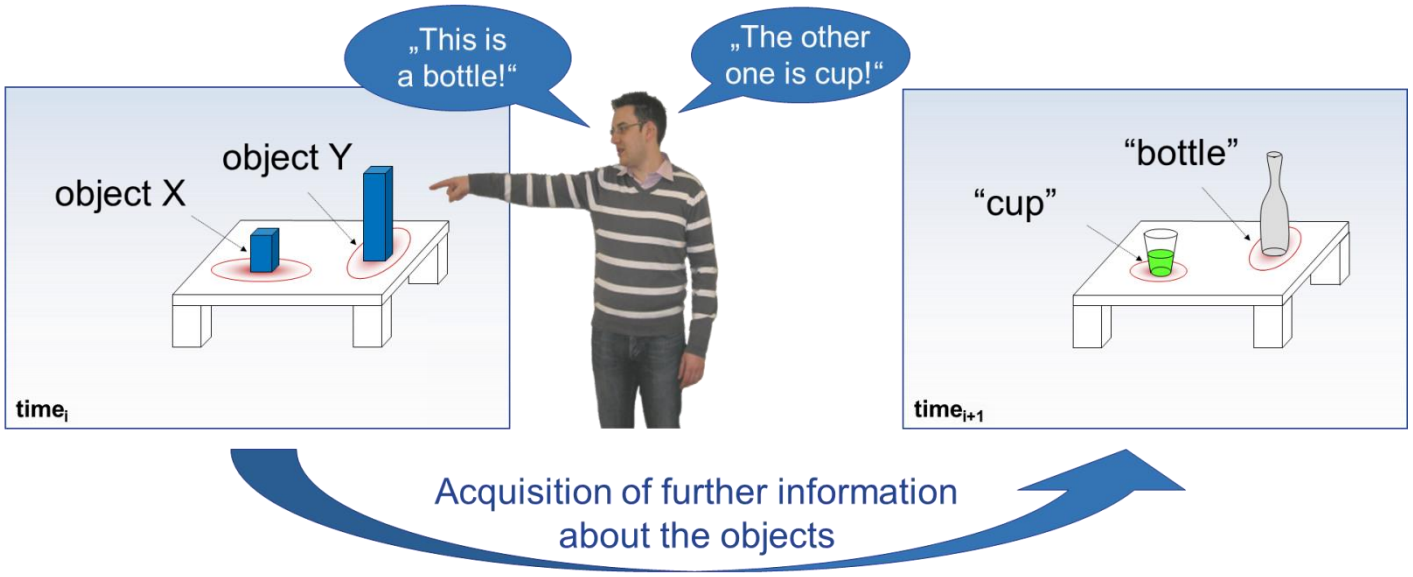
- **Berechnung:**
  - Bestimme **Kontaktpunkte** und –normalen zwischen Hand und Objekt.
  - Berechne sog. **Reibungskegel** an den Kontakten (Breite der Reibungskegel abh. vom Haftreibungskoeffizient).
  - Berechne **Grasp Wrench Space** GWS (in 6D) als konvexe Hülle über den Reibungskegeln.
  - Minimale Distanz vom Ursprung des GWS zu einer Facette des GWS ist ein Maß für die Stabilität eines Griffes.



Rot: Reibungskegel
Grün: Projektionen des Grasp Wrench Space

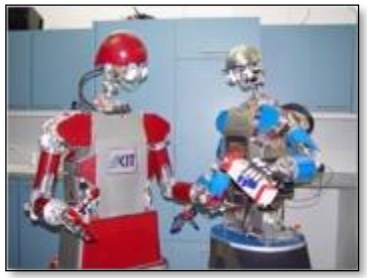
Interaktive Exploration und Lernen

- Multimodal saliencies- and knowledge based robot attention
- Modelling of multimodal referenced objects, f.e. pointing gestures and/or speech, in the environment of the robot
- Goal oriented, interactive dialogues for completion of knowledge
- Association of audio-visual object models of unknown objects with linguistic labels



Motivation für Lernen aus Beobachtung

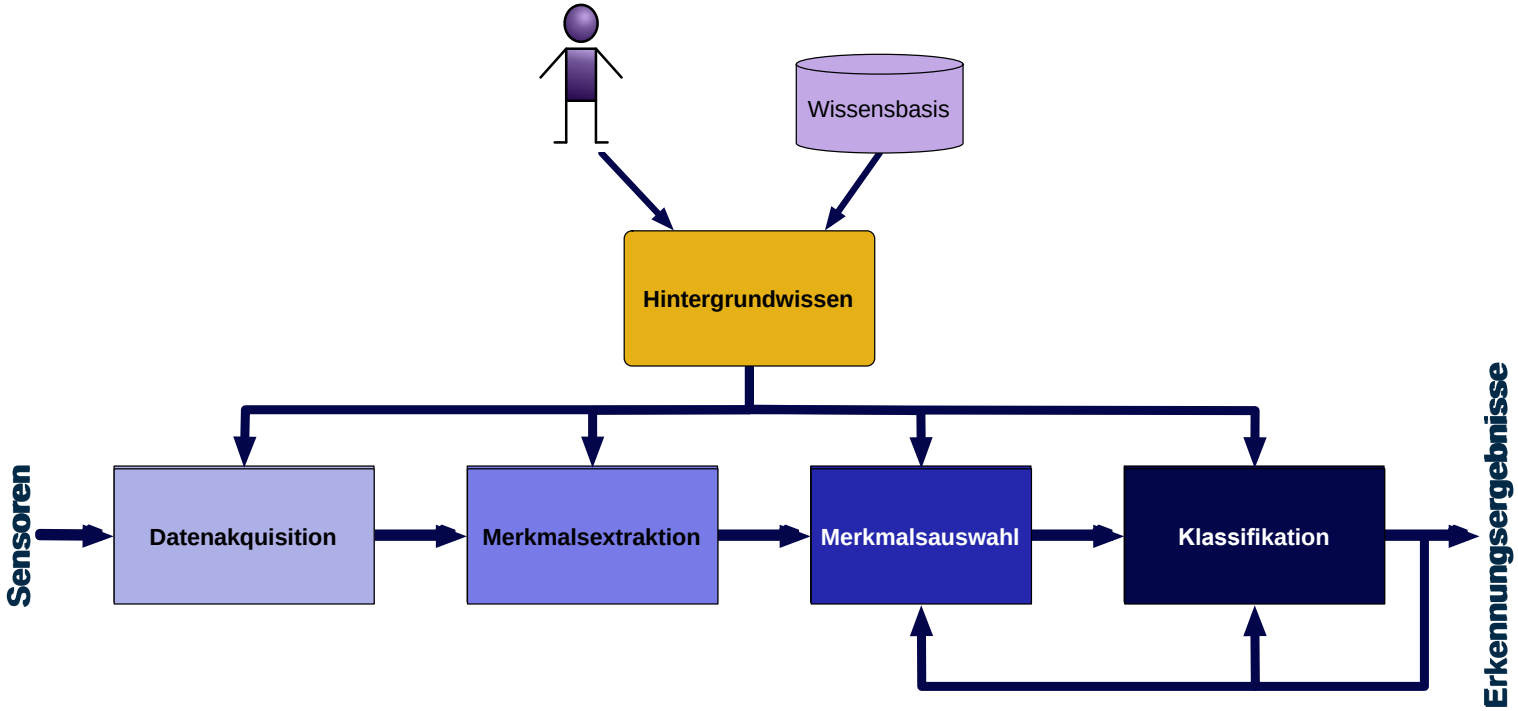


- Intuitive and effizeient Programmierung alltäglicher Manipulationsaufgaben für Serviceroboter und humanoide Roboter
- Problem: Große Anzahl unterschiedlicher Objekte, Hindernisse und Kontexte→ Flexible Ausführung erforderlich
- Ziel: Learning von Ziel-orientierter restriktionsberücksichtigender Bewegungsplanung
- Lösung: Programmieren durch Vormachen
- Problem: Spezialisierung gelernter Aufgabenmodelle an aktuelle Szenarien und Reduzierung von Planungsaufwand→Nutzung von Erfahrungswissen

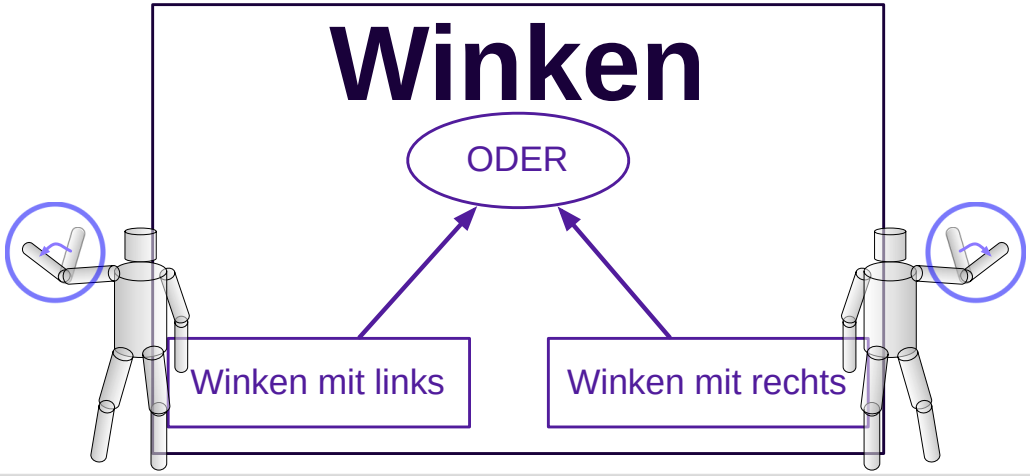
Lösungskonzept zur Handlungsbeobachtung

- Modellierung von Aktivitäten als Zeitreihe von beobachteten Bewegungen
- Verwendung abstrahierter, symbolisch deutbarer Merkmale zur Abstraktion konkreter Sensordaten
- Nutzung von Hintergrundwissen und Gelerntem zur Verbesserung der Lernprozesse



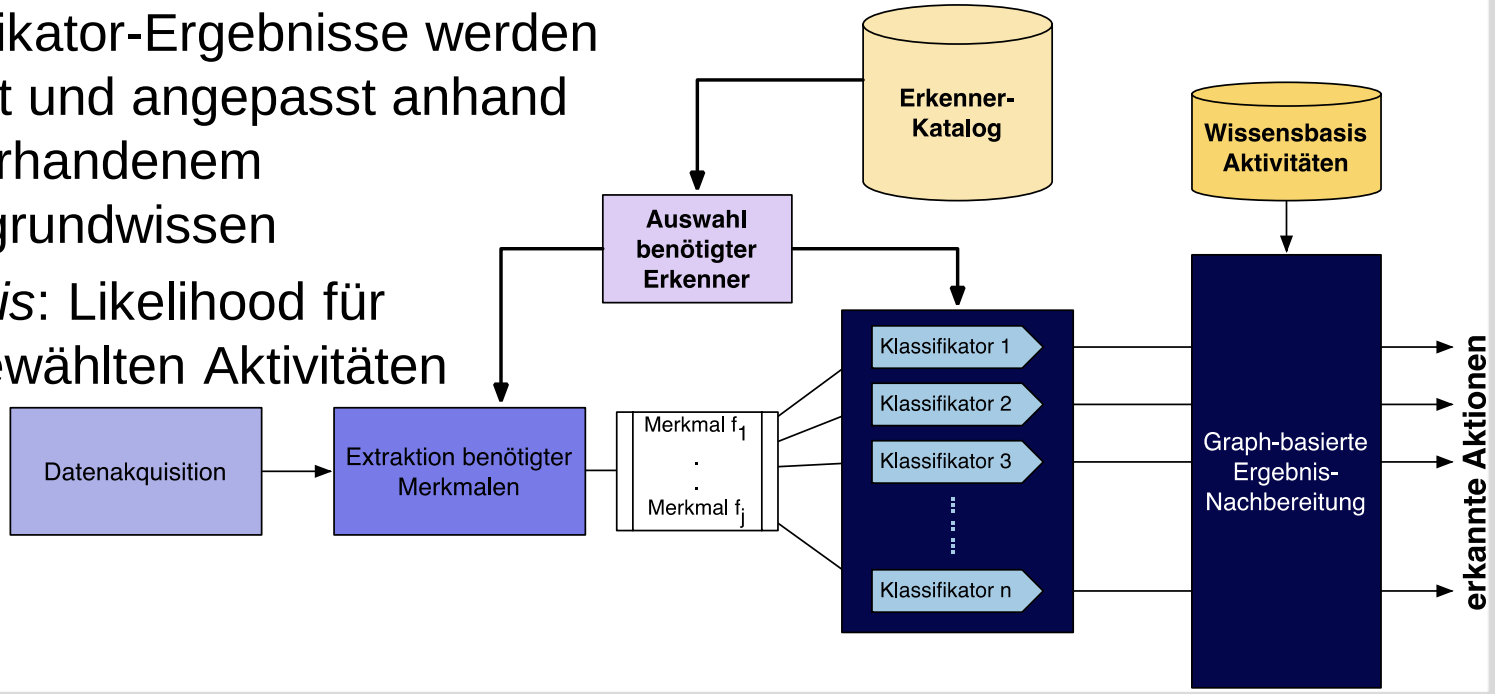
Eigenschaften der Handlungserkennung

- Nutzung nur ausgewählter Merkmale zum Trainieren von Erkennern
- keine Multiklassen-Klassifikation: für jede Aktivität wird ein unabhängiger Erkenner trainiert
  - ⇒ Erkenner können beliebig zur parallelen Erkennung verschiedener Aktivitäten zusammengefasst werden
  - mehrere Erkenner können zur Erkennung allgemeinerer/speziellerer Aktivitäten kombiniert werden

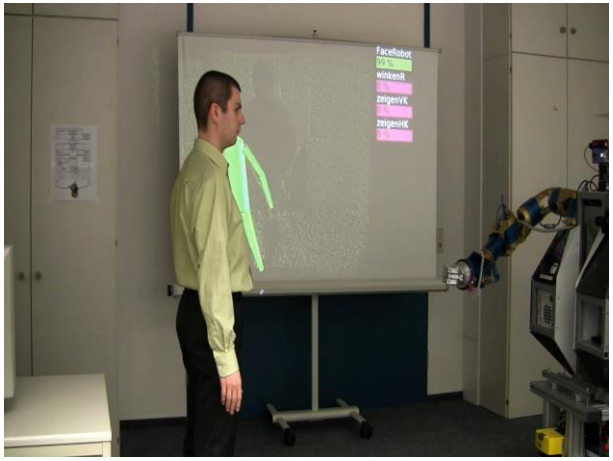


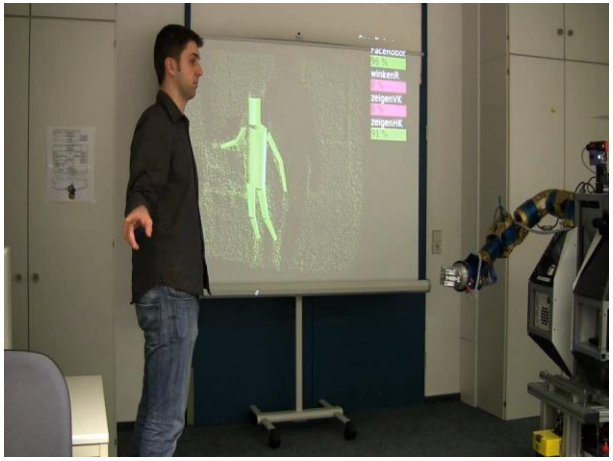
Durchführung der Erkennung

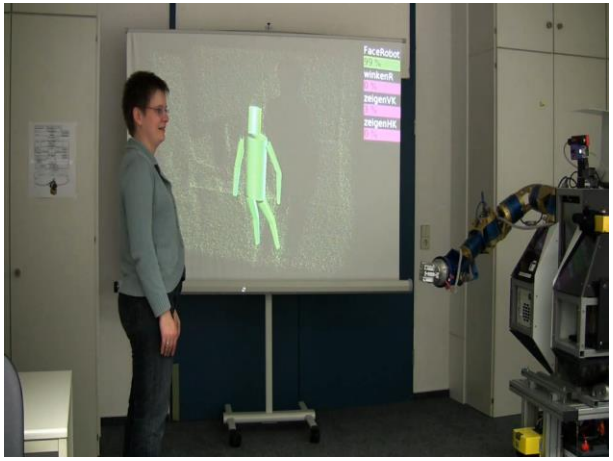
- *Vorbereitung*: Szenen/Kontext-abhängige Auswahl benötigter Erkenner
  - Merkmalsextraktion für benötigte Merkmale konfigurien
- *Schritte* in der Erkennung:
  - aus Rohdaten werden nur benötigte Merkmale extrahiert
  - jeder Klassifikator erhält „seine“ Merkmale
  - Klassifikator-Ergebnisse werden gefiltert und angepasst anhand von vorhandenem Hintergrundwissen
- *Endergebnis*: Likelihood für jede der gewählten Aktivitäten



Beispielszenen

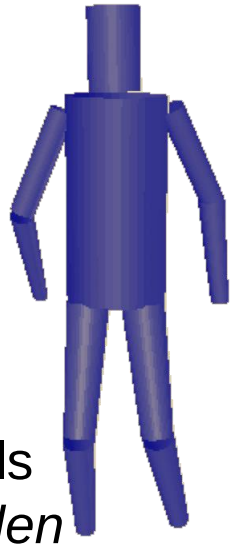






Modellierung von Bewegungsmerkmalen

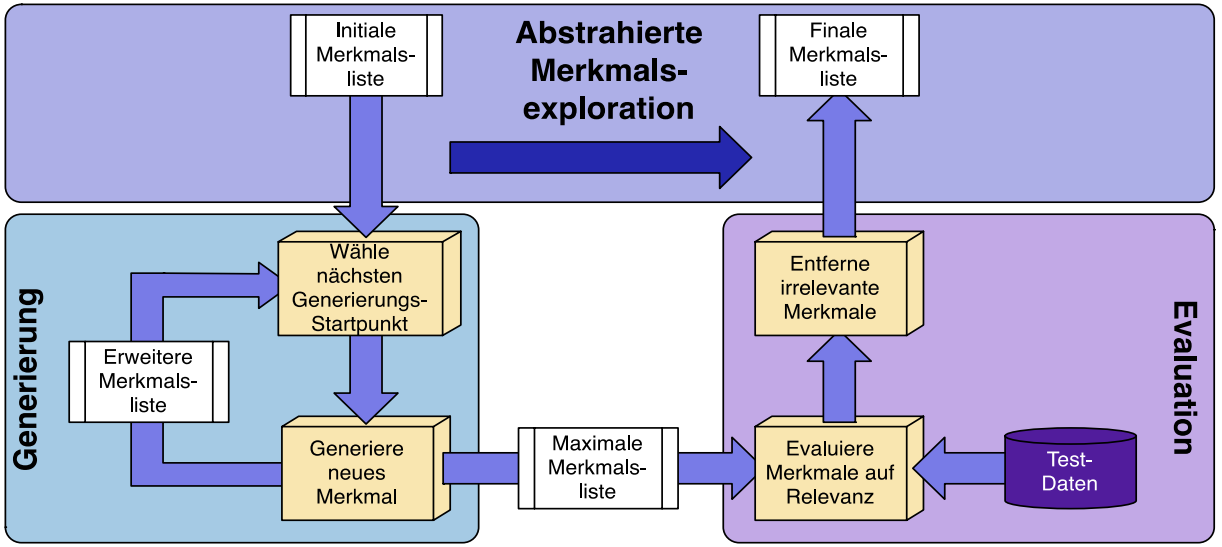
- Ziele
  - Erweiterbarkeit für neue Daten
  - Erweiterbarkeit für weitere Extraktionsmethoden
  - Symbolische Interpretierbarkeit



- Generische Repräsentation von Merkmalen als Baumstruktur von *Merkmalsextraktions-Modulen (MEM)*
  - MEMs sensorunabhängig definiert
  - wiederholte Anwendung von MEMs spannt hochdimensionalen Merkmalsraum auf

Verfahren zur Merkmalsexploration

- **Ziel:** Schnelle Erschließung neuer Domänen durch Merkmals-Exploration zur Suche potentiell relevanter Merkmale
- **Konzept:**

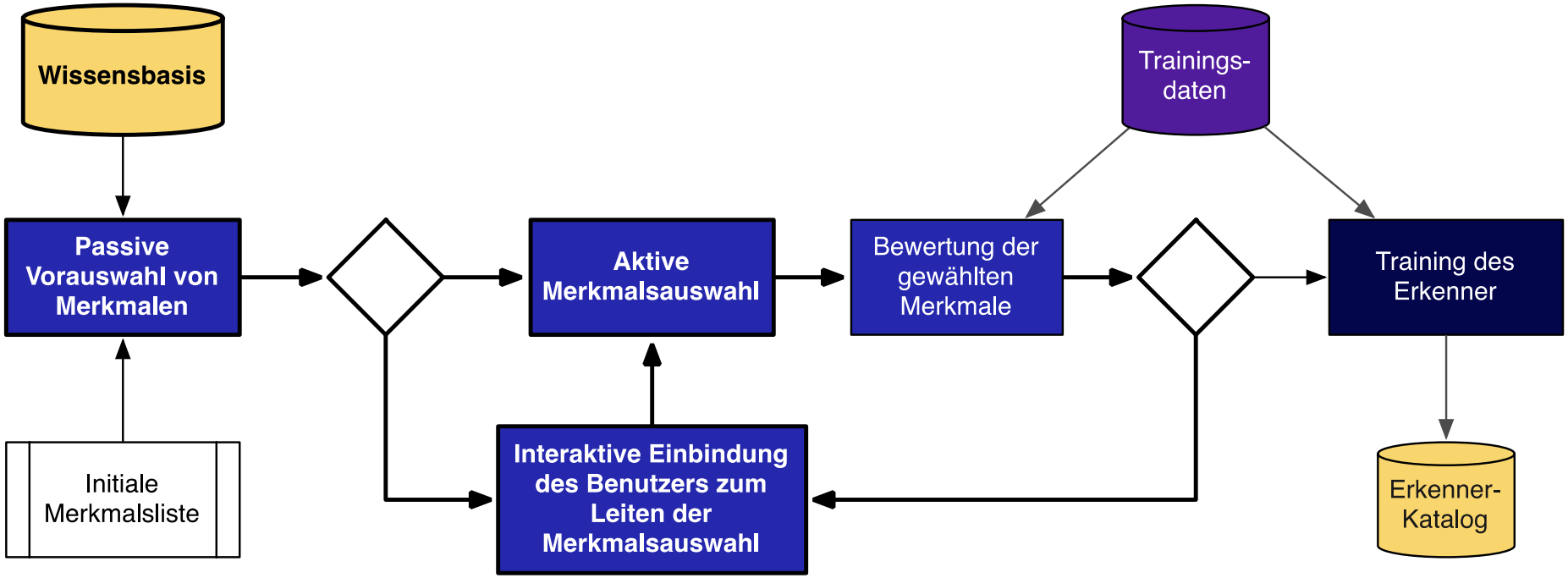


- **Ergebnisse:**
  - Generierung neuer Merkmale unter Verwendung verschiedener Strategien
  - Bewertung der Merkmale anhand repräsentativer Trainingsdaten

Merkmalsauswahl

- **Zweck:** Nur Verwendung jeweils relevanter Merkmale für Training und Erkennung spezifischer Aktivitäten
  - Verringerter Einfluss von Rauschen in Sensordaten
  - Bessere Generalisierung
- **Konzept:** Verbindung von
  - aktiver Auswahl mittels *Deep Correlation-Based Filter*-Algorithmen
  - passiver Auswahl auf Basis einer Hintergrund-Wissensbasis
  - interaktiver Auswahl unter Einbeziehung des Benutzer

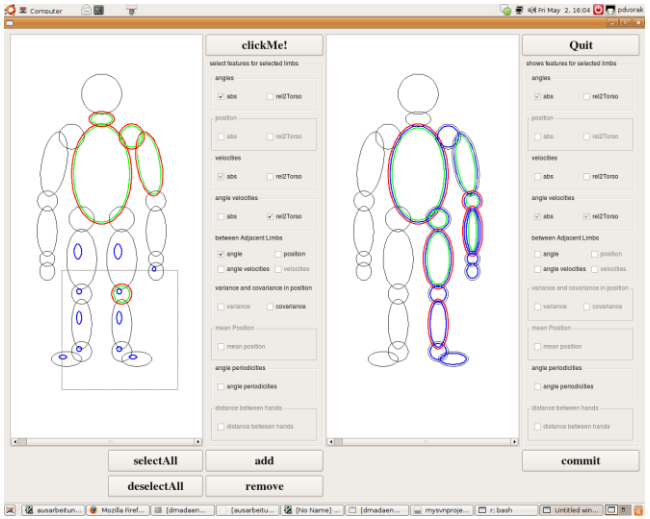
Schema der Merkmalsauswahl



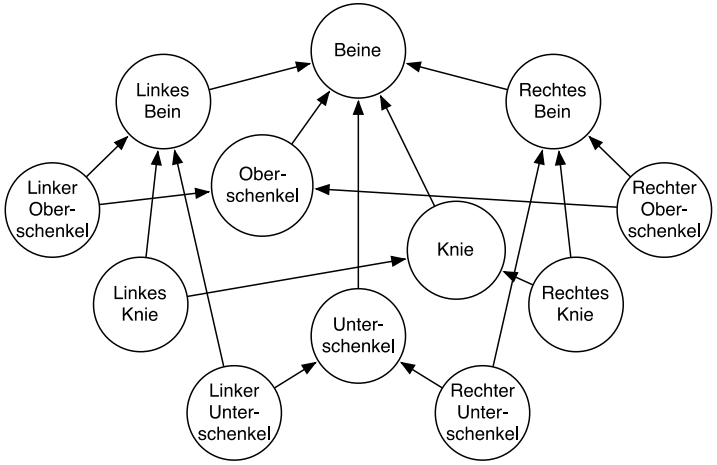
- Interaktiver Schritt kann iterativ ausgeführt werden

Interaktion in der Merkmalsauswahl

- Einbindung des Trainers über GUI
  - Vorauswahl beteiligter/unbeteiligter Körperteile
  - „Art“ der Aktivität

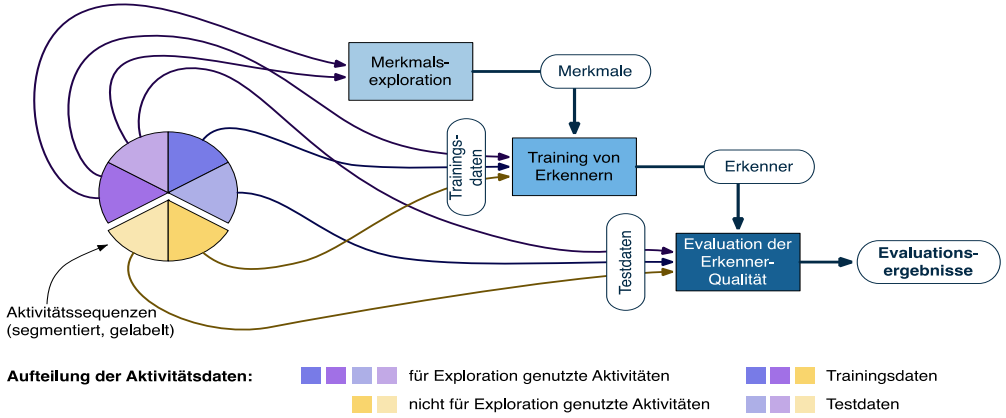


- Verbindung von Benutzereingaben zu Merkmalen über Modellierung von Merkmalstaxonomien
  - Modellierung des Zusammenhangs zwischen Merkmalsmengen als azyklische Graphen
  - Taxonomien haben Verbund-Struktur

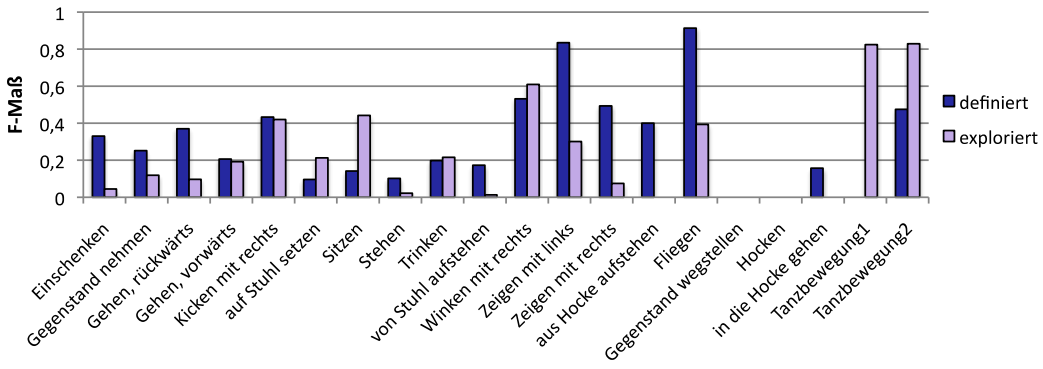


Ergebnisse Merkmalsexploration

- Direkte Bewertung explorierter Merkmale nicht ohne weiteres möglich
- Betrachtung der Erkennungsergebnisse:

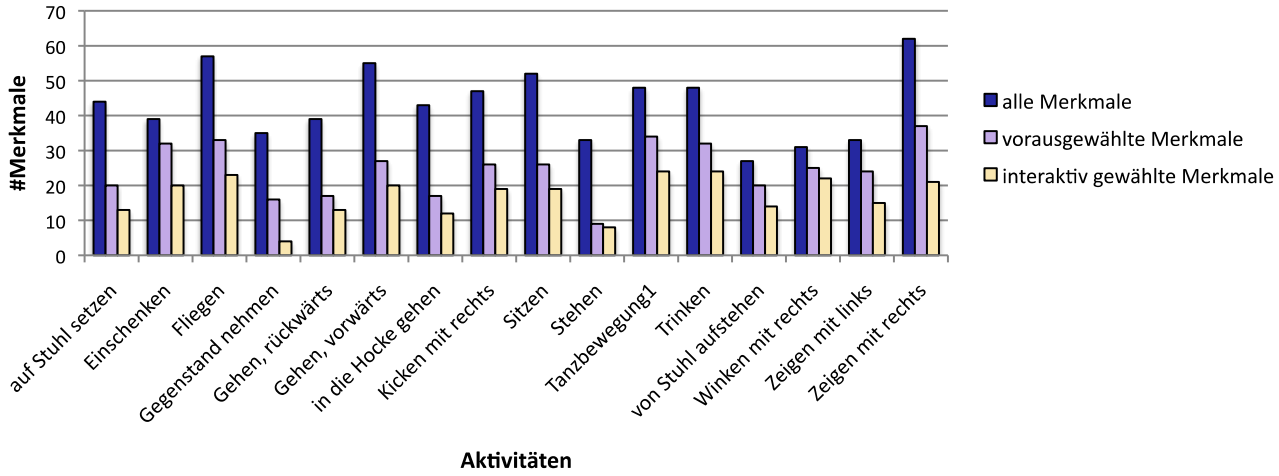


- Vergleich von Ergebnissen mit handmodellierten und explorierten Merkmalen

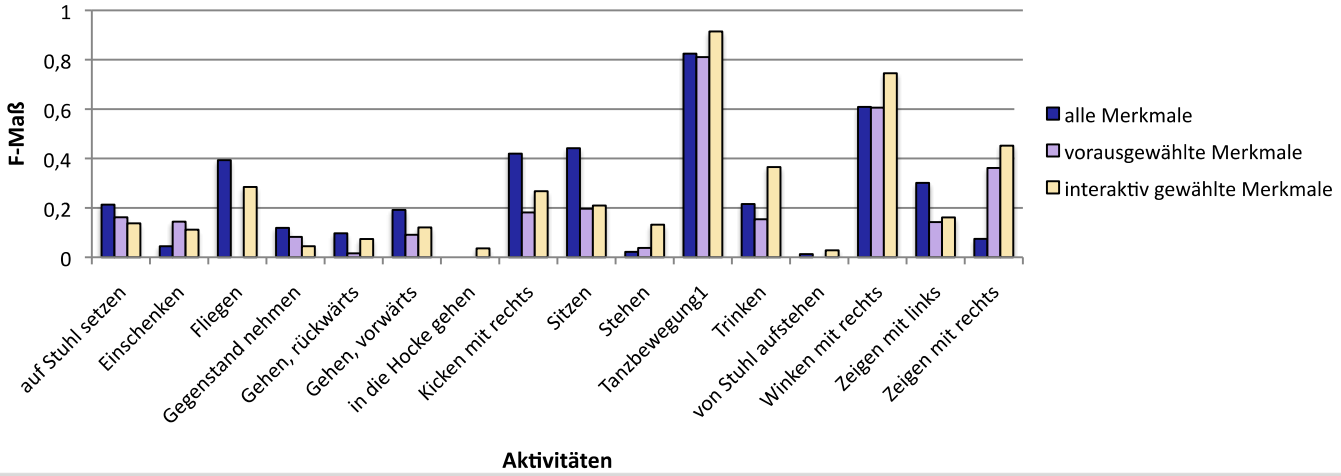


Ergebnisse Merkmalsauswahl

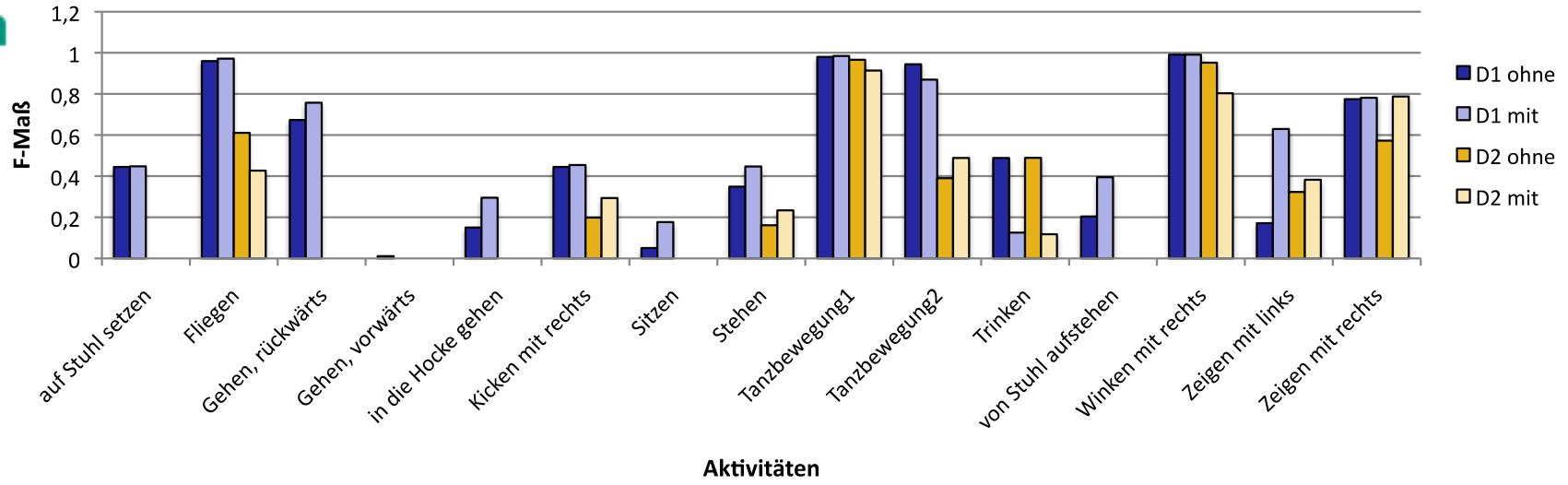
- Anzahl



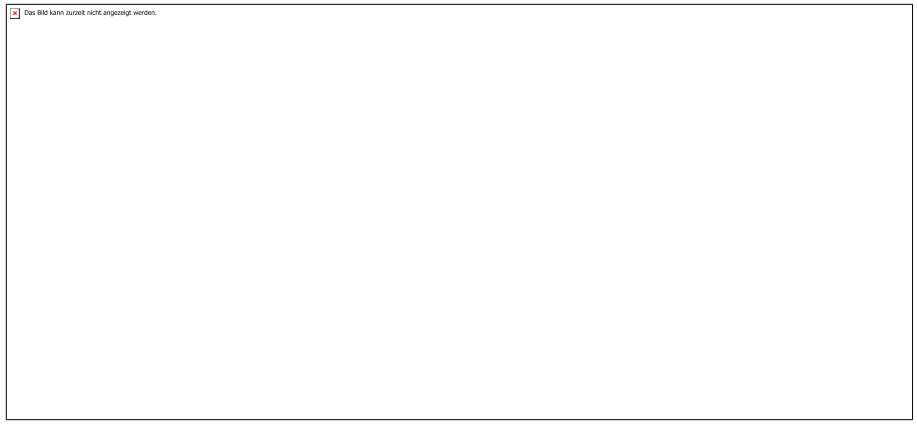
- Erkennungsqualität:

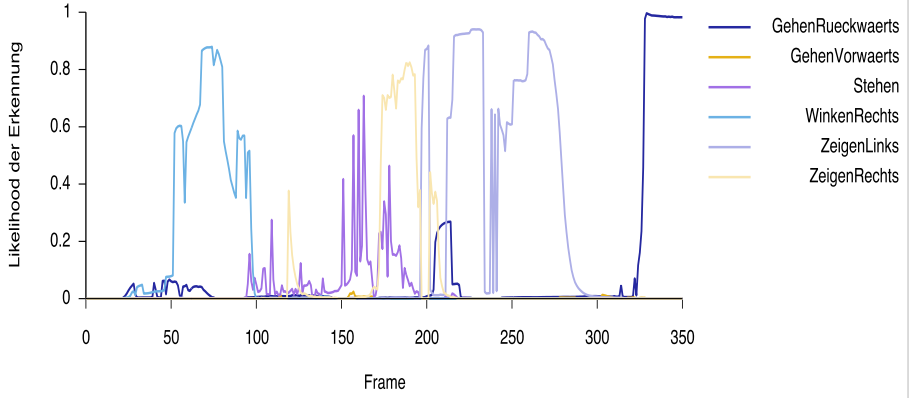


Ergebnisse Klassifikation



- Ergebnisse auf vollständiger Handlungssequenz:





Ausblick Aktivitätserkennung

- *Vision: Autonomes Lernen und Erkennen von Aktivitäten durch robotische Systeme*

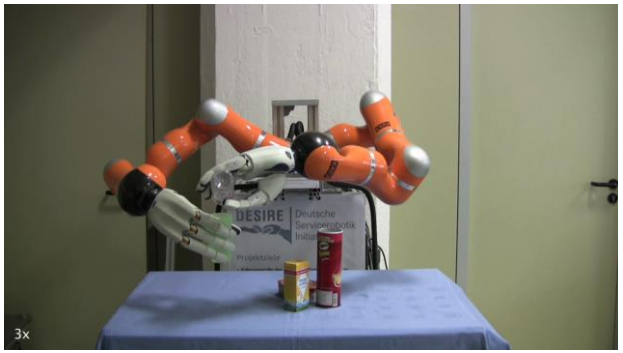
- Integration des Lernystems auf Roboter
  - Lernen vollständig in Interaktion mit Roboter statt Verwendung getrennter Rechner

- Verringerung des Aufwandes für Training durch
  - automatische Segmentierung und unüberwachte Lernverfahren
  - Adaptive Anpassung der Erkenner-Struktur abhängig von Aktivität
  - Nutzung von Bewegungsalphabeten
  - Nutzung von Bewegungsbibliotheken

Prinzip: Programmieren durch Demonstration

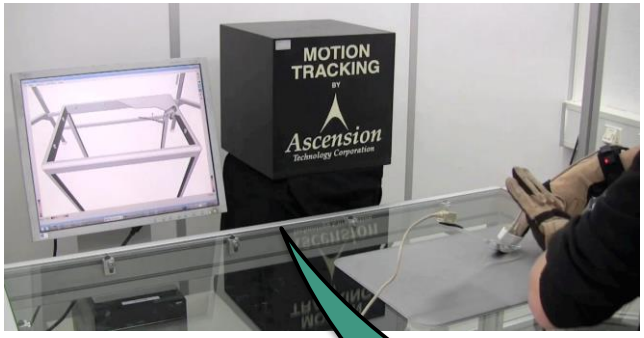
- Ein Mensch demonstriert auf natürliche Weise eine Handhabungsaufgabe unter Einbeziehung expliziten und impliziten Domänenwissens
- Die Demonstration wird aufgezeichnet und interpretiert

Specific PbD methodology

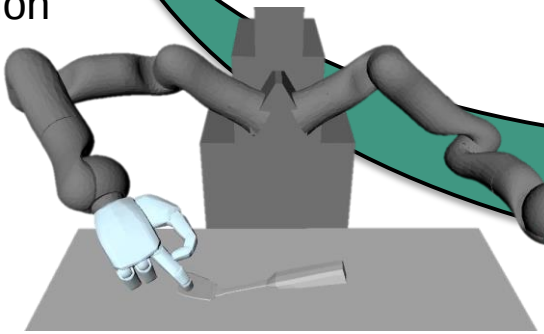


Natürliches Programmieren durch Vormachen

- Schema der intuitiven Roboterprogrammierung:
  Generierung flexibler, robuster Aufgabenrepräsentationen aus Beobachtun natürlicher Handlungen

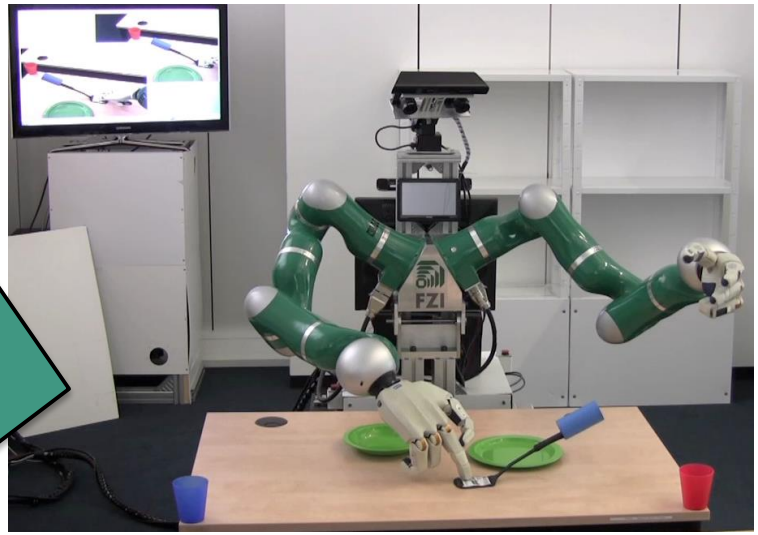


Observation



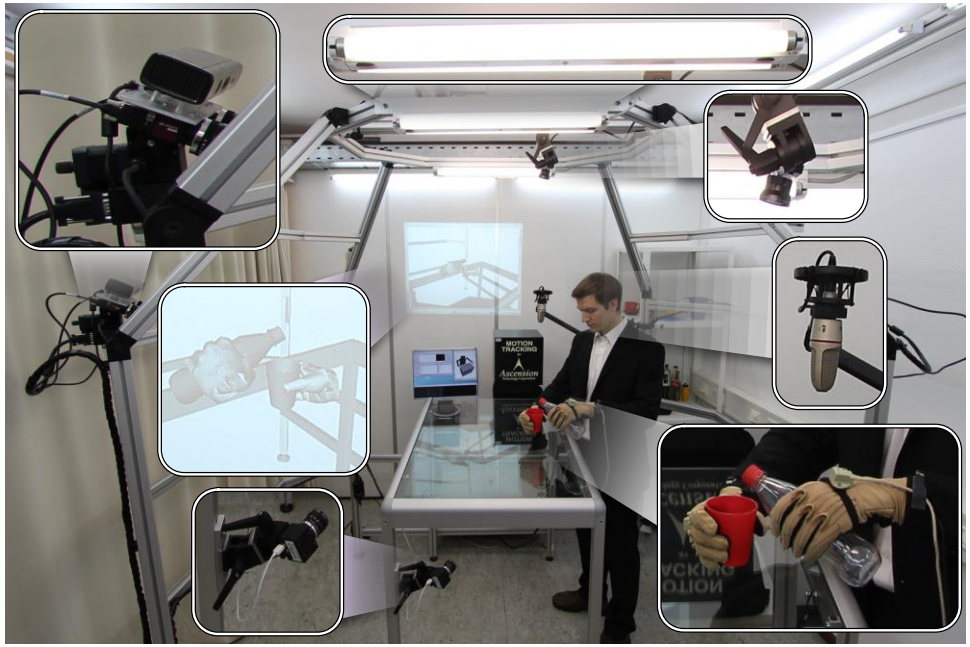
Task model

Autonomous execution

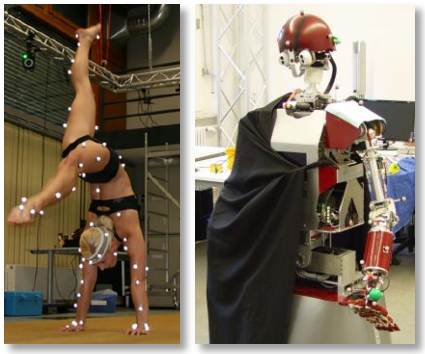


Sensoren zur Handlungsbeobachtung

- Object Modeling Center
  - Generation of high-precision, textured 3D-models
- PbD-Dome
  - Observation of human during manipulation
- Vicon Tracking System



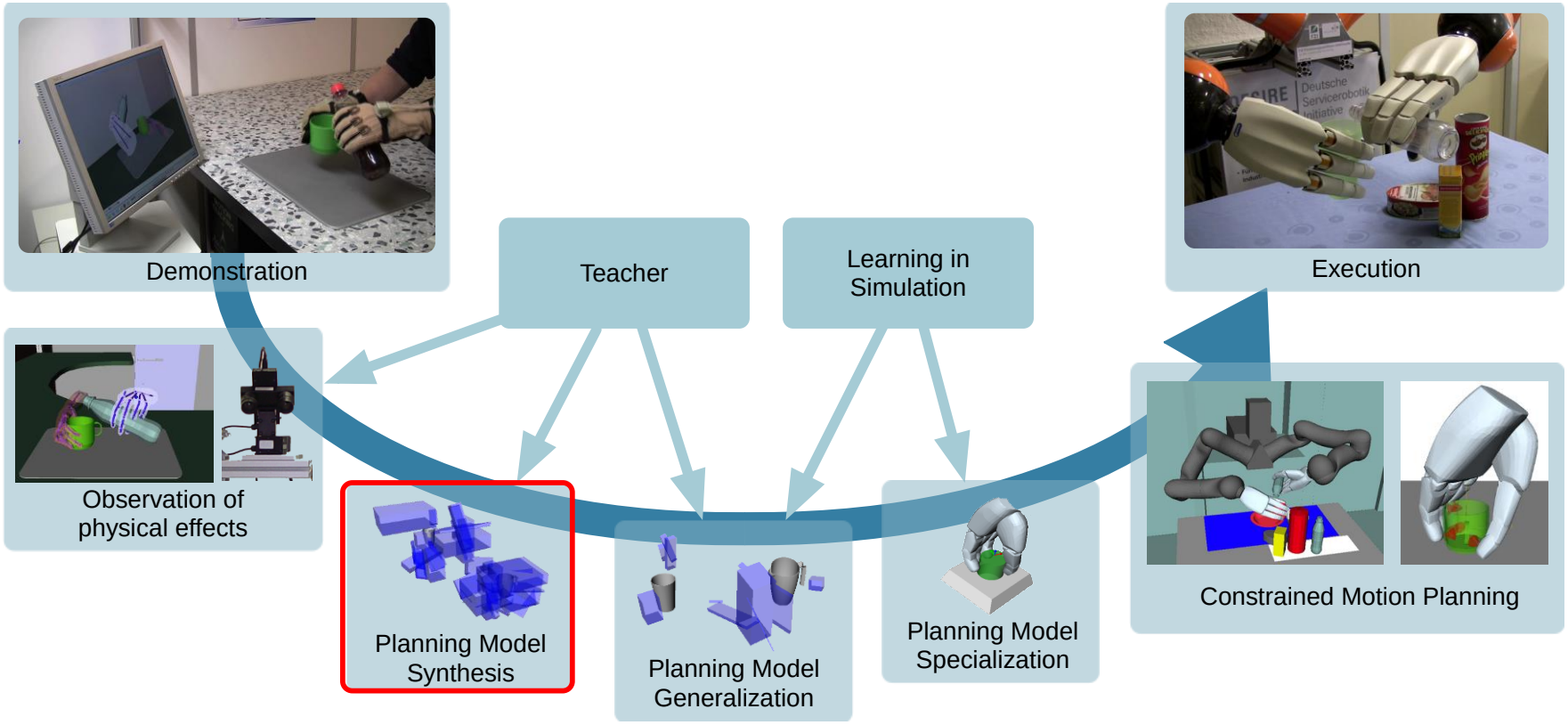
PbD-Dome



Vicon Tracking System

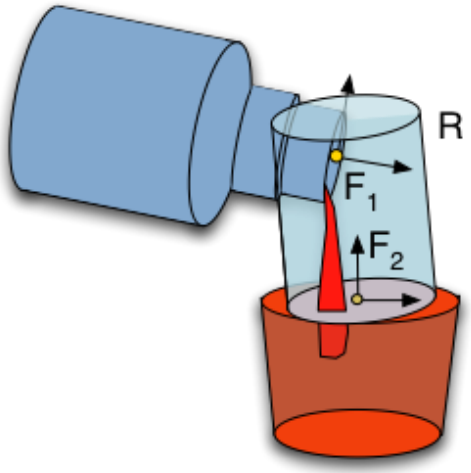
Zyklus – Programmieren durch Vormachen

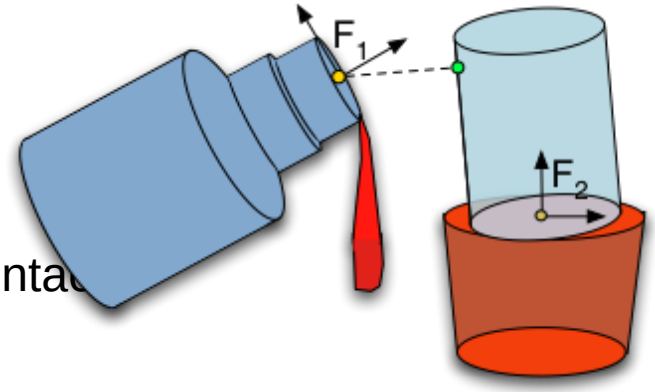
- PbD-process schema



Constraint Representation

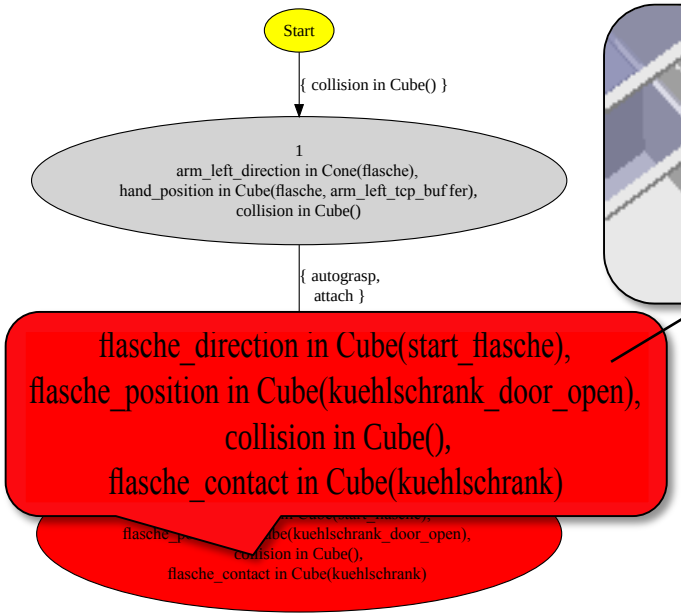
- Definition: position constraint
  - Parameters: coordinate frames F_1, F_2, region R
  - Distance
    - F_1 is transformed into F_2: ${}^2F_1 = F_2^{-1} F_1$
    - Euclidean distance of ${}^2F_1$ to R
  - Random sampling
    - Draw random sample r from R
    - Transformation of r into F_1 using $F_2 \cdot r$
  - Learning
    - For each training data θ_i:
      $F_1(\theta_i)$ is transformed into $F_2(\theta_i)$: ${}^2F_1(\theta_i)$
    - Generate parameters of R: $\forall i: {}^2F_1(\theta_i) \in R$
- Other constraints
  - Orientation, direction, force, momentum, contac
  - Compliance, spatial relations, time,…
- Implemented region types
  - cube-, cylinder-, sphere-, cone-sector, GMMs, convex hull



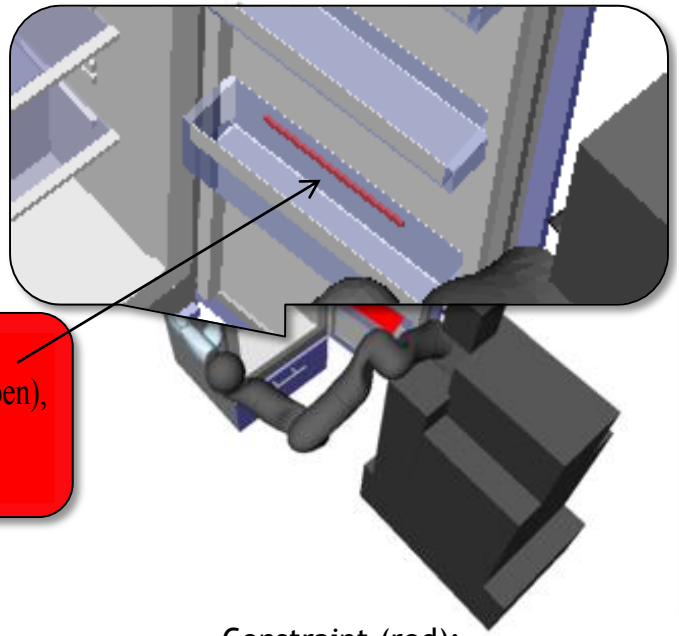


Planungsmodell

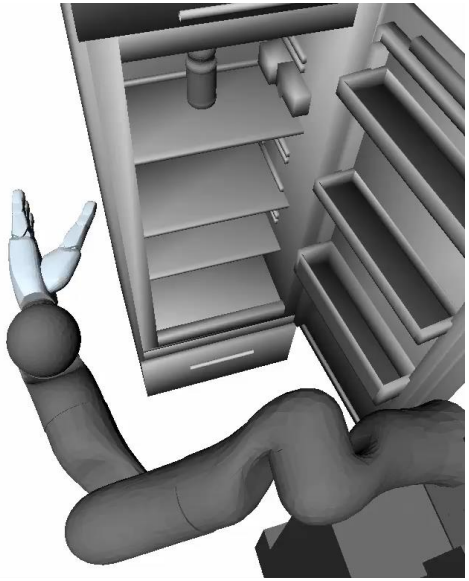
- Planning model: “Strategy graph”
  - Nodes = subgoals, e.g. “where to place the bottle in the fridge door”
  - Edges = transitions, e.g. “hold bottle upright during the transport”
  - Consistent representation based on constraints



Strategy graph



Constraint (red):
Bottle position relative to fridge door

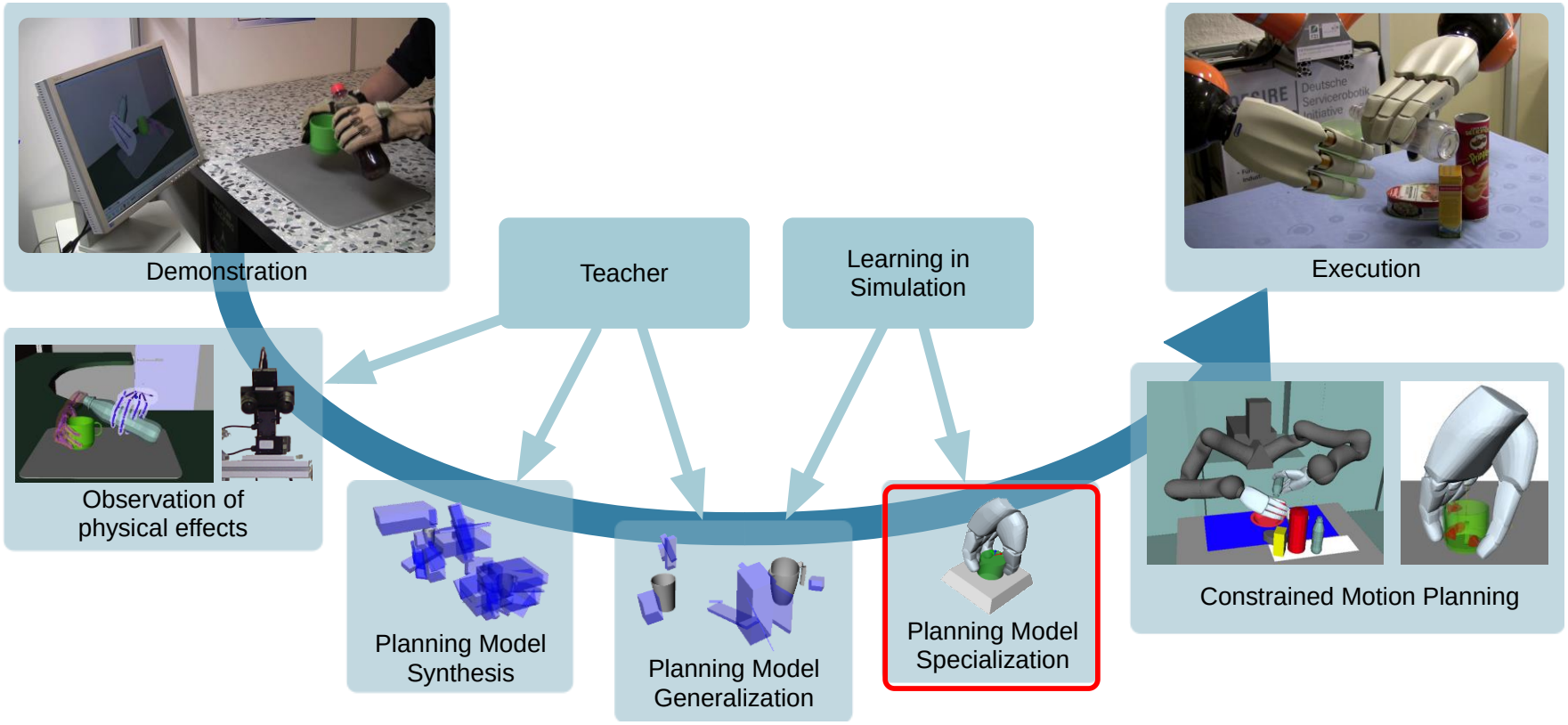


Planning and execution

[Jäkel et al., “Representation and constrained planning of manipulation strategies in the context of Programming by Demonstration”, ICRA’10]

Programmieren durch Vormachen

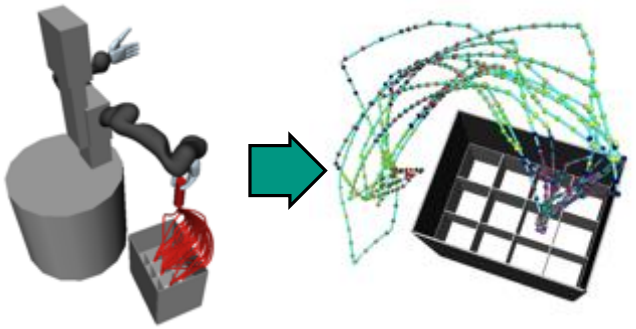
- Here: how to specialize learned planning models to the scene



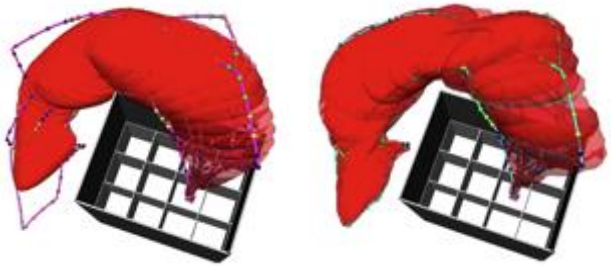
Institut für Anthropomatik und Robotik
Prof. Dr.-Ing. R. Dillmann
Kognitive Systeme

Lernen von Suchheuristiken

- Planning results are accumulated using original planning model
- Clustering of prior planning results
  - Distance: max. Dynamic Time Warping distance
  - Hierarchical agglomerative clustering
- Temporal alignment of cluster
  - Dynamic Time Warping using planning result with overall minimal distance
- Algorithms to create sequences of ellipsoids
  - GMM+GMR: Gaussian Mixture Regression based on Gaussian Mixture Model
  - Khachiyan: enclosing ellipsoid of subsequent time points



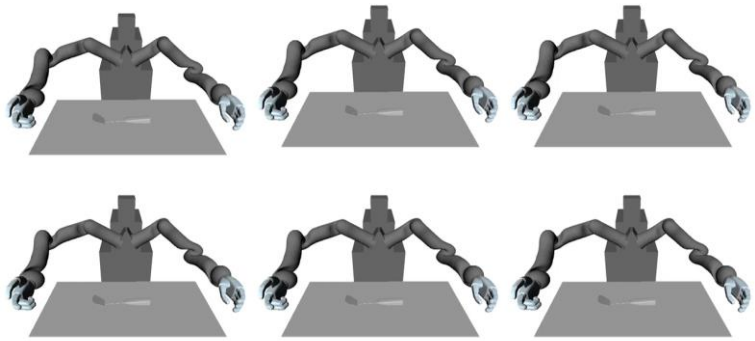
Planning results Temporal alignment



GMM+GMR Khachiyan

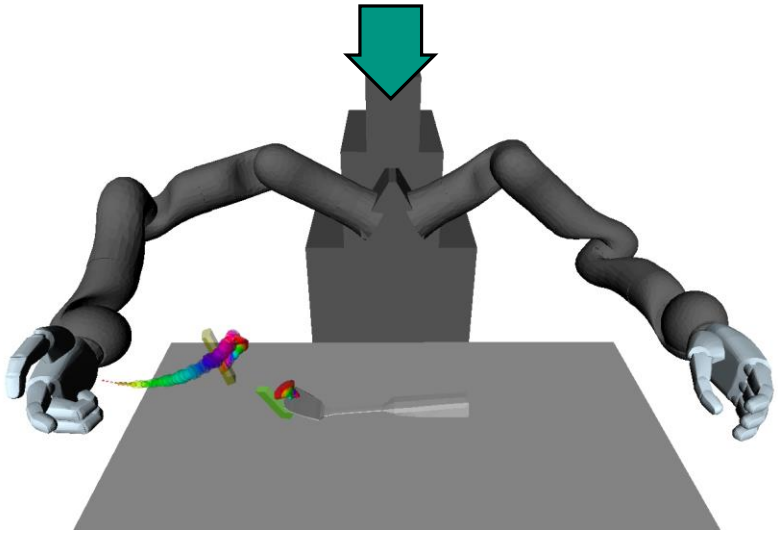
Bewegungensbeispiele

Human demonstrations



(Semi-)Automatic constraint selection

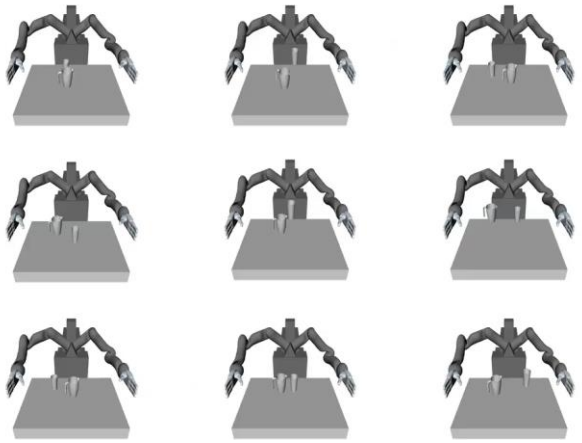
Execution using 2D-vision to localize objects



Automatic constraint optimization

Motions: Automatic model refinement II

- Results
  - 2 Robots, 8 Manipulation tasks
  - 34 CPU cores
  - Planning with/o dynamics simulatio
  - Generalization: 91%
  - Constraint reduction: 47%
  - Optimization time: up to 4h

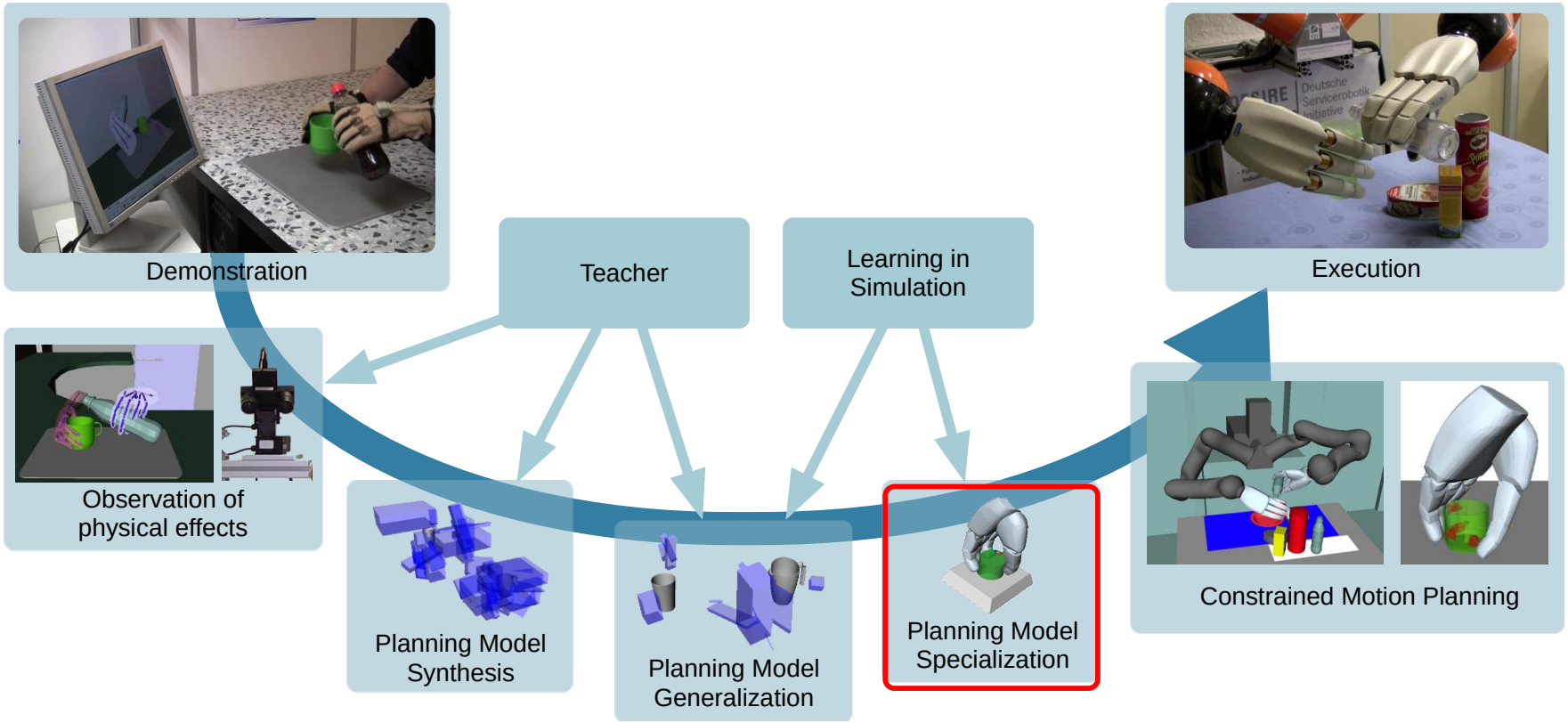


Parallelized Optimization („pour in“)

Task	Constraints (before)	Constraints (after)	Training data	Test data	Success
Flasche in Kühlschrank	5	2	2	10 (3)	100
Zweihändiges Einschenken	174	60	2	15 (3)	67
Flasche in Kiste	15	13	3	20 (4)	100
Taste auf Tastatur drücken	362	28	3	15 (3)	100
Tasse auf Untertasse	26	15	4	24 (6)	88
Löffel anheben	140	55	1	100 (2)	84
Zudrehen einer Flasche	16	8	1	20 (4)	90
Stuhl anheben	12	6	1	10 (3)	100

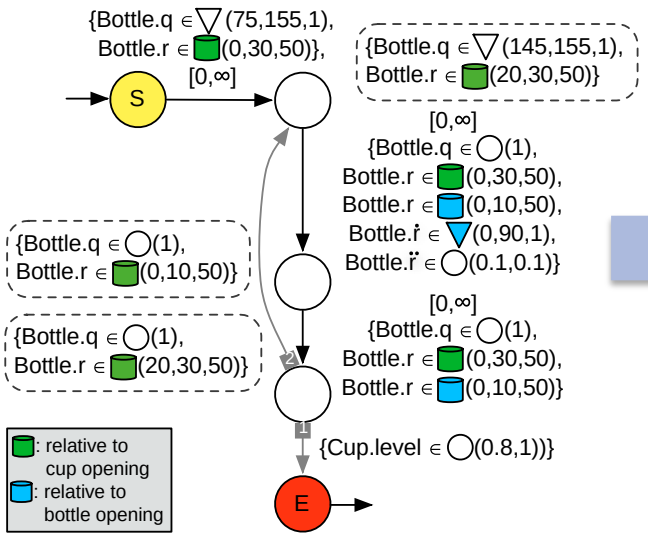
Overview – Programming by Demonstration

- Here: how to specialize learned planning models to the scene

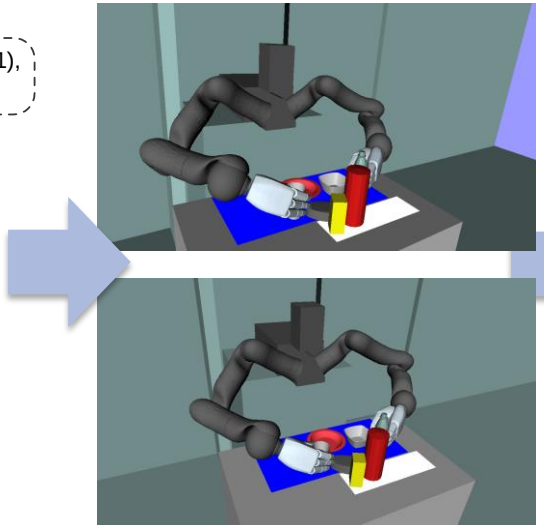


Strategy Learning

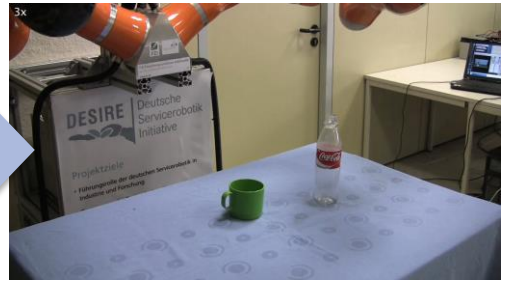
- Qualitative description based on domain constraints
- Generalisation and mapping
  → flexible representation of the search space for low-level planning
  → search space automatically adapts to obstacles and objects
- Full exploitation of robot capabilities



Strategy graph
"pouring"



Constrained
Motion Planning



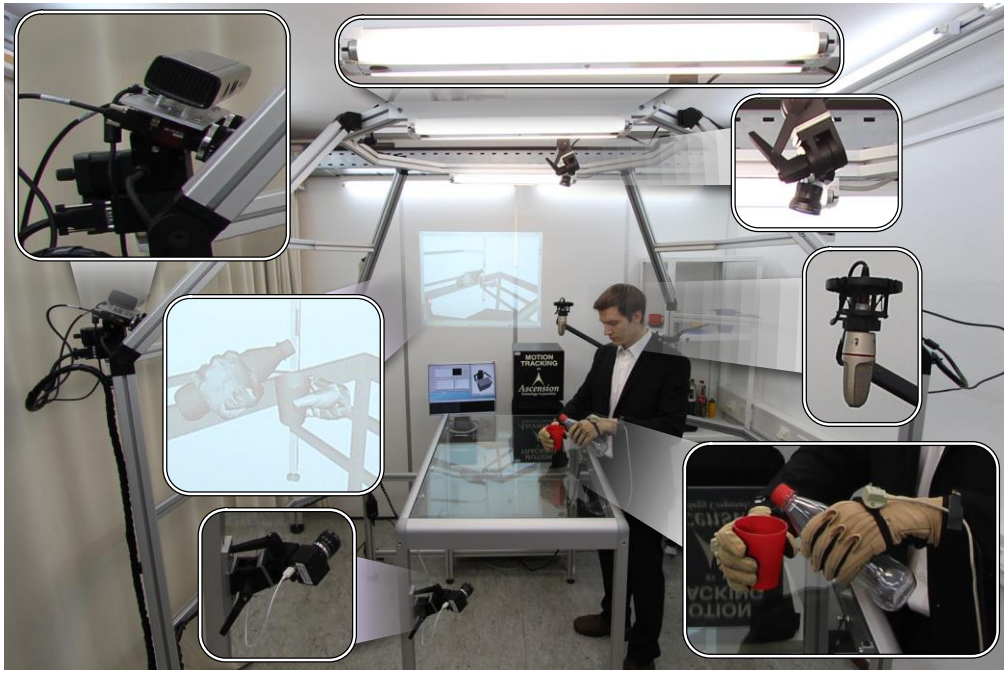
Execution

Motions: Manipulation strategies

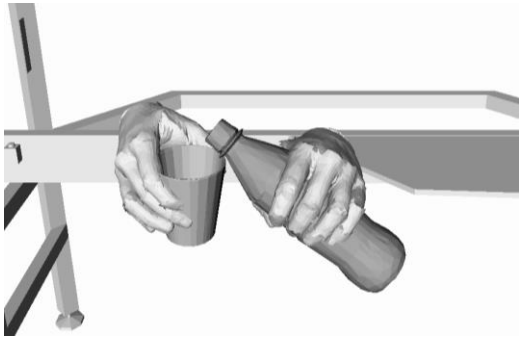
- Goal: Flexible manipulation strategies
  - Motion planning to consider high variance in objects, obstacles and environments
  - Scalability to different robot systems
  - Short planning times
- Planning model based on constraints: “Strategy graph”
  - Nodes = Subgoals, e.g. “where to place the bottle in the fridge door”
  - Edges = Transitions, e.g. “hold bottle upright during the motion”
- Constraints: qualitative and quantitative representation of relevant task aspects

Motions: Programming method

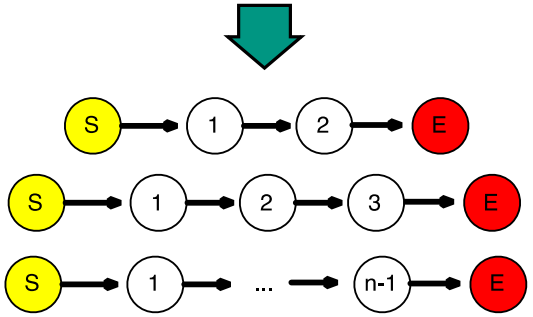
- Natural, human demonstrations of manipulation task
- Dedicated sensor systems to observe human
  - Datagloves, magnetic field trackers, 2D-, 3D-vision, tactile gloves
- Supported by 3D-visualization and -simulation



Natürliche Demonstrationen in Sensorumgebung



3D-Visualisierung und Kontaktberechnung

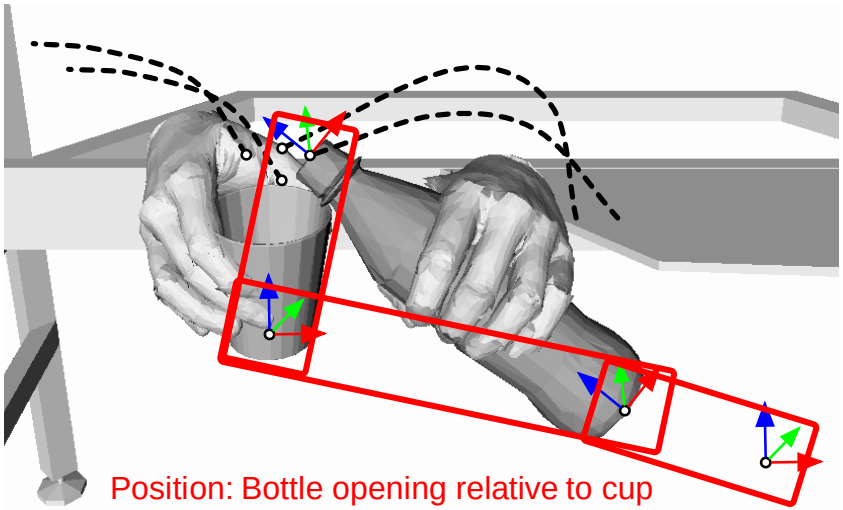


Struktur (Segmentierung)

Institut für Anthropomatik und Robotik
Prof. Dr.-Ing. R. Dillmann
Kognitive Systeme

Motions: Preliminary Model generation

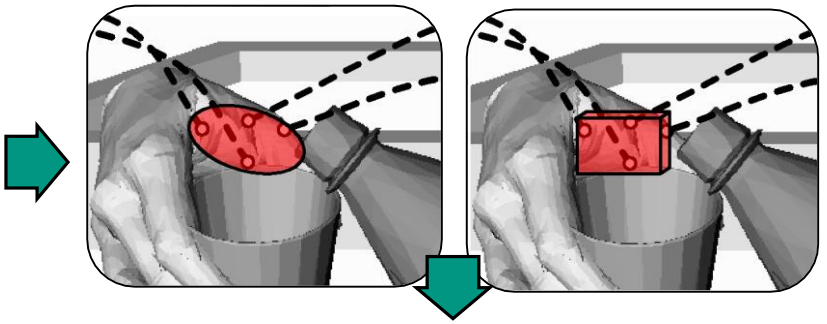
- Automatic generation of constraints
  - Basis: Combinations of coordinate systems (fingers, hands, detected objects)
    - Examples: Bottle opening relative to cup, bottle bottom relative to table
  - Optimization: automatic calculation of geometric primitives
  - Selection of geometric primitive with minimal surface area



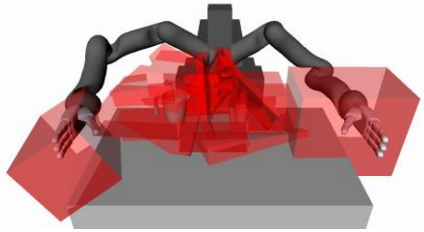
Position: Bottle opening relative to cup
Position: Bottle bottom relative to cup
Position: Bottle bottom relative to table

Multiple human demonstrations

Minimum



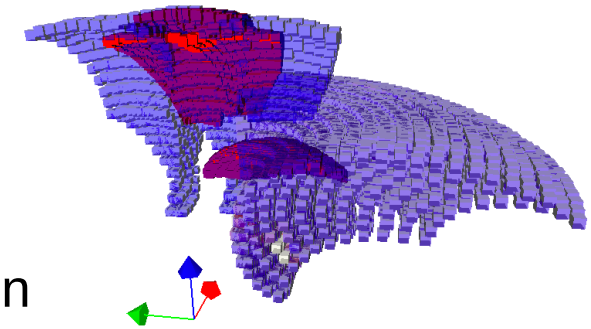
Optimization: generate parameters of geometic primitives



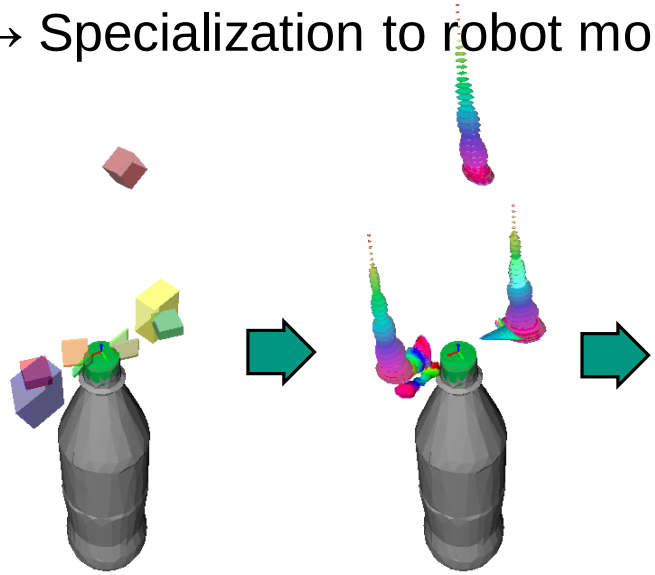
Constraints for combinations of coordinate systems

Motions: Correspondence Problem

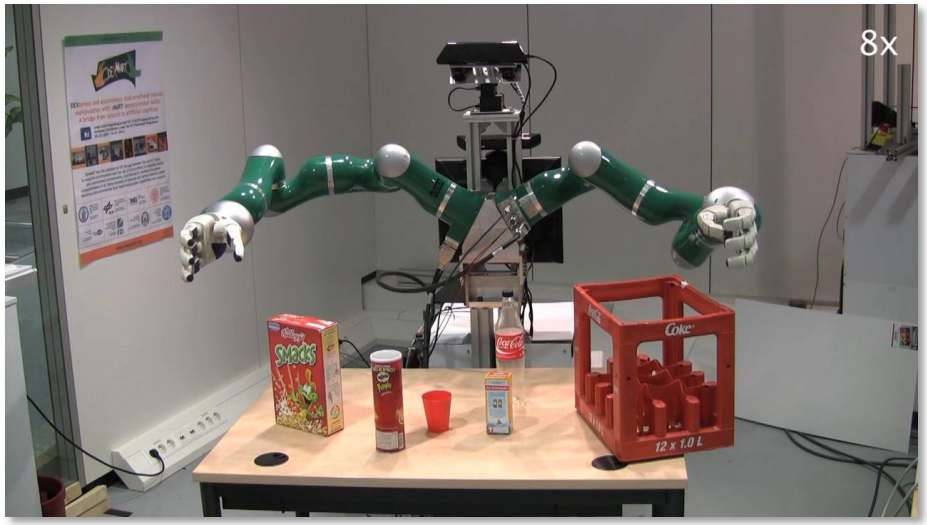
- Correspondence Problem: „How to consider differences in morphology of human and robot?“ [Skoglund09]
- Relaxation of finger and hand constraints based on workspace differences
- Constraint refinement using cycle-time reduction
  → Specialization to robot morphology



Workspace analysis



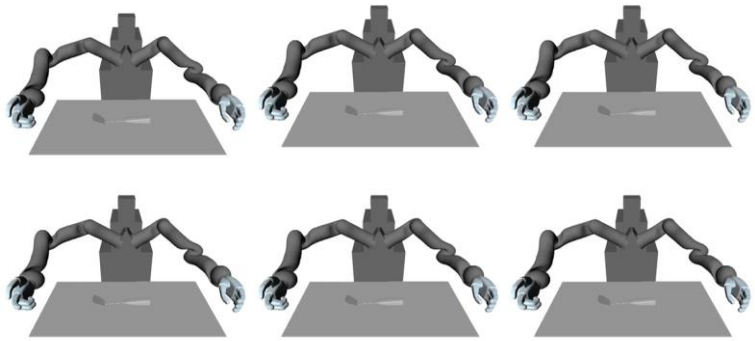
Original constraints Optimized constraints



Reduced time to generate complex motions

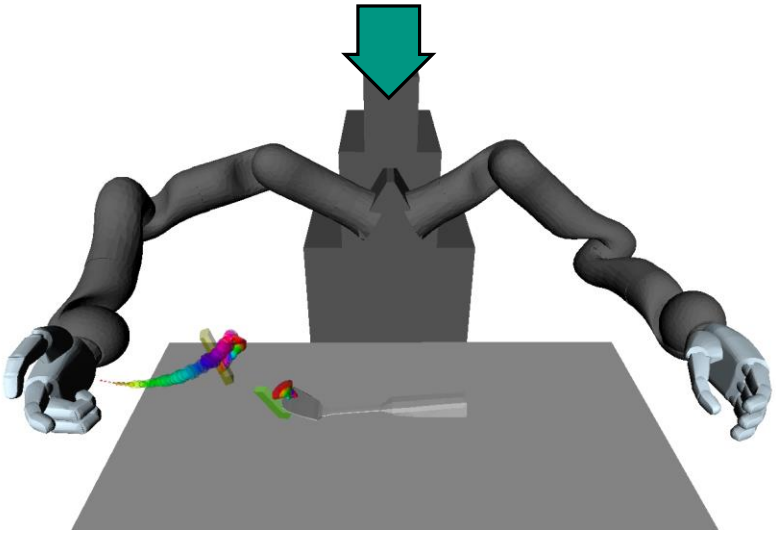
Motions: Example

Human demonstrations



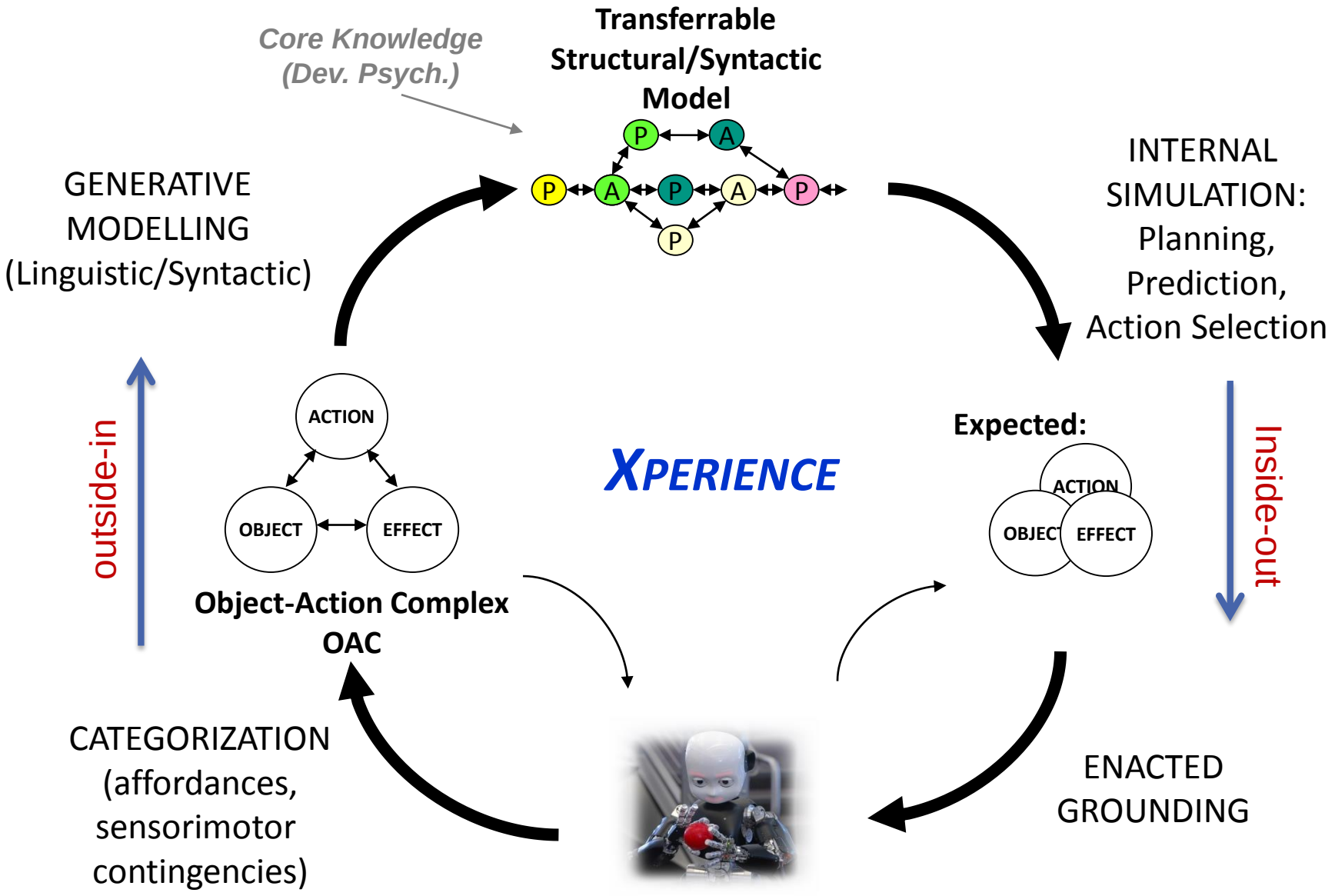
(Semi-)Automatic constraint selection

Execution using 2D-vision to localize objects



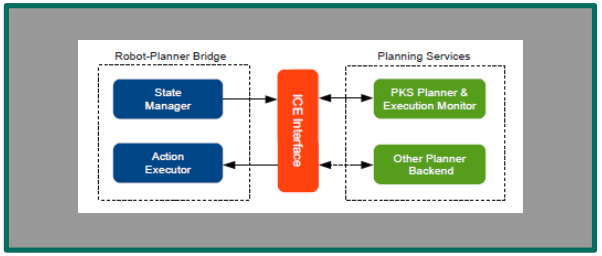
Automatic constraint optimization

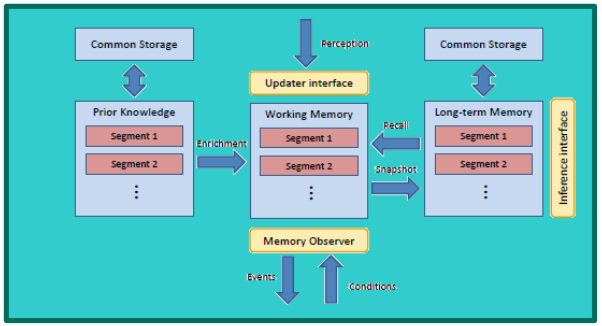
Generativer Lernzyklus

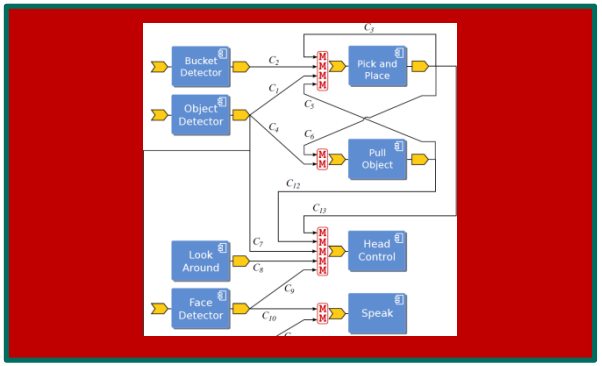


Kognitive Systemarchitektur

- The Xperience cognitive system architecture
  - Planning level
  - Mid-level (memory)
  - Behavior level
- Bootstrapping examples on all levels utilize grammatical correlations
- Benchmarking

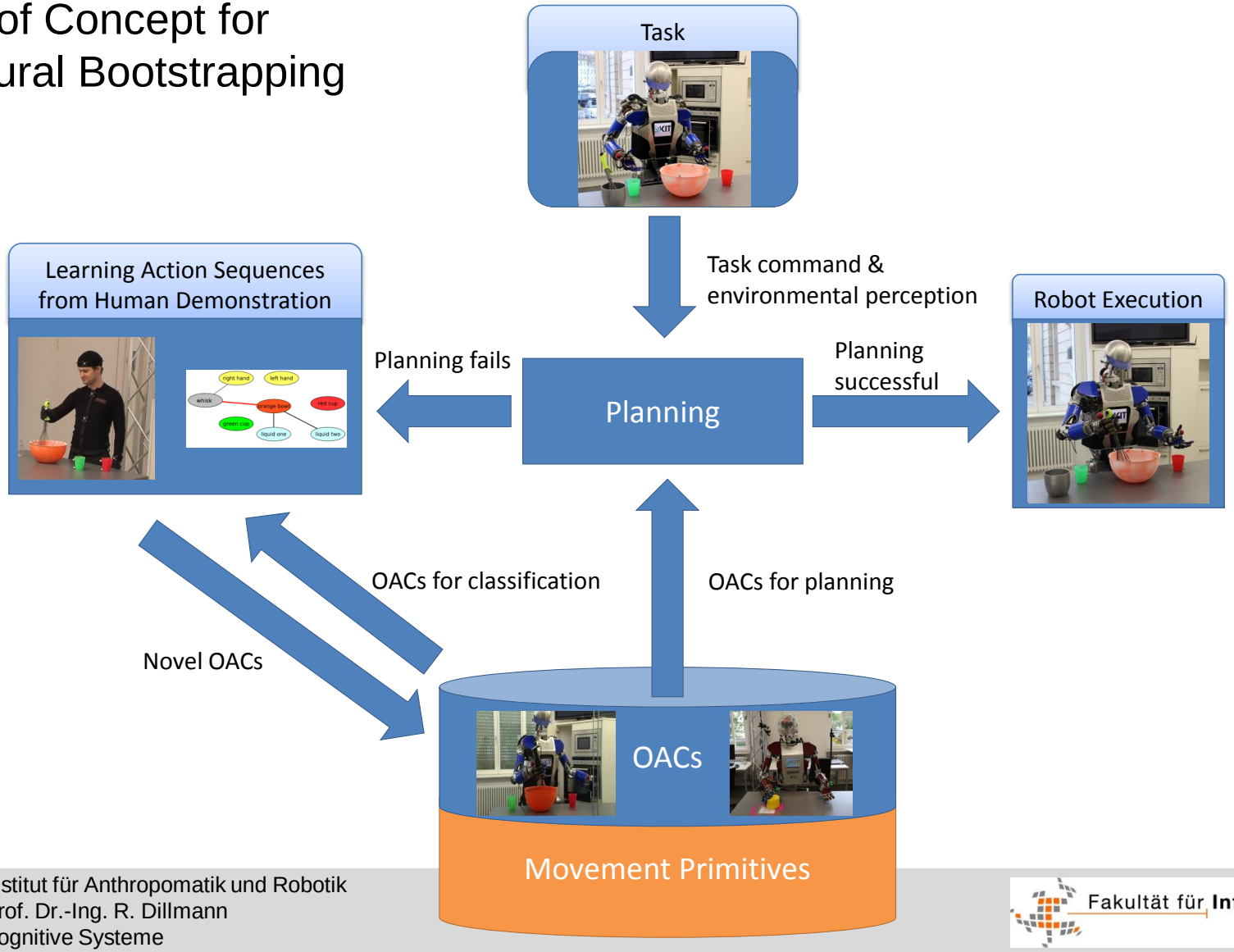






Struktural Bootstrapping von OACs

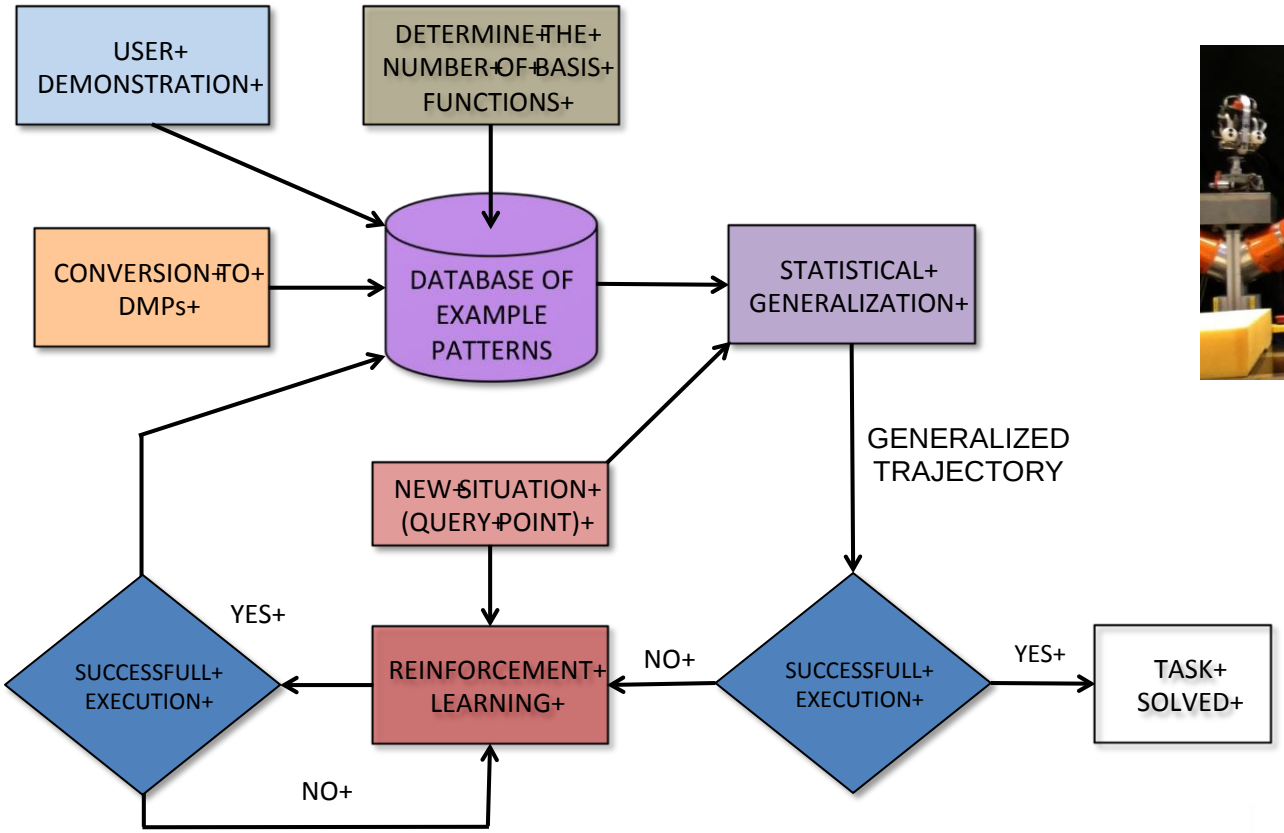
Proof of Concept for
Structural Bootstrapping

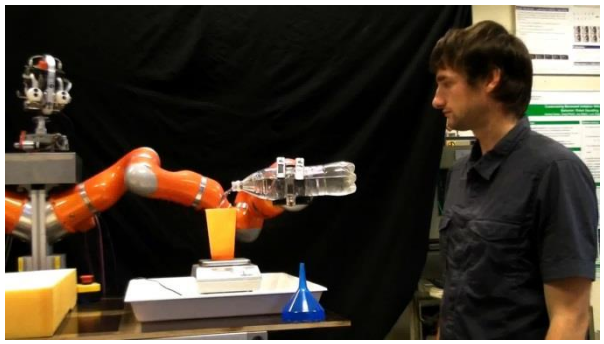


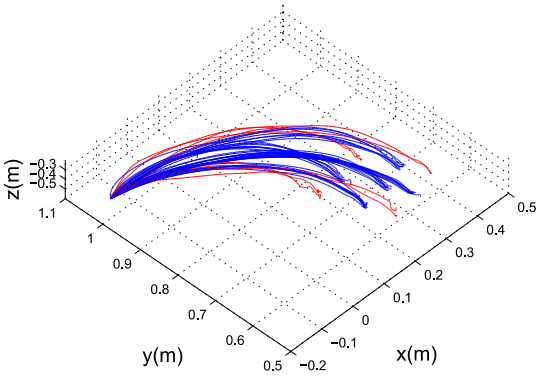
Institut für Anthropomatik und Robotik
Prof. Dr.-Ing. R. Dillmann
Kognitive Systeme

Autonomes Lernen von Skills

Autonomous skill learning by imitation and reinforcement learning



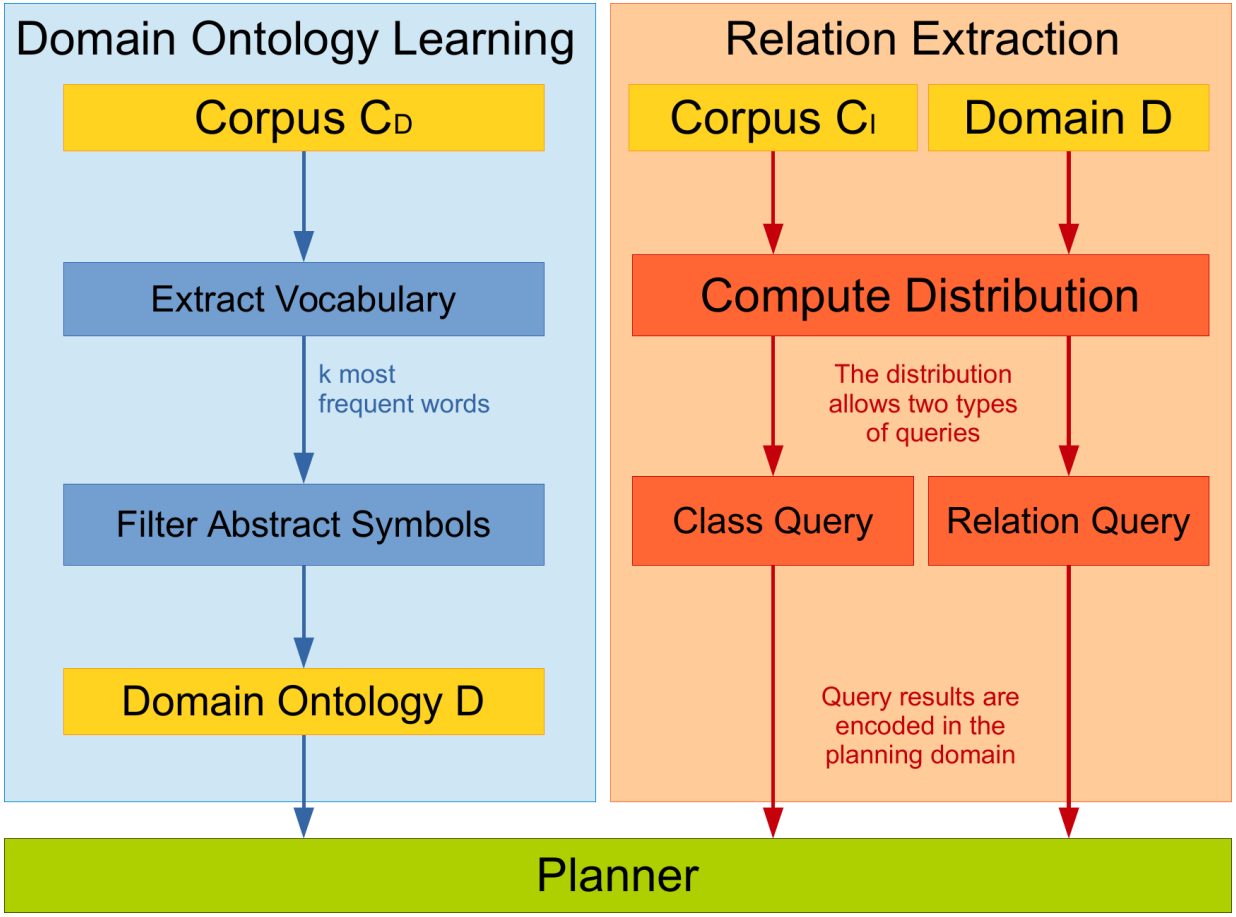




Institut für Anthropomatik und Robotik
Prof. Dr.-Ing. R. Dillmann
Kognitive Systeme

Lernen von Onthologien und Relationen

Learning common sense knowledge for planning



Institut für Anthropomatik und Robotik
Prof. Dr.-Ing. R. Dillmann
Kognitive Systeme