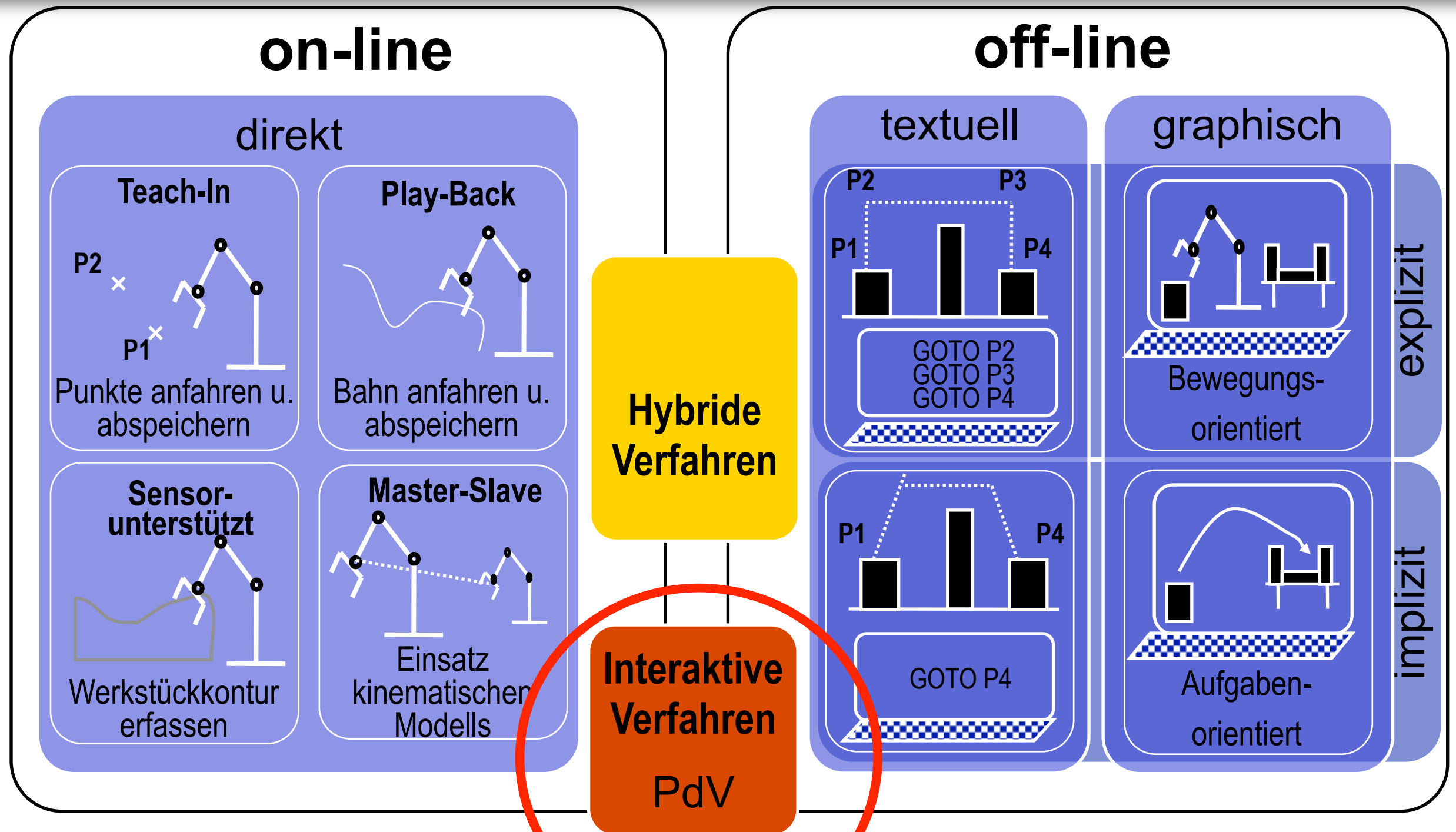


III

Interaktive Programmierung von Robotern

Roboterprogrammierverfahren



Übersicht

Interaktives Roboterprogrammieren I: Grundlagen (diese VL)

- Grundlagen
- Beispielsysteme: Calinon, Schaal, IPoR II
- Klassifikation von PdV-Verfahren

Interaktives Roboterprogrammieren II: Methoden (nächste VL)

- Bahnplanung
- Grifftaxonomie
- Griffplanung

Roboterprogrammierung: Neue Anforderungen in der Produktion

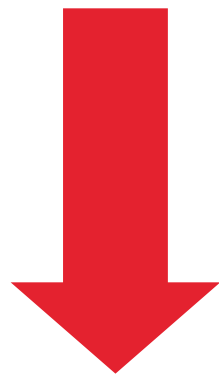
Klein- & Kleinstserienfertigung

Unikatfertigung (z.B. Prototyp)

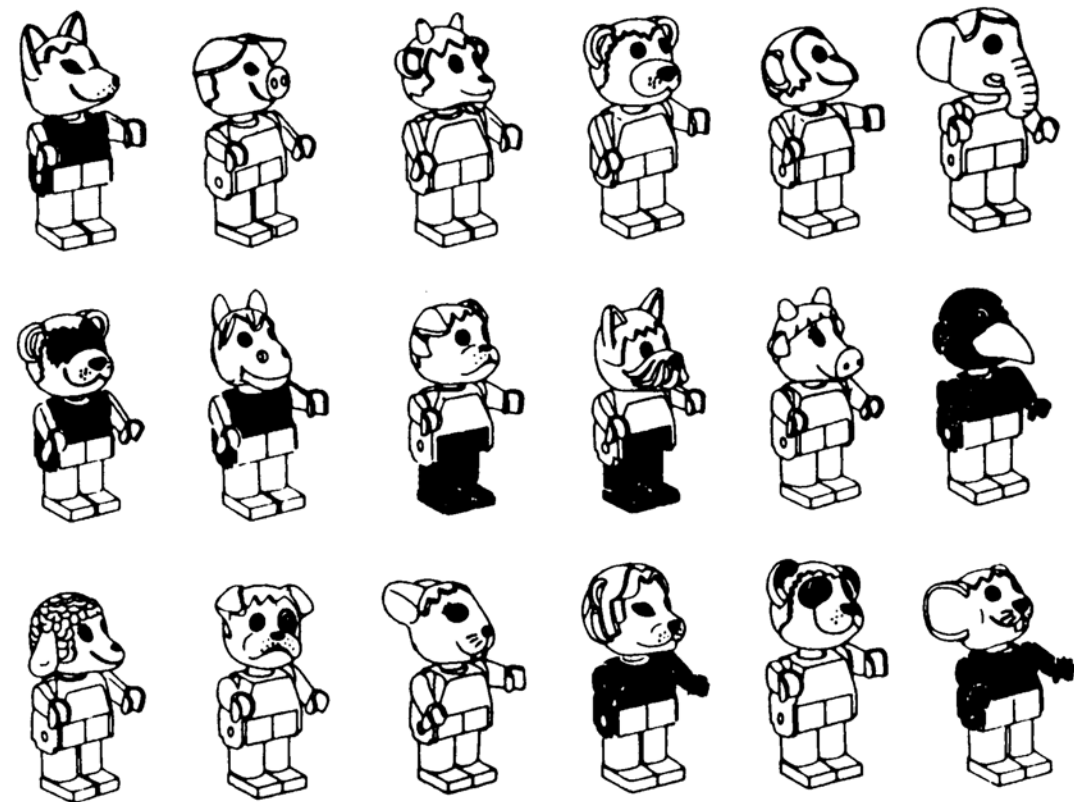
Produkte mit:

vielen Ausstattungsvarianten

hoher Rekonfigurierbarkeit



Flexible Fertigung



Roboterprogrammierung: Neue Anforderungen im Servicebereich

Handel:

- Kommissionierung und Palettierung von Waren
- Bestücken von Regalen

Pflege:

- Unterstützung von Rehabilitationsmaßnahmen durch Roboter
- Rollstuhl mit Manipulationshilfe

Handwerk:

- Handhabungen in Schreinereien und Schlossereien



Roboterprogrammierung: Anforderungen in der humanoiden Servicerobotik

- Manipulation beliebiger Objekte
- Selbstständiges Lösen komplexer Aufgaben
- Einsatz im menschlichen Umfeld



komplexe
Umgebung!

sehr viele
Freiheitsgrade!



➡ Wie Handlungswissen erzeugen?

Interaktive Programmierung: Grundidee

1. Mensch ist Domänenexperte (Manipulation)
2. Explizite Demonstrationen der Manipulationsaufgabe
3. Sensorielle Erfassung der Demonstrationen
4. Erzeugung der internen Repräsentation des Roboterprogramms
5. Abbildung auf das Robotersystem
6. Ausführung

Interaktive Programmierung: Anforderungen

1. Intuitive Interaktionsformen
2. Transparenz der Prozesse im Programmiersystem
3. Abgleich von Systemhypothesen mit der Benutzerintention
4. Flexibilisierung und Optimierung von Programmen
5. Wiederverwendung von Teillösungen

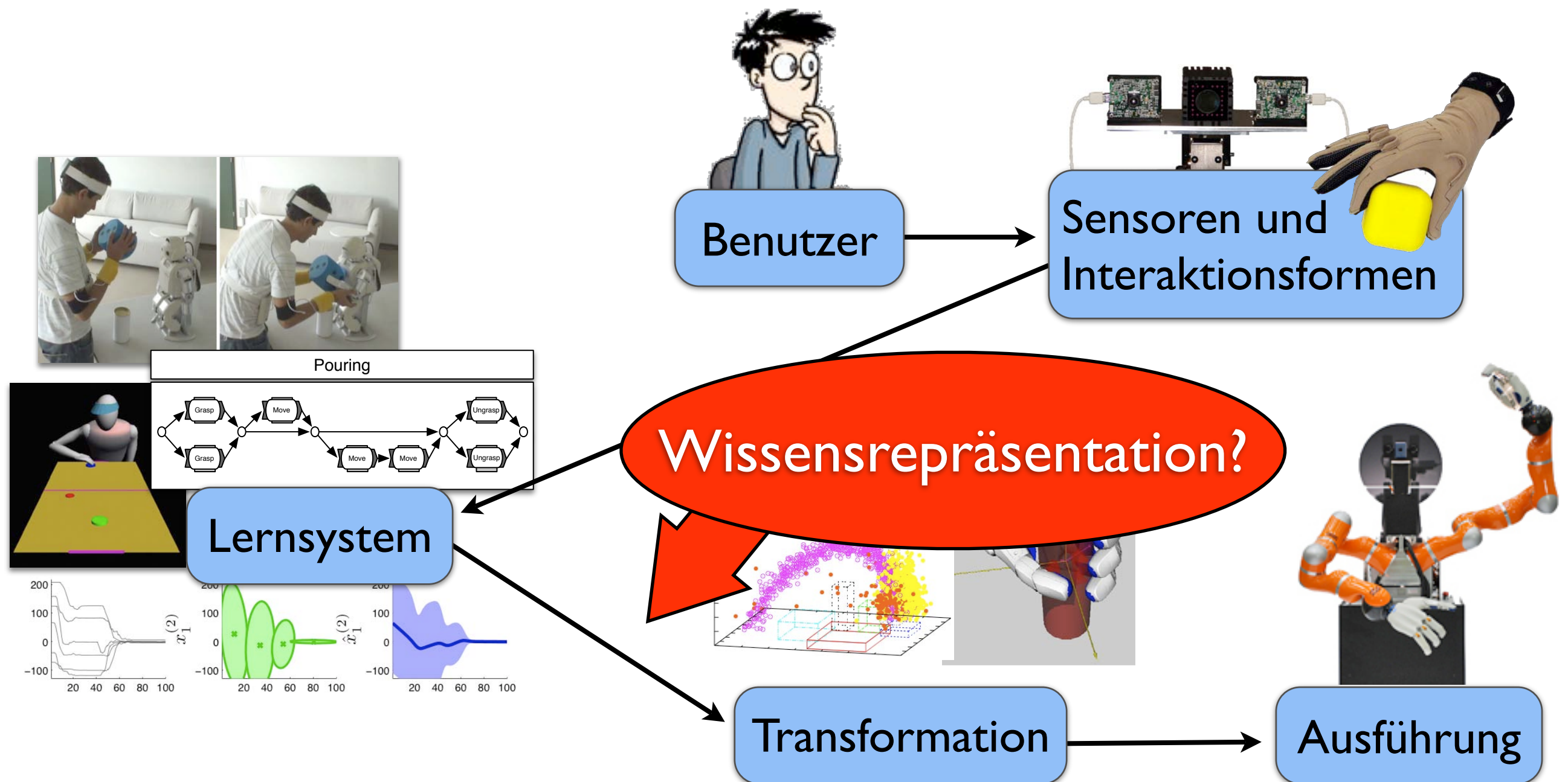


 ***Interaktive Programmierung***

Interaktive Programmierung: Randbedingungen

1. Weitgehend automatisierte Programmgenerierung
2. Beschränkung der Benutzerinteraktion auf das Nötigste
3. Maximierung des Informationsgewinns bei Benutzerinteraktion
4. Möglichst benutzerfreundliche Mensch-Maschine Interaktion

Interaktive Programmierung: Komponenten



Manipulatorabhängige vs. -unabhängige Repräsentation

Manipulatorabhängige Repräsentation durch Angabe von

- Aktionssequenz oder
- Gelenkwinkel-, Kraft und Momenttrajektorien

subsymbolisch

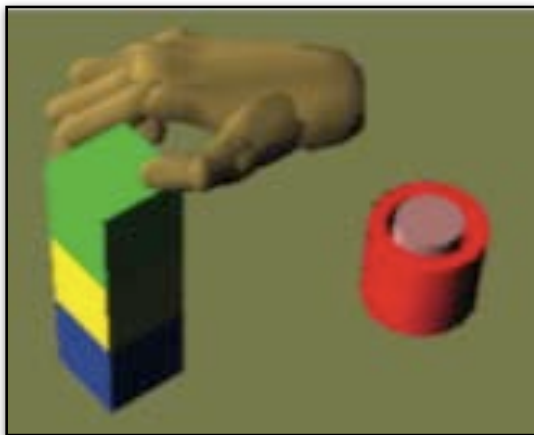
Manipulatorunabhängige Repräsentation durch Sequenzen von Elementaroperatoren

- Elementaroperatoren sind Regelungen mit Start-, End- und Fehlerkriterien
- Implementierung der Elementaroperatoren ist manipulatorunabhängig
- Effekte in der Umwelt sind manipulatorunabhängig

symbolisch

Interaktionsformen

- Physische Demonstration
- Graphische Demonstration
- Ikonische Demonstration
- Kommentierung



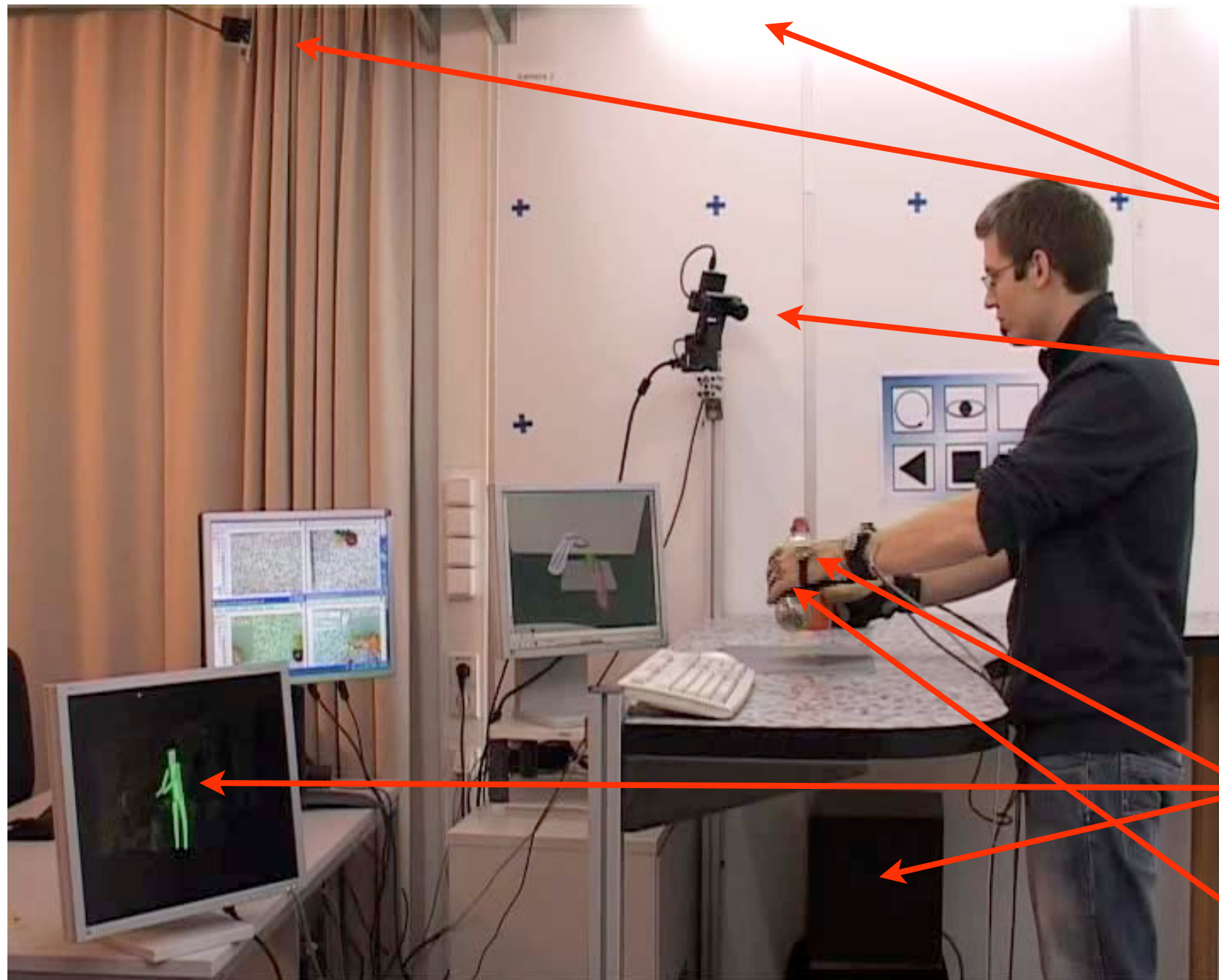
Interaktionsformen



Sensoren (I)

- Bildgebende Sensoren
- Magnetfeldbasierte Positionssensoren
- Datenhandschuhe & -anzüge
- Exoskelette
- Interne Robotersensoren

Sensoren (II)



fixierte
Deckenkamera

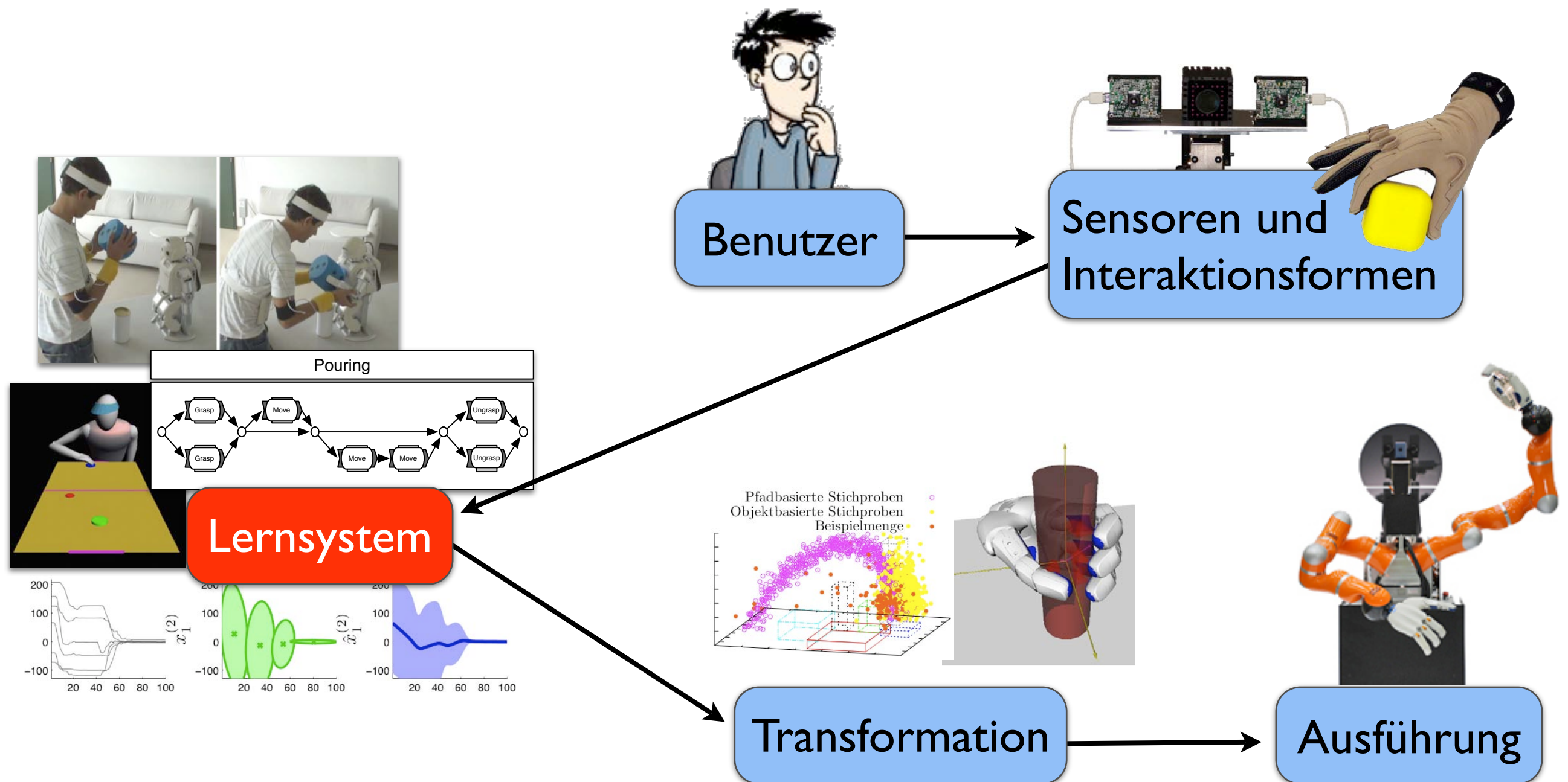
aktiver Kamerakopf

Taktile Sensoren im
Handbereich

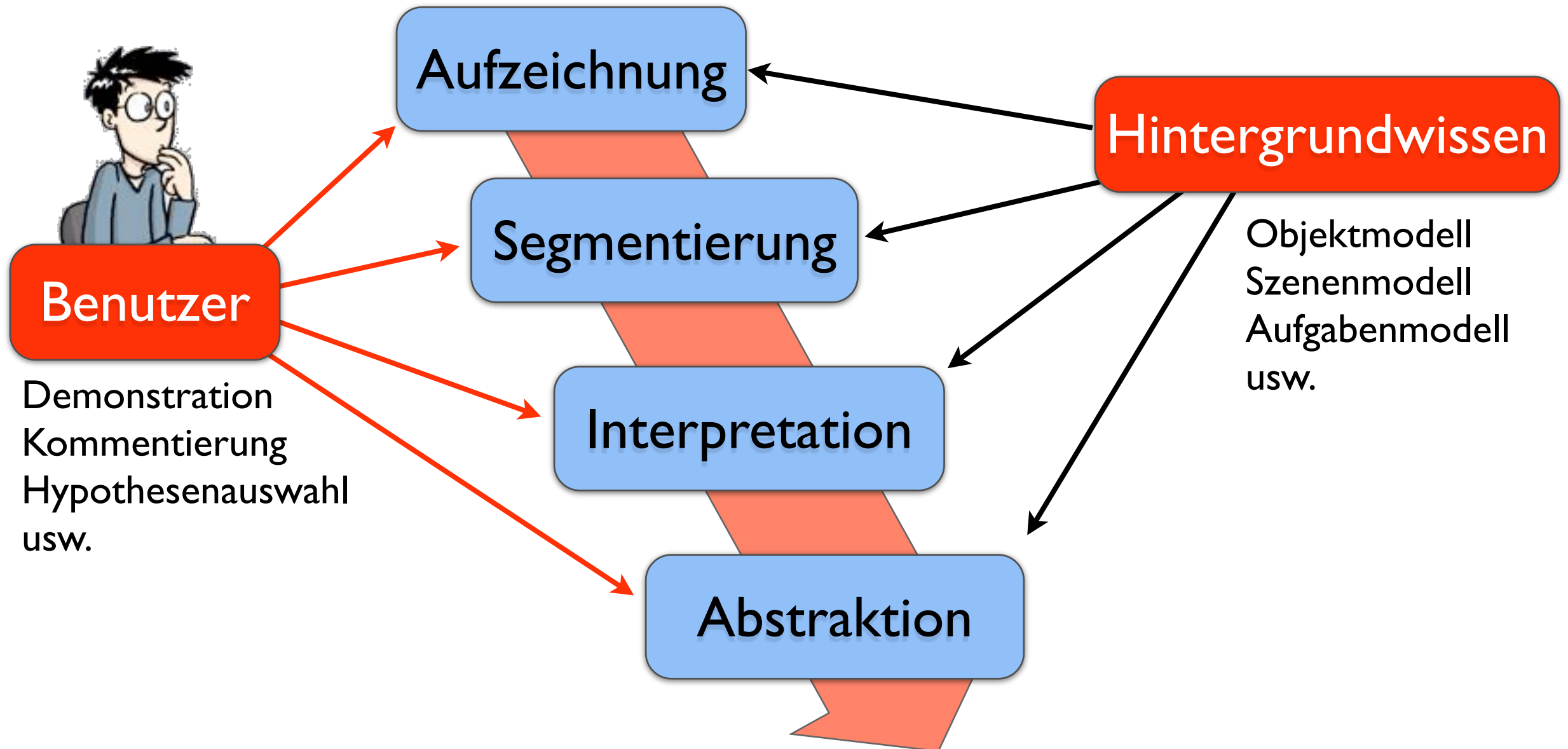
Trackingsystem

Datenhandschuh

Interaktive Programmierung: Komponenten

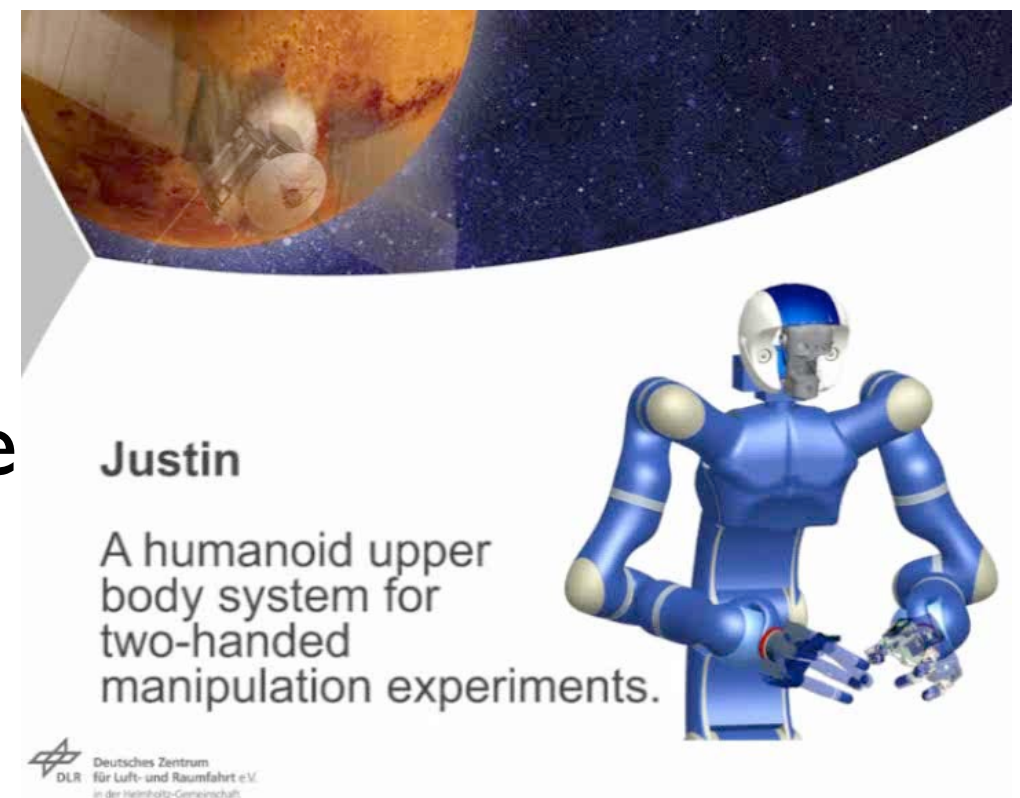


Lernsystem: Verarbeitungsschritte



Transformation

- Abbildung des generalisierten Handlungswissens auf ein Robotersystem in der Ausführungsumgebung
- Unterschiede in Demonstrations- und Ausführungsumgebung
 - Kinematik (insb. Arbeitsraum), Dynamik, Geometrie
 - Szene: Objektanzahl, -typen und -anordnung
- Berücksichtigung der Unterschiede
 - Planungsmethoden: Griff- und Bewegungsplanung
 - Details in Vorlesung: PdV II



Beispiele

Probabilistisch:

- Calinon / Billard

Dynamisch:

- Pastor / Schaal

Sub-/Symbolisch:

- IPOR II

Probabilistisch: Calinon, Billard (1/6)

Calinon07:

„What is the teacher's role in robot programming by demonstration?“

„On learning, representing and generalizing a task in a humanoid robot“

Lernen von Skills, z.B. Schachfigur bewegen

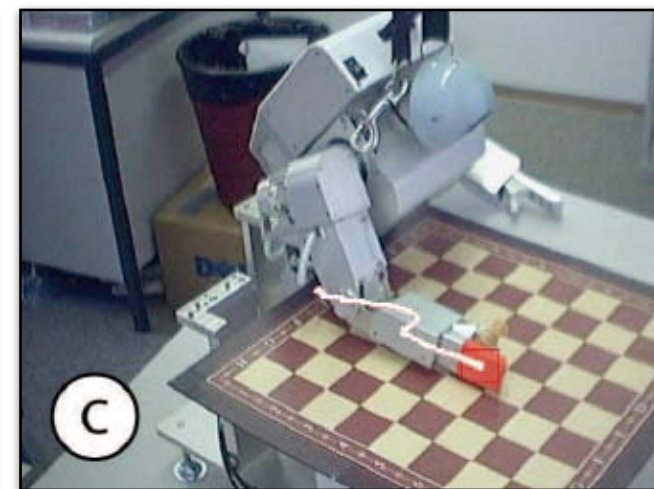
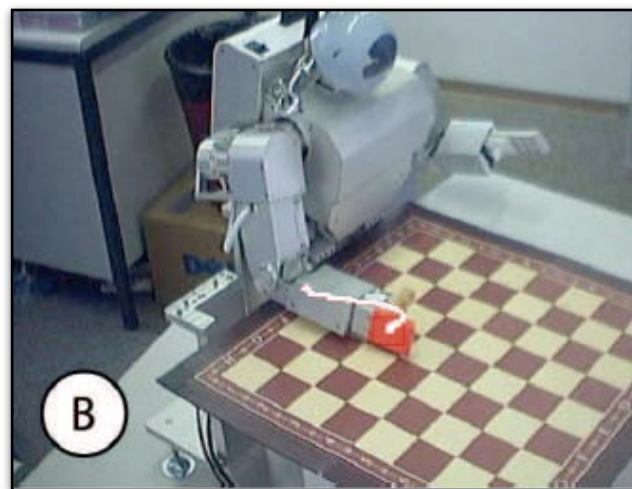
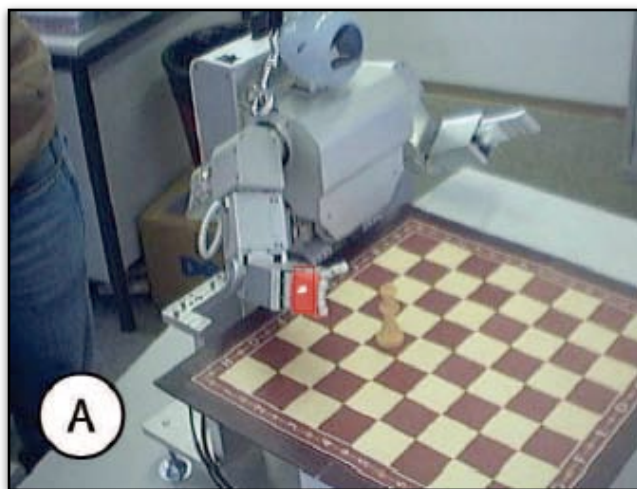
- Aktive, physische Demonstration am Roboter
→ kein Korrespondenzproblem
- Repräsentation durch Gaussian Mixture Models (GMM) → subsymbolisch
- Direkte Ausführung



Probabilistisch: Calinon, Billard (2/6)

Lerndaten:

- (θ, x, y, h) : n Demonstrationen mit je T Trajektoriepunkten
- θ : Gelenkwinkel des Roboters + Zeitstempel
- x : Kartesische Position der Hände + Zeitstempel
- y : Distanzvektor der Hände zur Startposition des Objekts + Zeitstempel
- h : Binärer Zustand des Greifers (offen, geschlossen) + Zeitstempel



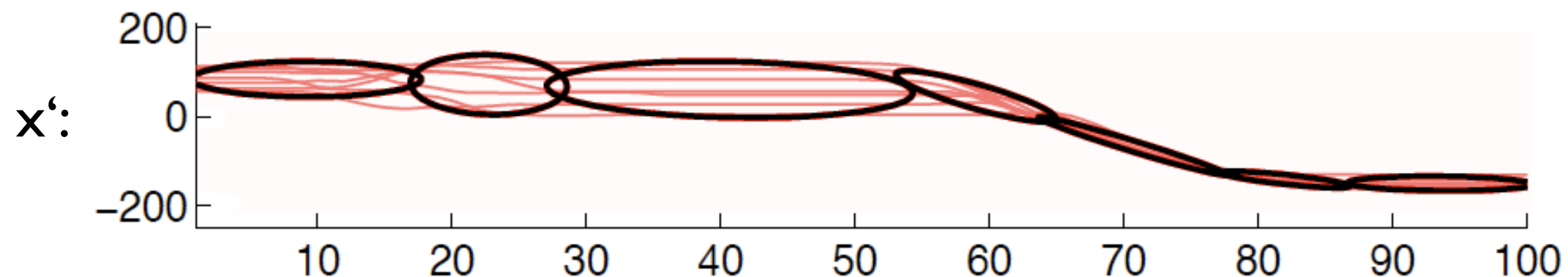
Probabilistisch: Calinon, Billard (3/6)

Lernverfahren:

- Dimensionsreduktion und zeitliche Angleichung: $(\theta, x, y, h) \rightarrow (\theta', x', y', h')$
- Lernen eines GMMs für alle Komponenten, z.B. x' (4-dim.) mit Dichte $p(x')$:

$$p(x') = \sum_{i=1}^k p(i) p(x'|i) = \sum_{i=1}^k \pi_i \mathcal{N}(x'; \mu_i, \Sigma_i)$$

- k = Anzahl der Normalverteilungen, π = Gewichtung, $\mathcal{N}$ = Normalverteilung
- Bestimmung von k : Bayes'sches Informationskriterium und EM-Algorithmus



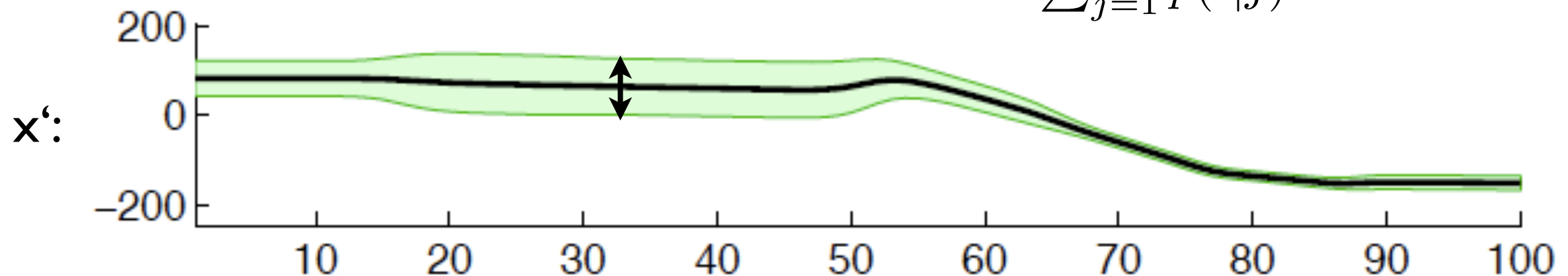
Probabilistisch: Calinon, Billard (4/6)

Repräsentation:

- Probabilistische Darstellung der Trajektorien $\mathbf{x}'(t)$: $t \mapsto \mathcal{N}(\mu_{x'}(t), \Sigma_{x'}(t))$
- Berechnung durch Gaussian Mixture Regression ~ Gewichtung der bedingten Wahrscheinlichkeiten $p(\mathbf{x}', i | t)$

$$\mu_{x'}(t) = \sum_{i=1}^k \beta_i(t) \mu_{i,x'|t} \quad \text{und} \quad \Sigma_{x'}(t) = \sum_{i=1}^k \beta_i(t)^2 \Sigma_{i,x'|t}$$

mit $p(\mathbf{x}', i | t) = \mathcal{N}(\mu_{i,x'|t}, \Sigma_{i,x'|t})$ und $\beta_i(t) = \frac{p(t|i)}{\sum_{j=1}^k p(t|j)}$



Probabilistisch: Calinon, Billard (5/6)

Ausführung:

- Definition eines quadratischen Ähnlichkeitsmaßes: „metric of imitation“
- Gewichtung der Abweichung von den Mittelwerten in t , z.B. $\mu_{x'}(t)$
- Wahl der Gewichtsmatrix: $(\Sigma_{x'}(t))^{-1}$
- Bestimmung des Nachfolgerzustands $\theta(t+1)$ bzw. der Transition $\theta(t) \rightarrow \theta(t+1)$, die das Ähnlichkeitsmaß minimiert
- Jacobi-Matrix zur Kombination von kartesischen und Gelenkwinkel-einschränkungen



Probabilistisch: Calinon, Billard (6/6)

Vorteile:

- schnelles Verfa
- automatische A

Nachteile:

- relevante Mer
- geringe Gener
→ keine zielge
- keine Validierung



en

Startposition

ionen

Wissensrepräsentation: Beispiele

Probabilistisch:

- Calinon / Billard

Dynamisch:

- Pastor / Schaal

Sub-/Symbolisch:

- IPOR II

Dynamisch: Pastor, Schaal (1/4)

Pastor09:

„Learning and generalization of motor skills by learning from demonstration“

- Lernen von Skills, z.B. Tennisschwung
- Aktive, physische Demonstration am Roboter → kein Korrespondenzproblem
- Repräsentation durch Dynamic Movement Primitives (Differentialgleichungen)
- Direkte Ausführung



Dynamisch: Pastor, Schaal (2/4)

Repräsentation:

- Implizite Darstellung durch Menge von Differentialgleichungen:

$$\begin{aligned}\tau \dot{v} &= K(g - x) - Dv + (g - x_0)f \\ \tau \dot{x} &= v\end{aligned}$$

- x = Position, v = Geschwindigkeit, K = Federkonstante, D = Dämpfung, g = Ziel, x_0 = Start, τ = zeitliche Skalierung
- f = nicht-lineare Funktion, die die Demonstrationsmenge approximiert:

$$f(s) = \frac{\sum_{i=1}^n w_i \psi_i(s) s}{\sum_{i=1}^n \psi_i(s)} \quad \text{mit } \tau \dot{s} = -\alpha s$$

- Vorteil: Gewichte hängen nicht von τ , x_0 und g ab
 - Änderungen von Start, Ziel und der zeitlichen Skalierung möglich
 - Generalisierung eingeschränkt möglich

Dynamisch: Pastor, Schaal (3/4)

Lernen:

- Berechnung von $v(t)$, $\dot{v}(t)$ für jede Demonstration $\mathbf{x}(t)$
- $s(t)$ wird durch Integration berechnet
- Der Wert $f(s)$ wird berechnet
- ψ_i sind nicht normalisierte Normalverteilungen („Gauss’sche Basisfunktionen“)
- Bestimmung der Parameter w_i durch lineare Regression

$$\tau \dot{v} = K(g - x) - Dv + (g - x_0)f$$

$$f(s) = \frac{\sum_{i=1}^n w_i \psi_i(s) s}{\sum_{i=1}^n \psi_i(s)}$$

Ausführung:

- Berechnung von $v(t)$, $\dot{v}(t)$ im aktuellen Zustand und Integration

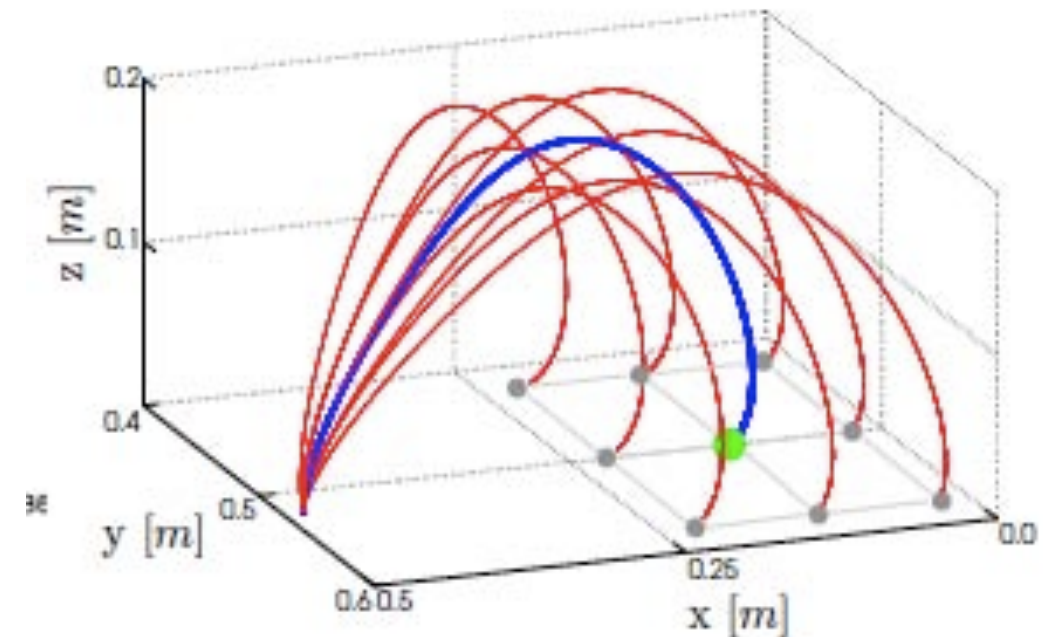
Dynamisch: Pastor, Schaal (4/4)

- Vorteile:

- schnelles Verfahren
- automatische Adaptierung an Start und Ziel
- lokale Hindernisvermeidung möglich

- Nachteile:

- relevante Merkmale manuell definiert
- geringe Generalisierung
- keine Validierung



Wissensrepräsentation: Beispiele

Probabilistisch:

- Calinon / Billard

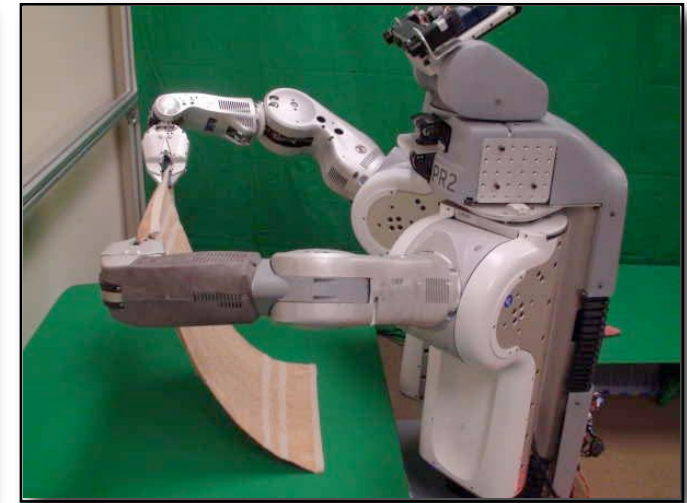
Dynamisch:

- Pastor / Schaal

Sub-/Symbolisch:

- IPOR II

Beispielsystem: *IPoR II*



Problembeschreibung:

- Robotersystem mit sehr vielen Freiheitsgraden (Kuka LBR / SAH/HIT: 40)
- Fingerfertige Manipulation fester Körper („rigid bodies“)
- Roboterarbeitsraum in realen Situationen stark eingeschränkt (z.B. Kollisionen)
- „Korrespondenzproblem“: Mensch und Roboter haben unterschiedliche Kinematik, d.h. keine direkte Abbildung menschlicher Bewegungen möglich

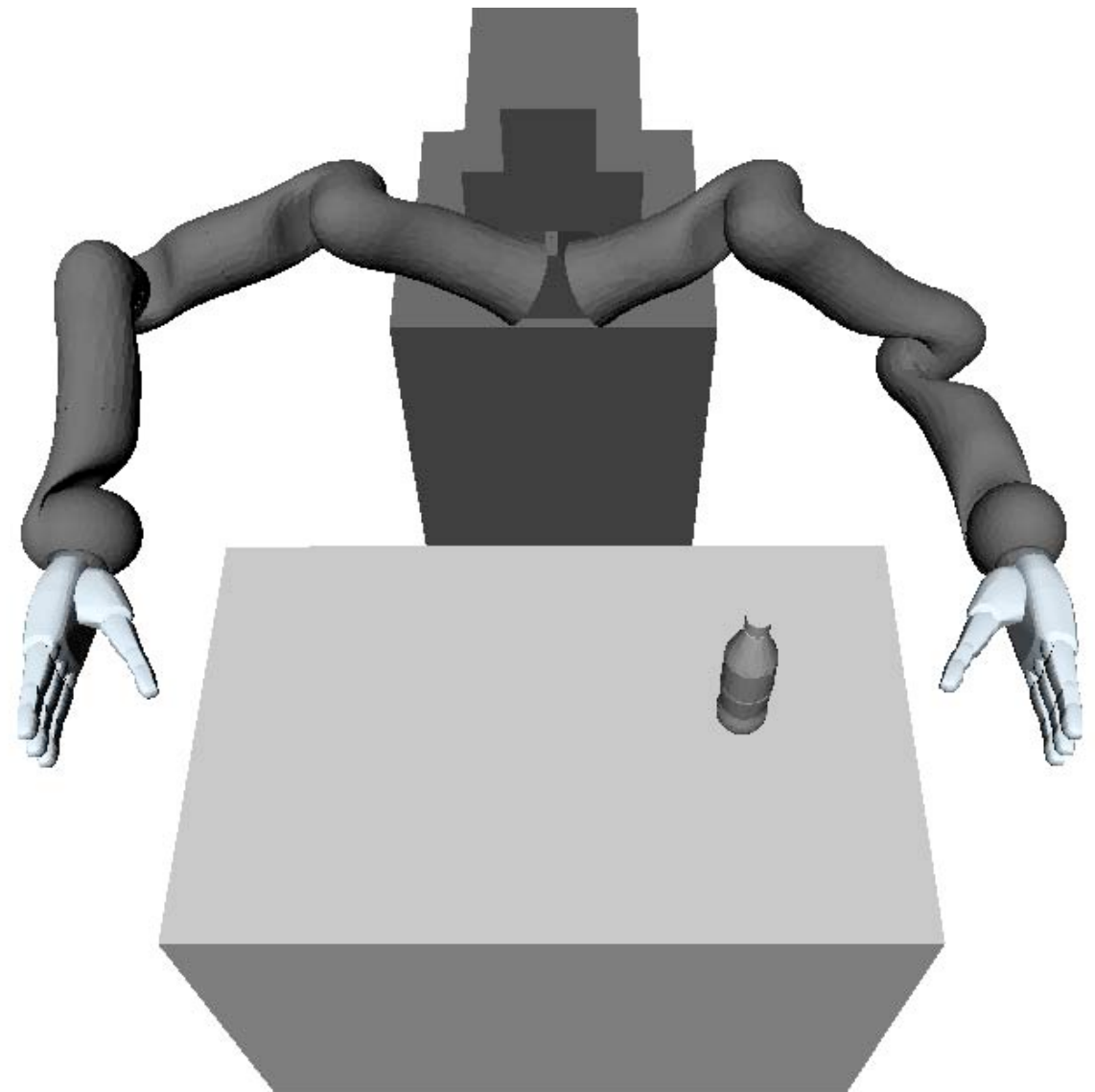
Lösungsansatz: *IPoR II*

Einsatz von Planungsmethoden:

- Repräsentation der Manipulationsaufgabe als Bahnplanungsproblem mit Einschränkungen
- Autonome Planung von Bewegungen, die das Ziel einer Manipulationsaufgabe erfüllen

Problem:

- Manuelle Definition des Planungsproblems ist komplex (z.B. ≥ 40 dofs)



➡ Lernen von Planungsproblemen aus der Beobachtung des Menschen

Repräsentation: Bahnplanungsproblem mit Einschränkungen

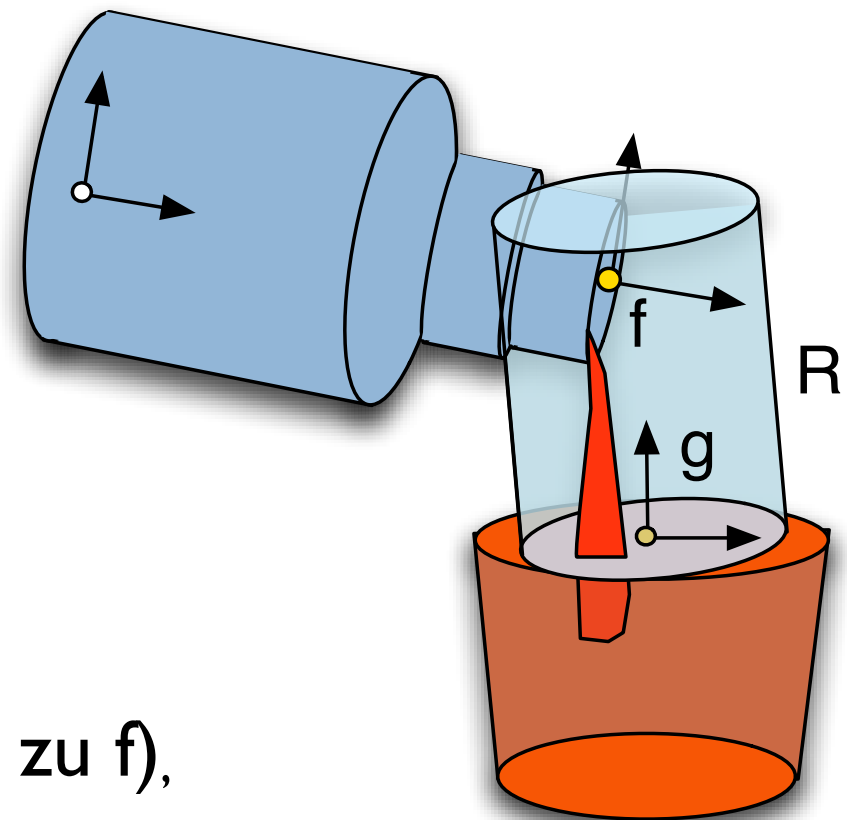
Grundlage: Bewegungseinschränkungen

- Beschränkung der Bewegung eines Koordinatensystems relativ zu einem zweiten Koordinatensystem (ähnlich „Task Frames“)
- 3 Typen: Positions-, Orientierungs- und Richtungseinschränkungen

Definition:

- Einschränkung (t, f, M, g, R) mit

Typ t ,
Koordinatensysteme f, g ,
homogene Transformationsmatrix M (relativ zu f),
Region R



Repräsentation: Bahnplanungsproblem mit Einschränkungen

Erinnerung: homogene Matrix (Robotik I)

- Rotationsmatrix R darstellbar als Quaternion:
(q_w q_x q_y q_z)
- Winkel der Rotation:
 $\alpha = 2 \cdot \arccos(q_w)$
- Achse der Rotation:
(p_x p_y p_z) = $1 / \sin \alpha \cdot (q_x$ q_y q_z)
- Skalierte Rotationsachse ist definiert als:
(r_x r_y r_z) = $\alpha \cdot (p_x$ p_y p_z)

$$M' = \begin{pmatrix} & R & & x \\ & & & y \\ & & & z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & x \\ r_{21} & r_{22} & r_{23} & y \\ r_{31} & r_{32} & r_{33} & z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Repräsentation: Bahnplanungsproblem mit Einschränkungen

Beispiele für M:

$$t = \text{Position} \quad \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{Position von } f$$

$$t = \text{Orientierung} \quad \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{Orientierung von } f$$

$$t = \text{Richtung} \quad \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{Richtung der z-Achse von } f$$

Wann ist die Einschränkung erfüllt?

- Transformation von M definiert in f relativ zu g :

$$\begin{aligned} M' &= {}^0H_g^{-1} \cdot {}^0H_f \cdot M \\ &= {}^gH_f \cdot M \end{aligned}$$

- Umwandlung von M' in 3d-Vektor m' :

t = Position, Richtung:

$$m' = (x \ y \ z)$$

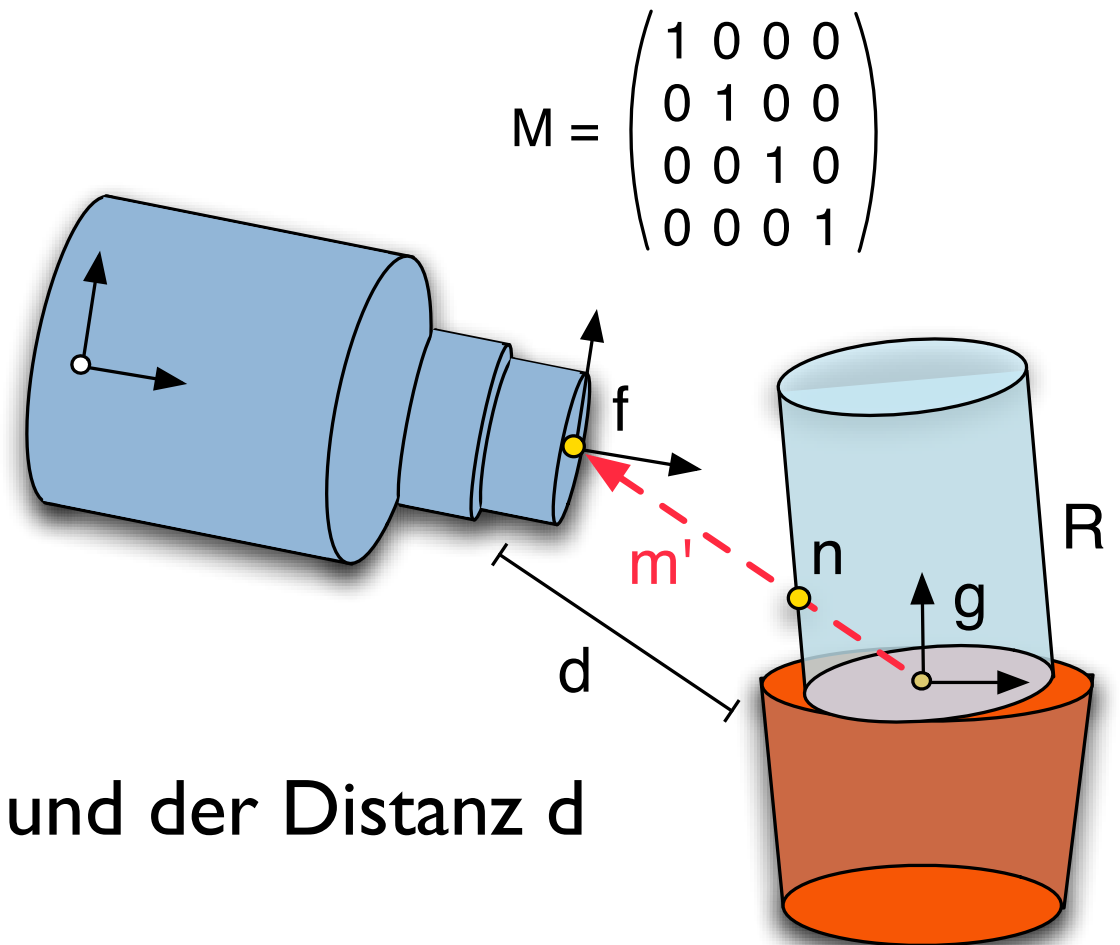
t = Orientierung:

$$m' = (r_x \ r_y \ r_z)$$

- Bestimmung des nächsten Punkts n in R und der Distanz d

$$d = |m' - n|$$

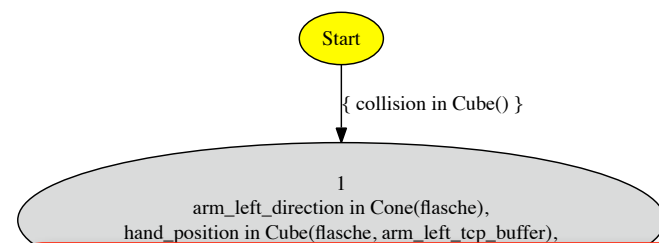
- Erfüllt, wenn $d < \varepsilon$



Repräsentation des Planungsproblems

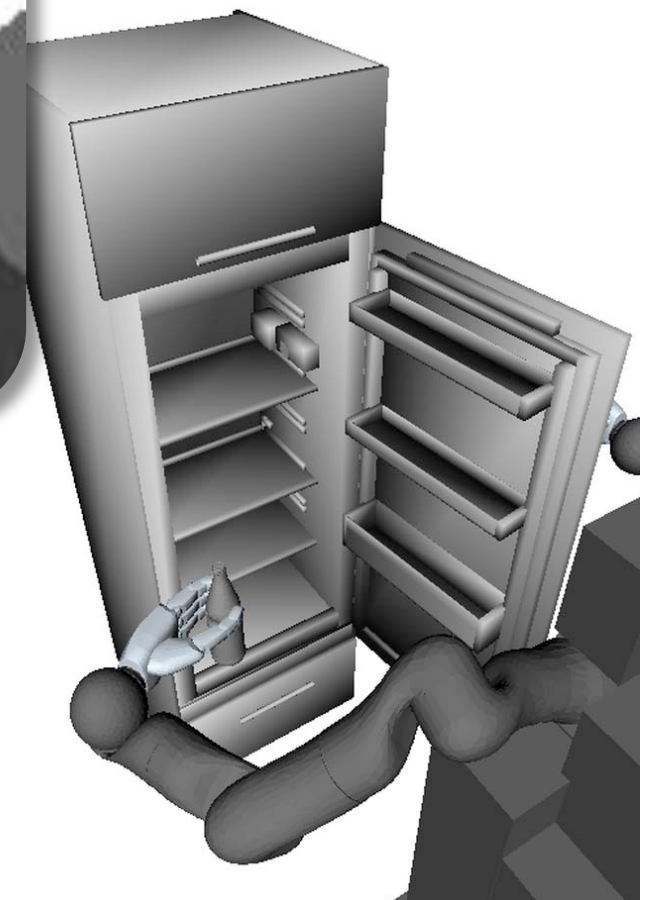
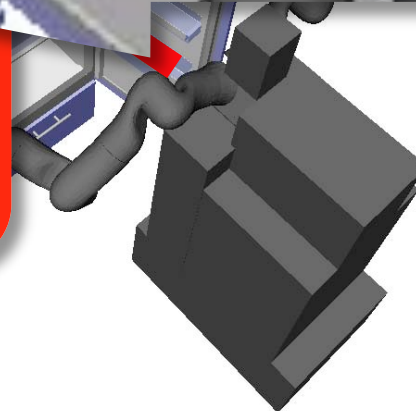
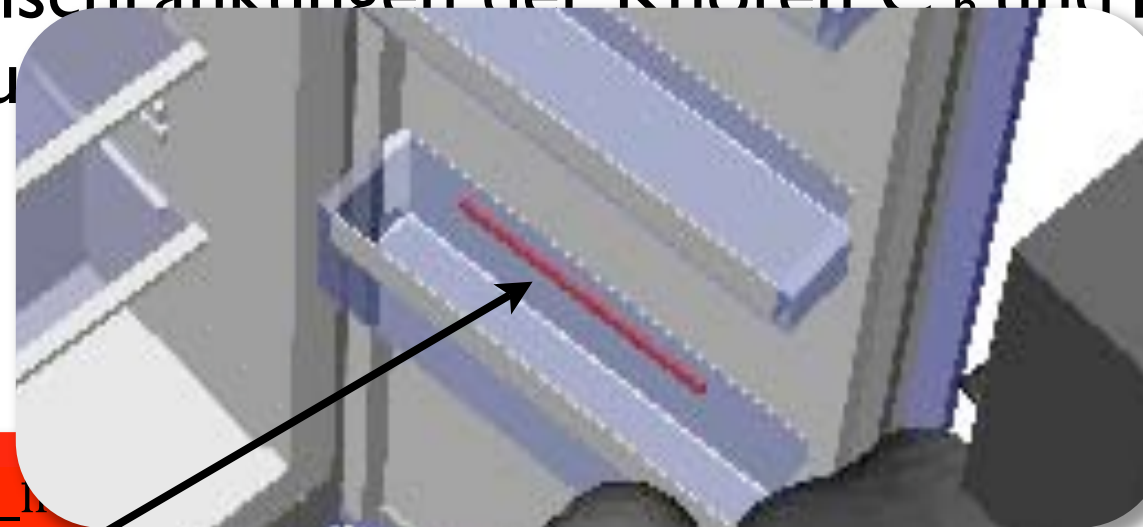
„Strategiegraph“:

- Tupel $(X, C_n^t, C_e^t, C_n^b, C_e^b)$
- Knoten X , zeitliche Einschränkungen der Knoten C_n^t und Kanten C_e^t , Bewegungseinschränkungen C_n^b und Kanten C_e^b



flasche_direction in Cube(start_position),
flasche_position in Cube(kuehlschrank_door_open),
collision in Cube(),
flasche_contact in Cube(kuehlschrank)

flasche_direction in Cone(flasche),
flasche_position in Cube(kuehlschrank_door_open),
collision in Cube(),
flasche_contact in Cube(kuehlschrank)

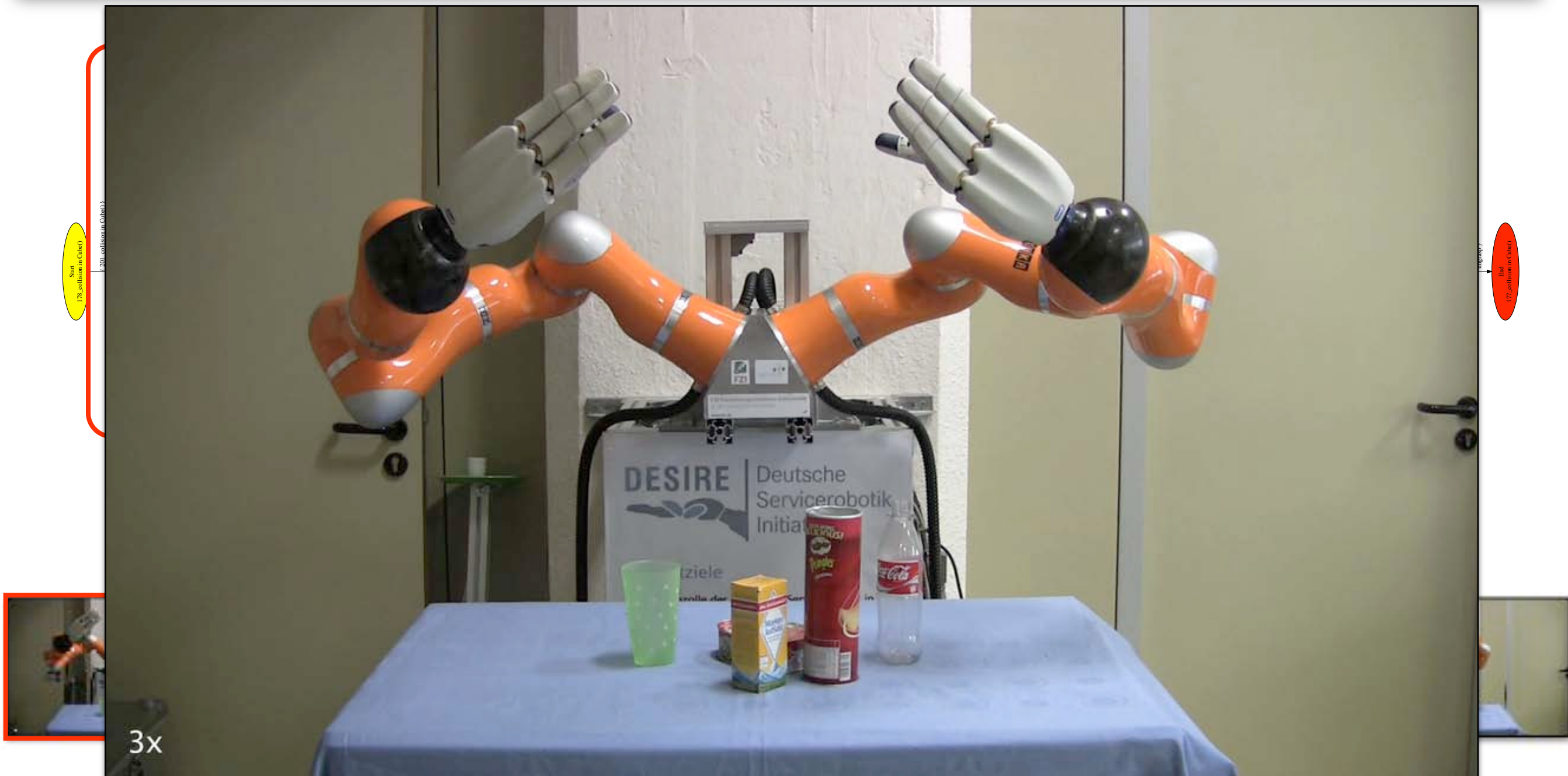


Strategiegraph

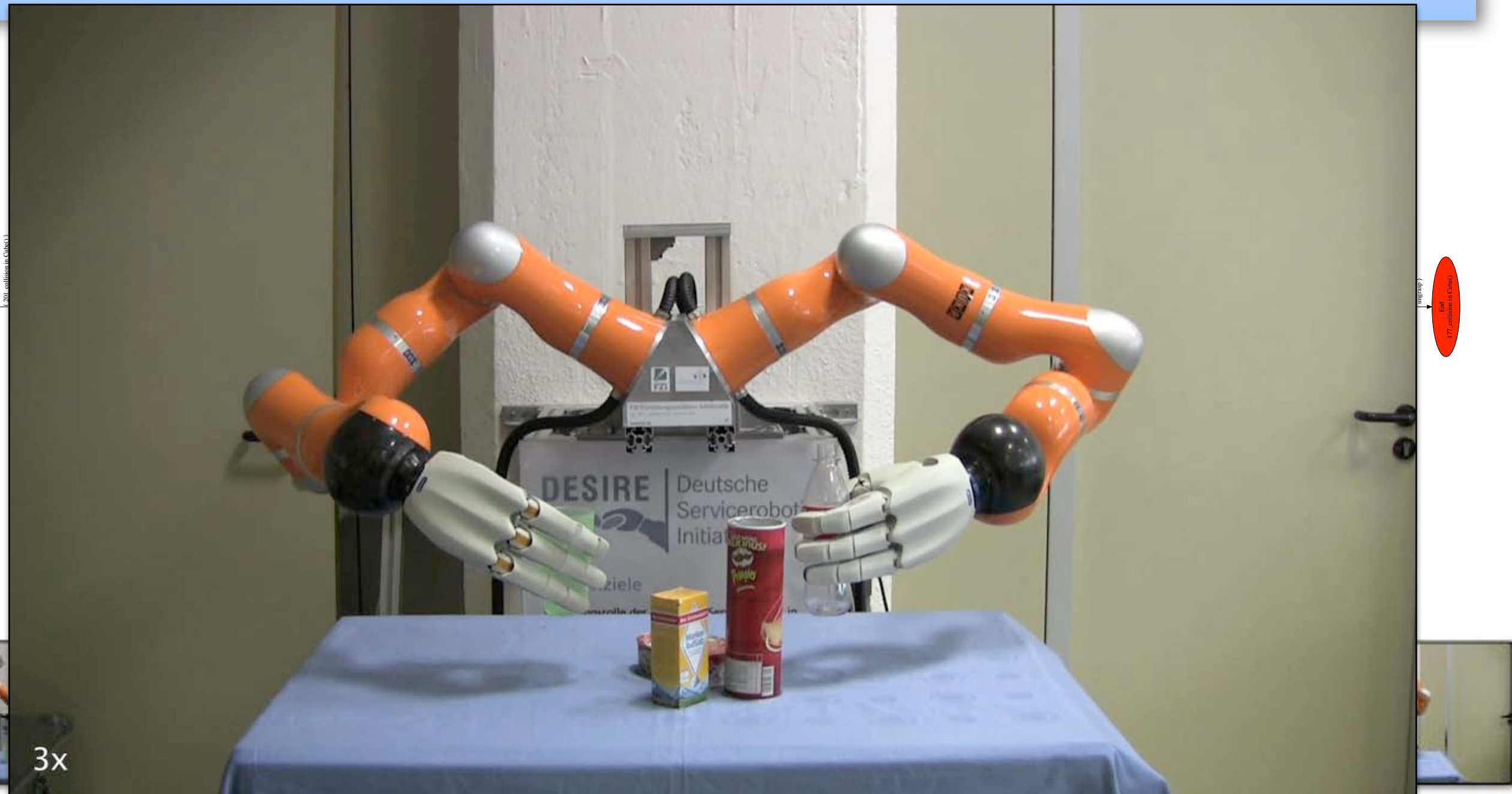
Bewegungseinschränkungen (rot)

Planung / Ausführung

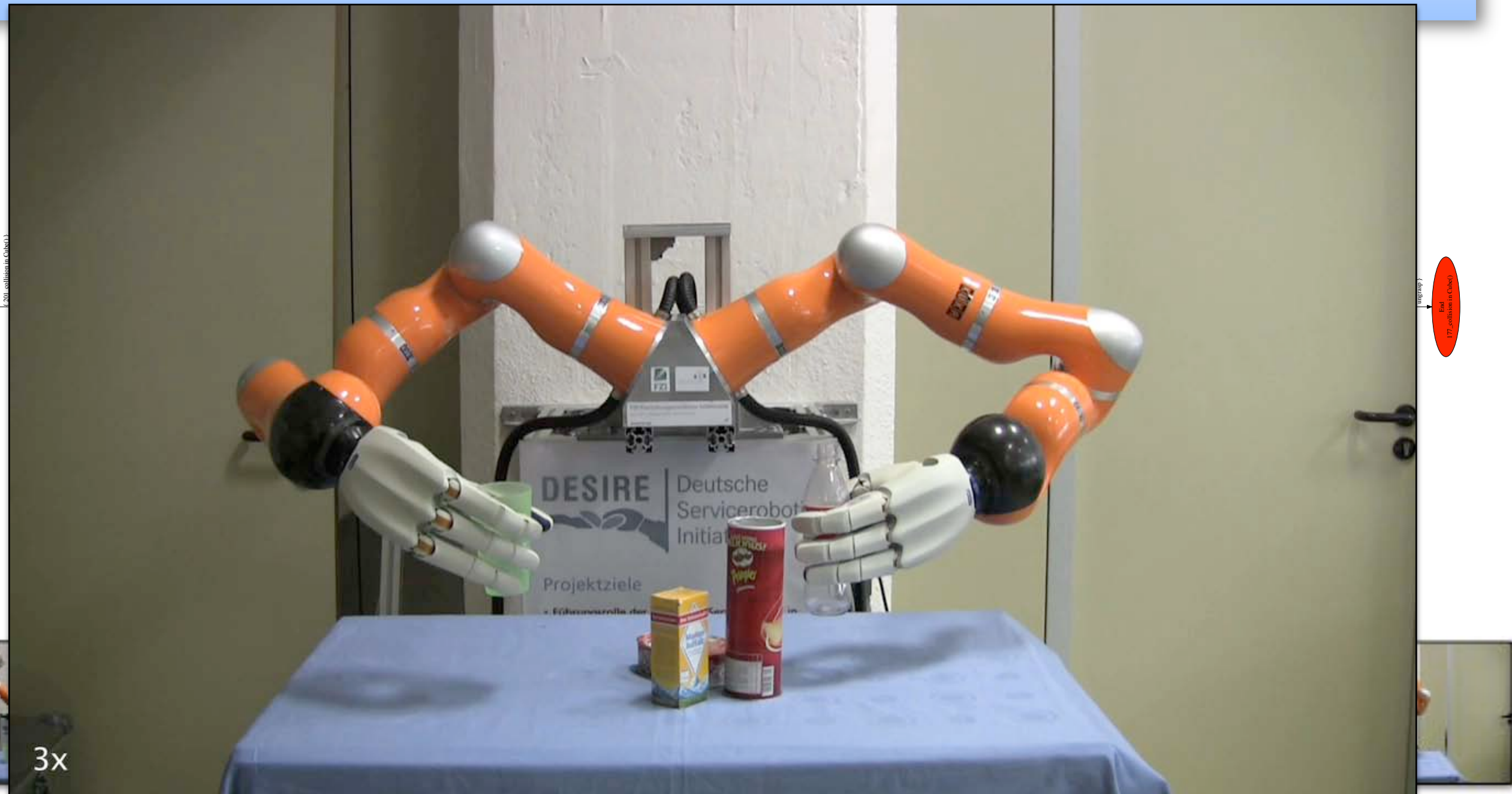
Repräsentation: Beispiel



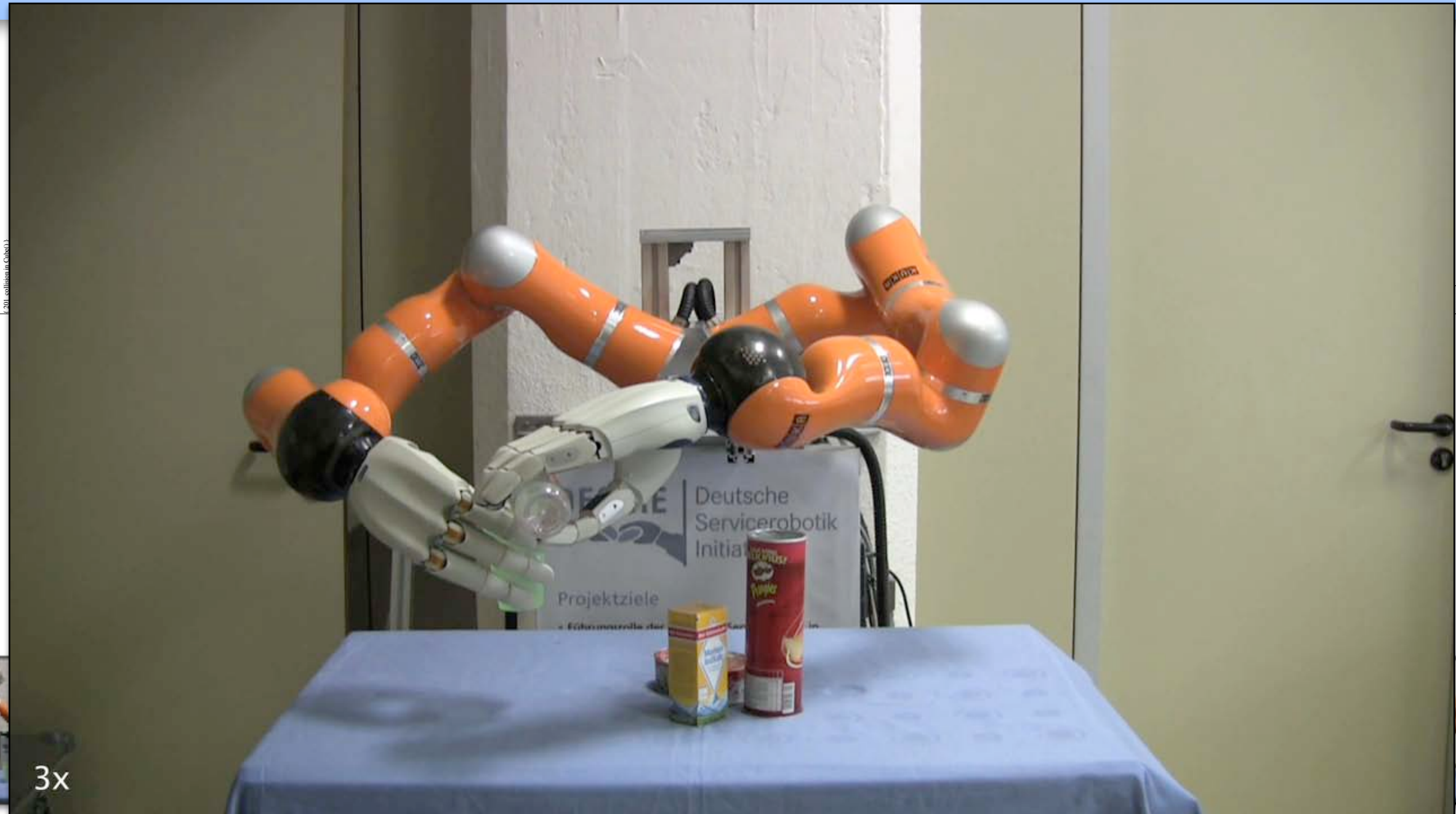
Repräsentation: Beispiel



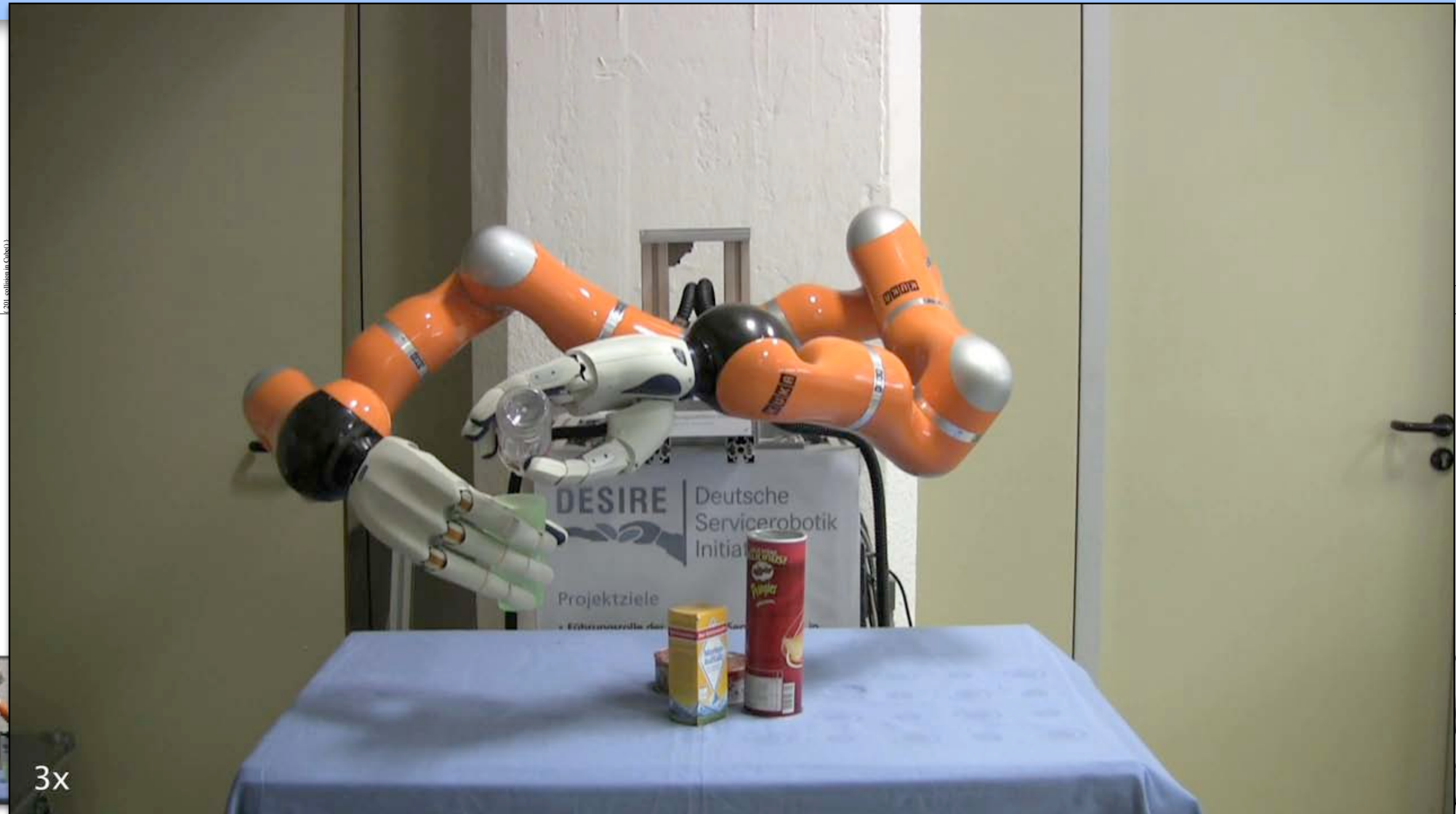
Repräsentation: Beispiel



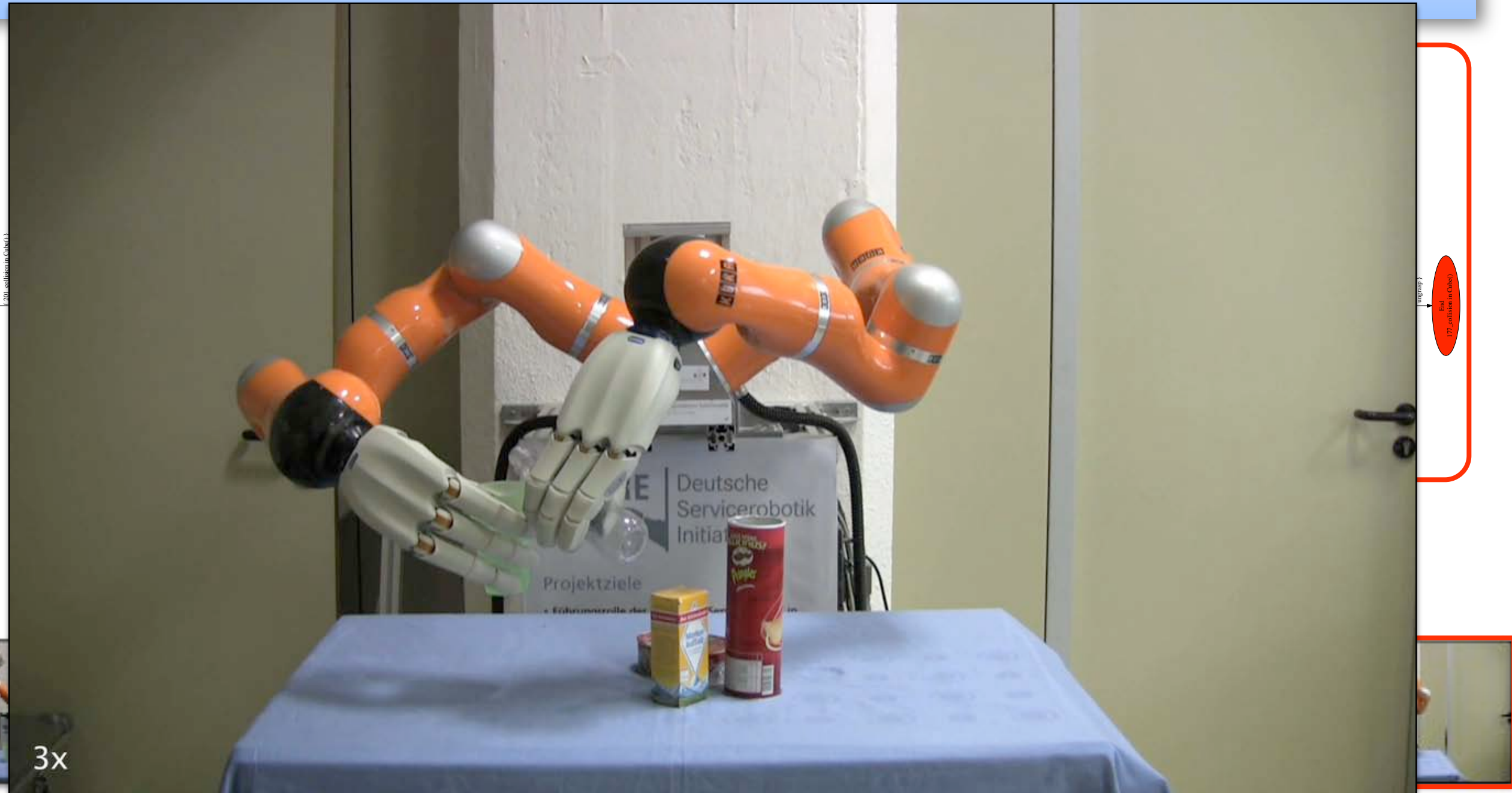
Repräsentation: Beispiel



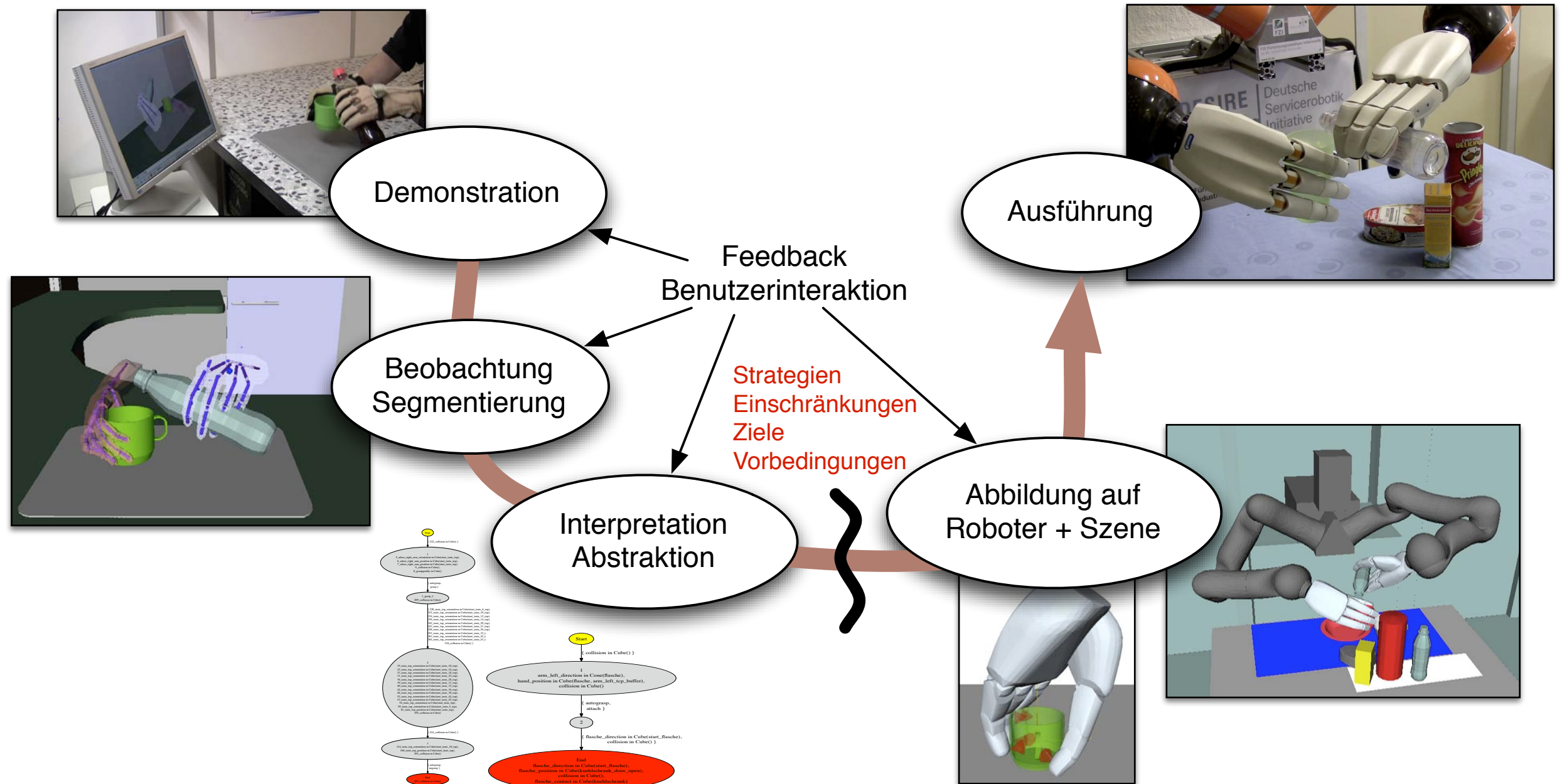
Repräsentation: Beispiel



Repräsentation: Beispiel



Lernen: PdV-Zyklus



Demonstration: sensorische Erfassung



Stereokamera mit
Pan-Tilt-Unit

Flock-of-Birds



Mikrofon

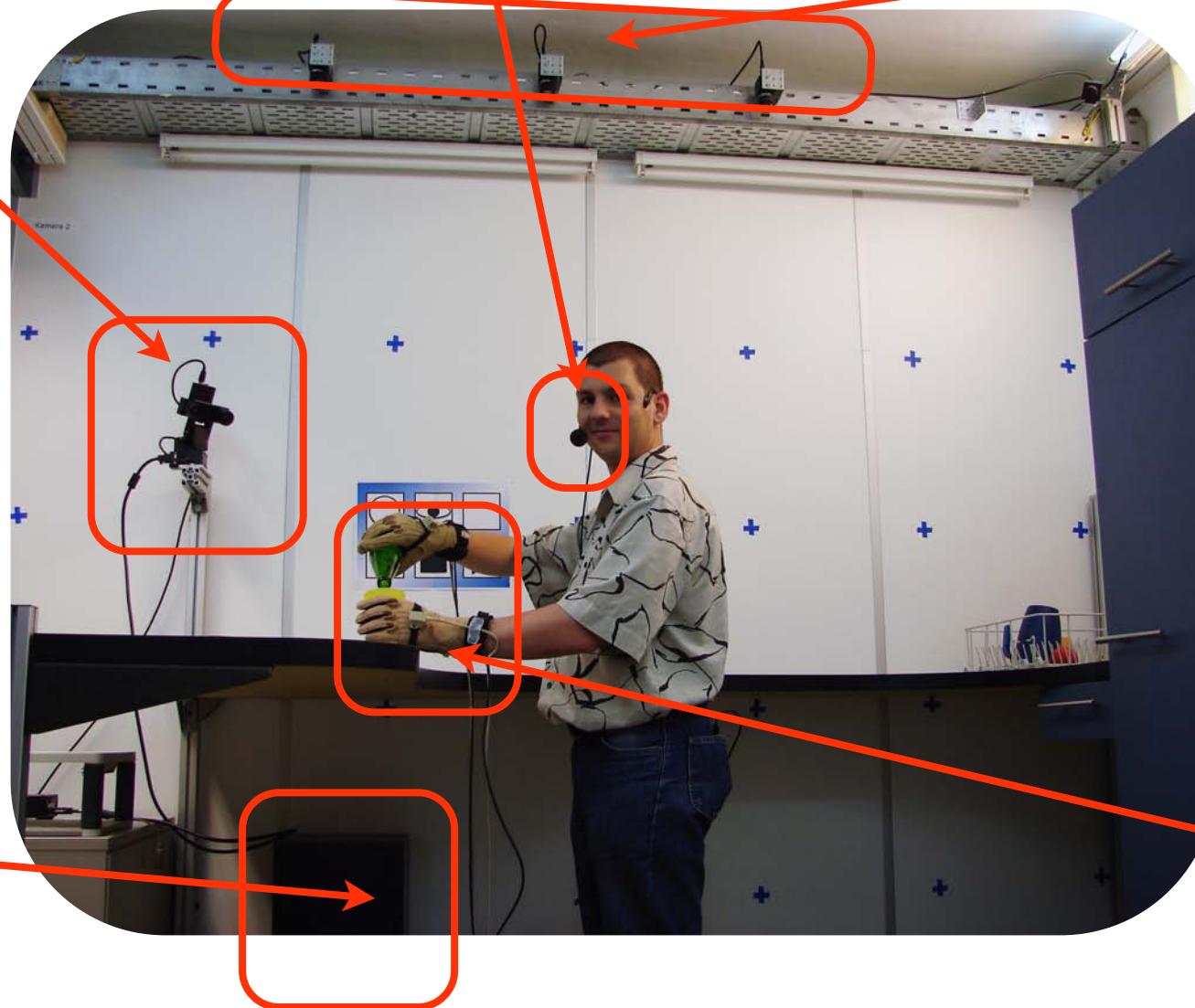


Deckenkameras

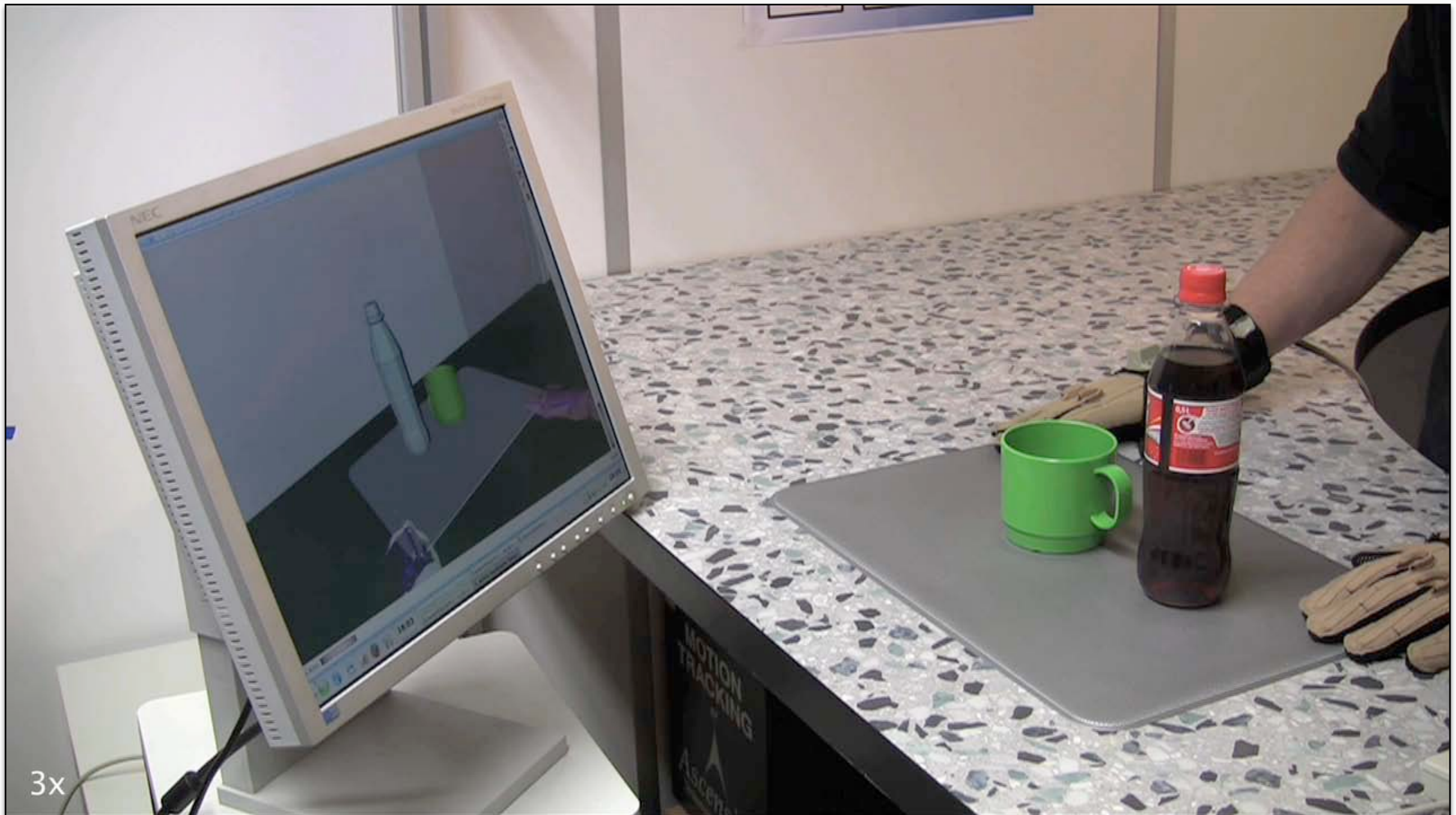


Voodoo

Cybergloves



Beobachtung (I)

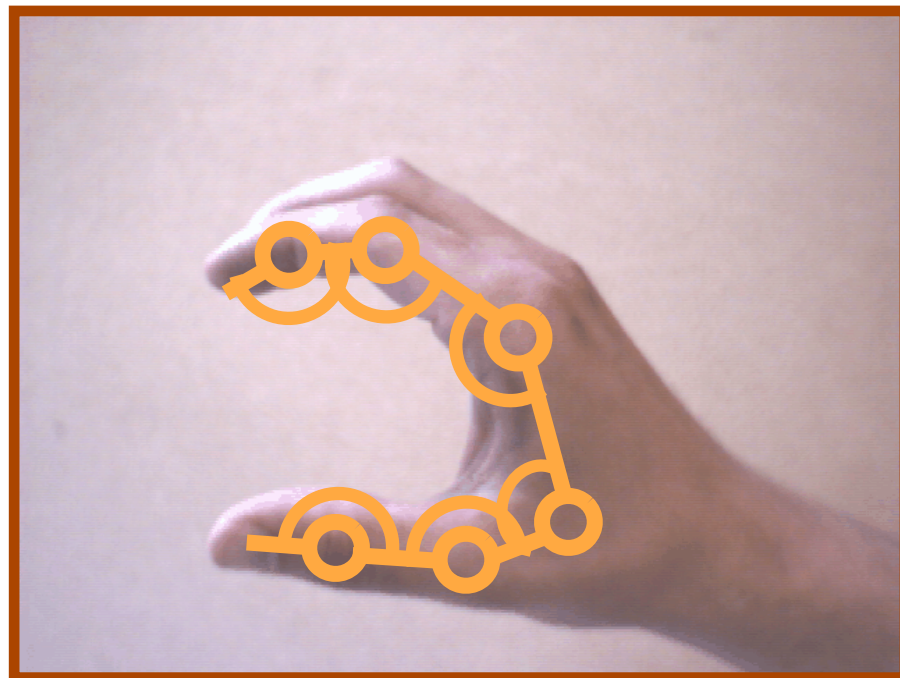
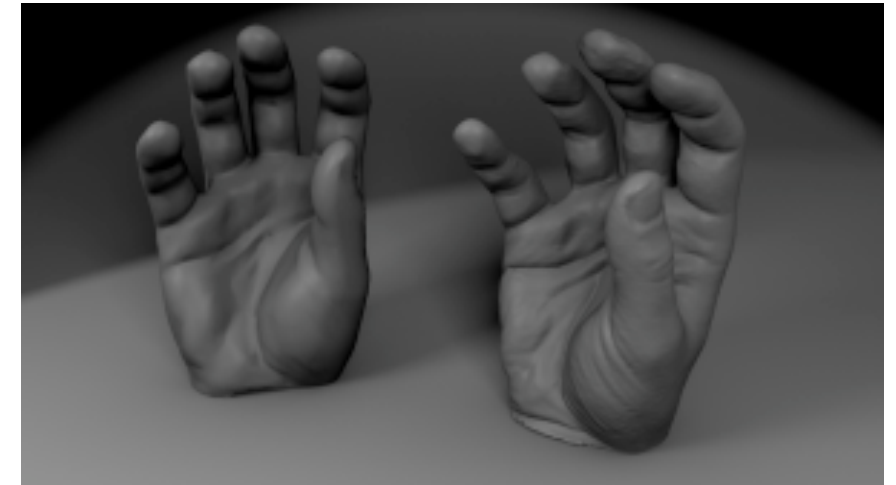


Beobachtung (II)

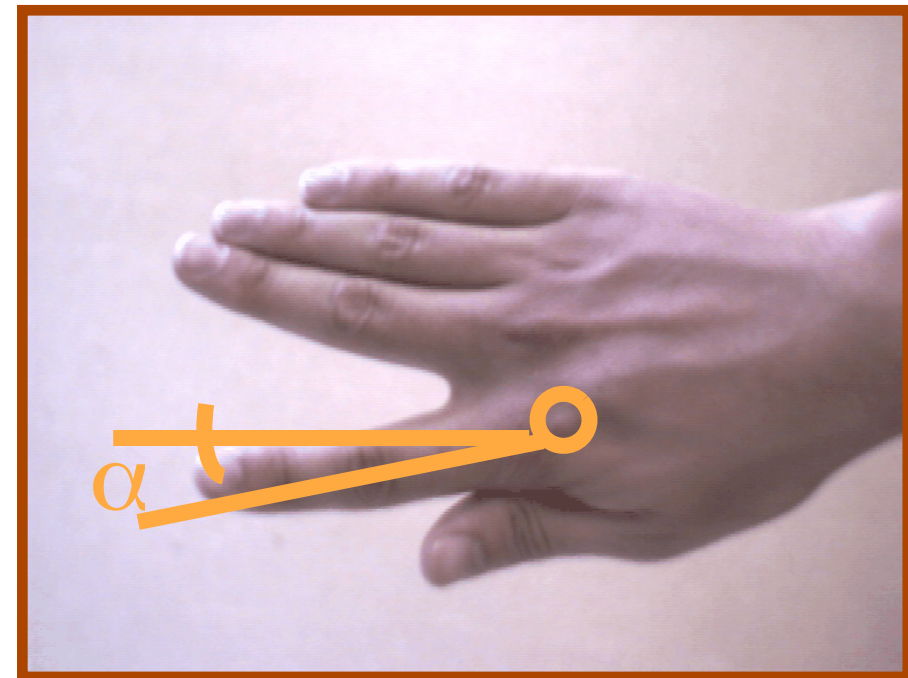
Virtual Technology - Datenhandschuh

Meßprinzip: Dehnmessstreifen

20 Fingerbeugungs- und Spreizwinkel +
2 Freiheitsgrade im Handgelenk



Beugungswinkel

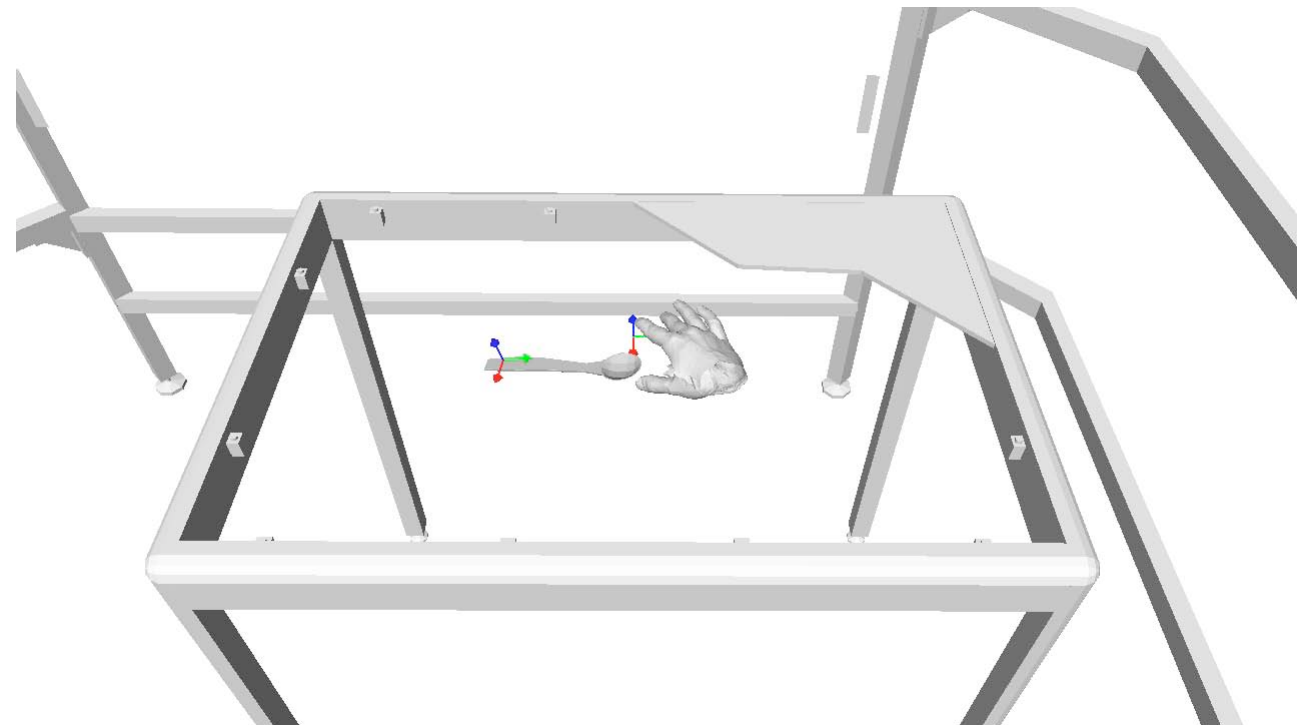


Spreizungswinkel

Beobachtung (III)

Bestimmung der Position und Orientierung der menschlichen Hände sowie von Objekten

- Magnetfeldbasierter Positionstracker
- Stereokamera



Segmentierung (I)

Ziel (mit Interpretationsphase):

- Repräsentation der demonstrierten Handlung durch Sequenz von zu erfüllenden Teilzielen (= Topologie des Strategiegraphs)

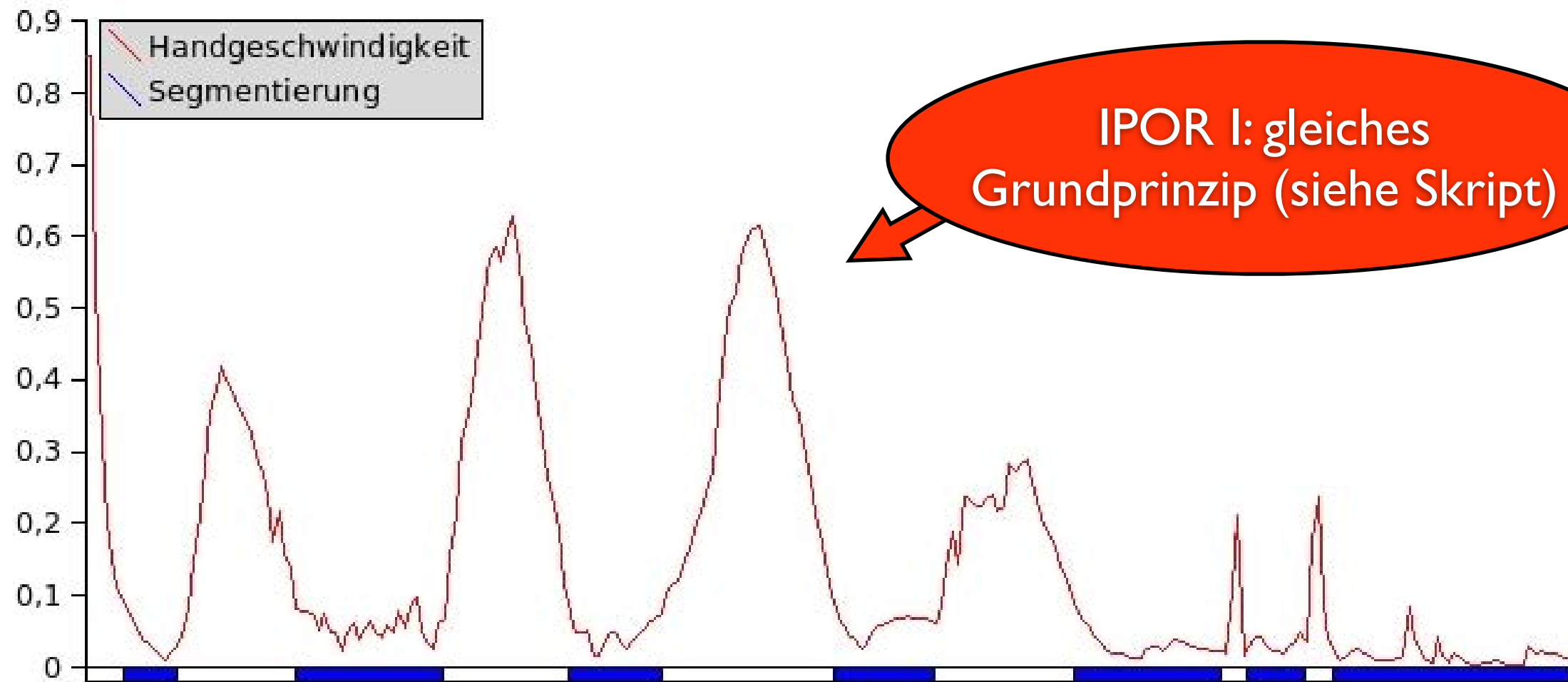
Ansatz:

- schwellwertbasierte Segmentierung zur Bestimmung von markanten Zeitpunkten der Demonstration

Vorteile:

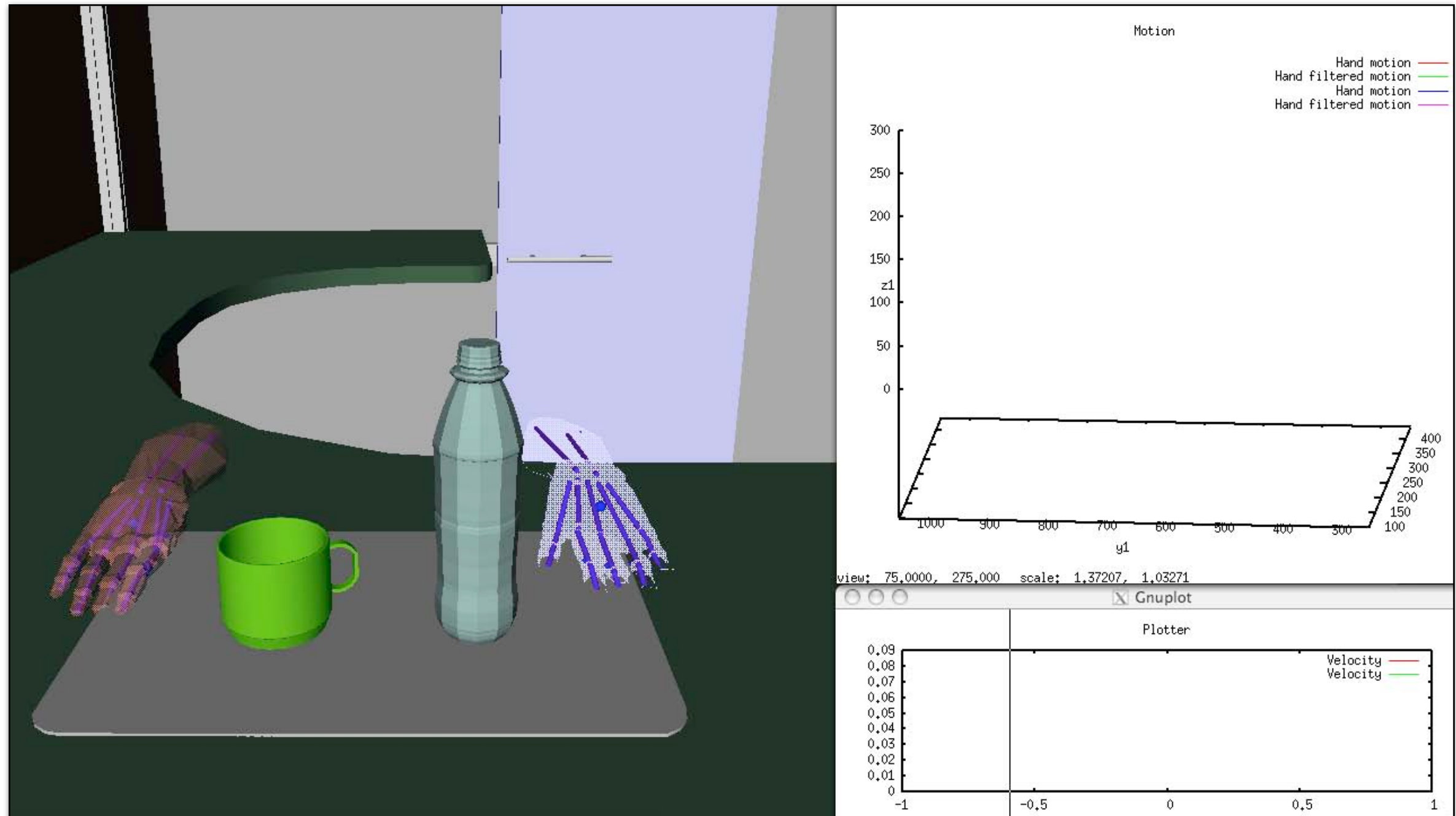
- Einfache Interaktionsmöglichkeit während der Demonstration
- Einfache Korrektur von Hypothesen

Segmentierung (II)



- Erzeugung eines Segmentierungspunkt, wenn Hand-, Fingergeschwindigkeit gering ist und mindestens ein Finger Objektkontakt hat

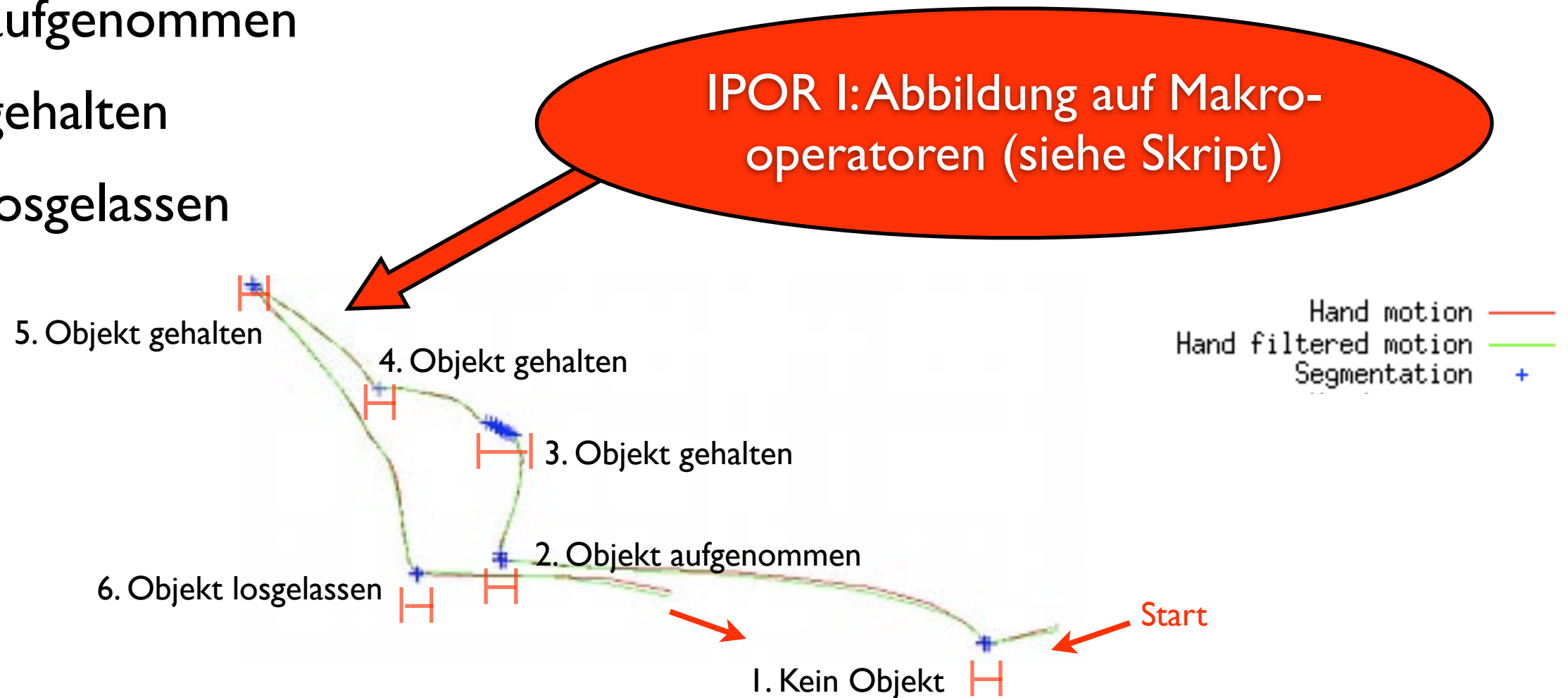
Segmentierung (III)



Interpretation (I)

Klassifikation der Segmentierungspunkte in 4 Typen auf Basis des Weltzustands an den Intervallgrenzen:

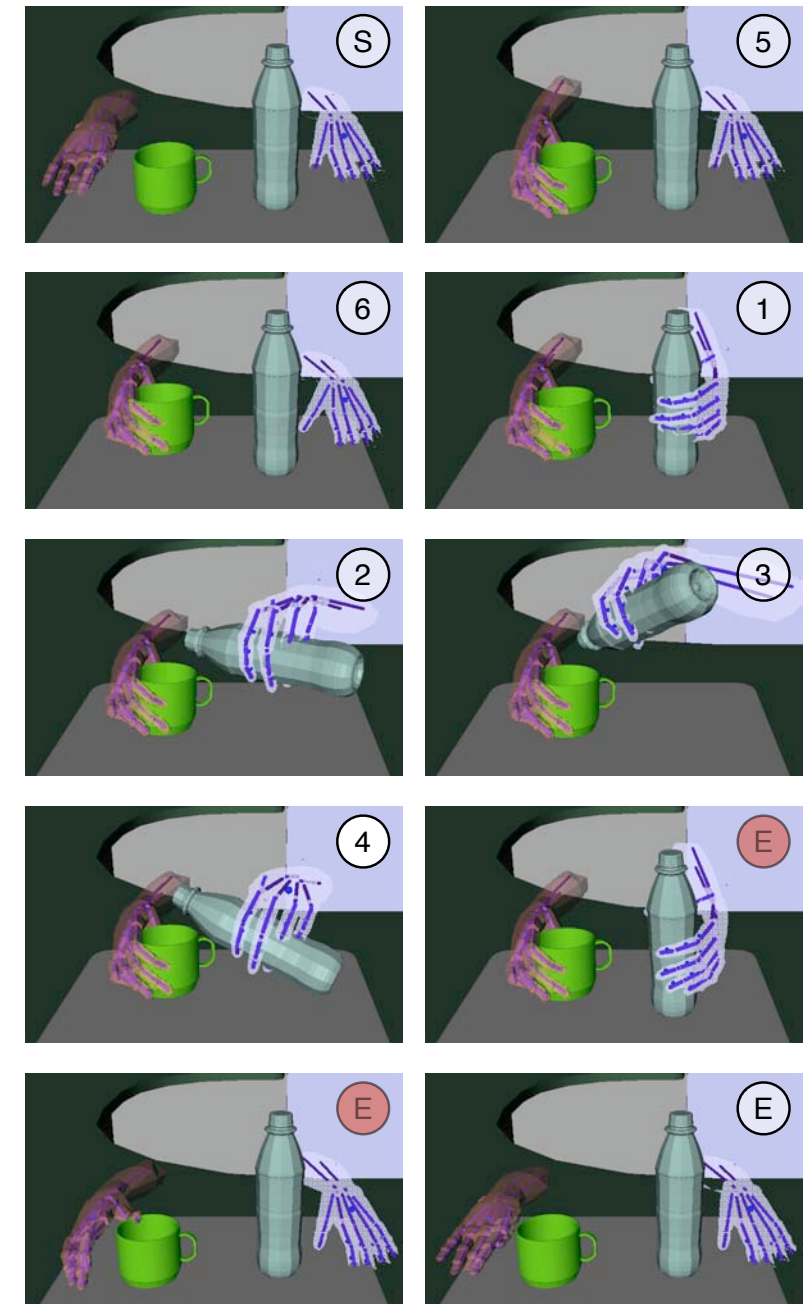
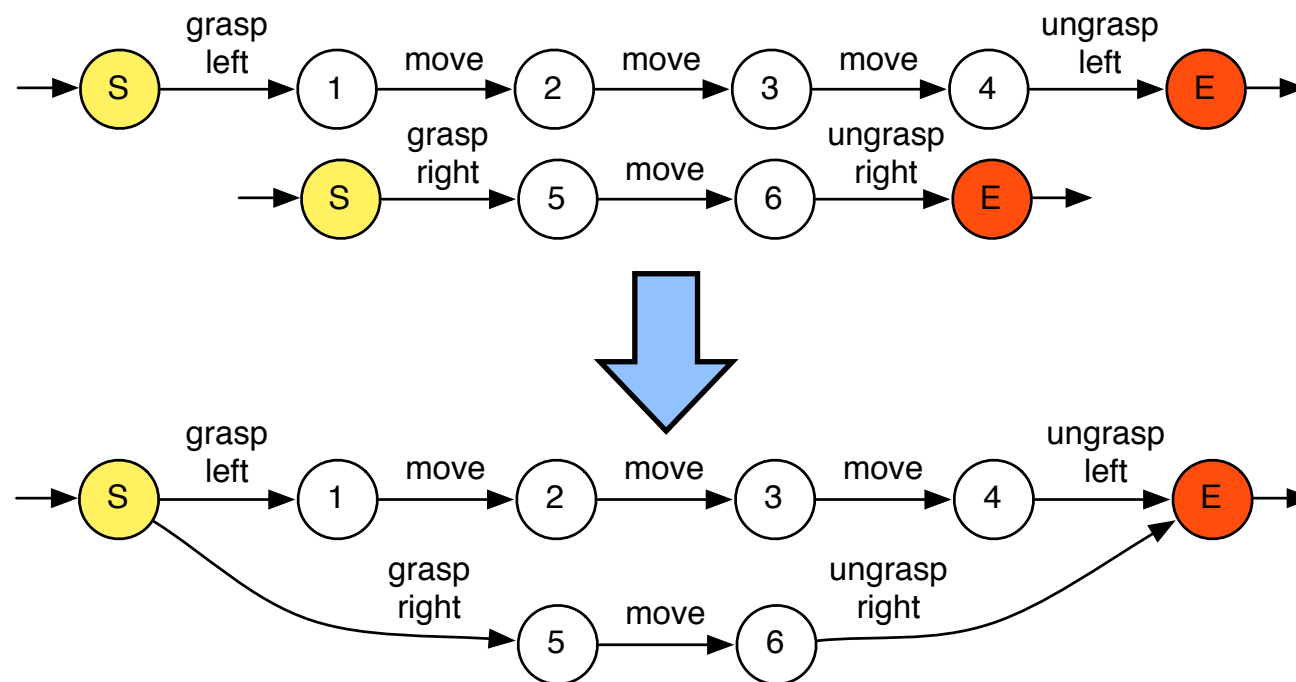
- Kein Objekt
- Objekt aufgenommen
- Objekt gehalten
- Objekt losgelassen



Interpretation (II)

Lernen des Planungsmodells

- Topologie des Strategiegraphs:
 - Segmentierung der Bewegungen des linken und rechten Arms
 - Kombination zu einem Graphen



Interpretation (III)

Lernen des Planungsmodells

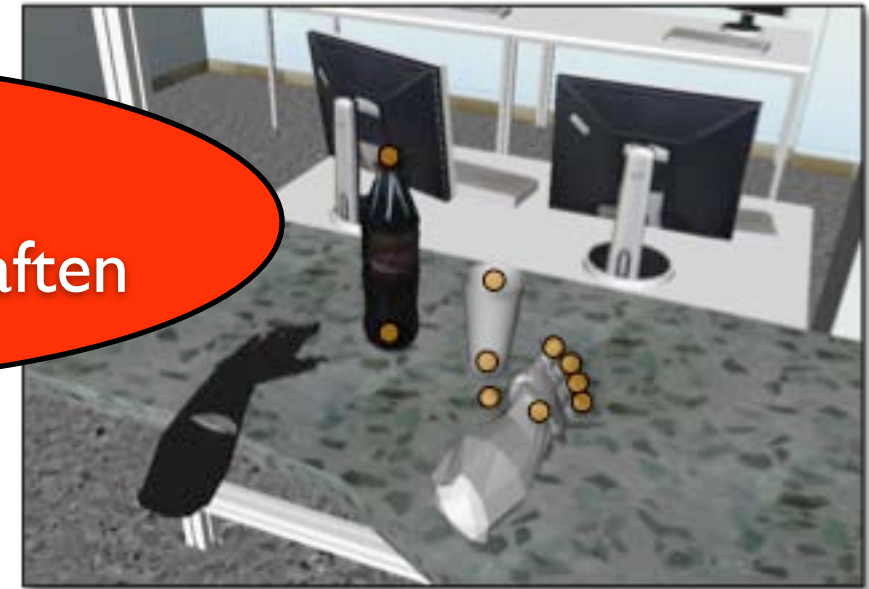
- Erzeugung der Bewegung

- Manuell definierte Kanten

alle Objekte:

$K = \{ \text{Flaschenöffnung, Becheröffnung,} \\ \text{Rechter Zeigefinger, Welt, ...} \}$

Objektmodell:
zusätzliche Eigenschaften

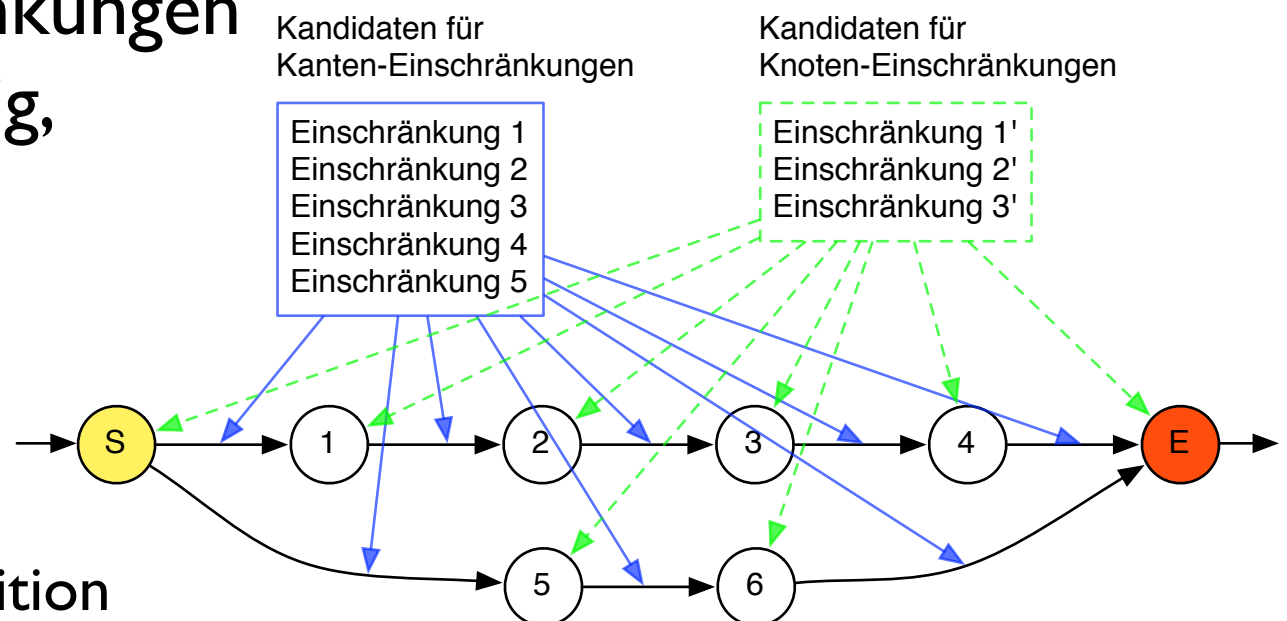


- Erzeugung aller möglichen Einschränkungen
(t, f, M, g, R) mit $f, g \in K$, Typ t beliebig,
für jeden Knoten und jede Kante
→ Region R muss bestimmt werden

- Beispiel:

$f = \text{Flaschenöffnung, } g = \text{Welt, } t = \text{Richtung}$

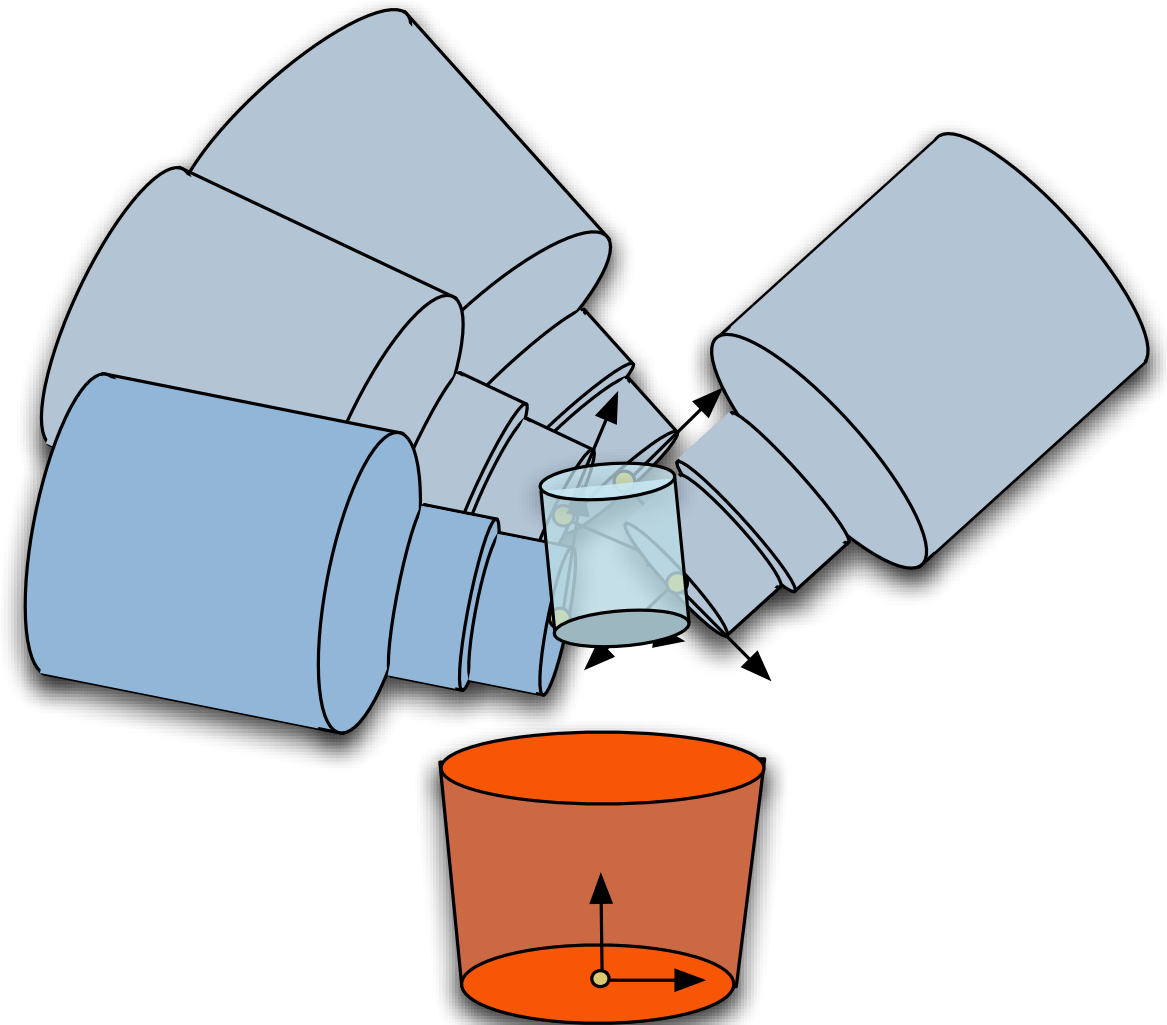
$f = \text{Flaschenöffnung, } g = \text{Becheröffnung, } t = \text{Position}$



Interpretation (IV)

Bestimmung der Region R

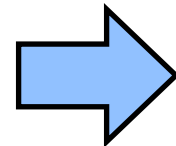
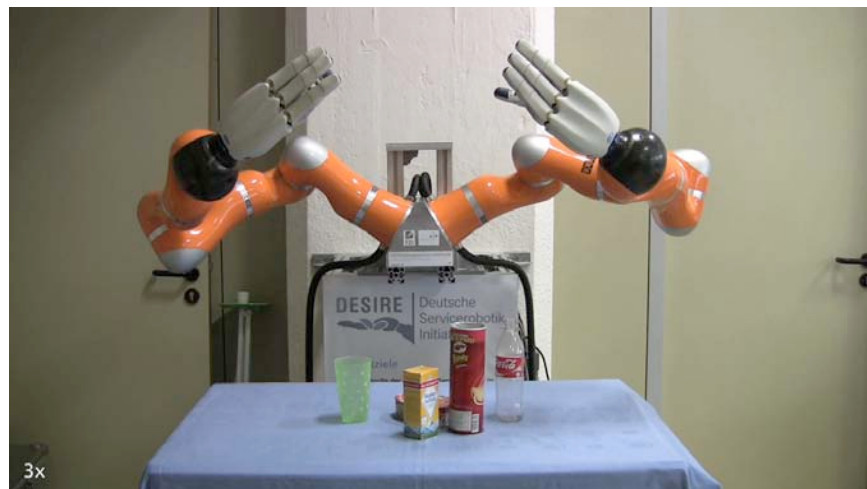
- Für jede Einschränkung (t, f, M, g, R):
 - Bestimmung des Werts von f in jedem Punkt t_i des Segments: ${}^gH_f(t_i) \cdot M$
 - Umwandlung in 3d-Vektor $m'(t_i)$
 - Ergebnis: Menge von 3d-Vektoren
- Bestimme Region R, die alle 3d-Vektoren einschließt
- Beispiel: Knoten-Einschränkung
 - f = Flaschenöffnung, g = Becheröffnung, t = Position



Abstraktion

Ziel: Anwendbarkeit des Planungsmodells in neuen Situationen auf verschiedenen Robotern

- Weitere Einschränkungen: Kräfte, Kontakte, Objektbewegung
- Wesentliche Abstraktion durch Koordinatensysteme und Einschränkungen
 - Abbildung von objektabhängigen Koordinatensystemen, z.B. Flasche.Öffnung → Milchpackung.Öffnung



Ausführung

- Gelernte Einschränkungen definieren Suchraum für Roboterbewegungen
- Einsatz von *Bahnplanung unter Einschränkungen* zur Ausführung von gelernten Planungsmodellen
- Einsatz von *Griffplanung* zur Bestimmung qualitativ hochwertiger Griffe

→ Vorlesung: PdV II

IPOR II

- Vorteile:

- Generalisierung auf Basis von Objekteigenschaften
- Start- und Zielbeschreibung, Validierbarkeit
- Hindernisvermeidung und Berücksichtigung von Einschränkungen
- mehrere Lösungen und beliebige Optimalitätskriterien

- Nachteile:

- hoher Aufwand (Planungszeit, Simulationszeit)
- 3d-Modelle der Objekte, menschlichen Hand notwendig
- automatische Segmentierung bei dynamischen Bewegungen schwierig

Programmierung durch Vormachen

Klassifikation und Vergleich

Klassifikation von PdV-Systemen

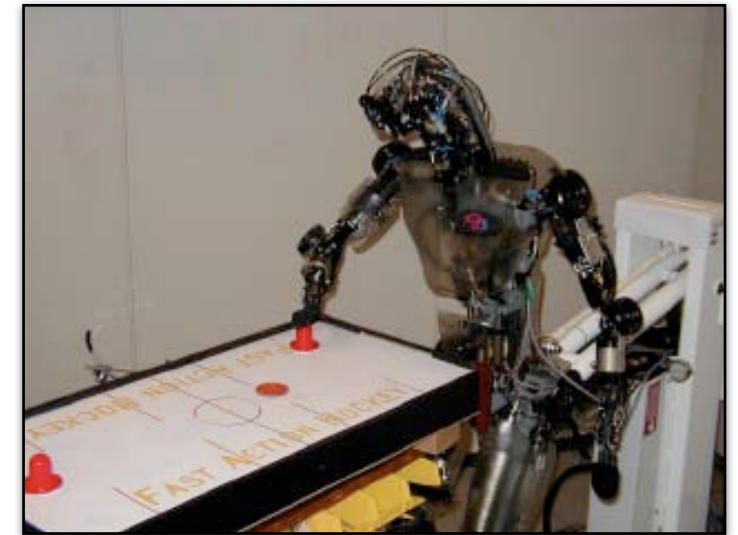
Klassifikationskriterien:

Elementarfähigkeiten (Skills)		Handhabungsfolgen	Komplexität der Aufgabe
aktiv	passiv	implizit	Art der Demonstration
Trajektorien	Umwelt- ereignisse	Positionen, Relationen	Repräsentation von Handlung und Umwelt
direkt		geplant	Technik der Trajektoriegenerierung

Elementarfähigkeiten (Reflexe, Basisskills, einfache Bewegungen)

- Lernen direkter Sensor-/Aktorzusammenhänge
- Beispiele: Verwendung neuronaler Netze auf adaptive Regelkreise, Zustandsautomaten
- Asada91, Koeppe95, Kaiser96, Ishikawa99, Bentivegna00, Calinon08, Pastor09

➔ Probleme: Explizite Beispiele, stark konfigurationsabhängig, viele Trainingsbeispiele notwendig



Komplexe Aufgaben (Task, Montageaufgaben)

- Interpretation von Handlungsfolgen
- Segre89, Inaba90, Kang94, Sagerer98, Aleotti06, Pardowitz07, Veeraraghavan08

➔ Probleme: Breites Hintergrund-, Planungs- und Modellwissen, Klärungsdialoge mit dem Benutzern



Pick



Transport



Place

Aktive Beispiele

- Benutzer führt explizit vor
- Beobachtung durch Sensorsystem (Datenhandschuh, Kameras)
- Friedrich98, Ikeuchi99, Inoue/Kuniyoshi94
Zöllner06



➔ Probleme:

- Aufwendige Sensorsysteme
- Identifikation relevanter Aktionen u. Ziele bzw. Zustände schwierig

Passive Beispiele

- Roboter wird durch externen „Master“ gesteuert, Signalaufzeichnung, Korrelation zwischen Sensor- und Aktordaten
- Kaiser96, Koeppe98, Billard07

➔ Probleme: Gelerntes Wissen ist auf konkretes Zielsystem festgelegt



Implizite Beispiele:

- Zielspezifikation durch Vorgabe graphischer Ikone
- Takahashi/Ogata97, Sagerer98, Riepp97

➔ Probleme: Dialog umfangreich, Anpassung an Zielsystem

Trajektorien

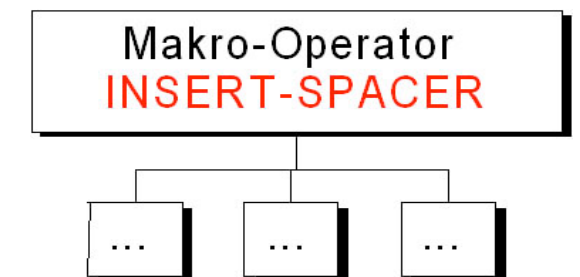
Umwelt-
ereignisse

Positionen,
Relationen

Repräsentation von
Handlung und Umwelt

Wirkungen und Reaktionen auf die Umgebung

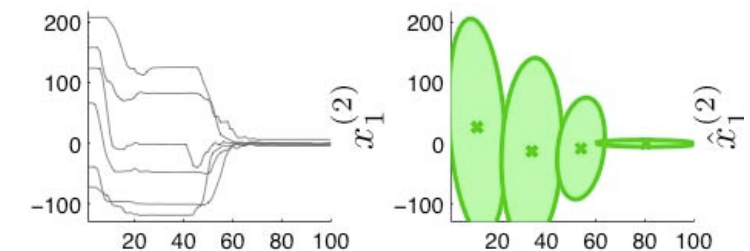
➔ Probleme: Identifikation von Kausalitäten und Zustandsfolgen durch kognitive Operatoren



Trajektorien

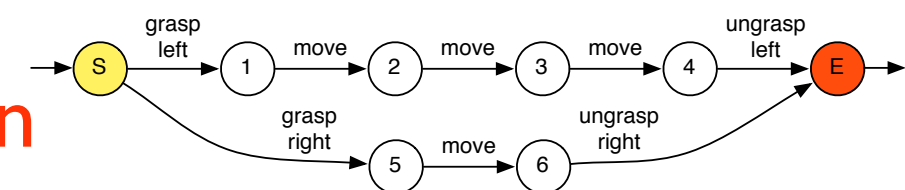
- Explizite analytische Repräsentation (exakt, approx.)

➔ Problem: keine wesentliche Generalisierung (1:1 Abbildung)



Objektlagen, Relationen und Operatoren

➔ Probleme: Beschränkung auf vorgegebenen Operatorumfang



direkt

geplant

Technik der
Trajektoriegenerierung

Planung von Roboterbewegungen / Aktionen

- Planung erforderlich zur Berücksichtigung der Unterschiede in Demonstrations- und Ausführungsumgebung
- Vollständige Nutzung der Leistungsfähigkeit des Robotersystems

➔ Probleme: Umwelt- u. Planungswissen erforderlich, intelligentes Planungssystem

Direkte Abbildung

- Explizites oder gelerntes Transformationsmodell
- Kang97, Kaneko97

➔ Probleme: Zielsystem u. Zielumgebung müssen korrespondieren

Ausblick: nächste Vorlesung

Interaktives Roboterprogrammieren II: Methoden

- Bahnplanung
- Grifftaxonomie
- Griffplanung

