

# Interaktive Programmierung

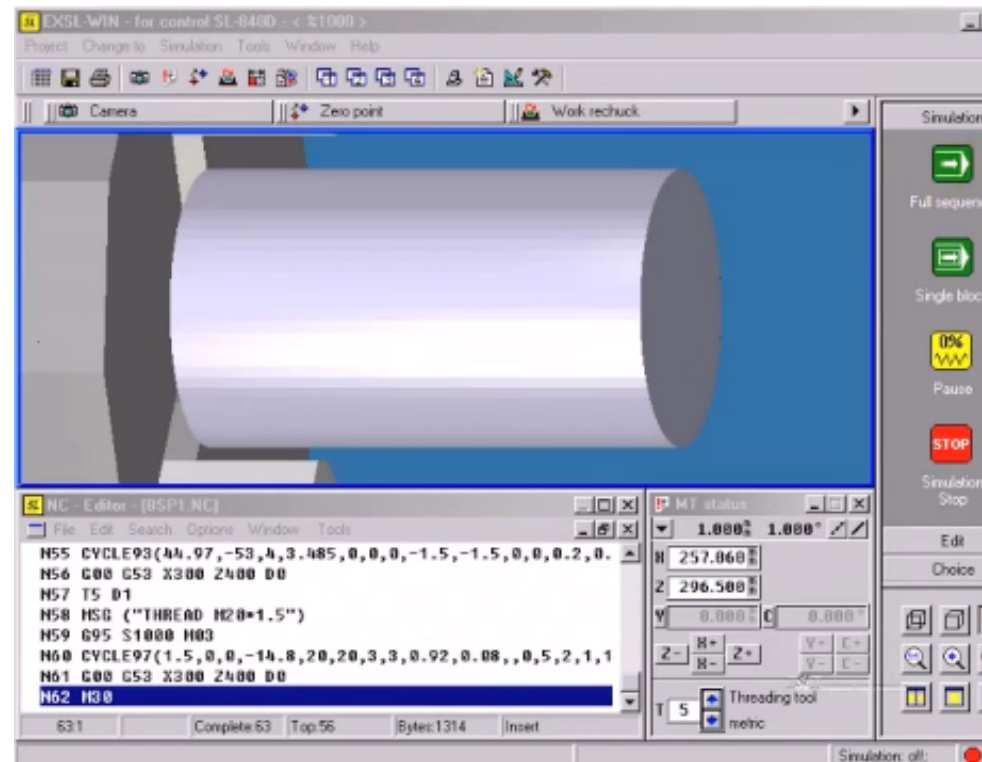
Prof. Dr.-Ing. Rüdiger Dillmann

- Motivation
- Grundlagen
- Interaktive Programmierverfahren
  - Probabilistisch
  - Dynamikbasiert
  - Planungsbasiert

**Klassische Roboterprogrammierung ist mit hohem Aufwand verbunden und erfordert Expertenkenntnisse**



Teach Panel



Textuelle Programmierung

## Neue Anforderungen in der Produktion

- Klein- & Kleinstserienfertigung
- Unikatfertigung (z.B. Prototyp)
- Produkte mit:
  - vielen Ausstattungsvarianten
  - hoher Rekonfigurierbarkeit



Flexible Fertigung



## Neue Anforderungen im Servicebereich

- Handel:
  - Kommissionierung und Palettierung von Waren
  - Bestücken von Regalen
- Qualitätssicherung
- Pflege:
  - Unterstützung von Rehabilitationsmaßnahmen
- Handwerk:
  - Handhabungen in Schreinereien und Schlossereien



## Anforderungen in der humanoiden Servicerobotik

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

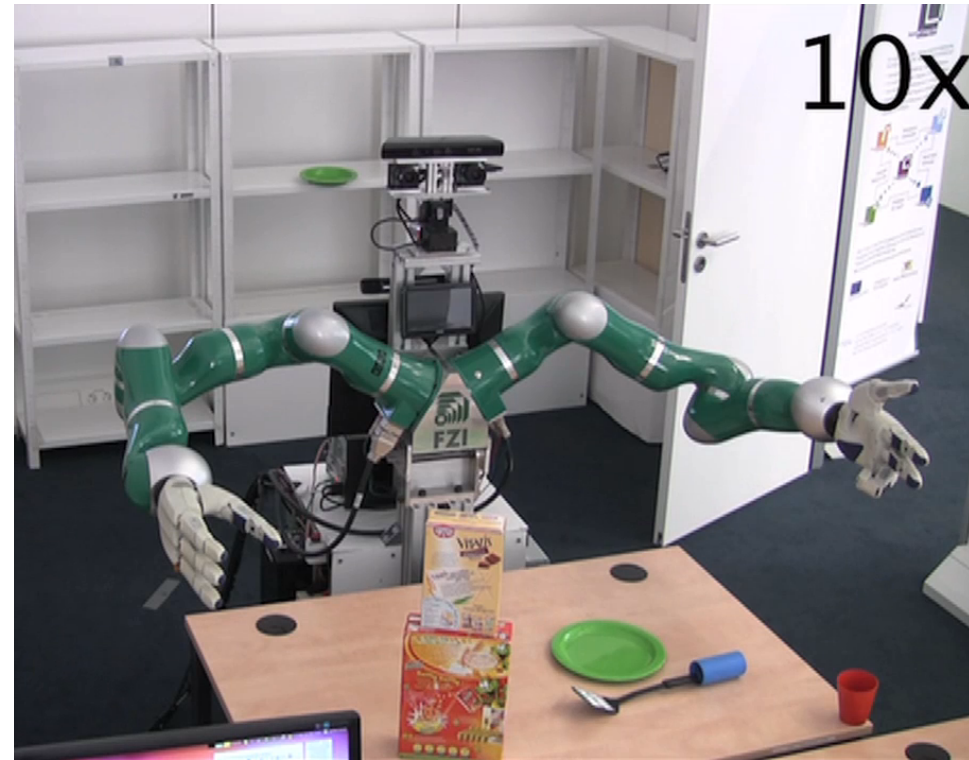


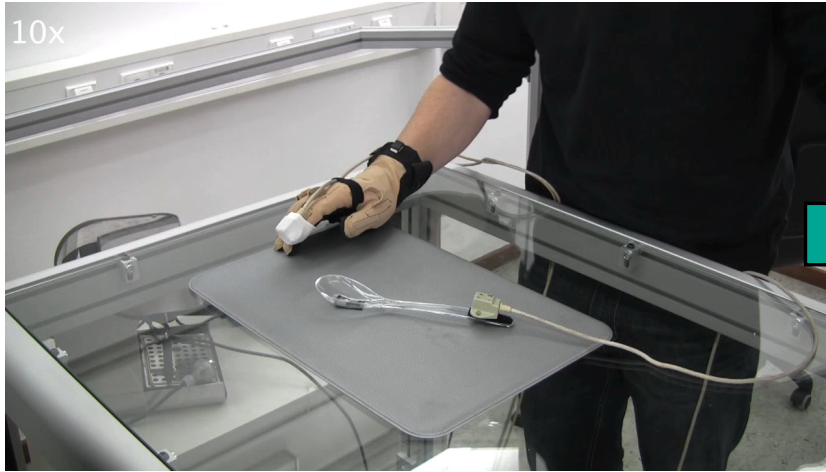
komplexe  
Umgebung!

sehr viele  
Freiheitsgrade!

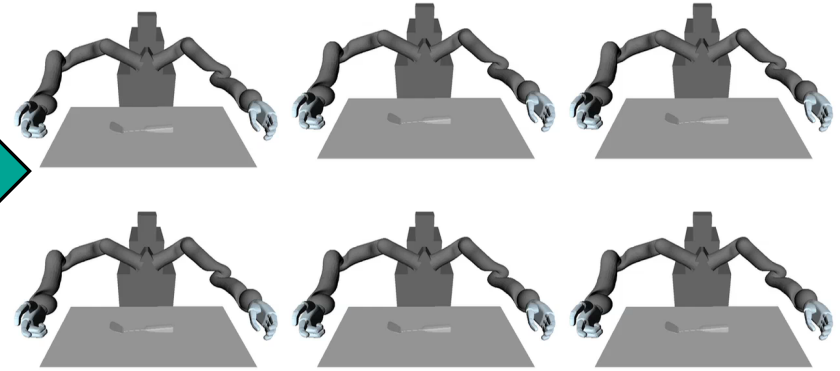


Programmierung?

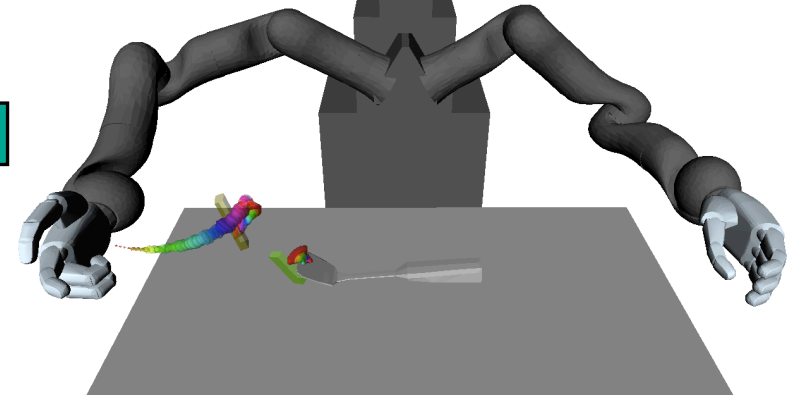




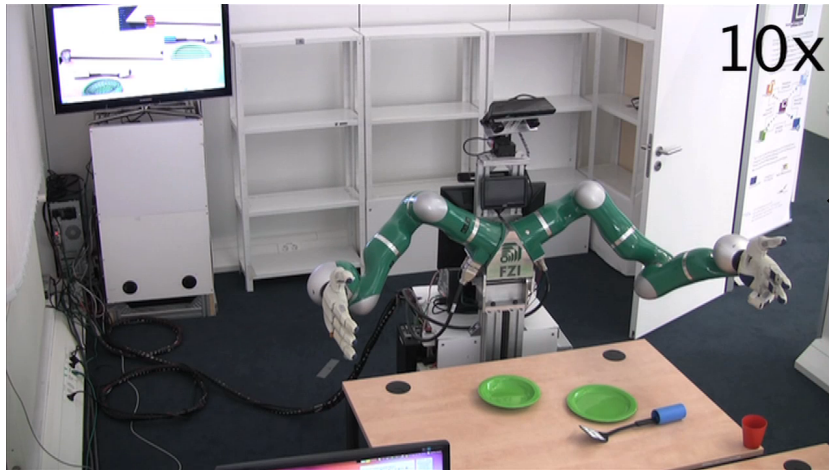
Demonstration & Aufzeichnung



Segmentierung & Interpretation



Abstraktion & Simulation



Ausführung

Bahnplanung

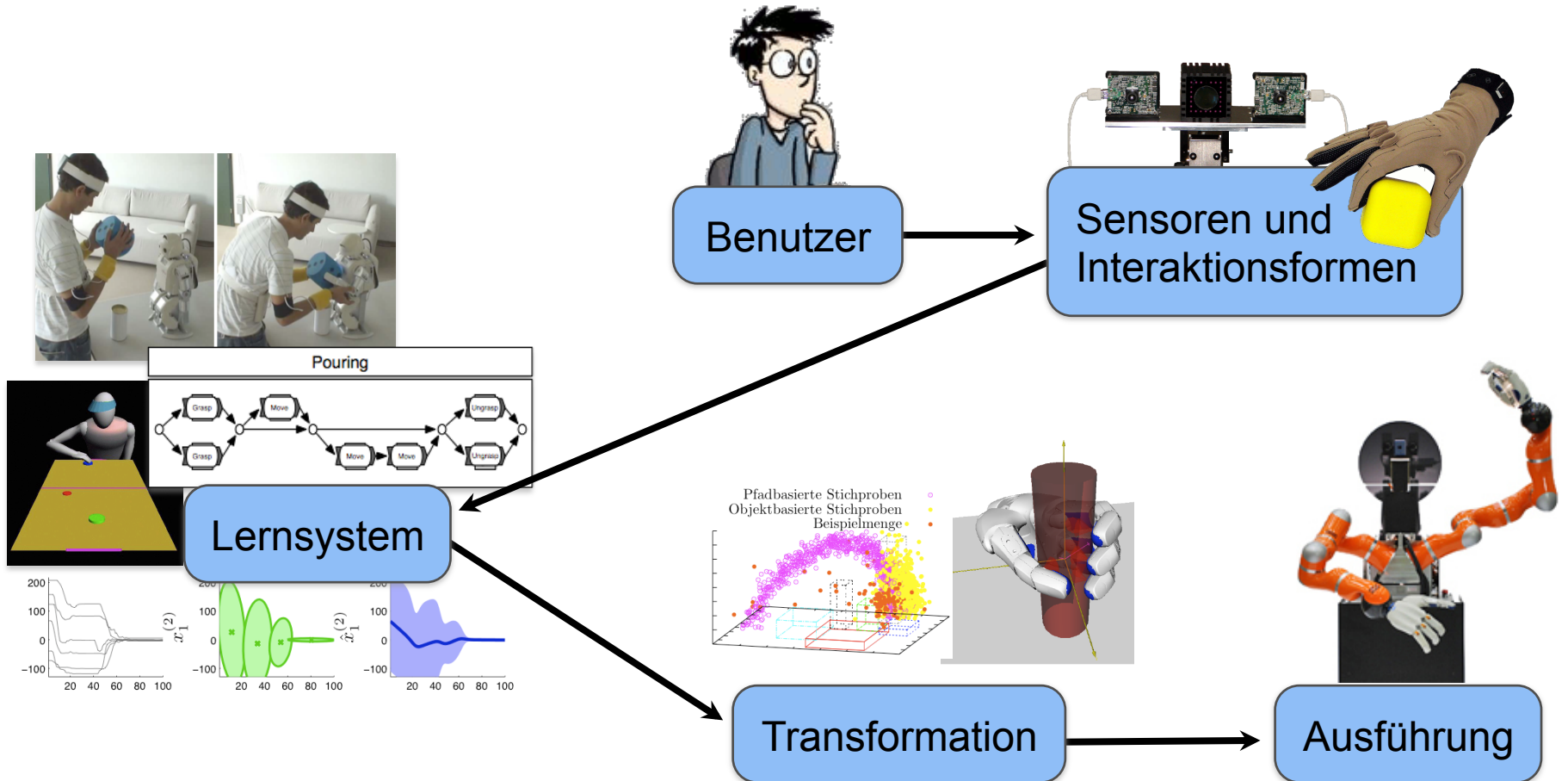
- Motivation
- Grundlagen
- Interaktive Programmierverfahren
  - Probabilistisch
  - Dynamikbasiert
  - Planungsbasiert



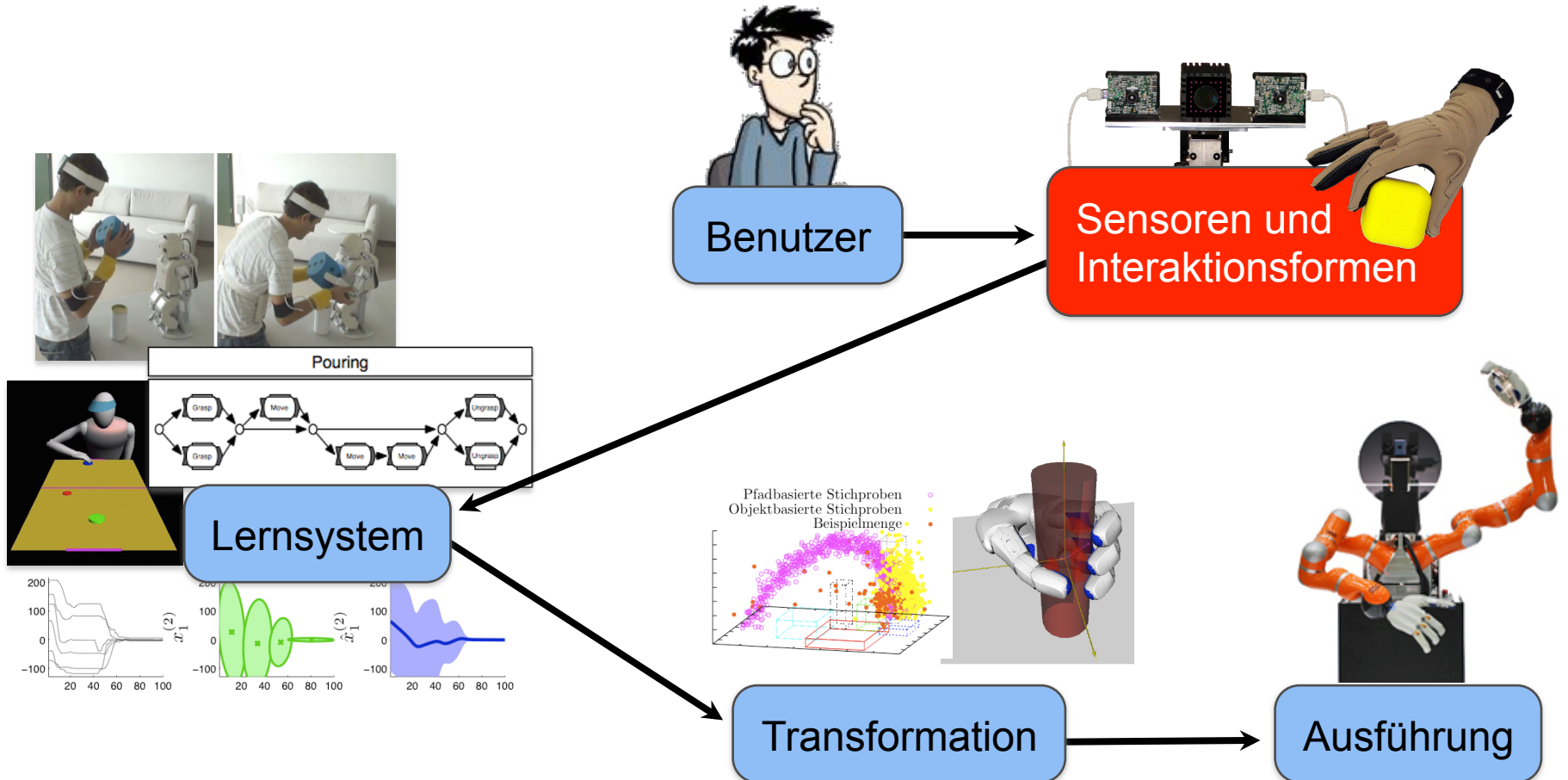
## Grundidee der Interaktiven Programmierung:

- Mensch ist Domänenexperte (Manipulation)
- Explizite Demonstrationen der Manipulationsaufgabe
- Sensorielle Erfassung der Demonstrationen
- Erzeugung der internen Repräsentation des Roboterprogramms
- Abbildung auf das Robotersystem
- Ausführung

## Komponenten:



## Komponenten:



## Interaktionsformen:

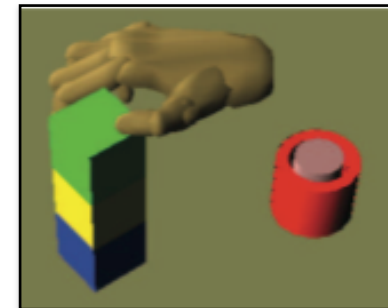
- Physische Demonstration
- Graphische Demonstration
- Ikonische Demonstration
- Kommentierung



Physisch:  
direkt



Haptisches Eingabegerät: indirekt



Graphisch:  
indirekt

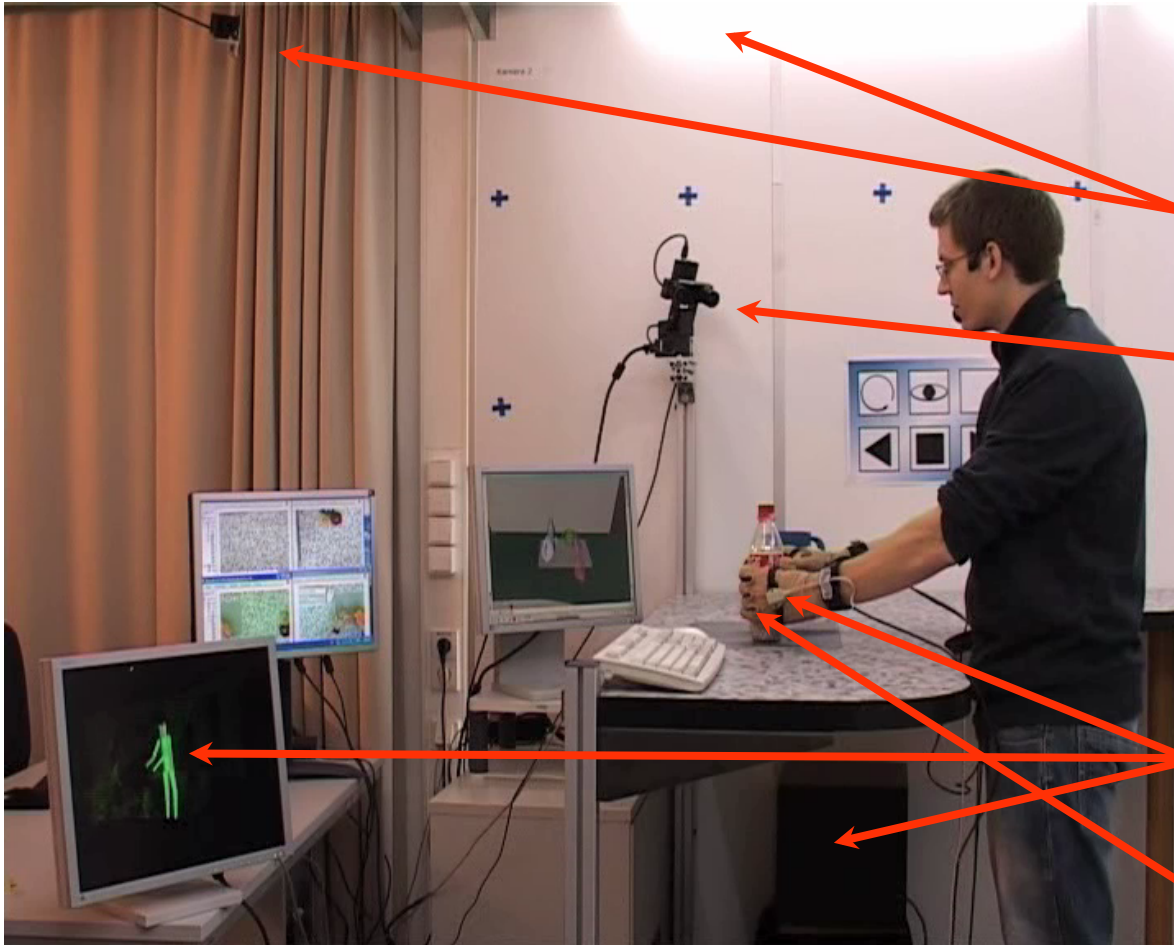
## Interaktionsformen



## Sensoren

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

## Sensoren



fixierte Deckenkamera

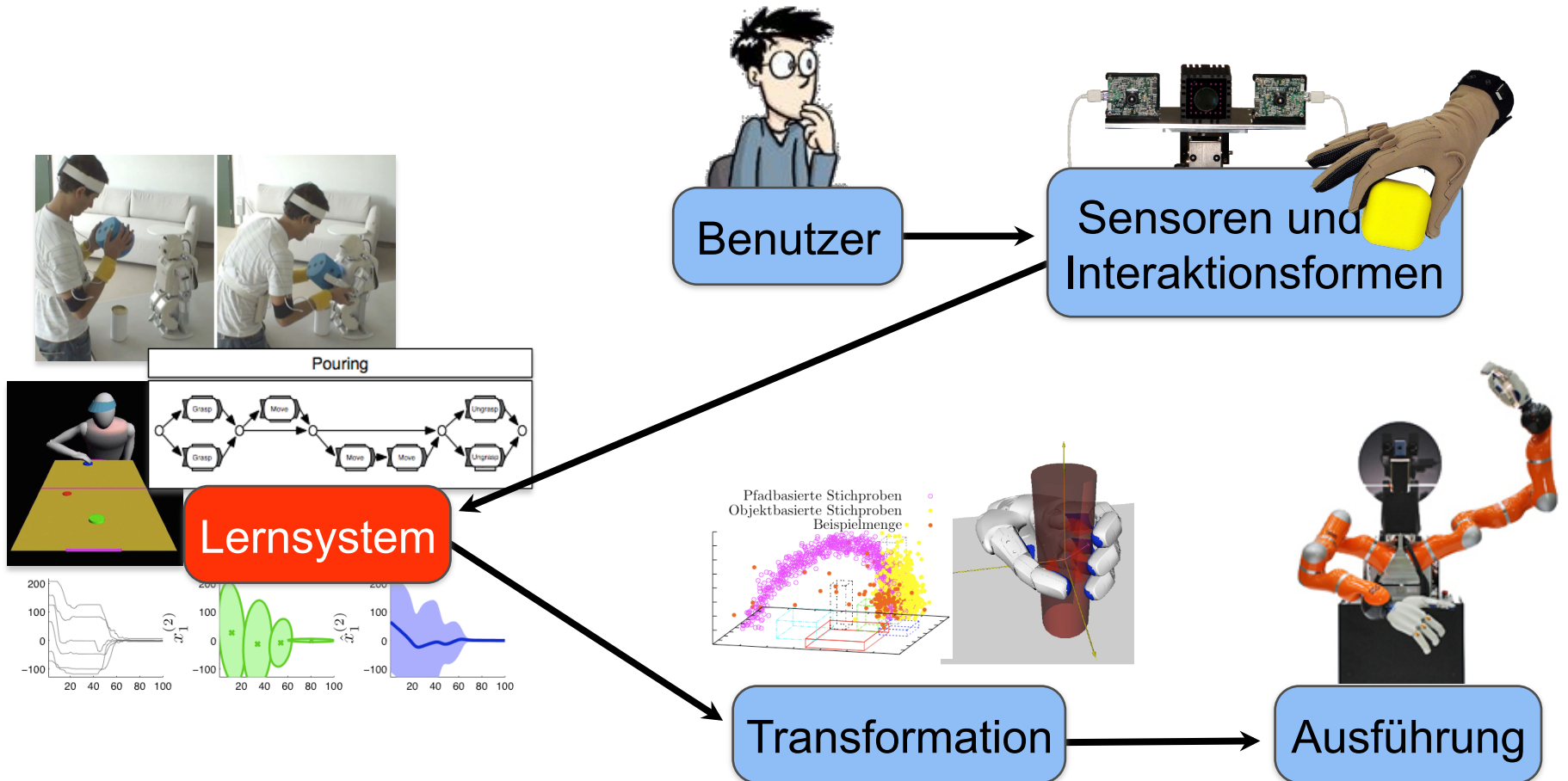
aktiver Kamerakopf

Taktile Sensoren im  
Handbereich

Trackingsystem

Datenhandschuh

## Komponenten

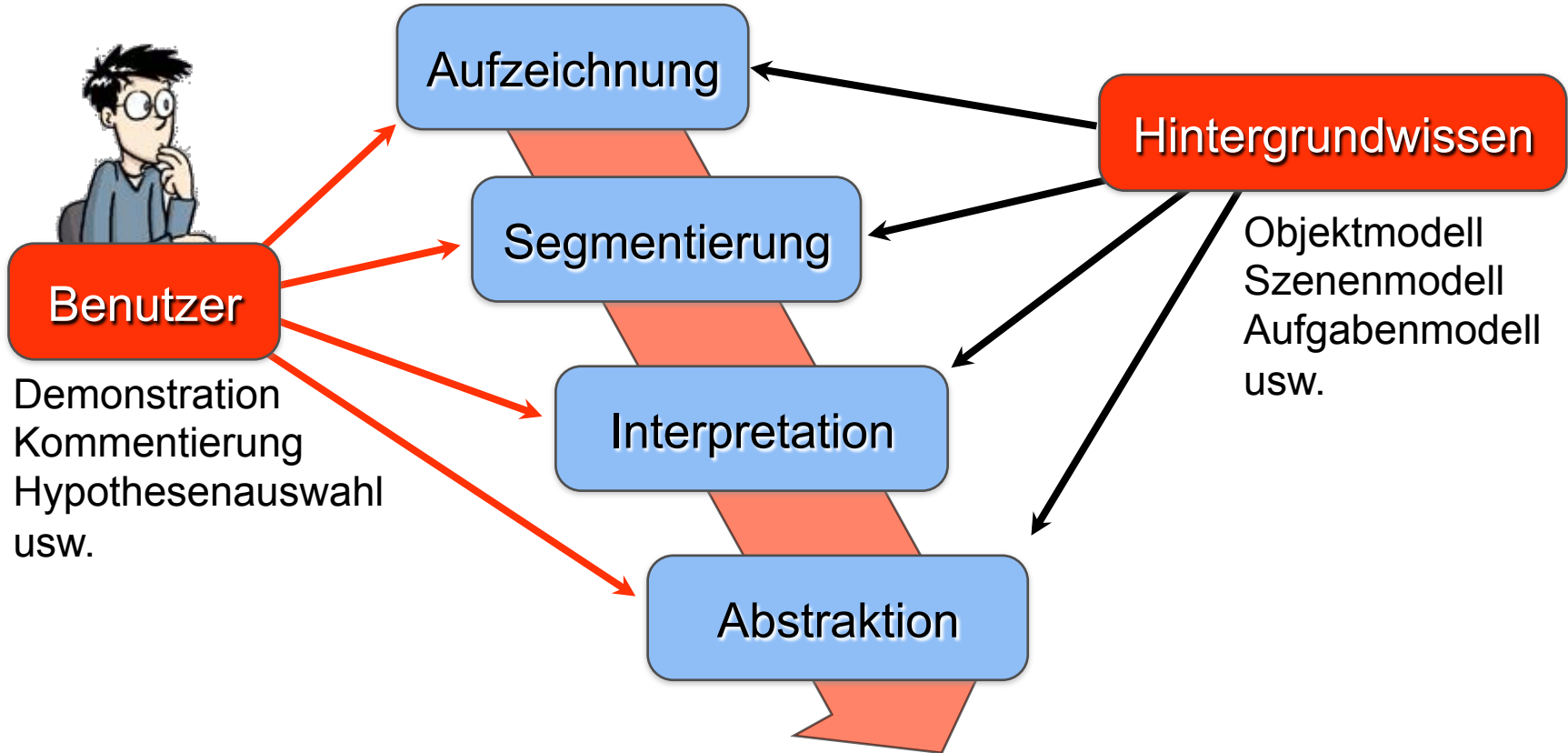




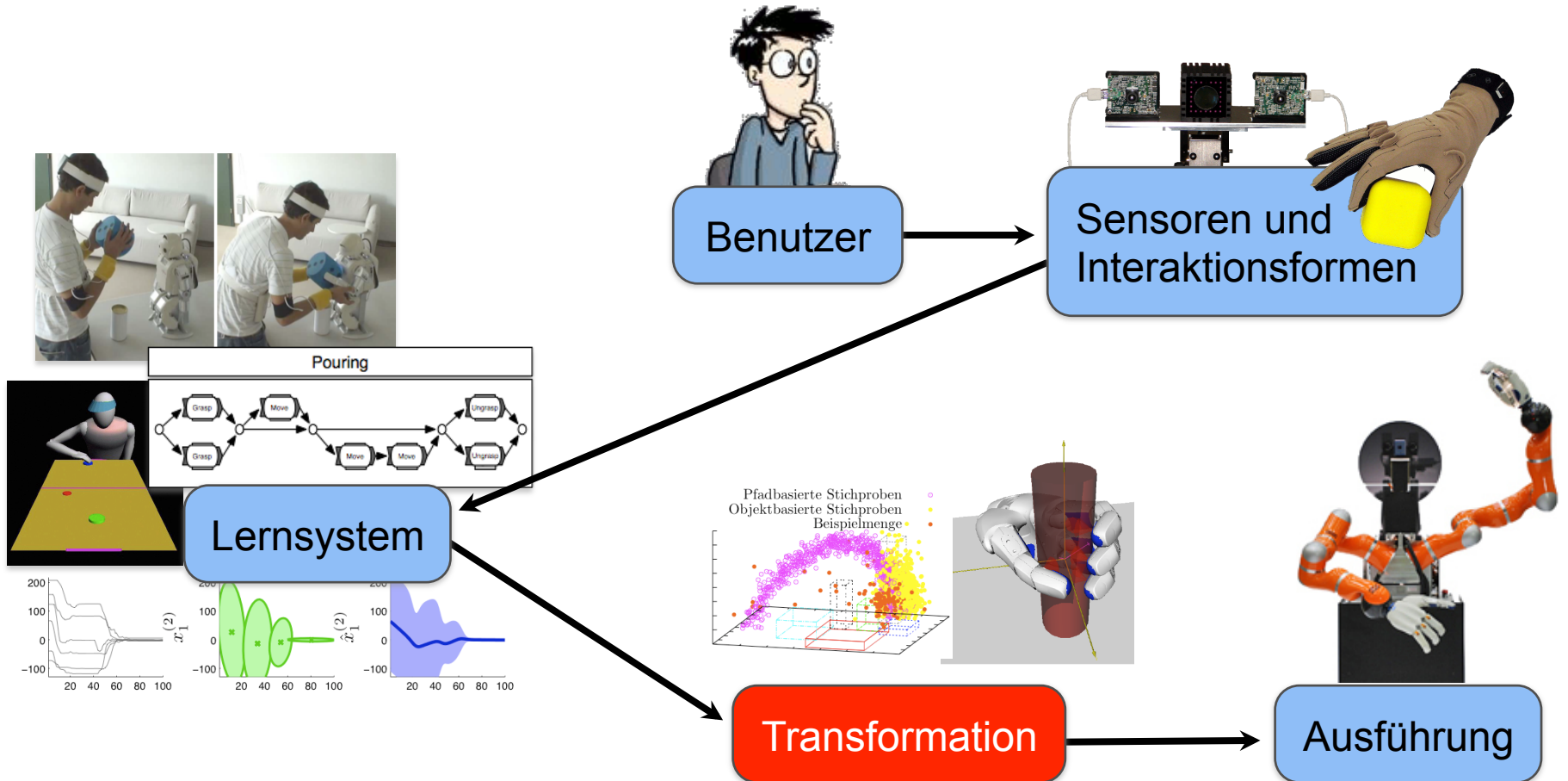
## Wissensrepräsentationen

- Manipulatorabhängige Repräsentation durch Angabe von
  - Aktionssequenz oder
  - Gelenkwinkel-, Kraft und Momenttrajektorien
- 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

## Lernsystem: Prozessschritte



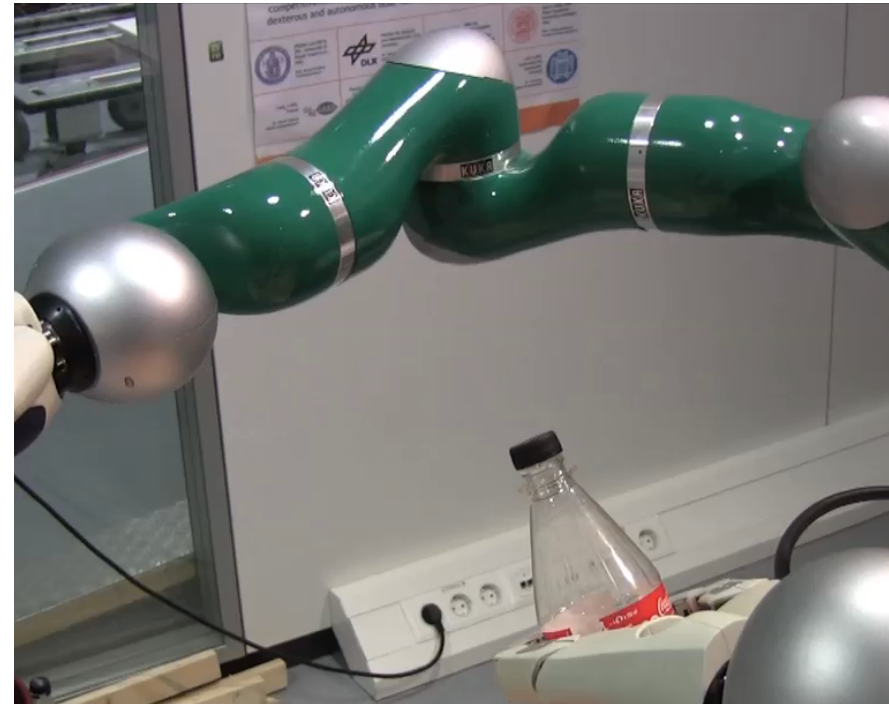
## Komponenten



## 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

Unterschiede in  
Handgröße



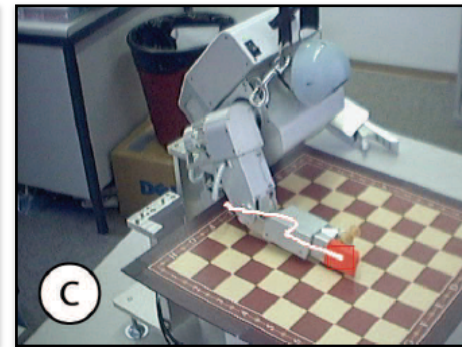
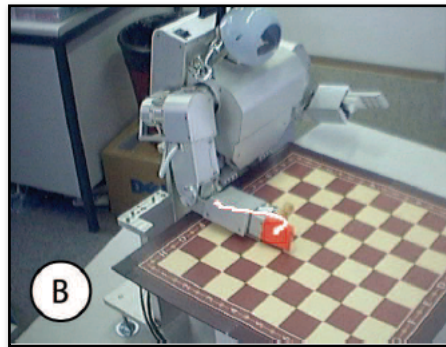
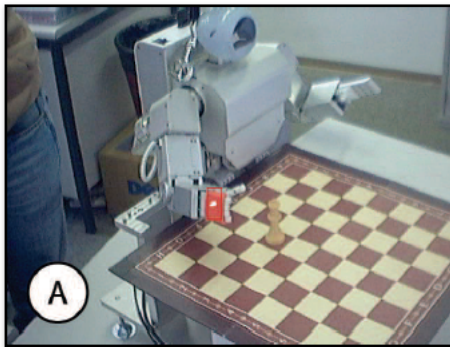
- Motivation
- Grundlagen
- Interaktive Programmierverfahren
  - Probabilistisch
  - Dynamikbasiert
  - Planungsbasiert

- 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)  
→ manipulatorabhängig
  - Direkte Ausführung



## Lerndaten

- $(\theta, x, y, h)$ : n Demonstrationen mit je T Trajektoriepunkten
- $\theta$ : 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



## Lernverfahren

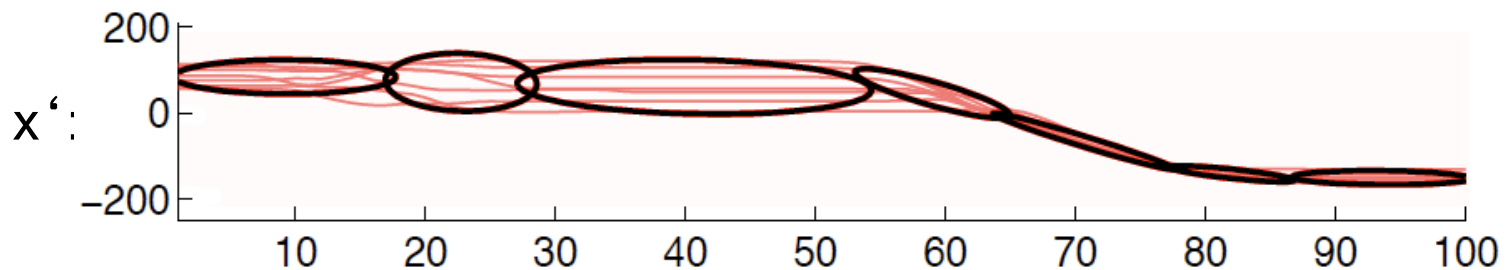
- Dimensionsreduktion, zeitliche Angleichung:  $(\theta, x, y, h) \rightarrow (\theta', x', y', h')$
- Lernen eines GMMs für alle Komponenten mit Dichte  $p(x')$ :

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

$k$  = Anzahl der Normalverteilungen,  $\pi$  = Gewichtung,  $\mathcal{N}$  = Normalverteilung

- Bestimmung von  $k$ :

Bayes'sches Informationskriterium und EM-Algorithmus



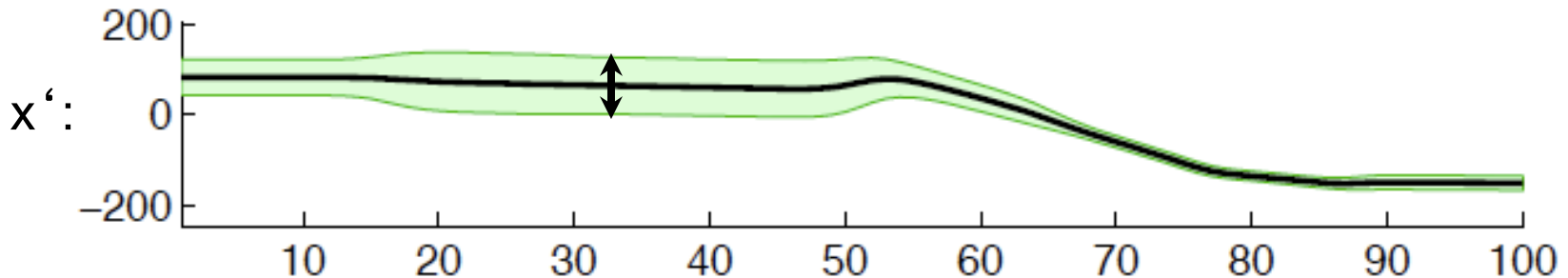


## Repräsentation

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

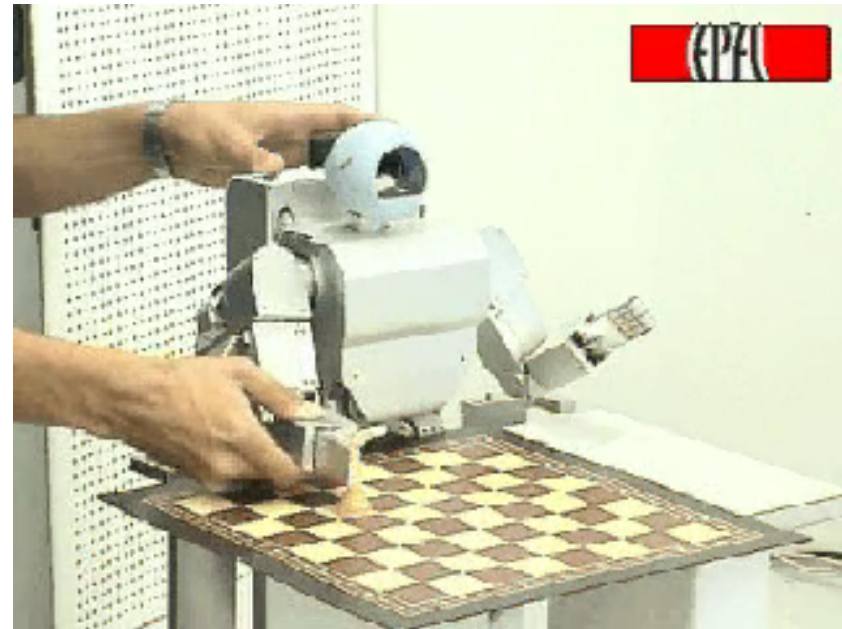
$$\mu_{x'}(t) = \sum_{i=1}^k \beta_i(t) \mu_{i,x'|t} \quad \text{und} \quad \Sigma_{x'}(t) = \sum_{i=1}^k \beta_i(t)^2 \Sigma_{i,x'|t}$$

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



## 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



## Vorteile

- schnelles Verfahren, ähnlich Playbackprogrammierung
- automatische Adaptierung an Änderungen der Objektpositionen

## Nachteile

- relevante Merkmale manuell definiert, hier z.B. nur Distanz zu Startposition
- geringe Generalisierung, da keine Vorbedingungen, Ziele, Kollisionen  
→ keine zielgerichtete Erzeugung von Bewegungen
- keine Validierung

- Motivation
- Grundlagen
- Interaktive Programmierverfahren
  - Probabilistisch
  - Dynamikbasiert
  - Planungsbasiert

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



## Repräsentation

- Implizite Darstellung durch Menge von Differentialgleichungen:

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

$$\tau \dot{x} = v$$

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

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

Vorteil: Gewichte hängen nicht von  $\tau$ ,  $x_0$  und  $g$  ab

- Änderungen von Start, Ziel und der zeitlichen Skalierung möglich
- Generalisierung eingeschränkt möglich

## Lernen

- Berechnung von  $v(t), \dot{v}(t)$  für jede Demonstration  $x(t)$
- $s(t)$  wird durch Integration berechnet  $\tau \dot{v} = K(g - x) - Dv + (g - x_0)f$
- Der Wert  $f(s)$  wird berechnet 
$$f(s) = \frac{\sum_{i=1}^n w_i \psi_i(s) s}{\sum_{i=1}^n \psi_i(s)}$$
- $\psi_i$  sind nicht normalisierte Normalverteilungen („Gauss‘ sche Basisfunktionen“)
- Bestimmung der Parameter  $w_i$  durch lineare Regression

## Ausführung

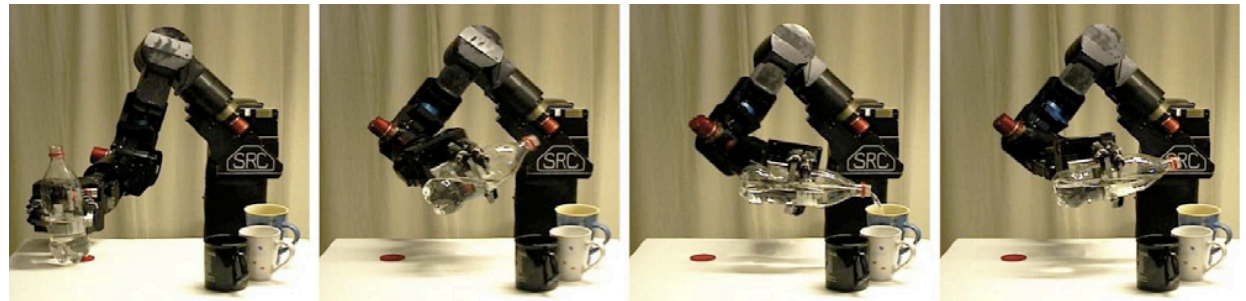
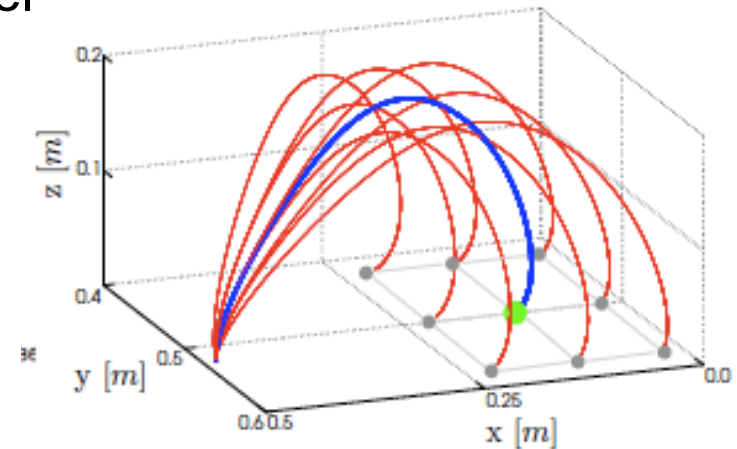
- Berechnung von  $v(t), \dot{v}(t)$  im aktuellen Zustand und Integration

## Vorteile

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

## Nachteile

- relevante Merkmale manuell definiert
- geringe Generalisierung
- keine Validierung





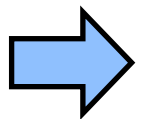
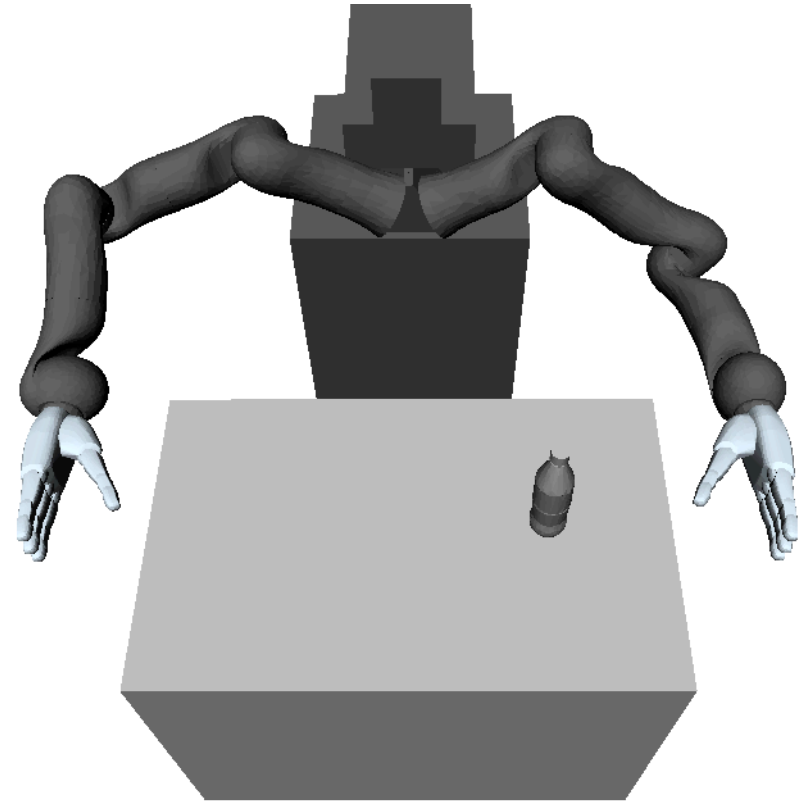
- Motivation
- Grundlagen
- Interaktive Programmierverfahren
  - Probabilistisch
  - Dynamikbasiert
  - Planungsbasiert

## 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.  $\geq 40$  dofs)

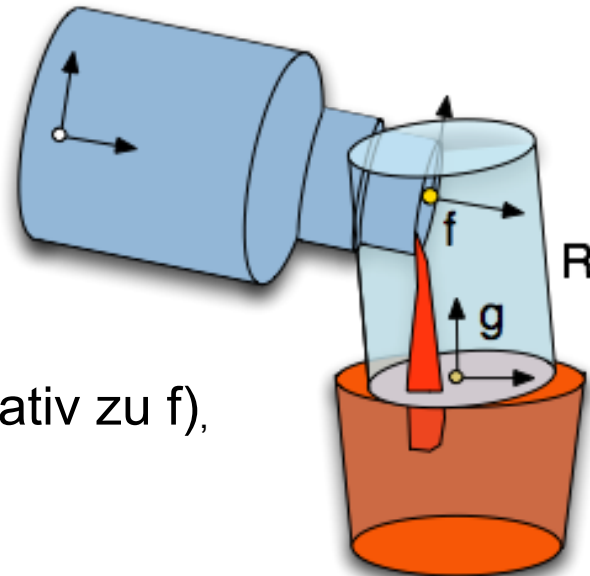


Lernen von Planungsproblemen aus der Beobachtung des Menschen

## Repräsentation: Grundlage Bewegungseinschränkung

- 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  $R$



## Erinnerung: homogene Matrix

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

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

## 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:

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

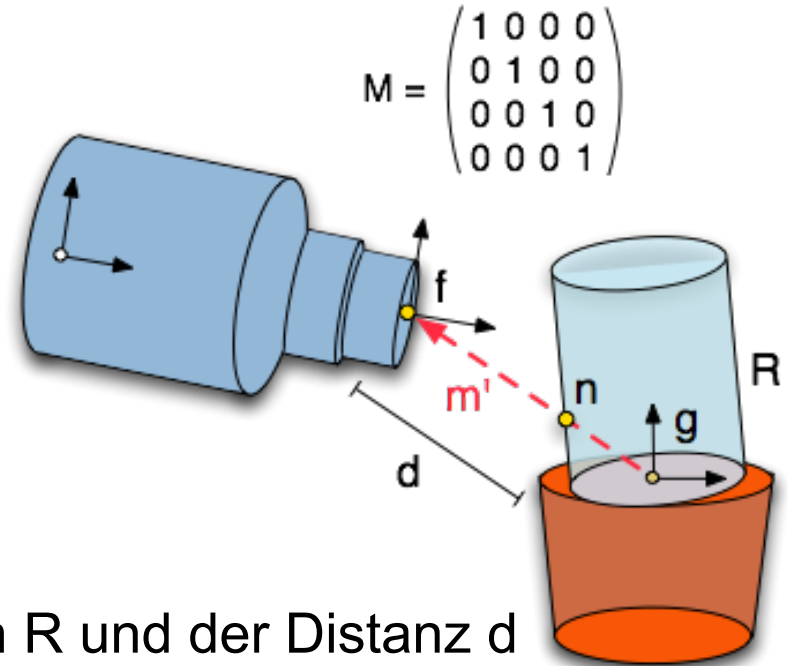
$t$  = Orientierung:

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

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

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

- Erfüllt, wenn  $d < \varepsilon$



## Beispiele für M:

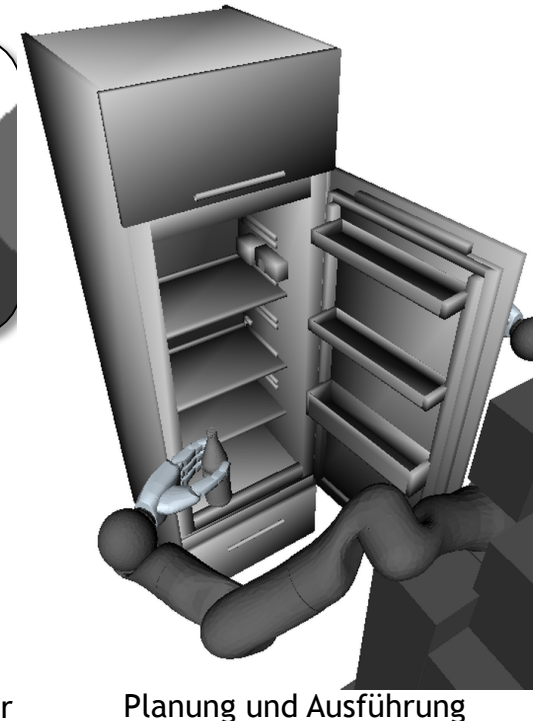
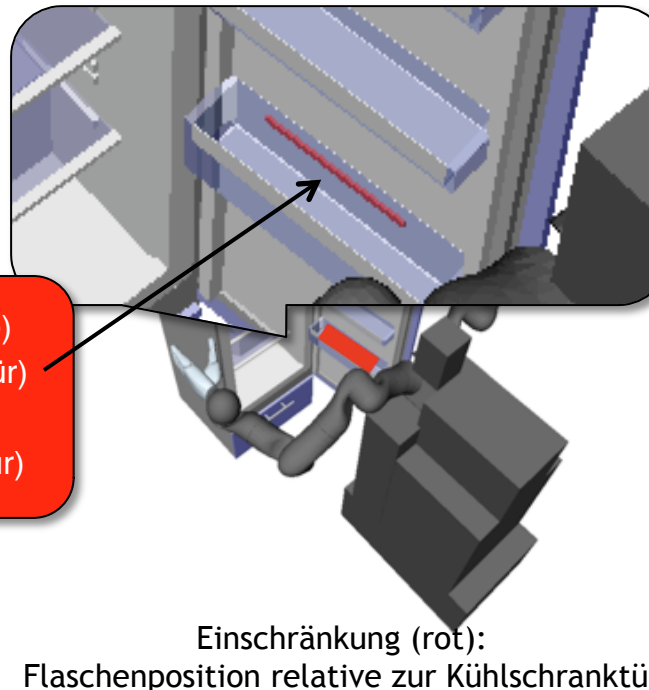
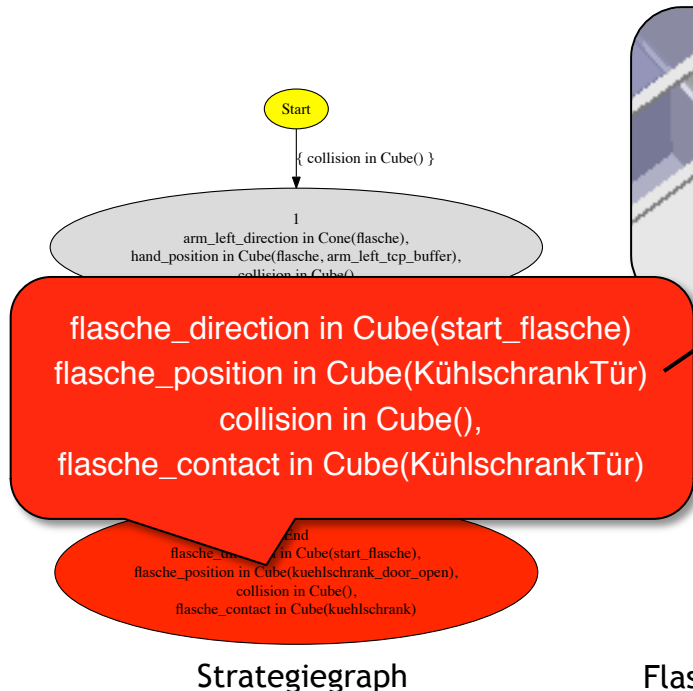
t = Position  $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$  *Position von f*

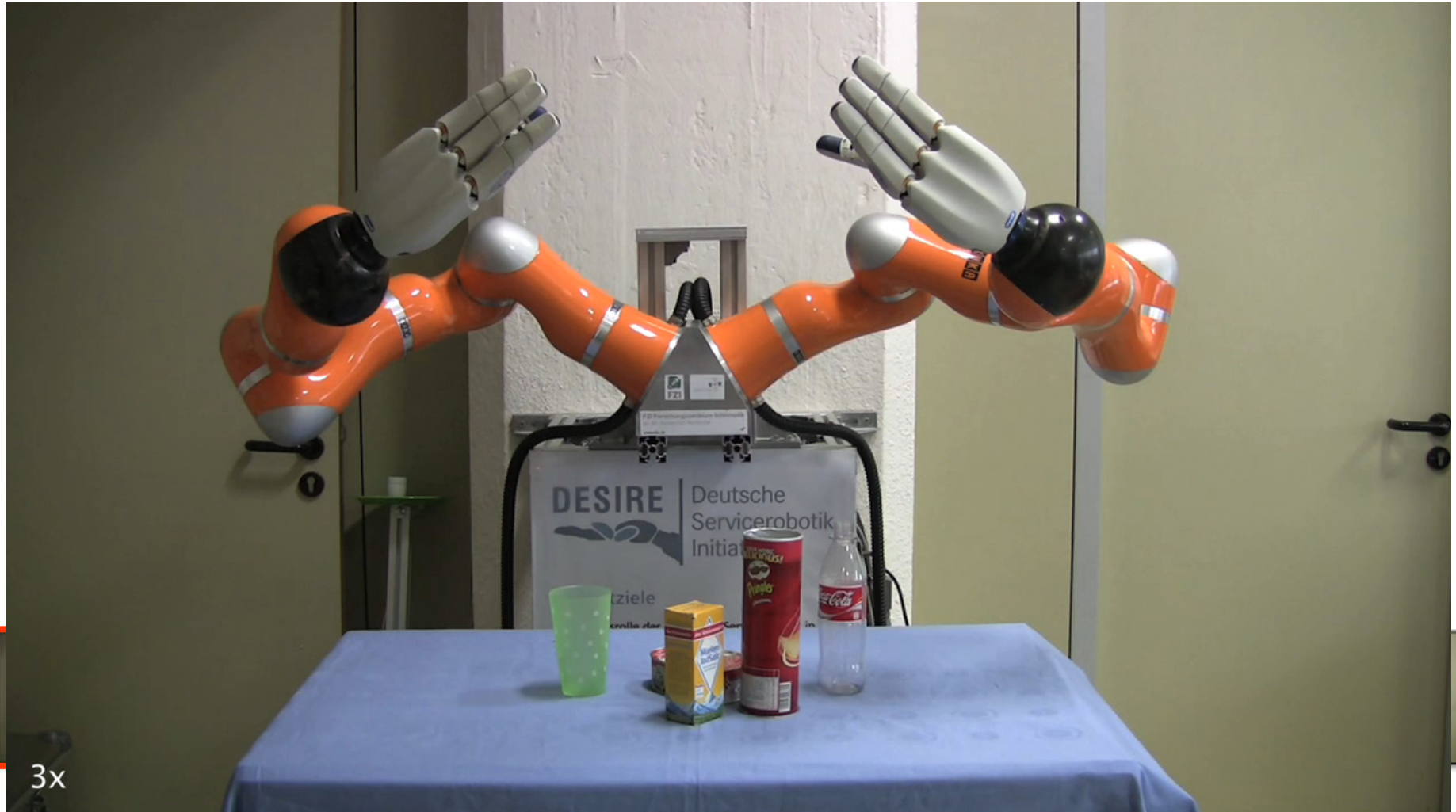
t = Orientierung  $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$  *Orientierung von f*

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

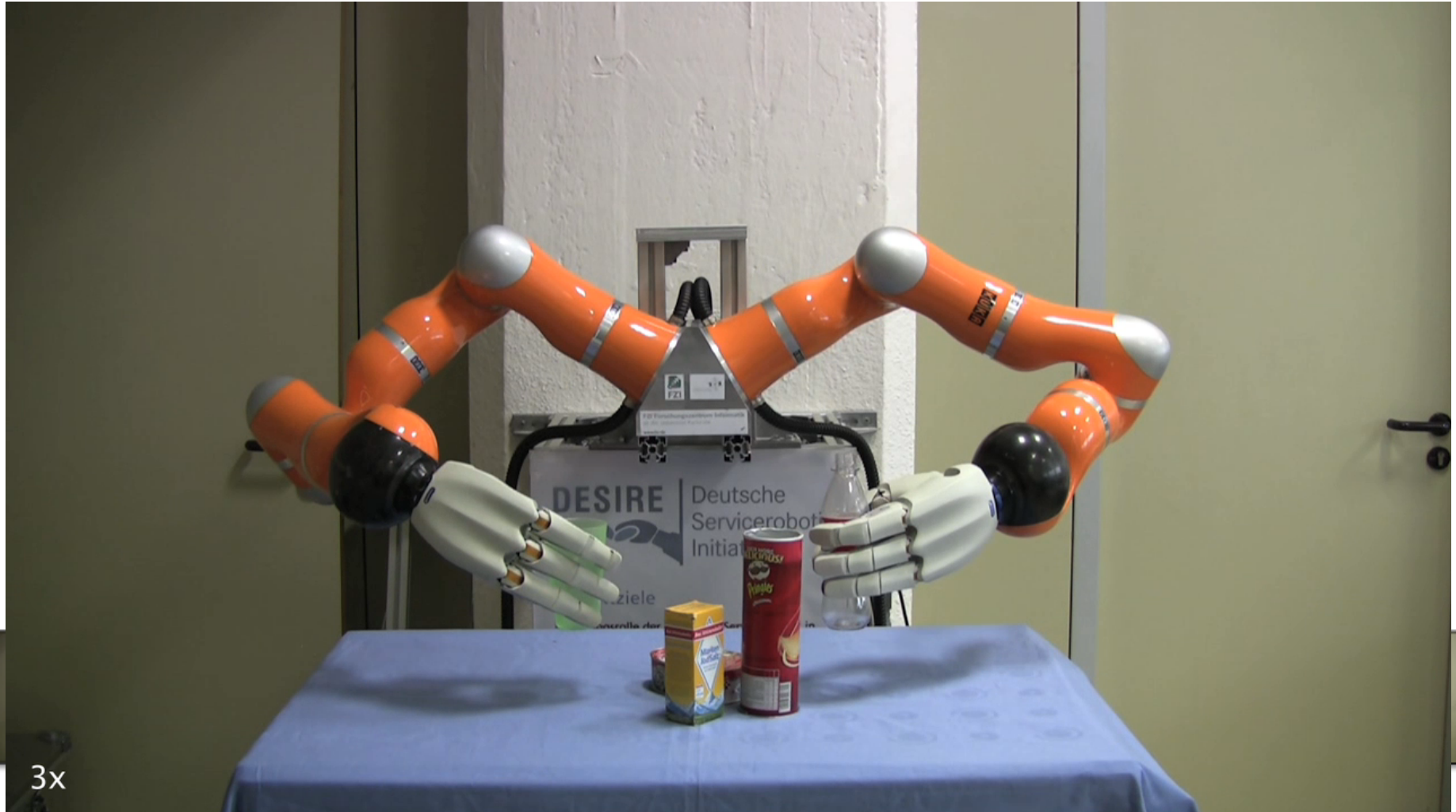
## Repräsentation: Bewegungsplanungsmodell „Strategiegraph“

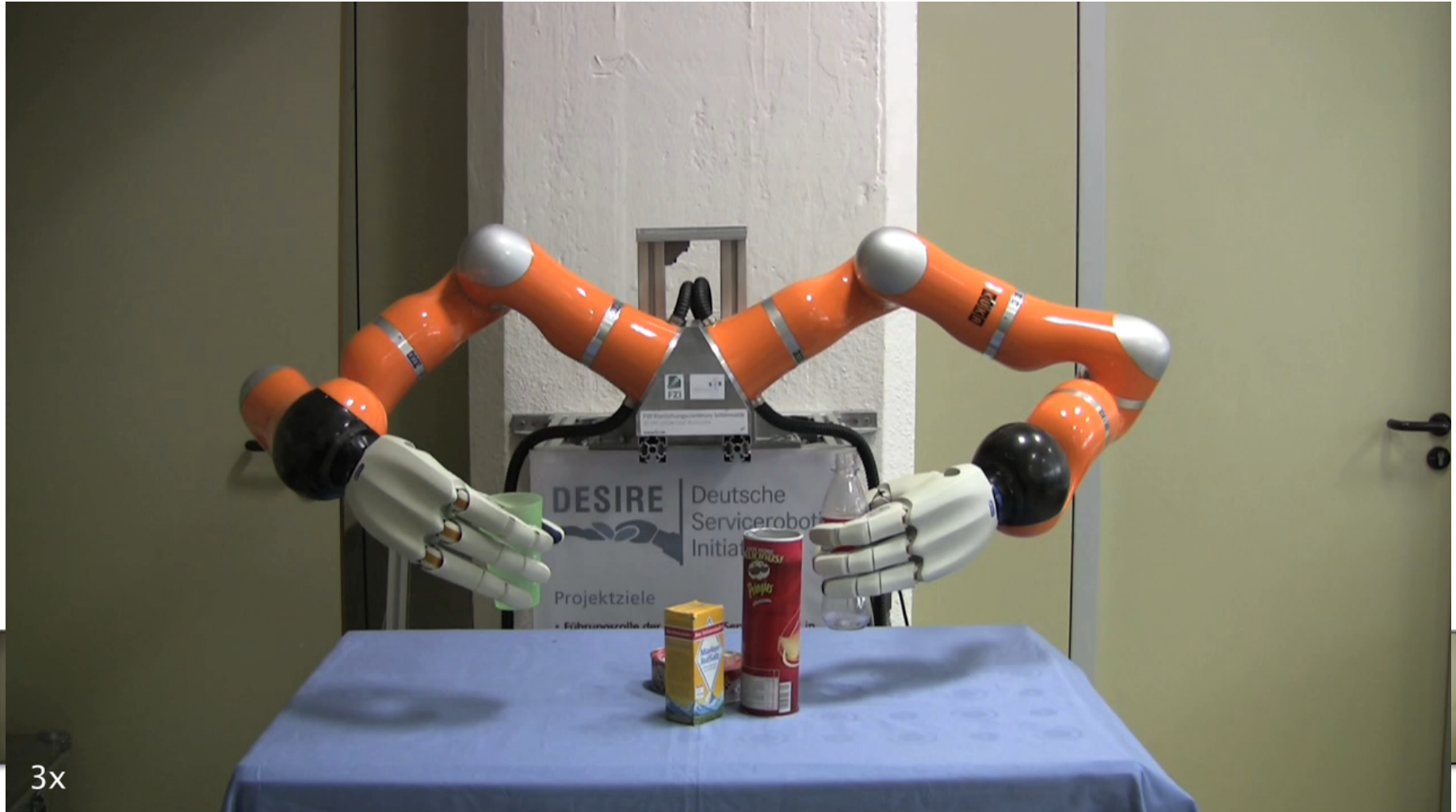
- Knoten = Teilziele, z.B. “wo wird die Flasche in der Kühlschranktür platziert”
- Kanten = Übergänge, z.B. “Flasche aufrecht halten bei der Bewegung”
- Einheitliche Beschreibung durch Einschränkungen

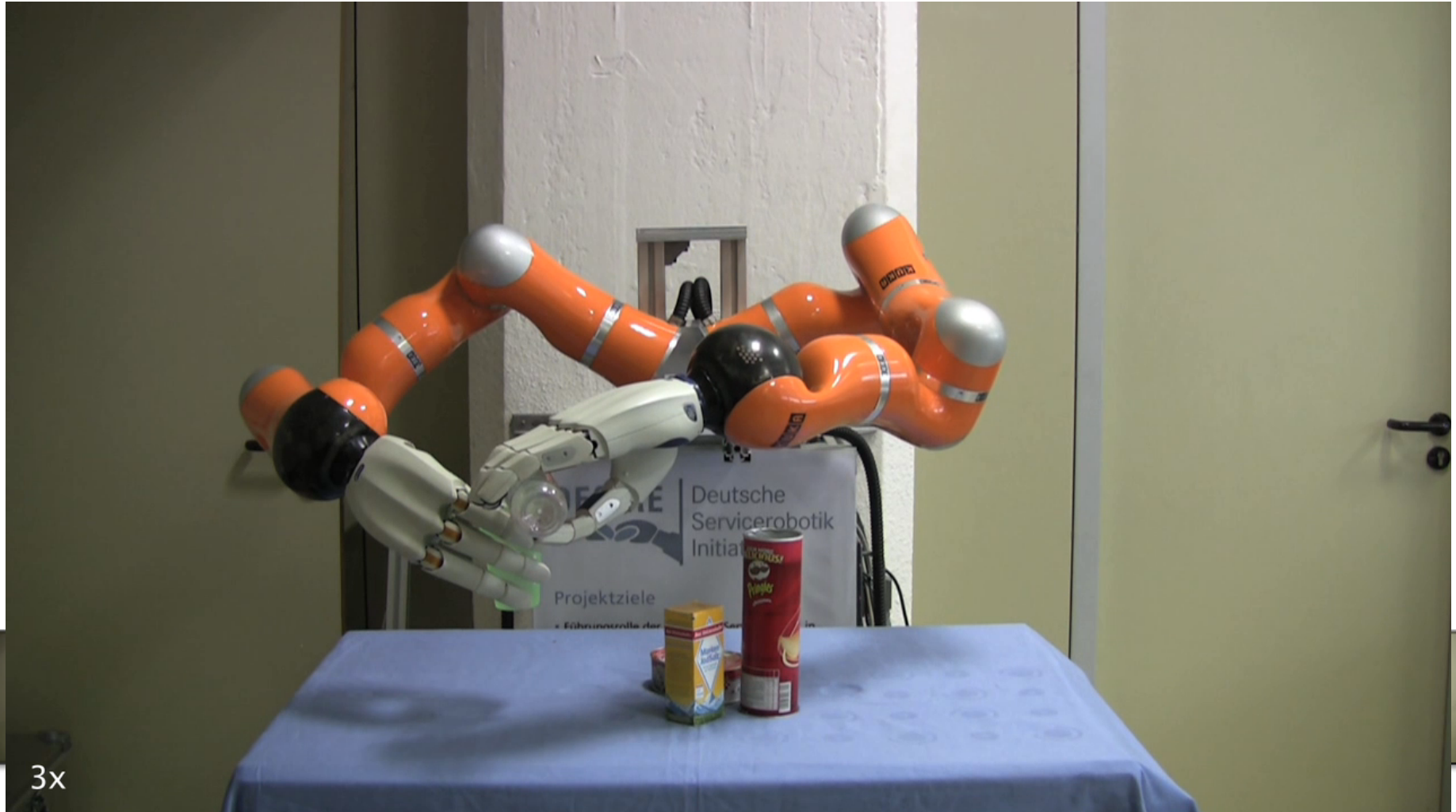


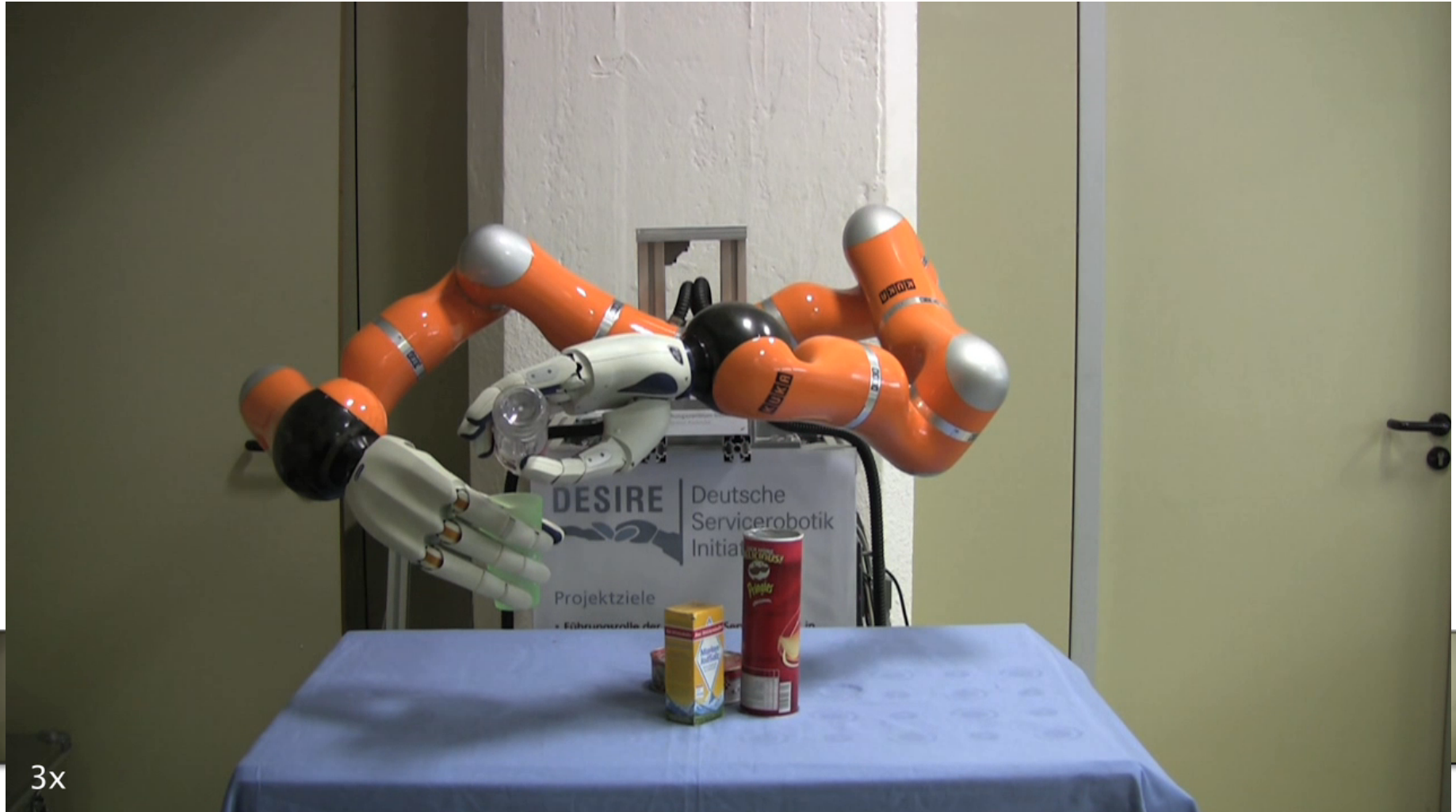


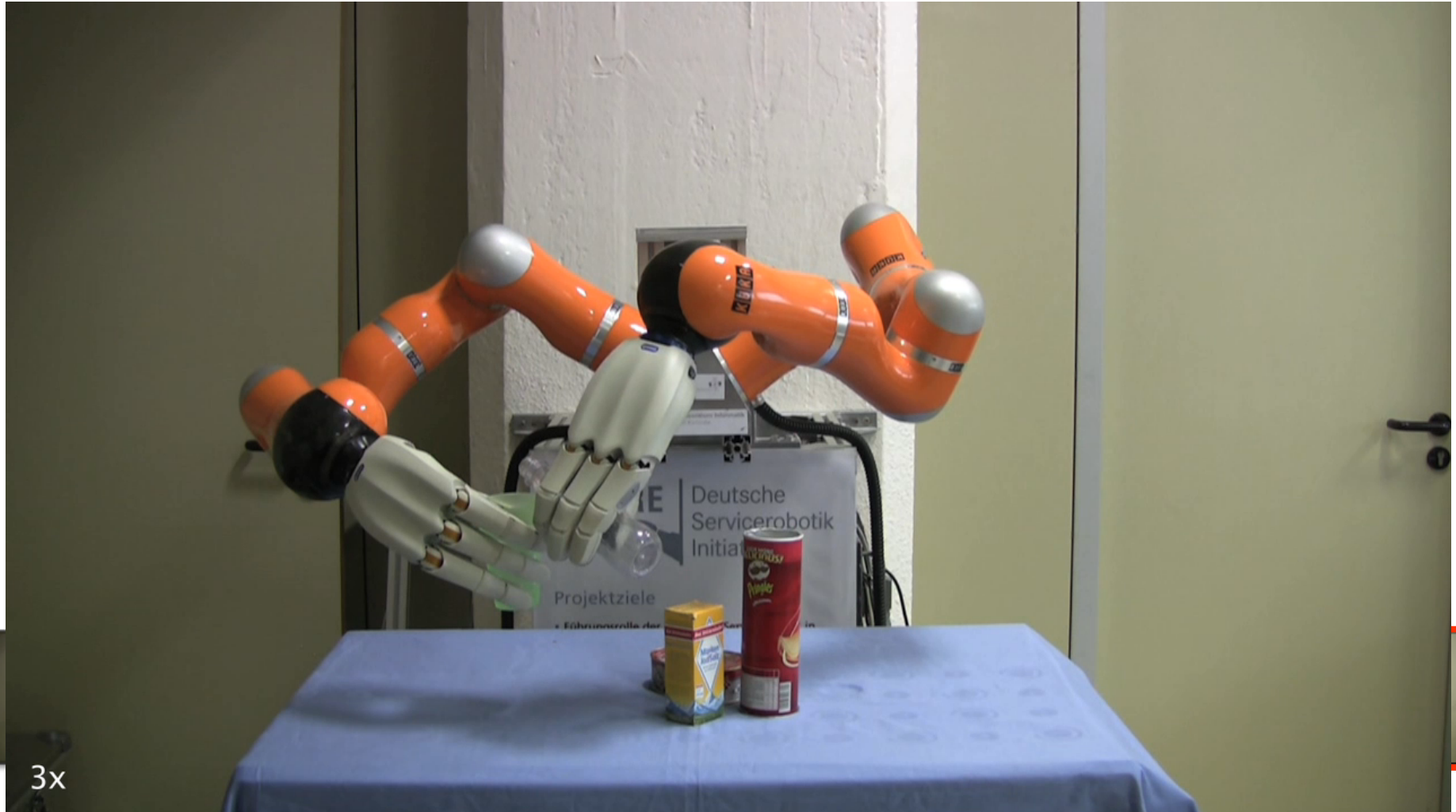




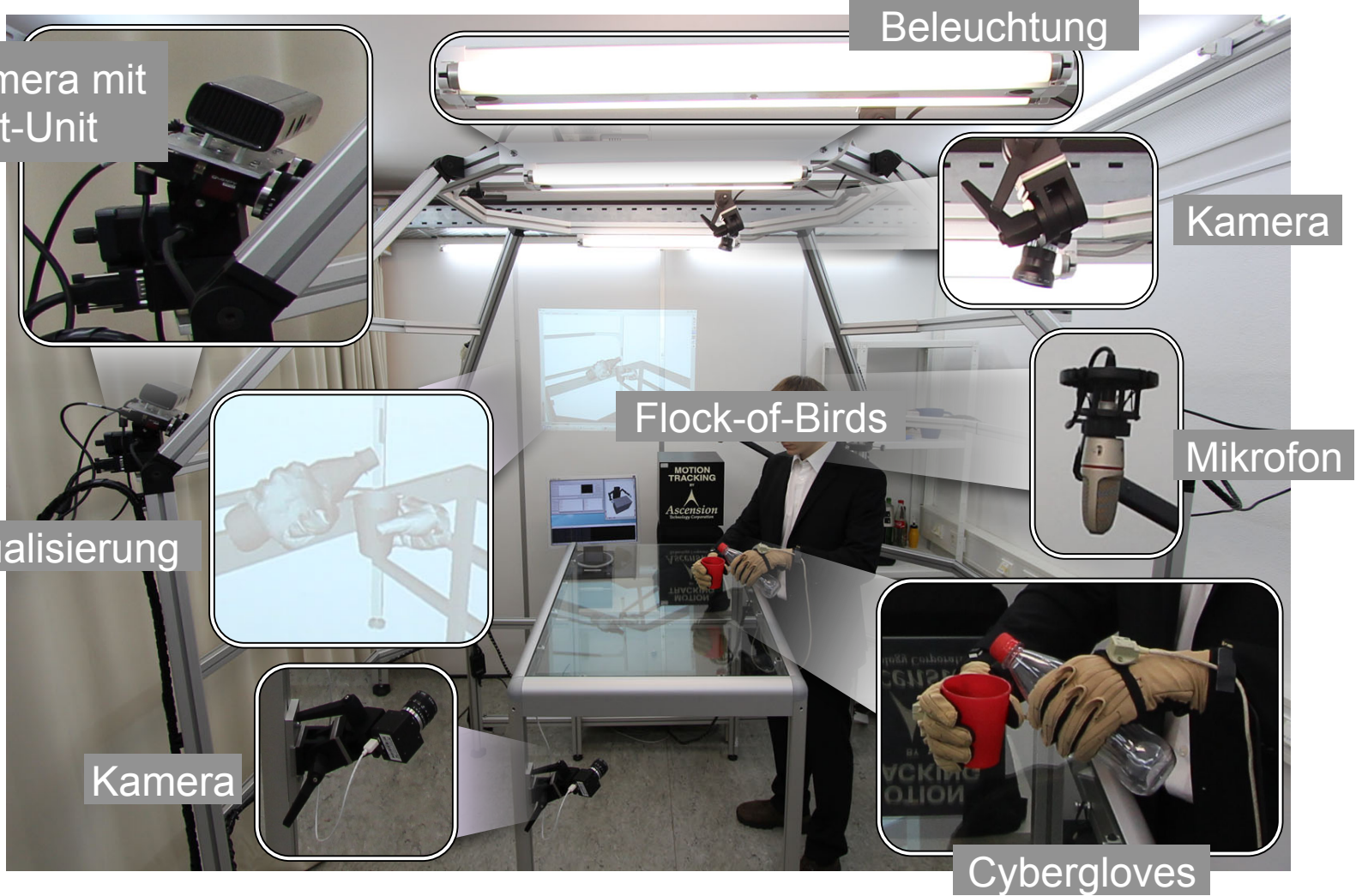






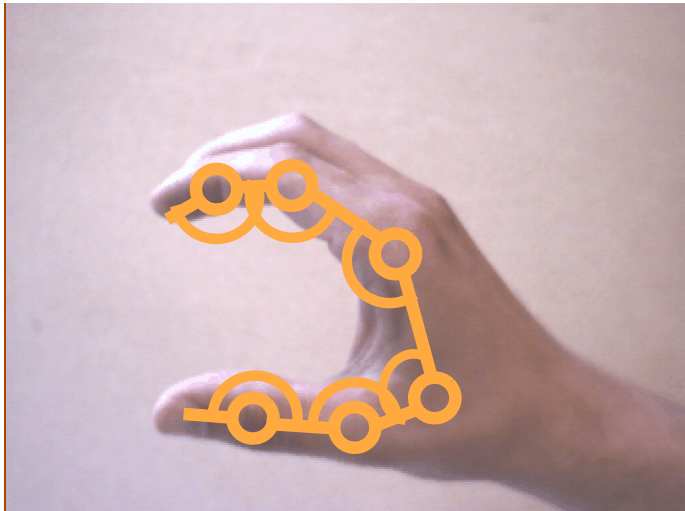
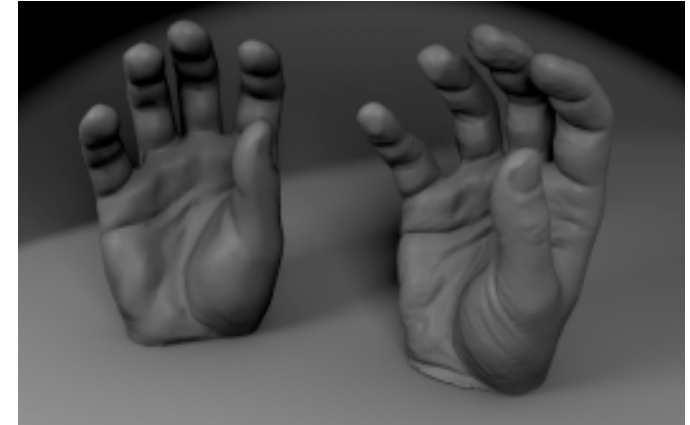


## Demonstration: sensorische Erfassung

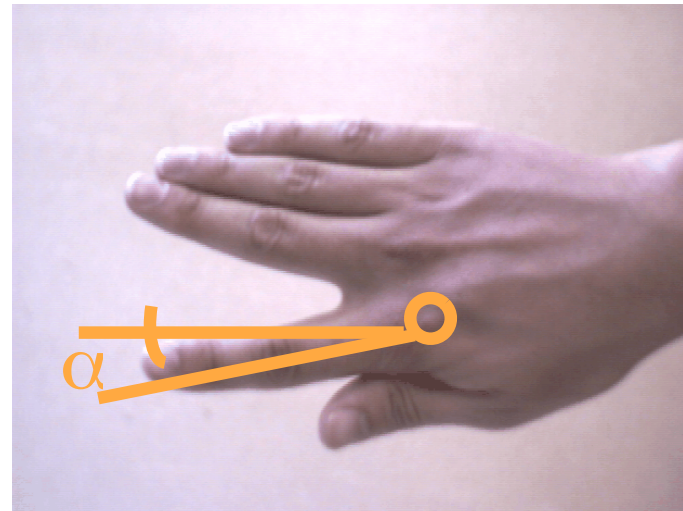


## Demonstration: sensorische Erfassung

- Virtual Technology - Datenhandschuh
- Meßprinzip: Dehnmessstreifen
- 20 Fingerbeugungs- und Spreizwinkel + 2 Freiheitsgrade im Handgelenk



Beugungswinkel



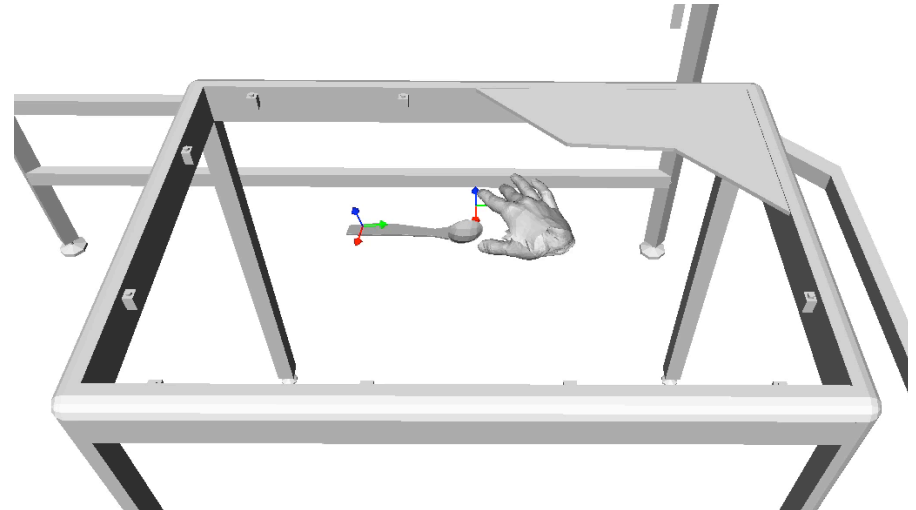
Spreizungswinkel

## Demonstration: sensorische Erfassung

- Bestimmung der Position und Orientierung der menschlichen Hände sowie von Objekten
  - Magnetfeldbasierter Positionstracker
  - Stereokamera



Demonstration

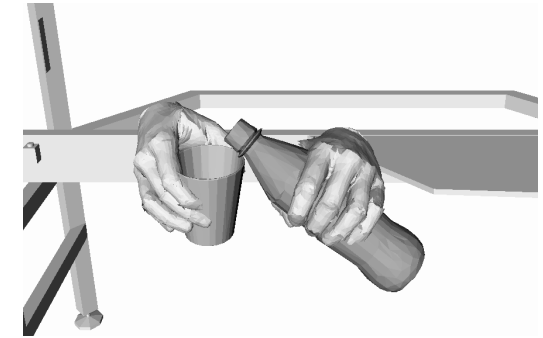


Beobachtung, 3D-Modell der Hand

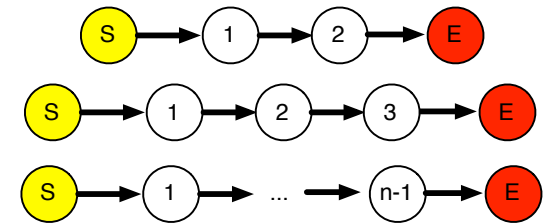


## Segmentierung:

- 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

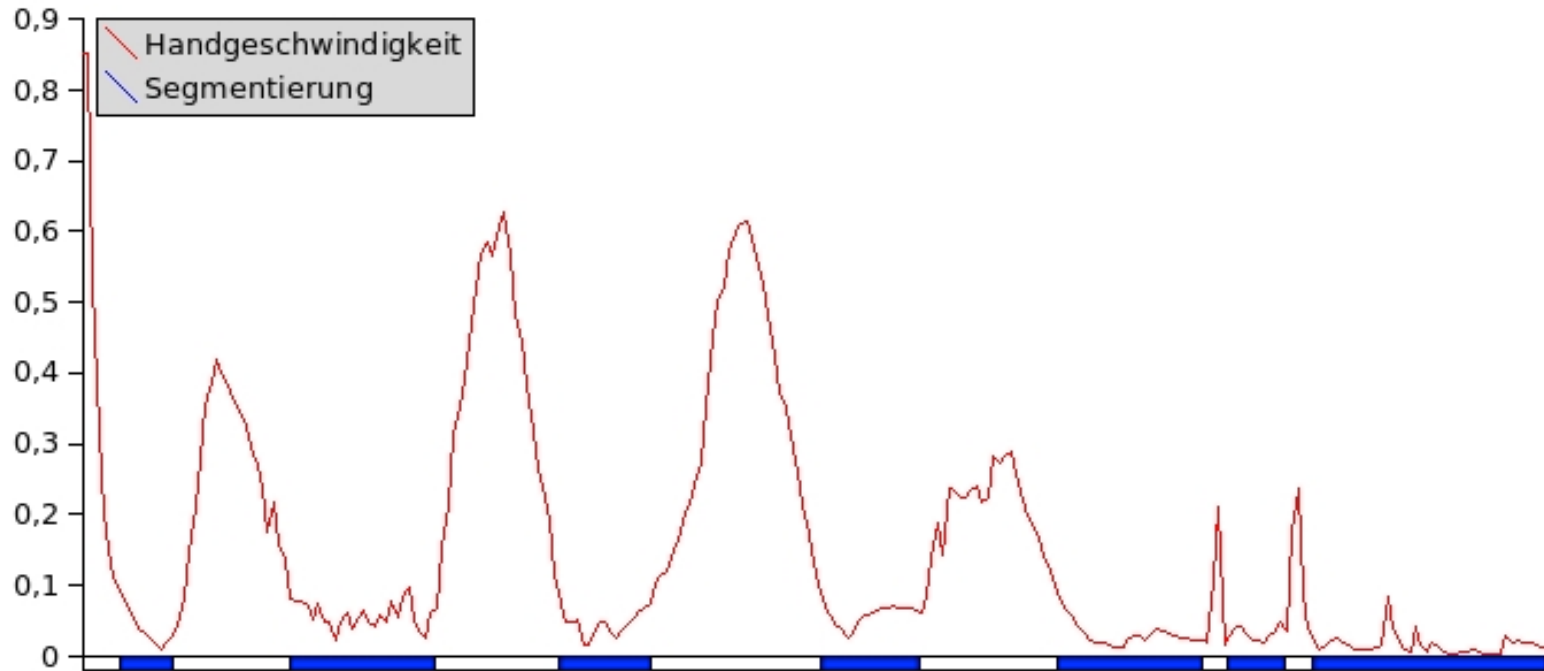


Beobachtete Demonstrationen



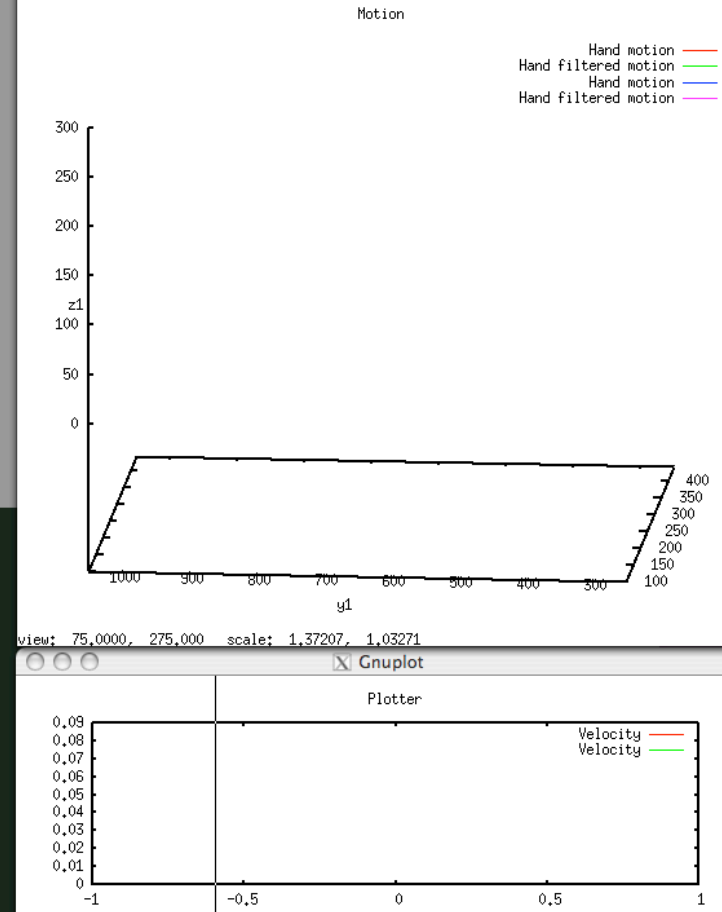
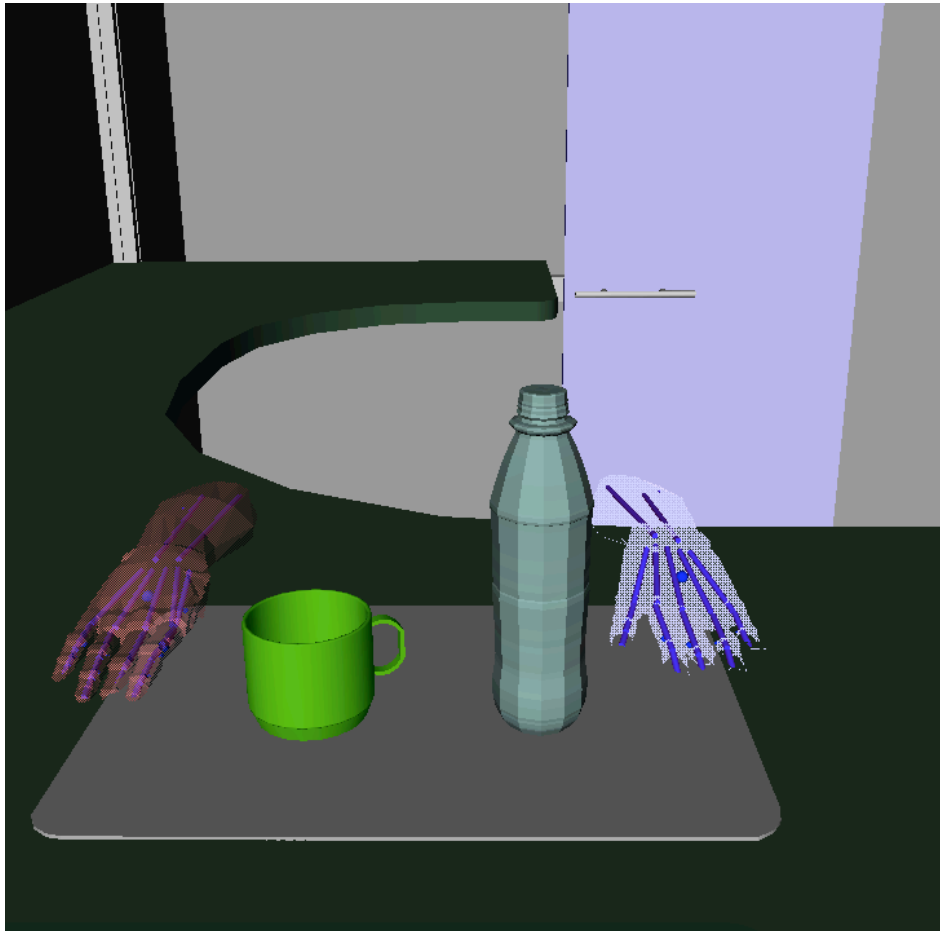
Topologie des Strategiegraphs

## Segmentierung



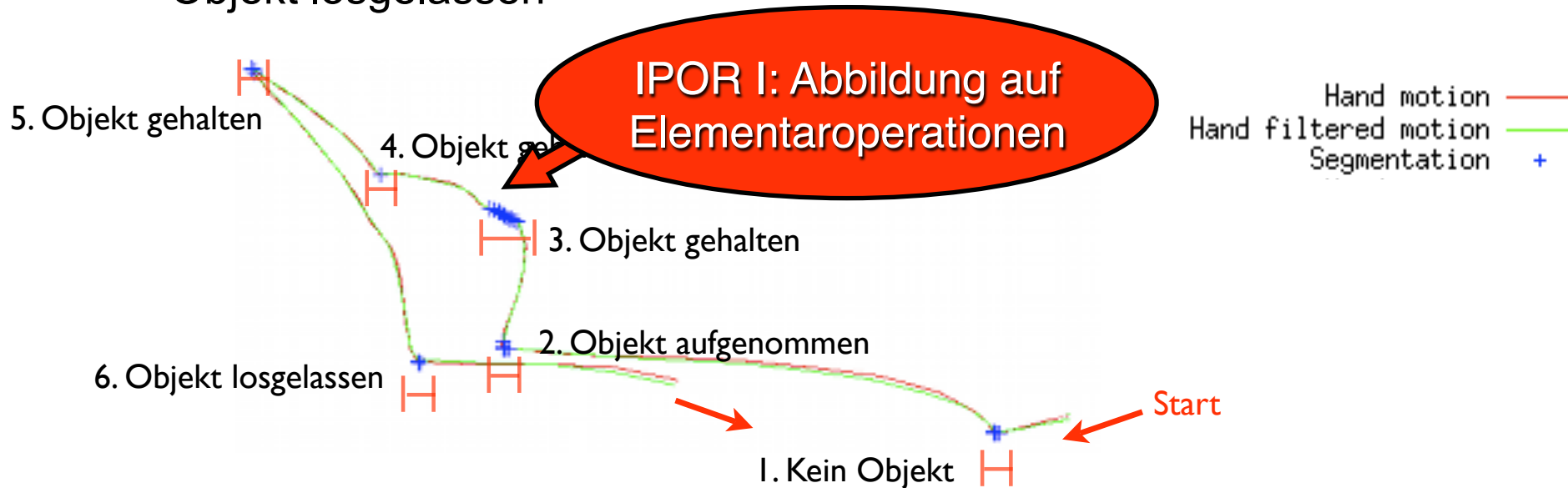
Erzeugung eines Segmentierungspunkt, wenn Hand-, Finger-  
geschwindigkeit gering ist und mindestens ein Finger Objektkontakt hat

## Segmentierung



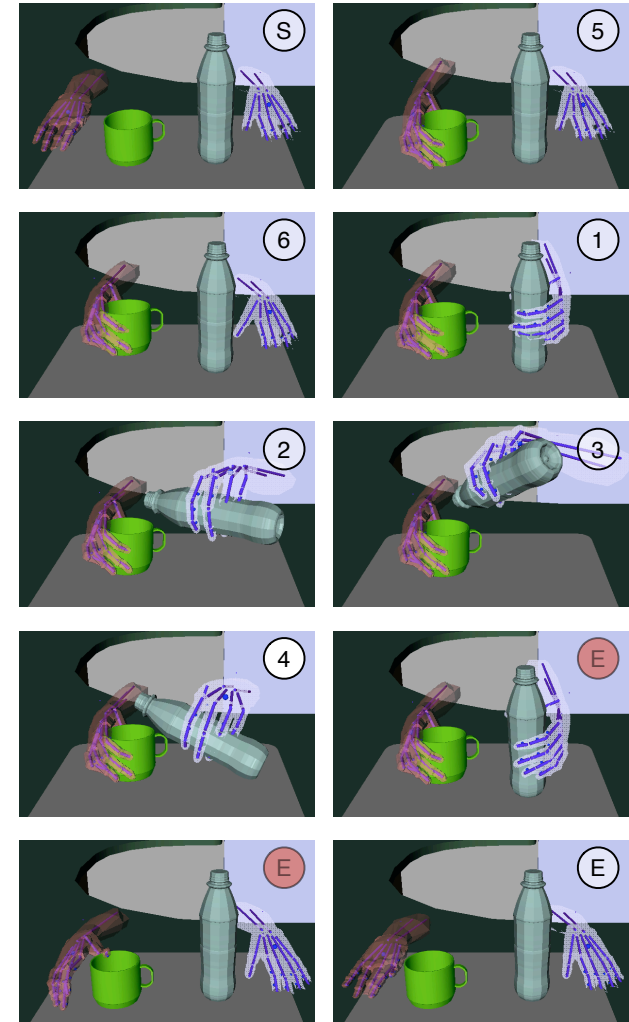
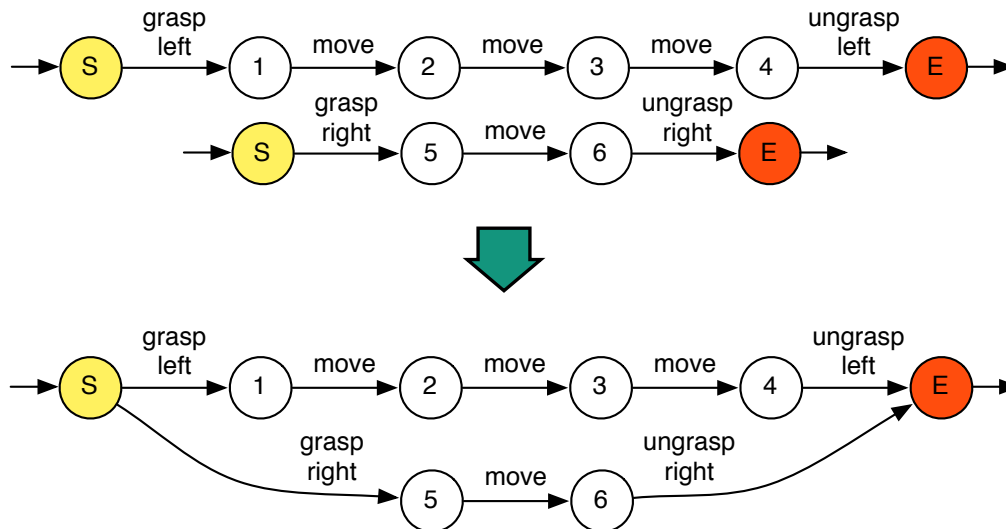
## Interpretation

- Klassifikation der Segmentierungspunkte in 4 Typen auf Basis des Weltzustands an den Intervallgrenzen:
  - Kein Objekt
  - Objekt aufgenommen
  - Objekt gehalten
  - Objekt losgelassen



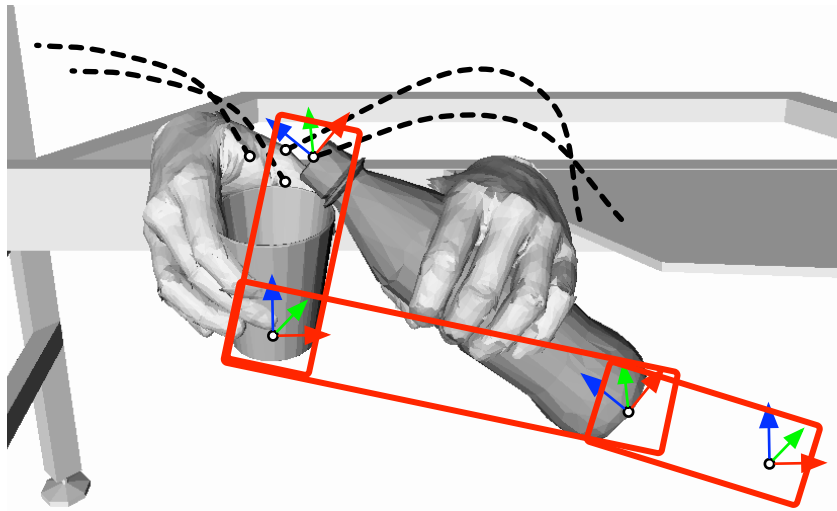
## Interpretation

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



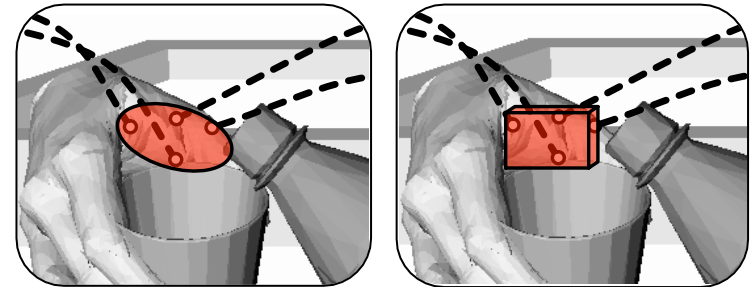
## Interpretation

- Automatische Erzeugung der Einschränkungen *pro Kante und Knoten*
  - Basis: Kombinationen von Koordinatensystemen: Finger, Hände, detektierte Objekte

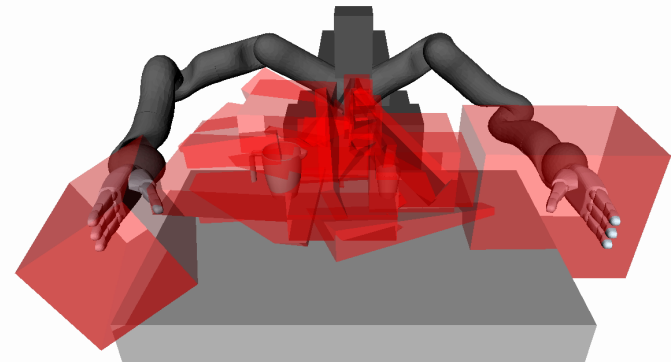


Position: Flaschenöffnung relativ zu Becher  
 Position: Flaschenboden relativ zu Becher  
 Position: Flaschenboden relativ zu Tisch

Mehrere menschliche Demonstrationen



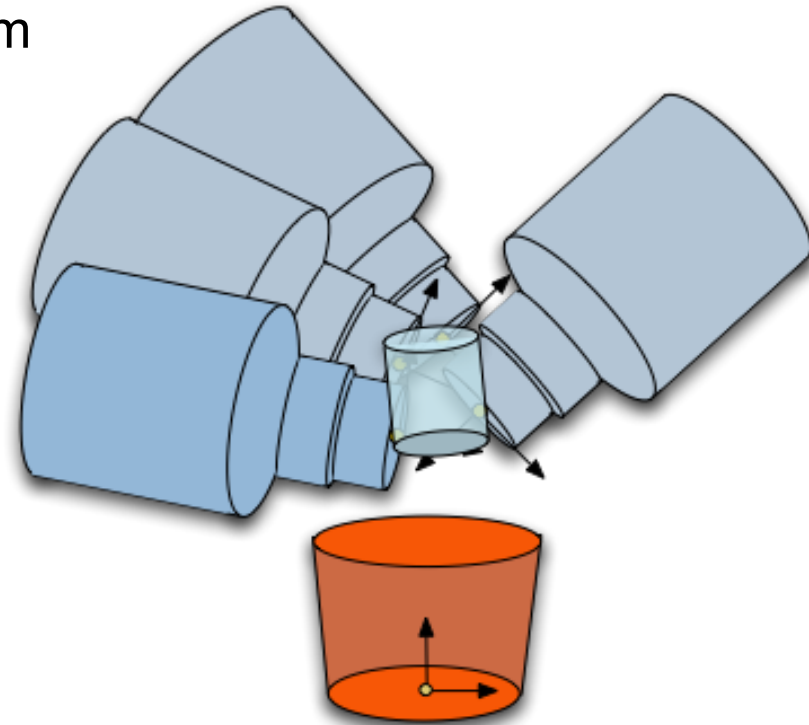
Parameterlernen der Einschränkungen



Einschränkungen für Kombinationen von Koordinatensystemen

## Interpretation

- Parameterlernen:
  - 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 (z.B. Rosenbrock-Optimierung)



Beispiel: Knoten-Einschränkung

f = Flaschenöffnung, g = Becheröffnung, t = Position

## 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: Bahnplanung





## Vorteile:

- Generalisierung auf Basis von Koordinatensystemen
- 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