IntellAct (2011-14): Intelligent observation and execution of **Act**ions and manipulations

SDU (co-ordinator)

MMI

UGOE

UIBK

JSI

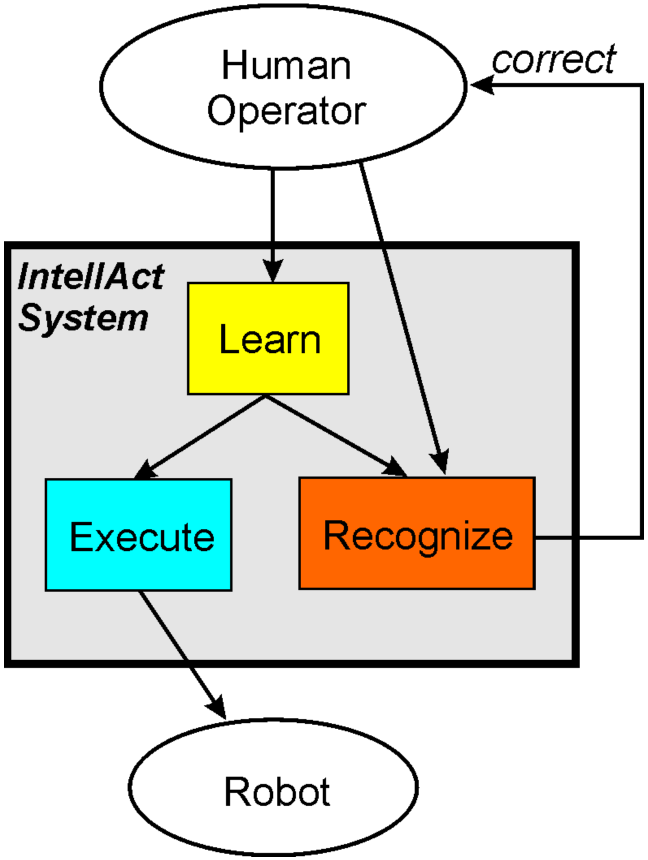
CSIC

SEVENTH FRAMEWORK PROGRAMME, ICT-2009.2.1, COGNITIVE SYSTEMS AND ROBOTICS

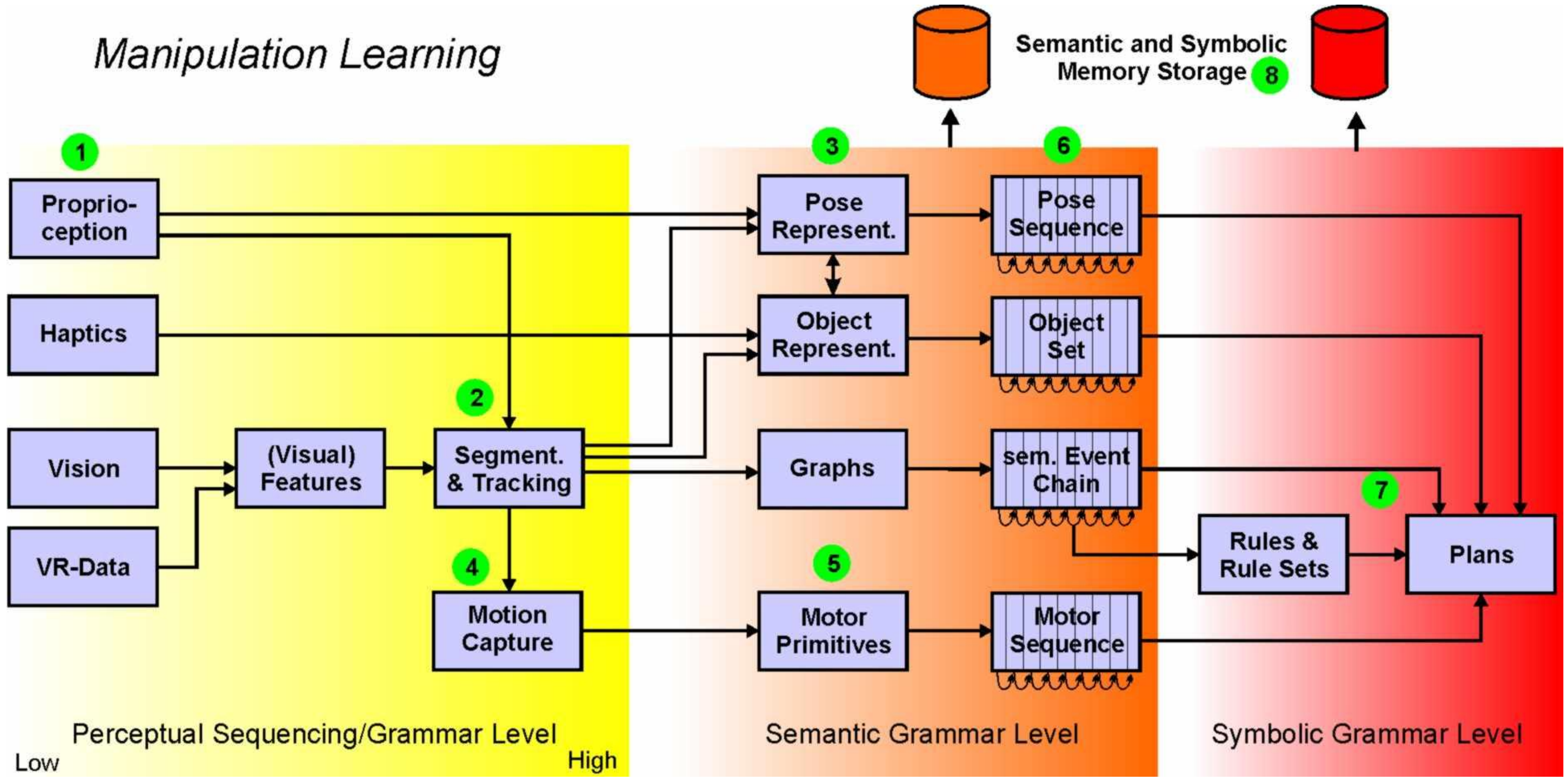
IntellAct Objectives

- Learning to
  - execute robot actions
  - monitor human and robot actions

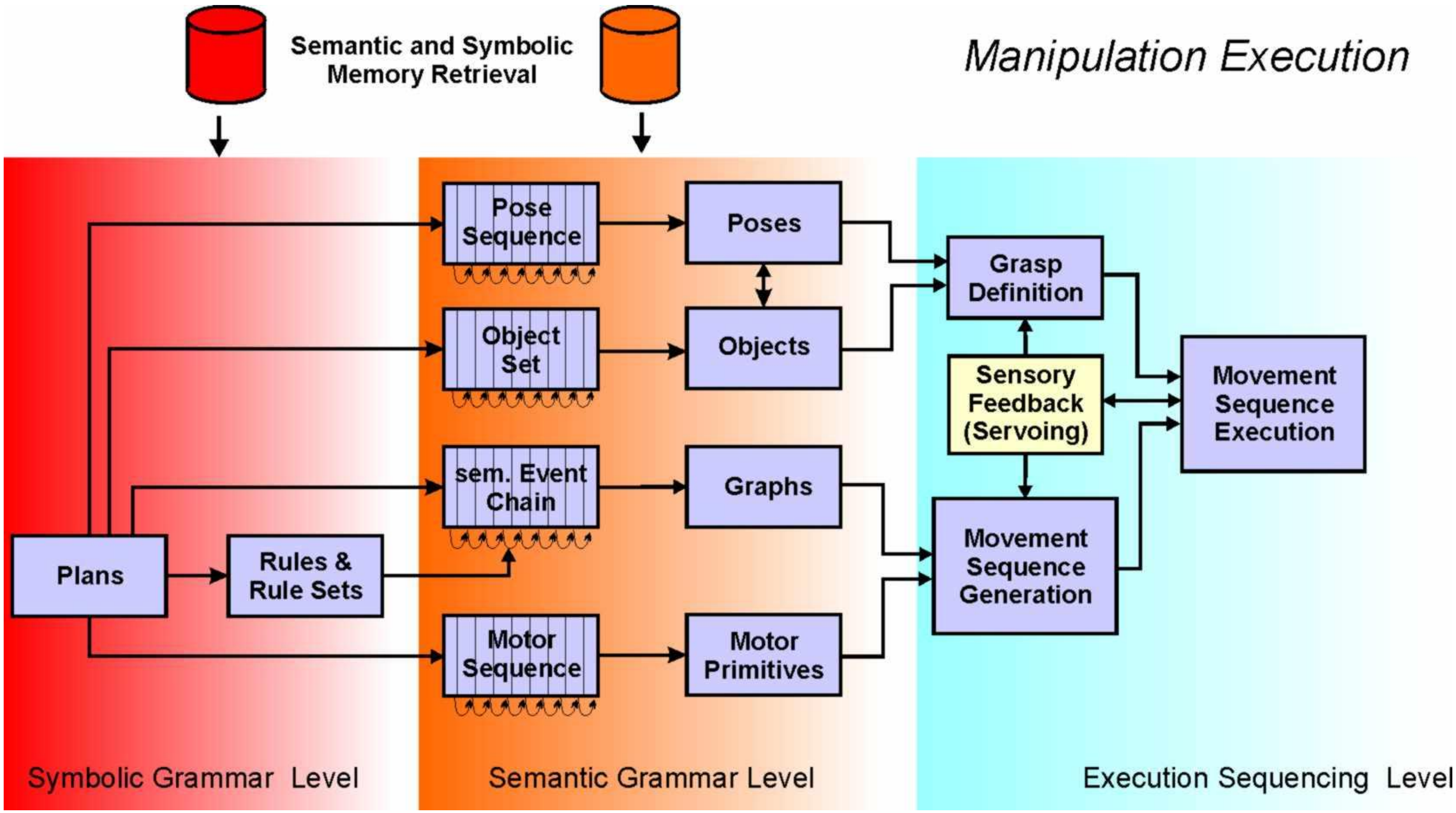
  Programming by Demonstration (PbD)
- Learning takes place on a semantic level not on a trajectory/motor level!
- Requires *scene and action understanding*!



Manipulation Learning

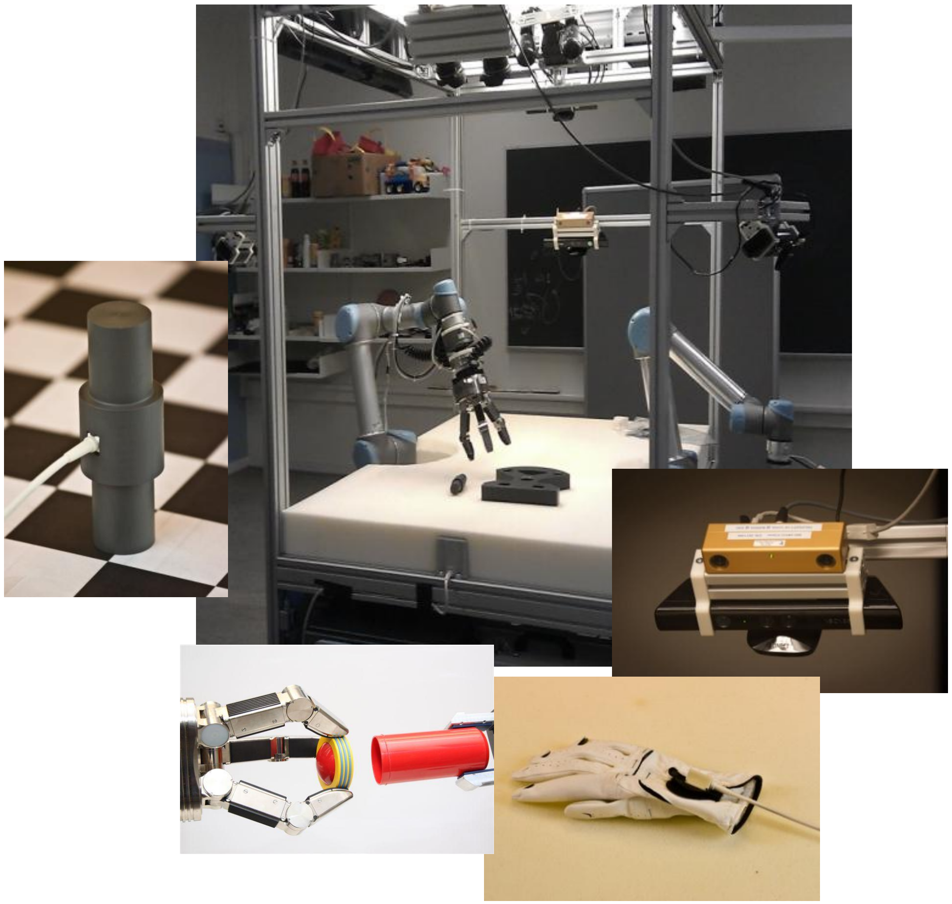


Manipulation Execution

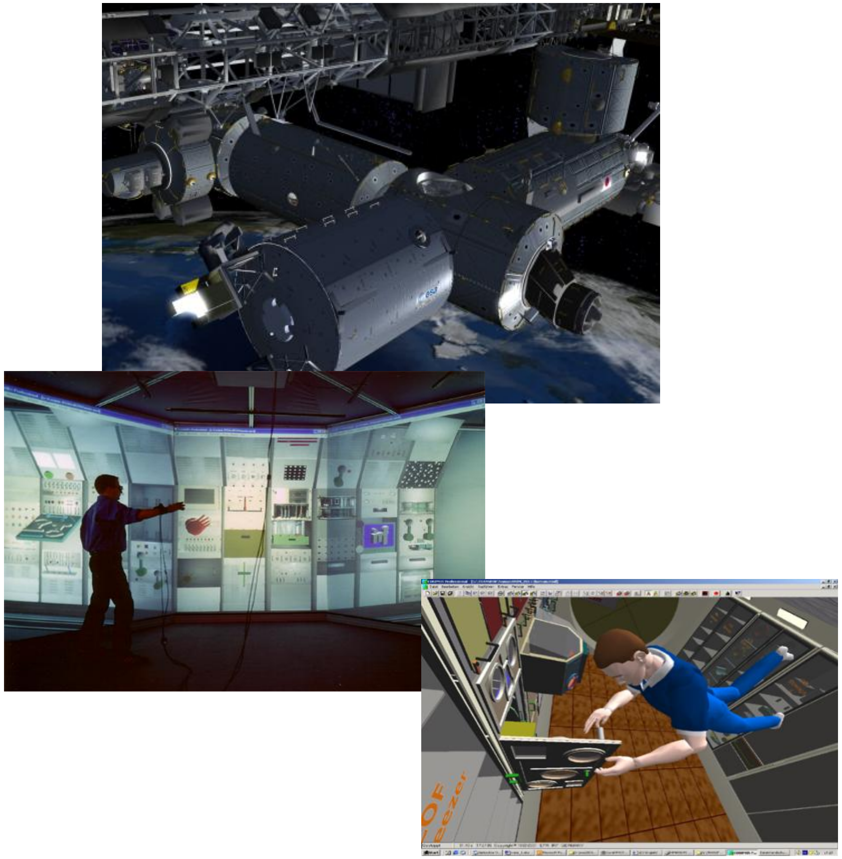


Two Scenarios

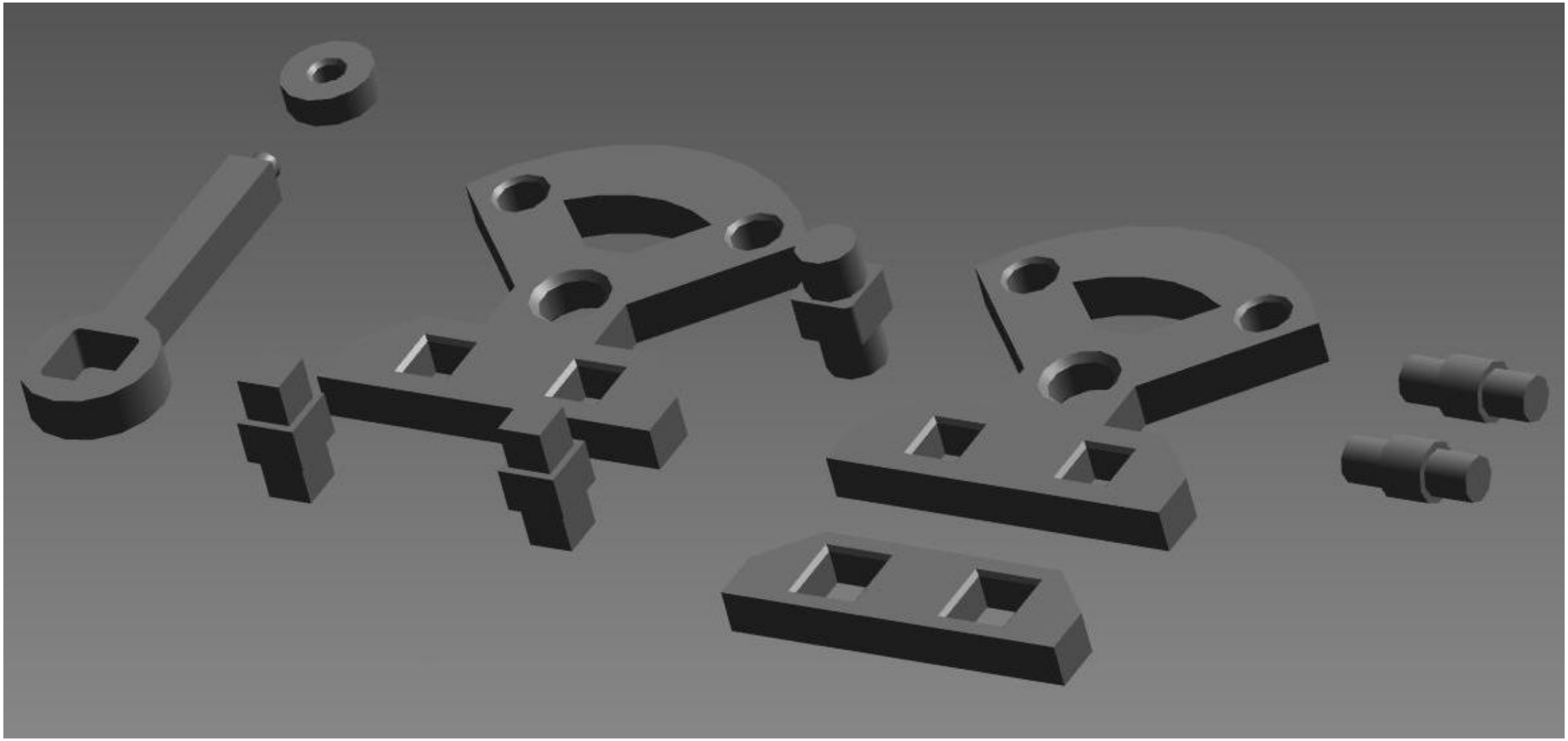
AUTAS



LABEX



Cranfield Benchmark



Key Techniques

KT1 Semantic Event Chains represent observed visual, motor and haptic information (Aksoy et al., 2010).

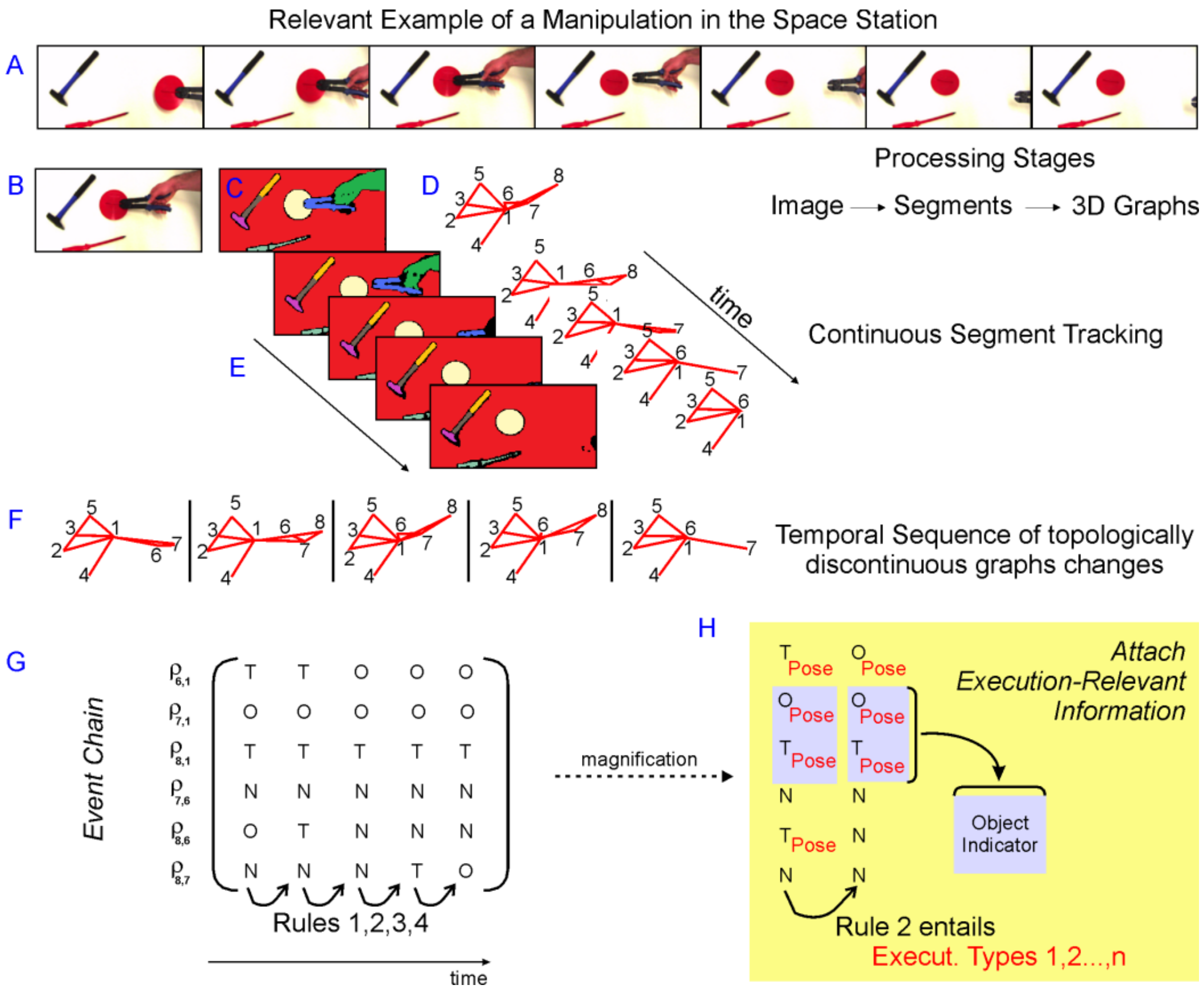
KT2 Probabilistic rule learning mines the semantic Event Chain data base (KT1) to identify prototypical object-action sub-sequences in the data base (Agostini et al. 2010).

KT3 Probabilistic Manipulation Functions represent the full, continuous set of observed manipulation affordances of specific objects (Detry et al., 2010).

KT4 Dynamic motor primitives are used to segment the action space and to translate rules and affordances to motor commands. (Ude et al., 2010; Nemec et al., 2009)

KT5 Virtual Reality provides a sophisticated, interactive, VR-enhanced simulation environment with a 3D visualization cave, available at MMI, provides an effective research environment.

KT1: Semantic Event Chains (SECs)



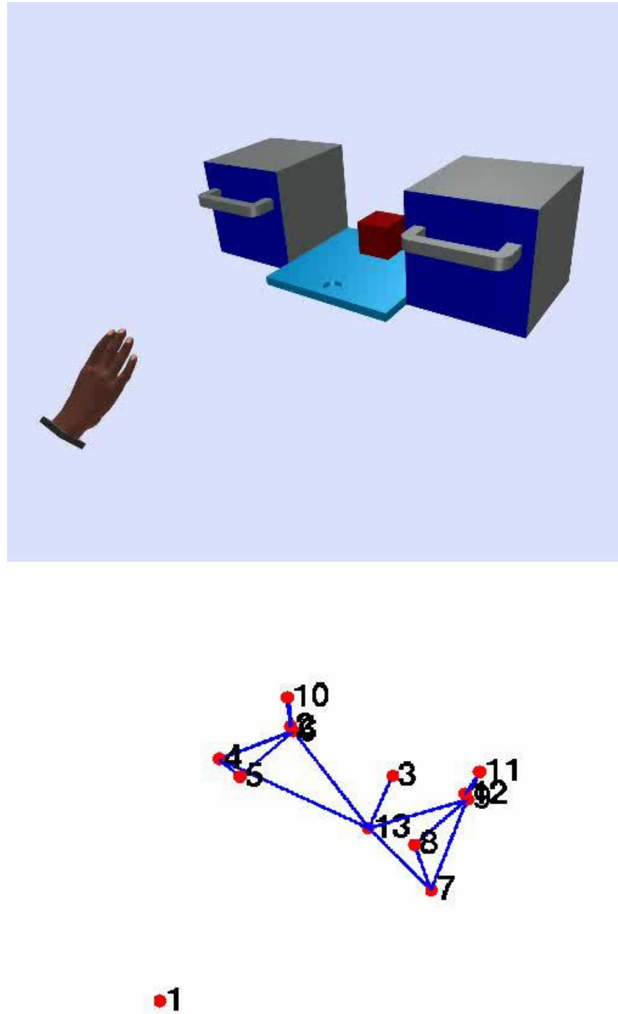
Aksoy, E. E., Abramov, A., Woergoetter, F. & Dellen, B (2010). Categorizing object-action relations from semantic scene graphs. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 398-405.

- SECs provide a condensed semantic description of the scene (Vision and Haptics) and the action sequence

LABEX: Version 1

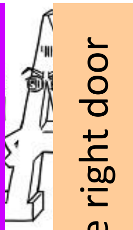
Actions

- Open left door
- Extract tray 1
- Close left door
- Take object 1
- Open left door
- Insert tray 1
- Close left door
- Open right door
- Extract tray 2
- Put object 1
- Take object 2
- Insert tray 2
- Close right door



Objects

- (1) Hand
- (2) Tray 1
- (3) Red cube
- (4) Left door
- (5) Left handle
- (6) Left cupboard
- (7) Right door
- (8) Right handle
- (9) Right cupboard
- (10) Object 1
- (11) Object 2
- (12) Tray 2
- (13) Blue plate



Initial Conditions			• Open left door			• Extract tray 1					• Close left door			• Take object 1						• Open left door			• Insert tray 1					• Close left door			• Open right door			• Extract tray 2					• Put object 1						• Take object 2						• Insert tray 2					• Close right door		
1	4	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	5	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	13	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	6	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	6	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	3	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	6	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	10	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	10	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
1	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
8	9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
9	11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
3	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
9	12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
1	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	0	1	1	0	0	0	0	1	1	1	1	0	0	0	0
10	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0
11	12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
11	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
9	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

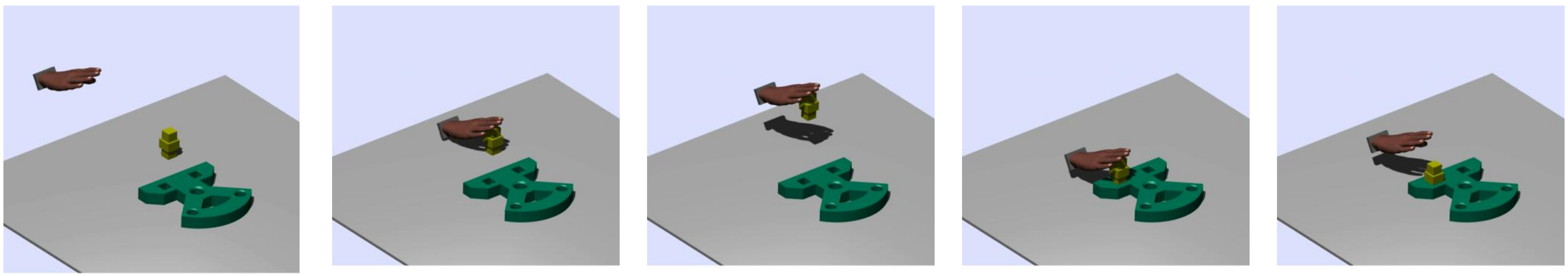
25 x 57

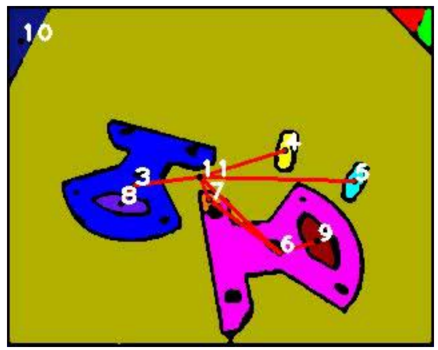
AUTAS: Peg in hole

$$\begin{array}{l} \rho_{1,3} \\ \rho_{2,3} \\ \rho_{3,4} \end{array} \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

(1) Hand
(2) Green plate
(3) Yellow peg
(4) Background

Frame 1 | Frame 26 | Frame 44 | Frame 100 | Frame 112

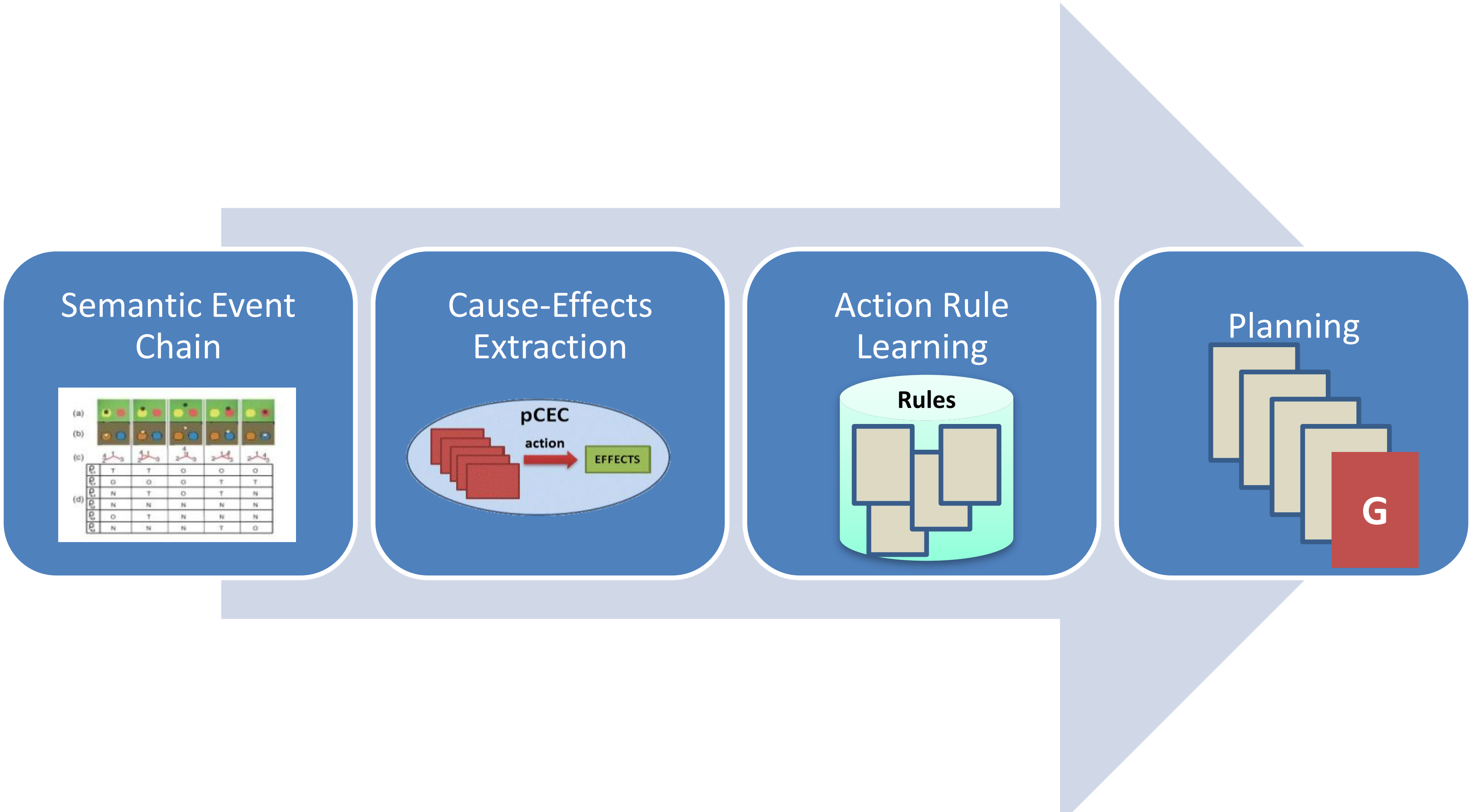




$$
\begin{array}{c}
\rho_{10,5} \\
\rho_{6,5} \\
\rho_{5,1} \\
\rho_{10,4} \\
\rho_{6,4} \\
\rho_{4,1} \\
\rho_{6,18} \\
\rho_{1,17}
\end{array}
\begin{pmatrix}
9 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
9 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 9 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 0 & 1 & 0 & 9 \\
9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 9 & 0 & 1 & 1 & 1 & 1 & 1
\end{pmatrix}
$$

(10) Hand
(6) Green plate
(5) Red peg 1
(4) Red peg 2
(1) Table
(17) Noise
(18) Noise

KT2: Probablistic Rule Learning



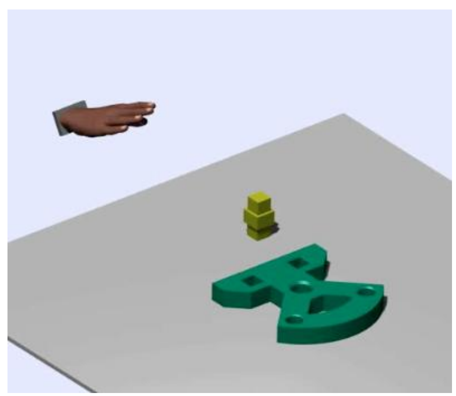
Mapping Graphs to Logic States

Graph state description → Logic Functions → **High-level state description**

$$\begin{array}{c} 1,3 \\ 2,3 \\ 3,4 \end{array} \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$ Touching relations

$$\begin{array}{c} 1,3 \\ 2,3 \\ 3,4 \end{array} \begin{bmatrix} F & T & T & T & T \\ F & F & F & T & T \\ F & F & F & F & F \end{bmatrix}$$ Alignment relations

- (1) Hand
- (2) Green plate (hole)
- (3) Yellow bolt (peg)
- (4) Background (table)



clear(hand)
clear(plate)
not(clear(peg))
not(clear(table))
ontable(peg)
not(holding(hand))
not(holding(plate))
not(holding(peg))
handempty
not(in(peg,plate))
not(aligned(hand,plate))
not(aligned(plate,bolt))
not(aligned(bolt,table))

Action Recognition from Graph Transitions

$$\begin{array}{c} 1,3 \\ 2,3 \\ 3,4 \end{array} \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Touching relations

$$\begin{array}{c} 1,3 \\ 2,3 \\ 3,4 \end{array} \begin{bmatrix} F & T & T & T & T \\ F & F & F & T & T \\ F & F & F & F & F \end{bmatrix}$$

Alignment relations

?

Action Recognition functions

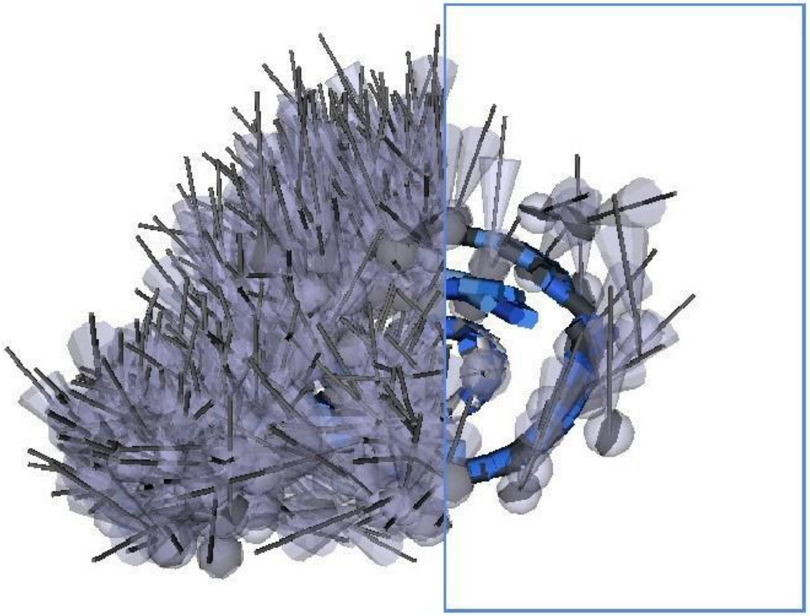
Actions

- grasp(O) Transition from not holding to holding. Grasp types?.
- release(O) Transition from holding to not holding.
- align(O1,O2) Transition from not aligned to aligned in objects relations.
- Insert(P,H) Transition from not inserted to inserted in peg-hole relations.
- lift(O) Transition from ontable to not ontable for object.

Execution of Semantic Event Chains

- KT2: Learn Action Plans from human observation
- Execution
  - KT3: Manipulation Densities
  - KT4: Dynamic Movement Primitives

KT3: Manipulation Densities

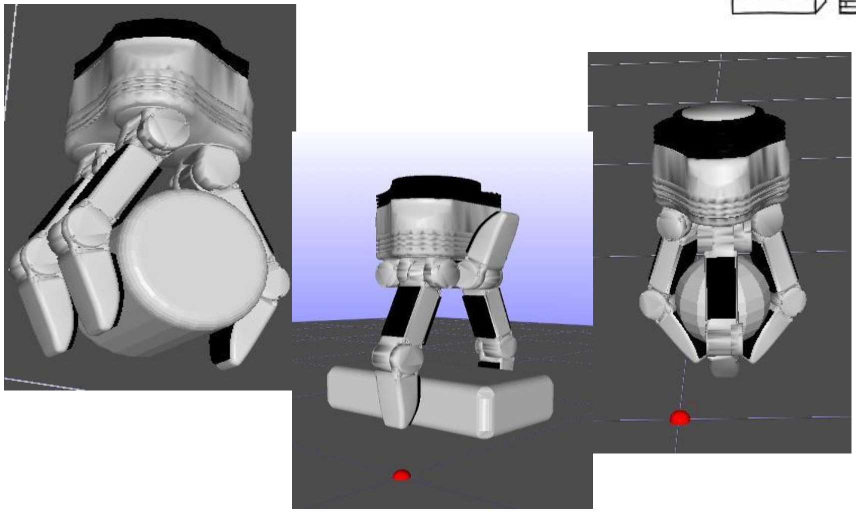


- Rich set of manipulation affordances on an object based on Kernel Density Estimation (KDE)
- Allows for efficient context specific use of these affordances

R. Detry, D. Kraft, O. Kroemer, L. Bodenhagen, J. Peters, N. Krüger, and J. Piater. Learning Grasp Affordance Densities. Paladyn. Journal of Behavioral Robotics, Volume 2, Number 1, 1-17, 2011.

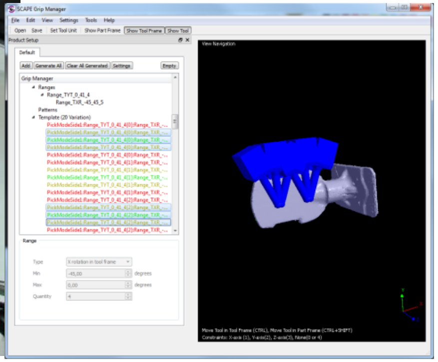
Extensions in IntellAct

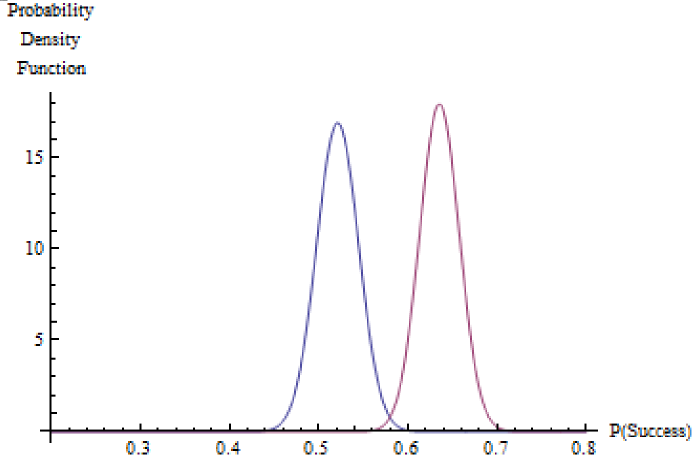
- Preshapes for the Schunk SDH-2 hand:
  - taking advantage of the flexibility of the Schunk hand
  - defining grasp types to cover relevant grasping scenarios
- Extension to actions beyond grasping
  - Peg-in-hole
  - Screwing



LearnBiP: Impact of Cognitive System Research to Industrial Robotics

- Learning in an Industrial Production Context
- Bin-Picking
  - Large sets of unused grasping experience
- Reduction of grasp errors by 25% in a real industrial Set-Up

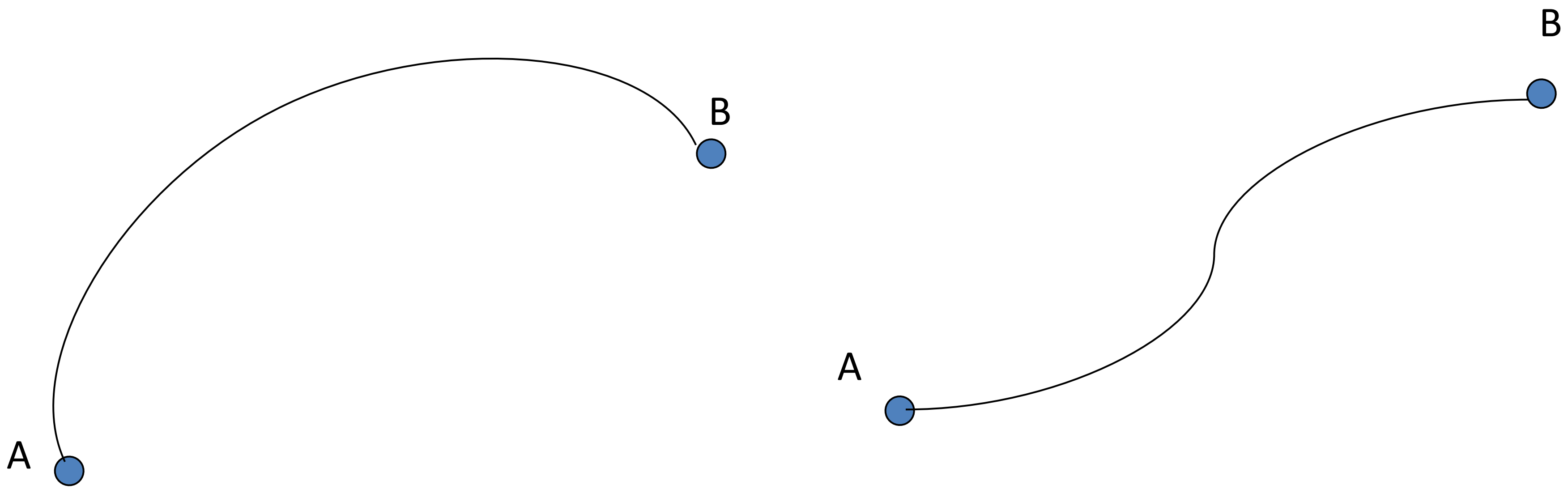




KT4: Dynamic Movement Primitives

"DMPs are units of actions that are formalized as stable nonlinear attractor systems" (Schaal, 2003, Ijspeert 2002)

DMP defines a movement trajectory from point "A" to point "B"



It's described by second order differential equation

Action Recognition from Graph Transitions

$$\begin{array}{c} 1,3 \\ 2,3 \\ 3,4 \end{array} \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Touching relations

$$\begin{array}{c} 1,3 \\ 2,3 \\ 3,4 \end{array} \begin{bmatrix} F & T & T & T & T \\ F & F & F & T & T \\ F & F & F & F & F \end{bmatrix}$$

Alignment relations

?

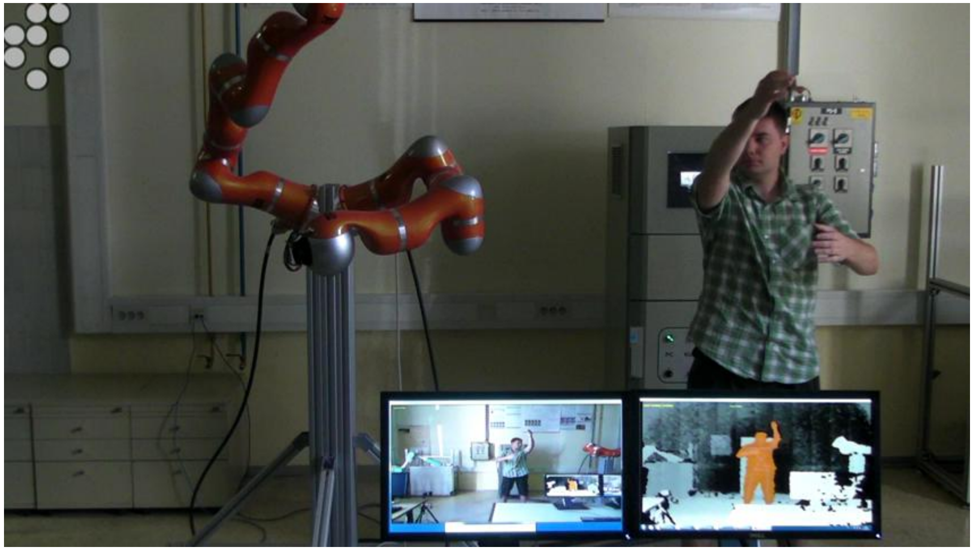
Action Recognition functions

Actions

- grasp(O) Transition from not holding to holding. Grasp types?.
- release(O) Transition from holding to not holding.
- align(O1,O2) Transition from not aligned to aligned in objects relations.
- Insert(P,H) Transition from not inserted to inserted in peg-hole relations.
- lift(O) Transition from ontable to not ontable for object.

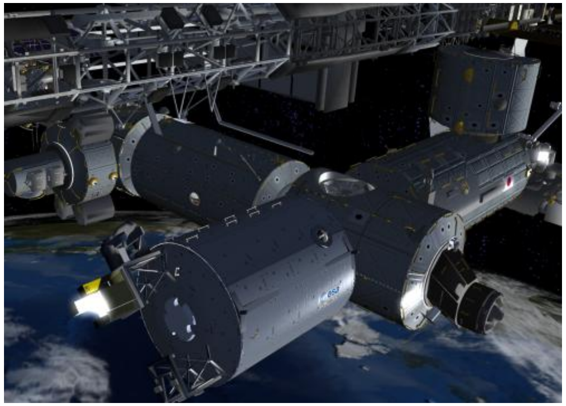
Programming by Demonstration with Self-Collision Avoidance

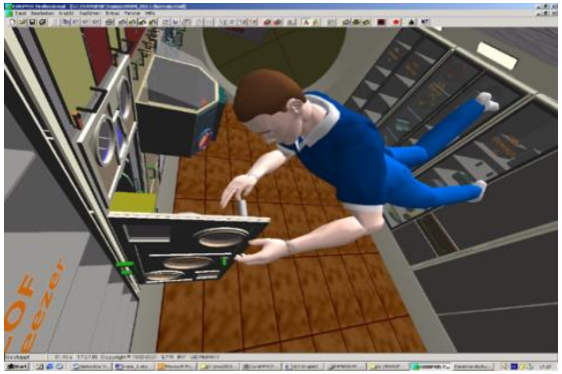
- <u>Primary task</u>: collision avoidance (only done when one arm approaches the other).
- <u>Secondary task</u>: track the demonstrated trajectory (suppressed if the distance between the arms becomes too small).



KT5: Use of VR in IntellAct

- Facilitating Vision and Haptics
  - Complexity and Noise can be controlled
  - Higher level people (planning and execution) can start directly
- Space station situation as such is a relevant scenario
  - Many training trials on earth before manipulations in space station are performed
  - Similar scenarios
    - Robots for undersea oil drilling instead of oil platforms
    - Actions in dangerous environments





Status of IntellAct after one Year

- Done
  - Semantic event chains, action recognition on VR data and rule learning
  - First recordings for Learning by Demonstration of grasping and PiH
  - Vision in rather simplified scenarios
- To be done
  - Learning of Plans
  - Studying of SECs in more realistic set-ups
  - Generalization of probabilistic manipulation functions to more complex actions
  - Execution of observed action sequences on the real robot

End