A ROS-integrated architecture to learn manipulation tasks from a single demonstration

L. Peppoloni, A. Di Fava, E. Ruffaldi and C.A. Avizzano
{l.peppoloni, a.difava, e.ruffaldi}@sssup.it

We present an architecture to learn and replicate households manipulation tasks by one demonstration.
The user is observed during the execution of the tasks, every action is analysed and its effect is translated into changes in the environment state. The logical sequence of the task is mapped into structured digital actions. From the map obtained the task can be performed. A planner adapts the execution to different initial environmental-conditions and possible in-task variations.

Problem and **Goal**:

- High-level representation of the skill (*primitives*)
- The problem is usually approached using multiple demonstrations [1, 2, 3] or with one demonstration using a-priori knowledge extracted from preliminary demonstrations [4].
- Improving efficiency by allowing the users to program new behaviors on the fly (1 demonstration).
- No a-priori knowledge of the task
- Robot observes user's manipulation tasks
- A state machine decodes user's actions into available primitives on objects and locations.

Operation and task representation

- Task is segmented through user interaction
- Generalization is achieved during execution through primitives, instead of statistical representation
- Action Primitives, e.g.

$$P^A = \{p_1^A, \dots, p_n^A\} = \{pick, place, \dots\}$$

- *Action Primitives* are inferred after every demonstration step from the changes in the environment state

$$S_k^i = \left\{ \begin{matrix} O_1 & O_2 & \cdots & O_n \\ pos_1 & pos_2 & \cdots & pos_n \end{matrix} \right\} \qquad S_{k+1}^i = p^A\left(S_k^i, O_s(k), pos_s(k)\right)$$

- A strip-like map of the task is obtained and saved

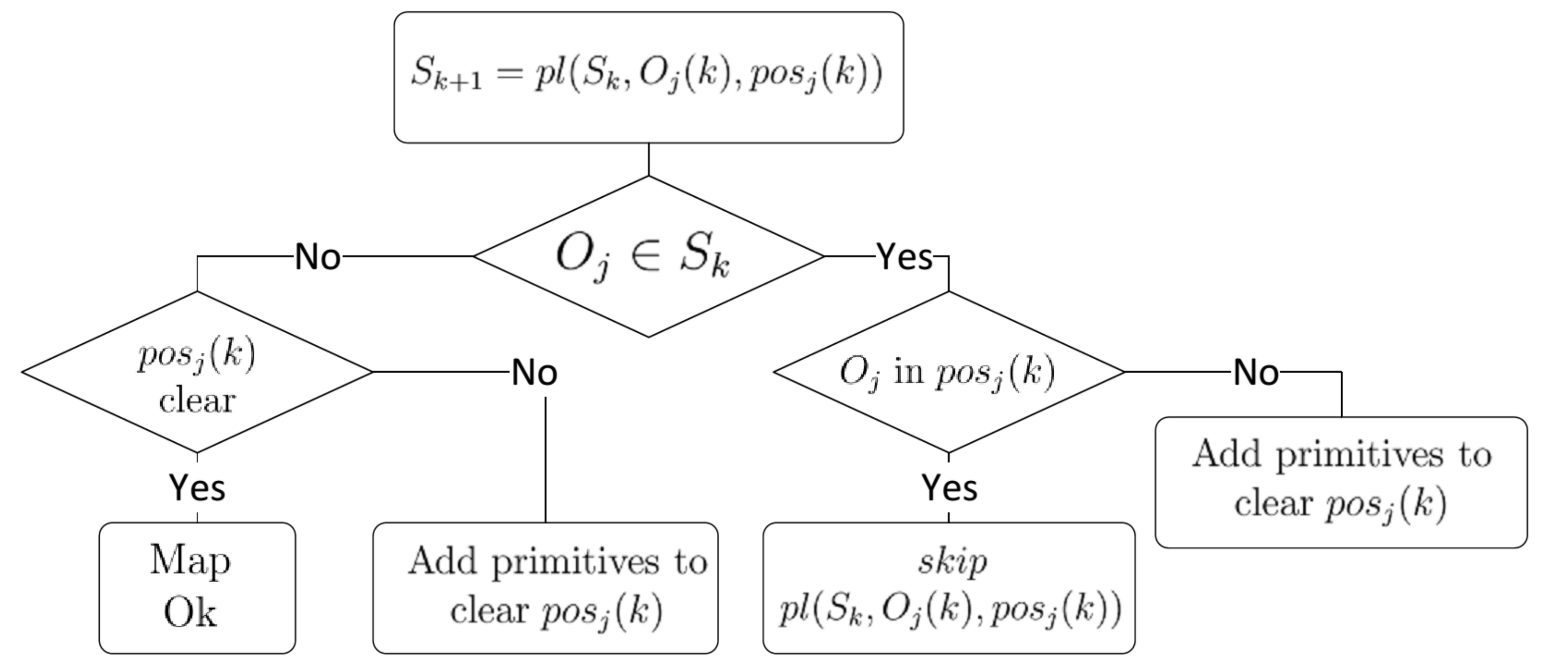
Task planning and execution (online planner)

An integrated state-machine provides a new ad-hoc *Execution map* that follows the task representation by adapting to environment state.

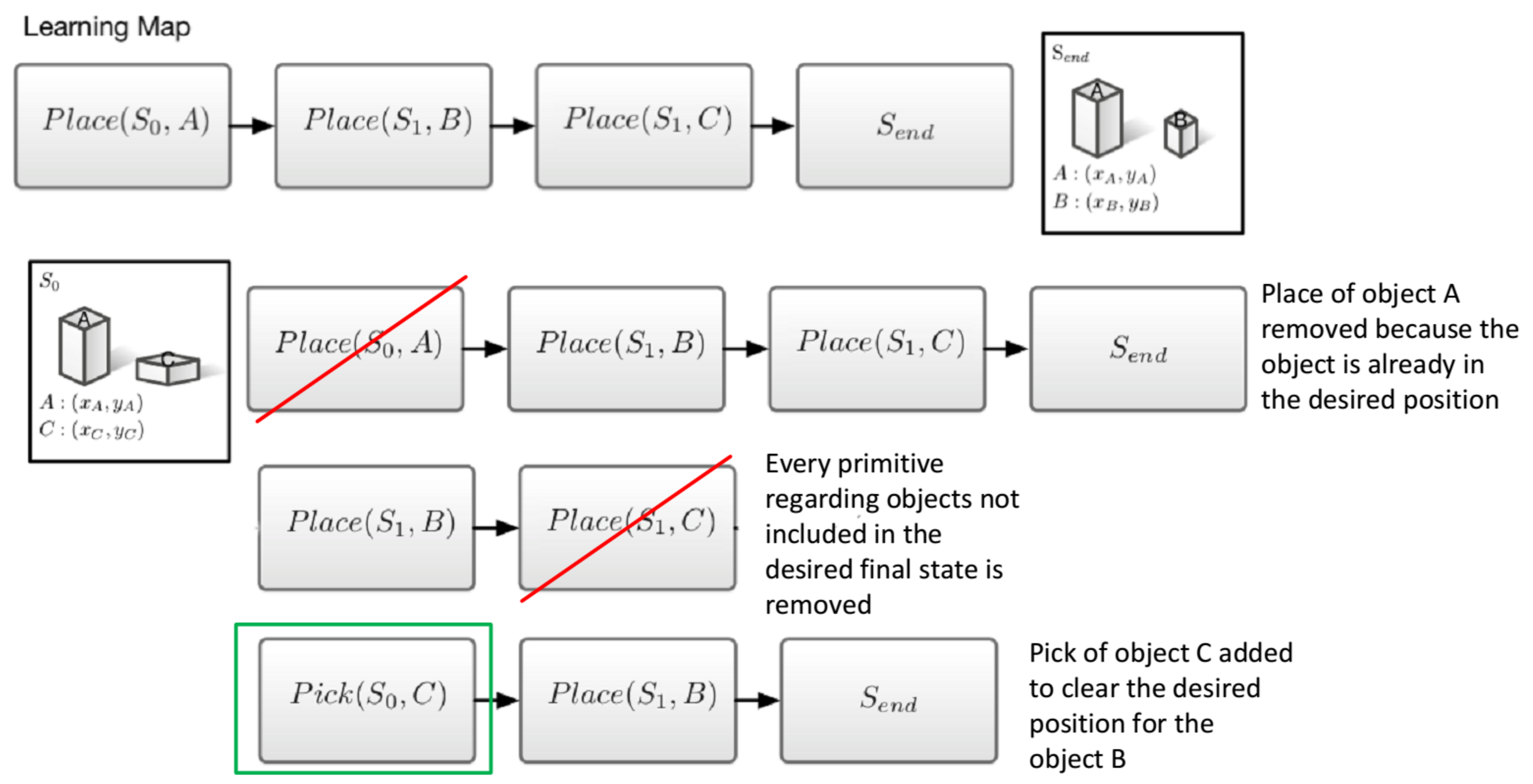
- Analyses the representation of the task and rearranges it taking into account the end goal, the actions order and possible new initial conditions
- Provides the robot with a action-fault tolerant behavior during the execution of the task. Every Action primitive effect is verified before proceeding in the execution.

[1] S. Ekvall and D. Kragic, "Learning task models from multiple human demonstrations," in Robot and Human Interactive Communication, 2006. ROMAN 2006. The 15th IEEE International Symposium on. IEEE, 2006, pp. 358–363.
[2] J. Saunders, C. L. Nehaniv, and K. Dautenhahn, "Teaching robots by moulding behavior and scaffolding the environment," in Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction. ACM, 2006, pp. 118–125.
[3] K. Ogawara, J. Takamatsu, H. Kimura, and K. Ikeuchi, "Generation of a task model by integrating multiple observations of human demonstrations," in Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on, vol. 2. IEEE, 2002, pp. 1545–1550.
[4] S. Ekvall and D. Kragic, "Learning task models from multiple human demonstrations," in Robot and Human Interactive Communication, 2006. ROMAN 2006. The 15th IEEE International Symposium on. IEEE, 2006, pp. 358–363.

Example of planner state machine (pick and place set)



Example of planner action starting from a Learning Map



Experimental Setup

KUKA YouBot

- Equipped with 2 Kinect Cameras

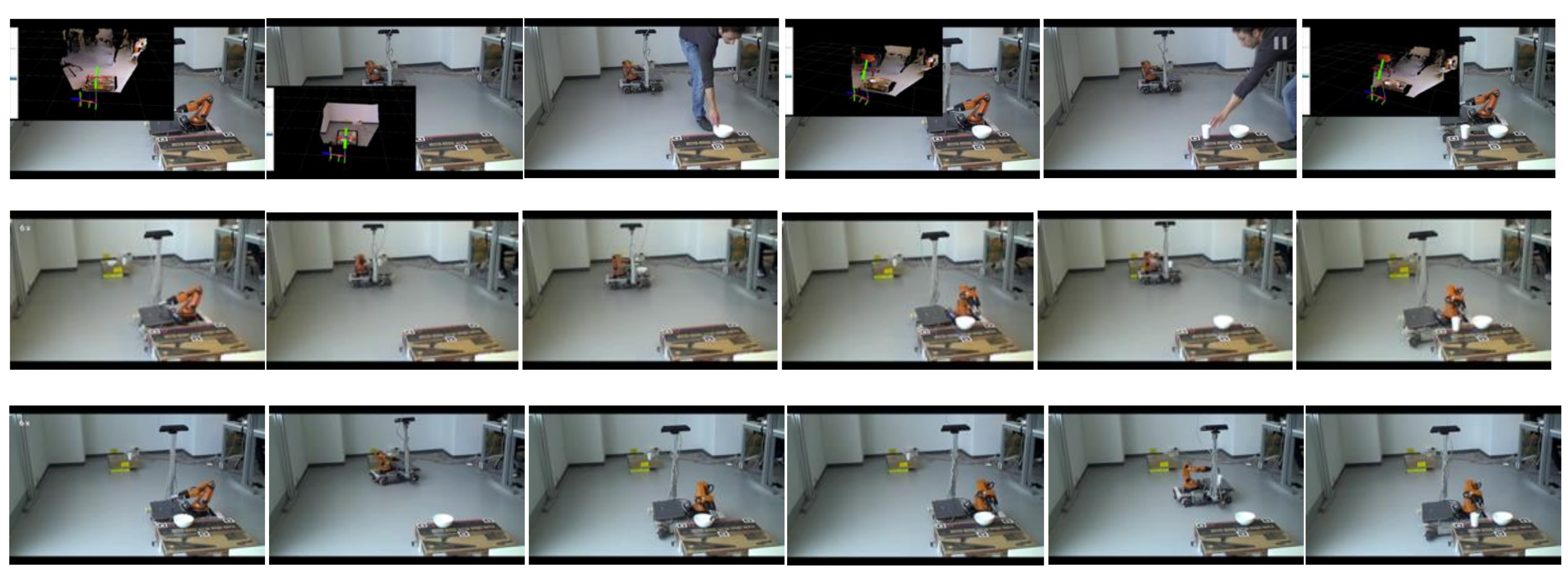
Kitchen-like Environment

- The chosen tasks are setting up and clearing a table for one person or more

Learning Test #1:
The robot learns to set up the table for one person

Performing Test #1:
The table is set up from 2 different environmental configurations

Results



During the experiments the robot has been capable of learning, planning and executing the task without problems. Further developments will regard issues related to specific setup scenarios:

- Improvement of vision architecture (e.g. objects very close, cluttering,…)
- Improvement of the navigation system robustness