Motion Descriptors for Intention Recognition in Robot Teleoperation Tasks

Mohamed Behery and Gerhard Lakemeyer

{behery, gerhard}@kbsg.rwth-aachen.de
Knowledge-Based Systems Group (KBSG)
RWTH Aachen University, Germany

Abstract

Real-time robotic assistance can offer an enhanced user experience if the robot is able to recognize the operator's intention. This is particularly important in the case of robot teleoperation using non-traditional input devices such as a brain computer interface. This paper proposes augmenting the PDDL action description with motion descriptors, which are geometric functions to describe motion primitives involved in each action. Using both the symbolic and geometric world state, we are able to filter the feasible actions narrowing down the search space. We conduct a preliminary user study that shows promising results in terms of usability. Our study shows that motion descriptors can be used with ease even by inexperienced users with no programming or robotics background.

Overview

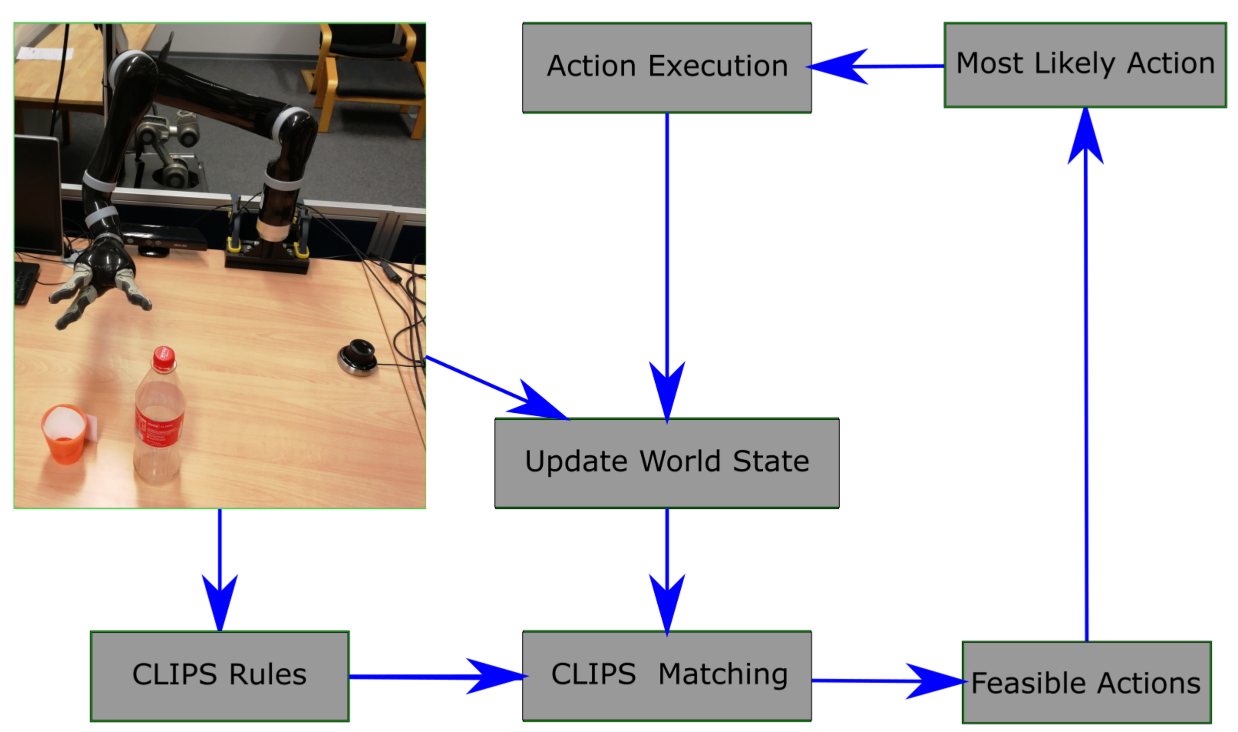

Figure 1: The proposed framework. The domain information is used to update the world state and the action PDDL definitions are translated into CLIPS rules, then the system is able to infer the feasible actions and perform the geometric reasoning given the motion description to recognize the user's most likely action

Our approach is based on a representation of actions that uses motion descriptors for the end-effector motion during manipulation actions. We use the CLIPS [2] rule-based production system to represent the agent's knowledge and reason about the world inferring the set of feasible actions. Afterwards, we use the operator's commands to reason about the geometric state of the world and find the action with the highest similarity to the operator's commands. We compare the operator's commands to the motion involved in the different actions and find which actions are most similar to the user's commands. The proposed representation allows a user to describe an action using motion primitives which facilitates describing new actions.

Motion Descriptors

We propose using descriptors of motion primitives as an extension to PDDL action definitions as seen in Table 1. We translate these descriptors into mathematical formulas and use them to judge the similarity between the motion and descriptors.

Name and Parameters	Preconditions	Effects	Motion
place (?s ?t ?m)	(bound ?s ?m) (sponge ?s) (table ?t) (manipulator ?m)	(on ?s ?t)	(perpendicular-to ?t roi) (direction-of ?t roi)
wipe (?s ?t ?m)	(bound ?s ?m) (sponge ?s) (table ?t) (manipulator ?m)	(clean ?t)	(parallel-to ?t roi) (direction-of ?t roi)
pour (?b ?c ?l ?m)	(bound ?b ?m) (contains ?b ?l) (bottle ?b) (cup ?c) (liquid ?l) (manipulator ?m)	(contains ?c ?l)	(direction-of ?c pouring)

Table 1: The PDDL definitions and motion descriptors used in this work. Names preceded by "?" are variable names (e.g., ?s) that are grounded at run-time. Lists of predicates in the different columns are treated as conjunctions.

At startup time, the PDDL action descriptions are translated into CLIPS rules, where the action preconditions are the transformed into LHS of a rule representing this class of actions. At run time, CLIPS uses pattern matching to ground the actions. The set of actions is filtered resulting in the set of *symbolically feasible* actions which significantly reduces the search space.An overview can be seen in Figure 1

Descriptor	Parameters	Function
direction-of(t)	t: point, line, or surface	$v = d(t,i) - d(m,i)$
perpendicular-to(t)	t: line or surface	$\theta = \arccos(\frac{i\cdot\bar{t}}{\lvert\bar{i}\rvert\cdot\lvert\bar{t}\rvert})$
parallel-to(t)	$\bar{t}$: line	$\theta = \arccos(\frac{i\cdot\bar{t}}{\lvert\bar{i}\rvert\cdot\lvert\bar{t}\rvert})$
parallel-to(t)	t: surface represented by 3 points a, b, c	$\bar{l_1} = a - b$ $\bar{l_2} = c - b$ $\bar{n} = \bar{l_1} \times \bar{l_2}$ $\theta = \arccos(\frac{\bar{i}\cdot\bar{t}}{\lvert\bar{i}\rvert\cdot\lvert\bar{t}\rvert})$

Table 2: Geometric functions to classify the motion according to the descriptors. In these functions, m is the manipulator pose and $\bar{i}$ is the user input.

Evaluation

Participants were asked to define several ADLs to evaluate the usability of these descriptors. After that they were asked to fill in a SUS questionnaire, the results are displayed in Table 3. They were also asked to perform ADLs which were recognized with modest accuracy.

Question	1	2	3	4	5
The task was mentally demanding	0	2	3	0	1
Actions were easy to describe	4	0	1	0	1
I need the support of a technical person to use this notation	1	0	0	4	1
People would learn to use this system very quickly	0	2	3	0	1
The System was difficult to use	2	2	1	1	0
I need to learn a lot to use this system	0	0	0	2	4

Table 3: A summary the results of the SUS questionnaire used in our study. These statements were rated on a scale from 1 to 5 where 1 means "Strongly Agree", and 5 means "Strongly Disagree". The table shows the number of responses for each selection.

Conclusion and Outlook

The proposed approach describes the motion involved in the action. During run time, the operator's commands are compared to the motion of the symbolically feasible actions to find the most similar one. The motion descriptors are human readable and use a set of geometric functions whose arguments are attributes of the world objects. New actions can be added and described using these descriptors, the functions can also be extended with minimal effort to describe more actions or fine tune the existing ones.
The next step is to integrate a deep-learning based motion primitive recognition approach and investigate the resulting accuracy. We also plan to test integrating these descriptors in to approaches such as SCTs [1].

Acknowledgements

We thank the anonymous reviewers for their helpful feedback.
Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC-2023 Internet of Production – 390621612.
This work is partially supported by the EU ICT-48 2020 project TAILOR (No. 952215)

References

[1] Gabriel Quere et al. "Shared Control Templates for Assistive Robotics". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020, pp. 1956–1962. DOI: 10.1109/ICRA40945.2020.9197041.

[2] Robert M Wygant. "CLIPS a powerful development and delivery expert system tool". In: *Computers & industrial engineering* 17.1-4 (1989), pp. 546–549.