Using Image Schemata to Support Autonomous Assembly Tasks

Nikolaos Tsiogkas[1,2]

[1]Department of Mechanical Engineering, KU Leuven, Celestijnenlaan 300, B-3001 Heverlee (Leuven), Belgium
[2]Core Lab ROB, Flanders Make, Gaston Geenslaan 8, 3001 Heverlee, Belgium

Abstract

Robots are an important part of the modern manufacturing industry. Newer generations of robots have improved sensing and control capabilities that allow the sharing of workspaces between humans and robots. In addition, these extra capabilities allow the robots to be agile, enabling the manufacturing of various goods, by using various skills developed by human experts. Unfortunately, such skills are sometimes hard to be reused, and are unable to recover from failures that were not anticipated from the designer, making them harder to be deployed in dynamic environments. This work proposes the use of image schemata as skill definition primitives. Such a representation will allow to make composable and easily reusable skills across domains using analogies. In addition, the image-schema-based representations can add a level of understanding of the skill performance and the environment, which can enable the recovery of unexpected errors during execution. Finally, this extended understanding can be used to explain the actions of the robot.

Keywords

Robotic assembly, Skill representation, Skill transfer

1. Introduction

Robots play an ever increasing role in the manufacturing industry. The first generation of industrial robots included static manipulators, manually preprogrammed to perform a specific task, which would blindly execute. This prevented the coexistence of humans with the robots as it was dangerous [1]. The emergence of collaborative robots, allowed humans to share the workspace with a robot, as it was safe to operate around each other, allowing closer collaboration between the two [2].

A further improvement is offered by the introduction of better sensing, planning, and control methods, which allowed robots to perform more than one task allowing for agile manufacturing. This enables the execution of more than one task using a set of skills that are prepogrammed by a human. A skill consists of a set of actions that a robot must execute, whose correct execution is monitored using the sensors. For example, a peg-in-a-hole skill [3], will be composed by an action picking the peg, then and action leading the peg to the hole, and finally, an action

The Sixth Image Schema Day (ISD6), March 24–25, 2022, Jönköping University, Sweden

nikolaos.tsiogkas@kuleuven.be (N. Tsiogkas)
https://www.tsiogkas.me/ (N. Tsiogkas)
0000-0003-2842-7316 (N. Tsiogkas)

performing the insertion of the peg in the hole. It can be seen that the combination of such skills can lead to complex behaviours.

Despite the greater agility provided by the use of skills and better sensing, multiple challenging problems still exist. Given that skills are developed by a human expert, the level of reusability and composability of each of them is dependent on the specific design decisions made, while the skill is created. For example, if a skill is making heavy use of robot specific primitives, such as specific joint or payload limits, it will be hard to use this skill with a different robot. In addition, a skill developed for solving a specific problem, most of the times cannot be used to solve an analogous problem in a different domain. For example, the aforementioned peg-in-a-hole skill at a high level can be analogous to insertion of an item in a box, used in kitting applications, but is unable to be reused for that purpose. This is caused, mainly, by including a lot of task-specific geometric information in the skill design, as well as, by hard-coded dependencies in specific sensing and monitoring capabilities of the robot.

Another problem is that skills cannot react to changes in the environment unless are explicitly coded to anticipate those changes. This means that some unexpected event can cause the skill to fail, without the robot being able to recover. To be able to cope with unexpected events, the robot would require a higher level of understanding of the situation. Such understanding is not encoded in state-of-the-art approaches for skill definition. Moreover, current approaches, use various ”magic numbers” that need to be hand-tuned for specific applications or robots. A higher level of understanding of the situation would allow the robot to automatically learn and configure all the relevant parameters for the task at hand. Finally, this higher level of understanding can be used by the robot to explain its behaviour.

To overcome such limitations we believe that image schemata [4] can be used as primitives to design a set of basic robotic skills, that are easy to compose and generate complex behaviours. Such an approach would allow the skills, and their compositions, to be easily transferred across domains through analogies, an inherent property of image schemata. In addition, using image schemata will allow understanding of the situation to be incorporated into the skill, making them resilient to disturbances and enabling them to recover from failure. Moreover, this understanding can be used to monitor the quality of the performance, and to actively try and avoid any failures. Finally, explainability can be achieved by using the knowledge and understanding of the situation, along with the intentions of the robot.

The rest of the paper is organized as follows: Section 2 presents the relevant literature regarding robotic skill definition. In section 3 the proposed skill definition approach is presented. Finally, section 4 concludes the work by presenting potential paths for future work.

2. Relevant work

In the robotics literature multiple skill definition approaches can be found. Unfortunately most of them are requiring a manual implementation, are defined on robot-centric primitives, and do not consider understanding of the skill execution to mitigate and recover errors. In [5] a framework for skill-based control framework based on the ROS middleware. Skills are defined as primitive actions that the robot has to perform like picking an object or placing it at a specific location. The skill definition is happening in software that implements the sensing, motion control, and monitoring in a hard-coded fashion. The work of [6] defines a skill as a directed graph where a vertex is a manipulation primitive, consisting of a twist and a feed forward wrench trajectory, and an edge is a transition between manipulation primitives triggered by a monitored condition. In addition, recovery strategies can be manually specified. In [7] a domain specific language for skill programming is defined. The skills are programmed by domain experts using a set of elemental actions, such as *OpenGripper*, *MoveTo* etc. Once a skill is defined control code is automatically generated. Another language-based approach for skill definition is presented in [8]. This language supports both code generation for control, as well as, symbolic representations of the skills to be used for reasoning and planning. The skill is defined using a set of robot specific primitives that are translated to a finite state machine for execution. In [9] a domain specific language is described that allows to define skills in a constraint based manner. In addition, it allows to specify monitors that can help with task execution, and it provides a reference implementation of a controller that can execute such tasks for a specific robot. In [10] a modelling approach for robot agnostic reusable skills is presented. It is based on hierarchical definition of skills, and the skills can be connected to concrete implementations that are hardware specific. The presented movement and grasping primitives require a manual skill definition by an expert, and despite claiming that the skills can be reusable across domains nothing is demonstrated.

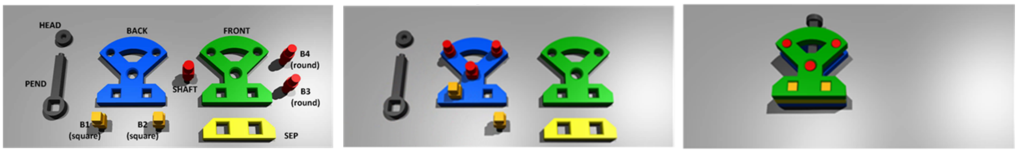
The work of [11] presents a way where image schemata can be used as an alternative to planning. There it is shown that image schemata can be used to reason about the environment, the various alternative ways to perform an action, and how the robot can use analogies to perform a variety of similar actions in diverse domains. It is rather relevant to the work presented in this paper, as we propose that skills can be described as a set of image schemata, which in turn can be used to automatically generate a controller to perform each skill. As skills and actions can be viewed as equivalent, this work agrees on the ability to transfer a skill to multiple domains.

In [12] a method that converts qualitative descriptions of scenes to quantitative equivalents is described. Such quantitative descriptions are then used in simulations to understand functional relations between objects in the world, and interpret them in a qualitative way. This is relevant to the work presented here, as understanding the functional relations will enable better monitoring for the skill execution.

3. Proposed approach

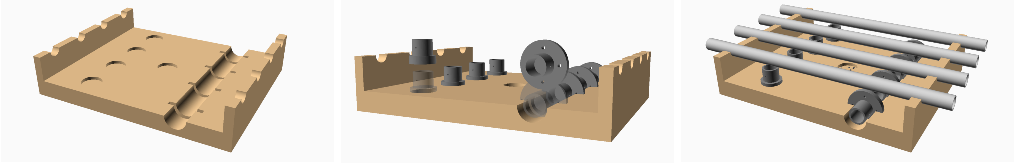
To demonstrate the use of image schemata for skill definition and how it can be transferred in different domains two examples of industrial applications will be used. A specific skill that is applicable in both applications will be encoded using image schemata and presented in detail.

The first application is a benchmark assembly application that can be seen in figures 1a, 1b, and 1c. The aim of this application is to assemble an industrial item that requires components to be placed in a specific partial order. To complete the assembly several skills are needed. For example, to insert the pegs and the shaft to appropriate places, a peg-in-a-hole skill is needed. In addition, there are some precedence constraints that need to be satisfied during the assembly, such as lower parts are supporting the placement of the higher parts.



(a) State of the item before... (b) ...during... (c) ...and after the assembly.

Figure 1: The Cranfield assembly task in various stages of execution. This task is used as a benchmark for testing solutions to the automated assembly problem using robots [13].



(a) Kitting base (b) Table feet inserted (c) Complete kit

Figure 2: Various stages of a kitting application. The base is supporting the other elements of the table. Feet and legs are inserted having precedence and position constraints.

The second application is a kitting application, where the robot has to collect and place a set of items in a package, and can be seen in figure 2. In this case a table is to be packed in a box, where the top of the table is on the bottom of the box, and the legs are placed on top using some support material. Again, there is a pick-and-place type of skill, which can be viewed as equivalent with the peg-in-a-hole skill. Their main difference is the tolerance and accuracy needed for the execution. Nevertheless, in state-of-the-art skill developments these two skills will not be compatible, meaning you cannot use one to achieve the other. However, if primitive skills were based on image schemata, their composition would be able to handle both cases, only needing a different type of monitoring and configuration.

To demonstrate the skill definition using image schemata, a peg-in-a-hole skill will be detailed. The objective of the skill is to insert a peg in a respective hole and is depicted in figure 3. The skill starts by moving the manipulator towards the peg, such a motion can be described by a *Source-Path-Goal* image schema (3c). Once the manipulator reaches the peg, it is grasped using the end effector, so the peg is *Contained* in the end effector. The next step involves moving the peg until it touches the surface containing the hole, as described by the *Contact* image schema (3d). Following is moving the peg until it cannot move further as it is *Blocked* by the hole (3e-3f). Finally, the peg is rotated to a vertical position above the hole, shown by a *Verticality* constraint and inserted until a secure *Link* is achieved (3f-3h).

The skill specification as a series of image schemata has the additional benefit of automatically defining what needs to be monitored during the skill execution. These monitors can detect what went wrong while executing the required actions, allowing the robot to recover and continue. For example, if the peg falls out of the end effector while it is moved towards the hole (3e) the containment constraint will not be satisfied. Then the robot can automatically reason that it needs to grasp the peg again, and continue the execution from where it was interrupted.

It can be seen that a similar approach can be used with placing the feet or legs of the table of

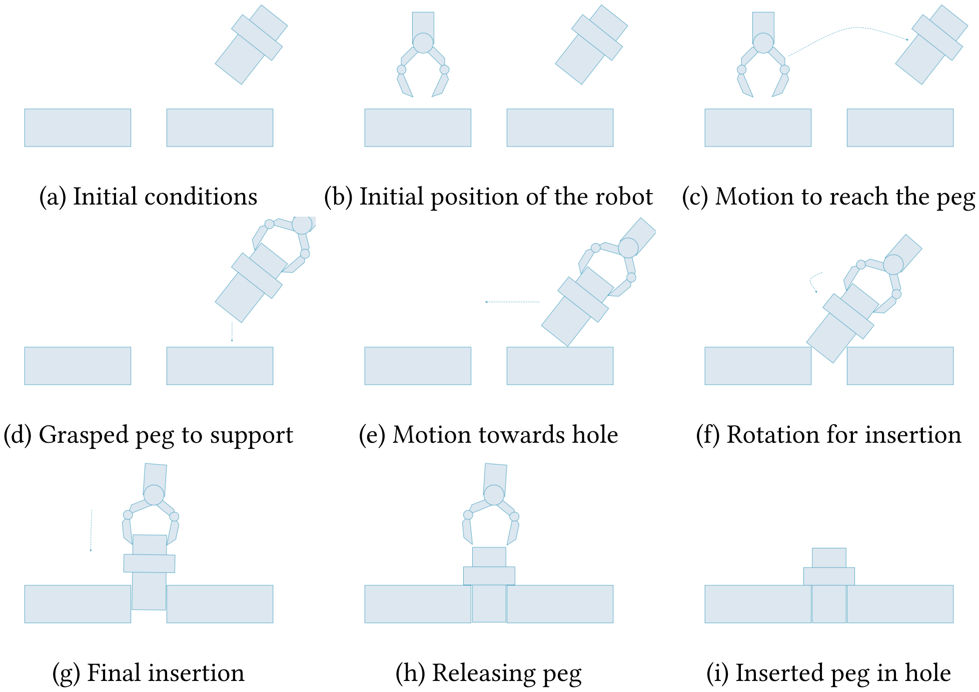

(a) Initial conditions
(b) Initial position of the robot
(c) Motion to reach the peg
(d) Grasped peg to support
(e) Motion towards hole
(f) Rotation for insertion
(g) Final insertion
(h) Releasing peg
(i) Inserted peg in hole

Figure 3: Various stages of a peg-in-a-hole skill. The robot needs to grasp a peg and successfully insert it into the respective hole. Initially the robot grasps the peg and moves it towards the surface that has the hole until there is contact 3a-3d. Then it moves the peg towards the hole until the motion is blocked by the peg being partly inserted 3e. Finally, the peg is rotated, inserted, and released 3f-3i.

the kitting application as seen in figure 2c. By using analogies the hole can be matched with the leg support portions of the base, and the peg is the leg itself. Then by just using the relevant parameters the leg can be inserted in the correct spots using the same skill.

4. Conclusion and future work

This work presented an approach to model robotic skills using image schemata as a core modelling primitive. This approach allows skills to be composable, and reusable in various tasks and contexts. In addition, they allow the robot to better monitor the execution of each skill, as well as, understand situations that can cause the execution of skills to fail, and be able to recover from such failures. Using such knowledge, the robot can also provide explanations, regarding its actions.

One of the most fruitful directions for future work is connecting these high level skills to concrete low level perception and control implementations that can be used by a robot. Since each sensor and manipulator have different capabilities, different software implementations will be required to perform each skill. For that a way to automatically define and parametrize the software used will be required. This will allow the easy deployment and adoption of such skills, as it will require minimal expert involvement.

Another interesting direction will be the use of such skills to explore, interact, and learn from the environment. As the robot understands basic concepts and can use them in skills to manipulate the environment, it can learn more about new skills, the affordances of the environment, as well as, the causality chain of events. One potential approach is to allow the robot to interact with objects, and detect and try to interpret the outcomes of its actions. A recent work towards that direction is presented in [14].

Finally, an important direction for future research is towards the explainability of the robot's actions. Using image schemata, and the increased understanding of the world they can bring, a robot must be able to explain why it chose a specific skill, or why it performed an action that is part of a skill in a specific way. For example, if the robot executes part of the insertion in the peg-in-a-hole skill using a different angle or a different approach, because there is something blocking the ”usual” way, it must be able to explain why it chose this alternative. This can increase the trust of users towards the system, and thus the adoption of more robotic systems in the real world.

Acknowledgments

This work was supported by the Flanders Make/VLAIO:SBO MULTIROB project.

References

[1] M. Vasic, A. Billard, Safety issues in human-robot interactions, in: 2013 ieee international conference on robotics and automation, IEEE, 2013, pp. 197–204.

[2] A. Cherubini, R. Passama, A. Crosnier, A. Lasnier, P. Fraisse, Collaborative manufacturing with physical human–robot interaction, Robotics and Computer-Integrated Manufacturing 40 (2016) 1–13.

[3] S.-k. Yun, Compliant manipulation for peg-in-hole: Is passive compliance a key to learn contact motion?, in: 2008 IEEE International Conference on Robotics and Automation, IEEE, 2008, pp. 1647–1652.

[4] M. M. Hedblom, Image schemas and concept invention: cognitive, logical, and linguistic investigations, Springer Nature, 2020.

[5] F. Rovida, M. Crosby, D. Holz, A. S. Polydoros, B. Großmann, R. Petrick, V. Krüger, Skiros—a skill-based robot control platform on top of ros, in: Robot operating system (ROS), Springer, 2017, pp. 121–160.

[6] L. Johannsmeier, M. Gerchow, S. Haddadin, A framework for robot manipulation: Skill formalism, meta learning and adaptive control, in: 2019 International Conference on Robotics and Automation (ICRA), IEEE, 2019, pp. 5844–5850.

[7] U. Thomas, G. Hirzinger, B. Rumpe, C. Schulze, A. Wortmann, A new skill based robot programming language using uml/p statecharts, in: 2013 IEEE International Conference on Robotics and Automation, IEEE, 2013, pp. 461–466.

[8] C. Lesire, D. Doose, C. Grand, Formalization of robot skills with descriptive and operational models, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2020, pp. 7227–7232.

[9] E. Aertbeliën, J. De Schutter, etasl/etc: A constraint-based task specification language and robot controller using expression graphs, in: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2014, pp. 1540–1546.

[10] S. Profanter, A. Breitkreuz, M. Rickert, A. Knoll, A hardware-agnostic opc ua skill model for robot manipulators and tools, in: 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, 2019, pp. 1061–1068.

[11] M. M. Hedblom, M. Pomarlan, R. Porzel, R. Malaka, M. Beetz, Dynamic action selection using image schema-based reasoning for robots, in: Proc. of the Joint Ontology Workshops, 2021.

[12] M. Pomarlan, J. A. Bateman, Embodied functional relations: A formal account combining abstract logical theory with grounding in simulation., in: FOIS, 2020, pp. 155–168.

[13] T. R. Savarimuthu, A. G. Buch, C. Schlette, N. Wantia, J. Roßmann, D. Martínez, G. Alenyà, C. Torras, A. Ude, B. Nemec, et al., Teaching a robot the semantics of assembly tasks, IEEE Transactions on Systems, Man, and Cybernetics: Systems 48 (2017) 670–692.

[14] M. Pomarlan, M. M. Hedblom, R. Porzel, Panta rhei: Curiosity-driven exploration to learn the image-schematic affordances of pouring liquids, in: The 29th Irish Conference on Artificial Intelligence and Cognitive Science 2021, Dublin, Republic of Ireland, December 9-10, 2021, CEUR-WS, 2021, pp. 106–117.