Using Robot Skills for Flexible Reprogramming of Pick Operations in Industrial Scenarios

Rasmus S. Andersen[1], Lazaros Nalpantidis[1], Volker Krüger[1], Ole Madsen[1] and Thomas B. Moeslund[2]
[1]*Department of Mechanical and Manufacturing Systems, Aalborg University, Aalborg, Denmark*
[2]*Department of Architecture, Design and Media Technology, Aalborg University, Aalborg, Denmark*

Keywords: Robot Vision, Robotic Skills, Industrial Robots, Tabletop Object Detector.

Abstract: Traditional robots used in manufacturing are very efficient for solving specific tasks that are repeated many times. The robots are, however, difficult to (re-)configure and (re-)program. This can often only be done by expert robotic programmers, computer vision experts, etc., and it requires additionally lots of time. In this paper we present and use a skill based framework for robotic programming. In this framework, we develop a flexible pick skill, that can easily be reprogrammed to solve new specific tasks, even by non-experts. Using the pick skill, a robot can detect rotational symmetric objects on tabletops and pick them up in a user-specified manner. The programming itself is primarily done through kinesthetic teaching. We show that the skill has robustness towards the location and shape of the object to pick, and that objects from a real industrial production line can be handled. Also, preliminary tests indicate that non-expert users can learn to use the skill after only a short introduction.

1 INTRODUCTION

In modern manufacturing industries automation using robots is very widespread, and automation has for several decades proven its potential to increase productivity many times. Still, areas exist where automation has not gained the same foothold. This is for instance the case in small to medium sized companies, where investment in automated production lines can be too big a risk. Also in large companies, automated production lines are typically required to run for several years to justify the investment. This can be a problem in shifting markets where consumer demand cannot always be predicted accurately, even for few years into the future. Especially for new products, there is a need for gradually increasing production, instead of constructing fully automated production lines all at once.

To make industrial robots better suited for these scenarios, it has been identified that improved configuration options and human-robot interaction are core requirements (EUROP, 2009). In short, making robots more flexible and easier to reconfigure and reprogram is a necessity for future manufacturing. In this paper we address the issue of making robot programming fast and flexible by taking advantage of a skill-based framework, which is described in Section 2.1.

The contribution of this paper is to show how such a skill-based framework can be used to develop a flexible vision-based pick skill, which fast and easily can be reprogrammed by non-experts. The developed pick skill uses a depth sensor to locate rotational symmetric objects, and then picks them up using a small number of taught parameters.

The paper is organized as follows: In Section 2.1 the concept of robotic skills is introduced. Both related research and the interpretation used here is presented. In Section 2.2 basic methods for performing object detection is described, and on this basis the approach taken here is presented.

In Section 3 the complete proposed system is described, consisting of a robot system (Section 3.1), a tabletop object detector (Section 3.2), and an integration of this into a pick skill (Section 3.3). Experimental results are presented in Section 4; first in Sections 4.1 and 4.2 with regards to robustness against variations in position and shape of the object to pick. In Section 4.3 the skill is tested on parts from an industrial production line from the Danish pump manufacturer Grundfos A/S, and in Section 4.3 reprogramming (teaching) of the pick skill is tested on non-expert users. Conclusions are finally drawn in Section 5.

S. Andersen R., Nalpantidis L., Krüger V., Madsen O. and Moeslund T..
Using Robot Skills for Flexible Reprogramming of Pick Operations in Industrial Scenarios.
DOI: 10.5220/0004752306780685
In *Proceedings of the 9th International Conference on Computer Vision Theory and Applications* (VISAPP-2014), pages 678-685
ISBN: 978-989-758-009-3

2 BASIC CONCEPTS AND RELATED WORK

The central concepts for the pick skill presented in this paper are robotic skills and object detection. These are covered here.

2.1 Robotic Skills

The concept of robotic skills it not new, and a significant amount of literature has attempted to define the concept in the most generic, useful way. The purpose of a robotic skill is to encapsulate complicated knowledge and present itself to a non-expert user in a way that allows the user to exploit this skill to make the robot perform new, repeatable operations on objects. One of the first to draw attention to this idea are (Fikes and Nilsson, 1972) with their STRIPS planner. Their focus is to make it possible to perform automatic planning of how to get from an initial state to a goal state by combining a set of simple actions or skills. In (Archibald and Petriu, 1993), the idea of skills is further generalized in a framework known as SKORP. Here, the focus is both on what a skill should consist of, and also on developing a (at the time) modern and user friendly interface to enable workers to program robots faster.

A broader view is taken in (Gat, 1998), where the complete planning and control structure of robots is treated. It is argued that a traditional sense-plan-act (SPA) architecture is insufficient for robots solving problems in a complex scenario, and he suggests instead a three-layered architecture, where all layers hold their own SPA structure. In the lowest layer, real-time control is placed. In the highest layer, time independent processes are run, that are related to the overall task of the robot. The middle layer, the *Sequencer*, continuously changes the behavior of the lowest level, in order to complete a goal. The functionality in this level can be interpreted as a type of skills, which encapsulates the functionality of the lower levels, and works towards a goal by sequencing hardware-near actions.

More recently, in (Mae et al., 2011) the three layered structure is augmented by yet another layer on top, which contains a high level scenario description presented in a user friendly interface. This is aimed at solving manipulation tasks for service robots. The focus here is also on the architecture, where skills constitute the third layer; just above the hardware control layer. Another approach to skills are taken at Lund University, where skills are seen as a means to reuse functionality across multiple platforms (Bjorkelund et al., 2011).

We interpret a skill as an object-centered ability, which can easily be parameterized by a non-expert. Rather than attempting to develop a large architecture, able to handle general problems in real world, we focus on making skills very easy and fast to reconfigure by non-expert humans. The focus is therefore especially on the human-robot interaction, and on the reusability of skills within the domain of tasks that are frequently required by the manufacturing industry. In its essence, a skill in our framework consists of a teaching part and an execution phase, as illustrated in Figure 1 (slightly modified from (Bøgh et al., 2012)).

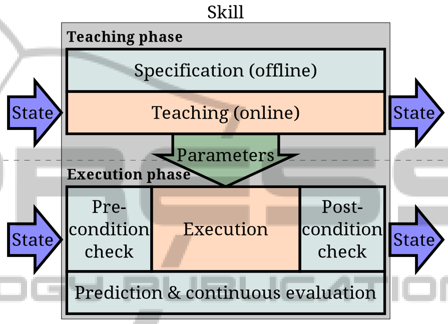


Figure 1: General structure of a skill. A skill consists of a teaching phase (top) and an execution phase (bottom). Both during teaching and execution, the skill transforms the world state in a way, which allows other skills to proceed.

The execution block in the figure captures the ability of the robot to perform a task. The remaining blocks is what makes this ability into a skill. The teaching phase is what makes the skill reprogrammable. Here, the user specifies all parameters, that transform the skill from a generic template into something, which performs a useful operation. The phase is divided into an offline specification part and an online teaching part. The specification is typically done while selecting to use this skill to solve (part of) a task. This can be done using e.g. a computer or a tablet. In the online teaching most of the parameters are specified, and this can be carried out using a robot controller, kinesthetic teaching, human demonstration etc.

The execution phase consists of the execution block combined with pre- and postcondition checks as well as prediction and continuous evaluation. The precondition check determines if the world state lives up to the requirements of the skill. For a pick operation this can for instance include checking whether the gripper is empty, whether the robot is properly calibrated, etc. If this check is passed execution can begin. During ongoing execution continuous evaluation ensures that the skill is executed as expected. When the execution has finished, a postcondition check determines if the current world state is as predicted. Together, the pre- and postcondition checks of a skill makes it fit with other skills, and thus allows it to be a piece in solving larger tasks.

We believe that a framework based on skills has the potential to increase the speed and ease of the way humans interact with industrial robots. In this paper, one such skill based on a tabletop object detector is proposed, and in Section 3.3 in particular it is described how it is realized as a skill.

2.2 Object Detection

Object detection, recognition and pose estimation making it possible to pick certain objects are some of the most fundamental problems to solve in robot vision, and many different approaches have been taken. (Klank et al., 2009) describe some of the most common methods. A depth sensor is used to detect a plane surface, and segment point clusters supported by this. An RGB camera then captures an image, and CAD models are fitted to the potential objects inside the regions given by the depth sensor. This provides poses of the objects. Good gripping points have been specified in advance in accordance with the CAD model, and one of these is used to pick the object.

Specifically for detecting objects on a surface, the field has matured enough to methods are available, e.g. through ROS. In general, these systems however rely on either predefined CAD models or require models to be learned before being used. This is a complicated and often time consuming step, which is necessary for some objects, but not for all. It is not always necessary to have a detailed model to be able to pick, place and handle simple objects. Specifically for rotational symmetric objects, which are very common in the industry, a cylindrical model is usually sufficient for determining their position precisely. Therefore, it is here chosen to use a *containing cylinder* model to estimate the position of all objects. The cylinder model is *containing* in the sense, that the entire object is inside the model. Therefore, the object can be grasped by opening the gripper more than the diameter of the cylinder, moving the gripper to the center point of the cylinder, and closing the gripper until the force sensors detect contact with the object. For complicated objects this approach can obviously cause problems, but for many objects encountered in industrial manufacturing, it will work satisfactory without the need for object-specific CAD models. At the same time, not using models simplifies the process significantly to human users. A more complicated object detector can then be used in situations where the object is very unsymmetrical.

3 PROPOSED SYSTEM

The proposed system consists of a robot system, a tabletop object detector, and the integration of this detector into a skill-based structure. These are described in the following subsections.

3.1 Robot System

The robot system used here is called "Little Helper" (Hvilshøj and Bøgh, 2011; Hvilshøj et al., 2009), and it is shown in Figure 2. It consists of a robot arm, a gripper, and has here a depth sensor mounted on its end effector.

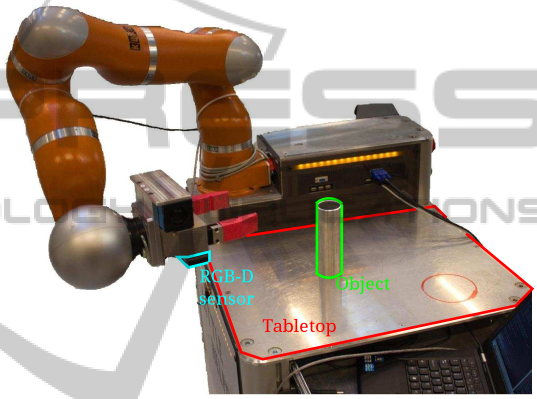


Figure 2: The robot system consists of a robot arm, gripper, depth sensor, tabletop, and an object to pick.

In the figure, the robot is about to grasp an object placed on a tabletop. The tabletop considered in this work is part of the robotic platform, although any surface within reach of the robot could be used.

The robot arm is a KUKA LWR, which has 7 degrees of freedom and supports force control. The gripper is a traditional parallel gripper. The depth sensor is a PrimeSense Carmine 1.09. This is similar to the Microsoft Kinect; only smaller and it functions at distances down to 35 cm. Is has both a RGB camera and a depth sensor, but here only the depth sensor is used. This captures the point clouds used for the table top object detector.

3.2 Tabletop Object Detector

The tabletop object detector described here has been developed specifically for this system. However, the type of object detector used is not essential for the skill-based approach. Thus, any available object detector could in principle be used.

The tabletop object detector takes as input a point cloud, and outputs the objects which are positioned on a supporting plane. The object models used as *containing cylinders*, in the sense that they enclose all points belonging to each object. The center position and diameter of these cylinders can be used directly for a pick operation. Figure 3 shows an example of a segmented point cloud.

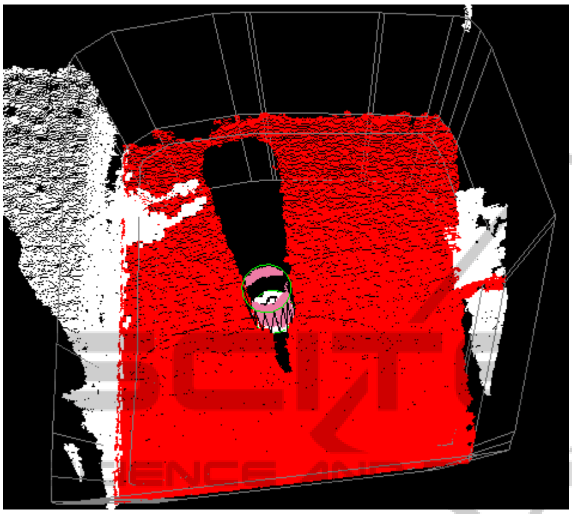


Figure 3: The segmented point cloud from the tabletop object detector seen from above. Points belonging to the detected plane are red, points belonging to the detected object are pink, and a green containing cylinder is fitted around the detected object.

The main steps in the object detector are shown in Figure 4. The first step is to actually find the dominant plane in the scene, i.e. the tabletop. This is done using RANSAC (Fischler and Bolles, 1981), followed by a least squares fitting to all inliers. The inliers are colored red in Figure 3.

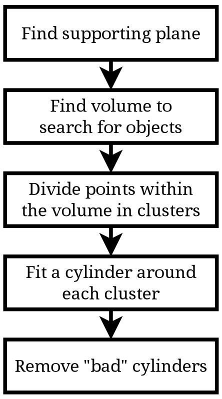


Figure 4: The main steps in the object detector.

The plane model is often supported by both the actual tabletop as well as other objects. Therefore, all inliers are clustered, and the largest cluster is chosen as the tabletop. Then, a convex hull is fitted around the tabletop, and it is extended along the norm of the plane to give a 3D volume. This volume is illustrated in Figure 3 as gray borders, and all objects supported by the plane must be within this. The points inside the volume are clustered, and very small objects as well as objects "hovering" over the plane are ignored. Cylinder models orthogonal to the detected plane are fitted around the rest. In Figure 3 one cylinder model has been fitted. The cylinders are containing in the sense that all points are inside or exactly on the borders of the model.

Finally, each cylinder model is checked for validity. Cylinders with a center very close to the border of the table are removed along with cylinders whose border is partly outside of the plane. This serves both to remove false objects, and also to remove objects that are too close to the border to be safely grasped.

3.3 Skill Realization

The purpose of the object detector is to be the core of a pick skill, which can easily be reprogrammed to solve new specific pick operations. This requires a number of other functions, as indicated in Figure 1. The most essential is which parameters it will be advantageous for the user to be able to specify and how to do this most efficiently. For this pick skill the most essential parameters considered are:

- The *camera pose* used for detecting objects.
- The *orientation* of the gripper used for grasping the object.
- The vectors used for *approaching* and *leaving* the grasping position.

Additionally, it is chosen to include the following parameters related to safety and robustness:

- The *velocity* of the robot.
- A *via position* for the robot, to help the robot avoid obstacles in the scene when approaching the object.
- The *height* of the gripper when grasping the object.
- The approximate *diameter* of the object to pick. This can be used to verify that the robot has grabbed the correct object.

The velocity of the robot can be provided most precisely by the user in the offline specification. The remaining parameters are provided through kinesthetic teaching; that is, by the human user manually moving the robot around. The sequence is illustrated in Figure 5.

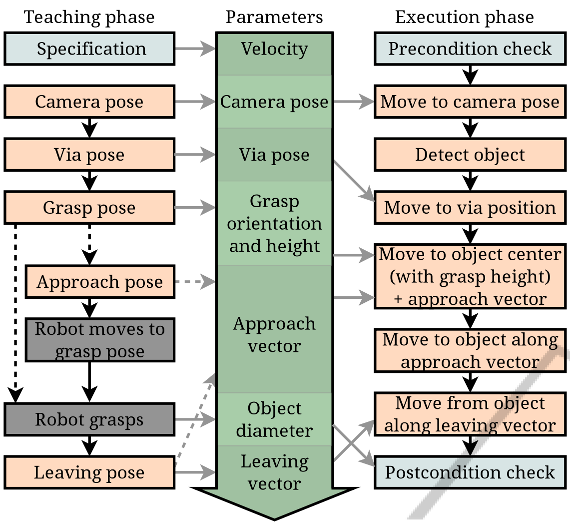


Figure 5: Teaching and execution phase of the pick skill. In the offline specification, the velocity of the robot is specified. It is also specified whether the same approach and leaving vector will be used. If that is the case, the "approach pose" is skipped during online teaching. All orange blocks in the teaching phase is input by the user through kinesthetic teaching.

During teaching of the camera pose the depth image from the camera is shown to the user. This allows him/her to precisely position the camera, so that the entire desired tabletop is in view. The pre- and postcondition checks verify that the gripper is initially empty and ends up holding an object of the approximate correct size (diameter). The continuous evaluation (see Figure 1) makes sure that the skill execution proceeds as planned, including that valid object(s) are detected.

One important feature to note for any skill is that the ending state must be the same for the teaching and execution phases. The same applies for the requirements for the starting state. This makes it possible to perform continuous teaching of several specified skills in a row. Since a successful execution of this pick skill will result in the gripper holding an object, the teaching phase of the pick skill is ordered to ensure, that the gripper also here ends up holding an object. This allows the user to continue teaching a skill that expects the gripper to hold an object - such as a place skill.

4 EXPERIMENTAL RESULTS

The pick skill was tested in a number of ways which is described in the following. First the robustness of the skill was tested with regards to variations in position and shape compared to the shape used for teaching. Then the skill is tested on industrial parts from a pump production line. Finally, the teaching part of the skill was tested on non-expert users. The objects used for the tests are shown in Figure 6. Where nothing else is specified, the perfect cylinder in Figure 6(a) was used.

(a) Cylindrical and deform objects. (b) Pump parts from an industrial production line.

Figure 6: Objects used for pick tests.

4.1 Variation in Position

During the teaching phase the object was placed in the middle of the tabletop shown in Figure 2, and the camera position was taught so that the object was in the center of the image. In a realistic scenario, the position of the object cannot be expected to always be the same during execution. To test the ability of the skill to handle this, the execution was executed 33 times. Each time, the object was placed between 0 and 20 cm from the teaching position. The results are shown in Figure 7.

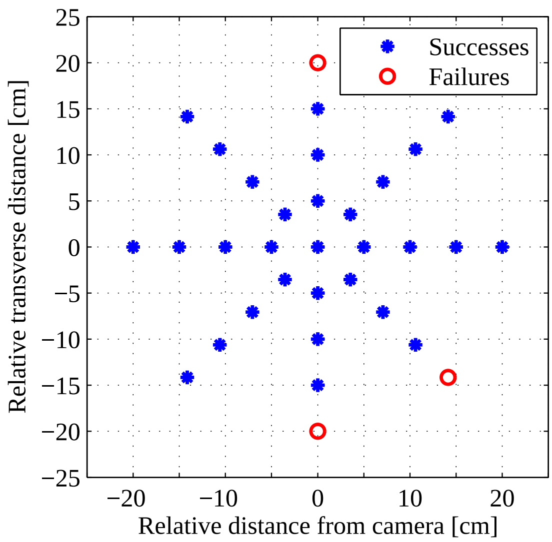


Figure 7: Results for pick with variation in position relative to position used during teaching. The figure shows the tabletop from above, and the camera position was to the left. The pick operation succeeded for all positions with variations up to 15 cm. For errors of 20 cm, the pick operation failed in 3 of 8 cases.

Of the 33 executions, the robot succeeded in picking up the object 30 times and failed 3 times. This corresponds to a success rate of 91%. Notably, all executions with a position deviation of 15 cm and below succeeded, while all errors occurred at a deviation of 20 cm. Visual inspection showed that the two errors at (0,-20) and (0,20) were caused by the object being very close to the border of the plane (see Figure 7). This caused the object detector to disregard the point clusters as valid objects. The last error was caused by an imprecise position estimation.

The time for each pick operation was on average 19.5 seconds. This includes 2.4 seconds on average for object detection.

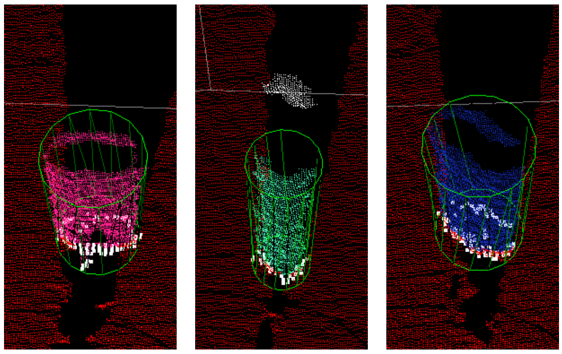
4.2 Variation in Shape

The object detector and pick skill were designed for rotational symmetric objects, as previously described. It has, however, some robustness to variations in shape. This robustness is tested in this section using the increasingly deformed cylinders shown in Figure 6(a).

The pick skill was first taught using the perfect cylinder shown to the left in Figure 6(a). Then, execution was carried out using increasingly deformed cylinders. It was attempted to pick each of the cylinders three times: Once seen from the wide side, once seen from the narrow side, and once seen in between (skewed).

In Figure 8 the most deformed cylinder is seen from the three different angles. The object detector has in all cases attempted to fit a cylinder model around the detected point cloud. For the wide and the skewed view, the model is fitted correctly. For the narrow view in Figure 8(b) the object detector has failed to classify all points from the cylinder together. In reality, both the green points inside the cylinder model as well as the white points behind the cylinder model come from the object. However, only the green points were used. The reason is, that the depth sensor is poor at detecting points on the top border of the physical cylinder as well as inside its narrow hole. Therefore, a gap arises in the point cloud between the front and back sides of the cylinder. Unfortunately, the gap is too large to allow it to be bridged by the clustering algorithm of the object detector. The effect is that the position of the object is estimated imprecisely.

Table 1 summarizes the tests of the deformed cylinders. The pick operation succeeded in all situations; even for the erroneously fitted narrow view orientation; in the sense, that the gripper held the object when execution had finished. In two situations the position of the object in the gripper was a bit off.



(a) Wide. (b) Narrow. (c) Skewed.

Figure 8: Rotations of the most deformed cylinder from Figure 6(a). For the wide and skewed orientations, a containing cylinder is fitted successfully. For the narrow orientation, the containing cylinder is only fitted to the frontal part of the physical cylinder.

The first of these situations, cylinder 4 seen from narrow side, is the one illustrated in Figure 8(b), and the reason is the imprecise position estimate. The second is the one illustrated in Figure 8(a). Here, the position estimate is relatively accurate. However, the position where the gripper grasps the object was taught, so that the object here was skewed in relation to the gripper. When it was grasped, it “slipped” into place in the gripper.

Table 1: Test results for picking the deformed cylinders from Figure 6(a). Only one result is given for cylinder 1, since its shape does not change with orientation. It was possible to grasp all cylinders with all tested orientations. In the situations marked as (+), the object was grasped, but its position in the gripper was not as expected.

No.	**Diameter**		**Seen from**		
	Wide	Narrow	Narrow	Wide	Skewed
1	4.9 cm	4.9 cm	+		
2	5.4 cm	4.4 cm	+	+	+
3	5.9 cm	3.8 cm	+	+	+
4	6.9 cm	3.1 cm	(+)	(+)	+

4.3 Application Specific Tests

The pick skill presented here was designed to function in industrial scenarios with industrial, rotational symmetric objects. Therefore it was tested on the industrial objects shown in Figure 6. All three objects are from a pump manufacturing line at the Danish pump manufacturer Grundfos A/S. From left, the objects are respectively a rotor cap, a rotor core, and an assembled rotor.

Obviously, the optimal place to grasp these objects is not the same. Therefore the pick skill was taught/parameterized specifically for each of the objects. Using these, the robot was able to successfully pick each of the objects three times in three attempts.

For the teaching phase of the skill, a preliminary test was carried out to verify that it is possible for humans to learn to use the skill fast. Two users with no previous knowledge about the pick skill were asked to teach the skill after receiving only a short introduction to the skill (∼5 min). The users succeeded to teach the skill in the first attempt, and the teaching were in all cases completed in less than 4 minutes. The parameterized pick skills could afterwards be executed. In one case the parameterized pick skill only functioned in a very limited area, because the robot otherwise was asked to move outside its limits after grasping the object. This was, however, easily visible and could if desired be corrected by re-teaching the skill.

5 CONCLUSIONS

In this paper, our interpretation of a flexible, robotic skill has been presented. We see a skill as an object centered ability, which encapsulates advanced functionality in a way that allows a non-expert user to easily program the robot to perform a new task. In this framework, a pick skill has been developed. The purpose of the skill is to make it possible for a robot to pick up rotational symmetric objects by using a depth sensor for detection. The skill features both a teaching and an execution phase. In the teaching phase, the user parameterizes the skill, mainly though kinesthetic teaching. In the execution phase, the taught parameters are used to pick up objects.

The skill uses an object detector which is specifically designed to detect the position of rotational symmetric objects, which it does sufficiently accurate to pick up the objects. Also, experiments have shown that the skill has robustness to deviations in position and shape. With regards to position, our robot was able to pick up 25 of 25 objects placed up to 15 cm of the location used for teaching the skill. At a distance of 20 cm it failed for 3 of 8 objects, mainly because the objects were very close to the border of the table. With regards to variation in shape, it was possible to pick up objects with different shapes, including a deformed cylinder with a diameter of more than double on one side compared to the other. Moreover, it was possible to pick up rotational symmetric objects from a real production line at Grundfos A/S.

Finally, the teaching phase of the skill was tested on users with no experience with the skill. With minimal introduction, the users were able to complete the teaching phase in less than 4 minutes.

We believe that with further development of functionality within a flexible skill based structure, there is a potential for robots to perform tasks, which it has previously not been profitable to automate. This is especially the case in small companies and in industries that manufacture products for rapidly changing markets. A skill-based approach to robot programming can make it possible for non-expert users to perform fast reprogramming of the robots to perform required tasks when it is required, without the need to call in robot programmers and other experts.

The pick skill presented in this paper is very easy to use, and it performs well in many scenarios. The largest restriction is perhaps that it cannot be guarantied to perform well with rotational asymmetric objects. The future research plan therefore includes investigation of how a generalization can best be integrated without complicating the teaching interface significantly to the user.

ACKNOWLEDGEMENTS

This research was partially funded by the European Union under the Seventh Framework Programme project 260026 TAPAS - Robotics-enabled logistics and assistive services for the transformable factory of the future.

REFERENCES

Archibald, C. and Petriu, E. (1993). Model for skills-oriented robot programming (skorp). In *Optical Engineering and Photonics in Aerospace Sensing*, pages 392–402. International Society for Optics and Photonics.

Bøgh, S., Nielsen, O., Pedersen, M., Krüger, V., and Madsen, O. (2012). Does your robot have skills? In *Proceedings of the 43rd International Symposium on Robotics*.

Bjorkelund, A., Edstrom, L., Haage, M., Malec, J., Nilsson, K., Nugues, P., Robertz, S. G., Storkle, D., Blomdell, A., Johansson, R., Linderoth, M., Nilsson, A., Robertsson, A., Stolt, A., and Bruyninckx, H. (2011). On the integration of skilled robot motions for productivity in manufacturing. In *Assembly and Manufacturing, IEEE International Symposium on*, pages 1–9.

EUROP (2009). Robotics visions to 2020 and beyond - the strategic research agenda for robotics in europe (SRA). Technical report, EUROP.

Fikes, R. E. and Nilsson, N. J. (1972). Strips: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3):189–208.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

Gat, E. (1998). On three-layer architectures. In Kortenkamp, D., Bonnasso, P. R., and Murphy, R., editors, *Artificial Intelligence and Mobile Robots*, pages 195–210.

Hvilshøj, M. and Bøgh, S. (2011). ”Little Helper” - an autonomous industrial mobile manipulator concept. *International Journal of Advanced Robotic Systems*, 8(2).

Hvilshøj, M., Bøgh, S., Madsen, O., and Kristiansen, M. (2009). The mobile robot ”Little Helper”: Concepts, ideas and working principles. In *ETFA*, pages 1–4.

Klank, U., Pangercic, D., Rusu, R. B., and Beetz, M. (2009). Real-time CAD model matching for mobile manipulation and grasping. In *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, pages 290–296. IEEE.

Mae, Y., Takahashi, H., Ohara, K., Takubo, T., and Arai, T. (2011). Component-based robot system design for grasping tasks. *Intelligent Service Robotics*, 4(1):91–98.