Robotic Roommates Making Pancakes - Look Into Perception-Manipulation Loop

Michael Beetz, Ulrich Klank, Alexis Maldonado, Dejan Pangercic, Thomas Rühr
{beetz, klank, maldonad, pangercic, ruehr}@cs.tum.edu
Technische Universität München, 85748 Munich, Germany

Abstract—**In this paper we report on a recent public experiment that shows two robots making pancakes using web instructions. In the experiment, the robots retrieve instructions for making pancakes from the World Wide Web and generate robot action plans from the instructions. This task is jointly performed by two autonomous robots: The first robot, TUM James, opens and closes cupboards and drawers, takes a pancake mix from the refrigerator, and hands it to the seoncd robot TUM Rosie. The second robot cooks and flips the pancakes, and then delivers them back to the first robot. While the robot plans in the scenario are all percept-guided, they are also limited in different ways and rely on manually implemented sub-plans for parts of the task.**

I. Introduction

Enabling robots to competently perform everyday manipulation activities such as cleaning up, setting a table, and preparing simple meals exceeds, in terms of task, activity, behavior and context complexity, anything that we have so far investigated or successfully implemented in motion planning, cognitive robotics, autonomous robot control and artificial intelligence at large. Robots that are to perform human-scale activities will get vague job descriptions such as clean up or fix the problem and must then decide on how to perform the task by doing the *appropriate actions* on the *appropriate objects* in the *appropriate ways* in all contexts. While getting the grounding of the actions and context correctly is certainly a big research issue we will in this paper ignore it and rather concentrate on the perception-action loop that had to be implemented for pancake making (Figure 1). The latter amounts to the following two steps: i) the robots must find and recognize the ingredients and necessary tools needed for making pancakes in their environment; ii) making pancakes requires manipulation actions with effects that go far beyond the effects of pick and place tasks. The robot must pour pancake mix onto the center of the pancake oven and monitor to forestall undesired effects such as spilling the pancake mix. The robot must also push the spatula under the baking pancake in order to flip the pancake. This requires the robot to flip the pancake with the appropriate force, to push the spatula strong enough to get it under the pancake but not too strong in order to avoid pushing of the pancake off the oven.

In a recent experiment [1] we have taken up the challenge to write a comprehensive robot control program that retrieved instructions for making pancakes from the world-wide web[2], converted the instructions into a robot action plan and executed the plan with the help of a second robot that fetched the needed ingredients and set the table. The purpose of this experiment was, among others, to show the midterm feasibility of the visions spelled out in the introductory paragraph.

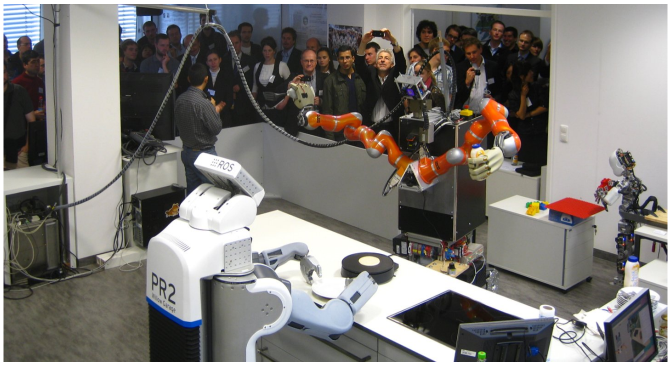


Fig. 1. TUM Rosie and TUM James demonstrating their abilities by preparing pancake for the visitors.

In the remainder of this paper we report on the perception-action loop side of this experiment and explain how we tackled what we identified as key problems. We will sketch the solutions to the individual problems, explain how they are used in the overall problem solving, and point to more detailed technical descriptions wherever possible. The paper is constructed as follows. In the first part we explain the concepts that have been used to solve the *serving* task using TUM James robot [1] and in the second part we then discuss the action of *making the pancake*, including perception and dexterous manipulation using a spatula tool on a TUM Rosie robot [2]. We conclude with a discussion of limitations and point at open research issues.

II. Perception-guided Serving

In the first part of the experiment the TUM James robot was tasked to deliver a pancake mix from a refrigerator and serve a plate and a cuttlery on a mock-up table. In this section we break up the task in three parts: i) detection and manipulation of a plate; ii) detection, recognition and manipulation of solid-state objects and iii) opening of doors and drawers with a priori unknown articulation mechanisms. All steps are percept-guided and detailed below.

A. Finding Action Related Places

In real household environments, objects are typically stored inside of cupboards and drawers and therefore the robot has

[1] **Please see the accompanying video: http://www.youtube.com/watch?v=lM_1BMIbhnA**

[2] http://www.wikihow.com/Make-Pancakes-Using-Mondamin-Pancake-Mix

to search for them before it can recognize them. Thus, to find the required objects quickly, a robot should search for the objects at their most likely places first. To do so, our robots use a semantic 3D object map of the environment, in which structured models of objects, such as cupboards consisting of the container, the door, the handle and hinges, are associated with first-order symbolic descriptions of the objects that mainly come from the robot's encyclopedic knowledge base KNOWROB-MAP [12]. The environment map also contains information about common locations of objects of daily use, also called action related places, which we used as robots' prior poses throughout the whole experiment.

B. Detecting and Picking-Up Plates

A uniformly colored plate is from the perception point of view a challenging object. It is neither tall nor flat and, thus, hard to be segmented as a cluster on the table using only depth information. To avoid this problem, we initialize the search for the plate with the query for the most circular, continuous edge in the RGB image.

This is used to calculate a good approximation of the position of the plate but with the uncertainty that the shadow might arise as the strongest edge. To prevent the latter case we cross check the position with the readings of a 3D stereo camera sensor inside the estimated volume of the plate. With this information we can estimate the height of the plate with a precision of up to $0.01m$. This approximate pose of the plate is then used to set up the approach pose for the bimanual grasp to the vicinity of the right and the left side of the plate. In the next step we start pushing grippers inwards to the center of the plate and use robot's capacitive fingertip sensors (see Figure 3, left) to detect collision with a plate. A compliant grasp is then executed, leading to a firm grip of the plate with both grippers. The compliant grasp adjusts the angle of the gripper and moves the wrist while closing it, thereby avoiding the collision of a plate with the gripper tips until both tips are closed.

For lifting, putting down and handling the plate during movements of the base, the arms are controlled in cartesian space, thus maintaining the relative position of the grippers while moving both arms.

C. Detecting, Recognizing and Picking-Up Textured Objects

Let us consider how the pancake mix is recognized. Many ingredients can be recognized based on the images on the front faces of their packages, which are often pictured in shopping websites. To use these information resources, we have downloaded the product descriptions of the web site *GermanDeli.com*, which contains about 3500 common products. The products of this website are categorized and include a picture of the front face of the package. To link the product descriptions to the robot's knowledge base the robot defines the product as a specialization of the product's category.

To make use of the product pictures for object recognition, we designed and implemented the Objects of Daily Use Finder (*ODUfinder*)[3], an open-source perception system that can deal with the detection of a large number of objects in a reliable and fast manner. Even though it can detect and recognize textured as well as untextured objects, we hereby do not report about the latter. The models for perceiving the objects to be detected and recognized can be acquired autonomously using either the robot's camera or by loading large object catalogs such as the one by *GermanDeli* into the system. Product pictures from online shops can provide good models of the texture of objects, but do not contain information about their scale. For manipulation, accurate scaling information is crucial and, in our system, it was obtained by combining the 2D image-based recognition with the information from a 3D tilting laser sensor.

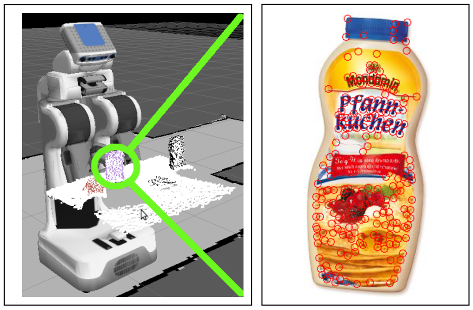


Fig. 2. Left: Region of Interest extraction using cluster segmentation and back-projection of 3D points, Right: Pancake mix with extracted SIFT features.

For obtaining a 3D pose hypothesis, we use the observation that, in human living environments, objects of daily use are typically standing on horizontal planar surfaces, or as physics-based image interpretation states it, they are in "stable force-dynamic states". The scenes they are part of can either be cluttered, or the objects are isolated in the scene. While the solution of the former is still an ongoing work, we solve the latter by a combined 2D-3D extraction of objects standing more or less isolated on planar surfaces.

This combined 2D-3D object detection takes a 3D point cloud, acquired by a stereo camera system, and a camera image of the same scene. Figure 2 left shows how the system detects major horizontal planar surfaces within the point cloud and segments out point clusters that are supported by these planes [11]. The identified clusters in the point cloud are then back-projected into the captured image to form the region of interest that corresponds to the object candidate.

The *ODUfinder* then employs a novel combination of Scale Invariant Features (SIFT) [9] for textured objects using a vocabulary tree [10], which we extend in two important ways: First, the comparison of object descriptions is done probabilistically instead of relying on the more error-prone original implementation with the accumulation of query sums. Second, the system detects candidates for textured object parts by over-segmenting image regions, and then combines the evidence of the detected candidate parts in order to infer the presence of the complete object. These extensions substantially increase the detection rate as well as the detection reliability,

[3] http://www.ros.org/wiki/objects_of_daily_use_finder

in particular in the presence of occlusions and difficult lighting conditions like specular reflections on object parts. In the current *ODUfinder* configuration, the robot is equipped with an object model library containing about 3500 objects from *Germandeli* and more than 40 objects from the *Semantic3D* database[4]. The system achieves an object detection rate of 10 frames per second and recognizes objects reliably with an accuracy of over 90%. Object detection and recognition is fast enough not to cause delays in the execution of robot tasks.

For picking up the bottle, a standard approach is used, employing a cluster based grasp planner, maximizing the coverage of the object while avoiding collision, together with a joint-space arm planner [6].

D. Detecting Handles and Opening Doors and Drawers

One of the aspects we investigated in more depth in the experiments was the opening of furniture entities. Figure 4 shows the robot opening various cupboards, drawers and appliances and generating the articulation models.

Algorithm 1: Controller for opening containers with unknown articulation model

```
Initialize pulling direction $D$ from plane normal
while Gripper did not slip off and Cartesian Error is below threshold $th = 0.035m$ do
    if Toolframe close to Robot footprint in (x,y) then
        Move base to displace Toolframe away from artificial workspace limit $L$
    Pull with the stepsize $0.05m$ in direction $d$
    Stabilize grasp using fingertip sensors (See Figure 3)
    Calculate relative transform $T$ between last trajectory pose $p_{t-1}$ and current one $p_t$
    Set pulling direction $D$ along transform $T$
Return a set of poses $P\{p_0 ... p_n\}$ representing the opening trajectory.
```

In this experiment we assume that all doors and drawers have handles which can be detected by first finding the front faces of furniture and then extracting and segmenting the clusters of pointclouds that fall in the polygonal prisms of previously detected faces [11]. To such obtained handle candidate we then fit RANSAC line and take line's geometric center to be handle's grasp point.

For opening we developed a general controller (See Algorithm 1) that employs the compliance of the TUM James's arms and the finger tip sensors to open different types of containers without a priori knowledge of the articulation model (rotational, prismatic). The robot moves the base during the process of opening containers when necessary. Lacking force sensors, the algorithm uses cartesian error of the tool coordinate frame to determine when the maximum opening is reached. The algorithm relies on the grippers maintaining a strong grasp while the arms are compliant. Like that, the mechanism that is to be opened steers the arm along its trajectory even when there is a considerable difference between the pulling and the opening direction. The controller memorizes a set of poses with the stable (aligned) grasps and returns those as an articulation model P. The controller works reliably as long as the force required to open the container is lower than the limit the friction of the gripper tips imposes.

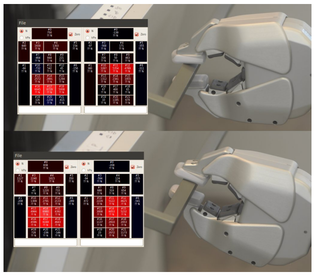


Fig. 3. TUM James's fingertip sensors (left) are used to adjust the tool frame rotation to the rotated handle (right).

A particular problem when opening the unknown containers is the possible collision of the containers with the robot. This could occur when e.g.x a drawer close to the floor is being opened and thus pulled into the robot's base. Since the articulation model is not known a priori, an actual motion planning is not possible. We thus propose a following heuristics: exclude poses whose projections of the gripper to the floor fall close to or within the projection of the robot's footprint from the allowed workspace limit L of the gripper. Like that the robot backs off and prevents the collisions.

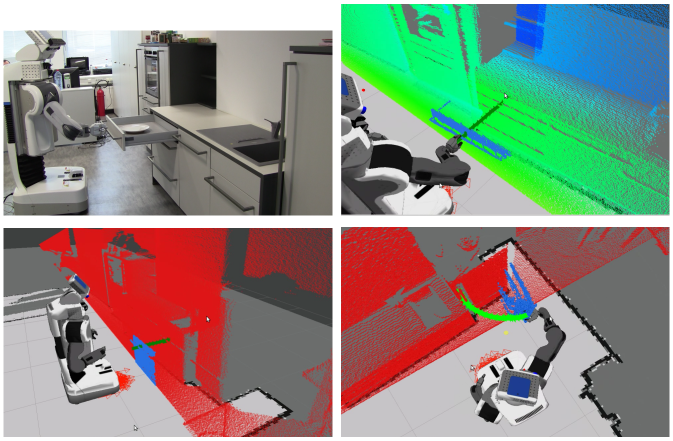


Fig. 4. Opening of various doors and generation of articulation models (green arrows).

III. Perception-guided Pancake Making

The experiment also includes the realization of a simple manipulation task that exhibits many characteristics of meal preparation tasks: cooking a pancake on a pan. Taking autonomous robot control from pick and place tasks to everyday object manipulation is a big step that requires robots to understand much better what they are doing, a much more capable perception, as well as sophisticated force-adaptive

[4] http://ias.cs.tum.edu/download/semantic-3d

control mechanisms that even involve the operation of tools such as the spatula.

In this section, we consider the process of making the pancakes by structuring it into the three steps specified in the instructions: 1) pouring the pancake mix; 2) flipping the pancake; and 3) putting the finished pancake on the plate. All steps are performed autonomously on TUM Rosie robot through the use of perception-guided control routines.

A. Pouring the Pancake Mix onto the Pancake Maker

The first step, pouring the pancake mix requires the robot to 1) detect and localize the cooking pan or pancake-maker as well as the bottle with the pancake mix, 2) pick up the pancake mix and position the tip of the bottle above the center of the pancake maker, and 3) pour the right amount of pancake mix onto the pancake maker. We will discuss these steps below.

1) Detecting and Localizing the Relevant Objects: The robot performs the detection and localization of the relevant objects using object type specific perception routines. The black color in combination with the metallic surface of the pancake maker makes the readings of time-of-flight sensors very noisy, and the heat of the pancake maker requires particularly high reliability of operation. On the other hand, the accuracy demands for successful action execution are less for the destination of the pouring action (roughly in the center of the object) than for successfully grasping an object. One basic principle that we used for the realization of perceptual mechanisms is that we apply a team of context specific perception mechanisms rather than aiming for a single but overly general perception mechanism [5].

Thus for the detection and rough localization of the pancake maker we provided the robot with a previously calibrated planar shape model of the top plane of the pancake maker and used this to localize the pancake maker. For matching in the online phase we used the method proposed by Hofhauser et al. [4] on images of a RGB-camera. This method is very fast, and gives an accurate result in less than half a second which is already cross checked over the second camera in the stereo pair.

The method for localizing the pancake mix also exploits the task context. Because the pancake mix is delivered by the other robot, it is reasonable and useful to assume that the pancake mix is placed where it is easily reachable by the robot and second that the location is approximately known. Thus, the robot uses a perception mechanisms that exploits these regularities and confines itself to finding a point cluster at the approximate position with the approximate dimensions of the pancake mix. This method is efficient as well as reliable and accurate enough to pick up the pancake mix (see [8] for details on the cluster detection). The pancake-mix is grasped with a power grasp coupled with a validation of the grasp success, which we discuss later.

2) Pouring the Adequate Amount of the Pancake Mix: In order to make pancakes of the appropriate size the robot has to pour the right amount of pancake mix onto the pancake maker. This is accomplished by estimating the weight of the mix that has been poured onto the pan. After successfully lifting the pancake-mix, the weight of the bottle is estimated using the measured joint torques.

To pour the pancake mix onto the pancake maker, the robot estimates the height of the top of the pancake mix bottle and uses this information to determine the right pose of the robot hand. The pouring time is adjusted using a hand crafted linear formula with the weight of the bottle as a parameter.

In order to validate the success and estimate the effects of the pouring action the robot applies a blob detection with the image region corresponding to the pancake maker as the search window. After a color-based segmentation, all components which are not similar in intensity to the pan are considered as pancakes or pancake parts. The noise removal on the segmentation results then gives the robot a sufficiently good model of the position (relative to the pan) and form of the pancake. This perception task is performed in real time and also works in the presence of the spatula.

B. Flipping the Pancake

The key steps in flipping the pancake are 1) to grasp and hold the spatula sufficiently well to use it as a tool, 2) to calibrate the spatula with the hand such that the robot can control and determine the accurate pose of the spatula through its internal encoders, and 3) to perform an adaptive stiffness control to push the spatula under the pancake without pushing the pancake off the pancake maker.

1) Picking Up and Holding the Spatula Properly: The spatula has been modified to give it a broader handle, so that the robot can hold it securely in its oversized hand.

The spatula is detected, localized, and approximately reconstructed through the use of our 3D sensors, in this case the ToF camera. To match the surface of the spatula with the current sensor data we use the method proposed by Drost et al. [3]. To train the object we observed it once on a table without clutter and took the result of a 3D cluster segmentation as the surface template.

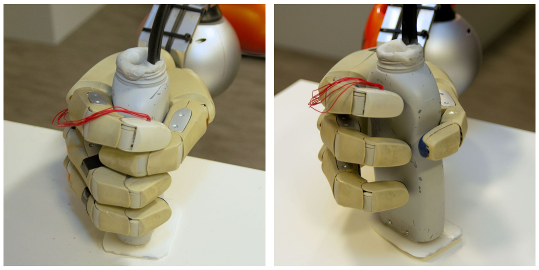


Fig. 5. A supervision system detects good (left) and bad (right) grasps.

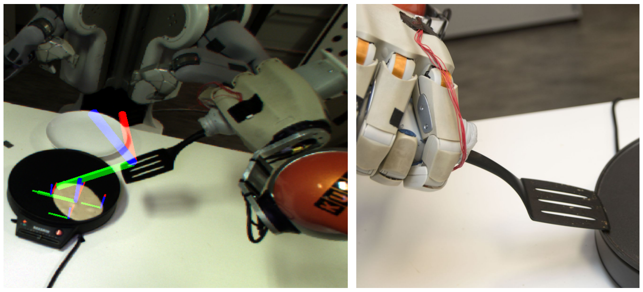
To deal with uncertainty in perception, that can lead to sub-optimal grasps, a simple system is used to evaluate grasp quality, using measured finger positions and torques. To this end, the data vector distances between current measurements and a known good and known bad grasps are calculated and used as a quality values. A low quality score leads to a grasp retry, and given another low value, the object is localized again and the complete grasping action is repeated.

Figure 5 shows a grasp that fulfills these properties on the left, and a failed one on the right. Grasps may fail due to unexpected contacts with parts of the object or delays in the control of the fingers.

2) Controlling the Spatula as an End Effector: To lift the pancake successfully, the robot should treat the spatula as a body part rather than an object that has to be manipulated. This means, the kinematic model of the arm is extended to include the spatula, and the algorithms used to detect collisions with the hand are modified to detect collisions on the spatula.

To use the spatula as a tool, its relative position to the hand has to be known precisely after the robot has grasped it. For this effect, the robot performs an online calibration using the same method that is used to localize the pancake maker. In this case the planar assumption is valid for the complete top part of our tool. To gain a higher accuracy, the matching is applied several times, always matching on both stereo images and validating the consistency of the results. The results from all matchings are taken as a set of hypotheses, which are used to calculate a robust mean value in translation and rotation. Figure 6 shows the position in which the robot holds the spatula (left) and the intrinsic view of the robot in visualization (middle) and the camera image at this point in time (right).

3) Movement Control of the Pancake Tip: To flip a pancake with a spatula, the robot must push the spatula under the center of the pancake without pushing the pancake off and deforming or destroying it. To do so, the robot pushes the spatula down until it touches the pan and the tip is parallel to the surface. The robot moves the spatula in a straight line between the point of contact with the pan and the center of the pancake.



(a) Approach the pancake (reference frames overlayed).

(b) First contact of the spatula with the pan.

Fig. 7. Flipping the pancake.

Figure 7(b) shows the moment when the robot has lowered the tool until it touched the pan. This contact produces measurable force changes in the fingers, so that the event can be reliably detected.

In order to correctly detect the contact of the tip with the pan, a band pass filter is applied to the 12 torque streams coming from the hand at 1kHz, eliminating the constant torques for holding the object and the high-frequency noise from the motor controllers. We calculate the dot product of the filtered torque vectors with a template vector, and a high value is measured shortly after the collision.

After touching the pan, its height is known precisely, and the rest of the movements take this into account.

4) Picking and Turning the Pancake: The trajectory to pick up the pancake, lift and turn it was taught by demonstration and is only parametrized with the pancake's position, corrected by the newly estimated height of the pan. The spatula has to be positioned under the pancake, then the pancake can be lifted. Afterwards, the pancake has to be turned and dropped back to the pan. The pancake tends to stick at this stage to the spatula, which requires the robot to apply various accelerations to the spatula to separate the pancake again. This introduces uncertainty about the position of the pancake after this action.

5) Checking the estimated Result: Dropping the pancake back onto the pan can have three possible outcomes: 1) the pancake falls pack to its original position in the center of the pan, 2) the pancake drops a little bit off the center (usually still on the pan) and 3) the pancake keeps sticking on the spatula. The first two cases can be detected by re-detecting the pancake on the pan and the third case follows if the pancake cannot be detected on the pan anymore. While case one does not require further actions, the second case is corrected by centering the pancake with the spatula again. In the third case, the robot continues moving the arm up and down until the pancake drops.

C. Putting the Pancake onto a Plate

After TUM James placed a plate close to the pan, TUM Rosie can move the pancake to the plate. The pancake is moved from its current position to the center of the plate. As the friction of a done pancake is too low to lift it, the robot pushes the pancake off the pan on to the plate.

IV. Conclusions and Research Issues

In this paper we have presented an experiment in which robots retrieved a simple instruction for a meal preparation task from the web and semi-automatically translated it into a robot plan that was jointly executed by the robots. The experiment was a feasibility study. Many aspects have been solved very specifically and some actions have been hand-coded.

While the robot plans in the scenario are all percept-guided they are also in many ways limited, use shallow, heuristic and ad-hoc solutions, and are overspecialized to the scenario. These limitations point us at fundamental research and technological questions that need to be answered in order to accomplish these kinds of everyday manipulation tasks in more general, flexible, reliable, and principled ways. In addition, the issues identified in [7] apply to our control task.

We believe that robots that are to scale towards human-scale activities require knowledge-enabled decision making and the control systems need to be knowledge intensive. We have seen in the paper that we have applied many specific mechanisms to do the job. We believe that this is not due to the state of the art of robot control but rather that more future advanced control system will also employ such specific methods. Generality and competence will lie in selecting the approprate mechanisms for large ranges of environment and task contexts.

This work was supported by CoTeSys (Cognition for Technical Systems) cluster of excellence at TUM and by MVTec Software GmbH, München.

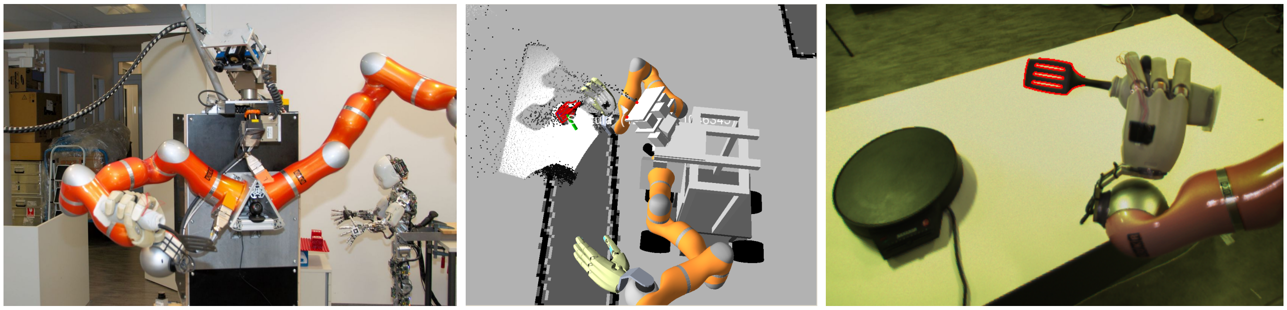


Fig. 6. Calibration of the spatula.

References

[1] Personal robot 2. URL http://www.willowgarage.com/pages/pr2/overview.

[2] Tum rosie robot. URL http://ias.cs.tum.edu/research-areas/robots/tum-robot.

[3] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010.

[4] Andreas Hofhauser, Carsten Steger, and Nassir Navab. Edge-based template matching with a harmonic deformation model. In *Computer Vision and Computer Graphics: Theory and Applications - VISIGRAPP 2008*, volume 24 of *Communications in Computer and Information Science*, pages 176–187, Berlin, 2009. Springer-Verlag.

[5] I. Horswill. Analysis of adaptation and environment. *Artificial Intelligence*, 73:1–30, 1995.

[6] Mrinal Kalakrishnan, Sachin Chitta, Evangelos Theodorou, Peter Pastor, and Stefan Schaal. STOMP: Stochastic Trajectory Optimization for Motion Planning. In *International Conference on Robotics and Automation*, 2011.

[7] C. Kemp, A. Edsinger, and E. Torres-Jara. Challenges for robot manipulation in human environments. *IEEE Robotics and Automation Magazine*, 14(1):20–29, 2007.

[8] Ulrich Klank, Dejan Pangercic, Radu Bogdan Rusu, and Michael Beetz. Real-time cad model matching for mobile manipulation and grasping. In *9th IEEE-RAS International Conference on Humanoid Robots*, pages 290–296, Paris, France, December 7-10 2009.

[9] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. ISSN 0920-5691.

[10] David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2161–2168, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2597-0.

[11] Radu Bogdan Rusu, Ioan Alexandru Sucan, Brian Gerkey, Sachin Chitta, Michael Beetz, and Lydia E. Kavraki. Real-time Perception-Guided Motion Planning for a Personal Robot. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4245–4252, St. Louis, MO, USA, October 11-15 2009.

[12] Moritz Tenorth, Lars Kunze, Dominik Jain, and Michael Beetz. Knowrob-map – knowledge-linked semantic object maps. In *Proceedings of 2010 IEEE-RAS International Conference on Humanoid Robots*, Nashville, TN, USA, December 6-8 2010.