Combining Analysis, Imitation, and Experience-based Learning to Acquire a Concept of Reachability in Robot Mobile Manipulation

Freek Stulp[‡], Andreas Fedrizzi[‡], Franziska Zacharias[§], Moritz Tenorth[‡], Jan Bandouch[‡], Michael Beetz[‡]

***Abstract*— Analytic modeling, imitation, and experience-based learning are three approaches that enable robots to acquire models of their morphology and skills. In this paper, we combine these three approaches to efficiently gather training data to learn a model of reachability for a typical mobile manipulation task: approaching a worksurface in order to grasp an object.**

The core of the approach is experience-based learning. For more effective exploration, we use capability maps [20] as analytic models of the robot's dexterity to constrain the area in which the robot gathers training data. Furthermore, we acquire a human model of reachability from human motion data [17] and use it to bias exploration. The acquired training data is used to learn Action-Related Places [16]. In an empirical evaluation we demonstrate that combining the three approaches enables the robot to acquire accurate models with far less data than with our previous exploration strategy.

I. Introduction

To acquire models of its capabilities and limitations, a robot has several options: **1) Analysis**: a designer specifies explicit analytic models for the robot; **2) Imitation**: the robot learns models by observing the behavior of humans **3) Experience-based Learning**: the robot learns models by observing its *own* interaction with the world in trial-and-error exploration. Each of these approaches have specific advantages and drawbacks, which we consider in the context of a typical task from mobile manipulation: determining base positions from which grasping an object will likely succeed. The articulated B21 robot we use for this task, as well as its simulation in Gazebo, is depicted in Fig. 1

Using analytical models of the robot and the world is still the dominant approach to manipulation, and "when the world's state is known and consists of rigid body motion, it's hard to imagine something better" [9]. For instance, robot kinematics are well-understood, and a wealth of methods and algorithms are available to determine if an object is reachable, i.e. if the end-effector can be brought into a position where the object can be grasped. However, considering kinematics alone does not address practical issues such as: What happens if the state estimation yields uncertainties? Or if the target object is not within the field of view? Or if the robot has a limited set of controllers that cannot reach to all positions? Such questions become crucial when performing everyday manipulation tasks, such as navigating to a work surface in order to fetch a cup.

Humans perform such manipulation tasks routinely and efficiently, and seem to take all these practical issues into account effortlessly. Acquiring a model by observing human behavior can be an alternative to explicit analytic models. This imitation approach drastically reduces the need for exploration. Of course, the robot's morphology is assumed to be similar enough to that of the human, in order to make such models portable between them.

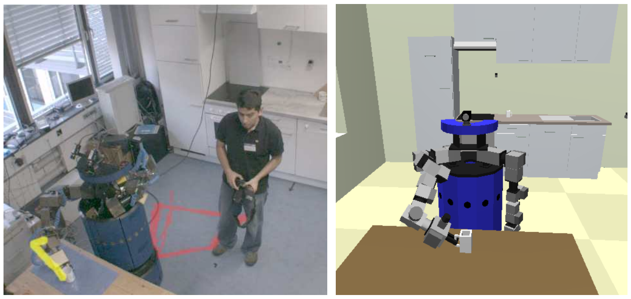


Fig. 1. Left: Our B21 mobile robot grasps a cup during a public demonstration. Right: Gazebo model of our real robot.

To tailor the model to its own skills, a robot should consider its *own* experience that arises from trial-and-error exploration of the environment. Using experience-based learning guarantees that the models are grounded in actual interaction between the robot and the world. Unfortunately, gathering data is often expensive, and the time required to learn an accurate model can be prohibitively long.

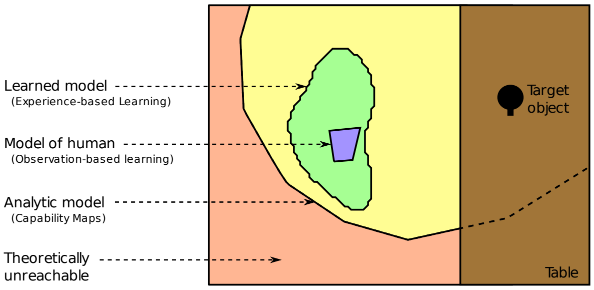

Learned model
(Experience-based Learning)
Model of human
(Observation-based learning)
Analytic model
(Capability Maps)
Theoretically unreachable
Target object
Table

Fig. 2. Three ways of acquiring a model of appropriate base positions for manipulation.

In this paper, we combine the aforementioned approaches to gather data to efficiently learn areas in which positioning the base will lead to successful grasping of objects, as

[‡]Intelligent Autonomous Systems Group, Technische Universität München, Department of Informatics, Garching bei München, Germany. This research has been partially funded by the CoTeSys cluster of excellence (Cognition for Technical Systems, `http://www.cotesys.org`), part of the Excellence Initiative of the DFG, and the DFG ActAr project.

[§]DLR - German Aerospace Center, Institute of Robotics and Mechatronics, Wessling, Germany. This research has been partially funded by the EC Seventh Framework Programme (FP7) under grant agreement no. 216239 as part of the IP DEXMART.

depicted in Fig. 2. The core of the work is experience-based learning, in which the robot navigates to different positions, and observes whether grasping a target object succeeds or fails from these positions. An example of randomly sampled base positions from which grasping succeeded or failed is depicted to the left in Fig. 3.

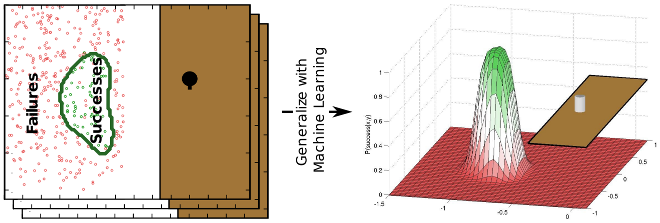


Fig. 3. Learning a concept of action-related place by generalizing over observed experience with machine learning. Green dots mark successful executions of the navigate-reach-grasp action sequence, while red dots mark failures.

Within this *experience-based learning* approach, we transform our previous random sampling into more goal-directed exploration by 1) constraining the search space with a capability map [20], which is a compact *analytic model* of the robot's reachable workspace; 2) guiding exploration by *imitating* observations of humans performing a similar task [17]. Combining experience-based learning, imitation and analytic models enable robots to exploit their respective advantages, and quickly gather relevant experience. The robot then generalizes over this experience using machine learning techniques. This yields a so-called *action-related place* (ARPLACE), which is a probability map that maps robot base locations to grasp success probabilities [16], as depicted to the right in Fig. 3.

The main contributions of this paper are: 1) Bounding robot exploration with theoretical kinematic limits, given by the capability map of the robot; 2) Evaluating the use of human motion data to guide exploration; 3) An empirical evaluation of the effect of these exploration strategies on the accuracy and speed of the learning process; 4) Deriving a concept of action-related place from the acquired data.

The rest of this paper is structured as follows. In the next section, we discuss related work. We describe how capability maps are computed, and how human data is acquired in Section III and IV respectively. In Section V, we describe how the analytic model and the human data are used to improve exploration for our experience-based learning approach. Learning more general models of action-related place from the resulting data is discussed in Section VI. An empirical evaluation is presented in Section VII, and we conclude with Section VIII.

II. Related Work

Okada et al. denote a good base placement for grasping a `spot` [13]. Different `spots` are defined for different tasks, such as manipulating a faucet, a cupboard, and a trashcan. In their work, `spot` information is hand-coded. Berenson et al. deal with the problem of finding optimal start and goal configurations for manipulating objects in pick-and-place operations [3]. They explicitly take the placement of the mobile base into account. As they are interested in the optimal start and goal configurations, instead of a probabilistic representation, this approach does not enable least-commitment planning. Diankov et al. use a model of the reachable workspace of the robot arm to decide where the robot may stand to grasp an object and to focus the search [7]. However, uncertainties in robot base position or the object are not considered, and can thus not be compensated for.

Recently, similar methods to the ones presented in this paper have been used to determine successful grasps, rather than base positions for grasping. For instance, Detry et al. determine a probability density function (PDF) that represents the graspability of specific objects [6]. This PDF is learned from samples of successful robot grasps, which are biased by observed human grasps. However, this approach does not take examples of failed grasps into account. As we shall see in Section VI, the distance between a failed and a successful grasp can be quite small, and can only be determined by taking failed grasps into account. Our classification boundaries in Section V are similar to Workspace Goal Regions [4], except that our boundaries refer to base positions, whereas in [4], they refer to grasp positions. Also, we generalize over these boundaries with a Point Distribution Model, and use it to generate a probabilistic concept of successful grasp positions.

Kuipers et al. [11] present a bootstrapping approach that enables robots to develop high level ontologies from low level sensor data including distinctive states, places, objects, and actions. These high level states are used to choose trajectory-following control laws to move from one state to another. Our approach is exactly the other way around: given the manipulation and navigation skills of the robot (which are far too high-dimensional to learn with trajectory-following control laws), learn places from which these skills (e.g. grasping) can be executed successfully.

Learning success models is a form of pre-condition learning. In robotics, the focus in pre-condition learning is on grounding pre-conditions in robot experience. For instance, 'Dexter' learns sequences of manipulation skills such as searching and then grasping an object [8]. Declarative knowledge such as the length of its arm is learned from experience. Learning success models has also been done in the context of robotic soccer, for instance learning the success rate of passing [5], or approaching the ball [15]. Our methods extend these approaches by explicitly representing the region in which successful instances were observed, and computing generalized success models from these regions.

III. Capability Maps

In this section, we describe how capability maps [20] model the reachable workspace of a robot arm. In this paper, we use a B21 robot equipped with two 6-DOF PowerCube lightweight arms from Amtec Robotics, and parallel grippers [2].

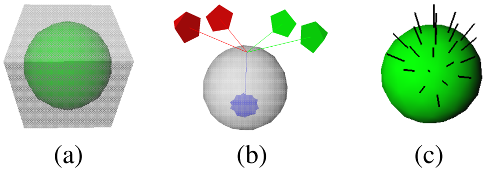


Fig. 4. (a) A sphere inscribed into the cube; (b) Exemplary frames for a point on the sphere; (c) Valid inverse kinematics solutions on a sphere.

First, the theoretically robot arm workspace is enveloped by a cube, and then subdivided into equally-sized cubes. A sphere is inscribed in each cube, and n points are uniformly distributed on this sphere (Fig. 4 (a), (c)). Frames (Fig. 4 (b)) are generated for each point on the sphere and serve as the target tool center point (TCP) for the inverse kinematics of the robot arm. The normal to the sphere at a point determines the z-axis (blue arrow) of the TCP frame. The orientation of the x- and y- axis is sampled equidistantly. If a solution is found for the specific frame then the frame is reachable, as visualized by a black line perforating the sphere at the corresponding point. The sphere diameter was chose empirically as a trade-off between memory consumption and being small enough so that the reachability substructure does not change.

The spheres visualize the reachability for a region and are therefore called reachability spheres. The reachability index D is the percentage of the points on the sphere that are reachable [20]. Fig. 5 shows examples of reachability spheres and corresponding values of D for the used robot arm.

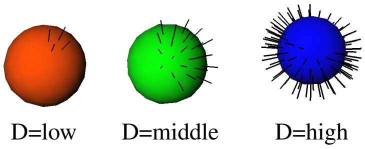


Fig. 5. Spheres with different reachability indices D.

The reachability sphere map is the aggregation of all spheres. This data structure can be used to gather statistics about the workspace, as well as to perform planning. It thus describes the capabilities of robot arms. Furthermore, this model can be used to visualize and inspect the orientation-dependent reachability across the workspace, and to approximate the shape of the robot arm workspace. The distribution of the reachability index D throughout the workspace is depicted in Fig. 6. The reachability sphere map is cut along two different planes. For reference the direction of the z-axis of the first link is shown as a gray arrow. In the blue region the reachability index is highest. In Fig. 6, the blue area is unfortunately useless for manipulation, as the robot is either in collision with itself or the environment if it tries to make use of it. For the arm used in this work, the reachability index is relatively low throughout the workspace, with D_{max}=52%.

Fig. 6 is a visualization of the question: given the position of my base, which positions can I reach with my end-effector? In this paper, we use the capability map to answer the 'inverse' question: given the position of the target object, and therefore the desired position of my end-effector, from which base positions can I achieve this end-effector position? In Fig. 7, the answer to this question is visualized for a specific target object position.

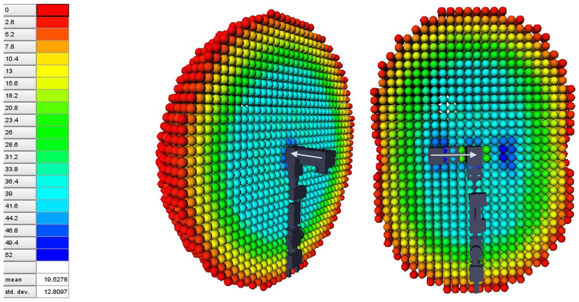


Fig. 6. The model of the PowerCube arm and the reachability index D across the workspace.

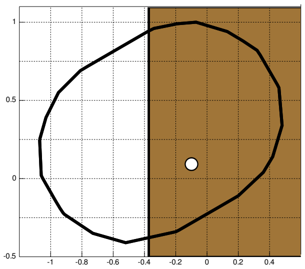


Fig. 7. Valid robot placements given a specific object position, determined using the capability map.

Summarizing, the capability map is an analytic model of the robot kinematics, and an upper bound on reachability. We shall use this upper bound to constrain exploration for experience-based learning in Section V-A.

IV. Human Activity Data

We observe humans in a sensor-equipped kitchen environment while performing tasks similar to the mobile manipulation task of our robot. Video data from four ceiling-mounted cameras is used as input to a markerless motion capture system [1]. The system is capable of tracking human manipulation actions with a 51 DOF articulated human model, as depicted in the left and central image of Fig. 8. Tracking is unintrusive and unconstrained, enabling the humans to perform as natural as possible. The video data is complemented by object detections by Radio Frequency IDentification (RFID) sensors, e.g. when placing a cup on the table.

To use the data to guide exploration, the system has to select relevant human poses from the continuous stream of tracked motions. First, the observed positions are loaded into a knowledge processing system and clustered with respect to their Euclidean distance. These clusters are represented as "places" in the knowledge base. Then, based on the RFID tag detections, the system learns a mapping from action properties (in our case an object being manipulated and

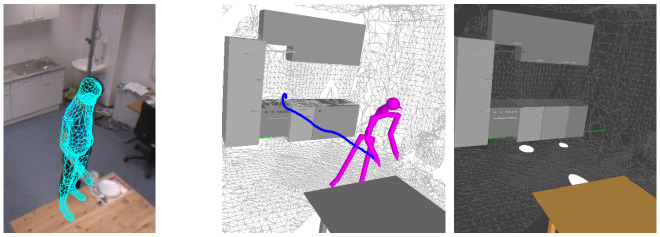


Fig. 8. Left: Tracking a human placing a cup on the table using our markerless fullbody tracker. Center: Playback of the action trajectory to visualize the recoded data. Right: The circle in the front corresponds to the learned place for action “place a cup on the table”.

its position) to human “places”. Given such a model, it is possible to either obtain a place for an action (like a place for picking up a cup), or to find the most probable action given an observation. The right image of Fig. 8 gives an example of learned action places in our knowledge base, with the circle in the front marking places where humans were standing when putting objects onto the table. A more detailed description of the knowledge processing system and the acquisition of action models as abstract specifications of action-related concepts is given in [18].

Summarizing, the human activity data is a model of places from which humans perform manipulation actions. In Section V-B, we demonstrate how this model is used to guide exploration.

V. Guiding Exploration

In this section, we first describe the default hand-coded exploration strategy used in our previous work [16], and then demonstrate how analytic models and imitation - capability maps and the human data respectively - are used to improve this exploration.

The default approach is to define a large rectangular area relative to the cup positions, and sample uniformly within this area. The robot is always oriented in the direction to the cup, as our controller can currently only deal with this relative positioning between cup and robot. The size of the rectangular area was essentially our generous estimation of the theoretical limits of the robot’s reachability. Fig. 9(a) depicts 300 training examples that were gathered for one cup position using this approach.

To generalize over these examples, we compute a classification boundary around the successful samples using Support Vector Machines (SVM). We used a Gaussian kernel with σ=0.03, and cost parameter C=20.0. Fig. 9(d) depicts the percentage of correctly classified examples over time, depending on the number of examples available for training the SVM. This measure was computed on a separate test set, which contained 150 additional examples. As expected, the accuracy of the model increases quickly in the beginning, but levels off as more examples become available.

A. Exploration with capability maps

The exploration strategy that uses capability maps also samples within the same rectangle, but immediately discards any positions from which the target is deemed unreachable. Querying the capability map for reachability is done by computing the pose P_{target} of the target object relative to the origin of the robot’s manipulator. Because the origin of the manipulator is also the origin of the capability map, P_{target} can be used directly by the capability map to compute whether the target object is reachable from the sampled position. Therefore the capability map determines in which of the equally-sized cubes P_{target} is located. For the resulting cube it is investigated if the inscribed sphere has solvable sphere points. If this is the case, the target object can be theoretically reached from the sampled base position. Otherwise the target object is unreachable. Using this approach, 132 out of 300 examples (44%) are thus excluded from the data that was used by the default exploration strategy. Fig. 9(a) and Fig. 9(b) visualize the reduction of required training examples. Note that there are many failed examples within the capability map. This is because the capability map only considers the reachability of a position, given arm kinematics; the failed examples are caused by collisions with the table, the fact that the robot only approaches the table perpendicularly, and the restrictions of our vector-field controller. Through experience-based learning, we refine the theoretical upper bound to take these practical restrictions into account. Because the discarded positions do not influence the classification boundary computed by the SVM, a model with the same accuracy can be learned with only 56% of the data. This is demonstrated by the graph in Fig. 9(d).

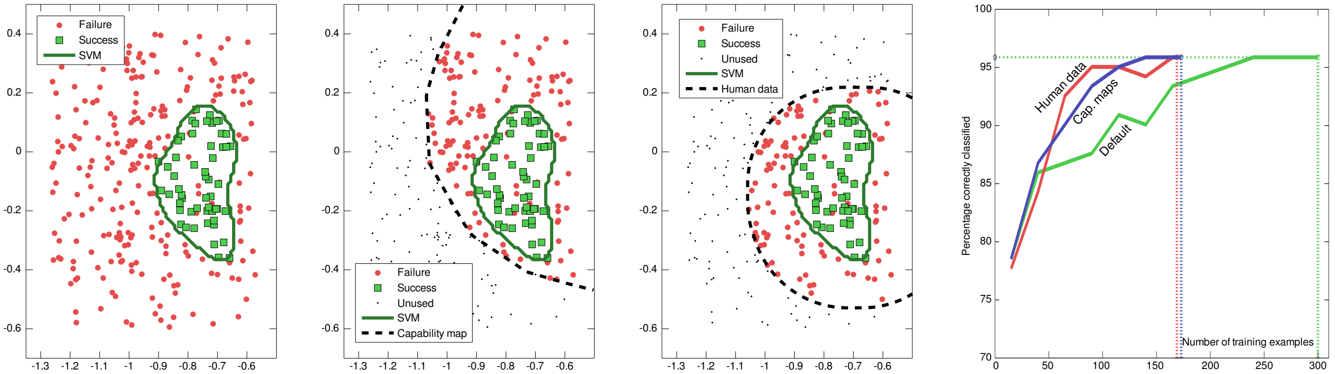
B. Exploration with human data

In Fig. 9(c), the area from which humans reach for the object lies almost completely within the area from which the robot successfully grasps. Therefore, we use the human data to bias the exploration during experience-based learning. That is, we start exploring in the region where humans stand when grasping, and incrementally explore from there.

Fig. 10 depicts the histogram of the distances of successful and failed robot grasp attempts to the convex hull around the human data. The image suggests that our approach is effective. Examples that lie in the convex hull have a distance of 0. Above a distance of approximately 0.3m from the human data, the robot does not succeed at grasping the object, although our default strategy still samples almost half of the examples beyond this distance.

We have used this histogram to implement a simple exploration strategy. First of all, the robot samples only within the convex hull enclosing the human data. Then, it samples within areas that are $5cm$, $10cm$, . . . away from the convex hull, as depicted to the left in Fig. 10. At each increment, the histogram is regenerated with the novel data, which contains successful and failed grasp attempts, as in Figure 10 (right). As soon as there are no more successful samples in the last histogram bin, data gathering is halted. In this example, this is the case after 6 increments corresponding to a distance of 0.325m to the convex hull enclosing the human data.

For an objective comparison, we use the same samples as in the default strategy. The failed examples which are never considered with this strategy (140, which is 47% of the data



(a) Default exploration.

(b) Exploration with CapMaps.

(c) Exploration with human data.

(d) SVM accuracy over time.

Fig. 9. (a)–(c): Data used in the exploration phase. In these figures green squares with a black border are positions from which the target cup was successfully grasped. Red circles represent failed attempts. Examples discarded in the different exploration strategies are depicted as black dots. (d): The number of correctly classified examples on a separate test set, depending on the size of the training set, for each of the exploration strategies.

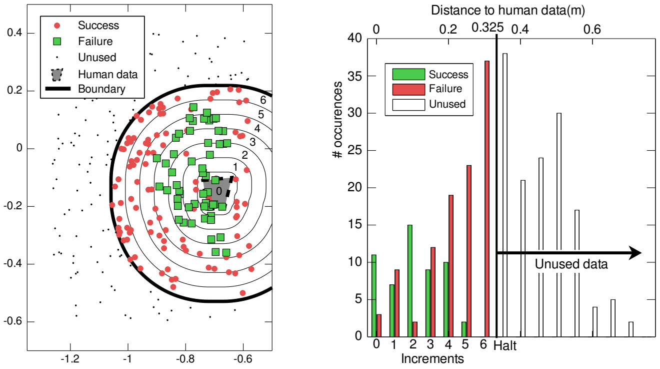


Fig. 10. Left: the human data, and the data gathered by the robot. The boundaries represent isolines of equal distance to the convex hull around the human data. Right: the histogram of the distances of the grasp attempts of the robot to this convex hull.

used in default exploration) and the percentage of correctly classified test-examples as more training-examples become available are depicted in Fig. 9(c) and 9(d) respectively.

The exploration algorithm just described is quite simple and has some disadvantages. For instance, it could never discover multi-modal distributions of successful examples if failures lie in between. However, within the scope of this paper, it is our goal to demonstrate that human data can in principle be used to bias and speed up robot exploration, not to provide the most complete and accurate algorithm to actually perform the exploration.

VI. Generalizing Reachability: ARPlace

A reduction from 300 to 132 or 140 examples might, at first, not seem to be a dramatic improvement. However, learning a SVM classification boundary for only one cup position is not very general. Therefore, we gather data and learn classification boundaries for 12 different cup positions, and generalize over them using a Point Distribution Model[1]. As we shall see in Section VII, using capability maps and human data reduces the total number from 3600 to 2213 and 2034 respectively, which saves executing over 1000 action sequences.

A. Generalized Success Model

After having gathered data for 12 different cup positions, depicted to the right in Fig. 11, we learn the classification boundary for each of them by training a SVM. Fig. 11 depicts the resulting classification boundaries for different configurations of task-relevant parameters, i.e. 12 different target cup positions. These have been learned from the data gathered with the default strategy. As we saw in Fig. 9, the boundaries acquired by using the other exploration strategies are exactly the same, or very similar. In Section VII, we compare the difference between these models.

From the classification boundaries, it becomes clear that the cup can be grasped from a larger area if it is closer to the edge of the table, i.e. the area from which reaching succeeds is different in each column. The transformation along the edge of the table should theoretically lead to exactly the same boundaries in each column, but as different data is sampled for each cup position, there are slight differences.

With these 12 classification boundaries, the robot can predict successful and failed grasps for 12 different cup positions. The next step is computing a so-called Point Distribution Model (PDM), that enables the robot to interpolate between these boundaries, and also compensates small differences between them.

As input a PDM requires n points that are distributed over the contour. We distribute 20 points equidistantly over each boundary, and determine the correspondence between points on different boundaries by minimizing the sum of the distances between corresponding points, while maintaining order between the points on the boundary.

Given the aligned points on the boundaries, we compute a PDM. Although PDMs are most well-known for their use in computer vision [19], we use the notation by Roduit et al. [14], who focus on robotic applications. First, the 2D boundaries are merged into one 40x12 matrix $\mathbf{H}$, where the

[1] This section is described in more detail in [16]; we summarize it here to demonstrate that the specific classification boundaries can be used to construct more general models.

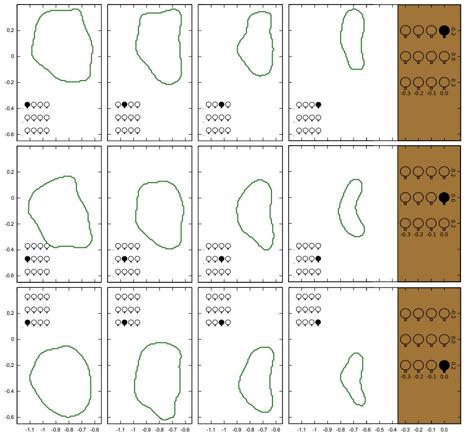


Fig. 11. Classification boundaries for 12 cup positions, visualized with the black cup in each of the 12 graphs. To save space, the table on which the cup is placed is only shown in the right-most graphs.

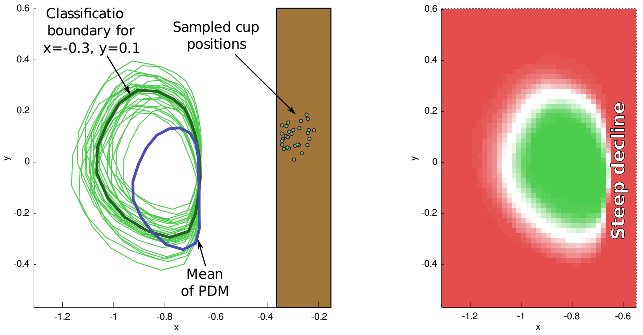
columns are the concatenation of the x and y coordinates of the 20 points along the classification boundary. Each row represents one boundary. The next step is to compute $\mathbf{P}$, which is the matrix of eigenvectors of the covariance matrix of $\mathbf{H}$. Given $\mathbf{P}$, we can decompose each boundary $\mathbf{h}_k$ in the set into the mean boundary and a linear combination of the columns of $\mathbf{P}$ as follows $\mathbf{h}_k = \overline{\mathbf{H}} + \mathbf{P} \cdot \mathbf{b}_k$. Here, $\mathbf{b}_k$ is the so-called deformation mode of the k^{th} boundary. The deformation modes are stored together in the matrix $\mathbf{B}$. By inspecting the eigenvalues of the covariance matrix of $\mathbf{H}$, we determined that the first 2 components already contain 96% of the deformation energy [16]. Therefore, we use only the first 2 deformation modes, without losing much accuracy. This is the Point Distribution Model.

The advantage of the PDM is not only that it substantially reduces the high dimensionality of the initial 40D boundaries. It also allows us to interpolate between them in a principled way using only two deformation parameters. The PDM is therefore a compact, general, yet accurate model for the classification boundaries. The final step of model learning is to relate the specific deformation of each boundary (contained in $\mathbf{B}$) to the values of the task-relevant parameters (like the x and y coordinates of cup position) that are varied during data collection. Since the correlation coefficients between the first and second deformation modes and the task relevant parameters $\mathbf{T}$ are 0.99 and 0.97 respectively, we simply compute the linear relation between them with $\mathbf{W} = [\mathbf{1}\ \mathbf{T}]/\mathbf{B}^T$.

B. *Action-Related Place:* ARPLACE

In this section, we describe how appropriate ARPLACEs for manipulation are determined on-line. Due to sensor noise and other factors that influence the state estimation, the task relevant parameters can never be known exactly, and uncertainty must be modeled. The belief state therefore also associates a covariance matrix with each position: $\begin{pmatrix}\sigma^2_{xx} & \sigma^2_{yx} \\ \sigma^2_{xy} & \sigma^2_{yy}\end{pmatrix}$, computed by our vision-based object localization module [10].

Because of this uncertainty, it does not suffice to compute only one classification boundary given the most probable position of the cup as the ARPLACE from which to grasp. This might lead to a failure if the cup is not at the position where it was expected. To solve this problem, we use a Monte-Carlo simulation to generate a probabilistic advice on where to navigate to grasp the cup. This is done by taking 100 samples from the Gaussian distribution of the cup position, given its mean and covariance matrix. This yields a matrix of task relevant parameters $\mathbf{t}_s = [\mathbf{x}_s\ \mathbf{y}_s]$. The corresponding classification boundaries $\mathbf{h}_s$ are computed for the samples. In Fig. 12(a), 30 out of the 100 boundaries are depicted. These were generated from the task relevant parameters x=-0.3, y=0.1, σ_{xx}=σ_{yy}=0.05, σ_{xy}=σ_{yx}=0.



(a) Sampled classification boundaries ($\mathbf{h}_s$).

(b) Discretized relative sum of the boundaries.

Fig. 12. Monte-Carlo simulation of classification boundaries to compute ARPLACE.

We then generate a discrete grid in which each cell measures $2.5 \times 2.5cm$, and compute the number of classifications boundaries that classify this cell as a success. Dividing the result by the overall number of boundaries yields the probability that grasping the cup will succeed from this position. The corresponding distribution, which takes the uncertainty of the cup position into account, is depicted in Fig. 12(b). A similar distribution in 3D (and for different values of the task-relevant parameters) is depicted in Fig. 3.

VII. EMPIRICAL EVALUATION

a) Number of examples required for exploration: In Table I we list the number of examples used by the different exploration strategies over all 12 cup positions. The capability maps and human biasing filter out 38.5% and 43.5% of the examples, but achieve the same model accuracy.

Exploration	Size of	% less than	% correctly
strategy	training set	training set	classified
Default	3600	0	94.0
Cap. Maps	2213	38.5	93.8
Human data	2034	43.5	94.0

TABLE I
OVERALL RESULTS OF THE DIFFERENT EXPLORATION STRATEGIES

b) Evaluation of ARPLACE*:* Finally, we evaluate the exploration strategies in the context of using ARPLACE to compute base positions for robust mobile manipulation. In this evaluation, the position to which the robot navigates is simply the position for which ARPLACE returns the highest probability that grasping the target object will succeed, as in [16]. We compare this strategy to our previous hand-coded implementation FIXED, which always navigates to a location that has the same relative offset to the target object, whilst at the same time taking care not to bump into the table.

Fig. II depicts the average percentage of successful grasp attempts, averaged over three different cup positions. The same experiment was performed for three different uncertainties about the cup position, represented by their covariance matrices $C(p_o)$. Overall 2626 evaluation experiments were performed, resulting in more than 400 experiments per cell entry. A first conclusion is that using ARPLACE always performs better than FIXED. As expected, the performance of both methods decreases as the robot becomes more uncertain about the cup pose. However, using ARPLACE is more robust, because as uncertainty increases, the performance of FIXED suffers much more. By explicitly taking uncertainties about the robot's estimation of the world's state into account, ARPLACE is more robust when used for this task.

	$C(p_o)$		
	$\begin{pmatrix}0.00 & 0.00\\ 0.00 & 0.00\end{pmatrix}$	$\begin{pmatrix}0.01 & 0.00\\ 0.00 & 0.01\end{pmatrix}$	$\begin{pmatrix}0.04 & 0.00\\ 0.00 & 0.04\end{pmatrix}$
Success rate Pla4Man	89%	73%	65%
Success rate Default	84%	58%	27%
Significance	$p > 0.1$	$p < 0.001$	$p < 0.001$

TABLE II

RESULT OF THE EMPIRICAL EVALUATION. DIFFERENCE BETWEEN DEFAULT AND PLA4MAN STRATEGIES, DEPENDENT ON $C(p_o)$.

VIII. CONCLUSION

We have presented an approach to acquire a model of action-related place through experience-based learning. We have empirically demonstrated that using analytic models and human data to bias the exploration significantly reduces the number of training examples required for learning, whilst yielding a model of the same accuracy. The analytic model is a capability map, which is a compiled representation of the robot's workspace. The human data is acquired with camera-based tracking and RFID tags.

As the model is learned from experience, this approach can easily be applied to different types of robot, for instance legged robots, rather than the mobile platform used here. Different robots will gather different training data, which will lead to models tailored to the specific skills of the robot.

In our current investigations, we are taking into account different types of objects and grasps, thus leading to a more general concept of ARPLACE. Instead of mapping specific objects to places, we will map object and grasp *properties* to deformation modes. Also, instead of using the capability map to filter out certain random samples, we are considering using it to *actively* assist in the sampling procedure. Leven and Hutchinson [12] have demonstrated that basing the sampling on a measure of manipulability derived from the Jacobian enables the robot to learn accurate Probabilistic Roadmaps with less samples. With such an active search strategy, we expect to require even less samples to learn an accurate model. This will enable us to learn more general models of place, that take different objects and grasps into account.

REFERENCES

[1] J. Bandouch and M. Beetz. Tracking humans interacting with the environment using efficient hierarchical sampling and layered observation models. IEEE Int. Workshop on Human-Computer Interaction. In conjunction with ICCV2009.

[2] M. Beetz et al. Generality and legibility in mobile manipulation. *Autonomous Robots Journal (Special Issue on Mobile Manipulation)*, 2010. To appear.

[3] D. Berenson, H. Choset, and J. Kuffner. An optimization approach to planning for mobile manipulation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2008

[4] D. Berenson, S. Srinivasa, D. Ferguson, Alvaro C. Romea, and J. Kuffner. Manipulation planning with Workspace Goal Regions. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2009.

[5] S. Buck and M. Riedmiller. Learning situation dependent success rates of actions in a RoboCup scenario. In *Pacific Rim International Conference on Artificial Intelligence*, page 809, 2000.

[6] R Detry, E. Baseski, M. Popovic, Y. Touati, N. Krueger, O. Kroemer, J. Peters, and J Piater. Learning object-specific grasp affordance densities. In *Proceedings of the International Conference on Development and Learning (ICDL)*, 2009.

[7] R. Diankov, N. Ratliff, D. Ferguson, S. Srinivasa, and J. Kuffner. Bispace planning: Concurrent multi-space exploration. In *Proc. Int. Conf. on Robotics: Science and Systems*, 2008.

[8] S. Hart, S. Ou, J. Sweeney, and R. Grupen. A framework for learning declarative structure. In *RSS-06 Workshop: Manipulation for Human Environments*, 2006.

[9] 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.

[10] U. Klank, M. Z. Zia, and M. Beetz. 3D Model Selection from an Internet Database for Robotic Vision. In *International Conference on Robotics and Automation (ICRA)*, 2009.

[11] B. Kuipers, P. Beeson, J. Modayil, and J. Provost. Bootstrap learning of foundational representations. *Connection Science*, 18:145–158, 2006.

[12] P. Leven and S. A. Hutchinson. Using manipulability to bias sampling during the construction of probabilistic roadmaps. *IEEE Transactions on Robotics & Automation*, 19(6):10201026, December 2003.

[13] K. Okada, M. Kojima, Y. Sagawa, T. Ichino, K. Sato, and M. Inaba. Vision based behavior verification system of humanoid robot for daily environment tasks. In *Proceedings of the 6th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 7–12, 2006.

[14] P. Roduit, A. Martinoli, and J. Jacot. A quantitative method for comparing trajectories of mobile robots using point distribution models. In *Proceedings of IROS*, 2007.

[15] F. Stulp and M. Beetz. Refining the execution of abstract actions with learned action models. *Journal of Artificial Intelligence Research (JAIR)*, 32, June 2008.

[16] F. Stulp, A. Fedrizzi, and M. Beetz. Action-related place-based mobile manipulation. In *Proceedings of IROS*, 2009.

[17] M. Tenorth, J. Bandouch, and M. Beetz. The TUM kitchen data set of everyday manipulation activities for motion tracking and action recognition. Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (ICCV), 2009.

[18] M. Tenorth and M. Beetz. KnowRob — Knowledge Processing for Autonomous Personal Robots. In *Proc. of IROS*, 2009.

[19] M. Wimmer, F. Stulp, S. Pietzsch, and B. Radig. Learning local objective functions for robust face model fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(8):1357–1370, 2008.

[20] F. Zacharias, Ch. Borst, and G. Hirzinger. Capturing robot workspace structure: representing robot capabilities. In *Proc. of the IEEE/RSJ International Conf. on Intelligent Robots and Systems (IROS)*, 2007.