2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Context Aware Shared Autonomy for Robotic Manipulation Tasks

Thomas Witzig, J. Marius Zöllner
FZI Forschungszentrum Informatik
Karlsruhe, Germany

Dejan Pangercic, Sarah Osentoski
Robert Bosch LLC
Palo Alto, USA

Rainer Jäkel, Rüdiger Dillmann
Karlsruhe Institute of Technology
Karlsruhe, Germany

***Abstract*— This paper describes a collaborative human-robot system that provides context information to enable more effective robotic manipulation. We take advantage of the semantic knowledge of a human co-worker who provides additional context information and interacts with the robot through a user interface. A Bayesian Network encodes the dependencies between this information provided by the user. The output of this model generates a ranked list of grasp poses best suitable for a given task which is then passed to the motion planner. Our system was implemented in ROS and tested on a PR2 robot. We compared the system to state-of-the-art implementations using quantitative (e.g. success rate, execution times) as well as qualitative (e.g. user convenience, cognitive load) metrics. We conducted a user study in which eight subjects were asked to perform a generic manipulation task, for instance to pour a bottle or move a cereal box, with a set of state-of-the-art shared autonomy interfaces. Our results indicate that an interface which is aware of the context provides benefits not currently provided by other state-of-the-art implementations.**

Keywords Robotic Manipulation, Context Awareness, Shared Autonomy, Bayesian Network

I. INTRODUCTION

In order to create solutions for tasks in highly unstructured environments, robotics has typically focused on either fully autonomous or fully teleoperated systems. State-of-the-art in fully autonomous approaches often provide good solutions that do not require human input but are limited in their application due to underlying assumptions about the environment. Fully teleoperated systems can robustly operate in such environments but do not save the end user time. While this is appealing for tasks in environments dangerous for humans, full teleoperation does not bring robotics closer to complete solutions appropriate for household tasks. The field of *shared autonomy* [1, 2] bridges this gap by studying approaches to bring humans into the loop while still leveraging the power of autonomous algorithms. We define shared autonomy as the full or partial replacement of a function that is required by the robot by a human. The goal is to create more reliable robust systems with limited human interventions.

In this paper, we focus on a context aware shared autonomy system for robot manipulation tasks. Figure 1 provides an overview of our general approach. As seen in Figure 1, there are a large number of potential grasps, visualized as red arrows in the picture, that can be selected for a given object. However, the best way to grasp an object depends upon the goal of the task. For example, when moving a cup to the sink for cleaning picking the cup up from the top or the side are equivalent; however, when picking a cup to pour the liquid to another container a top grasp would be a poor choice.

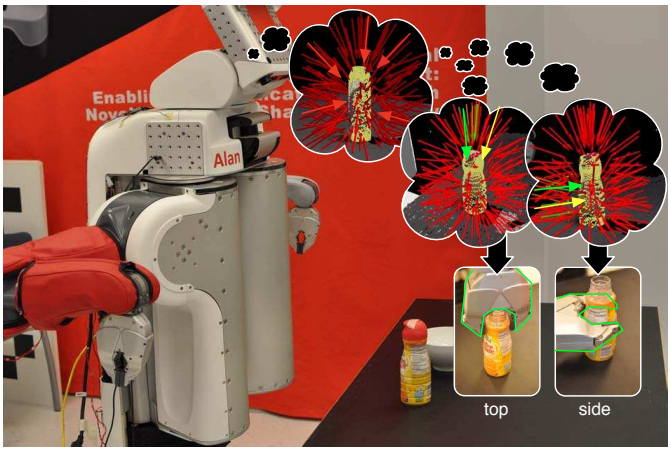


Fig. 1. Our system in a nutshell - PR2 robot computes grasping poses for an object bottle and is able to infer the right set of grasping poses for the task at hand, e.g. moving or pouring in this case, using probabilistic graphical models.

We present an approach that enables users to specify contextual information related to the task to help constrain the set of eligible grasp poses. In our approach, the robot autonomously generates a set of viable grasps and then uses the user provided constraints to aid in autonomously selecting a grasp. This contextual information is encoded into a probabilistic model based on multiple examples provided by the human teacher. We evaluate this approach by comparing it to state-of-the-art shared autonomy and autonomous systems through a user study.

The main contribution of this work is a novel dialog-based system, the Bosch Assistive Manipulation (BAM) library, which incorporates user provided semantic contextual information into a manipulation pipeline. The implementation of our system is based partly on and extends the work of Leeper et al. [3] and is available for download[1]. Our system uses a probabilistic model not to directly reproduce suitable grasp poses but instead to sort a set of given grasp hypotheses. The advantage of this approach is that it can be combined directly with advanced filtering techniques, e.g.

[1] http://www.ros.org/wiki/bosch_shared_autonomy_experimental

reachability, collision-checking and complex grasp quality metrics, of state-of-the-art grasp and motion planners. Such filtering techniques are essential for real-world applications and in general too complex to be integrated into the probability distribution. This approach also provides benefits with respect to data efficiency. Learning a probability distribution that generalizes well to different tasks and objects generally requires extensive amounts of (labelled) training data. The problem is solved by using a shared autonomy approach to refine the probability distribution on demand. During the execution, fully automatic grasp planning is used to generate a suitable set of grasp hypotheses which is sorted according to the learned probability distribution and filtered according to reachability and collisions. If the autonomous system fails, the human teacher selects suitable grasp hypotheses using a novel, graphical and dialog-based interface which are used to refine the learned probability distribution and encode the current context.

II. RELATED WORK

Grasp planning is a fundamental problem in the field of robotics which has been tackled by many researchers [4, 5]. The work proposed in these articles deals with generating successful and stable grasps for execution. However if we consider planning a grasp for a specific manipulation task, the stability of the grasp is no longer sufficient to describe all of the constraints on the grasp. Deriving quantified constraints from the task goals presents a formidable challenge since objects may have physical attributes that may constrain planning of a grasp and robots often have limited sensorimotor capabilities due to their various embodiments. Traditional approaches define these constraints using a task-oriented grasp quality metric which is based on wrench spaces [6, 7, 8, 9].

We extend the work of Song et al. [10] by implementing it on a real robot system and using a human teacher to provide the semantic constraints. A concept of providing expertise about task semantics through human tutoring has been implemented in [11] but never modeled and used in practice. Dang et al. [12] developed a definition of the semantic grasp which relates semantic label, sensory information and hand kinematics. To associate these grasps to the object knowledge they introduced a semantic affordance map. An even richer and more formal representation has been introduced in [13] and termed object action complexes (OAC). OACs allow for the representation of actions, objects, and the learning process that constructs such representations at all levels, from the high-level planning and reasoning processes that make use of them to the low-level sensors and effectors.

The problem with above approaches, however, is that they do not provide means for the human teacher to actively interact with the system in both training and execution stages. We alleviate this by leveraging approaches from the shared autonomy [1, 2] field where the original and still most common approach to shared autonomy in robotics applications is to assign human operators to supervisory or high level functions and machine intelligence to lower level functions. Sheridans widely cited description of 'levels of autonomy' [2] implicitly encodes this approach. This division has been successfully applied in many applications, especially in exploration and navigation tasks. Goodfellow et al. [14] for instance integrated user feedback into a robot to help with action selection. Leeper et al. [3] implemented four different strategies to grasp objects with shared autonomy systems. While they concentrated on interfaces to improve efficiency and collision avoidance, our work puts emphasize on capturing context information for the task at hand.

III. CONTEXT AWARE GRASP PLANNING

We define the context based on a set of features, which are either generated automatically or provided by the user through a novel dialog-based user interface. The context information is encoded into a probability density function using a Bayesian Network (BN) [15] with discrete and continuous variables. In a scene with different objects and obstacles, state-of-the art grasp planners are used to generate a set of grasp hypotheses. The user provides only high-level context information, which is combined with automatically generated context information derived from online perception. Using the combined context information as input, the probability density function is calculated and used to sort the set of grasp hypotheses. Following a maximum a-posteriori approach, we apply the (costly) planning filters, i.e. reachability and collision-checking [16], successively to each hypothesis and select the first hypotheses passing all filters. Since the user has to provide context information, training data is usually limited and automatic generalization might fail. In this case, we exploit the high usability of the developed dialog-based user interface to generate a shared autonomy process, in which the users provides additional training examples online to automatically adapt the probability density function and enable the robot to resume autonomous operation.

A. Context feature set

The context information is defined based on six features divided into three subsets motivated by [10]: task features (T), object features (O) and grasp features (G). We describe each feature and list the set of values used in the experiments.

A single task feature $T \in \{t_1, ..., t_n\}$ is used to model the task type as a discrete value:

1) Task type $T \in \{move, pourin\}$

This decision is grounded in our domain choice which is tabletop object manipulation. These two are most commonly used and can be extended easily.

Object features, O, encode information required to generalize across different object types. We model three discrete:

2) Object shape $O_O \in \{convex, concave\}$
3) Fill level $O_F \in \{filled, empty\}$
4) Side graspability $O_G \in \{true, false\}$

and one continuous object feature:

5) Object size $O_S \in \mathbb{R}^3_{\geq 0}$

The object size is defined by the width, height and depth of the automatically calculated oriented bounding box of

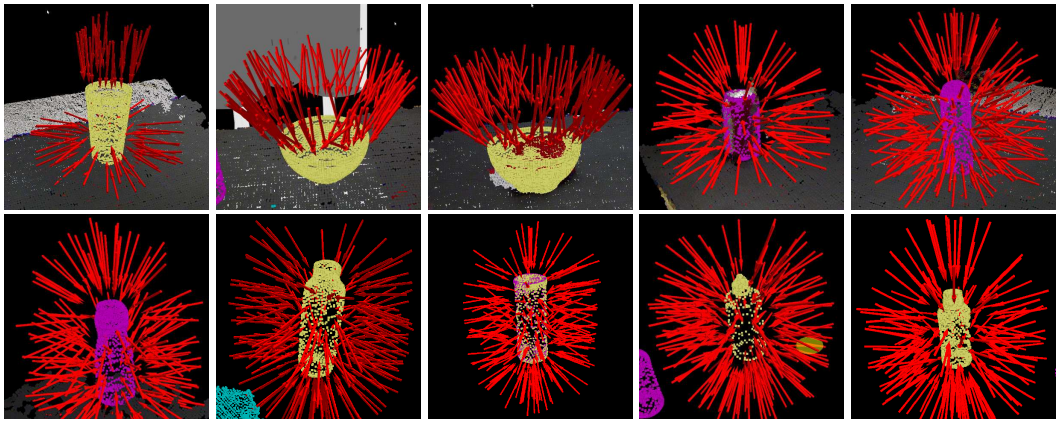


Fig. 2. Precomputed grasp poses from the database. The seven dimensional grasp poses (3D for position and 4D for orientation) are used as raw data for this approach and visualized as arrows. Yellow color implies a selected object.

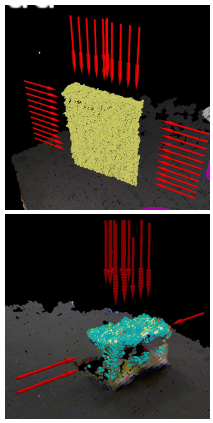


Fig. 3. Generated grasp poses from point cloud.

the object. Side graspability is false if the width and depth of the oriented bounding box exceed the maximum gripper opening. While side graspability is a constraint coming from our gripper setup, we used object shape and size as a fast approximation of an object recognition system. In the future, we plan to integrate additional local features such as the ones proposed by Montesano et al. [17] as well as softness and surface friction features presented by Chu et al. [18]. Fill level on the one hand constrains the robot from reaching into the object and on the other hand imposes an up-right constraint needed to e.g. avoid spills.

We use the pose of the Tool Center Point (TCP) relative to the object coordinate system as a single grasp feature G:

6) Grasp pose $G \in \mathbb{R}^7$

The grasp pose is represented as a 3D position vector and 4D orientation quaternion.

B. Grasp hypotheses

We use a discrete set of grasp poses as input, which is calculated using two existing approaches. Once the point cloud data from the depth sensor is obtained, the object recognition system segments flat surfaces and possible objects on it. The object candidates (as point clouds) are then compared to a database using the Iterative Closest Point algorithm (ICP) [19]. If the recognition succeeded, precomputed grasps from the database, see Figure 2, are used to generate grasp poses for this object [16]. If the recognition fails, possible grasps are generated automatically using local features of the point cloud itself [20], see Figure 3. We visualize each grasp as an arrow centered on the TCP pointing along the z-axis.

C. Bayesian Network Modeling

We define the probability density function using a BN. A BN is a common probabilistic graphical model to represent conditional dependencies of a set of random variables in a directed acyclic graph. The model consists of variables $X = \{X_1, X_2, \ldots, X_N\}$ which are represented by either unobservable (latent variables) or observable nodes. The structure of the graphical model encodes the relations between the variables X. Each node is represented by a conditional probability distribution (CPD) which is dependent on the node's parent variables. For instance, if one node has m boolean parents then the probability distribution is represented as a multinomial probability distribution by a table of 2^m entries, which is called conditional probability table (CPT).

We represent continuous features as Mixture of Gaussians (MOG). A MOG is a weighted sum of i unimodal Gaussian distributions, each described with mean μ, covariance Σ and weight π, and denoted as:

$$p(x) \quad = \quad \sum_{i=1}^{m} \pi_i \, \mathcal{N}(x - \mu_i, \Sigma_i) \tag{1}$$

The number of Gaussian distributions is encoded as a discrete latent variable which is directly connected to the node of the continuous feature. If we denote the network structure as S, a set of CPDs of each variable X_i, the parents p_i of each node X_i and its model parameters $\theta_S = (\theta_1, \ldots, \theta_n)$ then we compute the probability density function based as:

$$P(X|\theta_S, S) = \prod_{i=1}^{n} P(X_i \,|\, p_i, \theta_i, S) \tag{2}$$

We chose a BN to model the probability density function due to several important characteristics. It supports mixtures of discrete and continuous CPDs in a probabilistic framework, which is highly relevant to encode different types of context information. Inference mechanisms are used to deduce the posterior probability of G even with only partial context information. The latter is important to consider partial user input and partial observability due to online perception.

a) Bayesian Network Structure: The defined BN contains the task, object and grasp features as variables, see Figure 4. The task type and the object features directly affect the grasp features which are represented by directed edges to the respective node. Every possible task and object feature leads to a distinct probability distribution for the

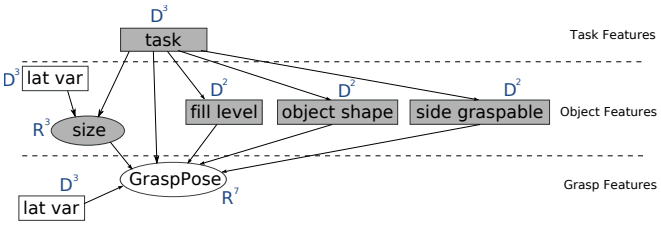


Fig. 4. BN structure with continuous nodes (ellipses, R), discrete nodes (rectangles, D), latent variables (white). The exponent D^n, R^m shows the number of elements n or dimension m.

grasp pose. Furthermore, the task type also constrains the object selection which means that object size, fill level, object shape and side graspability depend on the task type. For each continuous feature a discrete parent latent variable for the mixture coefficients of the MOG is introduced.

Based on this structure the joint probability distribution can be decomposed as:

$$P(T, O_S, O_F, O_O, O_G, G) = P(G \,|\, T, O_O, O_F, O_S, O_G) \\ P(O_S \,|\, T) P(O_F \,|\, T) P(O_O \,|\, T) P(O_G \,|\, T) P(T) \quad (3)$$

The conditional probability of the grasp pose given context information is simplified to:

$$P(G \,|\, T, O_O, O_F, O_S, O_G) = P(G \,|\, T, O) \quad (4)$$

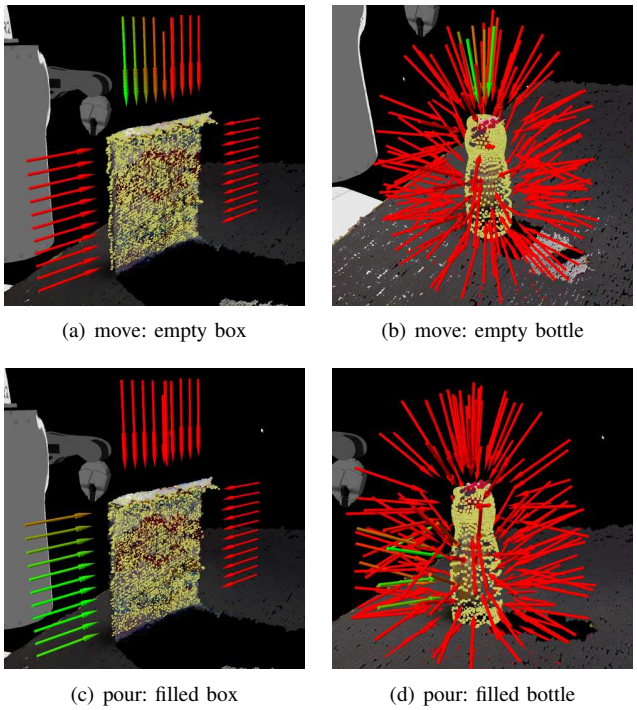
b) Bayesian Network Inference: The BN stores all statistical dependencies allowing the efficient calculation of the posterior distribution of variables given the observed data (evidence), which is called probabilistic inference. Once the network is learned, we infer the posterior distribution of grasp poses out of the evidence. The conditional probability of the grasp pose is denoted as $P(G \,|\, T, O)$. The evidence consists of task type T and object properties O (size, fill level, object shape and side graspable). We apply the junction-tree algorithm[21] to compute the exact inference.

c) Bayesian Network Learning: We use the expectation-maximization (EM) algorithm [22] to learn the parameters for the CPD and CPT of the BN on the basis of labelled training data. The algorithm converges to a local maxima for the complete likelihood of the BN. We run the algorithm with three random initial values and use the solution with maximum posterior probability.

D. Weighting of grasp hypotheses

In order to weight the set of grasp hypotheses for a given object, we compute $P(G \,|\, T, O)$ for each grasp pose using the BN Inference. Figure 5 shows an example output for the conditional probabilities. We visualize each grasp pose as an arrow. The color of the arrow is computed in RGB space. Given all likelihoods $L = \{P(G \,|\, T, O)_1 ... P(G \,|\, T, O)_n\}$ for a set of n grasp poses, the likelihood $P(G \,|\, T, O)_i$ is encoded as green g_i and red r_i color values where green represents a higher likelihood:

$$g_i = \frac{P(G \,|\, T, O)_i - min(L)}{max(L) - min(L)} \qquad r_i = 1.0 - g_i \quad (5)$$



(a) move: empty box (b) move: empty bottle

(c) pour: filled box (d) pour: filled bottle

Fig. 5. Visualization of conditional probability $P(G \,|\, T, O)$ for two different tasks (move, pour) using a single, trained BN.

IV. Shared Autonomy User Interface

Previous studies [3] discuss important aspects to successfully tele-operate a robot using commodity wireless network and commodity hardware interfaces such as desktop computer with a mouse and a keyboard. This is important in the light of an envisioned untethered deployment of robots in real households. They show that a dialog-based 3D point-and-click interface where the robot has a partial autonomy in choosing its actions is both efficient and less demanding in terms of cognitive load. Based on these findings, we decided to extend their point-and-click interface to fit our need to provide high-level context information and training data. Figure 6 shows the key steps to gather training data and encode the context information into the BN. First, the perception result is visualized to the user. Based on the perception result, the oriented bounding box of the object is calculated and the features object size O_S and side graspability O_G are derived automatically. Additional context information, i.e. T, O_O, O_F, is provided by the user with a dialog-based user interface (Figure 6(a)). With this step, we included a mouse-based menu and a iconic manipulation feature as is common in CAD software [3]. The BN automatically scales to situations, in which only a subset of context information is supplied. Grasp hypotheses are generated automatically and visualized to the user. The user selects a single or multiple grasp hypotheses to be added as training data to the BN (Figure 6(b)). In this step, the shared autonomy concept is realized. The user can adjust the training set online using the provided graphical tools such as the dialog interface and

(a) The user provides additional context information. Here, the user wants to pour a filled bottle.

(b) The human operator selects multiple grasp poses as positive training samples.

(c) Resulting grasp suggestion. The most likely grasp poses are encoded in green color.

Fig. 6. Visualization of developed dialog-based user interface to generate the initial probability density function.

the visualization of the grasp hypotheses. Finally, the BN is learned and the weighting of grasp hypotheses is visualized to the user (Figure 6(c)), which is the basis for the robot to select grasps automatically using the context information provided by the user.

V. EXPERIMENTAL SETUP

We implemented BAM as a set of ROS packages integrated in the ROS manipulation pipeline[2] and release them as open source. The implementation has been tested in the robot laboratory of the Bosch Research and Technology Center as well as in the laboratory of TU Munich on two different PR2 robots. For this paper, we present the results of the latter experiment. In order to analyse the importance of the context awareness and the presence of a human co-worker, we carried a rigorous user evaluation to compare our novel dialog-based user interface with the three other state-of-the-art approaches. We asked eight subjects to fulfill the following five simple manipulation tasks using the tele-operation interfaces:

- Task 1: Pour from box into bowl;
- Task 2: Move box;
- Task 3: Pour from bottle into bowl;
- Task 4: Move bottle;
- Task 5: Move filled cup.

The objects used in the experiments are shown in Figure 8. Each task consisted of two steps: grasping the object and a subsequent manipulation action. In addition to the BAM interface, subjects used the following three interfaces with the different level of autonomy as proposed by Leeper et al. [3]:

- Autonomous Grasping (AG);
- Ghosted Gripper (GG);
- Interactive Marker (IM).

The AG interface automatically segments objects on a flat surface and visualizes them to the user. The user selects an object which will automatically be grasped using either a set of pre-computed grasps or a heuristic grasp quality measurement. For the GG interface, the human operator moves a simulated 'ghosted' gripper to define the desired grasp pose and sends the pose to the motion execution algorithm, if the arm motion planner finds a feasible solution. The direct control strategy IM enables the operator to remote control the robot in Cartesian space in real-time by clicking and dragging a set of rings and arrows. It is worth mentioning that none of these three interfaces supports saving of data and learning.

Once the object has been grasped, the user fulfilled the manipulation action using the IM interface. This separation helped us to compare the effects of different grasp poses which were generated using the four interfaces with respect to the manipulation task. We decided to concentrate on a single-arm manipulation task since the prime goal was to create a baseline system which will allow us to evaluate the suitability and scalability of probabilistic graphical models as means to encode contextual knowledge in robotic manipulation.

We recruited eight subjects, one female and seven male ones. Four of the subjects had never worked on a robot and three never used rviz[3], a 3D visualization tool, before. Only one of the subjects never played 3D games but six subjects had experiences in robotics in general.

The subjects had an obstructed view to the test scene. During the evaluation, they familiarized themselves with the interfaces. To avoid bias towards one interface, we randomized the order in which we presented the interfaces to the subjects.

VI. EXPERIMENTAL RESULTS

We performed a quantitative as well as qualitative analysis of the evaluation results to assess if robot's context knowledge affected user's experience in working with the robot.

[2] http://www.ros.org/wiki/pr2_interactive_manipulation

[3] http://ros.org/wiki/rviz

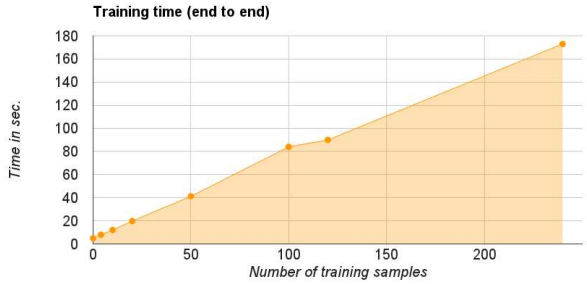


Fig. 7. BN average training times (end-to-end) for 40 trials.

Fig. 8. Testing set of objects for the interface evaluation. A bottle, cereal box and a filled cup (simulated with transparent foil) have been used to fulfill five simple manipulation tasks.

Fig. 9. Failed grasp using the AG interface. The grasp failed because the object was grasped from above for Task 3 - pour from a bottle into bowl.

A. Quantitative Results

We first time profiled the BAM library and measured how the training and query times of the BN vary with respect to the number of training samples. Secondly, we measured success rates presenting the number of successfully executed attempts summarized over all subjects and all interfaces. In addition to the success rates, we also measured the number of minor and major collisions as well as the times when user interacted with the robot and the times when the robot operated autonomously.

1) Training and Query Times for BAM interface: When training the Bayesian Network the training time approximately increases with the number of training samples in a linear fashion as depicted in Figure 7. Given that the training needs to be performed only once per objects we consider this as an acceptable solution. Beside the training times, we also measured the end to end query time (from user click to the grasp suggestion) given instances of the Bayesian Network with the different number of samples. The query times were constant at 5.8 seconds for a various amount of samples. Our client-server design pattern and the network communication delay caused approximately 2 seconds of overhead and the rest was spent on the inference itself. Given that motion planning and manipulation actions normally take in the order of 1 minute, we consider these times acceptable for real household tasks.

2) Success Rates: With the BAM interface the subjects successfully grasped and manipulated 37 out of 40 objects.

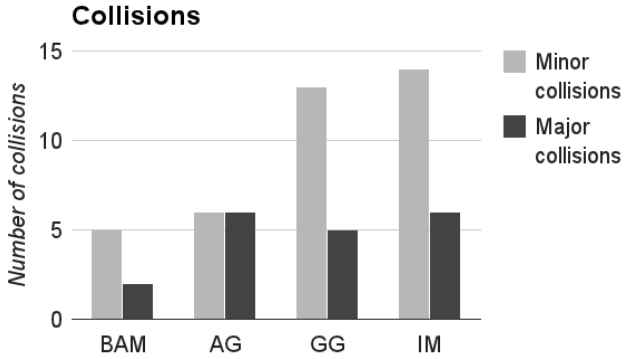


Fig. 10. Number of minor and major collisions for the four user interfaces which occurred during the grasp and manipulation execution.

Three failed attempts were caused by the human subject which forgot to select the correct context information. The other three interfaces all respectively resulted in 31 out of 40 succeeded attempts where failures were caused by different reasons. The failed task executions for AG were due to missing flexibility and the lack of situational context. For instance, if an object has been grasped from above the robot failed to pour from the bottle into a bowl as shown in Figure 9. In another instance the front part of the gripper reached into a filled cup. The failed attempts using GG were caused by three reasons: either the gripper was placed next to the object, shifted the object on the table or the grasped object slipped out of the gripper during the manipulation action caused by an unstable grasp pose. Using the IM for the whole grasp and manipulation execution, the human operator was in full control of the robot at any time. Since the user had an obstructed view to the processing scene, more collisions occurred for this interface.

3) Collisions: During the evaluation, we measured the number of minor and major collisions which occurred during the grasp execution and the manipulation task influenced by the chosen grasp pose. We considered collisions where an object toppled as major collisions and everything else as minor collisions. Figure 10 shows the number of minor and major collisions using the four interfaces. The best results were achieved using the BAM interface with only five minor and two major collisions in total, followed by the AG with six minor and six major collisions. A considerably increased number of collisions have been recorded using the GG (18 collisions) and the IM interface (20 collisions).

4) Execution times: For this evaluation, we measured the complete execution times for five manipulation tasks. To evaluate the effect, we separated the times in following categories:

- Autonomous time before grasp;
- Interactive time before object grasp;
- Interactive time after object grasp.

Moreover, only succeeded task executions are considered in time charts presented in Figure 11. AG and BAM approaches amounted to an increased computation time but considerably

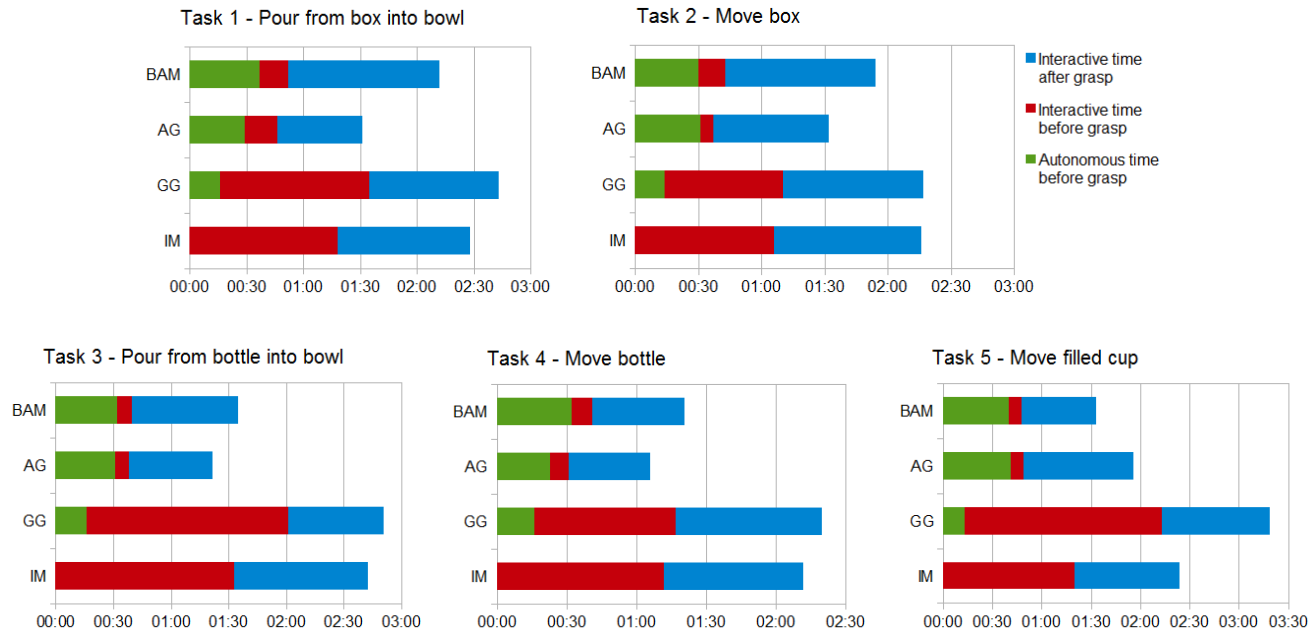


Fig. 11. Visualization of user interaction and computation times for five manipulation tasks using the four user interfaces. All times are in seconds.

decreased interaction time to grasp the object compared to IM and GG interfaces. Though the results show that our interface is rather intuitive and quick to adapt to the edge that AG got over BAM interface in the total computation time, is the compromise for having the human in the loop. This shows an importance in designing easy, intuitive and back-end algorithm agnostic interfaces.

5) Cognitive load: Additionally, we measured the cognitive load of the subjects during the grasp executions. We used the Sternberg memory test for task 3 as described by Tracy et al. [23]. The results showed a decreased cognitive load for the BAM and AG interface. However, we consider a detailed measurement of the cognitive load as future work.

The quantitative results presented in this paper show the significance of our approach compared to the experiments conducted by our competitors, but less in terms of all possible objects that can occur in everyday households all around the world.

B. Qualitative Results

After testing each interface, the subjects were asked to fill out a questionnaire to rate their experience. It was noted that the AG and BAM interface provided less 'fun' to the user who rather enjoyed having more control of the robot. On the other hand, it was easier to grasp and select objects using the autonomous tools which were but still easy to understand. In general, the subjects felt more comfortable if the system was equipped with additional knowledge using the AG and BAM interface. On the other hand, the subjects preferred to be in control of the robot using the GG and IM interfaces although this caused more failed attempts and collisions.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we present a user interface where the operator provides additional information to the system, trains, executes and improves grasp poses for household objects in collaboration with a real PR2 robot. A BN encodes context information and a set or pre-computed grasp poses to compute a grasp quality measurement. The human operator provides the context information through a dialog-based user interface. Beside the context information, the user trains and adapts the underlying system by incrementally inserting training samples to the BN. Once the model is trained, we infer the likelihood of the seven dimensional grasp pose given an evidence, the provided task and object properties. This likelihood is used to measure a grasp quality for a discrete set of precomputed grasps.

Additionally, a richer and more descriptive set of 2D and 3D features combined with a structure learning algorithm as described by Song et al. [24] will lead to more generalized results for context-aware object grasping. Furthermore, object features which are currently provided by the human operator can be replaced with automatically computed input data. The resulting grasp constrains from the BN can be used as an initial grasp pose for unrecognized objects. Local features can be used to learn how to grasp those objects after several iterations (active learning) as proposed by Montesano and Lopes [17]. In the future work we also plan to investigate cluttered scenes and bi-manual manipulation tasks.

We present our novel system in a short demo video[4].

[4] http://www.youtube.com/watch?v=BySOqr4FBUY

VIII. ACKNOWLEDGEMENTS

Many thanks to Prof. Gordon Cheng for providing laboratory space and a PR2 robot and to Dan Song for providing many insightful thoughts on the design of the Bayesian Networks.

REFERENCES

[1] P. Michelman and P. Allen, "Shared autonomy in a robot hand teleoperation system," in *Intelligent Robots and Systems '94*, Sep 1994.

[2] T. B. Sheridan, *Telerobotics, automation, and human supervisory control*. Cambridge, MA,: Technology and Society Magazine, IEEE, 1992.

[3] A. Leeper, K. Hsiao, M. Ciocarlie, L. Takayama, and D. Gossow, "Strategies for human-in-the-loop robotic grasping," in *Proc. of Human-Robot Interaction (HRI)*, 2012.

[4] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *IJRR*, 2008.

[5] M. Ciocarlie and P. Allen, "Hand posture subspaces for dexterous robotic grasping," *The International Journal of Robotics Research*, 07/2009 2009.

[6] M. Prats, P. J. Sanz, and A. P. del Pobil, "Task-oriented grasping using hand preshapes and task frames," in *IEEE International Conference on Robotics and Automation*, May 2007.

[7] R. Haschke, J. J. Steil, I. Steuwer, and H. Ritter, "Task-oriented quality measures for dextrous grasping," in *in Proc ICRA*, 2005.

[8] L. Zexiang and S. Sastry, "Task oriented optimal grasping by multifingered robot hands," in *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, Mar. 1987.

[9] C. Borst, M. Fischer, and G. Hirzinger, "Grasp planning: how to choose a suitable task wrench space," in *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, april-1 may 2004.

[10] D. Song, K. Huebner, V. Kyrki, and D. Kragic, "Learning task constraints for robot grasping using graphical models," in *International Conference on Intelligent RObots and Systems - IROS*, 2010.

[11] Z. Xue, A. Kasper, M. J. Zoellner, and R. Dillmann, "An automatic grasp planning system for service robots," in *14th International Conference on Advanced Robotics*, 2009.

[12] H. Dang and P. Allen, "Semantic grasping: planning robotic grasps functionally suitable for an object manipulation task," in *Intelligent Robots and Systems (IROS)*, Oct. 2012.

[13] N. Krüger, J. Piater, F. Wörgötter, C. Geib, R. Petrick, M. Steedman, A. Ude, T. Asfour, D. Kraft, D. Omrcen, B. Hommel, A. Agostino, D. Kragic, J. Eklundh, V. Krüger, and R. Dillmann, "A Formal Definition of Object Action Complexes and Examples at Different Levels of the Process Hierarchy," EU project PACO-PLUS, Tech. Rep., 2009.

[14] I. Goodfellow, N. Koenig, M. Muja, C. Pantofaru, A. Sorokin, and L. Takayama, "Help me help you: Interfaces for personal robots," in *Proc. of Human Robot Interaction (HRI)*, ACM Press. Osaka, Japan: ACM Press, 2010.

[15] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[16] S. Chitta, E. G. Jones, M. Ciocarlie, and K. Hsiao, "Perception, planning, and execution for mobile manipulation in unstructured environments," *IEEE Robotics and Automation Magazine, Special Issue on Mobile Manipulation*, 2012.

[17] L. Montesano and M. Lopes, "Active learning of visual descriptors for grasping using non-parametric smoothed beta distributions," *Robotics and Autonomous Systems*, vol. -, pp. 26–AUG–2011, 2011. [Online]. Available: http://hal.archives-ouvertes.fr/hal-00637575

[18] V. Chu, I. McMahon, L. Riano, C. McDonald, Q. He, J. M. Perez-Tejada, M. Arrigo, N. Fitter, J. C. Nappo, T. Darrell, and K. J. Kuchenbecker, "Using robotic exploratory procedures to learn the meaning of haptic adjectives," in *IEEE International Conference on Robotics and Automation (ICRA) Karlsruhe*, May 2013.

[19] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, Feb. 1992.

[20] K. Hsiao, S. Chitta, M. Ciocarlie, and G. E. Jones, "Contact-reactive grasping of objects with partial shape information," in *Intelligent Robots and Systems (IROS)*, 10 2010.

[21] C. Huang and A. Darwiche, "Inference in belief networks: A procedural guide," *International Journal of Approximate Reasoning*, 1996.

[22] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society, Series B*, 1977.

[23] J. Tracy, *Measuring Cognitive Load to Test the Usability of Websites*. University of Memphis, 2007.

[24] D. Song, C. H. Ek, K. Huebner, and D. Kragic, "Multivariate discretization for Bayesian Network structure learning in robot grasping," in *International Conference on Robotics and Automation*, 2011, pp. 1944–1950.