An Hyperreality Imagination based Reasoning and Evaluation System (HIRES)*

Sebastian Rockel, Denis Klimentjew, Liwei Zhang, Jianwei Zhang

Abstract— In this work we ask whether an integrated system based on the concept of human imagination and realized as a hyperreal setup can improve system robustness and autonomy. In particular we focus on how non-nominal failures in a planning-based system can be detected before actual failure. To investigate, we integrated a system combining an accurate physics-based simulation, robust object recognition and a symbolic planner to achieve realistic prediction of robot actions. A Gazebo simulation was used to reason about and evaluate situations before and during plan execution. The simulation enabled re-planning to take place in advance of actual plan failure. We present a restaurant scenario in which our system prevents plan failure and successfully lets the robot serve a drink on a table cluttered with objects. The results give us confidence in our approach to improving situations where unavoidable abstractions of robot action planning meet the real world.

I. Introduction

This work introduces a novel approach to the prediction of robot action results based on physical simulation during plan execution. This makes the scoring of plan operator parameters possible and allows re-planning. The simulation is combined with our recently introduced object recognition method to improve recognition results and add perceived objects to the simulation.

Since the early days of robotics the fundamental and interesting question was how robots could adopt human cognitive processes. In this work we propose a system implementing the cognitive principle of imagination. The approach combines, in a hyperreal setup, the cognitive concept of imagination for reasoning and evaluation into one system (HIRES). It does not rely on traditional spatial reasoning [1] but instead uses simulation as a tool to enable the testing of robot execution before actions are actually performed. This naturally saves robot resources and, when optimized, total execution time, as well as improving system robustness. Our approach is inspired by the concept of imagination from [2]. Imagination enables humans to consult rules or principles but not merely to apply those rules. Instead humans imagine what the consequences might be if the rules are, or are not, followed [3]. The author notes that humans constantly use imaginative projection and that imagination is essential to human reasoning.

According to Loasby [2] reasoning is expensive in time and energy and should therefore be used sparingly. Our approach is to combine the concept of imagination with the concept of "Hyperreality" [4]. The term Hyperreality is based on earlier work from Baudrillard [5]. It originates from semiotics and postmodern philosophy and is considered the condition where an agent cannot distinguish reality from a simulation of reality. Therefore reality and virtual reality can be considered interchangeable (referred to in our previous work as *Mixed Reality* [6], [7]), as can human and artificial intelligence [4].

The experimental domain is a restaurant-like environment in which the robot carries out tasks typical for a waiter, such as serving a beverage to a guest or cleaning a table. Aspects of these tasks, such as serving coffee to a guest, require reasoning in order to be performed correctly. For example to place the mug on the table near to the guest, the robot must use spatial reasoning to compute a suitable table-top area in which the mug is to be placed. The appropriate area might already be occupied by the guest's belongings, e.g. a mobile phone or a book, or by normal table decorations such as a vase or salt and pepper pots. Considering this scene in a restaurant causes an increase in "reasoning space" as computations have to consider each object in relation to all other objects.

Use of the proposed reasoning and evaluation approach is based upon reliable object recognition. Many different recognition methods and algorithms have been developed in recent decades. Nevertheless no available algorithms can deliver the necessary robustness and success rate. Therefore, in the work presented, the strategy used is to combine multiple algorithms (detectors) for different object properties. Thereby, object recognition is based on the combination of results from different detectors with varying accuracy.

An exact simulation of the scene is used in order to execute robot actions (e.g. picking-up the mug from the counter and placing it in front of the guest) before they are actually performed. The simulated consequences of the action (e.g. placing the mug failed) are used to adapt the current plan in order for the task to succeed.

The approach presented supports the assumption that applying imagination with the help of a hyperreal system improves robot autonomy and saves resources. In the next section we present the HIRES approach to resolving the proposed problem. HIRES is a central component for reasoning and evaluation of robot behavior and capabilities. Before the experiments and scenarios are described in Sec. III, the technical realization - the frameworks used, the adaptations and implementation details of HIRES and the object recognition method - is described.

II. APPROACH

HIRES is based upon the idea that imagination in human cognition improves human behavior. In computer science and especially robotics, imagination has received little attention. In this work we use an accurate and realistic simulation of the robot and its surroundings (as detailed as possible compared to the real world) to represent the robot's physical observations. This lets the system "imagine" the consequences of following or not following a plan before actually executing an action.

On the basis of an accurate and recent simulation, the borders between simulation (or virtual reality) and reality fade away. In philosophy and sociology the term *Hyperreality* has been established to describe exactly this condition [5].

Lodato [3] describes imagination as being central to human reasoning in that it allows us to organize and describe our experiences. And Loasby states that creation of new patterns rests on imagination, not logic. In this work, robot imagination enables reasoning about possible robot actions by inferring from simulation.

HIRES can be used to evaluate components or whole systems. In this work, it will be demonstrated by means of object recognition. In simulation many different table-top scenes with varying numbers and types of objects can be used to measure the algorithm's accuracy and performance.

A. Frameworks and Planning

In previous work we developed an ontology-based robot control architecture to allow robots to learn from experience [8]. To focus the research on high level tasks such as learning from experience, as employed in the RACE project[1], it is necessary to be able to take the robustness and functionality of lower level components for granted. Our platform provides basic capabilities such as navigation, manipulation and grasping.

Robots and robot control software become ever more complex. There is also a tendency to use ever more sensors and consequently the amount of data to be processed rises. As the complexity of tasks increases and as tasks are divided into many parts, coherence between capabilities grows rapidly. For example the PR2[2] robot produces data at approx. 50 MB/s, not including the RGB-D sensory output.

There are several coherent frameworks, like ROS, that integrate many different capabilities. One such capability is navigation, which typically consists of localization, path planning and robot control. The combination of 2D navigation and 3D collision avoidance is the state-of-the-art method in mobile robotics. 2D navigation provides the necessary performance; 3D collision avoidance the required fidelity.

Other essential capabilities are manipulation and grasping. For these it is important that a solution for inverse kinematics exists, in addition to solutions for the calculation of possible trajectories and grasps. One difficulty is the calibration between sensors, the robot and the gripper pose.

High grasp accuracy and the coordination of grasping with the recognition process must be ensured. Generally the task can be finished successfully only if all components are successful. If one component fails, the goal cannot be achieved. Other components, such as the planner, object recognition and the simulator are indispensable and belong to the basic capabilities.

Hierarchical Task Network (HTN) planning is one of the most popular high-level planning strategies for mobile robotics. In our work we currently use SHOP2 (Simple Hierarchical Ordered Planner) [9]. A huge advantage of SHOP2 over other HTN planners is its ability to generate partial-order plans. Unfortunately, HTN planners provide no possibilities for parallelization or re-planning. To compensate for this here, the planner is extended with a parallelization layer realized with an execution-based state machine, SMACH (**S**tate **Mach**ine) [10]. This allows the use of containers to assign different tasks to each state and is thereby able to execute any type of parallelized plan. This early work is described in [11].

[12] combines geometric and symbolic representations in an hierarchical planning architecture, called Hierarchical Planning in the Now, or HPN. Here a system is integrated to generate behavior in a real robot that robustly achieves tasks in uncertain domains. Simulation is used to demonstrate the system but not to exploit physical reasoning capabilities. [13], [14] define the Diverse Action Manipulation (DAMA) problem concerning a robot, a set of movable objects and a set of diverse manipulation actions. The objective is to find a sequence of actions that moves each of the objects to a goal configuration. In this work the manipulation problem is addressed as a multi-modal planning problem and it proposes a hierarchical algorithm to deal with it. Although the work addresses similar robot manipulations the authors do not use physical planning or simulation to improve robot autonomy or robustness.

B. Simulation

Simulation has become an essential tool in the development of complex mobile systems. The availability of physical robots is limited and researchers often work in separate physical locations on different system parts. Consequently whole-system evaluation and testing is often not possible. In such cases a realistic simulation is indispensable.

Harris et-al., [15] gives a good outline of the available simulators, frameworks and tool-kits. The authors suggest that most simulators are not well documented (for anything but Windows environments), have no sensors, or offer no possibilities for the integration of new robots, sensors, etc. Many tools are insufficiently precise, do not include physics engines or are no longer under active development. However, according to the authors, ROS has been proven very suitable. The Gazebo-based simulator provided by ROS integrates many robots and sensors as well as many libraries and tools. It also provides a physics engine and delivers accurate results. Furthermore ROS is designed to be a partially real-time system according to Harris. This, in addition to very

---

good PR2 integration, explains why ROS is currently one of the most popular robotics frameworks. [16] presents a qualitative evaluation of a number of free, publicly available, physics engines for simulation systems and game development. Among the ten evaluated physics engines, our work mainly uses ODE[3], which is used in the Gazebo simulator at version 1.0. [17] evaluates the Bullet engine as an ingredient of physical reasoning. It investigates the prediction accuracy of a physics engine when simulating a robot arm pushing flat objects across a table. The authors recommend reasoning about goals and actions using physics simulations. Successful reasoning is considered to imply the need for a strong understanding of dynamics and the effects of actions.

In [18] a system projecting the results of robot actions in everyday object manipulation (in this case, making a pancake) is proposed in order to obtain appropriate action parametrization for successful execution. This task, which for a robot is complicated, is broken down into simpler actions, e.g. breaking an egg, pouring the pancake mix etc. A model integrated with PROLOG allows chronological backtracking of actions and triggers physical simulations, monitoring and logging. It translates the logged data into first-order time-interval-based representations, called timelines. Naive physics and commonsense reasoning are used to improve the robot's manipulation capabilities. The framework includes the simulator Gazebo, the PROLOG logic programming language and a parametrizable robot control program. Evaluation focuses on qualitative results (e.g. a well broken egg) related to specific parameters. There is a discussion of the limitations of today's symbolic reasoning methods for drawing inferences about simple physical problems, as the necessary details of physical properties are abstracted away. The system presented uses a high-fidelity physics model that lacks real-time performance and hence must be applied offline (a-priori). This lack of real-time performance means that its predictions cannot be used for planning during execution. The work does not use higher level symbolic planning, e.g. HTN planners or state machines (e.g. SMACH).

In our work, ROS is used as a low-level framework and its integrated simulator proves to be suitable for research, testing and evaluation.

C. Object Recognition

Many existing methods for simultaneous object recognition try to match features using a database [19], [20] and may also integrate stereo cameras, time of flight cameras, thermal cameras, etc. [21]. However all of these methods have limitations. One method, for example, works only with rotationally symmetrical objects. Other algorithms cannot handle unknown objects and yet others work only with unknown objects.

The method presented in [22] is based upon a similar approach. The authors use two detectors, a combination of RANSAC and Mean Shift as well as the SIFT algorithm to recognize objects. Unfortunately the authors evaluate their

[3]http://www.ode.org

approach only for single objects in the scene. Furthermore the approach cannot handle partial occlusion.

In [23], point clouds are collected from different perspectives in order to improve clustering before recognition. In this method, point cloud collection probably takes a relatively long time and there is no guarantee that the best possible point cluster will be obtained. Nevertheless, clustering is an important part of the recognition method and improves their results.

The approach we presented in [24] extends the existing state-of-the-art and shows a possible way to improve scene analysis and to achieve more reliable and accurate object recognition. In the case of partial occlusion, this works by simultaneously using different detectors and by active interference with scene objects. The work goes a step towards active intelligent perception. In this case, many detectors are run in parallel, each concentrating on a different feature or object property. Depending upon the output of the detectors, an intelligent decision for the next step can be made. So it is possible to change the view of the sensors in various ways, for example by moving the robot's head, its torso or even the whole platform. The other possibility is to interfere with a scene, for example to grasp inside the scene or to change the arrangement of the objects on the table. To show more evaluation possibilities an object recognition approach based on [25] was used. Here, object recognition was improved by the developed detector. Foremost among the improvements was our implementation and integration of 3D recognition based upon ICP (Iterative Closest Point). This improved upon methods integrated into ROS by offering the ability to compare in three dimensions; in this case the method can be used for objects that are not rotationally symmetrical. Another advantage is that the objects need not have known orientation.

During experiments it became evident that if the shapes and curves of objects were similar and objects of different sizes were compared, the probability of false-positive comparison increased rapidly. To compensate for this disadvantage the ICP-based algorithm was improved by dividing the comparison processing into two steps. In the first step, the input cloud is compared with a database object; in the next step the same comparison is performed in reverse order. To adapt the resulting similarity, the quadratic error of distances between closest neighbor voxels is calculated for each step. The similarity is based upon the Euclidean Norm of a 3D vector. The method developed, dubbed ICP^2, combines both minimum normalized probabilities and computes a new probability value.

D. Hyperreality

Hyperreality is based upon a tight relationship between reality and virtual reality, i.e. the simulation. A fundamental aspect of the simulation is that it is updated whenever objects appear or change (in the robot's view) in reality. In this work we focus on the entry of new objects and do not consider changes to existing objects (which involves object grounding).
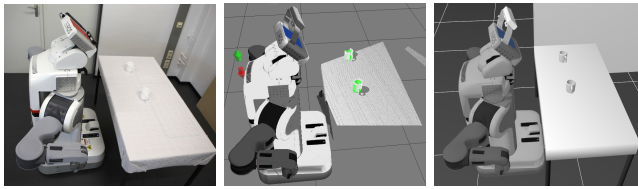
Fig. 1: (Table-top) objects recognized in reality are spawned into simulation concurrently. From left to right: reality, successful object recognition (RViz), spawned objects in simulation (Gazebo).

Fig. 2: The environment setup includes the counter, table and various table-top objects.

In a scenario in which the robot faces a table-top situation, it either will or will not successfully recognize the objects on the table. The key aspect of (traditional) 3D object recognition involves comparing a segmented RGB-D point cloud against an ideal 3D model stored in a database. The output of successful object recognition is the object identifier corresponding to the correct real world object. In our approach we extended the database with new attributes that store additional information about objects. This information is needed in order to create an instance of the object in the simulation. Objects in our database are referenced by their *Id* and include at least a *Label* and a *Category*. For Gazebo simulation compliance, we added the attributes *3D model* and *Gazebo description*. The 3D model holds a file name referring to a Collada[4] model which is used for object recognition and for spawning into Gazebo. The XML (Extensible Markup Language) Gazebo description represents the model information in either SDF (Simulation Description Format) or URDF[5] (Unified Robot Description Format) such that it can be spawned into Gazebo.

Successful object recognition provides the object identifier (*Id*) and also its 3D position (and orientation) which is transformed into the Gazebo world coordinate system and used to spawn the object at the correct location. Fig. 1 shows the robot situated in front of a counter recognizing the two mugs on top; the spawned objects are shown in the simulation. When re-creating a real situation in the simulation, the robot's position is also transformed after it has been retrieved from the AMCL (adaptive Monte Carlo localization). Any pose applied from the simulation into reality is transformed in the opposite direction. The applied affine transformation is dependent upon the robot's AMCL map relation to the Gazebo world and has to be calculated once for the environment setup in use.

III. SCENARIO AND EXPERIMENTS

In a restaurant a typical task for the waiter might be to serve a coffee in front of a guest. The setup of the physical environment is illustrated in Fig. 2. The robot is given the instruction *achieve serve_coffee_to_guest_Task guest1* through its command line interface (CLI). The planner is triggered and creates the plan shown in Listing 1 according to the problem description and current knowledge. Each line represents a plan operator with its arguments. The plan operators

[4]https://collada.org
[5]http://wiki.ros.org/urdf

correspond to robot capabilities and are bound to the robot. Arguments are instances from the robots ontology and are stored in the central component of the RACE architecture, the Blackboard. The Blackboard collects and distributes information between different system components. Fig. 4 shows the robot's environment with respect to its knowledge.

```
1   !tuck_arms armtuckedposture armtuckedposture
2   !move_base premanipulationareaeastcounter1
3   !move_torso torsoupposture
4    !tuck_arms armtuckedposture armuntuckedposture
5    !move_arm_to_side rightarm1
6   !move_base_blind manipulationareaeastcounter1
7   ! pick_up_object mug1 rightarm1
8   !move_base_blind premanipulationareaeastcounter1
9   !move_torso torsomiddleposture
10   ! move_arms_to_carryposture
11  !move_base premanipulationareasouthtable1
12  !move_torso torsoupposture
13   !move_arm_to_side rightarm1
14  !move_base_blind manipulationareasouthtable1
15  ! place_object mug1 rightarm1 placingareawestrighttable1
16  !move_base_blind premanipulationareasouthtable1
```

Listing 1: Generated (parallel) plan for serving a coffee.

```
11  !move_base premanipulationareanorthtable1
12  !move_torso torsoupposture
13   !move_arm_to_side rightarm1
14  !move_base_blind manipulationareanorthtable1
15  ! place_object mug1 rightarm1 placingareawestlefttable1
16  !move_base_blind premanipulationareanorthtable1
```

Listing 2: Generated (partial) alternative plan for serving a coffee. The adapted operator parameters are highlighted.

When the robot approaches the table from the south, as illustrated in Fig. 2, plan step 15 instructs the robot to place the mug on the table. This fails because various objects occupy the placing area and this would cause the complete plan to fail. HIRES is designed to reason about the next best action and to prevent the plan from failing. In the scenario presented, the "place" action is considered crucial and most likely to fail. Therefor it is chosen to be reasoned about before it is actually executed in reality.

Upon successfully picking up the mug (plan step 7) HIRES is started. HIRES uses the post-condition of the 14th plan step (the robot being at the manipulation area south of the table) and sets up the simulation accordingly (Fig. 3) in order to execute the next plan step (place the mug). This fails in the simulation too but this failure will not affect the original plan.
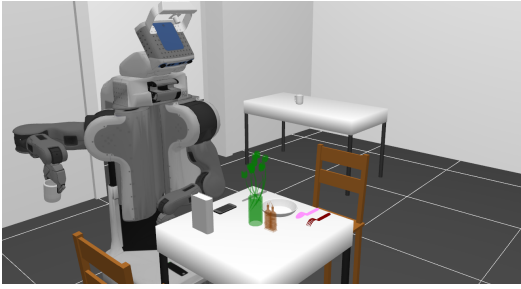
Fig. 3: The robot reaches the table to place the mug in front of the guest (omitted here, supposed to sit to the robot's left side of the table). As the placing area is occupied by various items the robot is not able to succeed.
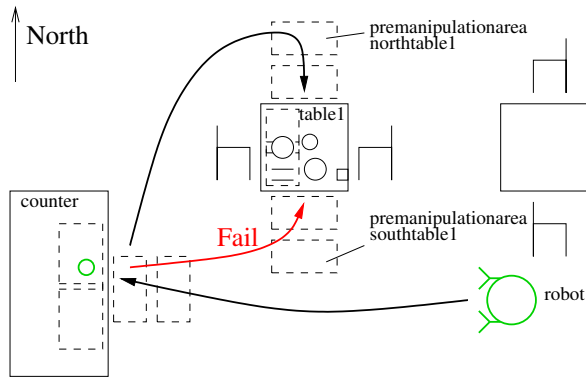

Fig. 4: Planning scene consisting of bounding boxes and coordinates for the table and counter and their approach positions. Labels correspond to plan operator parameters.

Moreover, it provides the plan with failure information before actual failure. The failure is detected by the return value of the "place" action. The planner, in return, is able to repair the current plan (a.k.a. re-planning) and to produce a (partial) new plan, as shown in Listing 2. This is possible because the two different manipulation areas (north and south) are valid parameters to the move base operator. As this work does not focus on particular re-planning methods, for the sake of simplicity the planner creates partial plans before executing them. This allows re-planning by simply changing the plan parameter and planning again (partially).

IV. EVALUATION

A. Object Recognition

The authors believe that a comprehensive evaluation of their approach to 3D recognition against methods that do not use the same sensor information is not possible. The strategy established here was to use many different detectors and various sensor information. The normal evaluation tools, such as [26] that provide more than 200,000 scans of 300 objects from 51 categories, are unsuitable. Our goal here was to evaluate object recognition w.r.t. sensors and perspectives. Both are typically robot platform dependent properties.

At first we evaluated our new object recognition detector in the simulation with all objects included in the ROS
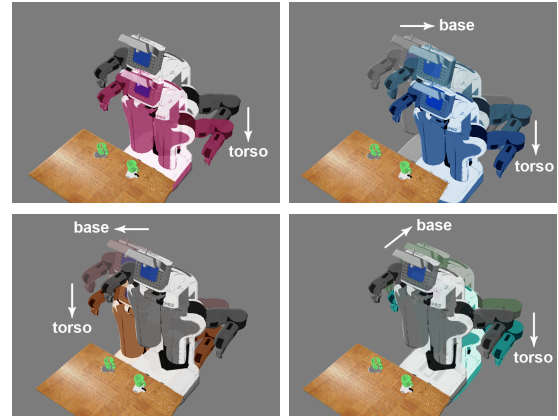

Fig. 6: Overlay of different poses tested within the simulation before giving a suggestion for object detection and grasp calculation. This image shows four tested poses with the robot in front of a table holding graspable objects.

household database[6]. We then tried many different evaluation strategies. For example the available objects were used to compare recognition results with the real world. A set of random or significant scenes was also investigated.

The results for mug recognition with our detector using the household database are presented in Fig. 5. The probability for the coffee mug is presented in red. The results confirm our assumptions presented in earlier publications: ICP^2 is stable and robust. The method first compares two possible object properties, namely size and volume. In this case the coffee mug has only the third-highest probability, ranking after two objects with extremely similar shapes, such as a glass and a can. The next step was to evaluate the same detector as part of the whole recognition system in order to investigate what can be achieved in combination with other detectors. Note that the position and orientation of the objects were changed automatically during the evaluation process, so an evaluation of the whole object recognition system w.r.t. partial occlusion has also been achieved.

B. Reasoning

One of the recurring tasks in our scenarios was to drive the robot to the counter, from which the robot picked up one or more objects. For our experiments we used the PR2 robot which has the ability to change its vertical torso position. This flexibility suggested two obvious questions for investigation: how does the robot's torso position affect grasping and object recognition and what kind of strategy gives the best results for both capabilities?

To find the answers, a table-top scene was recognized by the real robot. With this data, the real scene was replicated in the simulation. The task of the robot was to recognize the same scene in the simulation with the position of the simulated robot (base and torso) changed after each completed experiment. Fig. 6 visualizes the scenario used as well as the changes to the robot position.

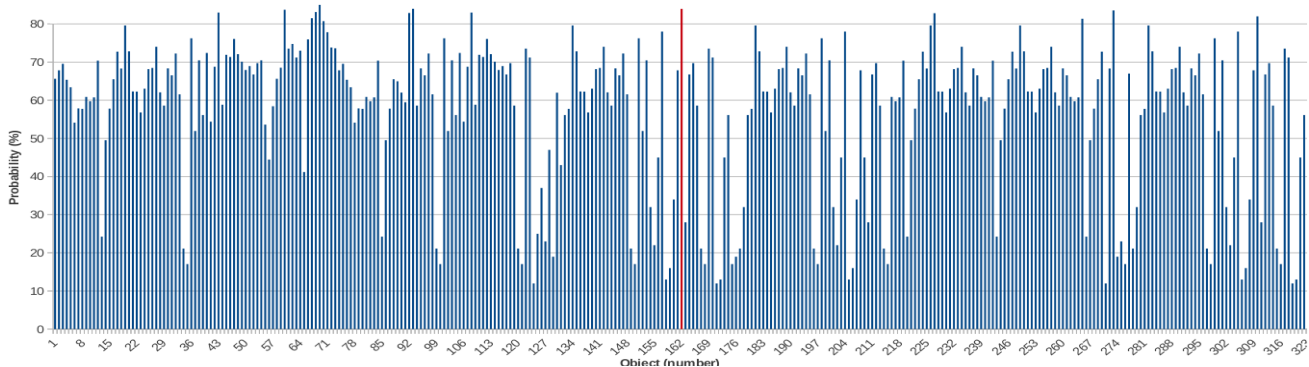[6]http://www.ros.org/wiki/household_objects_database

Fig. 5: Comparison of the mug with the 323 Objects (object number along the x-axes) of the ROS household database using the ICP^2 algorithm. The probability (y-axes) for the coffee mug is presented in red. The method compared only two possible object properties, namely size and volume. This was one reason why the coffee mug had only the third-highest probability, after objects with extremely similar shapes such as a glass and a can.
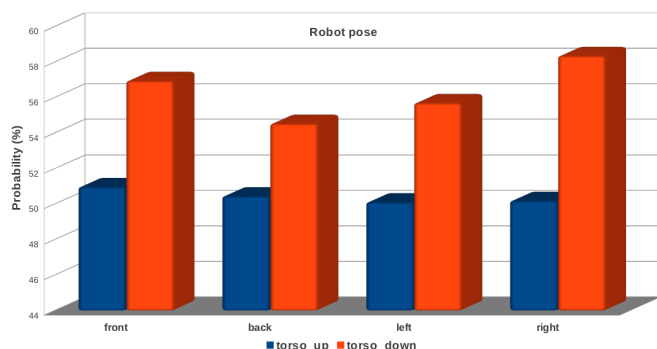

Fig. 7: Results for object recognition depend upon robot's torso and pose. This figure is related to the experiments illustrated in Fig. 6.

move_base parameter	premanipulationarea *southtable1*	premanipulationarea *northtable1*
Plan	fail	ok

TABLE I: Qualitative results: serving a coffee.

System	Default system + re-planning	HIRES
Duration	298 seconds	283 seconds

TABLE II: Averaged quantitative results of five scenario runs.

One of the first results was that the influence of torso position on object recognition has a linear character. The lower the torso position the more significant object voxels can be seen. Therefore only the "up" and "down" positions are considered to be relevant. As can be seen in Fig. 7 the main quality gain was measured when the robot's torso was in the down position. Other operators, such as moving the robot base left, right or back, did not significantly improve (and sometimes reduced) recognition quality.

The second result was that the number of grasps found for all torso positions tested was relatively constant. But the grasp quality increased by 32.5% when the robot's torso was driven to the "up" position from the initial pose. The information about the grasp quality was delivered from ROS' own grasp planning tool, which uses a simulated annealing optimization to search for gripper poses relative to the object or cluster [27]. Changes in the robot's base position to the left, right or back did not notably increase the overall grasp quality in that setup. For confirmation a few significant situations were reenacted and recognized in reality. The results confirmed the outcome of the simulation.

According to the experiments, the best resulting strategy is to use the "down" robot position for object recognition and to change it to "up" when searching for and calculating the necessary grasps. Consequently the use of this strategy

increases the robustness of both robot capabilities described below. This result can be used immediately by the planner or HIRES and influences the next calculated plan.

The scenario, presented in section III, shows a typical plan failure situation, in which the robot fails to perform a given task. The question posed is whether robot autonomy can be improved by HIRES. To compare the results, two runs of the scenario were performed. The first is executed by following the original plan but, instead of failing in the "place" task, re-planning at this point by driving the robot to the North (opposite) side of the table. The second run employs HIRES and therefore executes the (repaired) plan. Table I shows the qualitative results and Table II the quantitative results of the experiments performed in this scenario.

We increased the simulation time in this scenario by about 20 seconds. Although this is an order of magnitude slower than symbolic planning it prevents plan failure. Furthermore it would take approximately the same time to execute the plan and then re-plan but at the cost of spending robot resources. Future parallel simulation of additional parameters will compensate for this extra time cost.

V. CONCLUSIONS AND FUTURE WORK

In this work we introduced a reasoning and evaluation system based on Hyperreality and Imagination (HIRES). In order to measure its performance, we assessed the reasoning and evaluation parts in the context of a plan-based robot

system. We selected typical situations from a coffee serving scenario, (object recognition, grasping and placing a mug) for the evaluation of HIRES. Our recently introduced object recognition method was evaluated with HIRES; the results presented supported this new method as well as HIRES. We implemented HIRES and we contributed to improvements in overall system robustness and autonomy.

The contribution of HIRES was measured qualitatively and in terms of time costs. Time is a key resource in robotics, but the robot itself is also considered to be a resource. Our approach spared the robot from executing the dead-ends in a (failing) plan and prevented it from harming itself or its surroundings while acting in its real environment.

As our approach relies on real-time reasoning, it imposes a real-time requirement onto the simulation, especially by running multiple simulations in parallel to test different parametrization. This must be approached in future by choosing a simplified environment, changing the simulation engine, distributing the computing load or by interleaving simulation with execution. Parametrization can be scored according to the most likely success. We will use additional scenarios for evaluation, such as the robot carrying a pepper mill on a tray while parametrizing the driving, velocity, acceleration etc. A model to represent uncertain propositions, such as a fuzzy-model, will be evaluated.

Our future work will involve the application of HIRES to more scenarios and situations. Eventually, flexible integration with a symbolic planner will allow HIRES to be called from anywhere during plan execution. Performance issues were not considered in this work but have to be dealt with in future. Nevertheless we believe that increasing computing power will contribute to the targeted real time performance of HIRES.

ACKNOWLEDGMENT

REFERENCES

[1] J. Renz and B. Nebel, "Qualitative spatial reasoning using constraint calculi," in *Handbook of Spatial Logics*, M. Aiello, I. Pratt-Hartmann, and J. van Benthem, Eds. Springer, 2007, pp. 161–215.

[2] B. J. Loasby, "Cognition, imagination and institutions in demand creation," *Journal of Evolutionary Economics*, vol. 11, pp. 7–21, 2001.

[3] J. E. Lodato, "Moral imagination: Implications of cognitive science for ethics," *Dialogue: Canadian Philosophical Review/Revue canadienne de philosophie*, vol. 35, pp. 204–207, December 1996.

[4] N. Terashima and J. Tiffin, *HyperReality: Paradigm for the Third Millenium*. Routledge, 2001.

[5] J. Baudrillard, *Simulacra and simulation*. Univ. Michigan Press, 1994.

[6] S. Rockel, D. Klimentjew, and J. Zhang, "A multi-robot platform for mobile robots - a novel evaluation and development approach with multi-agent technology," in *IEEE Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2012, pp. 470–477.

[7] I.-H. Chen, B. MacDonald, and B. Wunsche, "Mixed reality simulation for mobile robots," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, May 2009, pp. 232–237.

[8] S. Rockel, B. Neuman, J. Zhang, K. S. R. Dubba, A. G. Cohn, Š. Konečný, M. Mansouri, F. Pecora, A. Saffiotti, M. Günther, S. Stock, J. Hertzberg, A. M. Tomé, A. J. Pinho, L. S. Lopes, S. von Riegen, and L. Hotz, "An ontology-based multi-level robot architecture for learning from experiences," in *Designing Intelligent Robots: Reintegrating AI II, AAAI Spring Symposium*, Stanford (USA), March 2013.

[9] D. Nau, H. M. noz Avila, Y. Cao, A. Lotem, and S. Mitchell, "Total-order planning with partially ordered subtasks," in *Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001)*, vol. 17, Seattle, 2001, pp. 425–430.

[10] J. Bohren, R. Rusu, E. Gil Jones, E. Marder-Eppstein, C. Pantofaru, M. Wise, L. Mosenlechner, W. Meeussen, and S. Holzer, "Towards autonomous robotic butlers: Lessons learned with the pr2," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 5568–5575.

[11] L. Einig, D. Klimentjew, S. Rockel, and J. Zhang, "Parallel plan execution and re-planning on a mobile robot using state machines with htn planning systems," in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, December 2013.

[12] L. P. Kaelbling and T. Lozano-Pérez, "Integrated robot task and motion planning in belief space," 2012.

[13] J. Barry, K. Hsiao, L. P. Kaelbling, and T. Lozano-Pérez, "Manipulation with multiple action types," in *International Symposium on Experimental Robotics*, June 2012.

[14] J. Barry, L. Kaelbling, and T. Lozano-Perez, "A hierarchical approach to manipulation with diverse actions," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013, pp. 1799–1806.

[15] A. Harris and J. M. Conrad, "Survey of popular robotics simulators, frameworks, and toolkits," in *IEEE Southeastcon*, 2011, pp. 243–249.

[16] A. Boeing and T. Bräunl, "Evaluation of real-time physics simulation systems," in *Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia*, ser. GRAPHITE '07. New York, NY, USA: ACM, 2007, pp. 281–288.

[17] E. Weitnauer, R. Haschke, and H. Ritter, "Evaluating a physics engine as an ingredient for physical reasoning," in *Simulation, Modeling, and Programming for Autonomous Robots*, ser. Lecture Notes in Computer Science, N. Ando, S. Balakirsky, T. Hemker, M. Reggiani, and O. Stryk, Eds. Springer Berlin Heidelberg, 2010, vol. 6472, pp. 144–155.

[18] L. Kunze, M. E. Dolha, and M. Beetz, "Logic programming with simulation-based temporal projection for everyday robot object manipulation," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, CA, USA, September, 25–30 2011, best Student Paper Finalist.

[19] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3D recognition and pose using the Viewpoint Feature Histogram," in *Proceedings of the 23rd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, October 18-22 2010.

[20] Sun, Xu, Bradski, and Savarese, "Depth-encoded hough voting for joint object detection and shape recovery," in *Proc. ECCV*, 2010.

[21] R. Rusu, N. Blodow, Z. Marton, and M. Beetz, "Close-range scene segmentation and reconstruction of 3d point cloud maps for mobile manipulation in domestic environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009.

[22] S. Srinivasa, D. Ferguson, C. Helfrich, D. Berenson, A. Collet, R. Diankov, G. Gallagher, G. Hollinger, J. Kuffner, and M. VandeWeghe, "Herb: A home exploring robotic butler," 2009.

[23] L. L. Wong, L. P. Kaelbling, and T. Lozano-Pérez, "Constructing semantic world models from partial views," in *Robotics: Science and Systems (RSS) Workshop on Robots in Clutter*, 2013.

[24] D. Klimentjew, S. Rockel, and J. Zhang, "Towards scene analysis based on multi-sensor fusion, active perception and mixed reality in mobile robotics," in *Proceedings of the IEEE First International Conference on Cognitive Systems and Information Processing (CSIP2012)*.

[25] ——, "Active scene analysis based on multi-sensor fusion and mixed reality on mobile systems," *Foundations and Practical Applications of Cognitive Systems and Information Processing (Chapter 69), Advances in Intelligent Systems and Computing*, vol. 215, pp. 795–810, 2013.

[26] K. Lai, L. BoR., X. Ren, and D. Fox, "A large-scale hierarchical multi-view rgb-d object database." *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[27] M. T. Ciocarlie, K. Hsiao, E. G. Jones, S. Chitta, R. B. Rusu, and I. A. Sucan, "Towards reliable grasping and manipulation in household environments," *ISER 2010*, pp. 241–252, 2010.