About us

The Institute for Artificial Intelligence (IAI) at the University of Bremen, Germany, headed by Prof. Michael Beetz, investigates methods for cognition-enabled robot control. The research is at the intersection of Robotics and Artificial Intelligence (AI) and includes methods for intelligent perception, dexterous object manipulation, plan-based robot control, and knowledge representation for robots.

Robots performing complex tasks in open domains, such as assisting humans in a household or collaboratively assembling products in a factory, need to have cognitive capabilities for interpreting their sensor data, understanding scenes, selecting and parameterizing their actions, recognizing and handling failures and interacting with humans. In our research, we are developing solutions for these kinds of issues and implement and test them on the robots in our laboratory. A particular focus of the group is on the integration of individual methods into complete cognition-enabled robot control systems and the release of the developed software as open-source libraries.

More information about the institute, the team and our research projects can be found on our website: http://ai.uni-bremen.de/

Four software libraries origin from our laboratory and are now used and supported by a larger user community. These are the KnowRob system for robot knowledge processing, the CRAM framework for plan-based robot control, openEASE for collecting and analyzing experiment data and RoboSherlock for cognitive perception.

KnowRob – Robot Knowledge Processing KnowRob is a knowledge processing system that combines knowledge representation and reasoning methods with techniques for acquiring knowledge from different sources and for grounding the knowledge in a physical system. It provides robots with knowledge to be used in their tasks, for example action descriptions, object models, environment maps, and models of the robot's hardware and capabilities. The knowledge base is complemented with reasoning methods and techniques for grounding abstract, high-level information about actions and objects in the perceived sensor data.

KnowRob became the main knowledge base in the ROS ecosystem and is actively being used in different academic and industrial research labs around the world. Several European research projects use the system for a wide range of applications, from understanding instructions from the Web (RoboHow), describing multi-robot search-and-rescue tasks (SHERPA), assisting elderly people in their homes (SRS) to industrial assembly tasks (SMERobotics).

KnowRob is an open-source project hosted at GitHub[1] that also provides extensive documentation on its website – from getting-started guides to tutorials for advanced topics in robot knowledge representation[2].

[1] http://github.com/knowrob

[2] http://www.knowrob.org/

CRAM – Robot Plans CRAM is a high-level system for designing and performing abstract robot plans to define intelligent robot behavior. It consists of a library of generic, robot platform independent plans, elaborate reasoning mechanisms for detecting and repairing plan failures, as well as interface modules for executing these plans on real robot hardware. It supplies robots with concurrent, reactive task execution capabilities and makes use of knowledge processing backends, such as KnowRob, for information retrieval.

CRAM builds ontop of the ROS ecosystem and is actively developed as an open-source project on GitHub[3]. It is the basis for high-level robot control in many parts of the world, especially in several European research projects covering applications from geometrically abstract object manipulation (RoboHow), multi-robot task coordination and execution (SHERPA), experience based task parameterization retrieval (RoboEarth), and safe human robot interaction (SAPHARI).
Further information, as well as documentation and application use-cases can be found at the CRAM website[4].

In our group, we have a very strong focus on open source software and active maintenance and integration of projects. The systems we develop are available under BSD license, and partly (L)GPL.

OpenEASE – Experiment Knowledge Database OpenEASE is a generic knowledge database for collecting and analysing experiment data. Its foundation is the KnowRob knowledge processing system and ROS, enhanced by reasoning mechanisms and a web interface developed for inspecting comprehensive experiment logs. These logs can be recorded for example from complex CRAM plan executions, virtual reality experiments, or human tracking systems. OpenEASE offers interfaces for both, human researchers that want to visually inspect what has happened during a robot experiment, and robots that want to reason about previous task executions in order to improve their behavior.

The OpenEASE web interface as well as further information and publication material can be accessed through its publicly available website[5]. It is meant to make complex experiment data available to research fields adjacent to robotics, and to foster an intuition about robot experience data.

OpenEASE is being developed as an Open Source, Open Science project aiming to make complex information available to researchers that otherwise don't have access to this kind of experiment data. All parts of the system are being developed under either the BSD license, or (L)GPL.

RoboSherlock – Framework for Cognitive Perception RoboSherlock is a common framework for cognitive perception, based on the principle of unstructured information management (UIM). UIM has proven itself to be a powerful paradigm for scaling intelligent information and question answering systems towards real-world complexity (i.e. the Watson system from IBM). Complexity in UIM is handled by identifying (or hypothesizing) pieces of structured information in unstructured documents, by applying ensembles of experts for annotating information pieces, and by testing and integrating these isolated annotations into a comprehensive interpretation of the document.

RoboSherlock builds on top of the ROS ecosystem and is able to wrap almost any existing perception algorithm/framework, and allows easy and coherent combination of the results of these. The framework has a close integration with two of the most popular libraries used in robotic perception, namely OpneCV and PCL. More details about RoboSherlock can be found on the project web-page[6]

RoboSherlock is being developed as an Open Source project under the terms of the Apache v2.0 license.

[3] http://github.com/cram-code
[4] http://cram-system.org/
[5] http://www.open-ease.org/
[6] www.robosherlock.org

Proposed Topics for Summer of Code-Projects

In the following, we list our proposals for the Google Summer of Code topics that contribute to the aforementioned open-source projects.

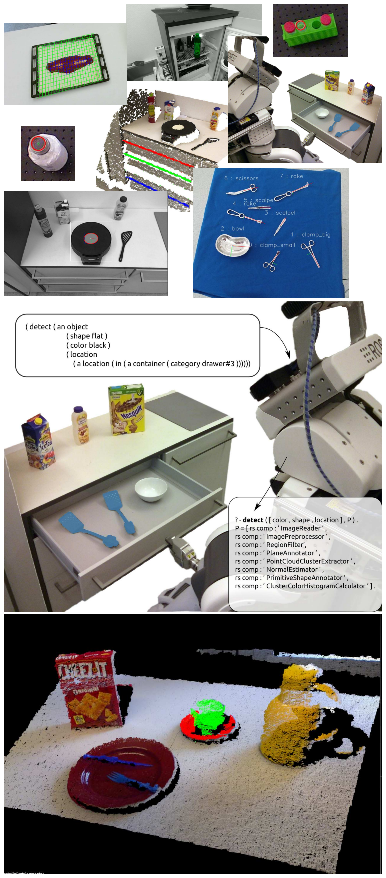
Topic 1: Multi-modal Cluttered Scene Analysis in Knowledge Intensive Scenarios

Main Objective The main objective of this topic is to enable robots in a human environment to recognize objects in difficult and challenging scenarios. To achieve this the participant will develop software components for RoboSherlock, called annotators, that are aided by background knowledge in order to detect objects. These scenarios include stacked,occluded objects placed on shelves, objects in drawers, refrigerators, dishwashers, cupboards etc. In typical scenarios, these confined spaces also bare an underlying structure, which will be exploited, and used as background knowledge, to aid perception (e.g. stacked plates would show up as parallel lines using an edge detection).

Task Difficulty The task is considered to be challenging, as it is still a hot research topic where general solutions do not exist.

Requirements Good programming skills in C++ and basic knowledge of CMake. Experience with PCL, OpenCV is prefered. Knowledge of Prolog is a plus.

Expected Results Currently the RoboSherlock framework lacks good perception algorithms that can generate object-hypotheses in challenging scenarios(clutter and/or occlusion). The expected results are several software components based on recent advances in cluttered scene analysis that are able to successfully recognized objects in the scenarios mentioned in the objectives, or a subset of these.

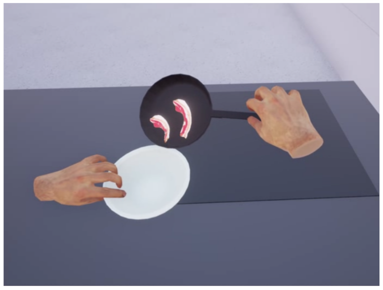


```
( detect ( an object
           ( shape flat )
           ( color black )
           ( location
             ( a location ( in ( a container ( category drawer#3 ))))))
```

```
? - detect ( [ color , shape , location ] , P ) .
P = [ rs comp : ' ImageReader ' ,
rs comp : ' ImagePreprocessor ' ,
rs comp : ' RegionFilter',
rs comp : ' PlaneAnnotator ' ,
rs comp : ' PointCloudClusterExtractor ' ,
rs comp : ' NormalEstimator ' ,
rs comp : ' PrimitiveShapeAnnotator ' ,
rs comp : ' ClusterColorHistogramCalculator ' ] .
```

Topic 2: Realistic Grasping using Unreal Engine

Main Objective The objective of the project is to implement various human-like grasping approaches in a game developed using Unreal Engine [7].
The game consist of a household environment where a user has to execute various given tasks, such as cooking a dish, setting the table, cleaning the dishes etc. The interaction is done using various sensors to map the users hands onto the virtual hands in the game.
In order to improve the ease of manipulating objects the user should be able to switch during runtime the type of grasp (pinch, power grasp, precision grip etc.) he/she would like to use.

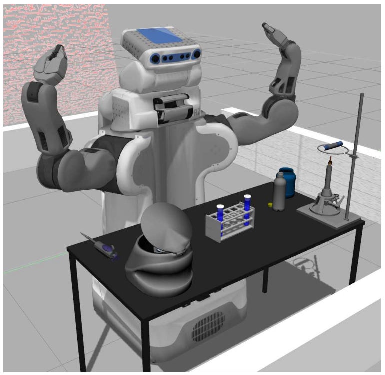


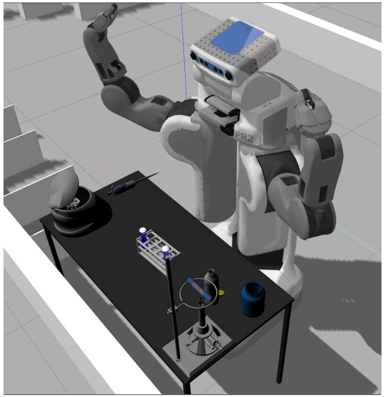
Task Difficulty The task is to be placed in the easy difficulty level, as it requires less algorithmic knowledge and more programming skills.

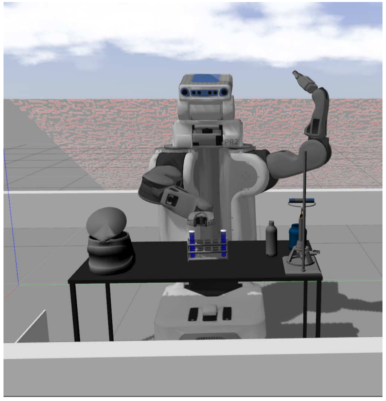
Requirements Good programming skills in C++. Good knowledge of the Unreal Engine API. Experience with skeletal control / animations / 3D models in Unreal Engine.

Expected Results We expect to enhance our currently developed robot learning game with realistic human-like grasping capabilities. These would allow users to interact more realistically with the given virtual environment. Having the possibility to manipulate objects of various shapes and sizes will allow to increase the repertoire of the executed tasks in the game. Being able to switch between specific grasps will allow us to learn grasping models specific to each manipulated object.

Topic 3: Plan Library for Autonomous Robots performing Chemical Experiments







Main Objective of this theme is to develop in Gazebo simulator [8] a set of plan-based control programs which will equip an autonomous mobile robot to perform a set of typical manipulations within a chemistry laboratory. The set of plan-based control programs resulted at the end of the program will be tested on the real PR2 robot from the Institute for Artificial Intelligence of the University of Bremen, Germany.

The successful candidate will use the domain specific language of CRAM [9] toolbox and code plan-based control programs which will enable the PR2 robot to perform manipulations like: simple grasping of different containers, screwing and unscrewing the cap of a test tube, pouring a substance from a container into another container, operating a centrifuge, etc.

In the first phase of the project the successful candidate will make sure he/she is familiar with the domain specific language of CRAM toolbox and the parameters of the plan-based control programs. This phase will culminate with the student having coded a simple complete and fully runnable plan-based control program.

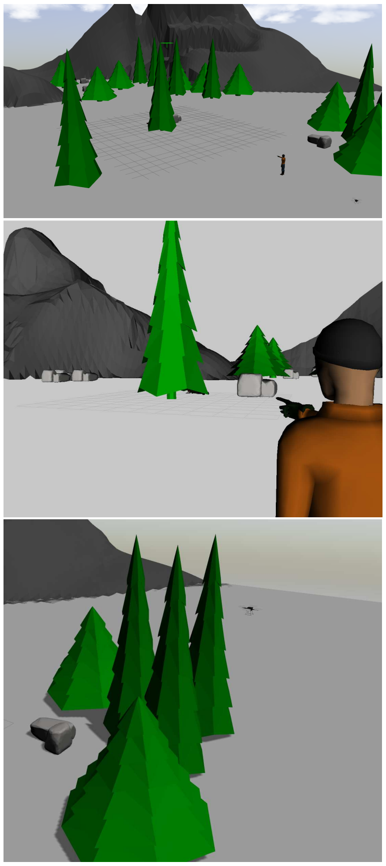
In the second phase of the project together with the successful candidate we will decide the set of manipulations which will be implemented in order to enable the robot to perform a simple and complete chemical experiment.

In the last phase of the project, the plan-based control programs developed in the second phase will be put together and the complete chemical experiment will be tested and fixed until it runs successfully. The set of plan-based control programs resulted at the end of the program will represent the execution basis of the future experiments which will be done at IAI in order to figure out how an autonomous robot can reproduce a chemical experiment represented with semantic web tools.

Requirements The ideal candidate must be comfortable programming in LISP and familiar with the ROS [10] concepts. The candidate familiar with the Gazebo simulator [11] and CRAM [12] toolbox will have a big plus.

Expected Results We expect to successfully code a library of plan-based control programs which will enable an autonomous robot to manipulate the typical chemical laboratory equipment and perform a small class of chemical experiments in Gazebo simulator.

Topic 4: Multi-modal Interaction in Human-Robot Teams in Outdoor Environments



Main Objective The objective of this project is to enable the external communication and interaction of a human with robots in the simulation in order to accomplish tasks together. To achieve this goal, the participant will develop an interface that enables to give written instructions, language instructions or instructions by clicking on an object in the environment that has to be performed by the robots. Furthermore, the history of the given instructions and the position of the robots such as paths and goal poses have to be stored as an action script in order to comprehend and to understand the interpreted task.

Task Difficulty The task is to be placed easy difficulty and requires good programming skills.

Requirements The ideal candidate must have good programming skills in C++ and Python and shall be familiar with the ROS concept and the Gazebo Simulator.

Expected Results We expect to enhance our currently developed human-robot simulation with an external human instead of a simulated human in order to ensure a realistic human-robot communication. The history of the task execution enables the understanding and reasoning about the behaviour that could support and improve human-robot interaction and understanding in real scenarios.