Serious Cybernetics Corporation 2024 Team Description Paper

B. Staehle, P. Prasad, S. Hafner and M. Knof

November 27, 2023

Abstract This paper introduces the team Serious Cybernetics Corporation(SCC) of the Ravensburg-Weingarten University of Applied Sciences, Germany, along with it's robots *Marvin*, *Kate* and *Kurt* as part of the qualification for the Robocup@Home 2024 tournament in Eindhoven. The primary research focus of the group lies in Machine-Learning applications for robotic systems. In this paper we describe an approach to anomaly detection in household environments, learning from demonstration, person tracking, re-identification and data augmentation. Also our efforts to make a robot system more scalable and reliable using container technology alongside with a graph based knowledge concept will be illustrated.

1 Introduction and Achievements in the @Home League

The Serious Cybernetics Corporation is the robocup@Home student team of Ravensburg-Weingarten University of Applied Sciences (RWU)[1] from south Germany. The team was formed in 2016 and has successfully participated in various robocup@Home competitions since 2017: Robocup World Cup: 1 x 8th Place(2023), 1 x Best Poster Award (2021), 1 x Open Challenge Award (2023), Robocup German Open: 3 x 5th Place (2017, 2018, 2019), European Robotics League: 1 x 1st Place tournament Oldenburg (2022).

The team consists of PhD / Master / Bachelor students and research assistants. Before focusing on Robocup@Home the group participated in the Robocup Soccer Middle Size League in 2005. The team is located and backed by the Institute for Artificial Intelligence (IKI)[2] where research in the fields of Machine-Learning and Service-Robotics (SR) is conducted.

The first SR platform *Kate* was developed in 2009 in a joint research program called ZAFH-Servicerobotik[3]. In 2013 the demand for a robotic platform with

[1] https://rwu.de
[2] https://iki.rwu.de
[3] http://www.zafh-servicerobotik.de

omnidirectional driving capabilities and the ability to manipulate objects on the floor as well as in heights around 2 meters above ground was formulated in a research project to aid people with physical disabilities AsRoBe[4]. This resulted in the development of *Marvin*. Since 2018 the team also has a commercial PAL Robotics TiaGO Robot at disposal that is named *Kurt* and has been used in the project ROBOTKOOP[5]. This project was focused on automated anomaly detection in household environments and the development of negotiation and cooperation strategies between robots and humans. All robots are using the same software stack based on docker and ROS. Further details can be found in the annex.

2 Technology and Scientific Contribution

2.1 Learning from Demonstration

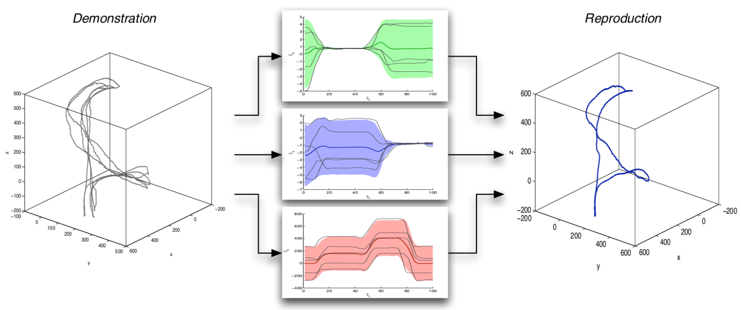


Figure 1: Using Gaussian Processes to form a generalized trajectory based on three demonstrations

Especially when there is little time to adapt to a new situation the complexity of solving a new task reliable in a generalised manner remains challenging. In this regard, Learning from Demonstration(LfD) is a intuitive and easy way to show the robot how to open an unknown drawer or grabbing a coat from a unknown hook without the need of programming and fine-tuning. We currently use a approach based on Gaussian processes [1] and Averaging Trajectories[2]. This method is able to extract all necessary information for a reproduction from a small number of demonstrations as shown in Figure 1. Those demonstrations are executed by manually moving the robot arm recording the arm joint movements with respect to the recognized objects in the scene. The result is a generalized trajectory that can be used in any scenario where the objects from

[4] http://asrobe.hs-weingarten.de/
[5] https://www.technik-zum-menschen-bringen.de/projekte/robotkoop

the demonstrations are present (e.g. pouring a drink into a cup). A reference implementation is available on github[6].
As this method is limited to the used arm during the demonstration phase, further research is conducted in this area and a more general version will be published soon.

2.2 Anomaly Detection and correctly arranging Objects

To perform any task in complex environments, like the @Home arena, the robot needs to have a deeper understanding of how its surroundings are structured. For instance how Objects are related to one another in specific scenarios, e.g. when setting up a table. We have developed a methodology that enables the robot to acquire the knowledge how to solve such a task by observing its environment in a normal state. We use a set of Gaussian Mixture Models (GMMs) to model object properties and object relations in scenes that are considered normal, in order to detect anomalies that occur when the objects are either misplaced in an absolute sense or relative to surrounding objects. The developed software is able to capture multiple configurations of objects and is also capable of learning object relations based on context as shown in Figure 2. After the training, the method is able to classify situations as shown in Figure 3 and can also provide instructions how to transform the scene into a normal state again (e.g. the cup has to be moved 5 cm to the left to achieve a normal state) by calculating the distance of the object to the mean average position resulting from the GMMs. This method can be applied in any scenario where the arrangement of objects is relevant.

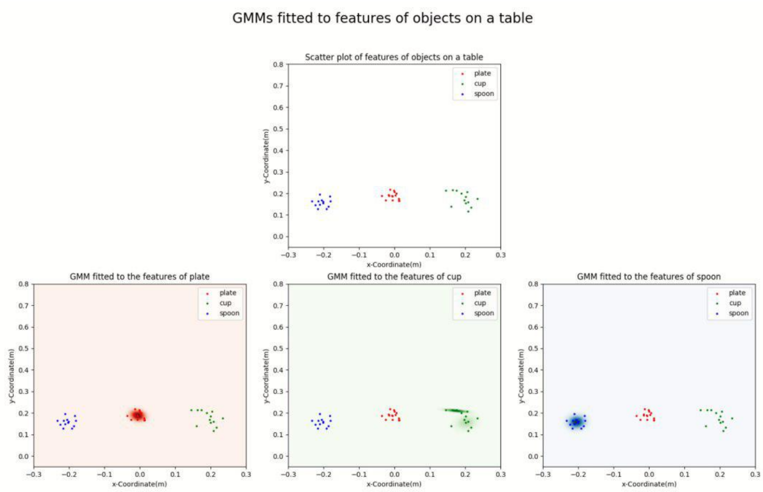


Figure 2: Learned GMMs derived from observing a table scene

[6] https://github.com/iki-wgt/Learning-from-demonstration-LAT

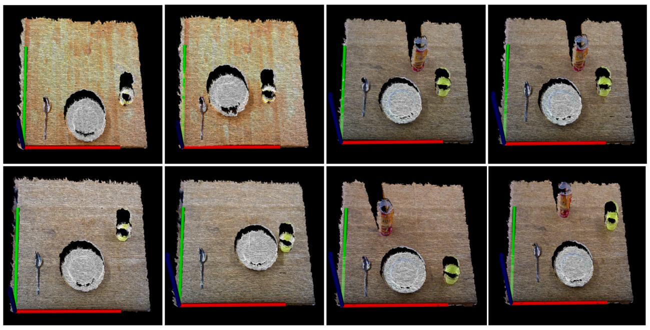
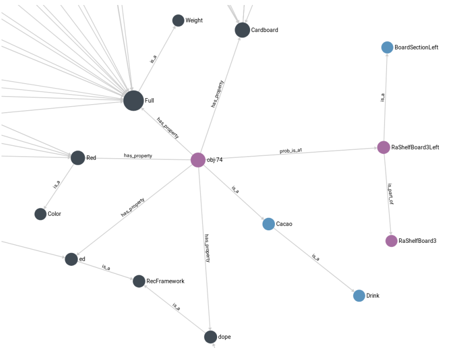


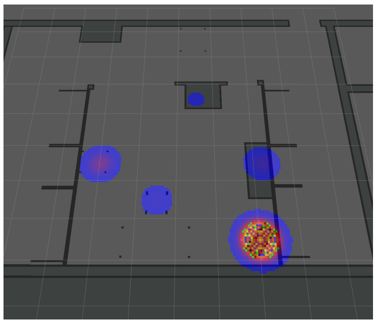
Figure 3: Validation scenarios for the learned GMM model

2.3 Knowledge Representation

In order to perform complex tasks pertaining to finding objects or answering specific questions about its surroundings, a knowledge storage system is essential [3]. We are currently developing a framework, which is based on the graph database system Arango DB[7]. This database model is suitable because it allows complex knowledge modeling that remains efficient when scaling to a large number of data points. Also the proposed system is cluster capable and could therefore be scaled up to multiple robots.



(a) Ontology example of a drink object.



(b) Learned heatmap for object locations.

Figure 4

The developed ontology is based on relationships between objects. Figure 4a shows a small part of such a relationship model. The relationship-model allows for arbitrarily complex models of the environment as different relationship types can be specified. Locations of an object, for example, are just relations

[7] https://www.arangodb.com/

between two different objects stored in the database. This relationships also get a probability. If the location of a item is requested later on, the knowledge-base can compose a list of different locations, where the object might be. Also more complex questions can be answered via this relationship network. The knowledge-base is coupled with our object recognition pipeline. As the robot drives around, any object recognized is added to the database creating a vast network of objects and their locations. The more the robot observes its environment, the better it learns, where the objects should be placed. If it gets the task to clean up a room, it can rely on this knowledge and perform the task to the users satisfaction. A second perk is the ability to inference knowledge from the underlying hierarchy of objects, this enables the robot to answer questions regarding the environment and can also form alternative suggestions. If, for example, the user asks for a Cola, but only a Fanta is available, the robot can suggest it as it is part of the same drink category.

2.4 Anomaly detection and Solution Strategies using Graph Neural Networks

This sections briefs about our Graph Neural Network(GNN) based approach for detecting anomalies and solving them [4]. Initially an ideal world model is created as a knowledge graph. This world model is constructed in the form of a graph database as mentioned in the previous section. Given any situation, the goal of the graph neural network is the detect anomalies and provide solutions for them in the form of robot actions. The training data for the GNN contains several situation graphs that contain anomalies(misplaced objects, misplaced furniture) and are labelled with action vectors which would solve these anomalies. The trained graph neural network can then be used to detect various anomalies and get solutions for them. An example is illustrated in Figure 5.

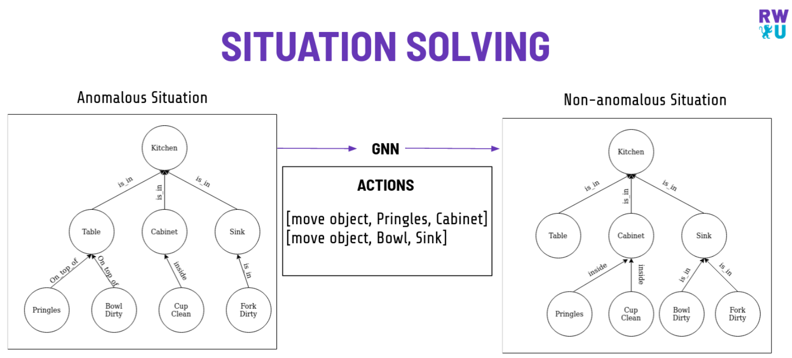


Figure 5: Anomaly detection and solving using GNN

3 Human Robot Interaction

Human robot interaction is an important component of human-centered service robots to function in a household environment. Therefore, we use a robot-dialogue framework to handle sophisticated robot-initiated interaction [5]. To become a truly collaborative companion, the assistant is able to engage in a proactive conversation for task assistance. The system actions are triggered by the recognition of user commands. The dialogue system uses a Rasa Natural language understanding (NLU) and a Rasa core to generate responses. A user interface is also provided with a text to speech engine to have text based interactions with the robot. Figure 6 shows the HRI system architecture.

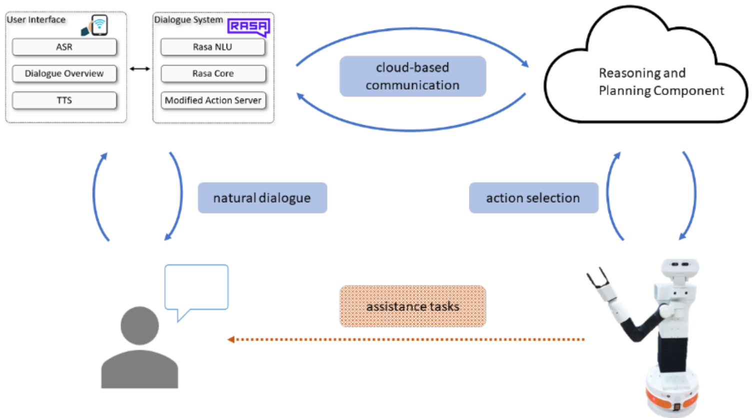


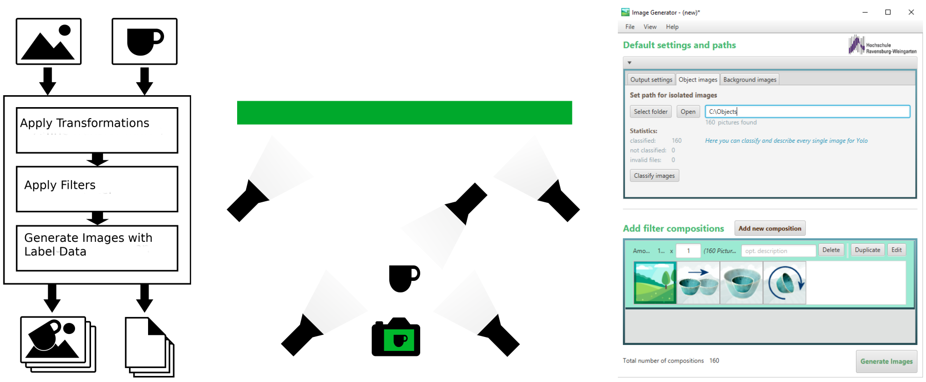
Figure 6: Human Robot Interaction Proactive dialogue System

4 Enhancing Robot Robustness using Docker

Service robots have to use complex software which sometimes is distributed on multiple systems with different architectures. As ROS is a system which already provides the capabilities to create highly distributed software through its system of independent nodes, every node, or sometimes group of nodes, get their own, specially designed Docker container. Such a container is started from a specially built image, that only contains the software needed to run that particular ROS-node. This enables us to also use software whose dependencies contradict each other. A further advantage is the flexibility this system provides. A ROS-node is no longer bound to run on a certain system which meets the dependency requirements but can be run on any system with Linux OS and Docker installed. On robots with multiple computing devices this also allows sophisticated load balancing strategies. The user only states which software module should be started and the system chooses the best available computing device depending on the current system load. This can also include external devices, for example a powerful GPU computer if available.

This docker based system is completed with a automatic image build on our GitLab code versioning tool. There the most recent, stable code is automatically built into Docker images and pushed to a Docker image registry. From there, the robots can automatically obtain those images if needed. This makes frequent software-updates easy to execute.
Regarding future improvements, an automated software testing is assembled. With unit tests to integration tests on fully simulated robots every aspect can be tested with a click of a button. This allows for more stable development as changes in huge software stacks tend to have effects in unexpected modules.

5 Automated Dataset Enhancement for Image Recognition Systems



(a) (b) (c)

Figure 7: a: Recommended light configuration for green screen shooting b: GUI to configure transformation / filter settings for dataset generation c: Data generator process overview

The acquisition of labeled training data for object recognition systems, e.g. deep learning frameworks such as YOLO[8] is of upmost importance as the performance of the later system depends on it. The availability of public datasets is small depending on the objects and especially for datasets with segmentation. Also most of the time, only common objects are included into a dataset, more special items are missing. Deep Learning frameworks require a lot of labeled training data in order to work satisfactory. The creation of such labeled data is therefore associated with considerable effort. A small research team can neither

[8] https://github.com/pjreddie/darknet

afford to buy such datasets nor can it create such themselves. For those reasons we have developed a framework which is able to place the object on arbitrary backgrounds and to compose a variety of images using filters and transformation methods as shown in Figure 7. There is also a java based GUI available as shown in Figure 7 to make the labeling and configuration process for different datasets easier. To perform such tasks the objects need to be separated from their respective background. We have automated this process using a green screen and have also evaluated different light configurations as shown in Figure 7b to reduce disturbing factors such as reflections and highlights. As the position in the image is known, the annotation of the image with a bounding and segmenting box is easy and is performed automatically. This reduces the amount of work to create a dataset remarkably. To determine the quality of the created training dataset, a YOLOv2[6] network was trained and tested. Our experiments conclude, that the artificial training data generation can be helpful for some objects, but not for all. The source code along with documentation and evaluation paper is available on github[9].

References

1. M. Schneider and W. Ertel. Robot learning by demonstration with local gaussian process regression. In *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)*, Taipeh, Taiwan, 2010.
2. H. Posenauer B. Reiner, W. Ertel and M. Schneider. Lat: A simple learning from demonstration method. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'14), Chicago, Illinois, USA*, 2014.
3. Pranav Krishna Prasad and Wolfgang Ertel. Knowledge acquisition and reasoning systems for service robots: A short review of the state of the art. In *2020 5th International Conference on Robotics and Automation Engineering (ICRAE)*, pages 36–45, 2020.
4. P.K. Prasad D. Hofer and M. Schneider. Comparison of anomaly detection and solution strategies for household service robotics using knowledge graphs. In *2021 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*, pages 1–6, 2021.
5. Matthias Kraus, Nicolas Wagner, Wolfgang Minker, Ankita Agrawal, Artur Schmidt, Pranav Krishna Prasad, and Wolfgang Ertel. Kurt: A household assistance robot capable of proactive dialogue. In *Proceedings of the 2022 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '22, page 855–859. IEEE Press, 2022.
6. Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.
7. Timm Linder, Stefan Breuers, Bastian Leibe, and Kai O Arras. On multi-modal people tracking from mobile platforms in very crowded and dynamic environments. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5512–5519. IEEE, 2016.

[9] https://github.com/iki-wgt/Yolo-training-data-generator

SCC Robots Hardware Description

(a) (b)

Figure 8: a: Self developed Robots at the IKI. Names (left to right) Kunibert, Kate and Marvin b: Kurt (TiaGO Pal Robotics)

We briefly describe our three robots that have been used in the @Home league so far. At the time of this writing the robots *Kate* and *Kurt* are planned to participate in the tournament this season.

Robot Name	Kate	Marvin	Kurt
Dimensions(LWH)	0.56 0.50 1.40	0.70 0.56 1.75	0.54 0.54 1.54
Weight	50Kg	150Kg	72Kg
Manufacturer	Self built	Self built	PAL Robotics
Motion Style	Differential	Omni-Directional	Differential
Manipulator	Katana 630 5DOF	Kinova Jaco 2 6DOF	Proprietary / unknown 6DOF
Payload	0.5Kg	2.5Kg	2Kg
Gripper	Angular actuated	Angular underactuated	Parallel
Base	Customized Pioneer 2AT	Neobotix MPO700	Proprietary / unknown
Battery	E-Bike LitiumIon 36V 14Ah	AGM sealed lead acid 48 V 28Ah	Proprietary / unknown
RGB-D Sensors	2	2	1
Laser Scanners	2	3	1
Sonar	5	0	3
Number of PCs (number of GPUs)	2(1)	2(1)	2(1)

Used Software

As every component is running in a separate docker container we are using mixed combinations of operating systems and ROS versions

Operating Systems	Ubuntu 16.04 / 18.04 / 20.04
Middleware	ROS Kinetic / Melodic / Noetic
Simulation	http://gazebosim.org
Localization	http://wiki.ros.org/amcl http://wiki.ros.org/slam_gmapping https://github.com/tue-robotics (Testing ED + Navigation)
Navigation	http://wiki.ros.org/move base
Arms control	http://moveit.ros.org/
Object recogntion	https://github.com/leggedrobotics/darknet_ros https://github.com/v4r-tuwien/v4r
Pose recognition	https://github.com/CMU-Perceptual-Computing-Lab/openpose
Face recognition	https://cmusatyalab.github.io/openface/
Person tracking	https://github.com/spencer-project/spencer_people_tracking/
Speech recognition	https://www.nuance.com/mobile/speech-recognition-solutions/vocon-hybrid.html
Speech generation	https://www.cereproc.com/

External Devices

– External GPU Server (optional)