

# Human-Friendly Interaction for Learning and Cooperation \*

Steen Kristensen Sven Horstmann Jesko Klandt Frieder Lohnert Andreas Stopp

DaimlerChrysler Research and Technology, Cognition and Robotics

Alt-Moabit 96A, D-10559 Berlin, Germany

E-mail: {steen.kristensen, sven.horstmann, jesko.klandt, frieder.lohnert, andreas.stopp}@daimlerchrysler.com

## Abstract

*In this paper, research towards a learning, cooperative robotic assistant is presented. The aim of this research is to develop a robot which can easily be instructed how to either perform tasks autonomously or in cooperation with humans. We describe the underlying representations and the methods developed for teaching new tasks and environments. The functionality has been demonstrated in a number of factory and office settings. In this paper, an example from a service scenario in an office environment is presented.*

## 1 Introduction

In this paper we describe past and ongoing research and development efforts at DaimlerChrysler Research and Technology's Cognition and Robotics Group where over the last years work has been conducted on human-friendly robots for space, office, and factory automation.

A major goal of this work has been (and still is) to develop robots that can assist, co-exist with, and be taught by humans. Therefore, apart from developing the "standard" mobile robot capabilities such as landmark recognition, path planning, obstacle avoidance etc. our research effort has been aimed at the development of learning capabilities that will allow the user to quickly and intuitively teach the robot new environments, new objects, new skills, and new tasks. We believe this is the only viable way of creating robotic assistants that can be flexible enough to function robustly in the very diverse habitats of humans and thus be accepted as truly helpful devices. In this paper we present some of the results of this work.

Current research is aimed towards improving the man-machine interaction by adding more advanced communication and cognition capabilities. This has the purpose

of further simplifying the teaching of the robot but also to make it more "cooperative" by having it interpret human commands and behaviour in the given context, allowing it to make better decisions about when, how, and where to assist the human co-worker(s). An important criterion is, however, that the robot can also perform tasks autonomously once instructed/taught by a human worker. Additionally it should be able to learn incrementally, i.e. to improve its performance during task execution by "passively" receiving or actively requesting information (the latter could for example be in the case where the robot detects ambiguities which it cannot autonomously resolve).

A typical scenario for a new robotic assistant in an industrial setting would be:

- The robot is led through the factory halls and is shown important places (stores, work stations, work cells etc.),
- the robot is shown relevant objects, e.g. tools, work pieces, and containers,
- the robot is shown how to dock by work cells, containers etc. in order to perform the relevant manipulation tasks,
- the robot is taught how to grasp various objects and how (and possibly in what sequence) to place them in corresponding containers or work cells,
- in case of a cooperation task, the robot is shown when and how to assist the human worker.

In Section 2 some more background in terms of related work and project specific constraints are outlined. In Section 3 it is explained how the representations necessary for the planning and execution of tasks are taught. In Section 4 an example of system operation is presented. Finally in Section 5, the results are discussed and summarised.

---

\*This research was partly sponsored by the German Ministry for Education and Research under the projects NEUROS, Neural Skills for Intelligent Robot Systems, and MORPHA, Intelligent Anthropomorphic Assistance Systems.

## 2 Environment Representations

All but the most simple, purely reactive robots internalise some kind of environment representation and can thus be said to “learn” their environment. Typically, an evidence grid representation of the obstacle/freespace conditions in the environment [1] is generated. The kind of representation we want the robot to learn, however, needs to fulfil the following conditions:

- It should be usable and thus valid over extended periods of time so that the robot only has to be taught once or whenever the environment has changed “significantly”. However, it should also be possible to adapt and modify existing models (either automatically or manually) to improve system performance.
- It should facilitate the user specification and automatic planning of mission commands such as “bring the parts to work station 5”, i.e., it needs to be a (partly) symbolic representation.
- It should contain the necessary information for the execution of planned tasks. This means that for example information about landmarks and objects must be present in the model.

A type of representation which satisfies these conditions is the annotated topological graph. In this graph representation, nodes represent distinctive places and edges the existence of traversable paths between the places. This purely topological representation is often annotated with metric information about the location of the places plus other information such as what local control strategies (LCSs)<sup>1</sup> to employ when traversing a given path. As indicated above, we would like to include information about landmarks and objects.

Topological maps are known from cognitive science but have also been widely employed in mobile robotics [3, 4, 5, 6, 7]. This research also shows how it is possible to automatically learn the topological graphs with exploration strategies. For most applications, however, we believe it is more sensible to let the robot learn the environment in interaction with the user since autonomous exploration has a number of problems:

- It cannot be guaranteed that the robot will see all the relevant parts of the surroundings due to obstacles, closed doors, people moving about etc.
- There may be areas where the robot is not allowed/supposed to go. This will often be the case for, e.g. office and factory applications.
- The resulting topological graphs are “un-grounded” in the sense that nodes do not have meaningful symbolic names such as “work cell”, “printer room” etc.

---

<sup>1</sup>After [2].

This can of course be added later, but then again there is no guarantee that the nodes to be named correspond to what the user intuitively would define as “places” or “rooms”.

Although autonomous exploration is possible with our system it has therefore been chosen to let the robot learn the topological graph in interaction with the user. This, however, only means that the user should be able to direct the robot through the environment and point out/name specific places and objects, i.e., it does *not* mean that the user should also point out appropriate landmarks nor provide metric information. How this information is taught is explained in Section 3.

## 3 Interactive Teaching

The basic concept behind the teaching of world model information is that it should be interactive (for reasons explained in Sect. 2) and that it should be done on-line and on-site with the real system. The latter has the advantage that the user gets direct feedback from the system regarding whether the robot can do what it is intended to do. Furthermore, it ensured that the representations are perfectly matched to the system in the sense that for example features detected and entered into the world model by a recogniser<sup>2</sup> are guaranteed to be detectable by at least one recogniser, namely the one originally entering it.

### 3.1 The World Model

In Sect. 2 it was argued that the annotated topological graph is a suitable choice of model representation because it satisfies the demands for longevity and because it supports the planning and execution of missions. In the following, the components of the topological graph, as we have chosen to implement it, will be elaborated.

**Nodes** The nodes represent “places” in the environment. Since the nodes are used for the mission planning as well as for the navigation, we will define places to be rooms, special locations of important objects, plus points significant for the navigation such as doorways and intersections. The nodes carry a 2D position and a covariance matrix representing the uncertainty of the position.

**Edges** The edges represent traversable paths between places, i.e. if such an edge exists the mission planner can assume that the robot can move between the connected places.

---

<sup>2</sup>A *recogniser* is a sensor data processing module specialised for recognising one or more types of features.

**Features** We define features as characteristics of the environment which can be used for navigational purposes and that can be detected “on the fly” by the robot’s recognisers. Typical examples of indoor environments are doors and walls. Features have—similar to nodes—a position and the corresponding uncertainty associated.

**3D-Objects** These objects are special objects used for tasks in which the 3D shape is of importance. In our system, these are docking objects and objects to be grasped and transported. 3D-Objects can—depending on the robot’s capabilities—also be features, but due to the normally rather limited on-board processing power and sensing capabilities of mobile robots, it is not feasible to continuously recognise these objects.

**Links** The links are used to relate the features and 3D-objects to the nodes. Features are linked to the nodes that have associated edges (paths) from which they can possibly be seen. This enables efficient indexing of the relevant features for landmark recognition, depending on where the robot currently is.

An example of a world model is shown in Fig. 5. The nodes are shown as small circles, the edges as lines between the nodes. Door and wall features are symbolised as black bars and lines, respectively. 3D-objects are shown as small squares. The links are not shown in this plot.

## 3.2 Teaching a World Model

The basic world model representation, the graph, is taught by simply leading the robot through the environment using a “virtual joystick”. As this joystick, we use the touch screen of a handheld computer displaying an evidence grid showing the local surroundings of the robot. The evidence grid, shown in Fig. 1, is generated using the robot’s laser scanner and thus delivers a very reliable and easily understandable “radar view”. The user can command the robot by simply clicking with a pointing device in the grid where he/she wants it to go. The advantage of using the virtual joystick versus a conventional joystick is that the commands from the user are converted into commands identical to those that will later be issued to the navigation controller when traversing the taught graph. Thus it is immediately checked if the robot can autonomously navigate to the commanded position. The edges are generated automatically simply by tracking where the user has driven the robot.

The features are learned automatically when the user is driving the robot around, teaching the graph. This is achieved by simply starting the appropriate recognisers and registering the features they deliver. When a feature has been seen  $N$  times, it is entered into the world model.



Figure 1: Evidence grid used as virtual joystick. White denotes occupied space, black freespace, and grey unknown. In the plot, the robot is standing in a room with a doorway approximately 2.5 meters in front of it.

Currently we let the user confirm the door hypotheses in the GUI interactively, since also phantoms can occur due to door-similar structures like windows. The links to the features are automatically generated by the teach controller which keeps track of where the features were seen by the recognisers.

In the future, we are going to add the option that the robot simply follows the user as he or she walks through the new environment directly pointing out relevant objects and places using a speech interface and a pointing device.

The 3D-objects are also taught interactively by the user. Different kinds of objects are taught using various dedicated interfaces. Polyhedral 3D-objects are taught using a teachbox with which the user can request a 3D depth image of some specified part of the environment. The 3D image is segmented into planar surfaces [8] and the result is presented to the user in the teachbox interface. A typical example is shown in Fig. 2. With a pointing device the user can now select the regions belonging to an object, name this, and define possible docking and grasp positions relative to the object. When done, the object is stored in the database and can subsequently be referenced in the world model at the positions where an object of this type exists.

For the recognition and localisation of objects not easily describable with geometric primitives we use the *iterative closest point* (ICP) algorithm [9]. This has the advantage that object models can be generated simply by scanning the object with a laser range finder and cutting out the relevant part of the resulting 3D point cloud. The algorithm iteratively seeks to match model points to scene points (using a nearest neighbour metric) and—based on this match—to register the model and the scene objects. When the iteration has converged we compare



Figure 2: Range image segmented into planar surfaces.

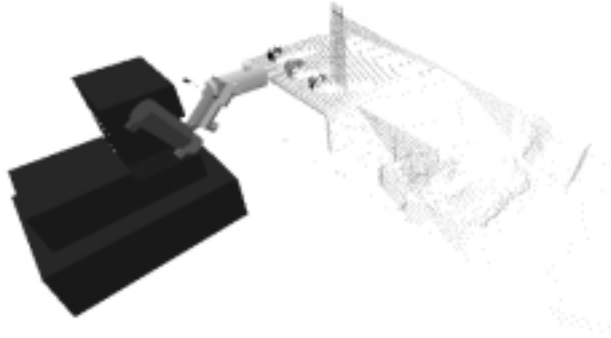


Figure 3: Example of the localisation of an object on a table. The scene consisting of a table with 3 cups on it (see Figure 6 for an image of a similar scene) has been scanned by the robot’s laser range finder resulting in the 3D point cloud in the right half of the image. The ICP algorithm has converged to the cup in the middle (the lighter model points cover the darker scene points). On the basis of this localisation it is calculated how to grasp the cup which is shown in the display as feedback to the user.

the mean square error between scene and model points, and if this is below a given threshold, we assume the scene object to be of the same type as the model object and that these furthermore coincide. Based on this, grasping and docking points are calculated. Due to its simplicity, the algorithm works rather quickly, but due to the fact that it is basically a gradient descent, it has problems with local minima. However, for scenes where foreground and background can easily be segmented, it has been shown to work well. In Figure 3 an example of the localisation of an object on a table is shown.

The interactive teaching method described here has been proven to be adequate for the fast and reliable generation of world models. An example of such a model is presented in Section 4.

### 3.3 Teaching of Arm Movements

For teaching the arm movements necessary to perform manipulation tasks we have, similar to the teaching of world models, pursued two directions; teaching via a

graphical user interface and teaching by directly moving the arm around. For this purpose, the arm, shown in Figure 4, has been equipped with a force-torque sensor between the gripper and the wrist enabling us to directly teach arm movements and grasping operations (skills). Normally, the “crude” movements are taught using the GUI while finer movements are better taught by directly moving the arm to the desired position and orientation.

During the execution of taught movements, the path of the manipulator is checked for obstacles which are detected with the robot’s laser range finder. If a potential collision is detected the system tries to re-plan a trajectory around the obstacle. Taught movements, e.g. for grasping, are generalised using planning in order to compensate for the fact that a given object will normally not have the same position relative to the robot as when the grasping was taught. We are currently investigating how to generalise taught grasping trajectories to different classes of objects.

### 3.4 Teaching of Missions

Currently the missions are planned (using a STRIPS-like planner) on the basis of timelines which are described in a high level specification language. Complex, difficult to plan missions or mission parts such as manipulation tasks can also be described directly in this language. Although this language is simple to use for a robot programmer it does not directly constitute an intuitive programming interface for non-expert users. Therefore we are developing algorithms allowing the user to specify missions by simply performing the desired task with the robot (programming by demonstration) using the “alphabet” of objects and arm movements already known to the robot. The output from this teaching is thus still a symbolic timeline description facilitating interactive confirmation and possibly correction by the human user.

## 4 An Example of a Taught World Model

In this section, we will present an example of a world model which has been taught by leading the real robot, shown in Fig. 4, around in a previously unknown but standard office environment.

In this example, the robot was driven from a starting point in a room out into a hallway which forms a rectangular loop. Back in the room from which it started it was taught two 3D-objects; a table and an automatic coffee maker. The resulting model is shown in Fig. 5. For the shown model, teach-in time was 5 minutes and total path length is approximately 68 meters. The 3D-objects were taught before teaching the world model which took approximately 2 minutes per object.

Using the taught world model it is possible right away and without any modifications to plan and execute rather



Figure 4: The robot “Clever” used in the experiments. The robot consists of a differential drive platform equipped with a 7-DOF Amtec manipulator and a Sick lidar plus a set of colour CCD cameras mounted on a pan/tilt unit. The arm is equipped with a force-torque sensor between the wrist and the gripper, which is a standard parallel jaw gripper with force sensors in the “fingers”.

complex missions. In this case we used a coffee serving scenario where the robot must pick up a cup from the table, bring it to the coffee maker, insert it into the machine, press the button for activating the machine, and pick up the cup and serve it to some human. In Figures 6–9 some scenes from this mission are shown. The necessary arm movements and manipulation capabilities were taught beforehand using the arm teachbox and the compliant motion described in Section 3.

## 5 Discussion and Conclusion

In this paper, we have described research done for a robotic assistant at DaimlerChrysler Research and Technology.

In particular, we presented the representations and methods developed for interactively teaching the robot world model information, a capability we consider fundamental to the practical use of robotic assistants. The basic idea behind the teaching is to let the user guide the robot through the environment as if it was driving autonomously and automatically record information about places, paths, and landmarks. Furthermore, the user must interactively teach relevant 3D-objects and arm movements. This produces compact models that are di-

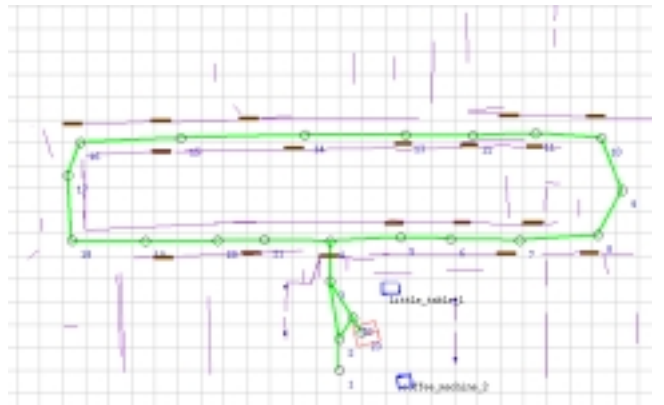


Figure 5: World model taught by leading the robot through a previously unvisited office environment. Grid size is 1 meter. See text for further details.



Figure 6: Scenes from the coffee serving scenario: after docking by the table the robot locates a cup and picks it up.

rectly verified because the robot has just been at the places, has driven the paths, and has seen the landmarks and objects that are stored in the model.

With an example from a standard type of indoor office environment we have shown that the information taught by the user is sufficient to plan and robustly execute missions including navigation, localisation, docking, and manipulation tasks.

It is important to note that due to the longevity of the taught models, the teaching load decreases very quickly to a level where it is normally only necessary to make occasional improvements and extensions to existing models.

Current research aims at improving the man-machine interaction by adding more communication channels and by extending the cognitive capabilities of the robot in order to make it more capable of analysing various tasks and the context in which they are carried out. This will lead to a robot which can carry out missions autonomously as well as in cooperation with humans.



Figure 7: Scenes from the coffee serving scenario: the robot drives to the coffee maker and inserts the cup in the coffee dispenser.



Figure 8: Scenes from the coffee serving scenario: the robot presses the button on the coffee maker.

## References

- [1] Elfes, A.: Sonar-Based Real-World Mapping and Navigation. *IEEE Journal of Robotics and Automation*, Vol. 3(3) (1987) 249–265
- [2] Kuipers, B.J., Buyn, Y-T.: A Robust, Qualitative Method for Robot Spatial Learning. In: *Proceedings of the Seventh National Conference on Artificial Intelligence*. AAAI Press, Menlo Park, Calif. (1988) 774–779
- [3] Kuipers, B.J., Buyn, Y-T.: A Robot Exploration and Mapping Strategy Based on a Semantic Hierarchy of Spatial Representations. *Robotics and Autonomous Systems*, Vol. 8 (1991), 47–63
- [4] Gutmann, J-S., Nebel, B.: Navigation mobiler Roboter mit Laserscans. In: *Autonome Mobile Systeme*. Springer (1997), in German.
- [5] Koenig, S., Simmons, R.G.: Xavier: A Robot Navigation Architecture Based on Partially Ob-



Figure 9: Scenes from the coffee serving scenario: the cup of coffee is handed over to the thirsty scientist. When doing this, the robot monitors the wrist forces and waits until the person has taken hold of the cup.

servable Markov Decision Process Models. In: Kortenkamp, D., Bonasso, R.P., Murphy, R.: *Artificial Intelligence and Mobile Robots*. AAAI Press/The MIT Press (1998) 91–122

- [6] Thrun, S., Bücken, A., Burgard, W., Fox, D., Frölinghaus, T., Hennig, D., Hofmann, T., Krell, M., Schmidt, T.: Map Learning and High-Speed Navigation in RHINO. In: Kortenkamp, D., Bonasso, R.P., Murphy, R.: *Artificial Intelligence and Mobile Robots*. AAAI Press/The MIT Press (1998) 21–52
- [7] Jensfelt, P., Kristensen, S.: Active Global Localisation for a Mobile Robot Using Multiple Hypothesis Tracking. In: *Workshop on Reasoning with Uncertainty in Robot Navigation (Workshop ROB-3 at the International Joint Conference on Artificial Intelligence)*, Stockholm, Sweden (1999) 13–22
- [8] Jiang, X. Bunke, H.: Fast Segmentation of Range Images into Planar Regions by Scan Line Grouping. *Machine Vision and Applications*, Vol. 7(2) (1994) 115–122
- [9] Besl, P.J. McKay, N.D.: A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14(2) (1992) 239–256