



# Johnny: An autonomous service robot for domestic environments

Thomas Breuer · Geovanny R. Giorgana  
Macedo · Ronny Hartanto · Nico  
Hochgeschwender · Dirk Holz · Frederik  
Hegger · Zha Jin · Christian Müller · Jan  
Paulus · Michael Reckhaus · José Antonio  
Álvarez Ruiz · Paul G. Plöger · Gerhard  
K. Kraetzschmar

Received: date / Accepted: date

**Abstract** In this article we describe the architecture, algorithms and real-world benchmarks performed by *Johnny Jackanapes*, an autonomous service robot for domestic environments. *Johnny* serves as a research and development platform to explore, develop and integrate capabilities required for real-world domestic service applications. We present a control architecture which allows to cope with various and changing domestic service robot tasks. A software architecture supporting the rapid integration of functionality into a complete system is as well presented. Further, we describe novel and robust algorithms centered around multi-modal human robot interaction, semantic scene understanding and SLAM. Evaluation of the complete system has been performed during the last years in the RoboCup@Home competition where *Johnnys* outstanding performance led to successful participation. The results and lessons learned of these benchmarks are explained in more detail.

**Keywords** Domestic service robots · Benchmarking · Human robot interaction

## 1 Introduction

During the last decades robotic research moved from stationary robotic systems in constrained environments to mobile and service-oriented robots operating in realistic and unconstrained environments. Based on recent progress in fundamental robotic algorithms as mapping, navigation, and perception mobile robots are almost ready to be deployed as assistants in challenging environments. One up-and-coming application of service robots are as daily-life assistants within domestic environments. Thereby robots could assist and support us in daily-life tasks like

---

Authors are affiliated at  
Bonn-Rhine-Sieg University  
E-mail: forename.surename@inf.h-brs.de

R. Hartanto  
DFKI - Robotics Innovation Center, Bremen, 28359, Germany  
E-mail: ronny.hartanto@dfki.de

cleaning, washing, or ironing [52], by performing such tasks in a reasonable manner. For instance, within an acceptable time frame or without constraining the environment. To do so, a domestic service robot must be equipped with diverse abilities such as: human robot interaction, person detection and tracking, planning, reasoning, object detection, classification, and manipulation. However, all these abilities are active and interdisciplinary research fields itself. The integrative nature of domestic service robot research opens thereby novel research challenges centered around the trade-off between precise and robust abilities. To meet requirements as robustness against environmental changes or the safe interaction with humans the abilities must be carefully selected, improved or even developed from scratch. Recently, research on complete domestic service robots has attracted the community. In [57] Srinivasa et al. presented *HERB*, a home exploring butler with promising object manipulation skills. Another complete service robot is the *PR2* by WillowGarage, a robot equipped with a dual arm system for e.g. opening doors [43]. Explicitly designed for domestic environments, the Care-O-bot 3 robot with the appearance of a friendly butler [53]. The *DESIRE* platform, a dual arm robot, is a research platform for studying abilities required in domestic environments as manipulation and perception [49]. In [5] Beetz et al. presents a robot which is able to perform everyday manipulation tasks incorporating knowledge from various sources.

However, evaluation of these complete systems in realistic and real-world environments is difficult due to the uniqueness of the robots, missing measures and procedures, and the lack of a benchmarking methodology. Though, one accepted and feasible way to perform benchmarking is through scientific competitions. Examples are the DARPA Grand Challenge events, the European Robot Trials and the various leagues under the umbrella of the RoboCup initiative. Along with the various robot soccer competitions, the RoboCup@Home<sup>1</sup> league explicitly targets the benchmarking of autonomous service robots in domestic environments. The competition defines a set of benchmarks or tests inspired by domestic environments and performance metrics centered around key abilities required to perform these tests. To support these benchmark efforts and to explore novel design and algorithm challenges in the field of domestic service robots we developed *Johnny Jackanapes* or just *Johnny*.

*Johnny* is an autonomous service robot for domestic environments participating since several years in the RoboCup@Home league. In this article we present the architecture, algorithms, and real-world benchmarks performed by *Johnny*. First, in Section 1.1, we describe a future application of domestic service robots. Thereby, we derive several capabilities which are required for a robot to be ready to be deployed in domestic environments. Namely, an architecture (Section 2), multi-modal human robot interaction (Section 3), and semantic scene understanding (Section 4). In Section 5 we discuss the benchmarking results obtained through the participation in the RoboCup@Home competition.

---

<sup>1</sup> Detailed information about the league can be found on <http://www.ai.rug.nl/robocupathome/>

### 1.1 Johnny in the restaurant

In the following a scenario where *Johnny* serves guests in a restaurant is described. The scenario is based on the real-world performance of *Johnny* during the finals of RoboCup@Home 2010 in Singapore<sup>2</sup>. The restaurant-like environment is densely composed of dynamic objects as guests walking around and static objects as tables, shelves, and chairs.

*Johnnys* tasks are the following:

- Receive seat reservations
- Welcome known and unknown guests at the entrance
- Escort guests to reserved and free seats
- Receive orders from guests like drinks, candies or chips
- Grasp and deliver the orders to the right guests
- Entertain guests by recognizing their mood and playing an appropriate song
- Find items lost or forgotten by guests in the restaurant

To perform the tasks the following capabilities are required:

**Semantic scene understanding:** For a domestic service robot it is not sufficient to simply perceive the environment. Modern robots are required to interpret raw sensor data in a more elaborated way. More precisely, understanding the semantics of a scene from raw sensor data is required. Thereby semantic scene understanding could range from the classification of laser-scans to places, e.g. kitchen or living room to the categorization of objects placed on shelves or tables. The process of semantic scene understanding is the key factor for achieving more complex tasks as described above. For instance, assuming a guest called *Bob* leaves the restaurant and recognizes, while being already at home, that he forgot his cell phone in the restaurant. *Bob* could call the restaurant and ask *Johnny* to search for the cell phone. In doing so *Johnny* requires several capabilities as the categorization of objects into e.g. cell phones or the background knowledge that cell-phones are often placed on tables or shelves.

**Human robot interaction:** A service robot which is not able to interact with its recipient is meaningless. For instance, in our scenario *Johnny* is required to perceive the orders by the guest. Therefore, means for human robot interaction are required. Thereby several modalities could be used. Ordering a drink through a speech interaction might be more feasible than pointing on a menu. However, a pointing gesture is more intuitive for signalling on which table the guest wants to sit.

**Object manipulation:** One mean for a service robot to interact with the environment is through the manipulation of objects. In the restaurant scenario *Johnny* is required to deliver orders, such as drinks or snacks. The delivery includes grasping of the orders from shelves or tables and the hand-over to the guest.

**Planning:** There are two approaches to perform tasks in the service robotic domain. In the first approach, the robot is equipped with pre-programmed capabilities, e.g. *navigate*, *following-a-person*, etc. With this approach, the robot can perform tasks based on user commands. However, in the long run it would not be able to solve problems in dynamic environments and perform more

---

<sup>2</sup> A complete video of the finals can be found on <http://b-it-bots.de/Media/Media.html>

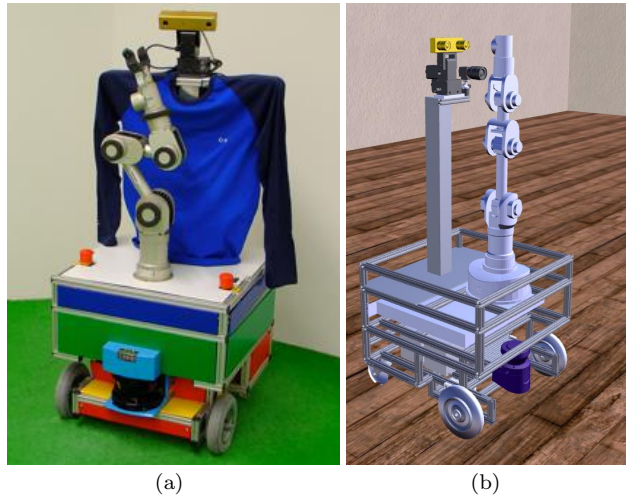
complex tasks. The second approach is using a planning system. The planning system enables the robot to solve complex tasks composed of several pre-programmed capabilities of the robot. The domestic service robot domain can be quite complex domain if one model all the objects in the planning domain. Such a complex domain could make the planning problem intractable. Hence, Johnny uses the Hybrid Deliberative Layer (HDL), where a Description Logic (DL) reasoner is used to store the planning domain and the world model. Details of HDL are presented in Section 2.2. Beside, for supporting the planning system, the DL system is used to store additional knowledge and support the Human Machine Interaction (HMI) system. It supports the HMI system by bridging the gap between the semantic information and the metric information. For example, DL stores the objects in its model. These objects contains some properties, from which some are needed by the planning system and some others by the users. In the case of dialog system, if a user ask Johnny to bring something, the DL reasoner provides Johnny with objects that have the property *graspable*, which excludes objects that are too heavy for its manipulator. If more than one object is available, Johnny can ask the user by providing what options the user has.

**Plan execution:** The output of a planning system is sequence of actions. These actions have a symbolic representation. For example, in the task of grasp and deliver order, it may consist of the following actions: *navigate-to-table*, *move-to-dexterous-workspace*, *grasp-chips*, *drive-backward* and so on. These actions are still in symbolic representation, which are not enough for the low level controller. The controller would not understand where is the table or how chips look like. Therefore, a plan execution system is needed to translate these actions into understandable commands for the low level controller. This translation process is supported by the HDL system, where the plan-execution system could ask additional information to interpret the actions. For example, it can ask the HDL system to provide the *SIFT* feature of the chips or ask the pose of the table it should navigate to. The plan-execution monitors the execution of these actions, so that the goal is achieved. In case of un-repairable failure, it will ask the planning system to produce new plan.

## 2 System architecture

### 2.1 Robot platform

*Johnny* is based on a modular mobile platform called VolksBot [61], which has been designed for rapid prototyping of robot applications in education, research and industry by the Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS). We use a customized variant, see Figure 1, equipped with a Neuronics Katana 6M180 robot arm with five Degrees of Freedom (DoF). The manipulator has a two-fingered gripper, which is equipped with infrared reflectance as well as force sensors. It can handle a maximum payload of 500 g and is mounted in a way to provide good reachability and maneuverability. One of the primary sensors for perceiving the environment is a SICK LMS 200 laser range finder mounted in the robot's center of rotation. It provides accurate range measurements to surrounding objects intersecting the 2D scan plane in an angular range



**Fig. 1** *Johnny Jackanapes* moving around the BRSU campus (a) and a simulated kitchen environment (b).

of 180 degrees. Further, *Johnny* is equipped with a Bumblebee stereo camera as well with a commercial and out of the shelf monocular webcam; both mounted on a pan tilt unit. The drive unit used for locomotion uses a differential drive with two actively driven wheels, powered by two 150W motors, and two caster wheels to enhance rotating and stability under load. The robot’s maximum velocity is 2 meter per second.

## 2.2 Robot control architecture

The overall control approach is based on a deliberative layer which is needed in a complex domain, such as domestic robotics. It provides the robots with additional cognitive capabilities to solve complex tasks. An example of such complex task in domestic robotics is pick and place (e.g. ”bring a coke to the guest in the armchair”). To perform this task, the robot needs to combine several actions such as navigation, object recognition, and object grasping. As the number number of capabilities of a robot grows, more complex combinations of tasks can be performed.

In our robot, we use a novel approach which is called *Hybrid Deliberative Layer* (HDL) [23]. It extends the planning component in hybrid control architecture that is usually used in mobile robotics with a *Description Logic* (DL) [3] reasoning system. Figure 2 shows the HDL as our robot control architecture. As planning component, HDL uses JSHOP2 [46], a *Hierarchical Task Network* (HTN) planner [21]. The DL reasoning component is implemented using *Pellet* [56].

The planning-related information and robot-specific information are stored in the ontology model instead of being merely planning problem descriptions. Two major benefits can be gained from this approach. Firstly, one can store huge number of objects or rooms in the ontology model without affecting the size of the planning-problem descriptions. Only planning-related objects or rooms are in-

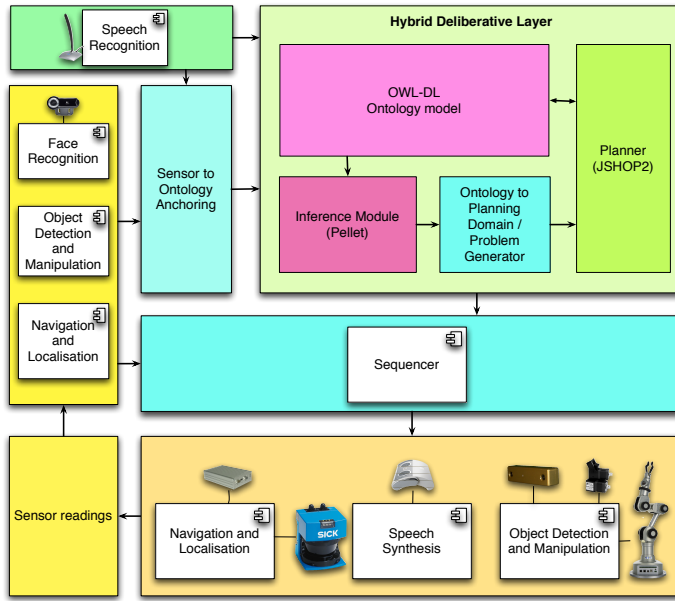


Fig. 2 HDL system architecture [23]

cluded in the problem descriptions. Secondly, the DL provides the capability to model domestic environments more naturally. Thus it can be used with other components such as a speech recognition engine, so that it can map the human objects with the planning system symbolically. The numerical information of the objects is modeled as property of the instances of the ontology model.

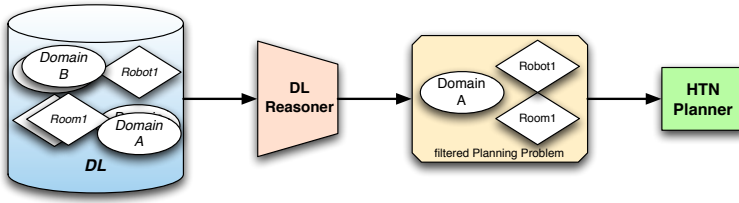
### 2.2.1 Concept

The reasoning process over the DL model for extracting a planning problem description is shown in Figure 3. All necessary information for the planning system and the robot is stored in the DL model. In addition to the robot's world model, it can store the planning domains as well. The planning domains are not limited to one domain only. Additional domains will not influence the overall planning performance, as only relevant instances and domains are considered for the problem description. The DL reasoner filters the DL model and generates a valid problem description for the JSHOP2 planner.

In the *Description Logic* representation, the objects and planning related information are modeled and stored in two boxes. The first box is the *Terminological Box (TBox)*, which stores the information as set of concepts. The second box is the *Assertional Box (ABox)*, which stores the instances of the conceptual information on the *TBox* [3].

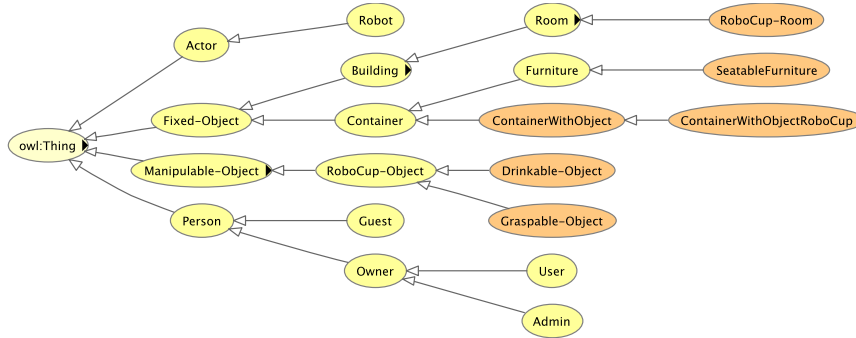
### 2.2.2 Modeling RoboCup@Home domain

Let us take a concrete example in RoboCup@Home domain, namely the pick and place task. Figure 4 shows the *TBox* model in this domain. As shown in this



**Fig. 3** A reasoning process over DL representation to extract a concrete HTN planning problem [23].

figure, the world model is captured and represented as set of concepts, which are denoted as ellipses. The robot itself is part of this model and represented as  $Robot \sqsubseteq Actor$ . Not of less significance than the robot, the objects including the furnitures are represented as well in the DL. Not every object has to be modeled in the DL. However, the planning related objects have to be captured and modeled in this model. As mentioned before, additional conceptual models will not affect the planning performance as the DL reasoning engine will filter irrelevant concepts.



**Fig. 4** An example of RoboCup@Home Ontology [23].

In the Figure 4, some concepts are shown in orange colored ellipses. These concepts do not have direct instances, which asserted by the system or user. The DL reasoner will reason about the model and fill these concepts with instances that fulfill the rule defined on the concept. For example the *Graspable-Object* is defined as follows:

$$\begin{aligned}
 \textit{Graspable-Object} &\equiv \textit{RoboCup-Object} \sqcap \\
 &\supseteq \textit{hasProperty}(\textit{small}) \sqcap \\
 &\supseteq \textit{hasProperty}(\textit{lessThan500g})
 \end{aligned}$$

Table 1 shows the number of instances on the RoboCup@Home concepts. The number of asserted instances represents the amount of objects which are explicitly asserted into the system. The DL reasoner reasons about the model and produces inferred instances as a result. As shown in the table, although some concepts



**Table 1** RoboCup@Home concepts and their instances [23].

Concept	# of Asserted Inst.	# of Inferred Inst.
<i>Fixed-Object</i>	0	62
<i>Building</i>	3	40
<i>Room</i>	16	19
<i>RoboCup-Room</i>	0	3
<i>Container</i>	5	22
<i>ContainerWithObject</i>	0	6
<i>ContainerWithObjectRoboCup</i>	0	4
<i>Furniture</i>	17	17
<i>SeatableFurniture</i>	0	4
<i>Manipulable-Object</i>	0	22
<i>RoboCup-Object</i>	15	15
<i>Drinkable-Object</i>	0	3
<i>Graspable-Object</i>	0	7

have no explicit instances they will have some instances that met their definitions. Therefore, we can easily extend some concepts by refining their definitions in order to reduce the amount of instances and remove irrelevant instances from the planning problem. As shown in the table, the number of *Manipulable-Object* is 22 but only 7 are *Graspable-Object* and 3 are *Drinkable-Object*.

### 2.3 Software architecture

The software architecture – realizing the control architecture – of our robot is inspired by the component-oriented paradigm [58]. Here, components encapsulate a functionality and expose it through well-defined interfaces. The resulting building blocks are decoupled from each other and therefore easier to reuse and to compose. In general the software architecture may be described best as a loosely integrated aggregation of dedicated autonomous components (ACos). This is, on one hand, in contrast to the classical three layer architecture (3T) consisting of controllers (skill level), a sequencer (execution level) and a planner (deliberation level). On the other, it resembles some principles of 3T, see [20]. For example employ one combined navigation, localization, drive unit (NLD) which works self sufficiently while executing tasks like path planning and following or tracking of a human operator. Similarly we have a combined manipulation/object recognition component and HRI components. These components do not match to any single layer in the 3T architectural pattern since they themselves already comprise several of these levels. The NLD, for example, offers services for low level motor control, guidance or path following yet also contains path and motion planners as well as own deliberation and sequencing components. Furthermore, the NLD maintains different environment representations and contains several components for simultaneous localization and mapping (SLAM). It can work self sufficiently on the achievement of goals (like: move to the refrigerator) and may also decide all by itself when to stop. The same holds true for our combined manipulation/object recognition/pan-tilt-camera component which takes care of searching, fixating,

identifying and grabbing a known object. An ACo is defined as a unit of composition, containing two (possibly all) of the aforementioned classical three levels and working self-sufficiently on goal achievement of its respective task. In our architecture, the integration of a communication between ACos is realized via the ICE middleware [24]. The middleware allows to compose heterogeneous components (e.g. components running under Windows, Linux and Mac OS X and developed in various programming languages). Table 2 exemplifies the heterogeneity in our robot. In general, integration takes place only on the level of ACos and a classical scheduler sequences all operations which are derived from the planning process described in section 2.2.

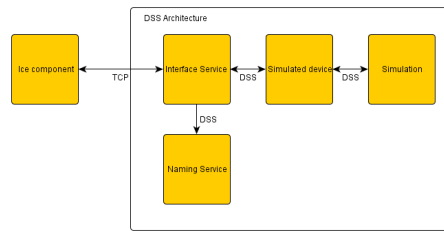
**Table 2** Overview of the capabilities of *Johnny Jackanapes* and the respective autonomous components realized in a programming language (PL) and running under a certain operating system (OS).

Capability	Autonomous Components	PL	OS
<b>Human robot interaction</b>	Face recognition	C++	Windows
	Facial expression recognition	C++	Windows
	People detection and tracking (laser-based)	C++	Linux
	People detection (sound based)	Matlab/C++	Windows
	Gesture recognition	C++	Windows
	Speech recognition	C++	Windows
	Speech synthesis	C++	Mac OS X
<b>Semantic scene understanding</b>	Object categorization	C++	Windows
	Text mining and understanding	Python/C++	Linux
	Simultaneous localization and mapping (SLAM)	C++	Linux
<b>Manipulation</b>	Vision-based manipulation	C++	Windows
<b>Planning</b>	DL-based hierarchical task networks	Java	Mac OS X
<b>Plan execution</b>	State machine based task execution and monitoring	C++	Windows
<b>System integration</b>	ICE-based integration framework	PL independent	OS independent
	Simulation and emulation framework	.NET	Windows

## 2.4 Service robot simulation and architecture integration

Simulation is a valuable tool for testing robot components or hardware designs without the availability of the real robot. Simulation can therefore fulfill several purposes while we are focusing on integration tests and component tests. Integration tests are used to test the whole software system on the robot integrating all hardware and software components like manipulation, navigation, HMI or planning. Component tests can be performed in simulation where single components can be tested in a virtual scenario. The simulator provides therefore efficient usage of resources since multiple developers can work on the same hardware simultaneously although only a single robot is available.

We use a kinematics and dynamics simulation of the robot in Microsoft Robotics Developer Studio (MRDS) [32]. The simulator is tightly coupled in the MRDS architecture which is based on the Decentralized Software Services (DSS), a custom Webservice Oriented Architecture developed with the .NET platform. The runtime environment called DSS Node offers a set of system services providing basic



**Fig. 5** Structural overview of the ICE to MRDS bridging of a simulated service

middleware functionalities e.g. Naming Services or Security Manager. Communication between DSS Services is defined by the Distributed System Services Protocol (DSSP) that is similar compatible to HTTP and adds further functionalities, like state manipulation and event notifications.

As mentioned afore, the simulation environment is integrated in the MRDS as an own service and can be interfaced in exact the same manner as every DSS service. The simulation service contains a virtual world. Every simulated element that shall be interfaced, e.g. sensors or robot actuators, has its own DSS service that acquires and preprocesses the data from the virtual world. Those services are used to publish the simulated data to other services.

For efficient use, the switching between real robot and simulation must be possible without changing the robot software. On the other hand the simulation replaces some essential part of the robot software, namely the perception and acting elements by a virtual counterpart. To perform realistic experiments in the simulation, this process must be transparent. In particular, the robot software framework shall not know whether it works with a simulated model or a real robot. In order to fulfill this requirement, a software bridge has been developed that allows communication between the ICE based framework and the Microsoft DSS framework.

In the presented software framework, all software components have an exclusive access to their used hardware. In a simulated scenario, those software components are replaced by an exact copy which does not directly access the hardware but uses a simulated representation instead. This exchange is transparent to the rest of the framework since both components, simulated and real one, supply the same ICE interface description file (slice). The simulated component uses a proprietary TCP communication to access a MRDS Service. This communication can be used to query simulated data in a synchronous communication or to set simulated actuators in a synchronous or asynchronous way. For the MRDS an interface service was developed that decomposes the incoming messages from the ICE framework and converts them in a DSS message. Further, it requests the DSS naming service to identify the target simulated sensor or actuator service via its string based unique identifier that is part of the message. Finally the message is forwarded to the identified target service. The reply message is handled in a similar way. The invoked MRDS simulation service will send the reply to the interface service which will convert the message in the proprietary TCP format and redirect it to the target ICE component where it is processed. A structural overview of this communication is given in Figure 5.

The performance of the system is analyzed by a set of round-trip messages for getting or setting simulator data with a payload of 10 bytes. The test was executed with simulation and robot software running on one computer and both running on different machines.

**Table 3** Performance of round-trip message with two different systems involved. System A is a 1.4Ghz Pentium M, system B an Intel quad core with 2.6Ghz each core

Architecture $\Rightarrow$ Simulator	Message	$\frac{\text{messages}}{\text{second}}$	$\frac{\text{seconds}}{\text{message}}$
A $\Rightarrow$ B	GET	227	$4.41 * 10^{-3}$
	SET	355	$2.82 * 10^{-3}$
B $\Rightarrow$ B	GET	1074	$0.93 * 10^{-3}$
	SET	1627	$0.62 * 10^{-3}$

The results of the performance evaluation is given in Table 3. It shows that, if a fast machine runs both architectures, the messaging time is in the  $\mu s$  range. Network communication adds some delay but the round trip time of the messages is still below 5ms.

At the moment we use the simulation to test the integration of different components. However, for the complete testing of all software we need to model the environment in which the robot acts. A very important aspect in this regard is the modeling of humans to integrate HRI.

### 3 Multi-modal human robot interaction

Human robot interaction (*HRI*) is a multidisciplinary field aiming to find ever faster and more intuitive manners of communication between humans and robots. Mostly humans express their intentions via speech, gestures, expressions and sounds. Domestic service robots (DSR) must be aware of those intentions and also be able to understand them. In this section we present our four HRI modules endowed into our DSR Johnny. For some scenarios the different modules have been combined to increase the robustness of our robot.

#### 3.1 Laser-based people detection and tracking

People detection and tracking is one crucial part of human-robot-interaction. HRI techniques like gesture- or facial expression recognition operate robustly only up to a certain distance between the robot and the human, e.g. in a range of 1 *m*. The presented people detection and tracking approach uses two sources of information - one Laser Range Finder (LRF) in leg height and one in waist height. Current LRFs provide ranges of up to 30 meters and allow a robot to sense people at farther distances. The detection mechanism is divided into three stages: preprocessing of the raw laser scans (1), detection of legs and waists in the respective layer (2) and fusion of both sensor information (3).

In the preprocessing stage (1), we have applied a *Point-Distance-based Method* (PDBS) [51] in order to cluster the raw laser scan into smaller segments. A laser

scan is processed as follows: let  $\mathbf{L} = \{p_i | i = 1 : N\}$  be a laser scan containing a sequence of  $N$  polar coordinates  $p_i = (r, \alpha)$ , then a new segment is established when  $Distance(r_i, r_{i+1}) > Threshold_{JumpDistance}$ . Otherwise  $p_i$  is added to the current segment. The applied threshold is also named as the *Jump Distance Criteria* (JDP). The segmentation results in an ordered sequence of segments where each segment can have various appearances. A segment which represents a wall includes usually many points and appears as a straight line. On the other side a garbage bin involves fewer points and appears as a circular object. Such kind of *geometrical properties* have been proposed by Arras et. al [1]. This set has been adopted and extended by an additional property - namely the distance to the respective segment. Since the appearance of an object is strongly dependent on how far it is away from the LRF, these additional features have been added to the feature set.

The actual detection (2) in both heights is realized by a supervised machine learning approach, namely *AdaBoost* [19]. The geometrical properties of a segment build the *feature space* for AdaBoost. Positive and negative training samples have been collected from different environments (e.g. office, corridor and apartment). Each layer is trained separately during the training phase. In the detection-/classification phase, the generic AdaBoost model is applied to the respective layer and each segment is labeled whether it belongs to a person or not. During each detection cycle, a list of possible leg and waist positions is generated.

The information of the leg- and the waist-layer are fused together in order to increase the detection accuracy and detect multiple people reliably in a clutter environment. The major idea of this layered architecture is the verification of each detection in one layer by possible detections in the other layer. A *Priority Shape Model* (PSM) has been applied which considers the spatial relations between a waist and two related legs (adopted from [45]). During our evaluation it turned out that the trained model of the waist-layer has a reasonable higher detection rate than the leg-layer. The performance difference comes from the simple fact, that there are many leg-like objects (chair-legs, table-legs and even other small objects) in low height which have been collected and are labeled as negative samples. Therefore, it occurred that both sample sets, positive and negative, include similar samples and the machine learner is not able to separate them in an optimal way. In the waist-layer, the positive samples consists of larger segments and because there are not so much similar object in the same height. Hence, the learned model in the waist-layer performs more accurate than the model of the leg-layer. According to the this observations, a *scoring system* has been established which assigns a higher confidence to waist detections.

Furthermore, a single person tracker has been implemented based on a *particle filter* (adopted from visual tracking [25]). The tracking is divided into two stages:

1. The tracker applies the static person detector which is looking for a person to track. The actual tracking is initialized when a person is detected in a specific range of the robot. This detection build the prior distribution  $\mathbf{P}(x_0)$  and all particles are initialized with random noise  $w_t$ .
2. The particle filter tracks the related laser scan segment of the initially detected person. The transition model  $\mathbf{P}(x_t | x_{t-1})$  is established on a second-order autoregressive model. This model does not consider only the last state  $(x_{t-1}, y_{t-1})$ , but also the second last state  $(x_{t-2}, y_{t-2})$  to predict the new state of a parti-

cle (for details we refer to [25]). Afterwards all particles are weighted by their likelihood according to the observation model  $\mathbf{P}(z_t|x_t)$ . Finally, all particles are resampled based on Sequential Importance Resampling (SIR). We only re-sample whenever the effective samples size  $N_{eff}$  falls below a certain threshold  $N_{th}$  [2].

The experimental results have shown a significant influence of the additional distance feature. We have achieved in average a 4.4% fewer misclassification rate than without the distance feature. This can be explained by the circumstance that the appearance of a leg or a waist in a laser scan is strongly dependent on how far it is away from the source of measurement. A table leg at close distance might look like a human leg at far distance and they might also share the same geometrical properties. In this case only the distance makes a distinction possible. The proposed shape model has mainly increased the overall performance regarding the false positive detection (see Table 4). Especially in the leg layer many false positive detections occurred ( $\approx 5.4\% - 13.2\%$ ). Through the application of the shape model we achieved a false positive rate of only  $\approx 1.3\% - 4.4\%$ .

**Table 4** The robot was navigated through different environments at 0.2 m/s. Simultaneously, the number of false positive detections were collected. The table shows the misclassification rate for each single layer (leg resp. waist) and for the fusion by the priority shape model.

	Location			
	Apartment	Lab	Corridor	Office
<b>Legs</b>	11.96%	13.20%	5.39%	12.26%
<b>Waists</b>	4.22%	3.23%	1.74%	6.91%
<b>PSM</b>	3.11%	2.76%	1.29%	4.39%

Further, we evaluated the performance of the shape model regarding the true positive detections. In one test, a person had to be detected at different distances and angles relatively to the robot. Table 5 shows the detection rates for the PSM model. If the model was not able to detect a person, there had been still detections in one of the single layers. The results of the proposed shape model have shown a significantly decreased false positive rate, while also providing a consistent true detection rate of  $\approx 92\%$ .

**Table 5** A person had been stand at different distances and orientations to the robot. For each position 50 laser scans were taken while the detection through the PSM was performed. If there was no detection by the PSM, we also checked the detections in the single layers.

	Distance to Person		
	1m	2m	3m
<b>Only Leg(s)</b>	1.23%	1.17%	1.71%
<b>Only Waist</b>	2.47%	1.59%	2.22%
<b>PSM</b>	93.40%	92.24%	93.07%

Although the results are quite promising, there are still some limitations. Due to the fixed mounted LRF, the detection only works for people which have a

certain height. People with less height - like children - can not be detected in the upper LRF and would not match to the shape model. Furthermore people which are sitting on a chair or a couch might not be detected as well since the legs are farther apart from the waist than in the defined shape model.

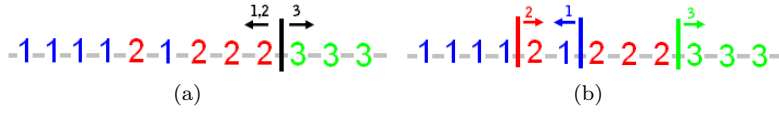
The described people detection mechanism build the primary component to find persons in the robots surrounding. The provided position information are used to approach a person and apply then e.g. facial expression recognition which requires a certain distance to the robot.

### 3.2 Facial expression recognition

Facial expression recognition (FER) offers domestic service robots (DSR) a natural way to interact with humans. This channel of information can be used by robots in order to receive feedback on their actions and also to adapt better to people surrounding them. However, there are many difficulties that have to be tackled before a domestic robot can completely exploit such mode of interaction. In home environments, faces might be completely unknown to the robot, they could also appear poorly illuminated and the inevitable mobility of both humans and robots can make faces look blurred, scaled, rotated or drastically occluded.

Our DSR has been endowed with the capabilities to recognize up to 7 different facial expressions in still images: *joy*, *surprise*, *sadness*, *fear*, *anger*, *disgust* and *neutrality*. The followed approach has as cornerstone the use of Gabor filters of different frequencies and orientations to extract the shape and texture information representative of each facial expression. Gabor filters are characterized by modeling some visual cortical cells [39] and also by providing a spatially localized frequency analysis of the signals (e.g., local line and edge detection). Among the most relevant works studying the performance of Gabor filters for FER we find [7], where authors use local Gabor filter banks together with PCA plus LDA for dimensionality reduction. Furthermore, [38] presented a local approach where Gabor features are extracted at the location of eighteen facial fiducial points. The most extensive study is presented in [4], where a comparison of different image sizes, feature selectors, classifiers and methods to extend them for the multi-class problem is presented. The result of their work is a high-accuracy, real-time, automatic, person-independent system. In the following we describe the architecture of our system and present the experimental results obtained after testing against images of the Cohn-Kanade AU-Coded Facial Expression Database [36] (hereinafter referred too as the C-K Database).

Training is carried out in 4 different stages and ends up with a model indicating what Gabor features are the most relevant to discriminate between the different classes (see Figure 7(a)). First the eye locations in all training samples are manually marked and used to normalize the face images. By normalization we actually aim to obtain face images of equal size, where both eyes are always fixed at the same position. The former is achieved by cropping the face region with a geometric face model based on [55] and scaling the result image to 48 x 48 pixels. After scaling,  $d$  Gabor features are extracted from each normalized image to create an  $n \times d$  observation matrix  $\mathcal{X} = (x_1, x_2, \dots, x_n)^T$ , where  $n$  is the number of training samples and  $x_i$  ( $i = 1, 2, \dots, n$ ) is a  $d$ -dimensional feature vector describing a training sample. In this work a bank of 40 Gabor

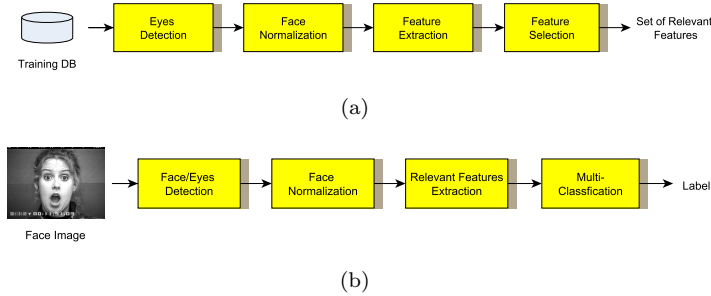


**Fig. 6** Examples of a) the single-threshold and b) the multi-threshold decision stumps.

filters of radial frequencies  $v = \frac{1}{2}, \frac{1}{2\sqrt{2}}, \frac{1}{4}, \frac{1}{4\sqrt{2}}, \frac{1}{8}$  cycles/pixel and orientations  $\phi = 0^\circ, 22.5^\circ, 45^\circ, 67.5^\circ, 90^\circ, 112.5^\circ, 135^\circ, 157.5^\circ$  has been used. However, as the result of convolving each image with the 40 Gabor filters is a feature vector of size  $d = 92, 160$  ( $= 48 \times 48 \times 40$ ), selection of the most discriminative Gabor features was necessary.

For feature selection, as well as for multi-classification, we have employed the AdaBoost.MH algorithm [54]. This algorithm is a multi-class, multi-label version of the original two-class AdaBoost algorithm proposed by Schapire and Freund [18]. Furthermore, we have analyzed two different kinds of decision stumps as weak learners: single-threshold and multi-threshold decision stumps<sup>3</sup>. The single-threshold version aims to find at each iteration the threshold that minimizes the overall weighted error. On the other hand, the multi-threshold approach finds at each iteration one threshold per class, particularly, the threshold that better separates one class from the others [9]. Figure 6 exemplifies the two previously described decision stumps.

Every time an image is input to the system for recognition, first the presence of a face is asserted. If a face is detected, then the eyes are located<sup>4</sup> in order to carry out normalization. Afterwards only the relevant Gabor features are extracted from the normalized image and then used for multi-classification. Figure 7(b) illustrates the system architecture for recognition.



**Fig. 7** System Architecture a) for training and b) for recognition.

In order to find out which of the two weak learners performs better for our task, we have measured their discriminative power on images of the C-K Database.

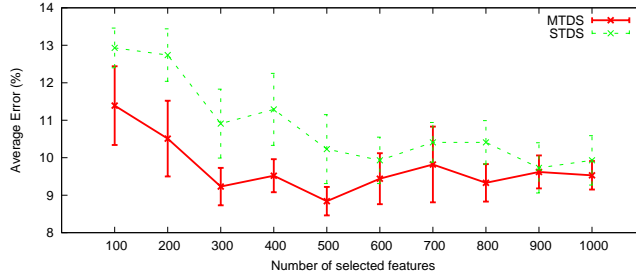
<sup>3</sup> The MultiBoost library of Norman Casagrande was used as implementation of the AdaBoost.MH algorithm and the two analyzed weak learners [10].

<sup>4</sup> The front-end module of the software development kit (SDK) for face recognition of the company L-1 Identity Solutions, Inc. has been used for face and eyes detection.



This standard database contains video sequences starting with a neutral face and proceeding until one of the 6 prototypical facial expressions is clearly visible. All video sequences used in this work come from 96 subjects, each playing at least one video sequence. The experiment was repeated five times for sake of precision, each time using different, randomly created training and testing sets. The image sets were always made up of the last image of each video sequence plus the first image of all video sequences of joy, this latter to collect neutral faces. Additionally, we have taken care that images of subjects in the training set do not appear in the testing set and vice versa. In each repetition of the experiment, images coming from 60% of the subjects were used for training, whereas the remaining were used for testing.

The experimental results are illustrated in Fig. 8. The figure shows the average error rate obtained by single-threshold decision stumps (green, dashed line), as well as by multi-threshold decision stumps (red, solid line). Additionally, error bars representing one standard error above and below the mean value were added in order to display the overall distribution of the data. In the graph we can see that the average error rate obtained by multi-threshold decision stumps is always smaller than the one obtained by single-threshold decision stumps. This former fact reveals the supremacy of multi-threshold against single-threshold. Nevertheless, the difference in accuracy between both weak learners decreases as the number of selected features increases. For both kinds of learners the error rate tends to decrease as more features are used, but after a certain number of features are included the error rate settles down with small oscillations. The previous described behavior is usual in learning approaches like AdaBoost and are a common signal of overfitting. For single-threshold decision stumps the minimum average error rate is obtained using 900 features (9.73%), whereas for multi-threshold decision stumps the minimum average error rate is 8.84%, obtained using only 500 features.



**Fig. 8** Average error obtained by single-threshold learners (dashed green) and multi-threshold learners (solid red) on 5 different training and test sets made up with images of the C-K Database. Error bars represent one standard error above and below the mean value.

### 3.3 Finding people by acoustic clues

For humans speaking is a very easy way to communicate. So the spoken word is probably also the most natural ways to interact with a service robot. For speech

recognition we employ a very mature speech SDK from Microsoft and some hand-crafted grammar of keywords to understand the commands of the user. This component has been analyzed in detail by Thomas Breuers' R&D1 [?] and is not the issue here. There are more ways to interact using the voice. In this section we propose an extended use of acoustic clues. Take for example an user, who calls for the robot from some distance and ask for assistance either in an apartment or an restaurant. Then it would be natural to turn the attention to the speaker and re-assure for the understood command. In sequel we will describe an algorithm which can approximately spot a somewhat distant speaker by exploiting only acoustical information.

The robot spots the position of the speaker in an polar coordinate system using angle and direction so the difference angle of central axis and this direction can immediately be used as a control variable to turn the robots towards the user to signal back that she successfully grasped the attention of the robot. This seemingly simple problem is substantially impeded during a RoboCup@Home tournament by very bad signal to noise ratios, typically as bad as 6dB where the environment noise may go up to 60dB. Thus we combined the underlying search algorithm with a noise reduction step. The proposed method may accordingly be called a grid based steered response power(GBNR) with noise reduction. Hardware-wise we use a very small microphone array of only four microphones in a slightly unsymmetrical configuration.

### 3.3.1 Noise reduction

For the noise reduction we use the spectrum subtraction method [?] (SSM). The main idea of SSM is to estimate a noise spectrum and then subtract it from the observed current signal spectrum. SSM first segments the signal in the time domain using a standard Hamming window followed by a Fast Fourier Transform (FFT) to calculate the spectrum. After this segmentation/transformation step, during the first ten frames, we assume that no speaker is present. The mean power  $\overline{P_{noise}}$  of these initial frames is calculated and this first estimation help to distinguish pure noise frames from speech frames. We then implement a voice active detector as thresholding filter based on the ratio  $\log \frac{\overline{P_{noise}}}{P_{speech}}$  where  $P_{speech}$  denotes the power of the current speech frame. Those fames which fail to pass the filter will dynamically adapt the initial mean power estimation  $\overline{P_{pure_{noise}}}$  with respect to the current conditions and furthermore we use them to build an averaged noise spectrum. This average is subtracted from the spectrum of all following speech frames. The frame with the best SNR is passed over to the Localization step described below. Finally a noise reduced version of the original signal is reconstructed via overlapping and adding the back transformed frames. The choice of frame with the best SNR can be done on the basis of one microphone only, here we do not take advantage of the multiple channels of our sound recording devices. This is different in the Localization step.

### 3.3.2 Localization

The sound localization relies on a low-noise speaker frame when trying to find a global maximum for the received power depending on the location of the sound

**Table 6** Microphone setup where  $Mx$  is the Microphone  $x$  and the values are measured from the robot's center

Microphone	x in meter	y in meter	z in meter
M1	0.4	0.4	0.5
M2	-0.28	0.28	0.5
M3	-0.4	-0.4	0.5
M4	0.28	-0.28	0.5

source. The underlying idea is called Steered Response Power (SRP) [?] and it based on a linear combiner having multiple differently delayed inputs just like in a beam-forming filter. The inputs are given by multiple audio streams from the different microphones. Fixing the weights of the linear combiner to the inverse of the magnitude of the frequency components gives a good estimate of the power spectrum of beam former output signal. This depends only on the position of the sound source. and is called  $PHAT_\beta$  filter. The beam former is used in a steered, inverse way: assuming a known fixed sound source position SRP estimates what will be the power of the output signal of the beam former. Current results show that if a signal sound source is in a predefined region the maximum SRP value will be located in the same 3D region. The sound source localization based beam-forming is then done in two steps. The first step calculates the SRP value in a predefined region taking the geometry information of the microphone array into consideration and then sampling SRP values at randomly chosen source locations inside a predefined region. In our implementation we have chosen a cylindric shaped region to sample the possible speaker locations. From the center a circle is drawn where every degree 200 points are sampled along the outgoing rays. This procedure is repeated for 100 predefined heights. When the SRP values are calculated for the sampled points, the surface produced by the  $PHAT_\beta$  filter is interpolated using cubic splines to achieve surface smoothing for the grid based SRP values. The smoothing of the SRP value surface eases the global maximum search very much. The search phase uses stochastic region contraction method to finally find the global maximum SRP value.

The approach has been validated using a reproducible speech source to guarantee the same input for the experiments yet allowing changeable volume. The following extracts section *evaluate the 2D result from the 3D result, because our purpose of the sound localize is to find out the 2D position of the user*. For the experimental evaluation four microphones were set up in an eye shape around the center of the robot with the distances from the center given in table 6.

We compared the conventional beam-form approach and our new GBNR-SRC approach as shown in Table 6. The evaluation was made by comparing the error on orientation and in the distance between the estimated source position and the true source position. The background noise level is 45dB. The SNR was adjusted by changing the signal volume.

This result shows the comparison between a conventional beam-form approach and our new proposed GBNR-SRC algorithm under different kinds of SNR condition. We found that the precision in orientation has been improved by the GBNR algorithm by 2 degree on average. The proposed algorithm has significantly improvement on the distance error by around 45%. which can efficiently prevent false navigation of the robot.

**Table 7** Comparison between conventional SRP method and GBNR method.

SNR	Angle (degree)		Distance (meter)	
	SRP	GBNR	SRP	GBNR
10.88	6.15	5.25	0.313	0.18
6.21	8.82	7.44	0.8511	0.47
5.522	9.13	8.87	0.957	0.52
3.79	13.7	11.1	1.3	0.78

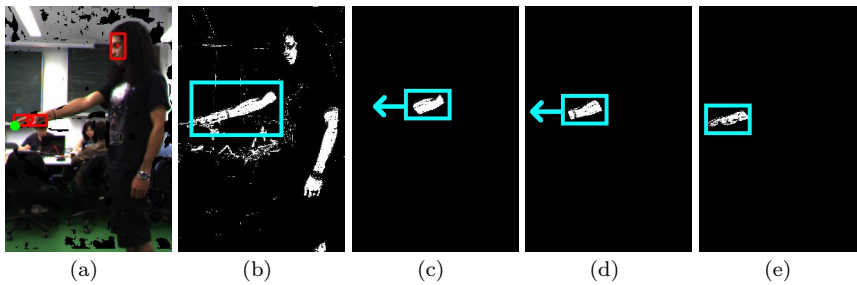
### 3.4 Gesture recognition

Especially pointing gestures are a promising and natural way for the interaction with a robot. As application for a service robot, pointing gestures could indicate objects and locations. It is easier and more accurate to point at an object than to verbally describe the object itself or its location [35]. However, pointing gestures are difficult to recognize [17]. The difficulty is to detect the precise 3D positions of the face and the fast moving hands of an unknown user in front of a dynamic background under unknown and changing lighting conditions. Further it is difficult to detect the point in time, when a user is performing a pointing action.

The developed person-independent dynamic pointing gesture recognition application works markerless and in real-time. It is able to cope with different skin colors without manual retraining, cope with variable and complex backgrounds (including skin colored areas like wood, paperboard, leather), and works under dynamically changing lighting conditions. The application further works if the user is wearing a t-shirt. It is also able to detect if the tracked face or hand is lost and can be reinitialized automatically.

To track the face and hand, first the frontal face is either detected by using OpenCV or a commercial software from L-1 Identity Solutions. The image is converted in the HSV color space and a skin color histogram (based on the hue values) of the face region is extracted. Based on the extracted histogram a skin color probability image (backprojection image) is created, smoothed and binarized. For the initialization of the hand tracking the hand has to be in front of the users chest. For both the face and the hand a 2D trackbox in the backprojection image and a 3D trackbox in the depth data of the used stereo vision camera is defined around the last known face / hand position. After deleting all pixels outside the trackboxes, the face / hand is tracked in the 2D backprojection image via the CAMSHIFT algorithm. The skin color histogram is continuously updated using the tracked face region to be able to cope with variable changing lighting conditions. To be able to track the hand even if the user is wearing a t-shirt the following steps are performed 3 times in a row for each frame (figure ?? shows the process of the algorithm until the hand is found in figure ??):

- the from the head farthest away pixel in the trackbox is used as new center for the trackbox
- The 3D position of the hand (or maybe the arm if user wears a t-shirt) is detected as usual (without moving the track-box towards the from the head farthest away skin colored hand pixel).
- In the found hand region it is searched for the from the head farthest away skin colored pixel. This pixel should be nearer to the hand (if hand is not already



**Fig. 9** Movement of trackbox towards the hand region to be able to track the hand even if user is wearing a t-shirt

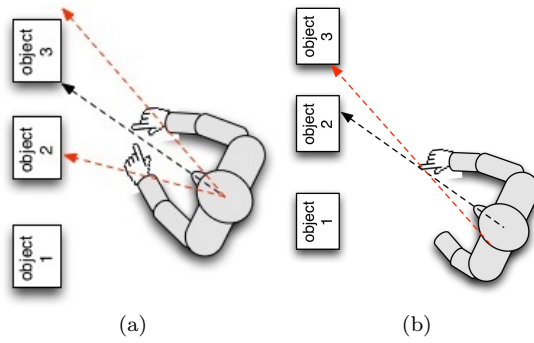
detected). All pixels farther away than 15 cm from this pixel are deleted (in a copied image). If some area of the arm is detected instead of the hand, this deletion will result in keeping skin colored pixels nearer to the hand.

- In the resulting image it is searched again for hand region pixels. The new found hand region is now nearer to the hand than before (if hand was not already detected).

The pointing target lies on the line of sight from the eyes to the fingertip [63] [37] [47]. The related works describe only how to detect / track the hand (instead of the fingertip) and the face (e.g. the chin instead of the center of the face). Figure ?? visualises how the detection of the fingertip and the center of the face improves the pointing target recognition. Compared to the related work, the pointing gesture recognition rate is therefore improved by a fingertip detection algorithm (instead of using the detected center of the hand) and by the detection of the width of the face and adding it to the measured depth value of the face (instead of using the detected depth of e.g. the chin). As fingertip the from the head farthest away pixel in the tracked hand region is taken. This assumption holds true in 92.5% of all performed pointing gestures in the evaluation. The determination of the width of the head is based on the (during the initial face detection) detected eye positions. The according face model is described in [7].

For the dynamic detection of the pointing gesture, the geometric movement of performed pointing actions of different subjects is evaluated. Based on the evaluation, rules could be defined which can classify a dynamic pointing gesture, rather than using time intensive machine learning algorithms. Therefore the application can easily be reimplemented. All described algorithms are further described in [8].

The implementation is integrated on the mobile manipulator and can be used in the context of the RoboCup@Home competition to dynamically detect pointing targets (the start of the pointing gesture is automatically detected). A function is provided to add the pointing targets for the detection. Therefore the pointing targets could be set manually by pointing on them and giving an appropriate speech command or by other components like the person detector or object classifier. For example the person detector could add every detected person as a pointing target. If the robot is asked to serve some person, the person to serve could easily be specified by pointing on it. With the above described abilities this application can



**Fig. 10** Left: Pointing target recognition accuracy is improved if fingertip is detected instead of just the hand region. Right: Pointing target recognition accuracy is improved if center of the head is detected instead of just the chin

be considered to work in the “real world” and is usable for “real” applications like grasping an object pointed at or serving a person pointed on.

The experimental evaluation with eight different subjects shows that the overall average pointing gesture recognition rate of the system for distances up to 250 cm (head to pointing target) is 86.63% (with a distance between objects of 23 cm). Considering just frontal pointing gestures for distances up to 250 cm (head to pointing target) the gesture recognition rate is 90.97% and for distances up to 194 cm (head to pointing target) even 95.31%. The average error angle (measured angle between the line from the head to the pointing target and the line-of-sight from the face through the hand towards the pointing target) is  $7.28^\circ$ .

#### 4 Semantic scene understanding

Objects are frequently involved in service tasks. Object understanding is important for the fulfillment of the task whereas a high level of flexibility is required to cope with real world conditions. Of particular concern are the different appearances of objects with common semantical concepts and the similar appearances of objects semantically unrelated. To solve this problem we are working on a two level process. In the first step a coarse categorization into specific object categories based on the statistical appearance of visual object properties is achieved. This allows to relate unknown objects-instances to known categories. Afterwards this information is used to support and guide a finer categorization based on text extracted from the object. With this, we intend to eliminate ambiguities. For example, reading “Pepsi<sup>®</sup>” from a magazine cover is weak evidence of the object being a “Pepsi<sup>®</sup>” in comparison with reading the same text out of a bottle or can.

##### 4.1 Object categorization

The presented visual object perception system categorizes *unknown* domestic object instances like cups, glasses, bottles or cell-phones to their respective category.

Such approach provides a new ability compared to commonly applied recognition ones, like for object-related service tasks where the semantical concept of a object instance is of interest (e.g. in serving tasks where (possibly unknown) instances of a glass are required) rather than the recognition of the individual object instance.

The system is grounded on 2D image information and relies on a *geometric-free* approach called *Bag of Features (BoF)* [14, 34, 48]. This approach has shown its reliability and robustness to object occlusions, illumination changes and especially, to geometric deformations of objects which belong to a common category, since the *BoF* approach does not rely on global geometric information; instead it relies on the extraction of local invariant features. The *BoF* approach is based on the assumption that each object category is distinguishable by its individual independent statistical appearance of salient-invariant-local features which are extracted from images.

In the first step of the *BoF*-based object categorization process, the *extraction of invariant features* from images is exploited to transform the visual image information into a compact representation, which provides rich recallable information of the image, i.e. similar information is extracted if the image content is transformed by scale, shift or rotation. Commonly Scale-Invariant-Feature-Transform (*SIFT*) has been successfully applied [33]; however our experiments have shown that Speeded-Up-Robust-Features (*SURF*) performs a better feature extraction, due to its feature recallability and computational lower cost. Next, a *visual dictionary* is created, which is used to analyze the feature frequencies from images that have passed the feature extraction process. Therein, the features of training images are grouped by similarity, in order to generate clusters of similar features. Based on a cluster, a generalized feature is constructed – *visual word* – which represents the center of a cluster. A *k*-means-based algorithms is applied for clustering due to its simplicity and low computational cost. An appropriate number of clusters *k* (*dictionary size*) is a crucial factor which influences the categorization performance. The discriminability is decreased if a too small or too large dictionary is used. Most approaches heuristically examine the dictionary size or they set the dictionary size to a fixed number [48, 44]. In contrast, we systematically analyze the dictionary size by *cluster validation* i.e. we use the *Dunn-validity index* [16] to examine the compactness of the cluster space. Additionally discriminative visual-words are *filtered* and *weighted* accordingly to their relevance and importance for each object category. After the dictionary is generated, the extracted features of a query image are assigned to the nearest visual words by *nearest-neighbor-search*. The comparison between the visual word frequencies, i.e. distribution of the visual words, of a query image and of labeled example images leads to a decision about the corresponding category of the query image. Supervised machine learning approaches like Support Vector Machines (*SVM*) are often applied [14, 48], since they have shown an enhanced robustness to discriminate sets of categories. The learners are trained with the visual word frequencies of training objects to generate a *prediction model*. However in our work we do not rely on the decision of a single classifier, since a single classifier provides a certain accuracy and also a high risk of misclassification bias for specific categories. To enhance the accuracy and to reduce the influences of those biases, a *set of six classifiers* combined with *feature-selection algorithms* is trained and their outcomes are *combined* by a modified majority-voting-based *sum-rule* to make a more *robust* and *reliable* decision. Moreover, our approach does not completely neglect the object shape information,

since it provides a useful indication about the corresponding category. We combine the set of *feature-based classifiers* with an additional *shape-based classifier* which is based on *shape descriptors*, in order to support an appropriate final decision. Also, we apply a basic, but sufficient *object detection* approach based on 2D image segmentation which allows to *detect multiple objects* on a table-top; thereby potential object boundaries are extracted. These boundaries are used to relate extracted features to objects. Afterwards the features of the detected objects are independently analyzed by the visual dictionary and classified by the feature- and shape-based classifiers.

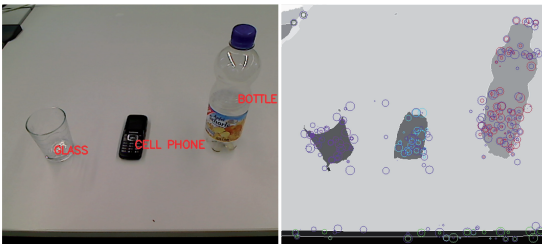
The experimental evaluation has shown that the classification accuracy is enhanced if the size of visual dictionary is indicated by the value of the *Dunn-validity-index measure* compared to randomly chosen dictionary-sizes. Thereby, dictionary sizes in the range from 100 to 1000 visual words are analyzed; dictionary sizes corresponding to *local maxima of Dunn-validity values* have shown to be a measure that leads to a discriminative visual dictionary as the results in table 8(green) show. Furthermore, the evaluation has shown that combining additional classifiers

**Table 8** The average classification error regarding the test set is shown of each classifier which is trained with randomly chosen and Dunn-validity-index indicated dictionary sizes. The classification error in the brackets shows the error if an appropriate dictionary size is chosen, i.e. the lowest classification has been achieved: 2-cat.=270 words, 3-cat.=400 words, 4-cat.=325 words.

Classifier	Number of supported object categories					
	2		3		4	
	Rand.	Dunn.	Rand.	Dunn.	Rand.	Dunn.
SVM	1.91%	0.23%(0%)	5.14%	2.04%(2.4%)	8.01%	6.65%(5.9%)
SVM+Entropy	1.71%	0.45%(0%)	4.84%	2.29%(1.2%)	7.38%	6.28%(2.7%)
SVM+PCA	1.91%	0.67%(0.9%)	3.93%	2.78%(1.2%)	6.73%	5.87%(4%)
AdaBoost	5.34%	5.65%(3.6%)	7.78%	7.50%(9%)	15.1%	13.17%(12.7%)
AdaBoost+PCA	3.62%	2.49%(2.7%)	7.27%	7.59%(7.8%)	10.98%	9.30%(9%)
AdaBoost+PCA+IAFS	4.23%	3.17%(2.7%)	6.66%	6.18%(6%)	10.37%	9.85%(9.5%)

*generally improves* the classification performance. However, experiments have revealed that a *certain combination of particular classifiers* can lead to the lowest classification error compared to the error of the most accurate *single* classifier or if the *entire set* of classifiers is combined – as shown in fig.9(left). The system has

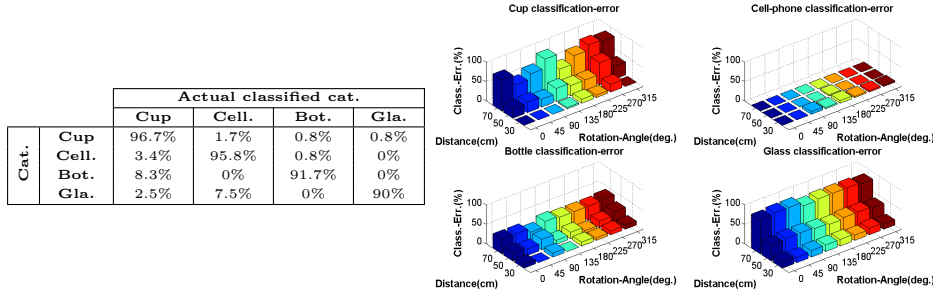
No. cat.	Classifier comb. with the lowest classification error	Error(All classifiers combined)
2	SVM, S.+Entropy, S.+PCA, AdaBoost, A.+PCA+IAFS	0%(0%)
3	S.+Entropy, S.+PCA, AdaBoost	0.6%(1.8%)
4	SVM, S.+Entropy A.+PCA	2.2%(3.1%)



**Fig. 11** Left: combinations of classifiers which result to the lowest classification error (test set) with respect to the number of supported categories. Right: example classification from the robot perspective: at the right the extracted object boundaries with the detected features are displayed; left, the system outcome is shown.



been trained with those combinations (including shape-based classifier) and integrated to the service robot whose camera is focused on a table top. Thereby, the system is trained for a robot-object distance of  $\approx 30\text{-}40\text{ cm}$ . A typical classification from the *robot-camera perspective* is depicted in fig.9(right). From such perspective,



**Fig. 12** Left: the classification accuracy w.r.t. the four categories (single objects on a table-top with fixed object perspective). Right: avg. classification error results of each category with respect to object-robot-distance and object-rotation-angle.

fig.10(left) presents the classification accuracy of perceived objects related to four categories: certain misclassification biases for particular categories are found. Also we investigated the categorization behavior depending on the *robot-object-distance* and *object-rotation-angle* – see fig.10(right). Different behaviors are observed due to the *presence* and *absence* of *descriptive category-related features* caused by variation of robot-object-distance and object-rotation-angle; also generic properties of the object categories like *object material* or *size* influences the classification result.

The object detection based on a basic image segmentation has shown a satisfying trade-off between computational cost and accuracy of the extracted objects boundaries. However to enhance the object detection, i.e. to provide robustness against object occlusions and cluttered environments such as in real world situations, approaches based on normalized cuts or 3D depth information could improve the process.

To gather more semantical information about a detected and categorized object, text can be localized on the object and used as input for text mining and understanding.

## 4.2 Text mining and understanding

Text constitutes a rich and readily available source of information with a large potential of applicability in *autonomous mobile robots*. However, the importance of text as source of *semantic information* has been neglected and is just starting to be considered as an alternative (refer to [50] for recent work). We are interested in using text for *product identification*. This would help to overcome the inherent *lack of generalization* suffered by *appearance* based object classification. This problem arises due to: different products having similar appearance, the same products having different appearance across different vendors and transient appearance, e.g. special Christmas product wrappings. This could easily render useless an appearance based classifier. Text on the other side, contains regularities that can be used

to identify products, e.g. Ketchup bottles will often exhibit text such as “Ketchup”, “Tomato”, “Sauce”, etc. whereas rat poison bottles are unlikely to have “delicious” written on them. Using text for product identification poses many challenges:

- Robust text information extraction (TIE) from natural scene images is still an open problem. Particularly due to the large variability in terms of font, size, color, layout, symbol repertoire, language, etc. Specially in product wrappings, text has a tendency to have non-standard looks and layouts.
- Common annoyances in computer vision tasks such as background clutter, noise, perspective distortion, occlusion, etc. are also present.
- Optical character recognition (OCR) systems require text images of large contrast, high resolution, clean background and standard fonts and layout.
- Context does matter, e.g. reading “Pepsi<sup>®</sup>” on the cover of a magazine is not a strong evidence that the object is a Pepsi<sup>®</sup>. Assistance of a *object categorization* approach as the one described in Section 4.1 would be a great help.
- Text can be ambiguous and interpreting it robustly can require *probabilistic models* and *ontologies* of the objects to be recognized and associated text. The web might be a good source for model generation and ontological knowledge.

In the present work, we focus on TIE using a connected components (CC) based method. The input image is first segmented using Niblack binarization. Then each of the resulting segments (i.e. CCs) is classified into *text* and *non-text* using a support vector machine (SVM) [13]; afterwards a simple heuristic removes isolated CCs under the assumption that text elements are usually found close to other text elements. CC classification using supervised machine learning based becomes cumbersome because of the need for labeled CCs to prepare training datasets for CC classification. We aim at decreasing this effort by using synthetically generated text. Training examples for the text class are created by a python script that renders random strings with random font, size and rotation. The text images are created in two versions (see Figure 11), i.e. a binary image representing an ideally segmented text image and a color version using random colors for background and foreground. For the non-text class, we use the ICDAR train dataset images. After binarizing the images and splitting them into planes, we use the available groundtruth (word bounding rectangles) and remove all the CCs overlapping the words, unless they completely contain the word. Features are extracted from the segments information (e.g. width and height), two binary images (one with and one without filling segment’s holes), in which the only element rendered is the CC being evaluated. The color version of the synthetic images are used to extract the contrast at the segments’ borders (during on-line operation, this information is extracted from the input image). Afterwards, we train an SVM with Gaussian kernel using a cross-validation procedure. Refer to [?] for a detailed description of the features. We limit to mention that we also use Hu moments [31], Zernike- and Pseudo-Zernike invariants [59].

We performed our experiments on the ICDAR test dataset and a custom groundtruth<sup>5</sup> containing the bounding boxes of text CCs in the test images. The results are given in terms of precision and recall as defined by [41] and [62]<sup>6</sup>. We trained four SVMs all using a set of seven features plus: H7) Using Hu moments

<sup>5</sup> Available at <http://home.inf.h-brs.de/~jalvar2s/>

<sup>6</sup> <http://liris.cnrs.fr/christian.wolf/software/deteval/index.html>



**Fig. 13** Examples of the images from which training examples are extracted. 11(a) and 11(b) are synthetically generated text images for the same random string; 11(a) is an ideally binarized version and 11(b) a color version. 11(c) and 11(d) non-text class examples

**Table 9** Classification performance table. Note that the classifiers trained using Zernike and Pseudo-Zernike invariants produce very similar results and outperform H7.

Classifier	ICDAR			Wolf		
	$p$	$r$	$f$	$p$	$r$	$f$
H7	0.57	0.55	0.56	0.53	0.51	0.52
Z10	0.68	0.56	0.62	0.63	0.52	0.57
P10	0.76	0.52	0.61	0.70	0.48	0.57
Z10-P10	0.77	0.51	0.61	0.71	0.48	0.57

Z10) Using Zernike invariants P10) Using Pseudo-Zernike invariants and Z10-P10) using Zernike- and Pseudo-Zernike invariants. Z10, P10 and Z10-P10 use moments up to the 10<sup>th</sup> order. Classification performance is given on Table 9.

Our results show that the SVMs were able to generalize from the synthetic text instances to real images. In general, the classifiers using Zernike and Pseudo-Zernike invariants performed better; however, this comes with a high performance penalty since running the evaluation with H7 takes just some minutes, whereas the other classifiers take hours. Using a staged classification scheme might solve this problem, by focusing more complex features on more promising CCs while rejecting the rest. In real robotic applications, TIE’s problems start with the robot acquiring “good” input images and we believe that sensor fusion and active vision are necessary to do this.

#### 4.3 Navigation – online SLAM, path planning and motion control

A fundamental prerequisite for the application of autonomous mobile service robots is safe navigation in domestic environments which tend to be cluttered and highly dynamic. Common approaches to mobile robot navigation address this problem in different stages. First a static map of the environment is built, e.g., by joysticking the robot around and processing the acquired sensory information offline to build a map. In the application phase, the robot localizes itself and plans paths to goal location in this static map. Since the map does not get adapted to changes in the environment, latest sensory information is taken into account by local path planners and reactive collision avoidance behaviors when actually traveling to the goal location.

In contrast to that, our primary goals when designing the navigation component was to not decouple generation and application of the map. That is, we want

to consider mapping and localization jointly as in standard *Simultaneous Localization and Mapping (SLAM)* approaches in order to continuously adapt the map to changes in the environment. This has the advantage that permanent changes to the robot’s workspace (such as re-arranging furniture) are represented in the robot’s map are taken into account when initially planning paths. In addition, it also allows for autonomously exploring the environment, i.e., to let the robot build a map of its environment on its own by, respectively, sensing previously unexplored regions and to fill holes in the so far built (and initially empty) map. However, it also requires for a fast (real-time applicable) and robust approach to SLAM. The latter thereby refers to not inducing unrecoverable errors in the map that may hinder the robot from accomplishing assigned tasks.

Over the last two decades, different algorithms for addressing the SLAM problem have been proposed. In recent years, there is a trend to probabilistic SLAM algorithms using, for example, Extended Kalman Filters (EKFs) [40], Unscented Kalman Filters (UKFs) [11], Sparse Extended Information Filters (SEIFs) [60] or Rao-Blackwellized Particle Filters (RBPFs) [22]. The latter one is going to be used to evaluate the performance (and robustness) of our SLAM approach. The approaches mentioned above explicitly handle uncertainties about the conducted estimates and the processed sensory information by estimating a probability distribution over the possible solutions. While they achieve robust and accurate results, the involved computational effort often prevents their application to large problem instances and hinders real-time applicability.

The fundamental idea of our approach is to address SLAM by means of incremental registration using the Iterative Closest Point (ICP) algorithm [6, 12, 64]. It operates on points clouds and can thus be used with any kind of range sensing device, such as 2D laser ranger finders. For simplicity, we build two-dimensional maps, but can account for the clutteriness of the environment by using 3D sensors and the concepts from [28] that allow for efficient 2D navigation using 3D data.

The idea of incremental registration is to, respectively, build a point map of the environment and a meta point cloud  $M$ . The first acquired point cloud  $D_0$  makes up the initial model  $M_0$ , i.e.,  $M_0 = D_0$  with the map’s origin coinciding with the robot’s pose where the first point has been acquired. To account for new information, subsequently acquired point clouds  $D_i$  are registered against the so far built model  $M_{i-1}$  in order to estimate the robot’s pose where  $D_i$  has been acquired and, finally, to add  $D_i$  in order to obtain the update point map

$$M_i = M_{i-1} \cup \{\mathbf{T}_i \mathbf{d}_i \mid \mathbf{d}_i \in D_i\}, \quad (1)$$

where  $\mathbf{T}_i$  is the transformation that correctly maps all points  $\mathbf{d}_i$  into the common coordinate frame of  $M_i$  and  $D_0$ .

Estimating  $\mathbf{T}_i$  by registering  $D_i$  and  $M_{i-1}$  is thereby done using the ICP algorithm. It iteratively estimates correspondences between  $D_i$  and  $M_{i-1}$  in the form of  $(\mathbf{d}_i, \mathbf{m}_j, e_{ij})$  where  $\mathbf{m}_j \in M_{i-1}$  is the closest point to  $\mathbf{d}_i$ , and  $e_{ij}$  the distance between  $\mathbf{m}_j$  and  $\mathbf{d}_i$ .  $\mathbf{T}_i$  (being composed of a rotation  $\mathbf{R}_i$  and a translation  $\mathbf{t}_i$ ) then results from aligning  $D_i$  with  $M_{i-1}$  by minimizing the distances  $e_{ij}$  between all  $k$  point correspondences:

$$\mathbf{T}_i = \arg \min_{(\mathbf{R}_i, \mathbf{t}_i)} \sum_k \|\check{\mathbf{m}}_k - (\mathbf{R}_i \check{\mathbf{d}}_k + \mathbf{t}_i)\|^2. \quad (2)$$

There are different closed form solutions for this optimization problem. We follow the SVD-based method in [30].

The approach as described has several shortcomings:

1. **False correspondences** can cause the registration to converge to incorrect local minima. In order to detect false correspondences and neglect them in the optimization, we 1) remove all correspondences with a distance  $e_{ij}$  larger than some threshold  $e_{\max}$  (exponentially decaying in the course of registration), 2) only consider points in  $M_{i-1}$  that are visible in  $D_i$  [42], and 3) we reject correspondences that contain the same map point  $\mathbf{m}_j$  and only keep the pair with smallest  $e_{ij}$  [65].
2. **Map size:** by updating  $M_i$  according to Eq. 1, we may store duplicates of points causing an unnecessary large size of the point map. In order to minimize the amount of memory for the map and to avoid the duplicate storage of points, we only add those points to  $M_i$ , that did not have a corresponding point in  $M_{i-1}$  within a distance of  $e_{\max}$ :

$$M_i = M_{i-1} \cup \{\check{\mathbf{d}}_{i,j} \mid \nexists \mathbf{m}_{i-1,k} \in M_{i-1} : \|\check{\mathbf{d}}_{i,j} - \mathbf{m}_{i-1,k}\| < e_{\max}\}. \quad (3)$$

3. **Changes in the environment:** The so far described approach only adds points to  $M_i$ , but never removes them. For an object moved from one location to another points modeling the object are added at the new location, but not removed from the old one. Similarly, new objects in the environment are accounted for, whereas objects being removed from the environment are not. To account for all possible types of changes, we additionally construct a grid map where each cell  $c$  models the probability  $p_{\text{ref}}$  that the respective region in the environment reflects laser beams. Points in  $M_i$  falling into regions with low reflection probability ( $< p_{\min}$ ) are removed from the map:

$$\check{M}_i = M_i \setminus \{\mathbf{m}_{i,j} \mid p_{\text{ref}}(c^{[\mathbf{m}_{i,j}]}) < p_{\min}\}. \quad (4)$$

A detailed description of the overall approach as well as additional extensions to ICP-based registration and experimental results can be found in [27]. These results show that the presented approach can 1) produce accurate and consistent maps of domestic environments while 2) being computationally efficient to process 2D laser range scans in real-time (75Hz with the used SICK laser range finders). How the SLAM approach is integrated into the navigation component is described in detail in [29]. In addition to moving to desired goal locations in the environment, the navigation component allows for showing the robot around in order to learn an initial environment model with semantically labeled places (human-guided exploration) as well as for exploring the robot’s workspace in a fully autonomous fashion (see [26]).

## 5 Results and lessons learned

In this article we have described an autonomous service robot *Johnny Jackanapes*. We focused on the component based software framework that is used to implement a deliberative robot control architecture based on the introduced Hybrid Deliberative Layer. Related to its application in the RoboCup@Home competition we

explained the algorithms for robust and multi-modal human machine interaction like gesture, speech and emotion detection as well as sound localization. As another major part of the robot system we have shown our implementations on semantic scene understanding namely object categorization and text mining as well as the robots navigation capabilities, online SLAM, path planning and motion control. The implementations have been evaluated under various conditions while the integration and overall system performance has been proven in the participation at various RoboCup@Home competitions.

**Table 10** Performance of *Johnny* in the last three RoboCup@Home competitions. **Ranking GO** lists the ranking of *Johnny* on the RoboCup GermanOpen. **Ranking WC** lists the ranking on the RoboCup WorldCup.

Year	Ranking GO	Ranking WC
2008	#2	#2
2009	#1	#1
2010	#1	#3

The RobCup@Home competition is held under very extreme environmental conditions. The environment itself is a typical household scenario with e.g. a living room and a kitchen. Most of the time the RoboCup is combined with an exhibition where the arenas are set up and even if the arena can not be entered by visitors they are open in a way that the visitors can watch the robots and vice versa. Further, the more people are attending, the higher is the noise level due to talking people oder moderators that explain the tests. Those conditions are very hard to reproduce in the laboratories where the robot software is developed. This environmental conditions lead to high requirements on the robustness of the individual robot components and overall system which is a key aspect for successful participation in those competitions. Beside the environmental conditions, those competitions afford a very well organized development structure because the times between different tests are usually only a couple of hours. During this breaks the arenas are blocked since the tests of different teams are carried out sequentially which allows no in place testing with the real hardware for the upcoming test. In this extreme situations a well designed debugging interface and simulation framework improve the ease of development by an order of a magnitude. Those lessons have a main reason for our quite successful participation in the RoboCup@Home competition as is shown in table 10.

Beside the work on the robots software, new hardware components are tested to keep track with the state of the art. This is in particular a different set of range sensors like 3D time of flight cameras or 3D laser range finders. with such hardware available also multiple components need coordinated access to the same hardware device like for example the gesture recognition and the object categorization. Further, such hardware create the need of sensor fusion to cope with uncertainty in the sensed environment. Such solutions would also improve the reliability and safety of the system and enable e.g. 3D path planning and grasp planning for the manipulation and navigation components.

**Acknowledgements** We gratefully acknowledge the support of our sponsors: The President of BRSU, the Department of Computer Science, the Association of friends and sponsors of BRSU, PMD Tech. Siegen, and DAAD.

## References

1. Kai Oliver Arras, Oscar Martinez Mozos, and Wolfram Burgard. Using boosted features for the detection of people in 2d range data. In *IEEE International Conference on Robotics and Automation 2007*, pages 3402–3407. IEEE, 2007.
2. Jun S. Liu Augustine Kong and Wing Hung Wong. Sequential imputations and bayesian missing data problems. *Journal of the American Statistical Association*, 89(425):278–288, 1994.
3. Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
4. Martin Steward Bartlett, Gwen Littlewort, Mark Frank, Claudia Lainscek, Ian Fasel, and Javier Movellan. Recognizing facial expression: machine learning and application to spontaneous behavior. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 568 – 573 vol. 2, 2005.
5. Michael Beetz, Dominik Jain, Lorenz Mösenlechner, and Moritz Tenorth. Towards Performing Everyday Manipulation Activities. *Robotics and Autonomous Systems*, 2010. To appear.
6. Paul J. Besl and Neil D. McKay. A method for Registration of 3–D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239 – 256, 1992.
7. Hong bo Deng, Lian wen Jin, Li xin Zhen, and Jian cheng Huang. A new facial expression recognition method based on local gabor filter bank and pca plus lda.
8. Thomas Breuer, Paul G. Ploeger, and Gerhard K. Kraetzschmar. Precise pointing target recognition for human-robot interaction. In *SIMPAR Workshop on Domestic Service Robots in the Real World '10*, 2010.
9. Norman Casagrande. Automatic music classification using boosting algorithms and auditory features. Master's thesis, University of Montreal, Department of Informatic and Operational Research, October 2005.
10. Norman Casagrande. Multiboost: An open source multi-class adaboost learner, 2005. <http://iro.umontreal.ca/casagran/multiboost/>.
11. Denis Chekhlov, Mark Pupilli, Walterio Mayol-Cuevas, and Andrew Calway. Real-Time and Robust Monocular SLAM Using Predictive Multi-resolution Descriptors. In *Proceedings of the 2nd International Symposium on Visual Computing*, pages 276–285, Lake Tahoe, Nevada, USA, 2006.
12. Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing, Special Issue on Range Image Understanding*, 10(3):145–155, 1992.
13. Corinna Cortes and Vladimir Vapnik. Support-vector networks. In *Machine Learning*, pages 273–297, 1995.
14. Chris Dance, Jutta Willamowski, Lixin Fan, Cedric Bray, and Gabriela Csurka. Visual categorization with bags of keypoints. In *ECCV International Workshop on Statistical Learning in Computer Vision*, 2004.
15. Hong Deng, Lian Jin, Li Zhen, and Jian Huang. A new facial expression recognition method based on local gabor filter bank and pca plus lda, 2006.
16. J. C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Cybernetics and Systems*, 3:32–57, 1973.
17. Christian Eckes, Konstantin Biatov, Frank Hülsken, Joachim Köhler, Pia Breuer, Pedro Branco, and L. Miguel Encarnação. Towards sociable virtual humans: Multimodal recognition of human input and behavior. *IJVR*, 6(4):21–30, 2007.
18. Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55:119–139, August 1997.
19. Yoav Freund, Robert E Schapire, and Murray Hill. Experiments with a new boosting algorithm. In *Machine Learning: Proceedings of the Thirteenth International Conference*, 1996.

20. Erann Gat. Integrating planning and reacting in a heterogeneous asynchronous architecture for controlling realworld mobile robots. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 1992.
21. Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated Planning: Theory and Practice*. Morgan Kaufmann, 2004.
22. Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters. *IEEE Transactions on Robotics*, 23(1):34–46, 2007.
23. Ronny Hartanto. *Fusing DL Reasoning with HTN Planning as a Deliberative Layer in Mobile Robotics*. PhD thesis, University of Osnabrück, November 2009.
24. Michi Henning. A new approach to object-oriented middleware. *IEEE Internet Computing*, 2004.
25. Rob Hess and Alan Fern. Discriminatively trained particle filters for complex multi-object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 240–247. IEEE, 2009.
26. Dirk Holz, Nicola Basilico, Francesco Amigoni, and Sven Behnke. Evaluating the Efficiency of Frontier-Based Exploration Strategies. In *Proceedings of the joint conference of the 41st International Symposium on Robotics (ISR 2010) and the 6th German Conference on Robotics (ROBOTIK 2010)*, pages 36–43, Munich, Germany, June 2010.
27. Dirk Holz and Sven Behnke. Sancta Simplicitas – On the efficiency and achievable results of SLAM using ICP-Based Incremental Registration. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1380–1387, Anchorage, Alaska, USA, May 2010.
28. Dirk Holz, David Droschel, Sven Behnke, Stefan May, and Hartmut Surmann. Fast 3D Perception for Collision Avoidance and SLAM in Domestic Environments. In Alejandra Barrera, editor, *Mobile Robots Navigation*, pages 53–84. IN-TECH Education and Publishing, Vienna, Austria, March 2010.
29. Dirk Holz, Gerhard K. Kraetzschmar, and Erich Rome. Robust and Computationally Efficient Navigation in Domestic Environments. In J. Balthes, M.G. Lagoudakis, T. Naruse, and S. Shiry, editors, *RoboCup 2009: Robot Soccer World Cup XIII*, volume 5949/2010 of *Lecture Notes in Computer Science*, pages 104–115. Springer, Germany, 2009.
30. Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987.
31. Ming-Kuei Hu. Visual Pattern Recognition by Moment invariants. *Information Theory, IRE Transactions on*, 8(2):179187, 1962.
32. Jared Jackson. Microsoft Robotics Studio: A Technical Introduction. *IEEE Robotics & Automation Magazine*, 14:82–87, 2007.
33. Yu G. Jiang, Chong W. Ngo, and Jun Yang. Towards optimal bag-of-features for object categorization and semantic video retrieval. In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, New York, NY, USA, 2007. ACM.
34. Frederic Jurie and Bill Triggs. Creating efficient codebooks for visual recognition. In *ICCV '05: Tenth IEEE Intl. Conf. on Computer Vision*, volume 1, pages 604–610, 2005.
35. Roger E. Kahn, Michael J. Swain, Peter N. Prokopowicz, and R. James Firby. Gesture recognition using the perseus architecture. In *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 734–741, 1996.
36. Takeo Kanade, Jeffrey Cohn, and Ying-li Tian. Comprehensive database for facial expression analysis. In *Proc. of International Conference on Automatic Face and Gesture Recognition*, pages:46–53, 2000.
37. Roland Kehl and Luc Van Gool. Real-time pointing gesture recognition for an immersive environment. In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 577–582, 2004.
38. Anastasios Koutlas and Dimitrios I. Fotiadis. An automatic region based methodology for facial expression recognition. In *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*, pages 662–666, 2008.
39. J. Kulikowski, S. Marčelja, and P. Bishop. Theory of spatial position and spatial frequency relations in the receptive fields of simple cells in the visual cortex. *Biological Cybernetics*, 43:187–198, 1982. 10.1007/BF00319978.
40. John J. Leonard and Hans Jacob S. Feder. A computationally efficient method for large-scale concurrent mapping and localization. In D. Koditschek J. Hollerbach, editor, *International Symposium on Robotics Research*, Snowbird, Utah, USA, 1999.



41. Simon M. Lucas, Alex Panaretos, Luis Sosa, Anthony Tang, Shirley Wong, Robert Young, Kazuki Ashida, Hiroki Nagai, Masayuki Okamoto, Hiroaki Yamamoto, and Others. ICDAR 2003 robust reading competitions: entries, results, and future directions. *International Journal on Document Analysis and Recognition*, 7(2):105–122, 2005.
42. Stefan May, David Droeschel, Dirk Holz, Stefan Fuchs, Ezio Malis, Andreas Nüchter, and Joachim Hertzberg. Three-dimensional mapping with time-of-flight cameras. *Journal of Field Robotics, Special Issue on Three-Dimensional Mapping, Part 2*, 26(11-12):934–965, December 2009.
43. Wim Meeussen, Melonee Wise, Stuart Glaser, Sachin Chitta, Conor McGann, Patrick Mihelich, Eitan Marder-Eppstein, Marius Muja, Victor Eruhimov, Tully Foote, John Hsu, Radu Bogdan Rusu, Bhaskara Marthi, Gary Bradski, Kurt Konolige, Brian P. Gerkey, and Eric Berger. Autonomous door opening and plugging in with a personal robot. In *ICRA*, 2010.
44. Frank Moosmann, Bill Triggs, and Frederic Jurie. Fast discriminative visual codebooks using randomized clustering forests. In *Advances in Neural Information Processing Systems*, pages 985–992, 2006.
45. Oscar Martinez Mozos, Ryo Kurazume, and Tsutomu Hasegawa. Multi-layer people detection using 2d range data. In *IEEE ICRA 2009 Workshop on People Detection and Tracking*, 2009.
46. Dana Nau, Okhtay Ilghami, Ugur Kuter, J. William Murdock, Dan Wu, and Fusun Yaman. SHOP2: An HTN Planning System. *Journal of Artificial Intelligence Research*, 20:379–404, 2003.
47. Kai Nickel and Rainer Stiefelhagen. Visual recognition of pointing gestures for human-robot interaction. *Image Vision Comput.*, 25(12):1875–1884, 2007.
48. Eric Nowak, Frederic Jurie, and Bill Triggs. Sampling strategies for bag-of-features image classification. *European Conference on Computer Vision*, pages 490–503, 2006.
49. Paul Ploeger, Kai Pervoelz, Christoph Mies, Patrick Eyerich, Michael Brenner, and Bernhard Nebel. The desire service robotics initiative. *KI - Zeitschrift Künstliche Intelligenz*, 4, 2008.
50. Ingmar Posner, Peter Corke, and Paul Newman. Using Text-Spotting to Query the World. In *To appear in Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010.
51. Cristiano Premevida and Urbano Nunes. Segmentation and geometric primitives extraction from 2d laser range data for mobile robot applications. In *Robotica 2005 Scientific meeting of the 5th National Robotics Festival*, pages 17–25, Coimbra, Portugal, 2005.
52. Céline Ray, Francesco Mondada, and Roland Siegwart. What do people expect from robots? In *Proceedings of the IEEE/RSJ 2008 International Conference on Intelligent Robots and Systems*, pages 3816–3821. IEEE Press, 2008.
53. Ulrich Reiser, Christian Connette, Jan Fischer, Jens Kubacki, Alexander Bubeck, Florian Weisshardt, Theo Jacobs, Christopher Parlitz, Martin Hgele, and Alexander Verl. Care-bot 3 - creating a product vision for service robot applications by integrating design and technology. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2009)*, 2009.
54. Robert E. Schapire. A brief Introduction to Boosting. *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, 1999*, 1999.
55. Frank Y. Shih and Chao-Fa Chuang. Automatic extraction of head and face boundaries and facial features. *Information Sciences Informatics and Computer Science*, 158:117–130, January 2004.
56. Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A Practical OWL-DL Reasoner. *Journal of Web Semantics*, 5(2), 2007.
57. Siddhartha S. Srinivasa, Dave Ferguson, Casey J. Helfrich, Dmitry Berenson, Alvaro Collet, Rosen Diankov, Garrat Gallagher, Geoffrey Hollinger, James Kuffner, and Michael Vande Weghe. Herb: A home exploring robotic butler. 2009.
58. Clemens Szyperski. *Component Software: Beyond Object-Oriented Programming*. Addison-Wesley Professional, 2002.
59. Cho-Huak Teh and Roland T. Chin. On image analysis by the methods of moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):496–513, July 1988.
60. Sebastian Thrun, Yufeng Liu, Daphne Koller, Andrew Y. Ng, Zoubin Ghahramani, and Hugh Durrant-Whyte. Simultaneous Localization and Mapping with Sparse Extended Information Filters. *International Journal of Robotics Research*, 23(7-8):693–716, 2004.

61. Thomas Wisspeintner, Walter Nowak, and Ansgar Bredenfeld. Volksbot - a flexible component-based mobile robot system. In *Proceedings of the RoboCup Symposium*, 2005.
62. Christian Wolf and Jean-Michel Jolion. Object count/area graphs for the evaluation of object detection and segmentation algorithms. *International Journal on Document Analysis and Recognition*, 8(4):280–296, 2006.
63. Yu Yamamoto, Ikushi Yoda, and Katsuhiko Sakaue. Arm-pointing gesture interface using surrounded stereo cameras system. In *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 4*, pages 965–970, Washington, DC, USA, 2004. IEEE Computer Society.
64. Zhengyou Zhang. Iterative Point Matching for Registration of Free-Form Curves. IRA Rapports de Recherche, Programme 4: Robotique, Image et Vision 1658, Institut National de Recherche en Informatique et en Automatique (INRIA), Valbonne Cedex, France, 1992.
65. Timo Zinßer, Jochen Schmidt, and Heinrich Niemann. A Refined ICP Algorithm for Robust 3-D Correspondence Estimation. In *Proceedings of the International Conference on Image Processing (ICIP)*, pages 695–698, Barcelona, Spain, 2003.