Using an Autonomous Robot to Maintain Privacy in Assistive Environments

Christopher Armbrust, Syed Atif Mehdi, Max Reichardt, Jan Koch, and Karsten Berns
Robotics Research Lab, University of Kaiserslautern,
P.O. Box 3049, D-67653 Kaiserslautern, Germany
{armbrust, mehdi, reichardt, koch, berns}@cs.uni-kl.de

Abstract

In our societies, the number of senior citizens living on their own is increasing steadily. The lack of permanent attention results in the late detection of emergency situations. Labour-intensive care is already a high burden for the society, therefore it seems reasonable to promote technology that helps to detect and react in case of emergency situations that elderly people may encounter.

In the last decade, assistive environments have been established by integrating surveillance devices into the living environments giving remote operators access to monitor the senior inhabitant at home for detecting emergency situations. However, due to poor privacy in terms of intrusion into the private life of an elderly person, there will be an unfavourably low acceptance of such systems.

This paper introduces a two-stage strategy and proposes to replace a possibly large number of human-controlled monitoring devices by a single autonomous mobile system. The first stage will be performed by the autonomous system to detect an emergency situation. The human operator will be obligatory only at the final stage when the system assumes that an emergency has occurred and the final evaluation of the situation is required. The self-assessment will reduce the human factor related to privacy issues.

Keywords *Privacy, Security, Assistive Environment, Assisted Living, Autonomous Mobile Robot, Elderly Care*

1. Introduction

It is a well-known fact that the demographic development in many developed countries is showing a steadily growing percentage of senior citizens. With increasing age, often living alone and due to immobility, these people start having problems with loneliness and missing interaction with relatives or friends. Senior citizens are often not able to perform all activities of their daily life without the help of caregivers, and face a higher risk of experiencing a medical emergency in unattended situations. Inpatient care would be a solution, but it would be a large financial burden for society and also would be seen as a setback for the quality of life of senior citizens. Therefore the approach should be to support elderly people to stay within their familiar living environment for as long as possible.

One methodology to increase the safety of elderly citizens would be to install a dense video supervision framework within an apartment to cover every corner of it (see, for example, [1]). This is not only expensive in terms of installation cost and monitoring cost, but also requires manipulation to the living environment, for instance installing wires and mounting cameras. Even more important, it gives the inhabitant the feeling of living within a "Big Brother" environment. The results of a study conducted by [2], unveil that elderly people require that there should not by any surveillance of them at home. Moreover, the authors' enquiries have shown that there is in fact a small group of people who would accept such conditions. It is the group of senior citizens who have already experienced heavy falls and have sometimes had to hold out all night, waiting for help.

1.1. Contribution

The authors introduce an electronic helper that serves as a communication help for elderly person and provides unobtrusive support in the normal living environment. It will improve the situation of an elderly person, by offering attention, providing safety and still maintaining privacy. The authors take the definition of privacy from one of the oldest definitions that is still influential, "Privacy is the right of the individual to be left alone." by Samuel Warren and Louis Brandeis presented in 1890 [1]. For this reason, the design of the electronic helper is such that it should minimise the external

[1] http://science.jrank.org/pages/10852/Privacy-DEFINITIONS.html

human intervention and maximise the autonomous performance of the robot.

The Autonomous Robot for Transport and Service (ARTOS) is a prototype for assisting senior citizens who are living on their own. Fig. 1 shows this robot, which is not larger than a common vacuum cleaner. It is an example of a mobile system that can serve as communication platform and is able to roam around autonomously. Its size and shape are optimised for obstructed living environments. Equipped with a certain amount of technical infrastructure, the robot can become a mobile interactive agent for an assistive environment.

Figure 1. The mobile robot ARTOS

Besides the standard use for transportation and communication, the robot also serves as a support device in emergency situations. This scenario is described in detail in Section 2.

1.2. Recent Approaches

Many assisted living labs have been established in the last decade, and also sophisticated mobile robotics is evolving closer and closer to real-life products. But not many projects combine the two areas to become a consistent scenario.

With the ambient assisted living hype, numerous environments have been equipped with such technology. The *Aware Home* at Georgia Tech in Atlanta [3], *house_n* [4] as part of *Placelab* at MIT, the *Assisted Living Lab* in Kaiserslautern, Germany [5] and *HomeLab* from Philips in Eindhoven, NL, are only some to be mentioned. Of course, all have the common aim of unobtrusively keeping the human being in the centre of attention. The integrated systems shall maximise comfort and convenience while keeping the amount of technical interaction as minimal as possible. [1], is one of the solution that may allow a remote caregiver to analyse the situation where multiple cameras have been used for fall detection. But the number of cameras to cover the complete home area and the expense of altering the environment is quite high.

The study conducted by [2], revealed that elderly people require that the Ambient Assisted home should do a lot of good things for them but there should not by any surveillance of inhabitants or any senor system that would collect any data. Moreover the question, “Are there any hidden cameras or hidden microphones in the home?”, always comes ahead whenever there is a reference to the Ambient Assisted Living.

Monitoring of elderly persons at home can also be achieved using tele-presence services provided by a mobile robot. Despite mounting fixed cameras in the home environment, a mobile robot with camera, controllable by the remote caregiver personnel to drive around the home can be used to enquire the health conditions of the elderly person. Telepresence for assisted living or medical scenarios using mobile robots is, for example, investigated by [6], [7] and [8]. [9] presents an approach of tele-presence, where a robot is being used to monitor the human being at home. The movement of the robot is controlled by a remote caregiver who can monitor the elderly person using the camera mounted on the robot. The *Robocup@Home* initiative even proposes and runs competitions concerning the abilities of home service robots [10].

Concerning safety and security, the privacy issue has always played a major role. For example, [11] states that already on the infrastructure level (in this case the ubiquitous middle-ware *I-Living*), means to ensure privacy shall be provided when collecting and merging all sorts of information about activities of daily life. The authors of [12] invented an algorithm for perturbing user location data in order to maintain privacy. They also proposed privacy metrics to evaluate their work.

According to [13] there is a considerable lack of security and privacy in the household robots. A hacked robot can be used to frighten the elderly person or it can even cause harm to the person by placing or dropping objects on the way. The tests conducted by [13], revealed that default accounts on the indoor robot WowWee Rovio[2] are not password protected. Moreover, some robots perform periodic upload of data for storage on the external servers. Sometimes its just the log-in information or an alive signal and sometimes it is a video stream. The video streams from web-cams have always been an issue but the difference between a robot’s video stream and the web-cams’ is that web-cams are static and cover a very limited area and mobile robots can move around in home and hence can cause much harm to privacy.

As mentioned in [13], Spykee robot[3] tries to register with the Spykee website by using wireless network

[2] http://www.wowwee.com/

[3] http://www.spykeeworld.com/

and sending unencrypted information to the website. Hackers launching attacks within wireless range or just outside the home like sniffing packets that are travelling by Internet Protocol (IP) can easily pickup such information. Once information of the robot is compromised, it is quite easy to cause harm using that robot.

In view of the above mentioned problems, the authors have developed an autonomous mobile robot, ARTOS, to take care of transportation and service needs of an elderly person. Besides that, certain measures with respect to privacy and security of the elderly person have also been taken into account in ARTOS.

1.3. Contents

This paper is structured as follows:

Section 2 introduces the scenario for the primary use case. It presents hardware and software aspects of the robotic platform that shall unobtrusively serve the user in transporting objects and establishing channels for communication. The section also gives an overview of the experimental setup and the lab environment. Section 3 explains how authorised personnel can use the robot as communication platform in emergency situations. This section also describes the means of autonomous navigation, supporting the teleoperation and arriving at the site of the potential emergency as fast as possible. Section 4 explains how to deal with obstacles in proximity, while Section 5 depicts the systems for localising a human and the robot.

In order to maintain a certain level of privacy, no information about the user and his apartment shall be stored on any external database. Hence, the robot itself must be able to create a model of its environment from the information gathered using the installed sensor system. How this is done is described in Section 6. The robot shall be unobtrusive and stay far away from the user unless an emergency is detected. In such a case, however, driving to the user as fast as possible is a critical requirement. An approach to meet this demand is described in Section 7. Experiments conducted in the testing apartment are presented in Section 8. Section 9 explains the steps taken to secure the information of the inhabitant of the home that would also respect the privacy of the elderly person. Finally, Section 10 concludes the work and presents an outlook on possible future work.

2. Experimental Setup

This section explains the primary use case and gives a brief overview of the robot's hardware and software features and the assistive environment. A detailed description of ARTOS' mechatronics system can be found in [14].

2.1. Emergency Scenario

As mentioned in Section 1.1, coping with emergency situations is the application scenario for illustrating the benefit of the robotic system regarding the inhabitant's privacy. Since the robot is not yet fully equipped with the necessary sensors, the authors rely partially on the availability of technical infrastructure of an ambient assisted living environment (see Section 2.3). This environment provides the location of the human inhabitant [15] and informs the robot about possible emergency situations. These events can derive from direct sensor information (e.g. fall sensors) or the information that a deviation from the expected inhabitant's behaviour has occurred. Examples for Activity of Daily Life (ADL) are given in [5]. An obvious deviation would be not going to the bathroom during a certain length of time.

The robot itself features sophisticated audio and video teleoperation and autonomous navigation capabilities. When an emergency is suspected, the system begins to move towards the human's last known position. If the suspicion is confirmed, a human operator is called for assistance. At any time during the robot's movement, the operator can log-in to the robot's teleoperation system to conveniently use camera and audio connection to evaluate the situation. When necessary, manual control of the navigation system can be taken. The final interpretation of the situation remains with the human operator. This part of situation assessment, at the current state-of-the-art, cannot be handled fully by a machine. But obviously, the possibility for a human other than the operator to breach the privacy of the elderly person is kept as small as possible, explained in Section 9.

Just as a side note: The system is not meant to be limited to emergency situations. It can be used for daily communication with relatives or friends.

2.2. Robot Hardware and Software Architecture

ARTOS is a small robot (59 cm long, 37 cm wide, and 41 cm high) with a differential drive kinematics that is equipped with a Hokuyo[4] laser range finder as well as two custom-made chains with a total of 14 ultrasonic sensors for collision avoidance, and two bumpers for collision detection. Collision in this context is that the robot may hit any object. This can be a human being

[4] http://www.hokuyo-aut.jp/

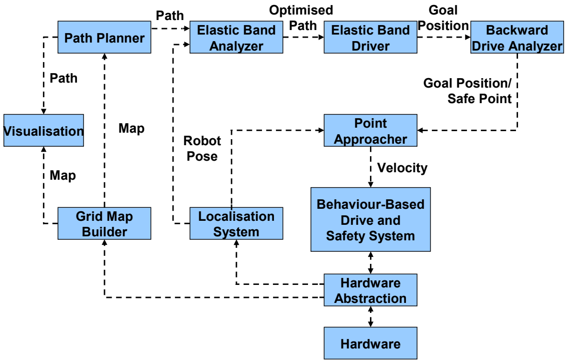


Figure 2. The elements of the control system

and eventually may cause harm to the person. Therefore, to protect the human being and also the robot, it is necessary that the robot deviates from the path of any collision.

The robot also features a Pan-Tilt-Zoom (PTZ) camera that can be used by authorised personnel to get an impression of the situation in the assisted living environment and a combination of a microphone and two speakers which can be used for establishing communication between the monitored person and the medical personnel.

ARTOS' control system has been implemented using MCA-KL[5], a framework for the support of robot control systems developed at the Robotics Research Lab of the University of Kaiserslautern. It is built up in a modular way, with many components being realised as behaviours of the behaviour-based architecture iB2C[6] (see [16]).

The structure of the control system is depicted in Fig. 2. A Hardware Abstraction Layer serves as an interface between the real robot and the higher layers of the control system. It contains the basic sensor processing and the actuator control. A behaviour-based drive and anti-collision system realises collision-free robot movements (see [17]). The mapping and localisation components generate the basic data used by the high-level navigation components, which realise path planning and path following.

2.3. Assistive Environment

A lab environment, serving as an example apartment, has been established recently[7]. It is a fully furnished apartment with an area of 60 m^2 (approx. 600 sq.ft.), comprising a living room, a bedroom, a bathroom and a kitchen (see Fig. 3). Equipped with a powerful collision avoidance and being able to create a map of its environment, the robot can safely manoeuvre between the rooms, which is very helpful for a human operator who is allowed, under certain conditions, to remotely take control of the robot.

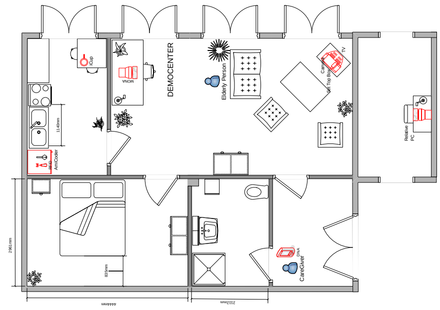


Figure 3. The BelAmI testing apartment

As previously mentioned, that the elderly people don't want to be monitored all the time. To respect their need of privacy, in terms of not being observed everywhere, a platform such as ARTOS should remain idle in some remote location when its services are not needed. This way, the environment is not monitored by any camera for most of the time. However, especially in case of an emergency, it is crucial that the robot finds the person needing help as quickly and reliably as possible. This requires advanced, robust navigation mechanisms and equipment with appropriate sensors. Therefore, the central subsystems and ideas implemented in ARTOS to reach these goals are presented in this context.

3. Teleoperation

As introduced in Section 2.1, at a certain point a human operator has to remotely operate the robot. The idea is to have a human to evaluate the situation in the living environment and to assess the condition of its human inhabitant. Based on web standards, the robot's graphical control interface can be accessed from any Internet enabled computer with a web browser – an advantageous solution with respect to ease of setup and deployment. Of course, access to the robot should only be granted to authorised personnel, such as care-givers and relatives – a crucial requirement with respect to privacy and security of the elderly person. How this is realised is explained in Section 9. The following section describes the modular and extensible web interface framework the authors have created to provide a convenient and simple way of teleoperation.

[5] MCA-KL: Modular Controller Architecture — Kaiserslautern Branch (see http://rrlib.cs.uni-kl.de/)

[6] iB2C: integrated Behaviour-Based Control

[7] http://www.belami-project.org/

3.1. The Web Interface

For the robot ARTOS, a framework for Java-based web interfaces and a respective editor were developed [18]. This Graphical User Interface (GUI)-layer is independent from the robot's internal control system. It operates on a set of interfaces provided by MCA-KL, (see Section 2.2). Notably, usage is not limited to ARTOS or our framework. A plug-in mechanism actually allows adding support for virtually any robotic platform. When a browser connects to the robot and valid credentials are provided, an applet is transferred from the GUI-server to the browser. This applet shows the GUI elements previously composed by the GUI editor and manages the network connection to the robot.

The network transport mechanism is a critical aspect of a teleoperated system such as ARTOS. In particular, the side-effects which arise with system control over the Internet need to be taken into account. As already stated in [19], limited bandwidth and unpredictable delays, in addition to different protocol stacks for direct remote control, can cause problems. Wireless connections have similar issues. In the worst case, temporal connection losses occur. Therefore, a semi-autonomous control approach in terms of *Goal-Point* or *Position Point* [9] has been chosen (see Section 3.3).

Regarding the network implementation, fault-tolerance, efficiency, low latency, data encryption, and quality-of-service are important features in this context. A suitable mechanism needs to deal with temporary connection losses and adapt to the available bandwidth. Having evaluated the alternatives, a slim, custom, TCP-based solution, tailored specifically to the applications' requirements, has been chosen and implemented.

For audio connections, using standard technology appeared most feasible. Skype[8] offers an out-of-the-box application which is used to enable the robot for telephonic services. Noise-reduction and echo-filtering features work fine in this context. These are important, since the elderly inhabitant is not required to wear a headset, rather the robot itself carries a microphone.

3.2. GUI Deployment

From a technical point of view, there are three ways of deploying a Graphical User Interface. Fig. 4 illustrates the different variants:

1. Standalone: The robot can be operated using the editor, a Java application, directly. This is useful as GUIs can be designed and tested simultaneously. In this case, however, a native connection to the robot's control framework is necessary, which is insecure and not optimised for wireless connections in terms of bandwidth and fluctuating availability.

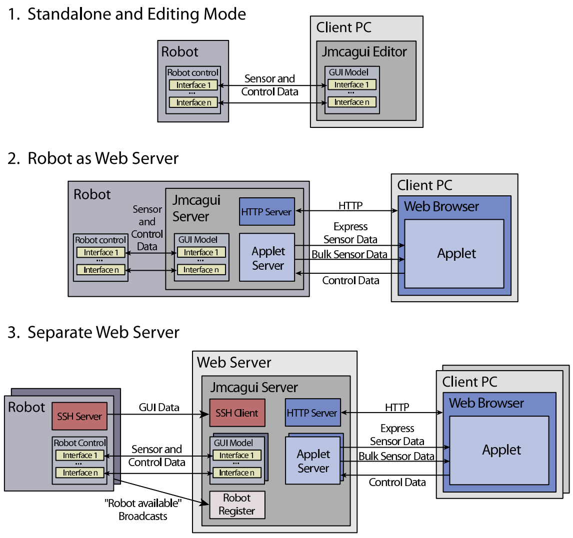


Figure 4. Deployment of Graphical User Interface

2. Server on Robot: In the second case, the GUI is deployed on the robot itself – publishing the GUI as an applet using an integrated web server. This example is shown in Fig. 5. Transferring data from the server to a client web browser – based on the network mechanism tailored specifically to this problem – is typically more efficient than connecting the editor natively with a robotic framework, as data transfer is optimised and latency is controlled. In this deployment scenario, a standard web browser can be used to connect.

Appropriate precautions should be taken in order to ensure security and therewith privacy in such configurations. When deployed behind a firewall, the robot can be accessed by either opening a certain port, by using an SSH[9] tunnel or by setting up a VPN. In the case of using a firewall, special rules can be used to allow only a group of machines to connect to that port. Besides deploying the VPN connection and firewalls in the wireless network of the robot, Section 9 explains the actions taken for securing communication by using secure sockets.

3. Separate Web Server: If the firewall is not to be levered, a generic instance of the server can be installed separately – possibly integrated in a central web server (see case 3 in Fig.4). In this case, the GUI data are still stored on the robot and the generic server retrieves them via SSH for publication.

[8] `http://www.skype.com/`

[9] SSH: Secure Shell

3.3. Control Paradigm

For respective user groups, GUIs of different manipulation complexity can be provided. The server can host different GUIs at the same time – accessible via tabs in the applet.

The implemented network mechanism for dealing with varying connection quality has proven valuable in practice. With latencies introduced by network connections and a limited field of vision, however, driving the robot directly raises difficulties. It therefore appears favourable to (optionally) let the operator control the robot with high-level commands. The robot's navigation system will deal independently and autonomously with such navigation orders (see Sections 5 to 7). Using a semi-autonomous approach, high latency and connection losses are far less of an issue. The operator can conveniently focus on the camera image and on the control of the PTZ camera while the robot navigates by itself. This makes teleoperation of the robot very simple. People are usually able to operate the robot without any prior training. In the application scenario presented here (see Fig. 5), the operator has two choices:

1. He may control the robot indirectly in an autonomous mode by clicking on the 2D representation of the living environment – thus specifying goal coordinates. The robot will drive independently to the specified position, even if the connection to the operator temporarily fails.

2. The operator may control the robot manually by using the joystick widget arranged at the bottom of the GUI. If the wireless connection fails, the robot will apply default values – in this case setting the speed to zero and stopping. However, the robot's navigation system also assists when using manual control. The obstacle avoidance (see Section 4), for instance, ensures that the robot does not collide with its surroundings.

The GUI, furthermore, contains a camera widget and a geometry renderer widget. The latter visualises a map together with the robot's distance sensor data. Apart from that, there are LED widgets indicating the motors' activation state, a button for toggling this state, an LCD widget indicating the robot's control cycle period, and so on.

4. Fast Reactions in Dynamic Environments

An important aspect in the life of an elderly person, is the security that no one is spying on him but

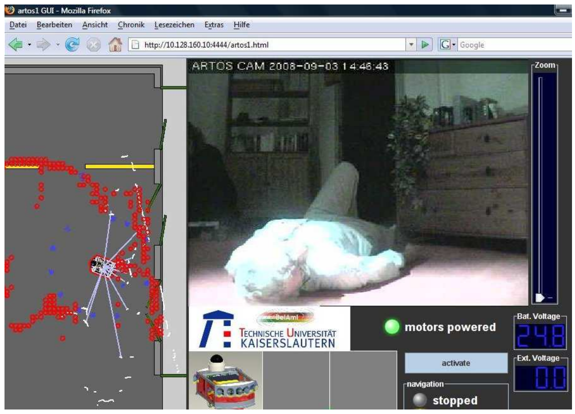


Figure 5. Teleoperation in an emergency scenario

yet he is been taken care of. A robot system without an autonomous behaviour is like all the other sensor systems that are monitoring a person all the time and continuously transmitting the information to the healthcare personnel. Therefore, one of the main aspects of the ARTOS project is the development of a robot that can move in the living environments without the help of an operator and limit external intervention to maintain the user's privacy. Therefore the abilities to properly detect different types of obstacles and to navigate between them without causing collisions are essential. Even if the robot is teleoperated by authorised personnel (see Section 3), an anti-collision system can be used to support the operator.

ARTOS collects information about the presence of obstacles with the help of its laser range finder and its two chains of ultrasonic sensors. Fig. 6 shows the area that is covered by these sensors. The information collected is just precise enough for collision avoidance. Detailed information about the apartment's interior, such as daily routine of the resident or furniture or articles placed in the environment, is not collected.

From the data of these sensors, three so-called virtual sensors are created. Each of them is represented by a sector map – a uniform data structure that abstracts away from the technical details of different sensor systems and offers a generic sensor interface to the anti-collision system. Each sector covers a certain area around the robot and provides information about the presence of obstacles contained within. Usually information about the closest obstacle is stored within a sector.

The collision avoidance system has been implemented as behaviour-based system using the iB2C (see [20]). It consists of a set of behaviours that can

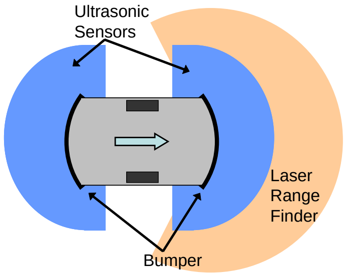


Figure 6. The sensors of ARTOS and the regions they monitor.

inhibit the motion commands coming from higher navigation layers, or issue different commands to keep the robot away from obstacles. A behaviour-based approach has been chosen in order to make the system robust against sensor failures, and easily extendable if new sensors are to be attached to the robot. As a last resort, two bumpers are attached to the robot's front and back to stop its motion in case a collision has occurred.

Additional and more detailed information about the anti-collision system can be found in [17].

5. Localisation

As a health care and service robot, ARTOS should always and precisely know its location and consequently the position of the human being in the environment. Always tracking the human being using a camera or a laser scanner may breach the human's privacy in a scenario where the person wants some time alone and does not want to be seen or observed. Therefore, there should be a mechanism to track the person even without physically seeing him. Such a mechanism is described in the following.

5.1. A Discreet Human Localisation

For many assisted living applications, knowing the position of the person is important. A typical example is the assumption that an elderly person getting up in the middle of the night and not returning to bed for more than an hour could be a sign of an emergency that requires further investigation.

Many approaches for localising humans in assistive environments use camera-based systems to keep them under observation at all times, but these approaches come with high costs and the loss of privacy. To ensure maximum safety, it is necessary to know the person's position at all times. But to ensure privacy, it is mandatory for an assistive system to collect only as much data as absolutely necessary. Therefore an approach based on RFID[10] tags has been chosen here. A grid of about 4,000 passive RFID tags with a grid size of 12.5 cm by 12.5 cm (5 in by 5 in) has been implanted under the carpet (see Fig. 7) of the aforementioned test apartment (see Fig. 3). The RFID tags used are standard ISO15693 tags operating at 13.56 MHz, as these tags and the corresponding readers are easily available and have satisfying characteristics (see [21]).

At first sight, laying out so many RFID tags does not seem to be very cost-efficient. However, the price of RFID tags is likely to decrease to only few cents per tag as soon as the carpet companies start selling smart carpets as a successful product. Hence, compared to the cost of the carpet itself, the expenses for RFID tags can be expected to be insignificant. Furthermore, a step-wise reduction in the grid density by successively hiding tags in experiments by the means of software is also planned, with the goal of determining the number of tags needed for a sufficiently good localisation estimation. This would help in reducing the number of RFID tags to be used in commercial carpets.

Fig. 8 shows an RFID reader that has been attached to a shoe that shall be worn by a human being in the assistive environment. It is equipped with a bluetooth transmitter which is used to connect the foot sensor to an intelligent ambient system. It should be mentioned here that no specialised or experimental components have been used to build up the foot sensor. It has been assembled from components of a mobile bluetooth RFID reader that is generally used for reading RFID tags on appliances in logistics. With a sensing range of about 10 cm (4 in), the reader is able to detect at least two tags with the mentioned grid density. This means that a person can be detected very reliably. In contrast to many other human detection systems, this approach does not suffer from blind areas due to shielding, diffraction, reflexion or other field strength issues.

$$N = \text{number of RFID tags in range} \quad (1)$$

$$\text{Pos}_{x,y} = \frac{1}{N}\sum_{i=1}^{N}\begin{pmatrix} x_i \\ y_i \end{pmatrix} \quad (2)$$

Equations 1 and 2 show how a position estimate Pos is calculated: It is simply the mean value of the 2D positions of all N RFID tags in range.

By sending the position estimate to an ambient system and connecting this system to the mobile robot, the latter will know where the person is without directly monitoring him or using a large set of firmly mounted sensors in the environment, see Fig. 9. With respect to

[10] RFID: radio frequency identification

Figure 7. RFID Under the Carpet for Localisation

Figure 8. Foot-attached RFID Reader

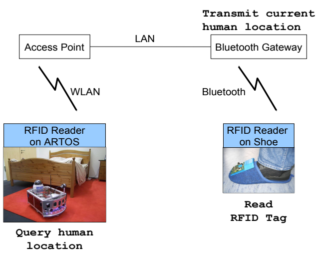


Figure 9. Communication between ARTOS and the shoe

privacy, the system has another advantage: As only location information about the elderly person wearing the foot sensor is created, everyone else is invisible to the system.

For an even higher degree of privacy, additional techniques could be employed. Several approaches are described in the literature. For example, a detailed analysis of privacy and security in RFID systems can be found in [22]. Its author developed new concepts for providing security and ensuring privacy and created a framework that supports their technical realisation.

It is hard to ensure that an elderly person always wears the RFID-enabled shoe. Therefore, a strategy based on the probability of the presence of a human being at a certain place needs to be developed, which would help the robot to find the human even in case when the shoe is not being used. The methodology is currently being developed and beyond the scope of current paper.

5.2. Robot Localisation

Mapping and navigation of robots have always been a challenge to researchers. Various techniques have been devised for precise mapping and navigation. Simultaneous Localisation and Mapping (SLAM) has been developed to localise the robot while building a map of the environment [23], with promising results. But according to [24], different implementations of SLAM pose certain limitations. SLAM works better in corridors and areas which are less populated and where there are predictable features in the environment. The dynamic nature of living environments makes it harder to track features, e.g. moving a chair may hide some of them, and hence reliable localisation may not be possible.

Of course, using SLAM can be an option, but since the assistive environment is already equipped with RFID tags under the carpet, these tags are being used as more reliable landmarks to localise ARTOS in the environment while moving autonomously or semi-autonomously instead of using a computationally intensive SLAM approach. These RFID tags are read during the movement of ARTOS and the information is then used to refine the odometry-based pose estimation.

Like the foot sensor, the robot is equipped with a small RFID reader board of 5 cm by 5 cm developed by FEIG Electronics[11] that is attached to the bottom of the robot. It features an integrated antenna and is connected via a UART[12] interface to the robot's on-board computer. While a position estimate can be computed

[11] `http://www.feig.de/`

[12] Universal asynchronous receiver/transmitter

in the way described above, the orientation estimation is more difficult. It is based on detecting several landmarks while the robot is moving. A combined arithmetic and heuristic calculation is performed to estimate the robot's orientation. Many experiments have shown that this approach is appropriate for indoor navigation.

The major benefit of the localisation methodology presented here is the high robustness of the system. If an RFID tag is detected, the information is accurate with a known maximum error. If no tag is detected, this information is also valid. Triangulating systems, in contrast, always yield a position estimate and the error that might have occurred is unknown.

Readers interested in more details about the RFID localisation system are invited to read [15].

6. Mapping Living Environments

In order to preserve privacy and security of the elderly person as best as possible, the approach described in the work at hand shall work without complex monitoring of the environment. Hence, there shall neither be cameras, laser scanners, nor other similar devices attached to the apartment's walls.

Regarding aspects of security, creating maps of assisted living apartments as a help for service robots might cause some concerns: Those maps would probably be stored on central servers, which would create data about people's personal environments over which they would not have direct control. A solution to this problem would be to not provide the robots with maps, but to fully rely on their ability to create the maps on their own. For the sake of security, no map of the environment is required to be given to ARTOS. Therefore, it is necessary for ARTOS to generate the map of the environment on its own and then use that map for path planning, obstacle avoidance and navigation. The quality of the maps generated by ARTOS is already sufficient for assistive tasks. For more complex maps, approaches like the one described in [25] can be employed.

In case of detection of a possible emergency, the robot has to drive to the person's estimated location and will probably encounter several new obstacles on its way there. It becomes mandatory that the robot is able to generate a map fast and update it continuously.

As mentioned in Section 5, the robot's localisation is realised by the combination of odometry and usage of RFID tags (see Section 5.2), so there is no need to use SLAM here. Hence, a simpler approach can be employed that is less computationally expensive. As more powerful computational hardware is more expensive and energy-consuming, keeping the computational load low is important.

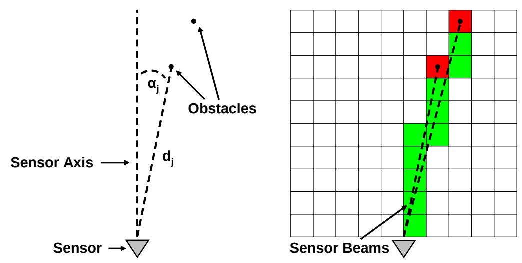
6.1. Single Map Creation

The main target of the mapping system is to support precise navigation, and hence a grid map approach has been chosen for map building. A grid map represents the world around the robot as an array of (usually uniform) grid cells. Each cell stores information about the area it covers, with the most important information usually being whether the area is occupied or not. The current occupancy belief is represented by an occupancy counter. Positive values are used for occupied cells, negative values for free cells and an occupancy counter of zero reflects an unknown occupancy state. The occupancy counters are limited so that a belief in an occupancy state cannot get too strong.

The laser range finder as well as the two chains of ultrasonic sensors are used as sources for the grid map creation process. The design of the *Grid Map Builder*, the component that contains the mapping functionality, allows for the addition of an arbitrary number of sensor systems which can be used simultaneously. The sensor data can be obtained in one of the two following formats:

1. Polar format: The sensor values are stored as a series of distance-angle-pairs. This format is used for the data from the laser range finder.

2. Sector map format: The polar sector maps introduced in Section 4 are used here to access the data generated by the ultrasonic sensors.



(a) Two obstacles in the environment. (b) Two sensor beams have been traced.

Figure 10. The creation of a grid map by tracing sensor beams (green/light grey: free; red/dark grey: occupied; white: unknown).

Depending on the format, the grid map creation is carried out in two different ways.

- When using the polar format, sensor beam tracing is used to alter the occupancy counters within the field of view of the sensor. Each sensor beam is traced in small steps. Whenever it hits a grid cell, the corresponding occupancy counter is decremented by 1. The counter of the last cell, i.e. the cell in which the obstacle lies, is incremented. The principle is illustrated by Fig. 10.

- When using polar sector maps, simply tracing sensor beams is not the best approach because the fact that the polar coordinates provide information about the *closest* obstacle in a sector is not used. Given a sector containing an obstacle, all grid cells in this sector which are closer to the sensor can be regarded as free. But with sensor beam tracing, only the occupancy counters of cells which lie on a line between the sensor and the obstacle would be decremented.

Algorithm 6.1 Polar Sector Map

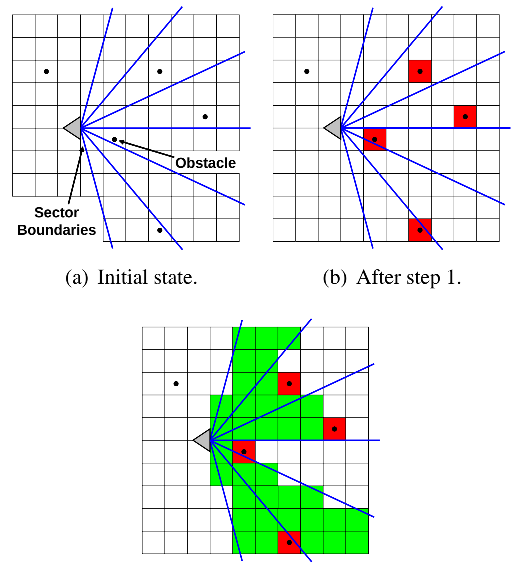
Step 1:
for all sectors s with obstacle o **do**
  calculate cell c containing o
  increment occupancy counter of c
end for

Step 2:
for all cells c in rectangular area around sensor **do**
  calculate sector s containing c
  if (distance(c) $<$ distance(o)) **then**
    decrement occupancy counter of c
  end if
end for

A solution is to process the sector map data in two steps as shown in Algorithm 6.1. In the first step, the cells in which obstacles lie are marked. Each sector contains at most one obstacle. All sectors have to be traversed, the polar obstacle data has to be transformed into map coordinates, and finally the cells containing obstacles have to be marked. In the second step, all cells in a rectangular area around the sensor are traversed. For each cell, the coordinates of its centre are transformed into polar coordinates of the sensor coordinate system. The information is then processed to determine whether the cell is closer to the sensor than the obstacle of this sector. If this is the case, it is regarded as free and its occupancy counter is decremented. Fig. 11 illustrates the two steps.

6.2. Map Integration

Using the above-mentioned procedure, a grid map is created from the data of *one* sensor. The integration of *several* sensors requires that the procedure is repeated



(a) Initial state.

(b) After step 1.

(c) After step 2.

Figure 11. The creation of a grid map using a polar sector map as data source.

for each of them. This also requires the use of different arrays of occupancy counters for different sensors, otherwise an obstacle that is *only* visible to sensor S_1 would be removed when processing the data of sensor S_2, which does not see it. Therefore, in every execution cycle, the data of each sensor are processed and the corresponding array of occupancy counters is updated. Then the data of these arrays are aggregated to build one combined grid map according to the following rules:

1. If a cell is occupied in at least one sensor's array, then its counter in the combined grid map is set to $+1$.

2. If a cell is not occupied in any of the sensors' arrays and is free in at least one sensor's array, then its counter in the combined grid map is set to -1.

3. If a cell is not occupied or free in any of the sensors' arrays, then its counter in the combined grid map is set to 0.

A grid cell stores additional information such as an occupancy counter and the information whether the counter may be changed or not. The most relevant information fields contained in one grid cell are shown in Table 1. The *is_fixed* field is set for cells whose occupancy counters shall not be changed. *dist_to_start* and *est_length* are used by the path-planning mechanism described in the next section. The combined grid map

Table 1. The main information fields contained in one grid cell.

Field	Stored Information
occupancy_counter	belief about the occupancy
is_near_obstacle	whether cell is close to an occupied cell or not
is_fixed	whether occupancy belief is fixed or not
dist_to_start	distance to the start point of a path
est_length	estimated length of a path going through this cell
update_time	time of the last update of the counter

is provided to the path planning component by using a blackboard mechanism offered by MCA.

It shall be pointed out that the mapping approach described here only stores the information that is necessary for navigation. There is no tracking or identification of single objects, so the user can be sure that no one will collect information about the things he stores in his apartment.

7. Getting to the Place of an Emergency

The mapping system described in Section 6 is capable of creating and updating a model of a dynamic assistive environment using the data of the robot's main sensors. Based on this model, the robot must be able to drive to a specific location as fast as possible. Especially in emergency situations, the time to reach the person should be minimal and the robot should adopt the shortest possible path from its current location to the location of the person. But to respect the elderly person's privacy and behave as unobtrusively as possible, the robot should normally stay at a distance from the person.

In the following, it is explained how the robot calculates shortest paths, keeps a certain distance to objects while following those paths, and deals with the presence of unexpected obstacles. Refer to Fig. 2 for the names of the control system's components.

7.1. Creating Paths

When a new path has to be created, the *Path Planner* first reads the current grid map from the blackboard. It then executes the A*-algorithm [26] to calculate the shortest path from start point s to destination point d. This algorithm processes cell by cell, starting with s, until a path to d has been found. In each processing step, the cell with the lowest cost is chosen as the next cell to be processed. For a cell c, the cost $f(c)$ is

$$f(c) = g(c) + h(c), \tag{3}$$

where $g(c)$ denotes the cost for the shortest *known* path going from s to c and $h(c)$ denotes the *estimated* cost of a path going from c to d. The Euclidean distance is used as heuristic function. Before planning a path, the obstacles in the map are enlarged by marking the cells close to them as "neighbours". High costs are assigned to these neighbours so that other free cells may be preferred. As a result, paths do not lead the robot close to obstacles unless it is necessary.

When a path has been created as a sequence of grid cells, the *Path Planner* calculates the "relevance" of points. A point is either *intermediate* or *relevant*. In contrast to intermediate points, the direction of the path changes at relevant points. In order to get an optimised path and a smooth robot movement, it is necessary to tune some of the path points. This can be achieved by considering the fact that the direction of the path does not change at intermediate points and thus they can be altered (see Section 7.2). By contrast, relevant path points change the direction of the robot's movement and hence cannot be modified. To accommodate the dynamically changing environment, the path is replanned after a certain interval of time.

Finally, a path is represented as a series of triples in the form (x-coordinate, y-coordinate, relevance). The aforementioned blackboard mechanism is used to provide a path to other components of the navigation system and to the visualisation module.

7.2. Driving along Paths

The functionality of driving forward along paths is realised by the *Elastic Band Analyzer*, *Elastic Band Driver*, and *Point Approacher* modules. The path planned by the *Path Planner* is the shortest path from the source to the destination. Following exactly those points might result in getting the robot too close to some obstacles which will cause reduction in speed and thus the robot might take a longer time to reach its destination. To overcome this problem, the elastic band approach [27] has been used. The algorithm has been implemented in the *Elastic Band Analyzer*, which reads the path from the blackboard and optimises it with respect to a smooth robot motion, a short driving time, and a safe distance to obstacles.

The *Elastic Band Driver* reads the path optimised by the *Elastic Band Analyzer* from the blackboard and sends its relevant points one by one to the *Point Approacher*. It also offers additional functionality such as calculating a target orientation, which is important for the use of the *Elastic Band Driver* on non-holonomic robots.

The *Point Approacher* receives target coordinates and the current robot pose as input, and calculates a desired velocity v_{des} and angular velocity ω_{des} depending on the distance d and absolute angle $|\alpha|$ to the target (see Equations 4 and 5):

$$v_{des} = \begin{cases} 0 & ; d \leq d_{min} \\ 1 & ; d \geq d_{max} \\ \frac{1}{2} + \frac{1}{2} \cdot \sin\left(\left(\frac{d - d_{min}}{d_{max} - d_{min}} - \frac{1}{2}\right) \cdot \pi\right) & ; \text{else} \end{cases} \quad (4)$$

$$|\omega_{des}| = \begin{cases} 0 & ; |\alpha| \leq \alpha_{min} \\ 1 & ; |\alpha| \geq \alpha_{max} \\ \frac{1}{2} + \frac{1}{2} \cdot \sin\left(\left(\frac{|\alpha| - \alpha_{min}}{\alpha_{max} - \alpha_{min}} - \frac{1}{2}\right) \cdot \pi\right) & ; \text{else} \end{cases} \quad (5)$$

By comparing the robot's orientation to α, ω_{des} can be calculated from $|\omega_{des}|$. d_{min}, d_{max}, α_{min}, α_{max} mark the distances and angles at which v_{des} and ω_{des}, respectively, take their extreme values. If d_{max} is reduced, for example, the robot will drive longer at the maximum speed when approaching a target. If it should decelerate earlier, d_{max} has to be increased. To make the changes of v_{des} and ω_{des} smoother, sigmoid functions are used.

7.3. Backing off

Living environments are very dynamic in nature. Despite every effort to generate a map of all the fixed and dynamic obstacles, there are many dynamic obstacles that obstruct the shortest path to the destination and sometimes render the destination unreachable. Therefore, a component is required to make the robot back off after acknowledging that it cannot go any further and there is not even enough space to turn around. In this case, the only choice left is to come back to a previously visited position where it was safe for the robot to move, and re-plan the path to the destination. After backing off, the map will contain new information about the dynamic obstacles that prevented the robot from following the original path, and the new path will lead the robot around these obstacles.

Drive Backward shall realise this behaviour. It considers the robot's velocity after a certain time period. If the velocity at some point is above a certain threshold, the robot is in a comfortable zone where it can easily move around. The current point is recorded as *safe point* so that the robot can drive back to it later if necessary. There can be two cases in which the velocity can be below the threshold. First, it is possible that the robot is moving through a tight corridor and there are lots of obstacles around it. Second, the robot has got stuck at some place and can neither go forward nor turn around. In the first case, after crossing the corridor the robot can regain a high velocity. Therefore, a certain time is given as a grace period to the robot to regain a high velocity or change the path by turning around before driving back. In the second case, there is no option other than to drive backward to the safe point.

During the experiments, it was observed that while driving back, dynamic obstacles may keep ARTOS from reaching the safe point (see Fig. 12). The robot will remain in *Drive Backward* mode, trying to reach the *safe point*, which might no longer be possible. This is avoided by deactivating the *Drive Backward* after a certain time. Afterwards, a path is planned from the current point to the destination and the robot moves forward.

8. Experiments

In both control modes described in Section 3.3, ARTOS' ability to manoeuvre around obstacles is essential. If the collision avoidance failed, the robot easily could collide with a piece of furniture or the elderly person in the apartment, causing harm instead of comfort to the person. For a teleoperator, controlling the robot by only clicking on a target location in a map additionally requires the mapping and navigation systems to operate properly. In the operation mode that maintains the highest degree of privacy – the fully autonomous mode – no operator shall be needed for the robot to be employed. This necessitates a powerful combination of anti-collision, mapping, and navigation systems.

Figure 13. The main obstacles in the living room that ARTOS encountered when driving from the entrance to the kitchen.

To evaluate the performance of the overall system, numerous experiments have been conducted in the assistive apartment described in Section 1 (see Fig. 3). Details of such experiments can be found in [17] and [28]. Two significant examples with respect to naviga-

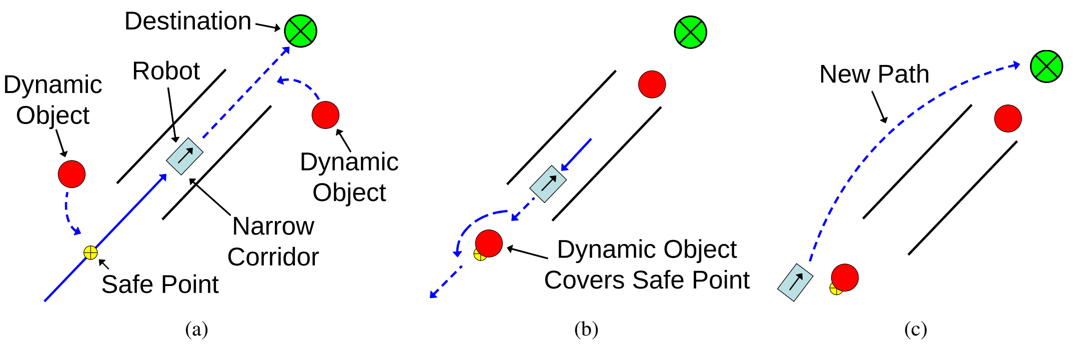


Figure 12. Scenario of Backing off: (a) While approaching the destination, the robot marks the *safe point*. (b) It backs off from the first obstacle and drives around another obstacle blocking the *safe point*. (c) The time for backing off has expired and the robot has planned a new path.

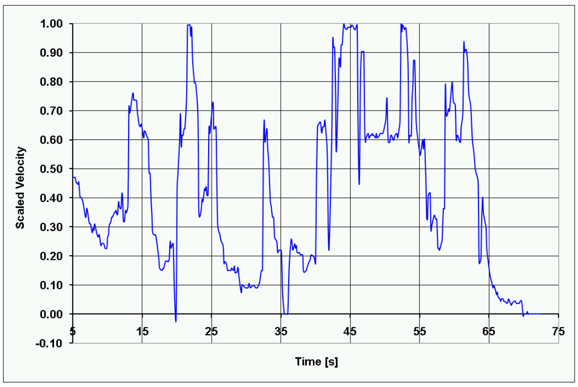


Figure 14. ARTOS' velocity during its tour from the entrance to the kitchen (scaled to $[-1.0; 1.0]$**).**

tion shall be presented here. At the beginning of the experiments, the robot was provided with an initial map containing *only* the walls of the rooms. Furniture and other dynamic objects standing or lying on the ground were not represented in the map and so the initially created paths led through obstacles. The challenge for the control system was to detect path obstructions and to find a way around them. In both the experiments presented here, the robot had to drive from the entrance to the kitchen. Fig. 13 shows the furniture in the living room through which ARTOS had to plan a path while navigating autonomously.

The robot's path during the first experiment is shown in Fig. 15. The gaps in the path are caused by the RFID-based pose corrections, which made the estimated pose "jump". A diagram of its velocity is shown in Fig. 14. As can be seen, the robot did not back off while following the path. This is an important result as it demonstrates that the mapping and obstacle avoidance components detected the obstructions so early that a way leading around them could be calculated before the robot got stuck.

In the second experiment, the robot took a detour before entering the kitchen. This was caused by the collision avoidance system which detected the door and made the robot turn in the wrong direction. As can be seen in Fig. 16, the navigation system adapted to the new situation and led the robot along a curve back to the correct course.

Finally, an emergency situation was created in the assistive apartment, where a man fell down on the floor and the assistive system detected the fall (see Fig. 17). After realising the situation, ARTOS – currently operating in the autonomous mode to ensure privacy of the person – sent an emergency call to the health care centre for urgent assistance. As mentioned in Section 3, the operator could then view the surroundings of the robot using the web interface. He had the two options to control the robot: either directly manoeuvring the robot using the joystick widget in the web interface, or activating the semi-autonomous mode by clicking on the 2D map of the apartment.

In the current scenario, the operator initially used the semi-autonomous mode to reach the human being and the joystick mode to have a better view of the incapacitated person. The remote person was able to focus the PTZ camera onto the fallen person to see his condition. Afterwards, the medical person established ARTOS' audio communication link to talk to the fallen person in order to find out how he was feeling and to assure him that medical help was on its way.

On the left side of the GUI visible in Fig. 17, the data yielded by ARTOS' laser scanner and ultrasonic sensors is displayed. This widget can also be used to

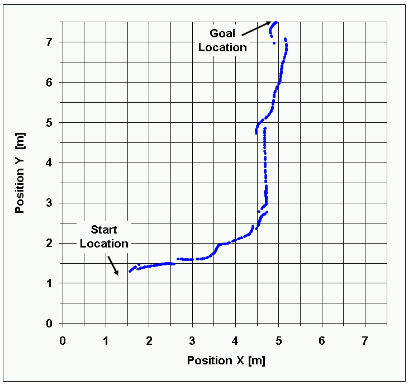


Figure 15. The path of the robot from the entrance to the kitchen during Experiment 1

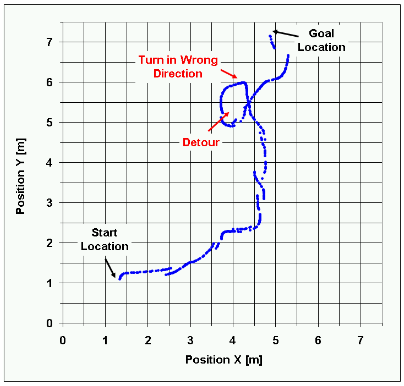


Figure 16. The path of the robot from the entrance to the kitchen during Experiment 2

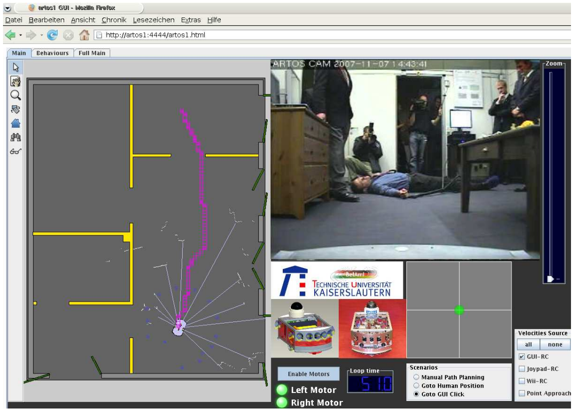


Figure 17. Teleoperation in an emergency scenario

display the grid map generated by the robot. It is important to mention here that this data, as well as the video image on the right, are *only* displayed and *not* stored in any way on the operator's computer. This again is in accordance with the approach keeping as much data as possible on the robot and only transferring (not storing) data that are needed in special, precisely-defined situations.

As visible in Fig. 17, the experiment was witnessed by a group of dignitaries. ARTOS was able to navigate between those persons without any collision and successfully established the communication setup after realising the situation.

9. Security and Privacy

Being an elderly care robot, security and privacy of the inhabitant of the house is one of the concerns for the development of ARTOS. It is a system in which security measures have been taken to secure the data. As mentioned in Section 3, ARTOS is able to communicate with the remote caregivers using wireless Internet. To ensure that only the caregivers may be able to use the information from ARTOS, certain steps have been taken. It is worth mentioning that the authors are not focusing on the aspect that any government agency gets access to the robot and hence access to the private information about the persons. This paper is focused on developing a secure robot for elderly people, which is secured from conventional hacking techniques and remote agents that might harm the privacy of the people.

Following are some of the measures taken to protect the information on the robot.

9.1. Encrypted Data

Firstly, to ensure privacy of the human being, no information about the daily schedule of the person is kept on the robot. Secondly, the images and videos that are captured using the camera are sent to the caregiver person only in case of an emergency situation and are not stored on the system.

To ensure that data on the disk, such as maps of the environment etc, are encrypted, the authors use TrueCrypt[13]. TrueCrypt creates an encrypted file which can be mounted as a device and all the operations can be performed as a normal drive. All the encryption and de-

[13] http://www.truecrypt.org

cryption is done on-the-fly and therefore cause no delay in computation. TrueCrypt has been configured to use three ciphers in a cascade operation in XTS[14] mode for encryption and decryption. These three ciphers are Advanced Encryption Standard (AES), Twofish and Serpent. Each block of information is first encrypted with Serpent using a 256 bit key. Then it is encrypted with Twofish using a 256 bit key and finally with AES using a 256 bit key. Independent keys are being used for these operations. (More information on different configurations and setup could be found at TrueCrypt website).

Storing encrypted information on ARTOS ensures that only relevant and authorised people have access to the stored data.

9.2. Network Security

Since ARTOS is reachable via wireless Internet, securing the network is the primary task that could prevent any illegal access to the robot. Therefore, the following actions have been taken.

9.2.1. Use of SSL. Secure Socket Layer (SSL) encrypts the segments of network connections at the Application Layer to ensure secure end-to-end transit at the Transport Layer. The Transport Layer Security (TLS) protocol allows client/server applications to communicate across a network in a way designed to prevent tampering and listening. It also provides endpoint authentication and communications confidentiality over the Internet using cryptography. TLS provides RSA security with 1024 and 2048 bit strengths.

In typical end-user/browser usage, TLS authentication is unilateral, that means, only the server is authenticated (the client knows the server's identity), but not vice versa (the client remains unauthenticated or anonymous). TLS also supports the more secure bilateral connection mode, in which both ends of the "conversation" can be assured with whom they are communicating. This is known as mutual authentication, or 2SSL. Mutual authentication requires that the TLS client-side also holds a certificate.

When a TLS or SSL connection is established, the client and server negotiate a CipherSuite, exchanging CipherSuite codes in the client hello and server hello messages, which specifies a combination of cryptographic algorithms to be used for the connection. The key exchange and authentication algorithms are typically public key algorithms.

Based on the security provided by TLS and SSL, all the network communication on ARTOS uses SSL. The server on the PTZ camera is configured to issue self-signed digital certificates to authenticate the user accessing the camera. The RSA algorithm (an algorithm for public-key cryptography) has been employed to encrypt the public-key. The transmission of the video stream is carried over a secure HTTPS channel which prevents any kind of tampering or listening.

The web server on the computer of the robot is also SSL based. Over here client authentication is also required. All the information for authentication of the client is first encrypted at the client side and then transferred over the Internet and finally after reaching the robot, it is decrypted and used. In this way it is assured that only the authorised operator is accessing the information from ARTOS.

The network security can be enhanced by establishing a Virtual Private Network (VPN) and installing firewalls to limit access to the robot.

9.2.2. Use of Skype. The authors have used the third party telephonic conversation tool, Skype[15]. The main reason for the choice was again the security and privacy of the elderly person. The information of logging into Skype using user-name and password is sent using Secure Socket Layer (SSL). Hence all the information is encrypted before it leaves the computer and can only be decrypted by the Skype server. Skype uses the AES algorithm with 256 bit encryption which has a total of $1.1x10^{77}$ possible keys. Skype also uses 1024 bit RSA to negotiate symmetric AES keys. User public keys are certified by the Skype server at log-in using 1536 or 2048-bit RSA certificates.

To add a level of security, automatic addition of users in Skype is not permissible on ARTOS. Similarly, calls and chat requests from unknown users are not allowed. Moreover, Skype is configured not to accept any invitation or files from users that are not added to the Skype list.

In case the security of Skype is compromised in any way, although the chances are low, the hacker could get access to the Skype server effecting the privacy of the elderly person, but there is no way the robot can be controlled by the illegal user to cause any harm.

10. Conclusion and Future Work

In this paper, an approach was described for improving safety but maintaining privacy of a person in assistive environments. The approach presented is only a first step towards a full-featured personal robot that shall unobtrusively monitor the user, respect the privacy, maintain a certain level of security and only call for help

[14] XTS: XEX-based Tweaked CodeBook mode with CipherText Stealing

[15] http://www.skype.com

if it detects a possible medical emergency. The focus was laid on the mobile system that replaces many monitoring devices whose deployment would tremendously reduce acceptance of such an environment by the inhabitants. The assisted teleoperation for emergency evaluation was used as an application scenario. Technical solutions regarding usability, deployability, security, and privacy were depicted. The concepts of the robot's navigational subsystems and the sophisticated teleoperation have been described in detail.

Up to now, especial tests revolving around the function of the navigational components have been evaluated. Future work will focus on real-life tests with the target group, with care institutions and with first-responders. The results of these tests will enable to define the privacy level acceptable by the elderly people.

To ensure the authentication of the client accessing the robot, 2SSL will be implemented which not only requires the server to have an SSL certificate but also demands that the client should also have a similar certificate for mutual authentication. The Java implementation allows a secure connection to the machine using 2SSL. Attempts for hacking the system and decrypting the information on the computer of the robot will also be launched to thoroughly test the security of the contents on the robotic system.

As has been mentioned, there is no guarantee that the elderly person wears the RFID-equipped shoe that would enable the robot to estimate the location of the person in the environment. Therefore, integrating a methodology for predicting the position of the human being in the environment based on a probabilistic belief of the presence of the human at a particular time is required to be developed.

Finally to enhance the state of privacy, interventions from the care-givers should be limited. This can be achieved by enhancing the capabilities of the robot to assess the situation and allow any involvement of an external person in case of a real emergency.

Acknowledgements

The authors gratefully acknowledge the funding of Syed Atif Mehdi by the Higher Education Commission of Pakistan and the DAAD, Germany and the funding of Christopher Armbrust by the PhD Programme of the University of Kaiserslautern's Department of Computer Sciences. Also, the authors want to thank Fraunhofer IESE in Kaiserslautern for cultivating the Assisted Living Lab.

References

[1] R. Cucchiara, A. Prati, and R. Vezzani, "A multi-camera vision system for fall detection and alarm generation," *Expert Systems*, vol. 24, no. 5, pp. 334–345, November 2007.

[2] W. L. Zagler, P. Panek, and M. Rauhala, "Ambient assisted living systems - the conflicts between technology, acceptance, ethics and privacy," in *Assisted Living Systems - Models, Architectures and Engineering Approaches*, ser. Dagstuhl Seminar Proceedings, A. Karshmer, J. Nehmer, H. Raffler, and G. Tröster, Eds., no. 07462. Dagstuhl, Germany: Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2008.

[3] G. Abowd, A. Bobick, I. Essa, E. Mynatt, and W. Roger, "The aware home: Developing technologies for successful aging," in *Procceedings of the Workshop on Automation as a Care Giver at the American Association of Artificial Intelligence (AAAI)*, Alberta, Canada, July 2002.

[4] S. Intille, K. Larson, and E. M. Tapia, "Designing and evaluating technology for independent aging in the home," in *International Conference on Aging, Disability and Independence (ICADI)*, Washington DC, USA, December 2003.

[5] J. Nehmer, A. Karshmer, M. Becker, and R. Lamm, "Living assistance systems - an ambient intelligence approach," in *Proceedings of the 28th International Conference on Software Engineering (ICSE)*, Shanghai, China, May 20-28 2006.

[6] P. Deegan, R. Grupen, A. Hanson, E. Horrell, S. Ou, E. Riseman, S. Sen, B. Thibodeau, A. Williams, and D. Xie, "Mobile manipulators for assisted living in residential settings," *Autonomous Robots, Special Issue on Socially Assistive Robotics*, vol. 24, no. 2, pp. 179–192, February 2008.

[7] F. Michaud, P. Boissy, H. Corriveau, A. Grant, M. Lauria, D. Labonté, R. Cloutier, M.-A. Roux, M.-P. Royer, and D. Iannuzzi, "Telepresence robot for home care assistance," in *AAAI Spring Symposium on Multidisciplinary Collaboration for Socially Assistive Robotics*, Palo Alto, USA, March 2007.

[8] A. Tapus, M. Mataric, and B. Scassellati, "The grand challenges in socially assistive robotics," *Robotics and Automation Magazin*, vol. 14, no. 1, pp. 35–42, 2007.

[9] D. Labonte, F. Michaud, P. Boissy, H. Corriveau, R. Cloutier, and M. Roux, "A pilot study on teleoperated mobile robots in home environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, October 9-15 2006, pp. 4466–4471.

[10] T. van der Zant and T. Wisspeintner, "Robocup@home: Creating and benchmarking tomorrows service robot applications," in *Robotic Soccer*, P. Lima, Ed. Vienna, Austria: Itech Education and Publication, December 2007, no. ISBN: 978-3-902613-21-9, ch. 26, pp. 521–528.

[11] Q. Wang, W. Shin, X. Liu, Z. Zeng, C. Oh, B. K. Alshebli, M. Caccamo, C. A. Gunter, E. L. Gunter, J. Hou, K. Karahalios, and L. Sha, "I-living: An open system

architecture for assisted living," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Taipei, Taiwan, October 2006.

[12] Y. Ouyang, Y. Xu, Z. Le, G. Chen, and F. Makedon, "Providing location privacy in assisted living environments," in *PETRA 08: Proceedings of the 1st international conference on PErvasive Technologies Related to Assistive Environments*. New York, NY, USA: ACM, 2008, pp. 1–8.

[13] T. Denning, C. Matuszek, K. Koscher, J. R. Smith, and T. Kohno, "A spotlight on security and privacy risks with future household robots: attacks and lessons," in *Ubicomp '09: Proceedings of the 11th international conference on Ubiquitous computing*. New York, NY, USA: ACM, September 30 - October 3 2009, pp. 105–114.

[14] J. Koch, C. Armbrust, and K. Berns, "Small service robots for assisted living environments," in *VDI/VDE Fachtagung Robotik*, Munich, Germany, June 11-12 2008.

[15] J. Koch, J. Wettach, E. Bloch, and K. Berns, "Indoor localisation of humans, objects, and mobile robots with RFID infrastructure," in *7th International Conference on Hybrid Intelligent Systems (HIS07)*, Kaiserslautern, Germany, September 17-19 2007, pp. 271–276.

[16] M. Proetzsch, T. Luksch, and K. Berns, "The behaviour-based control architecture iB2C for complex robotic systems," in *Proceedings of the 30th Annual German Conference on Artificial Intelligence (KI)*, Osnabrück, Germany, September 10-13 2007, pp. 494–497.

[17] C. Armbrust, J. Koch, U. Stocker, and K. Berns, "Mobile robot navigation support in living environments," in *20. Fachgespräch Autonome Mobile Systeme (AMS)*. Kaiserslautern, Germany: Springer-Verlag, October 2007, pp. 341–346.

[18] J. Koch, M. Reichardt, and K. Berns, "Universal web interfaces for robot control frameworks," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nice, France, September 22-26 2008.

[19] P. Fiorini and R. Oboe, "Internet-based telerobotics: Problems and approaches," in *International Conference on Advanced Robotics (ICAR)*, Monterey, USA, 1997.

[20] M. Proetzsch, T. Luksch, and K. Berns, "Development of complex robotic systems using the behavior-based control architecture iB2C," *Robotics and Autonomous Systems*, vol. 58, no. 1, pp. 46–67, 2010.

[21] K. Finkenzeller, *RFID Handbook: Radio-Frequency Identification Fundamentals and Applications*. New York: Wiley, 2000.

[22] D. Henrici, *RFID Security and Privacy - Concepts, Protocols, and Architectures*, ser. Lecture Notes in Electrical Engineering, Springer. Springer Berlin, 2008, ISBN: 978-3-540-79075-4.

[23] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proceedings of the 18th National Conference on Artificial Intelligence and 14th Conference on Innovative Applications of Artificial Intelligence*. Edmonton, Alberta, Canada: AAAI Press, California, USA, August 2002, pp. 593–598.

[24] R. Ouellette and K. Hirasawa, "A comparison of SLAM implementations for indoor mobile robots," *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007 (IROS 2007)*, pp. 1479–1484, October 29-November 2 2007.

[25] J. Wettach and K. Berns, "3D reconstruction for exploration of indoor environments," in *Autonome Mobile Systeme 2007*, ser. Informatik aktuell, K. Berns and T. Luksch, Eds. Kaiserslautern: Springer-Verlag, October 18-19 2007, pp. 57–63.

[26] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," in *IEEE Transactions of Systems Science and Cybernetics*, vol. 4, no. 2, July 2 1968, pp. 100–107.

[27] S. Quinlan and O. Khatib, "Elastic bands: Connecting path planning and control," in *Proceedings of IEEE Int. Conference on Robotics and Automation*, Atlanta, 1993, pp. 802–807.

[28] S. A. Mehdi, C. Armbrust, J. Koch, and K. Berns, "Methodology for robot mapping and navigation in assisted living environments," in *PETRA '09: Proceedings of the 2nd International Conference on PErvasive Technologies Related to Assistive Environments*, no. ISBN: 978-1-60558-409-6. Corfu, Greece: ACM, New York, NY, USA, June 9-13 2009.