The ever increasing population of elderly people in our societies requires development of robotic platforms that can provide services to them at their own homes. In order to serve the person, the most essential task is to find the human in the home environment. There are two components in finding the person. The first is to autonomously traverse to a location where human is present, and the second is to detect the human using various sensors installed on the robot. The state-of-the-art approaches do not focus on autonomously identifying locations where it is most probable to find the person. Moreover, the sequence of navigating these locations can be used to determine the optimization of the search process.

This thesis focuses on the fundamental task of identifying locations in the home environment, which are suitable for the robot and can be used for finding the elderly person. The methodology adapts itself according to the daily routine of the person to traverse these locations; hence, optimize the human search process.

The developed methodology has been tested in simulation and in real environment using an autonomous mobile robot, ARTOS. The experimental results show that the developed approach enables the robot to efficiently and effectively search the human being.
Using the Human Daily Routine for Optimizing Search Processes using a Service Robot in Elderly Care Applications

Syed Atif Mehdi

Vom Fachbereich Informatik der Technischen Universität Kaiserslautern zur Verleihung des akademischen Grades Doktor-Ingenieur (Dr.-Ing.) genehmigte Dissertation.
Zur Begutachtung eingereicht am: 5. Februar 2014

Vorsitzender: Prof. Dr.-Ing. Stefan Dessloch
Erster Berichterstatter: Prof. Dr. rer. nat. Karsten Berns
Zweiter Berichterstatter: Dr. Mian Muhammad Awais
Dekan: Prof. Dr. rer. nat. Klaus Schneider

Zeichen der TU im Bibliotheksverkehr: D 386
Acknowledgements

All praises be to Allah, the most gracious and the most merciful.

First of all, I am extremely thankful to Prof. Dr. Karsten Berns for supervising me and having valuable discussions about the work. I have learned a lot from him. I am especially impressed with his deep insight in the field of robotics. I also like to thank Dr. Mian Muhammad Awais for being reviewer of this thesis.

I would like to thank members of Robotics Research Lab for their support and maintaining a pleasant atmosphere for research and generating new ideas. Special thanks to Jan Koch who started ARTOS project and worked hard to keep it running. Thanks goes to my office roommates Christopher Armbrust, Jochen Hirth and Gregor Zolynski, specially Christopher and Gregor for reviewing some parts of this thesis. I am also thankful to Mrs. Broschart, Mrs. Stumpf, and Mrs. Sakdapolrak who made all the administrative tasks easier for me.

I would like to thank Dr. Abubakr Muhammad for valuable discussions and comments regarding various aspects of the thesis. I am also thankful to anonymous reviewers of various journals, conferences and workshops who reviewed my papers and gave valuable suggestions in polishing my work. I am also thankful to IESE Fraunhofer, Kaiserslautern for permitting the use of assisted living lab and keeping it free for many days at multiple occasions, specially during the final experiments which took several weeks.

I am extremely grateful to Higher Education Commission of Pakistan (HEC) and Deutscher Akademischer Austausch Dienst (DAAD) for providing me scholarship and an opportunity to work in a very productive environment. A note of thanks goes to Pakistani community in Kaiserslautern for making every event a memorable one.

Much appreciation to my friends Junaid Haroon Siddiqui, Yasir Niazi Khan, Adil Khan, and Kh. Muhammad Umar Suleman. Their sincerity and friendship is an unmatched gift.

Above all, I am greatly in debt to my parents and my siblings for their continuous support and guidance in every matter of my life specially my wife who supported me during tough days and encouraged me to complete my PhD. May Allah reward them the best. Amen.
Abstract

The number of elderly people in our societies is increasing steadily. Most of the times, these elderly people like to live alone in their homes which gives them a sense of ownership and control over their lives. This growing trend has led many researchers to develop robotic platforms to assist elderly people in their daily life activities at their homes. To serve an elderly person at home, it is necessary that the mobile robot approaches the human to initiate an interaction or inquire about any services. Mostly it is done by pre-defining some locations in the home environment and the robot is called upon by the elderly person at these locations. In such scenarios, the robot behaves as a reactive device which waits for a signal to navigate to a specific location.

The aim of this thesis is to develop a methodology which enables an autonomous indoor mobile robot to pro-actively search an elderly person in a home environment. Besides being helpful in scenarios like initiating communication, or reminding an event, such pro-active approach is essentially required in scenarios of autonomously detecting an emergency situation where the person might not be able to call the robot for help. For an effective search in the environment, the developed methodology is inspired from human behavior during the search of lost objects or fellow humans. Their search approach relies on several factors. Among those, the most prominent one is that they learn from their experience where a fellow human is usually seen at a particular time of the day. This learned knowledge makes the search process much faster as compared to searching randomly or everywhere in the living environment.

The proposed methodology for an effective robotic search is based on the same concept. In order to develop the cognition of presence of person in the environment, several locations are computed by the robot which gives best possibility to observe the environment. The robot, then, autonomously navigates to these locations and gather information of human presence. Initially, the robot does not have any knowledge of presence of the person at these places, therefore, all locations are treated equally during the learning phase of the search process. The situation changes gradually as the robot finds the person at some locations at certain times of the day. The next search process utilizes this information and the robot gives priority to these locations.

The experiments in real environment as well as in simulation show that the mobile robot, ARTOS, autonomously and pro-actively initiates the search process and traverse to different locations in the home environment to find the human. The results show that with the passage of time, the robot successfully learns the daily routine of the person and uses this information to expedite the search process. A promising success percentage of more than 90% of finding the human shows the effectiveness of the proposed methodology. For future directions, the developed methodology can be used to monitor an elderly person and to
determine any anomaly in the daily routine of the person. These anomalies can be used to predict a possible emergency situation.
## Contents

1 Introduction ................................. 1
   1.1 Technical Solutions for Elderly Care .......... 4
   1.2 Objectives ............................... 6
   1.3 Document Structure ....................... 7

2 Mobile Robots for Serving Humans .......... 11
   2.1 Commercial and Prototype Robots for Helping Humans ........ 11
      2.1.1 Robots for Home Service ............... 13
      2.1.2 Transportation Robots ................. 15
      2.1.3 Robots for Facilitating Communication .... 16
      2.1.4 Robots for Monitoring Purposes in Home Environments .... 18
   2.2 Control Architectures for Service Robots .......... 22
   2.3 Discussion ............................... 25
      2.3.1 Functional Requirements for an Elderly Care Robot .... 25
      2.3.2 Non-Functional Requirements for an Elderly Care Robot ... 26

3 Fundamentals ............................... 29
   3.1 Autonomous Robot ARTOS ................... 29
      3.1.1 Design and Mechatronics of ARTOS .......... 29
      3.1.2 Sensor Systems on ARTOS ................ 33
      3.1.3 Mapping on ARTOS ..................... 35
      3.1.4 Localization of ARTOS in the environment .... 38
      3.1.5 Navigation in ARTOS ................... 40
   3.2 Assisted Living Lab ...................... 46
   3.3 Simulated Environment .................... 49

4 Design Concept for Searching Elderly People at Home .... 51
   4.1 Human Behavior for Searching ............... 51
      4.1.1 Finding Lost Objects ................... 52
      4.1.2 Finding Fellow Humans ................. 54
   4.2 Concept of a Robotic Solution ................ 54

5 View Points for Finding Human ............. 57
   5.1 Placement of an Observer for Area Coverage .... 58
   5.2 Locations for Finding Human ............... 62
   5.3 Experiments and Results ................... 67
   5.4 Discussion ............................... 73
# 6 Human Location Estimation Based on Daily Routine

## 6.1 Approaches for Human Localization

- 6.1.1 Human Localization by Installing Sensors in the Environment  
- 6.1.2 Wearable Sensor Based Human Localization  
- 6.1.3 Finding Humans using Robots

## 6.2 Analysis of Life of Elderly People

## 6.3 Concept for Human Search using Daily Routine

- 6.3.1 Learning Daily Routine of an Elderly Person

## 6.4 Modeling of Human Search as Markov Decision Process

- 6.4.1 Behavior for Human Daily Routine based Search
- 6.4.2 Behavior for Last Known Location based Search
- 6.4.3 Fusion of behaviors

## 6.5 Human Detection

## 6.6 Experiments and Results

## 6.7 Discussion

# 7 Experiments

## 7.1 Experimental Setup
- 7.1.1 Simulation of Human
- 7.1.2 Data Sets

## 7.2 Evaluation of View Points

## 7.3 Evaluation of Learning Human Routine

## 7.4 Evaluation of Navigation Policies for Human Search

## 7.5 Overall Performance of Developed Methodology

- 7.5.1 Success Rate
- 7.5.2 Reliability of Searching Human

# 8 Conclusion

## 8.1 Future Directions

# A Human Probabilities and Learned Routine

# B Implementation

- 8.1 Modular Controller Architecture
- 8.2 Integrated Behavior Based Control Architecture (iB2C)

# C Simulation and Visualization Environment

# Bibliography
1. Introduction

The overall demographic situation of the world shows an unprecedented increase in the elderly population, aging 60 years or more. The current growth rate of the elderly population in the world is 2.6 percent per year, which is significantly more than the growth rate of the population as a whole, which is 1.2 percent per year [United-Nations 09]. This difference of growth rates inherently implies that there will be less people in the working age and more elderly people who will depend on others for performing simple daily activities. According to the report prepared by Berlin Institute for Population and Development [Lehr 07], about one tenth of the world population, 672 million people, are older than 60 years.

The growth of elderly population is much more dominant in developed countries, like Germany, France, Japan etc. The demographic situation in such countries can be understood by the report presented by [United-Nations 09], which shows the number of elderly people, aged more than 60 years, in such countries has already exceeded the number of children, aged under 15 years, in 1998. For any country it is desirable to have a large population of youth or younger generation, which will provide the basis of the working economy. Thus, the population should be a pyramid like structure with a greater percentage of the population at the base as youth and less number of elderly people on the top. The effect of increase in elderly population can be seen by looking at the population of Germany. Back in 1950, the population structure used to be a pyramid shape which now is transforming into a mushroom like structure as can be seen in Figure 1.1 [Snetterlin 08].

The issue of growth in elderly population is not specific only to the developed countries. It is also rising in the developing countries, where it could get even worse as no preemptive measures have been taken due to lack of facilities and awareness. According to the estimates presented by the United Nations Population Fund (UNFPA), until the year 2050, the average number of children per woman is going to reduce from 5 to 2.6 in the 50 least developed countries [Lehr 07]. Figure 1.2 shows that it took 115 years in France to double the population of people aged 60 or older from 7 to 14 percent. In contrast, in Thailand, it will take only 20 years to reach the same level of percentage [Lehr 07]. The consequence of this rapid growth will be, as projected by World Population Ageing 2009
Figure 1.1: Population trend in Germany - from pyramid to mushroom [Suetterlin 08]. The pyramid structure shows a larger younger generation and fewer older people in the society whereas the mushroom structure shows a larger group of elderly people as compared to young people. Image has been modified for better viewing.
Elderly population is speedily doubling from 7 to 14 percent in many countries. In developing and under-developing countries it is taking less time, as compared to developed countries, in changing the numbers thus stirring more challenges for these countries in coming years. Reproduced from [Lehr 07]

United-Nations 09, that by 2050 one fifth of the population in developing countries will consist of old people.

Among many others, there are two important factors that define the increase in old-age population. Firstly, the increase in life expectancy due to better health care facilities, hygienic environments, and emphasis on physical and mental exercise. This factor is more prominent in developed countries. Secondly, low fertility rates have reduced the number of children per woman, which is very much the case in developing countries. Due to these factors, the relative population of elderly people is rising in the communities.

Aging itself is a natural phenomenon, but there are several changes that happen to the body of a person as he grows older. With the passage of time, changes in nervous, cardio-pulmonary, and musculo-skeletal systems occur. By the age of 70, the mass of muscles decreases by 40%, especially in arms and legs, thus contributing to weakness and fatigue [Koncelik 03].

With the increase in age, poorer senses of balance and kinesthesia affect spatial awareness causing change in gait and motion patterns. Walking speed, stride length, upper-lower extremity synchrony and arm swing amplitude may contribute to loss of firm and balanced foot grip and thus causing increased risk of falling [Crews 05]. According to the report entitled “The state of Aging and Health in America” by Centers for Disease Control and Prevention (CDC), USA, [CDC 13], every year one out of three adults aged 65 year or more falls. Falling is the most common cause of hospital admissions for injury among the older people. It is the leading cause of injury death in older adults. Fall related injuries, especially fractures, cause significant mortality, disability, and loss of independence.

Due to muscular weakness and danger of falling down while performing daily life activities, elderly people cannot be left alone on their own. It is required that someone is always
there with them to look after them and help them in performing simple tasks like moving from one room to another.

According to a report by the US Census Bureau, the number of elderly people in the world will exceed the number of children under 5 within 10 years [Kinsella 09]. This change will place greater demands on shrinking number of younger caregivers. According to the Berlin Institute for Population and Development, in Germany, currently about one third of all people aged 80 or more are dependent on caregivers. Statistically this number will be doubled by 2030 [Lehr 07].

In economics the term age dependency ratio is used to identify the age-population ratio of non-labor force to labor force. It is used to measure the pressure on productive population and is defined as in Equation 1.1, which simply implies that the lower the number, the less pressure on the working class of the country.

\[
\text{Age Dependency Ratio} = \frac{\text{number of people aged 65 and over}}{\text{number of people aged 15-64}} \times 100.
\]  

According to the Berlin Institute of Population and Development [Lehr 07], the current old age dependency ratio of Germany has turned out to be 26.5 and this ratio will keep on increasing up to 39.3 in 2025 and 55.8 in 2050. With such high ratios, the changes in the methods of elderly care are mandatory.

Elderly people 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 an unattended situation. Specially, the Western societies with very high life expectancy and small number of family members cannot provide support in daily activities of elderly person. This results in development of professional elderly care homes where old people live with their age fellows and are supported and supervised by a well qualified nursing staff. In an effort to facilitate old people at these elderly care homes, the total costs of such facilities have started rising sky high.

Several surveys show that not all elderly people like to move to elderly care homes in the early part of their old age. They desire to live in their own homes but require that similar care services are provided at their homes. This trend not only reduces the ever-growing costs of the elderly care homes but also fulfills the desire of elderly people being independent at their homes.

1.1 Technical Solutions for Elderly Care

With the increase of elderly population, it is becoming necessary to use modern technologies to maintain standard of living and to provide elderly people with better health care services at their homes. Due to the limited number of nursing staff and increasing costs of such facilities, many research groups focus on modern technologies to assist elderly people at their homes. The aim is to lower the nursing costs and increase the quality of life of persons with disabilities [Hähnel 04, Kulyukin 05, Montesano 06, Cherubini 07].

The technological advancement can make it possible to detect an accident to the elderly person at home or even report the accident immediately to the caregivers. Several research
groups have focused on developing wearable sensor systems. These include dresses, shoes, watches, etc. with embedded sensors that can monitor human activities and body conditions such as blood pressure, heart beat etc. These wearable sensors perform pretty well in monitoring the humans and can transmit the information to a monitoring server for further processing.

There is also a branch of researchers who are working hard on developing sensor systems for home automation. These include developing cameras and other sensors to be installed in home environments. These sensors monitor the human activity and log the patterns of human daily routine. Several methodologies have been developed for detecting if the human being has fallen down or is unable to perform certain activities, which are then reported to the health care staff for immediate assistance.

Besides above-mentioned methods for supporting and monitoring elderly persons at home, robots are also being used to monitor the aged persons. Domestic service robots are becoming useful, multipurpose, autonomous and reliable. Besides their use as household appliance, like robotic vacuum cleaners, many research groups are focusing on the development of robots for home monitoring and elderly care situations where they can interact with humans and objects in the real world. The goal of such robots is usually to investigate the needs of the elderly person, determine emergency situations that might have happened to the person and serve as a communication channel between the inhabitant of the house and the caretakers. The added benefits of using robots, besides others, are that they can

- help the elderly person in performing different tasks at home,
- be a companion to the elderly person,
- act as an interaction partner,
- be tele-operated in case of an emergency situation.

To achieve any of the above-mentioned benefits from an autonomous mobile robot, it is necessary for the robot to search the human in the environment. In case the robot does not have the capability to find the human, it is not possible to provide any service to the person. The search process using an autonomous mobile robot involves many challenges to be addressed, including perceiving work environment, evaluating the current location of the robot, determining a way from the current position to a specific destination in the environment, traversing to the destination safely and last but not least detecting the people at their favorite places.

Usually, a typical home environment is cluttered with obstacles, thus limiting the autonomous mobility of the robot and creating hindrances for the robot to perceive information of the surroundings. Furthermore, several areas can also be beyond the reach of the mobile robot. Such environmental situations require that the robot navigates to the important locations in the environment from where it can efficiently find the human being.

Some researchers have taken up the challenge to build an autonomous mobile platform to facilitate elderly people. A few of the robotic solutions are also being developed to carry out the routine tasks of the caregivers at old-age homes or hospitals. These robots have been developed according to the specific needs of the staff at these institutes. Examples of such requirements are helping old people to move from one room to another [Pollack 02]
Montemerlo 02, carrying carts that human workers used to carry [Bloss 11], and delivering objects to elderly people [Graf 09a]. These solutions are customized according to the requirements of the general elderly people. However, there is still a need to develop autonomous robots that are personalized in a sense that they can adapt themselves according to the needs and routines of one specific person at home.

1.2 Objectives

In real-life scenarios, it can be easily observed that humans are quite capable of moving from one place to another in their home environments. One can argue that humans have many degrees of freedom and can move freely in the environment or climb up objects, but to perform complex tasks, humans rely on their perception of the environment and the objects in the environment. Even if a perfect two legged robot, with all the possible degrees of movements as humans have, is developed, it will not be possible for the robot to perform complex tasks without having a certain level of perception. Specially, when it comes to searching objects or fellow humans in the home environment, the humans perform the task very successfully mainly due to a variety of search strategies and higher level of perception system. On the other hand, current robotic solutions are lacking behind in effectively searching a person in the environment. Therefore, this thesis explores the human behaviors for performing the complex task of searching other humans in the home environment and attempts to generate similar behaviors in an autonomous mobile robot.

Considering the complex aspects of human behaviors and nature of the environment, the hypothesis can be stated as follows

The daily routine of an elderly person can be learned by an autonomous mobile robot to efficiently search the person within the unstructured home environment. For the purpose of human searching, identification of good viewing locations in the environment and finding the human can be autonomously performed by the robot without any intervention from an operator or a supervisor of the robot.

Having outlined the hypothesis, the following aspects have been investigated in this thesis.

Determination of Locations in the Environment Suitable for Observation

The first and the foremost task is to perceive the home environment using sensor system installed on the mobile robot. Once the environment is perceived in a suitable manner, the challenge of determining suitable locations for viewing humans arises. For humans, locations for observation in the home environment do not matter as humans can easily walk from one location to another or climb up the furniture to see behind or above them. But in case of a mobile robot, these locations are very important as these locations provide a good possibility to observe the home environment which may be filled with obstacles that humans do not realize as obstacles. These locations can be anywhere in the environment. However, for determining or marking some locations as suitable locations for viewing or observing the home environment, the constraints of the mobile robotic platform need to be kept into consideration.
Estimating Human Location in the Home Environment One of the key challenges in finding the human being in the home environment is to estimate the location of the person at a particular time of the day. It is a general observation that old people tend to have a very organized daily life. For example, they take breakfast at a particular time and perform other activities according to a well-defined schedule. In this work, this observation has been taken into account and the presence of a person is determined using the chances of him being at particular places in the home at different times of the day.

Simulated Environment for Validation For validating the implemented methodologies described above, a simulated environment has also been developed which includes a home-like scenario with furniture and different objects in the environment, a simulated human being that can move to different rooms to generate real-life-like situations and a simulated robot with the necessary sensor system and motion model.

The goals, having said easily, are quite hard to achieve. The challenges include choosing an accurate representation of the environment, determining the best locations in the environment for monitoring the person and last but not least estimating the location of the person and navigating to the required destination to find the person.

1.3 Document Structure

This thesis document is organized as follows:

Firstly, an overview of commercially available robotic platforms along with robots developed for research in the field of elderly care has been provided in Chapter 2. These robots are being in various settings and perform different domestic tasks. Some of these robots are also used in hospital environments or elderly care facilities where they are helping caregivers. This chapter also describes some technical details about these robots like size of the robot, capabilities, and sensor systems installed on the robot. The objective is to extract qualities or features that are expected from a robotic platform that can be used to provide services to an elderly person living alone in his apartment.

Chapter 3 focuses on the robotic platform (ARTOS) developed over the last 8 years at Robotics Research Lab for conducting research in the context of providing services to an elderly person in an home environment. It is a prototype platform to research and develop methodologies that can be used in mobile robots in various elderly care scenarios. This chapter provides an insight to the fundamentals that are required to understand this thesis containing technical details of the robot and sensors on the robot along with their measurement accuracies and ranges. This chapter also explains the process of generating map of the environment using the installed sensors and localization process to reliably and accurately determine the current location of the robot in the environment. Furthermore, components of autonomous navigation used in ARTOS and path planning in indoor environment has also been described. Besides providing details about the robotic platform, this chapter also focuses on the test environment which is a typical household environment specially developed for conducting research activities in the field of assisted living. Towards the end of the chapter a brief introduction to the simulation developed for performing experiments has been provided.
The emphasis of Chapter 4 is to explain the human behaviors when they are indulge in searching a lost object or a fellow human. This chapter also explains how humans start developing cognition about objects and humans and how this memory is used during the search process which results in various types of search strategies that are used by humans at different times. This insight into the complex strategies for searching objects has been used to develop a strategy for efficiently finding an elderly person in the home environment using an autonomous mobile robot. Towards the end, this chapter presents the design concepts of the developed methodology.

Following the footsteps of humans, a robotic platform requires to navigate from one location to another in order to find an elderly person in the home environment. Traditionally, this process requires human intervention and these locations are provided to the robot by operator or supervisor of the robot. Chapter 5 presents the argument that these locations can be determined by the mobile robot on its own without human interference. Naturally, some criterion are required to be identified that can provide the basis for formulating locations for searching the human. This chapter provides these criterion and methodology to autonomously determine the locations in the home environment. The experiments performed in the real home like apartment confirms the validity and effectiveness of the developed methodology and results have been provided at the end of this chapter.

After finding the suitable locations in the environment, the next step is to develop a perception of presence of an old person in home. As human beings keep on updating the information about the environment, a robotic solution should also be able to learn over a longer period of time and adapt itself according to the changes in daily routine of the elderly person. These aspects are more elaborated in Chapter 6 which also provides an overview of already developed methodologies used for finding or localizing a person in the home environment. The chapter focuses on aspects of life long learning in the robotic platform and using these memories to expedite the process of searching in the environment. The experiments performed in the real home apartment, showing promising results, have been provided at the end of the chapter.

The experiments and results in Chapter 5 and Chapter 6 are drafted from real world scenarios. In order to perform statistical analysis, experiments have to be performed repeatedly and for a longer period of time. To accomplish this task, a simulated environment with a human character depicting the behavior of a real human has been created. Details of this human character and statistical analysis of the results obtained by performing experiments in simulation have been explained in Chapter 7. This chapter also describes the overall performance of the developed approach of using an autonomous mobile robot to efficiently find an elderly person by learning daily routine of the person.

Chapter 8 summarizes the results and achievements of the thesis at hand and gives an outlook on the future work in the direction for applying the developed methodology in various scenarios and possible extensions for future research.

The data set used in experiments for learning human routine and results of experiments has been provided in Appendix A.

The implementation details and the framework used for controlling the robot are described in Appendix B. It elaborates the concept of “Modular Controller Architecture” and “Integrated Behavior Based Control Architecture” and provides an overview of some of the tools available for using them.
An extended description of the simulation framework has been provided in Appendix C. It also describes the creation of 3D scene for simulation of environment, robot and simulated character.
2. Mobile Robots for Serving Humans

The use of robots for performing various tasks is increasing day by day. Initially designed to work in the industry to help mankind, the robots are now penetrating in the homes of human beings. This trend can be seen in Figure 2.1 which depicts the increase in the number of robots from 2006 till the end of 2008. Though there is a steady growth in number of industrial robots, the increase in service robots is tremendously high and had already crossed the 7 million mark.

According to the International Federation of Robotics (IFR) [IFR 12], a service robot is a robot that operates semi- or fully autonomously to perform services useful to the well being of humans, excluding manufacturing operations. IFR further divides service robots into two categories: robots for professional use and robots for personal and domestic use. It is predicted that by 2016, there will be more than 10 million service robots living with the humans in their houses and helping them in various tasks, majority of which will be vacuum cleaning robots. Figure 2.2 shows the pattern of growth of service robots for domestic and personal use. The trends clearly indicate that people like to use robots for household tasks though robots for entertainment are also popular.

In the following, a general overview of service robot used for performing different activities as a help to humans has been provided. The focus is to look into some technical details of commercially available and research oriented robots and extract capabilities of these robots. This will lead to the discussion about requirements of a robot which can be used in indoor house environments for providing services to an elderly person living alone.

2.1 Commercial and Prototype Robots for Helping Humans

There is a variety of service robots that are being developed for helping people in their daily life. These robots have been specially designed and developed for performing predefined specific tasks. Although there is quite an active development in robots for performing outdoor activities but the main focus here is to elaborate development of robotic platforms
Figure 2.1: There is a steady growth in industrial robots but a much rapid growth in service robots. The numbers show that service robots have already crossed the 7 million units mark by 2008. Reproduced from [Guizzo 10].

Figure 2.2: Among the service robots, number of robots for personal use increased tremendously in 2011 and 2012 [IFR 12]. It is projected that during 2013 - 2016 household robots will exceed 10 million. During the same time, service robots for entertainment will be less than 7 million.
2.1. Commercial and Prototype Robots for Helping Humans

![Figure 2.3](image)

Figure 2.3: (a) Roomba (b) Kaercher RC (c) Naeto XV, are examples of robotic vacuum cleaners used in home environments. Images from [http://www.irobot.com/](http://www.irobot.com/), [http://www.avsale.ru/inf/9798/](http://www.avsale.ru/inf/9798/) and [http://www.neatorobotics.com/](http://www.neatorobotics.com/) respectively for performing tasks in complex indoor environments. These indoor tasks can be categorized into four major categories as follows.

- Home service tasks
- Transportation of objects from one place to another
- Enhancement of interaction between the person at home and a remote user by establishing a communication link
- Monitoring the person at home

In the following, an overview of robots used for performing above-mentioned tasks is given.

### 2.1.1 Robots for Home Service

Robots that perform tasks at home are the most common ones and their production and use is increasing day by day. In 2008, around 4.4 million service robots were sold for home applications. Among that, the largest share was taken by the vacuum cleaning robots that summed up around 1 million [Guizzo 10].

Robotic vacuum cleaners are the most cost effective and popular among all kinds of service robots. In order to keep down the costs of a single machine, usually these are equipped with very basic sensor systems and thus have a very limited sensing range which is sufficient for cleaning the floor. Though some of these robots have a kind of predefined strategy to navigate in home environments like starting from the middle and moving in spirals or follow walls and then move in rectangular manner towards the center of the room, but most of them follow a random path to traverse through the room.

Roomba by iRobot, see Figure 2.3(a), is equipped with bumper sensors to detect obstacles in the environment. Having a simple sensor system and limited computation, it is not capable of planning a path or learning from the environment. It determines its way by colliding obstacles. After collision with furniture and walls, it changes its course and turns to a different direction. Although Roomba has some predefined behaviors like follow the
Figure 2.4: Motion Pattern of (a) Roomba (b) Naeto. Roomba moves randomly in the environment and therefore sometimes misses a spot or move over a spot multiple times. Naeto on the other hand plans a strategy to vacuum the room and thus moves minimally over one spot. Images from: [http://signaltheorist.com/?p=91](http://signaltheorist.com/?p=91) and [http://www.i-robot.su/item.php?id=102](http://www.i-robot.su/item.php?id=102) respectively.

wall, move in spiral, its overall strategy to traverse the working space is completely random [Forlizzi 06]. Figure 2.4(a) shows the cleaning strategy of Roomba. It can be seen very easily that during random movements, the robot may vacuum the same area multiple times and it is equally possible that any part of the room may be left dirty and needs to be manually cleaned.

Kaercher RC, Figure 2.3(b), is also similar to the Roomba in the sense of its random movement pattern and detecting obstacles by colliding with the object sensed by the bumper sensor. Additionally, it is equipped with infrared sensors to detect stairways and thus can prevent itself from falling down. The infrared sensors are also used to guide the robot to autonomously dock to the charging station.

Naeto XV by Naeto Robotics, Figure 2.3(c), is equipped with a laser range finder, along with bumper sensors. It creates a map of the environment and then starts cleaning in a systematic manner. Based on the distance information, it calculates the trajectory and prevents hitting obstacles. It localizes itself in the environment using Simultaneous Localization and Mapping (SLAM). Figure 2.4(b) shows the cleaning methodology used by Naeto. As can be seen that a place is visited only once, hence resulting in a better and quicker cleaning of the environment.

There are many reasons for the increasing popularity of these vacuum cleaning robots. For example, they are performing the tedious job of cleaning the home, they can be easily programmed to perform the cleaning on specific timings and with the advancement in technology they are getting cheaper and cheaper. However, there are two main factors that outstand: the size of these robots and the fact that they are plug-and-play.

The small size of these robots makes them easier to carry from one place to another. They can easily move through closely placed furniture and also under furnitures like beds, tables, and chairs. The small size also makes them non-obtrusive to the people in the home.
2.1. Commercial and Prototype Robots for Helping Humans

The most significant feature of vacuum cleaning robots is the plug-and-play feature, which enables their owners to carry them to any place and use them. These robots do not require any changes in the infrastructure of the home environment and can perform their task with their limited sensing and computational capabilities.

As stated by [Forlizzi 06], “Simply put, homes are simply not designed to accommodate autonomous robotic technology — nor should they be. Rather, if autonomous mobile robots are to be used in the domestic environment the robots need to be designed to ‘artfully integrate’ with the structures and practices of the home”. Therefore, any autonomous mobile robot that should be used as an elderly care solution should be designed in a way that it blends in the environment easily. Moreover, the robot should not require changes in the already existing infrastructure of the home environment for its operations. It is also necessary that the robot should work as a plug-and-play unit and can easily perform all its required duties even if the environment is completely changed.

2.1.2 Transportation Robots

For indoor environments, like offices and hospitals, robots have been developed to help humans for transporting objects from one place to another. These robots are more complex than the vacuum cleaning robots in the sense that they have more sophisticated sensor systems and better strategies to navigate in the environment. These robots have pre-assigned destinations and can navigate to the desired location autonomously. The benefit of such robots is a significant reduction in transportation times and the increase in the amount of load that can be transferred using these robots.

The TUG, developed by Aethon\footnote{http://www.aethon.com}, is an automated system developed to transport supplies such as medication, linens, and food from one place to another in a hospital environment.
The robot moves through hospital corridors, elevators and departments at any time during the day to make either scheduled or on-demand deliveries [Bloss 11]. The robot itself weighs about 25 kg and can pull carts up to a total of 227 kg.

It is equipped with sonar sensors, infrared sensors and laser scanners to detect human beings in the environment. The map of the environment is provided to the robot for autonomous movement within the hospital environment. Using this map, the robot plans an optimum path from the source to the destination. In case, any unexpected obstacle is discovered in the desired path, the robot can modify and plan a new path to the destination. Radio Frequency Identification (RFID) tags can be used as reference points for the TUG to update the location accuracy. The robot drives close to walls, giving more space to human beings in the center of the passages and hallways.

The Swisslog RoboCourier Figure 2.5(b), is an autonomous mobile robot developed to dispatch and deliver specimens, medications, and supplies throughout the hospital. It can carry about 20 kg from one place to another. Once the robot is carrying the objects to be delivered, a person chooses the destination and the robot selects the most efficient route to deliver the materials by planning a path using electronic maps. The robot uses a laser scanner to ensure precise and safe navigation and voice-activated messages alert staff of the robot’s presence. It is equipped with 3D sensors and touch bumpers for obstacle detection and collision avoidance. Being a part of building automation, the robot can signal the doors to automatically open so it can move through.

Though these robots are very useful in hospitals and office scenarios, their usability is limited in home environments. Being larger in size and sometimes requiring modifications in the existing structures, these robots are not suitable for navigating in the home for monitoring or providing services to elderly people. Moreover, these robots generally do not have the capability to build a map of the environment on their own and need pre-assigned locations for navigation. Nevertheless, their capability of carrying weights from one place to another is a desired feature for a robot to be used in indoor environments for elderly care.

### 2.1.3 Robots for Facilitating Communication

A more recent class of robots that is now available for personal use is telepresence robots. These robots have been developed especially for the purpose of facilitating interaction between a person at home and a remote user. They have been used in office environments as well as in home environments especially for elderly care purposes. Their design reflects the need for traversing through closely placed furniture and narrow corridors in the home environment. The added benefit of robotic telepresence over traditional telepresence methodologies is that the robot can be controlled over the Internet giving the impression of being present at the place rather than just looking from one specific location.

QB, see Figure 2.6(a), has been developed by AnyBots to provide telepresence at offices and homes. It is equipped with a laser range finder that assists during the movements through narrow corridors and closely placed furniture. With it, a user can simply point the robot in the direction of a doorway and guide it forward without worrying about colliding with objects. The communication is carried over wireless LAN and video is captured using...
2.1. Commercial and Prototype Robots for Helping Humans

Figure 2.6: Telepresence Robots like, (a) QB (b) Giraff, are becoming more popular for remote presence. Images from: http://www.cnet.com/news/the-telepresence-robots-are-coming/ and http://www.aal-europe.eu/projects/excite respectively

(a)  
(b)  

Figure 2.6: Telepresence Robots like, (a) QB (b) Giraff, are becoming more popular for remote presence. Images from: http://www.cnet.com/news/the-telepresence-robots-are-coming/ and http://www.aal-europe.eu/projects/excite respectively

a 5 mega-pixel camera. Being connected to WiFi, there have been some issues in quality of communication and control of the robot.

The Giraff, see Figure 2.6(b), is a mobile robot to facilitate the elderly people to contact the outside world. It allows a two-way video call similar to Skype over the Internet. It can be remotely controlled via a PC to traverse from one place to another [Kristoffersson 11].

It is equipped with an LCD panel mounted as a head unit and has a 2 mega-pixel video camera with a 120° field of view. The head unit can be tilted up to 90° using a powerful servo motor. The base unit consists of 15 cm wheels that are differentially driven by high-power motors enabling speeds of up to two meters per second. There are 4 IR sensors that are used for avoiding collisions. A remote user can charge the batteries of Giraff by driving it onto the docking station. A full charge requires about 2 hours and once fully charged the robot can be used for more than 2 hours.

The new developments on the Giraff include autonomous docking to the charging station from a distance of about 2.5 meters, obstacle detection with a warning system to the remote user and localization of the robot in the environment with graphical feedback to the remote user [Kristoffersson 11]. The map of the environment is built only once and the operator verifies the map before being used. Giraff relies on particle filter for localization in the environment.

Although these telepresence robots have a limited degree of autonomous movements in the sense that they can sometimes avoid collisions from obstacles or can dock to the charging station from a specific distance, they are not capable of building a map of the environment or localizing themselves in the environment. Their use in elderly care scenarios is, therefore, limited to providing a communication channel between the person at home and the remote user, which is required by the caregivers and family members. Nevertheless, these robots also have the property of using them right out of the box and do not require
2. Mobile Robots for Serving Humans

2.1.4 Robots for Monitoring Purposes in Home Environments

The concept of using robots for elderly care is not new, but these robots are still in the phase of research and development and currently none of the systems is commercially available for domestic use. Several research groups have already been working on the concept of developing autonomous mobile robots specifically for the help of old people. The task of providing help to elderly people is multi-dimensional. At some point it is sufficient to develop a system that can provide telepresence services and at another side one likes to see a robotic solution that can work in kitchen to help in cooking and baking. Figure 2.7 depicts the futuristic vision of the tasks that robots will be able to perform for humans.

Pearl is developed as a nursing robot to assist elderly people at homes and elderly care facilities, see Figure 2.8(a). It is the first robot to be tested in an elderly care facility. It is equipped with a differential drive system, wireless LAN, a laser range finder, sonar sensors, and a stereo camera system. The main purposes of the robot are to remind elderly people about their daily activities and to safely navigate them from one place to another in the environment. The robot autonomously learns the map of the environment using the laser range finder. The location of the people in the elderly care environment is determined by map differencing [Pollack 02], that is by detecting a significant deviation from the already learned map. They have restricted the operating area of the robot by manually augmenting the map to avoid densely populated regions in the environment. This restriction ensures a safe navigation and prevents any collision with the elderly people [Montemerlo 02].

Figure 2.7: Artistic view of futuristic elderly care robots, where the robot helps the elderly person in performing household tasks, facilitate in movements and serves the person or guests with food. Images from: [Reiser 13] p.106 and [http://www.flickr.com/photos/willowgarage/4788324470/] respectively.
2.1. Commercial and Prototype Robots for Helping Humans

SCITOS G3 [Merten 12b] is developed by Metralab, see Figure 2.8(b). It is developed to facilitate the interaction between an elderly person and the rest of the family members and the caregivers. It can be used for telepresence and also for remote monitoring of an elderly person. It is equipped with a laser scanner and 24 ultrasonic sensors for navigation and obstacle avoidance. The robot is equipped with a 360° camera system in order to detect people. This allows the robot to move independently as well as safely and speak to the user directly. The robot interacts with people through both verbal commands and a touch screen interface. It can carry small objects, like keys, but is primarily meant to be a personal organizer. It can remind to take medication, alert scheduled appointments, and suggest activities.

Care-O-bot is a series of mobile service robots for domestic purposes developed as a research platform by Fraunhofer Institute of Manufacturing Engineering and Automation (IPA), Stuttgart, Germany, see Figure 2.9. The latest in the series is Care-O-bot III, Figure 2.9(c), a mobile service robot which has the capability to autonomously perform fetch and carry operations in home environments. It is able to move safely among humans to detect and grasp typical household objects. The objects are passed on to the humans using the tray mounted on the robot [Graf 09a].

Care-O-bot III has an omni-directional platform with four steered and driven wheels. This kinematic system enables the robot to move in any desired direction and to safely pass through narrow passages. In doing so, it is able to autonomously plan and follow a collision free path to a given target. Dynamic obstacles such as persons are detected by sensors and are avoided during the navigation. It is equipped with a flexible, commercial arm with seven degrees of freedom and has a hand with three fingers to grasp objects. Using tactile sensors in the fingers, it is able to adjust the grasping force. Moreover, it is able to

Figure 2.8: (a) Pearl, (b) SCITOS G3, are used to assist elderly people in home environment. Images from: [http://www.cs.cmu.edu/~flo/images9.html](http://www.cs.cmu.edu/~flo/images9.html) and [http://www.tu-ilmenau.de/?id=23601](http://www.tu-ilmenau.de/?id=23601) respectively.
autonomously open closed doors that block its path to the target location by synchronizing the arm and the platform movements.

The interaction between the human and the robot is carried out using a retractable tray that includes a touch screen for receiving commands. The exchange of objects also takes place using this tray. The robot places objects on the tray using its arm to pass on to the humans. The movement of the arm is stopped immediately as soon as people are detected in the working space of the robot. The robot is equipped with a stereo color imaging camera system, a laser scanner, and a 3D range imaging camera to detect objects, obstacles, and humans in the environment.

Willow Garage has developed PR-2, see Figure 2.10(a), as a research and development platform for home assistance. It has a mobile base with four steered and driven caster wheels, a torso, a head, two arms, and sixteen cores of computation. The torso can move up and down relative to the base giving it a height ranging from 1.33 m to 1.645 m. The two arms have seven degrees of freedom (DoF) each. A color camera with $640 \times 480$ pixels resolution is attached to the forearm to always keep the grippers in view. The grippers consist of a central palm with four articulating fingers. There is a 3-DoF accelerometer in the grippers and also an LED. The LED can be turned on or off to assist in calibration or locating the gripper in the camera images. The pan-tilt head is equipped with wide and narrow stereo cameras, a texture projector, and a laser range finder.

Overall, PR-2 is roughly the size of a person with four steered and driven wheels allowing it to move in any direction. It has been developed for the open source community, who are developing various scenarios where PR-2 can perform defined tasks. Being an open source and world wide development platform, the robot is capable of performing various household tasks like folding the laundry, opening doors, plug-in switches, etc. [Wyrobek 08] [Ciocarlie 10].

**Figure 2.9:** Care-O-bot I, Care-O-bot II and Care-O-bot III are designed to perform household tasks. From [Graf 09b], p. 139.
Rosie, see Figure 2.10(b), is developed by a group of researchers at Technical University of Munich (Technische Universität München) as an autonomous helper in an assistive kitchen environment. It is equipped with two KUKA arms to manipulate objects in the kitchen scenario. It has Mecanum wheels for omni-directional movement, a Kinect camera for 3D perception of the environment, proximity sensors in fingers for slip detection, a stereo camera, a time-of-flight camera and 3 laser scanners for environment perception. The task of Rosie is to work in a kitchen environment and manipulate objects to make a meal.

Kompaï, see Figure 2.11(a), by Robosoft has been developed to assist elderly people at home. The robot can be remotely controlled by authorized persons to investigate the health of a person at home. It is equipped with a Pan-tilt web cam, a multi-touch tablet as user interface, ultrasonic sensors, a laser scanner, and infrared sensors for obstacles, holes, and stairways detection, and a sensitive bumper. It can autonomously drive in a home environment avoiding obstacles in the path [Rumeau 12]. The main purpose of the robot is to maintain the schedule of the daily life of a person and to give reminders for meeting or for taking medicines.

uBot, see Figure 2.11(b), is a small and lightweight research platform that has been developed by a group of researchers at the University of Massachusetts Amherst for elderly care. It is a two-wheeled robot that is differentially driven with a rotating trunk. It is equipped with a camera system that is used for telepresence. It has two arms with 4 DoF with integrated force feedback and passive compliance that can be remotely controlled to manipulate objects in the environment [Deegan 06]. Each arm is about 0.5 m in length and can carry a load of about 1 kg.

An array of cameras has been installed in the home environment with overlapping fields of view [Deegan 08]. This distributed network of cameras is used for localization of the
2. Mobile Robots for Serving Humans

Figure 2.11: (a) Kompaï by Robosoft (b) uBot, used for telepresence in elderly care setup. Images from http://www.robosoft.com/robotic-solutions/healthcare/kompai/index.html and http://ageinplace.com/aging-in-place-technology/robots-for-those-aging-in-place/ respectively.

The robot does not have the capability to generate a map of the environment and relies on being operated remotely to communicate with the person at home.

2.2 Control Architectures for Service Robots

In the following, a few of the generally used control architectures have been described to develop an idea of working of an elderly care robot in the home environment.

The main components of the system architecture of SCITOS are shown in Figure 2.12. The architecture has been divided into five main groups, namely Interaction, Control, Sensors, Power Supply and Drive. These groups have been further composed of several modules interacting with each other to perform certain tasks. The Interaction module is mainly responsible for communicating with humans. The commands from humans are received via the Touch Display or by speech synthesis performed by the Multimedia unit. The Robot Head includes the robot’s eyes and a pet sensor. A tray equipped with an RFID reader is used to keep the belongings of the human and to remind him about his objects.

The interface required by the robot application can be classified into two groups: the human-robot interface, enabling the interaction between users or operators with the robot; and the robot supporting system interface, allowing the communication between the robot and other systems to broaden the robot’s functions. The human-robot interaction is mainly realized by the touch screen, microphones, loudspeakers, and the robot head as well as the RFID reader for administrator access.

The Vision Sensors include a front camera and a back camera which is used to support the docking to the charging station. A laser range finder and ultrasonic sensors constitute the
2.2. Control Architectures for Service Robots

**Figure 2.12:** Overall system architecture of SCITOS representing the functional view of architecture. From [Merten 12a], p. 113.

**Figure 2.13:** Basic Modules of system architecture of Care-O-bot. From [Hans 02], p. 383.
Distance Sensor module. The Drive System of the robot is differentially driven with two wheels attached to the center axis of the robot. The two motors for driving the robot are controlled by the Drive System module. The power is managed by the batteries installed in the robot. The charging and regulation of the power is managed by the Power Supply component. The interaction between different modules is carried out by the Control component which consists of the Embedded PC and the Robot Control Module.

The Main Control Segment is responsible for the execution of complex software algorithms and for the processing of real-time tasks. The Motion Sensor Segment includes the collision sensor; and the Vision Sensor Segment involves a front-camera and a camera at the backside of the robot supporting the docking process to the charging station. The Distance Sensor Segment includes a laser range finder and ultrasonic sensors. For the detection of higher obstacles, a depth camera is added to this segment. The Low-Level Interaction Segment is composed of an RFID tray, small objects can be stored.

Care-O-bot is developed with a hybrid system architecture having deliberative and reactive components. Basic modules of the system architecture of Care-O-bot are shown in Figure 2.13. The regular arrows represent data flow and dashed arrows represent the control flow. The Robot control is responsible for the perception of the environment using the sensor system installed in the robot and defining the trajectory of the robot by issuing commands to the actuators. The interaction between the human and the robot is carried out via the man-machine-interface module which mainly consists of the retractable touch screen tray. All commands are sent to the symbolic planner, which generates a list of possible actions that can be performed to complete a task. The execution module receives the list of possible actions, picks a suitable action from the list and triggers the robot control to execute it. All modules need a previously learned world-model containing information about the environment, which is available from the database. With its sensors the robot assistant analyses its environment and updates the database. Furthermore, it displays the progress of task execution via the man-machine interface [Hans 01].

Important components of the user interface include a fusion module, which merges all information given through the communication channels, and a separation module, which relates the output on the adequate channels. The control software of Care-O-bot is based on the “Robotics Toolbox”, which contains modules for implementing all necessary mobile robot control functions in several independent packages. Complex robot control can be achieved using various modules. With the given system architecture, Care-O-bot is able to plan and execute complex tasks autonomously.

For navigation, the control system for Care-O-bot is split into three hierarchical levels: The first level is the trajectory-tracking controller, which keeps the robot on its path. Then follows a controller instance that coordinates the four wheel modules and a third instance for each individual wheel module [Graf 09a]. Care-O-bot uses a flexible path planning method for nonholonomic mobile robots. An intelligent planner based on a static map of the robot’s environment has been developed. The generated path is smoothened and eventually modified in reaction to dynamic obstacles or other external forces [Hans 02].

In the above-mentioned architectures, a major task is assigned to one of the module or component which is fully responsible for the operation of that task. The modular architecture makes it easier to develop new smaller modules that can be easily integrated into the system at a later stage. These architectures focus more on the reactive and
2.3. Discussion

There has been a remarkable growth in the field of service robots specially in robots for personal use. Many solutions are now commercially available for the use in domestic and household environments. One of the key area that has evolved over the past few years is the development of elderly care robots which focuses on using a mobile robot to help elderly people at their homes or care-giving facilities. The research in this field is not mature enough to provide solutions to every problem faced by the mobile robots in complex indoor environments. Nevertheless several new issues arise when finding solutions to the existing ones.

From the above discussion, several functional and non-functional requirements for an elderly care robot can be extracted. In the following such requirements are mentioned.

2.3.1 Functional Requirements for an Elderly Care Robot

**Plug-and-Play** The robot for the elderly people should be workable right after taken out of the box. Unnecessary configuration and installations will make the robot less attractive to the people.

**Possibility of tele-operation** The robot should have the possibility to be remotely controlled. This is specially desired in case of any emergency situation that might have happened to the person at home.

**Carry weights** The robot should be able to carry some weights from one place to another for the elderly person.

**Drive over various floor surfaces** A typical home usually has a variety of surfaces. For the comfort of the people, sometimes the floor is made of wood and sometimes there are carpets on the floor. In order to perform any task for the benefit of elderly people, the robot should be robust enough to drive over different floor surfaces. Moreover, it is also very necessary that the robot can easily move from one surface to another. There can be uneven floor surfaces which can cause problems to the robotic navigation and thus need to be specially taken into account.

**Autonomous mapping of the environment** The robot should be able to generate a map of the environment on its own. This is specially required in situations when the location of the furniture has been changed or the robot has to work in a new apartment.

**Autonomous navigation** The robot should be able to move on its own in the environment. It should be able to generate a path from one location to another and should also be able to follow the generated path.
Collision avoidance The home environment usually consists of static and dynamic obstacles. The static obstacles, for example walls, furniture, will stay at their own place and collision can be avoided by planning a path that leads away from them but the dynamic obstacles need to be taken into account during movement. Therefore, the robot should be able to avoid any collision with both the dynamic and static obstacles.

Autonomous determination of locations for searching humans In case, a robot has to provide any services to an elderly person, it is necessary that it should be able to find the person in the home environment. For searching humans, several locations are required where a robot can navigate and look for the person. Currently, these locations are provided by the operator of the robot and are absolute in nature. In case the environment changes, these locations need to be identified again and given to the robot. In essence, a robot should be able to determine on its own the best locations to find the human in the environment. These locations are not easy to be found due to obstacles but once ascertained can facilitate the human search process. This requirement is lacking in the existing mobile robots.

Autonomous human search One of the most important factors in elderly care robots is that they should be able to search humans in the environment on their own. Human can be anywhere in the environment and searching them requires a well-defined strategy that would facilitate swift robotic movement and alleviate the search process. It is specially required as the robot might have to remind the person for daily schedule or investigate the health conditions of the person at various times of the day. This requirement is lacking in the existing mobile robots.

2.3.2 Non-Functional Requirements for an Elderly Care Robot

Working area of the robot An elderly care robot has to perform its duties within the home environment. Due to privacy reasons, the toilet should be excluded from the robotic movement.

Size of the robot Home environments are cluttered with furniture and items of daily use. Therefore, the robot should be small in size so that it can easily navigate through narrow passages and closely placed furniture.

Installation in the environment Any installation of sensors in the environment will cost time, effort, and money. Moreover, if the person changes the position of the furniture, all the calibrations of the sensors need to be performed again. Nevertheless, if the person changes the apartment, all the installations need to be done from scratch and this is not desired at all. Specially in order to enhance the usability of elderly care robots, it is important that no installations in the environment need to be performed.

Till now, the focus of many researchers is to develop a mobile robot that may be used in home environment for performing different tasks. Among others, autonomous human search and autonomous location estimation for conducting the search process are the most neglected requirements though being the most important in case of establishing any fruitful
human-robot interaction. The thesis in hand will especially focus on these requirements besides explaining the development of a mobile robot for elderly people that is smaller in size and can navigate autonomously in the complex indoor environment and can be used for providing services to the elderly person living alone in a home environment. The robot can be easily tele-operated by a novice user and can also be used as a communication channel between the elderly person and the caregivers.
3. Fundamentals

Considering the requirements mentioned in Chapter 2, this chapter focuses on the fundamentals that will be beneficial in understanding the design concepts for developing a robotic platform that is more suitable for monitoring the humans at their homes. The methodologies for searching human has been developed for the Autonomous Robot for Transport and Service (ARTOS) and has been tested in a real home like environment. Besides performing experiments in the real life scenarios, a simulated environment has also been created to fully understand the dynamics of the environment and to validate the approach beforehand.

In the following, an account of ARTOS, the real life test environment and the simulated environment are provided.

3.1 Autonomous Robot ARTOS

The Autonomous Robot for Transport and Service (ARTOS) [Koch 08] (see Figure 3.1) is an initiative to provide services to elderly people, living alone in their homes. It has been specially designed for indoor living environments and is able to navigate through narrow corridors and closely placed furniture in household environments. It can transport different objects within the home environment by navigating autonomously to different rooms and avoiding collisions. A telecommunication link can be established between the elderly person and the caregivers using wireless Internet available on the robot. Similarly, an audio and video link between the elderly person and the remote caregiver can also be established by initiating a Skype call to the robot. Moreover, it can be tele-operated in the environment by the health care personnel to detect medical emergencies. The control of the robot is based on the Modular Control Architecture (MCA2-KL) [Koch 07a]. Details about MCA2-KL can be found in Appendix B.

3.1.1 Design and Mechatronics of ARTOS

In case of an elderly care robot working in a home environment, the first and foremost concern is the dimensions of the robot. As described earlier, a typical home usually has
narrow passages that instill restrictions on the size of the robot. This leads to design a compact and small sized robot that is small enough to match the size of a vacuum cleaner and large enough to carry weights from one place to another in the home environment. Moreover, the width of the robot should be small enough to pass through these passages. Thus, in designing ARTOS, the width of the robot has been set to 37 cm.

Similarly, if the size of the robot is too small it can become a hazard to elderly people and may cause them to trip over the robot. Therefore, a height of 26 cm would be suitable for the current scenario. The length of the robot should be sufficient enough that objects can be placed on its top to be transported from one place to another. A typical example of transportation could be a case of bottles which is heavy enough that the elderly person cannot carry it and therefore need transportation within the home. Considering this requirement, the robot has been designed with a length of 55 cm. The details about the hardware of the robot are provided in Table 3.1.

Another concern in a typical home environment is that they usually do not have an even

<table>
<thead>
<tr>
<th>Size</th>
<th>length: 55 cm, width: 37 cm, height: 26 cm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight</td>
<td>self: 25 kg, payload: 50 kg</td>
</tr>
<tr>
<td>Kinematics</td>
<td>differential drive</td>
</tr>
<tr>
<td>Wheels</td>
<td>2 active on centered axis, 2 passive at front and rear</td>
</tr>
<tr>
<td>Motors</td>
<td>2 Faulhaber 2657WO24CR</td>
</tr>
<tr>
<td>Dynamics</td>
<td>max. speed: 50 cm/s, max acc.: 25 cm/s²</td>
</tr>
<tr>
<td>Power</td>
<td>2 lead batteries 12V each with 10Ah connected in series</td>
</tr>
<tr>
<td>Operation Time</td>
<td>approx. 4 hours</td>
</tr>
<tr>
<td>Computer</td>
<td>Intel Core2 Solo ULV 1.2 GHz, 2GB RAM, CAN Interface</td>
</tr>
<tr>
<td>DSP Boards</td>
<td>developed at the Robotics Research Lab</td>
</tr>
<tr>
<td>Motor Control</td>
<td>Motorola DSP 56F803</td>
</tr>
<tr>
<td>Operating System</td>
<td>Robuntu 12.04</td>
</tr>
<tr>
<td>Multimedia</td>
<td>Speakers, microphone, serial display</td>
</tr>
</tbody>
</table>

Table 3.1: Control Hardware of ARTOS
3.1. Autonomous Robot ARTOS

Figure 3.2: (a) CAD model of ARTOS shows the tilt mechanism. The blue point marked in the picture is the kinematic center of the robot. Following the conventions described in [Berns 09b], (b) Sensing area of ARTOS with planer laser range finder and ultrasonic sensors

surface at all the places. Some areas are carpeted and some are not. Similarly, at some places there are two layers of carpets. Such uneven surfaces can provide hindrance to the smooth navigation of the robot. To incorporate such variations in the navigational plane, a tilting mechanism has been designed in ARTOS that ensures that the driving wheels and at least one of the coaster wheel are always on the ground [Armbrust 07c]. Figure 3.2 shows the Computer Aided Design (CAD) model of the tilting mechanism in ARTOS which consists of a pivot-mounted plate to which the centered components are attached. Due to this mechanism, ARTOS can easily be driven over varying floor surfaces.

ARTOS is a differentially driven robot. Two active standard wheels have been mounted on a common axis on either side of the robot, which are controlled by two independent actuators. The linear velocity of the wheels along the wheel plane can be defined as \( r \dot{\psi} \), where \( r \) is the radius of the wheel and \( \dot{\psi} \) is the rotational speed of the wheel. As these are standard wheels there should be no sliding orthogonal to the wheel plane (i.e. the velocity must be zero).

Following the conventions described in [Berns 09b], the central point of the common axis is taken as the kinematic center of ARTOS. The origin of the coordinate frame of the robot lies on this kinematic center. The wheel coordinate system has its x-axis in the rolling direction and the y-axis as the normal to the wheel plane. The linear speed of the wheel is in the direction of the x-axis of the wheel frame. If \( \alpha \) is the angle between the x-axis of the coordinate frame of the robot and the wheel mount point and \( \beta \) is the angle between the straight line through the kinematic center and the fixing point of the wheel and the y-axis of the wheel coordinate frame, (see Figure 3.2a), then the values for the left and right wheels are \( \alpha_l = 90^\circ \), \( \beta_l = 0^\circ \), \( \alpha_r = -90^\circ \) and \( \beta_r = 180^\circ \) respectively. Let \( d \) be the distance between the kinematic center and each wheel, then based on the defined parameters, the linear velocity vector \( \mathbf{v} = (\dot{x}, \dot{y}, \dot{\theta})^T \) of the kinematic center can be determined using Equation 3.1 for left and right standard wheels.
\[
\begin{align*}
\sin(\alpha_l + \beta_l) \dot{x} - \cos(\alpha_l + \beta_l) \dot{y} - \cos \beta_l \dot{d} \dot{\theta} &= r_l \dot{\psi}_l \\
\cos(\alpha_l + \beta_l) \dot{x} + \sin(\alpha_l + \beta_l) \dot{y} + \sin \beta_l \dot{d} \dot{\theta} &= 0 \\
\sin(\alpha_r + \beta_r) \dot{x} - \cos(\alpha_r + \beta_r) \dot{y} - \cos \beta_r \dot{d} \dot{\theta} &= r_r \dot{\psi}_r \\
\cos(\alpha_r + \alpha_r) \dot{x} + \sin(\alpha_r + \beta_r) \dot{y} + \sin \beta_r \dot{d} \dot{\theta} &= 0
\end{align*}
\] (3.1)

By substituting the values of the parameters defined above and solving the equations result in

\[
\begin{pmatrix}
\dot{x} \\
\dot{y} \\
\dot{\theta}
\end{pmatrix} =
\begin{pmatrix}
\frac{1}{2} (r_l \dot{\psi}_l + r_r \dot{\psi}_r) \\
0 \\
\frac{1}{2d} (-r_l \dot{\psi}_l + r_r \dot{\psi}_r)
\end{pmatrix}
\] (3.2)

From Equation 3.2 it can be seen that the motion vector of the robot is the average of the independent wheel motions. The straight line motion is obtained by turning the wheels at the same rate in the same direction. The robot will rotate on the central point of the axis if both wheels are turned at the same rate in opposite directions. Other motion paths can be implemented by dynamically modifying the angular velocities and/or direction of the active wheels.

In order to balance the robot, two caster wheels - one at the front and one at the back of the robot - have been mounted. These wheels can cause problems in situations where the robot may move in reverse direction as the caster wheels must turn 180°. The offset in turn can result in an undesired motion vector of the robot. This could be avoided if the robot change the direction by always moving forward and turning.

The degree of maneuverability describes the degree of freedom of the robot and is defined as \( \delta_M = \delta_m + \delta_S \), where \( \delta_m \) is the degree of mobility and \( \delta_S \) is the degree of steerability [Siegwart 04]. The degree of mobility defines the effect of sliding constraints that affect the movement of the robot. The degree of steerability determines the degree of controllable freedom and is determined via the number of independently controllable steering wheels. Since Artos is a differentially driven robot, its degree of mobility is 2 and the degree of steerability is 0. Thus the degree of maneuverability comes out to be 2, which describes that the robot can be navigated in the home environment by controlling the motion of the two wheels. By only rotating one wheel, rotation around the second wheel as center can be achieved and by rotating both wheels in opposite direction, rotation around the kinematic center can be performed. These rotations with very small or none radius are very critical specially in the home environment, which does not have much space for turning with large radius.

Another benefit of using a differential drive on Artos in elderly care situation is that it has a simple kinematic model which is easy to realize. On the other hand, with a differential drive it is quite difficult to make the robot move in a straight line. Since the drive wheels are independent, if they are not turning at exactly the same rate the robot will veer to one side. Making the drive motors turn at the same rate is a challenge due to slight differences in the motors, friction differences in the drive trains, and the wheel-ground interface.
3.1. Autonomous Robot ARTOS

Figure 3.3: (a) The ultrasonic sensors are placed at an angle of 30° to each other and covers an angle range of 240° (b) Placement of ultrasonic sensors in front chain. From [Stocker 07], p. 52.

3.1.2 Sensor Systems on ARTOS

As mentioned in the functional requirements in Chapter 2, the elderly care robot should be plug-and-play. This implies that all the sensor systems need to be installed on the robot itself and no other sensing unit is desired in the environment. Home environments usually have narrow passages which are sufficient enough for the humans, but an autonomous mobile robot needs a precise information for navigation without hitting any of the obstacles. Therefore, for perceiving the environment, ARTOS is equipped with a variety of sensor systems. These sensor systems work on different principles and perform well in combination.

The most important sensors in such situations are the distance sensors, which provide the distance between the nearest obstacle and the robot. The data from these sensors can be used to generate a map of the environment. The well appreciated distance sensor in robotics for stable and reliable distance information is a laser scanner. The distance measuring principle is based on the time delay of reflected laser light. On ARTOS, to detect obstacles at a distance, the planer laser scanner Hokuyo URG-04LX has been installed at the front of the robot very close to the floor with a ground clearance of about 2.5 cm. The benefit of such an arrangement is that the scanner can detect obstacles that are very close to the ground, for example human feet. It has a measuring range of 20 mm to 4000 mm with a 240° field of view, linear resolution of 1 mm, and angular resolution of 0.36°. It can measure the distance with an accuracy of ±10 mm in case the object is in between 60 mm to 1000 mm, afterwards it is ±1 of the measured distance. The sensor requires 100 ms to complete one complete scan.

Since the laser scanner can scan in only one plane, it cannot properly detect furniture that does not have a continuous base and is standing on its feet. This furniture, like tables, is still low enough to cause a collision with the robot. Therefore, the planer laser scanner is supplemented with the ultrasonic sensors. These ultrasonic sensors generate sound waves to detect the distance to the obstacles. The sound waves are conic in nature and hence the low hanging obstacles can be easily detected. ARTOS is equipped with 14 of these sensors, separated as two chains for front and back, respectively [Stocker 07]. The central axis is pitched upwards at an angle of 15° to detect furniture standing on feet. These sensors have a range of about 80 cm and a cone angle of 60°. They have a precision of 2 cm and operate with 400 ms response time. Each chain of sensors covers an angle range of 240°.
The placement of sensors can be seen in Figure 3.3. The working area of the laser and ultrasonic sensors is shown in Figure 3.2b.

Besides distance measuring sensors, tactile sensors have also been installed on ARTOS. The purpose of these sensors is to immediately stop the movement of the robot in an unlikely event where both the laser scanner and the ultrasonic sensor might fail to detect a dynamic obstacle that comes in the way of the robot. Two sets of normal open bumper sensors have been mounted at the front and back of the robot, respectively. The response time of these bumpers is 1 ms, resulting in 0.5 mm movement of the robot in case the robot is moving with maximum speed and collides with an obstacle, which is a sufficient distance for stopping the robot without harming any object.

As mentioned in Section 3.2, there are RFID tags placed under the carpet of the assisted living apartment. These RFID tags have fixed coordinates and are being used as landmarks for various scenarios. To make use of the already installed infrastructure, an RFID tag reading module from Feig Electronics (IS ISC M02-B) has been attached to the bottom of ARTOS to localize the robot in the environment. It is a 50 mm × 50 mm × 15 mm chip with integrated antenna (see Figure 3.4) and has a detection range of 100 mm. It supports the ISO 15693 standard with 13.56 MHz operating frequency. It is connected via a UART interface to the on-board computer of the robot.

Finally, in order to facilitate applications like the detection of people in emergency situations, tracking of people in communication, monitoring humans, and estimating poses, a pan-tilt-zoom camera from Axis (PTZ 215) has been installed on ARTOS. It has a zoom of 48x (12x optical zoom and 4x digital zoom), pan range of 360° with a pan speed of 180°/s and tilt range of 180° with a tilt speed of 140°/s. The pan-tilt capability is especially required in ARTOS to detect persons while the robot is standing at one location. The camera has automatic day and night functionalities, providing color images in sufficient light and black/white images in dark conditions. It has a 1/4” interlaced CCD as image sensor with NTSC resolution of 704 × 480 and MPEG-4 and Motion JPEG video compression.

The sensors installed on ARTOS are sufficient for the task of finding a human in a home environment. The obstacle detection is carried out using a laser scanner and sonar sensor.
with an emergency stop realized using bumper sensor. The localization of the robot is performed using an RFID reader and human is detected using PTZ camera. With this combination of sensors, several experiments have been performed and reliable results have been achieved.

3.1.3 Mapping on ARTOS

The goal of robotic mapping is to generate a spatial map of the environment around the robot. This information is usually represented in either of the two forms of maps, world centric or robot centric. The world centric maps maintain information about the objects in global space with a fixed location marked as the start of the coordinate system. The robot centric maps, on the other hand, contain information about objects that are observed by the sensors on the robot. In this case, kinematic center of the robot is usually taken as the origin of the coordinate system.

[Thrun 02] has identified five main issues related to designing and implementing the robotic mapping system. These are as follows:

- Since sensor system itself is not perfect, therefore, all the data acquired by the robot using its sensor system is subject to measurement errors. All of these noises induced in the measurement are statistically dependent and therefore need to be taken into account.

- As the robot takes various measurements, the issue of correspondence between two sensor measurements raises. It needs to be determined that the new information belongs to a different object in the environment or is it the same object. Simply adding the new information in the map might make it completely useless.

- As the robot generate map of the environment, each additional measurement point added to the map increases the dimension of that map.

- Another issue during robotic mapping is the fact that the environment around the robot is dynamic in nature. Positions and states of objects change in the environment causing the need to re-map the environment.

- The final issue in robotic mapping is the robotic exploration relating to the aspects of mapping by an autonomous robot. Generally, a robot cannot observe the complete environment from a single location and, therefore, it has to navigate in the environment to develop a complete map. Thus the robot faces the problem that on the one hand its environment changes over time and on the other hand its self-localization has changing offsets, implying that the issues mentioned above rise again.

Similarly, depending upon the application and requirements, a variety of maps can be used to maintain the information of the environment.

A feature map besides having the information about occupancy of the cell also maintains information about the actual object that is present. The representation of objects is often in the form of features and coordinates information. These features are usually lines or general geometric shapes representing wall, corners, which are determined using range sensors.
Gasós 97 has detected lines from set of points obtained from sonar sensors. They have used fuzzy sets to represent the uncertainty related to the position of the detected features. The perception of features itself is unreliable due to the inaccuracies in the sensor measurements. Therefore, accurate and precise mapping becomes very difficult when using feature maps.

Topological maps represent the environment as a graph. The edges represent the distinctive places in the environment and the arcs show the connection between the places. Topological maps describe the connectivity of different places in the environment. These maps are used to map large environments and maintain less complex information. Such a map is usually represented by a graphical structure with nodes denoting the important locations in the environment and edges specifying the connectivity of these nodes. The edges can maintain extra information like costs for navigation etc. These maps are easy to develop but difficult to use for navigation purposes. An advantage of these maps is that they only have the information about important locations and even if the underlying environment suffers changes, these map remains valid as they represent only the connectivity between different locations.

Althaus 02, Althaus 03, Althaus 04 have used topological map for mapping the environment. The robot follows a person to generate the map. Whenever the person enters a room or leaves the corridor, he specifies this to the robot. Thus, the robot marks the nodes and edges of the topological map. The odometry information and sonar sensor data is used to localize the robot.

The topological maps require that the robot has already explored the places and a more exhaustive exploration of the environment is required for higher precision for position estimation. Moreover, due to sensor errors, sometimes it becomes difficult to define places in dynamic environments.

Metric maps maintain information about the geometry of the environment around the robot and are a detailed representation of the area. A lot of information can be stored in these maps that make them more convenient to use for navigation purposes. Since these maps store much information, they are usually difficult to construct.

In grid maps, the working environment of the robot is separated into grid elements or cells. Generally, a square shape grid is used in grid maps but triangular and hexagonal grids can also be used. Each element of the grid stores information about the area of the environment it is representing. In case of an occupancy grid map, it usually stores only the information that the grid element is occupied by an obstacle or not. But generally, it is also possible to store more complicated information about the terrain type, structure etc. Typically range sensors are used to generate these maps as the most valuable information is the occupancy of the cell.

Carpin 03 has used laser scanner to generate grid map of the environment. They have developed occupancy grid map to maintain the information about the obstacles and free spaces in the environment. Each cell in the grid map can be in one of three states namely, occupied, free and unknown. The state is determined by an integer value called occupancy counter which is limited in positive and negative directions.

Thrun 02 has implemented a probabilistic mapping technique for RHINO, a Type B21-Robot designed by the company Real World Interfaces Inc. (RWI) and developed as a
3.1. Autonomous Robot ARTOS

tour-guide robot at the “Deutsches Museum” Bonn by a collaboration of Carnegie Mellon University and the University of Bonn. The basis of the robot is a cylindrical body, 56 cm in diameter with a holonomous kinematic. That is, the robot is able to rotate with its four synchronous steerable wheels around the Z-axis [Thrun 98]. The occupancy grid map and histogram grid maps have been developed using the 56 infrared sensors and 24 sonar sensors installed on the robot.

A variant of grid map is a grid map that uses quadtree data structure to store the information about the obstacles in the environment. The basic idea behind the quadtree maps is that the cells with the same occupancy are combined and treated as a one big single cell, thus the level of detail increases top down from the root to the leaves of the tree. Every node on level \( i \) consists of four nodes on level \( i + 1 \). Each node of the tree stores the information about its level in the quadtree, and the occupancy type, which could be occupied, empty or mixed. The benefit of combining cells is apparent in path planning algorithms where less number of cells are required to be processed for planning a path. An example of such a map has been developed by [Shao 05], where a hierarchical list of nodes representing grid cells and a list of neighboring cells is created. From these lists, a quadtree map is generated.

Object maps maintain the geometric structure like lines, polygons, circles etc that represent objects in the real world. In order to generate an object map, extra computational effort is required to extract the geometric information of the objects in the environment from the data obtained by the sensor system of the robot.

[Anguelov 04] describes a probabilistic framework for detecting and modeling doors and walls in the office environment. They have used expectation maximization algorithm to classify doors and walls based upon their difference in colors, shapes and motion properties.

Though, object maps are a natural choice for building the map of the environment, current object perception methodologies can detect only simple objects. Moreover, sensor inaccuracies make it even more difficult to reliably detect an object and real environments are usually more complex to represent as simple shapes and objects.

Considering the benefits of occupancy grid maps, same has been implemented on ARTOS [Armbrust 07b, Armbrust 10]. The map is developed using distance sensors, laser scanner and sonar sensors, installed on the robot. Each cell of the grid map can have either of three values, namely, free, occupied, or unknown, specifying the presence of obstacle or observation status of the cell. These values are maintained in form of a bounded counter to limit the belief about the state, where positive values indicates an obstacle, a negative value is used for free cell and the unobserved cell is marked with zero. Figure 3.5 shows the result of mapping on ARTOS.

Two different maps are generated by laser scanner and sonar sensors respectively which are then combined together to form a single global map. The maps are combined according to the following rules:

- Case 1: A cell is marked to be occupied by at least one sensor type in its map, then the counter in the corresponding cell of the combined grid map is set to a positive value.
Figure 3.5: Map building process on ARTOS. (a) There are some obstacles in the environment. (b) Obstacles in-viewing range of the sensor system are detected and registered in the grid map (marked as red cells). (c) The cells in the grid map that are visible to the sensor system are registered as empty cells (marked green). The occupancy belief of white cells is unknown. Images from [Berns 09b].

- Case 2: A cell is not set occupied by any of the sensor type and is marked free by at least one sensor type in its map, then the counter in the corresponding cell of the combined grid map is set to a negative value.

- Case 3: A cell is not occupied or marked free in any of the sensor type, then the counter in the corresponding cell of the combined grid map is set to 0.

The map generation described so far considers that the robot knows its location in the World Coordinates and therefore obstacles can be registered in the world map. In practice, the robot has to move and change its direction and location to build up map of the rest of the environment. For an accurate mapping, it is mandatory that the precise location of the robot in the world is know. On ARTOS this is realized by developing localization mechanism and is described in the next section.

3.1.4 Localization of ARTOS in the environment

The localization is usually classified into three classes, namely geometric, topological and hybrid corresponding to the type of underlying mapping used. Geometric localization methods rely on two-dimensional grids for map representation. The goal in such localizations is to keep track of exact location of the robot in the map coordinates. Topological localization approaches, on the other hand, maintains an adjacency graphs for representing the map. Their focus is to determine the node of the graph that correspond to the location of the robot. Hybrid approaches, as the name indicate, combines both the methods for localization of the robot [Ulrich 00].

[Ulrich 00] has presented an approach based on detection of appearance for topological localization. They have used panoramic images to identify the location of the robot in the unmodified environments. The training has been performed using nearest-neighbor learning on the color bands in the image. The results are then classified using an unanimous
voting scheme. Since the methodology is based on color space, it is not useful in home environments where robot has to navigate at night when all the lights are turned off. Topological localization is workable with topological maps. Since in home environments, grid maps are usually used to represent the obstacles, therefore, some technique based on grid maps is required for localization.

[Biswas 10] has also defined map as a topological map and described the localization using WiFi signal strength on the floor of a university building. During the learning phase, they have built up a map with 223 vertices corresponding to 106 unique WiFi access points. They have reported a mean error of 1.2 meters during navigating the environment. The reported error is workable in corridors in a university environment but using this approach in an home environment which is already cluttered with furniture is not feasible due to unavailability of that many access points and obstacles that may disturb the WiFi signal strength. [Yeh 12] has used combination of WiFi signal strength and passive RFID tags to localize the robot. Using WiFi signals in home environment is very difficult as the signal strength varies in different rooms and it is also possible that in some room WiFi signals are not even available.

[Graf 09a] explains that localization in Care-O-bot is performed in two steps. At first, the position and orientation of the robot is estimated by integrating the route traveled. Therefore, only odometry information is taken into account which results in small unavoidable errors that add up over time. As a second step, additional information from sensors installed on the robot are used to detect significant features like walls or poles in the environment. These features are checked against their reference position stored in a global map already provided to the robot. This helps in localizing the robot in relation to the features detected in the environment. Although it is a workable approach, but requires a global map to be given to the robot for estimating its location with respect to the features in the environment. Therefore, it is not feasible as the location of objects or furniture changes in the home environment.

[Park 09] has used passive RFID tags to localize the robot in an indoor environment. They have used 13.56 MHz frequency band antenna to determine the RFID-tags installed on the floor. The antenna is attached to the under side of the robot keeping a distance of about 5 cm to accurately read the 198 tags covering an area of 420 cm × 620 cm with 34 cm spacing between the tags. They have reported a maximum error of 13.3 cm with their proposed approach.

The localization on ARTOS is carried out using two different ways for robust and accurate estimation of robot position in the environment [Koch 07b]. Primarily localization is achieved by performing incremental localization using odometry information from the wheels. The odometry based localization is further improved using artificial landmarks that are already installed in the form of passive RFID tags under the carpets of the assisted living lab, see Figure 6.8. More information about installed RFID tags can be found in Section 3.2. These tags have been cataloged to represent the absolute position in the World Coordinates [Bloch 07]. Therefore, these tags can be used to determine the position and orientation of the robot.

These tags can easily be detected within a range of 5 to 15 centimeters, thus the risk of interference from neighboring tags is extremely low. These RFID tags are readable via a 13.56 MHz reader installed under ARTOS. This is the similar methodology that has been
Figure 3.6: Path of ARTOS during navigation from start location to goal location in IESE Fraunhofer. ARTOS localizes itself using odometry information (shown as connected line). The jumps in the path show that an RFID tag has been read and localization error has been corrected. Image from Armbrust 10.

used to localize a human being in the environment with a shoe integrated with RFID card reader as described in Section 6.1.2.

\[
N = \text{number of RFID tags in range} \quad (3.3)
\]

\[
\text{Robot Position}_{x,y} = \frac{1}{N} \sum_{i=1}^{N} \begin{pmatrix} x_i \\ y_i \end{pmatrix} \quad (3.4)
\]

\[
\text{Robot Orientation}_\theta = \arcsin(\Delta y/\sqrt{\Delta x^2 + \Delta y^2}) \quad (3.5)
\]

The localization achieved using the RFID tags and the RFID reader results in quite stable position readings and maximum error is within the range of one cell of the RFID grid (12.5 cm). The error is increased if the robot moves too fast on the floor and then reader may not able to read the tags properly. The fusion of odometry and RFID based localization ensures that in case the robot is moving fast and is not able to read the RFID tags properly, an estimate of robot location is still valid. Figure 3.6 shows the localization process on ARTOS at IESE Fraunhofer. The error in localization due to odometry information is corrected as soon as an RFID tag is read by the robot. As a result, ARTOS is better localized in the environment.

3.1.5 Navigation in ARTOS

A variety of algorithms are being used to autonomously navigate the robot in the indoor environments. In the following a few of them have been mentioned.
Thrun 99 has developed a path planner namely coastal planner. The paths are generated based on path length and available information contents. Since the work has been done for a museum guide robot, the paths generated were more close to the walls as compared to the center of the environment. The developed approach is suited for tour guides but is not feasible in home environments as path close to the walls are more prone to collision with obstacles because of unstructured obstacles in the environment.

Hans 02, Graf 04 have used potential grids with wave front expansion for planning paths in indoor environments. The path planner works on a static map of the robot’s working environment. The path is smoothed and modified in reaction to dynamic obstacles in the environment using elastic bands [Quinlan 93].

Bennewitz 03, Bennewitz 05 has used A*-algorithm to determine the minimum-cost path in the indoor environment represented as an occupancy grid map. The changes in the environment are detected using laser scanner and camera installed on the robot and the path is adjusted accordingly.

Garrido 06 has presented a sensor based global path planner that works in two stages. During the first phase, safe areas in the environment are identified using Voronoi diagram. The second phase uses Fast Marching method on the extracted Voronoi areas to obtain the shortest path. To obtain realistic trajectories, they have dilated obstacles to ensure a safe distance between robot and the obstacles.

Avilés 07, Avilés-Arriaga 09 has implemented a navigation module using the machine learning techniques namely, behavioral cloning, inductive logic programming, and a simple grammar learning algorithm. Behavioral cloning is to show the robot what to do, instead of how to do a task (learning by example), and thus the system learns to avoid obstacles and orient itself towards a goal by following the human behaviors. More complex behaviors were learned using first order learning from sequences to learn grammars from sequences based on association rule learning. As mentioned by the authors, the trajectories learned in this way are not optimal in terms of distance traveled. Moreover, learning the motion pattern in home environment is difficult to realize.

Svenstrup 11 has optimized widely used Rapidly-exploring Random Trees (RRT’s) algorithm for path generation. The issue with RRT’s algorithm is that it is not deterministic. A path found from start location to the goal location is not necessarily the same when computed next time. Moreover, it is difficult to use in case of narrow passages. Finally, the determined path contains sharp edges which are not easily driven by a mobile robot unless modified.

The navigation system of ARTOS has been designed in a way that the robot may autonomously and efficiently traverse to destinations by avoiding obstacles and re-planning paths if required. It has been divided into two main components, namely global navigation and local navigation.

### Global Navigation

The global navigation on ARTOS estimates a path from current location to the goal location using the generated grid map of the environment. It is possible that the path planning algorithms may evaluate a path that may lead the robot very close to obstacles. In order to avoid such situations, the neighboring cells of the obstacles are assigned high costs which prevents selection of these cells in the path of the robot.
Three algorithms have been described in [Armbrust 07b] to determine a path from current location to destination that have been implemented on ARTOS. These algorithms are Breadth First Search, Dijkstra’s Algorithm, and $A^*$ Algorithm. These algorithms are discussed in more detail in [Stout 96, Tietz 96, Hart 68].

The advantage of Breadth First Search is that it never gets trapped in exploring the useless path forever and in case a solution exists, it will definitely find it out. Moreover, in situations, where there are multiple possible solutions, it always finds the solution with minimal number of steps. On the other hand, Breadth First Search is memory dependent search and stores all the nodes at a level before moving to the next level. This implies, that if the goal node is further down in the hierarchy it will consume more time to reach the goal node.

An improvement over the Breadth First Search is Dijkstra’s algorithm which takes into account the costs of edges from one node to another. This search strategy always processes those neighbors that have lowest cost. To perform the lowest cost search, a priority queue is used to store and sort the nodes. The priority of a node depends on the minimum known distance between the start node and the goal node. Thus smaller distance means higher priority.

Since, the cells near to the obstacles are given higher cost values, therefore, the path obtained using Dijkstra’s algorithm is better than the Breadth first search which do not incorporate the higher cost values during computation of path.

The $A^*$ algorithm uses a heuristic function for calculating the distance between the start and the end nodes. In many cases using heuristic is beneficial as it results in less node traversal as compared to Dijkstra’s algorithm or Breadth First Search. The heuristic process of determining the distance between the two nodes depends on two values. Firstly it takes into account the real cost to reach the current node from the start node and secondly it estimates the cost of reaching the goal node from the current node.

The benefit of using $A^*$ algorithm is that it expands the fewest number of nodes and use heuristics to reach the goal faster. Moreover, if there exists any solution from start to goal, $A^*$ algorithm will find it in shortest possible way. Finally, by using heuristic function it generates better path as compared to Dijkstra’s algorithm. $A^*$ is both complete (finds a path if one exists) and optimal (always finds the shortest path).

Considering the benefits of $A^*$ algorithm, it is mainly used for path planning on ARTOS. The path generated is in the form of points which have been categorized into relevant points and intermediate points. The relevant points are those where direction of movement is changed and intermediate points are those where there is no change in direction.

**Local Navigation**

The main objective of the local navigation is to determine the intermediate goals from the information received in the form of a planned path from global navigation and to control the trajectories to reach these intermediate goals. One of the challenging tasks in this regard is to avoid obstacles during navigation which will be explained later. These obstacles can be either known prior in the form of map or can be dynamic obstacles, such as moving humans that appear during robot navigation. The components of local navigation have been described in [Armbrust 07a, Armbrust 07c, Mehdi 09, Armbrust 10, Wettach 10a] and an overview is provided in the following.
There are two main reasons to improve the path. Firstly, the grid map divides the world into adjacent rectangular grid elements. The path thus generated is dependent on the dimensions of the grid element. Secondly, although the planned path keeps the distance between the robot and the obstacles into consideration, but the robot may still drive unnecessarily close to the obstacles and thus the speed of the robot is reduced.

Besides that, there are several other reasons when the robot is not able to follow the originally planned path, for example,

1. The environment has changed since the last path was planned and a dynamic obstacle has been encountered during following the path, resulting in a need to update the original path.

2. The self-localization process, explained in Section 3.1.4, induce a small offset during movement. This offset and afterwards correction of this offset some times result in a slight deviation from the planned trajectories. As a result, there is a chance that the robot can come closer to an obstacle than anticipated.

In order to overcome the above mentioned issues, two concrete strategies have been adopted. Firstly, a heuristic approach to environment changes and inaccuracies during localization by periodically planning the path from the currently reached location to the goal location. In this way, the most recent changes in the environment are reflected in the path.

Secondly, change the intermediate points by positioning them away from the obstacles. This has been realized by incorporating elastic band algorithm [Quinlan 93, Philippsen 03], described in [Berns 09a, Wettach 10a], and has been explained in the following.

Although the path planned is the shortest path from the source to the destination, it may bring the robot very close to obstacles. Consequently the speed of the robot may have to be reduced or may lead the robot through very narrow corridor of obstacles. In order to overcome such situations Elastic Band [Quinlan 93] has been employed to keep the robot at a safe distance from obstacles and, thus, to ensure that maximum velocity can be achieved. Besides that, Elastic Bands also optimize the path for smooth movement of the robot.

The Elastic Band consists of \( n \) number of bubble \( b_i \), where \( i = 1, \ldots, n \), each described by its centers \( \vec{c}_{b_i} \), its radius \( r_{b_i} \) and obstacle masking distance \( D_{mi} \) [Berns 09a]. These bubbles are created along the path overlapping each other according to the dimensions of the robot. This ensures that the area covered by the bubbles is a collision-free space along the planned path for the robotic movement.

The masking distance can be described as a linear relation to the position of the bubbles \( L_i \) along the path by Equations [3.6] and [3.7]

\[
L_i = \sum_{j=1}^{i} \| \vec{c}_{bj-1} - \vec{c}_{bj} \| \tag{3.6}
\]

\[
D_{mi} = D_{m,\text{max}} \cdot \begin{cases} 
0 & \text{if } L_i \leq L_{\text{min}} \\
1 & \text{if } L_i \geq L_{\text{max}} \\
\frac{L_i - L_{\text{min}}}{L_{\text{max}} - L_{\text{min}}} & \text{otherwise}
\end{cases} \tag{3.7}
\]
where \( L_{\text{min}} \) and \( L_{\text{max}} \) represent cumulative path lengths over which \( D_{\text{m}} \) is stretched and \( D_{\text{m, max}} \) is the maximum range at which the obstacles can be ignored.

The obstacles are represented as points \( \vec{p}_j = (x_{pj}, y_{pj}) \) of the grid cells and thus a set of masked obstacles \( \{ \vec{p}_m,ij \} \) related to the each bubble can be described by Equation 3.8. The obstacle \( \vec{p}_i^* \) closest to bubble \( b_i \) can be determined using Equation 3.9 and is used to determine the radius \( r_{bi} \) of the bubble by Equation 3.10 which is the guaranteed free space for the robot.

\[
\{ \vec{p}_m,ij \} = \{ \vec{p}_j : \| \vec{c}_{bi} - \vec{p}_j \| > D_{\text{mi}} \} \tag{3.8}
\]

\[
\vec{p}_i^* = \arg \min_{\vec{p} \in \{ \vec{p}_m,ij \}} \| \vec{c}_{bi} - \vec{p} \| \tag{3.9}
\]

\[
r_{bi} = \min_{\vec{p} \in \{ \vec{p}_m,ij \}} \| \vec{c}_{bi} - \vec{p} \| \tag{3.10}
\]

A bubble \( b_i \) is completely defined from the above mentioned relations and these bubbles can be translated to a given path with a minimal set of bubbles representing the free space relative to dimensions of the robot. The next step is to move the bubbles and adapt their size and position during the movement of the robot along the path. For this, the first bubble moves with the robot and the last bubble is fixed at the goal. All other intermediate bubbles are moved iteratively under the influence of two forces \( \vec{f}_{\text{int}} \) and \( \vec{f}_{\text{ext}} \) on their centers \( \vec{c}_{bi} \). The movement from time \( t \) to \( t + 1 \) is given in Equation 3.11.

\[
\begin{align*}
\vec{c}_{bi,t+1} &= \vec{c}_{bi,t} + \Delta \vec{c}_{bi} \\
\Delta \vec{c}_{bi} &= \alpha_{\text{tot},i} \cdot \left( \vec{f}_{\text{int},i,j} + \vec{f}_{\text{int},i,j+1} + \vec{f}_{\text{ext},i} \right) \\
\alpha_{\text{tot},i} &= \begin{cases} 
1 & \text{if } r_{bi} > r_{\text{lim}} \\
\frac{r_{\lim}}{r_{bi}} & \text{otherwise}
\end{cases}
\end{align*} \tag{3.11}
\]

The parameter \( r_{\text{lim}} \) defines the distance at which the elastic band starts to react to obstacles, \( \alpha_{\text{tot},i} \) is a proportional relation between bubble size, and \( \vec{f}_{\text{int},i,j}, \vec{f}_{\text{ext},i} \) are internal and external forces and are defined by Equation 3.12 and 3.13 respectively.

\[
\vec{f}_{\text{int},i,j} = \alpha_{\text{int}} \cdot \begin{cases} 
0 & \text{if } \| \vec{c}_{bi} - \vec{c}_{bj} \| \leq \varepsilon \\
\frac{\| \vec{c}_{bi} - \vec{c}_{bj} \|}{\| \vec{c}_{bi} - \vec{c}_{bj} \|} & \text{otherwise}
\end{cases} \tag{3.12}
\]

\[
\vec{f}_{\text{ext},i} = \alpha_{\text{ext}} \cdot \begin{cases} 
0 & \text{if } r_{bi} \leq \varepsilon \text{ or } r_{bi} \geq r_{\text{lim}} \\
\frac{r_{\text{lim}} - r_{bi}}{r_{bi}} (\vec{c}_{bi} - \vec{p}_i^*) & \text{otherwise}
\end{cases} \tag{3.13}
\]

The internal force \( \vec{f}_{\text{int},ij} \) defines the strength of cohesion between adjacent bubbles \( \vec{c}_{bi} \) and \( \vec{c}_{bj} \). The external force \( \vec{f}_{\text{ext},i} \) on the other hand, represents a repulsion of bubble \( b_i \) from its nearest obstacle \( \vec{p}_i^* \). The parameters \( \alpha_{\text{int}} \) and \( \alpha_{\text{ext}} \) are force weighting factors and \( \varepsilon \) is used to avoid division by zero.
Using the above mentioned equations, center of the bubbles are iteratively moved at each computational cycle. The radii of the bubbles are also updated taking into account any dynamic obstacle that has been detected in the vicinity by the sensor system of the robot. As soon as the first bubble, which is moving along the robot, coincides with its neighboring bubble, it is removed and the next bubble moves with the robot. As a result, the band represents a dynamically smoothed variant of the original path from the robot to the goal position. The continuously updating radius marks the narrow passages and free areas for the robot where speed can be adjusted accordingly. The elastic band process is stopped as soon as the robot reaches the goal position, that means, first and last bubble become the same.

Sometimes, during the robotic navigation, bubbles shrink below the dimension of the robot signifying that there is not much free space for traversal. In case if the bubbles are stable, indicating that the dynamic obstacle has blocked the path and it is not moving, a re-planning has to be initiated to plan a new path avoiding the newly detected obstacle. The new plan is then provided as an input to the elastic band algorithm for re-evaluating the bubbles from the current location to the destination.

In order to avoid obstacles during navigation, a behavior based approach has been adopted by implementing four safety behaviors that compete against each other to avoid obstacles. This approach has been described in [Armbrust 07a, Armbrust 07c]. These behaviors are Emergency Stop Behavior, Anti-Collision Behavior, Keep Distance Rotatory Behavior, and Evasion Behavior.

Emergency Stop Behavior monitors the bumper information and gets fully activated when the value reaches a certain threshold value. The activation of Emergency Stop Behavior inhibits other behaviors resulting in a complete stop of motion in the direction of the obstacle.

The purpose of Anti-Collision Behaviors is to monitor the front of the robot. In case the robot moves close to an obstacle, this behavior reduces the speed as a function of distance to the nearest object and actual speed of the robot.

The activity of the behavior is calculated as follows: If the distance to the nearest object is between upper and lower threshold values, then the speed is reduced linearly. If it is above upper threshold, the robot does not reduce the speed at all and if it is below the lower threshold value, the speed is reduced to zero.

In order to prevent collisions during driving the curves, Keep Distance Rotatory Behaviors has been implemented to ensure side ways safety. The purpose of these behaviors is to keep a certain distance between the robot sides and the objects in the environment during rotational movements. In case an object is detected near to one side of the robot, the turning in direction of obstacle is inhibited and an opposite direction turn is initiated.

The Evasion Behavior monitors the middle front of the robot and calculates distances and angles to nearest objects. When an obstacle gets too close to the robot, its activity increases and turning is initiated which does not stop until the distance to the nearest object exceeds a certain limit. The turning direction is determined by checking whether the obstacle is on left or right side of the robot. This behavior always turns the robot in the opposite direction.

Employing the above mentioned methodology for global and local navigation, ARTOS is able to navigate safely in the environment. The distance and orientation to the goal
Figure 3.7: Navigation process in ARTOS. (a) The robot plans a path from start location to the goal location using A*-algorithm. The path is shown in purple color. The path is modified for maintaining safe distance from the obstacle (red color) and having smooth trajectory using elastic bands (blue color). (b) As the robot moves in the direction of goal location, both path and bubbles are updated. (c) The robot passes through the doorways keeping itself in the center. The Safety Behaviors keep the robot away from colliding the obstacles.

3.2 Assisted Living Lab

In order to perform tests, a real-life environment was established at the IESE Fraunhofer Kaiserslautern. It is about 60 m² apartment that contains furniture required for living. Figure 3.8 shows the floor plan of the living apartment marked with installation locations of various sensor systems. Figure 3.9 shows the interior of the apartment, visible are different rooms of the apartment with furniture. This facility has been set up as a platform for conducting assisted living related research projects.

To meet the needs of these projects, a variety of sensor systems have been installed to monitor various activities and parameters in the environment. A state-of-the-art building automation system based on the European Installation Bus (EIB) has been installed in the apartment, which provides the facility to monitor all the switches and power plugs. Additionally, movement sensors and door sensors have been installed to monitor the activities of the person in the apartment. The main purpose of all these sensors is to determine any deviation in the normal routine life of the person living at home [Kleinberger 09].

For monitoring the food placed in the fridge, an intelligent fridge has been developed that is equipped with an RFID reader. It reads the passive RFID tags attached to edible products and notifies if a product has passed beyond its expiry date to the central computer at the assisted lab. It also notifies when a refill of items is required in the fridge. Furthermore, there is an interactive TV-based video-telephony system which is used to
Figure 3.8: The floor plan of testing apartment at IESE Fraunhofer. Location and type of various sensors installed in the environment are also mentioned.
provide multimodal interaction between the person at home and caregivers. There are also vital data monitoring devices for continuous monitoring and emergency situation recognition. Moreover an intelligent walking aid has been developed to recognizes falls [Becker 07]. The envisioned middle-ware communication architecture between different devices and sensor systems has been described in [Anastasopoulos 05].

In addition to these sensors, passive RFID\footnote{RFID: radio frequency identification} tags have been installed under the carpet for localizing the human being in the environment. A grid of about 4,000 passive RFID tags with a grid size of 12.5 cm × 12.5 cm (5 in × 5 in) has been implanted under the carpet. These tags are low cost and are in the form of flexible cards, usually called “Smart Labels”, see Figure 3.10. The RFID tags used are standard ISO 15693 tags operating at 13.56 MHz, as these tags and the corresponding readers are easily available and have satisfying characteristics, see [Finkenzeller 00] for more details about the RFID.

Besides the installation of various sensor systems in the environment, several user centered devices have been developed that are used in daily life. As an example, a walking stick has also been developed that is equipped with an accelerometer and communicates with a central computer via bluetooth. Several experiments have been conducted which determine the location of the human in case the stick detects a fall of the person carrying it, see [Nick 07] for more details. Similarly for monitoring the liquid consumption by the elderly person, an intelligent cup has been developed that measures the liquid intake by the person. Moreover, devices like bracelets to determine vital information like pulse, temperature, skin humidity have also been developed and tested in this assisted living facility [Kleiberger 07]. A separate room has also been established for monitoring the activities in the environment that facilitates non-obtrusive experiments.

In short, the assisted living lab serves as a hub for not only robotics related research activities, but also for focusing on monitoring humans and determining their movements in the environment. Several accessibility and usability tests have been successfully performed in the lab to evaluate the developed assisted living environment and demonstrate the platform for representing the results of research in various scenarios.

Despite the fact that there are numerous sensors installed in the environment, the goal of the thesis is to develop an independent setup that does not require any information from

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure3.9.jpg}
\caption{The real life apartment setup at IESE Fraunhofer for performing research activities in range of elderly care. Visible are the living room and bedroom with all the furniture essential in the home environment. Objects lying on the ground poses navigational challenges to the autonomous mobile robot.}
\end{figure}
3.3 Simulated Environment

externally installed sensors and focus on developing the perception of the environment on its own with all the sensors installed on the mobile platform. Therefore, Artos should only rely on its own sensors for finding the human being in the environment and establishing the location estimation for the person in the home.

3.3 Simulated Environment

An elderly care robot has to work in a very delicate environment and it has to deal not only with the uncertainty of the environment but also with the uncertainty in regard to the elderly person. Testing and validation of the robotic behavior is not possible in the real environment since any malfunction can harm the elderly person. It is, therefore, fundamentally desired to extensively validate the working of the robot before performing experiments in real life scenarios. Moreover, it is almost impossible to conduct test cases repeatedly with the same environmental conditions to regenerate and improve the results. Therefore, it becomes necessary to develop a simulated environment that is as close to the real scenarios as possible and should provide all the necessary parameters that can influence the working of the robot in real life situations.

For the above mentioned reasons, a simulated environment has been created. It is based on SimVis3D framework [Braun 07, Wettach 10b], see Appendix C for more details. The simulated environment resembles the real apartment with all the furniture models as are present in the real scenario. The modeling of the rooms, furniture, robot, and the human has been done using a 3D modeling tool, Blender. The placement of simulated furniture can be easily changed in order to replicate the movement of furniture in the real home.

The visualization of ARTOS is composed of chassis, wheels, camera, and laser scanner. The simulated sensors and the actuators for differential drive have been realized using components of simple simulated sensors and a motor simulation within the SimVis3D framework.

In a real environment, there is also a human being moving around in the home. Therefore, it is also necessary to have a simulated human character in the simulated environment which can move to different rooms and behaves in a manner as a real human does. In

---

Figure 3.10: Passive RFID tags by Euro I.D. (www.euroid.com). These tags operates at 13.56 MHz and are compatible with commercially available RFID readers.
order to incorporate this requirement, a human model based on well-defined standard H-Anim\(^6\) has been used. Figure 3.11 shows the human avatar and the simulated ARTOS in the simulated environment. The movements of different body parts of the character have been defined in order to make the simulated avatar close to a real human. These movements include sitting, standing, walking, and lying. For a more close to reality simulation, some behavioral attitude has also been described for the avatar which includes moving to different rooms at different times and performing above-mentioned movements in a manner which depicts the behavior of a real human.

With that many details in the simulated environment, it became very convenient to perform experiments in simulation and then afterwards transfer the system to the real robot for conducting experiments in the real apartment without any fear of malfunctioning or abrupt behavior of the real robot. It is also worth mentioning that switching between the simulation and the real robot can be performed very easily by replacing the hardware abstraction layer, explained in Section 4.2.

\(^6\)http://www.h-anim.org

Figure 3.11: Simulated Environment with human character and simulated ARTOS. Furniture is also visible in the scene that hinders the movement of the robot.
4. Design Concept for Searching Elderly People at Home

In order to develop a system that may be used to find an elderly person to inquire about needs, health, initiate communication or provide assistance, it is wise to observe how humans perform the complicated task of searching in everyday life. Therefore, this chapter focuses on the strategies that humans adopt to search lost objects and fellow human beings in their homes. Based on the experience learned from humans, a methodology for developing an elderly care robot that can search the elderly person in home environments has been presented afterwards.

4.1 Human Behavior for Searching

For understanding human behavior during search process, it is important to know how humans receive the information about their surroundings and how the brain manipulates this information into valuable resource. Most importantly, why does a person forgets about his objects and how he tries to retrieve or re-call the location of the object from his memory. In case no such memory exists, how a lost object is found?

The cognitive psychologists state that there is one memory system that is normally divided into three functions for storage, namely, sensory memory, short-term (often called working) memory, and long-term (often called permanent) memory [Anderson 00]. The sensory memory receives information from different senses. This memory serves as the information entry point to the mind. The decay process in this memory ensures that always an updated value of the senses is present. Some of the information is discarded and the rest is transferred to the short term memory. The short term memory (STM), like sensory memory, has a limited capacity and here also the information decays with the passage of time.

After processing, some of the information is discarded and the rest is passed from STM to the long term memory (LTM). The long term memory is the main repository for storing information over an extended period of time. The limits of its capacity are not known. It organizes the information in schemas which are related together. An important factor
Figure 4.1: Information processing performed by a human brain. A lot information is received from the sensors but very few information is stored in the long term memory. The rest is discarded by the brain as irrelevant information.

for retention of learned information in LTM is rehearsal that provides developing deeper understanding of the knowledge gained. Figure 4.1 shows the relationship between different types of memories.

It is evident from the brief description of memories, that some of the information is lost even before it reaches the LTM for long term storage. Furthermore, lack of rehearsal and repetition may cause the loss of valuable information in the LTM. Finally, as the information is maintained as collection of related schemas, a change in some related schema may cause in forgetfulness of valued information.

The searching behavior in humans is a reflection of retrieval of information from LTM. The schemas related to the target are combined with the schemas concerning their locations and as a result humans try to develop a methodology for searching a lost object. This human behavior is easily observable in two different situations, when they have lost an object or when they have to find a person. This behavior in these situations has been described in the following.

4.1.1 Finding Lost Objects

Due to the decaying process in sensory memory and short term memory, it is a tendency of humans to place an object at a place and then may forget the location of the object. People often comment that an important item cannot be found just when it is needed most or that an object that was recently used cannot be found. There can be many factors involved in not finding the object, such as the person has forgot the actual location of placement or some other person has moved the object intentionally or unintentionally to other place. In either case, the human thinks that the object is lost and it is not present at the place where it was supposed to be. An interesting behavior of human beings is that they exert effort in finding the “lost” object. This human behavior is very interesting to observe and reveals how human brain maintains information about the location of objects and its location in the environment. The factors that influence the finding of objects can be numerous and complex.
The researchers at College of Computing and School of Psychology of Georgia institute have carried out a detailed survey about the issue of losing the objects in a home environment and common ways that humans adopt to find these objects. In [Peters 04], they have discussed factors which lead to a frequent loss of objects and also factors which effect the retrieval of the lost object. The survey is conducted on 71 participants through a questionnaire focusing on their personal details, reasons of losing objects, successful strategies to find them and a systematic solution to find them. According to the results of this survey, losing objects is an everyday phenomenon and a significant problem is faced by majority of population. The reasons of losing an object vary significantly with age groups e.g. it is “inattentiveness” for young people, however, it is mostly a memory related problem with older people. As an outcome of the survey, they figured out the strategies that humans mostly follow in order to find a lost object. Most frequent among these strategies are:

**Locus search** - In this search strategy, locations are searched based on the information where the objects are normally found. It is a very targeted search technique that considers only a few locations for searching the object. Since only some locations are being searched, it finishes very quickly with either success or failure in finding the object. For locus search to work, it is necessary that the object has been seen several times and a cognition about its normal place is developed in the brain. This is the most commonly used method as about 33% of the participants of the survey start the search of the lost objects with this strategy.

**Exhaustive search** - During this kind of search methodology, locations are searched without any preference. As the name suggests, it considers each and every possible location for searching a lost object. It is apparent that this search strategy takes much more time for searching the lost object. Since all locations need to be searched, there is no special information is required to be developed in the brain. It is most helpful when searching an object that has not been seen many times and its usual location is not known. About 24% people used this search strategy to find their objects.

**Retrace search** - Humans use this kind of search, when it is known that they have recently lost the object. They sequentially search the locations where they were physically present. It is like back tracing the foot steps for searching the object. In this process, only some locations are searched for the lost object. The survey concludes that around 19% people use this search to find the objects.

**Memory search** - This strategy is based on the fact, that people search the locations where they have prior interaction with the object. In this strategy the focus is on the object itself. It is a walk-through the memories of the object and its relation with location. It has been seen that related information like an event or activity helps in recalling the location of the object. In such scenarios only some key locations are looked for the lost object. Around 11% of the participants search the lost object using this technique.

**Delegation search** - This strategy also helps where the person himself do not participate in the searching of the object, rather the duty is delegated to some other person.
This kind of search strategy is usually opted by the elderly people who sometimes do not like to search for an object. It is then on the other person, how he would like to search for the object in the environment. About 11% of the survey participant used this strategy to search the lost objects.

The above mentioned search strategies give a detailed insight how humans search for their lost objects in the home environment. The most widely used search among all is locus search, adopted by 33% people, relies on previous knowledge about the location of the object. This knowledge is developed with experience and with the passage of time. Moreover, this search strategy is extremely targeted and only locations where there is a possibility to find the object are searched.

It is important to mention that humans do not rely only on one particular strategy to find the lost objects. They adopt one particular methodology; search for the object; if it is not found, they use other techniques.

### 4.1.2 Finding Fellow Humans

The behavior of human beings in finding other humans in the environment is similar to that of finding the lost object. In order to observe the human behavior of finding the other humans in the environment, lets look at how a child behaves while searching for his mother.

The child comes home from the school. After entering the home, he directly goes to kitchen to find his mother. In his LTM, he has established the premises that whenever he returns from school, his mother will be in the kitchen preparing food for him. The child goes to other places like living room or bedroom in case he could not find his mother in the kitchen. His search continues and he goes from one room to another until he finds the mother or establish the conclusion that the mother is not at home. The information of presence of the mother in different rooms was established with experience which is developed by repetition of the same occurrences and successful efforts in finding the mother.

Similarly, let us look at another example where a student wants to find his professor. From the schedule during the semester, the student drives the information that the professor will be in the lecture or in his office at a particular time of day. This information is different for every hour. The belief of finding the professor strengthen with every successful search and deteriorates in case of failures.

These examples show that humans have a tendency to develop cognitive information about the presence of person at a particular location at a specific time. This information is enhanced with every successful result of search process.

### 4.2 Concept of a Robotic Solution

For a robot to efficiently and effectively find a person in the environment, it is necessary to follow the foot steps of the expert in the field of searching i.e. humans. There are multiple dimensions about how humans try to find an object or a person in an environment. Over a period of time they develop a notion of presence of objects or humans and afterwards use this information for searching. They search the object at one place, if they find it,
task completed; but in case they can not find it, they search other places and update the knowledge at the same time.

In order to replicate the human behavior for searching human being in the home environment, a mobile platform is required that can develop the perception of human presence and use this information in future for efficient searches. This perception needs to be updated upon every successful search and enhance with the passage of time.

The mobile platform should be autonomous or semi-autonomous to carry out the search on its own and it should be equipped with necessary sensor system to perceive the environment. The size of the robotic platform also needs to be optimized that may assist it in easy navigation in the home environment which is usually cluttered with closely placed furniture. The navigation should be collision free as the robot should neither harm the objects in the environment nor should get itself destroyed. For such a collision free navigation, position of obstacles in the environment needs to be remembered.

Based on the above mentioned requirements, a concept has been developed that can be realized to enhance the capabilities of a mobile robot to be an active agent and a companion to the elderly people in their homes. The main components of the strategy are depicted in Figure 4.2 and are explained in this thesis.

At the very basic level hardware support in terms of hardware controllers, sensor systems and motors is needed which has been encapsulated as “Hardware” in the “Hardware Layer”,

Figure 4.2: Design concept for searching human in home environment using mobile robot.
as shown in the Figure 4.2. In order to incorporate simulation, a “Hardware Abstraction” layer is required that can facilitate smooth and unnoticeable interchange between the hardware and the simulation. It is also necessary that all developments for real robot can easily be transferred to the simulation for performing experiments in simulation and vice versa.

For retrieving sensor information and controlling the movement of the robot, mapping and navigation modules are required, shown as “Control Layer” in the Figure 4.2. The “Grid Map Builder” receives the sensor information from “Hardware Abstraction” and delivers an occupancy grid map of the environment. This grid map is used by the “Navigation” for generating paths and to ensure a collision free navigation in the home environment. “Navigation” transfers the commands to control the motors to “Hardware Abstraction”. During navigation, the estimate of current location of the robot is maintained by “Localization” which also uses the grid map information.

The components in Hardware and Control layers have already been implemented and have been described in Chapter 3.

The Perception layer depicts the proposed concept of a search methodology that can be employed to search the human in the home environment. First of all, for establishing the perception of presence of humans and learning daily routine of the person, a self-learning mechanism is required that strengthen the state of belief whenever a human is found in the environment. This has been shown as “Learning Human Position” in the overall concept (Figure 4.2). This module will learn the location of human being in the environment at various times of the day. It is supported by “Human Detection” which will provide the information of human detection in the environment.

This can only be achieved if there is a mechanism that can determine some locations in the home environment which are suitable for searching the human, shown as “Search Locations Determination” in the Figure 4.2. It can use occupancy grid map of the environment to evaluate locations from where a maximum area is visible. These locations should be traversable by the mobile robot. It is a necessary condition otherwise robot may collide with obstacles in the home environment.

Once human routine has been learned and the locations are known, this information can be used to find the human in the home environment. All it needs is “Human Search” that can combine this information into a meaningful way. The “Human Search” will select among the possible locations based on locus search and memory search (described in Section 3.1.1) for searching the human. The locus search is the most popular search method used by the humans in real life. Logically it is also more appropriate to search for object at those locations where it is more likely to find the object. The memory search has been selected as humans are usually at the same place in short time intervals. Thus the chances are very high to find the human at his previous location. The robot will search the human at these possible locations and in case of successful search it will update its perception of human presence at different times of the day.

The above mentioned concepts are the main themes of this thesis and have been described in detail in the coming chapters.
5. View Points for Finding Human

The most important task of a personal mobile robot is to provide assistance to a person in every day life. Assistance can be of any type as in case of an elderly care scenario, for example, to inform the elderly person about someone at the door or to remind about appointments or even to perform transportation tasks at home. For providing such assistance, a mobile robot has to navigate and perform a search in the home environment to reach the person. The search process can be speed up, if the robot has already a knowledge of the environment and knows which locations are suitable for searching the person.

Identification of such places in the home environment is not an easy task as there are many obstacles in the environment which hinder the view of the robot. Traditionally, these locations are provided to the robot by a human operator. For example, [Volkhardt 11a] and [Granata 11] have used pre-defined locations which are specified manually by a human administrator or operator before using the robot in the environment. The robot navigates to these fixed locations and try to detect the human inhabitant from these locations. Though, this pre-determination of locations tremendously simplifies the process but it is not suitable in elderly care environment as position of obstacles may change or the robot may have to move to a new house with the elderly person. Therefore, there is a need to develop a methodology that can be used by an autonomous mobile robot to determine the suitable locations in the home environment on its own without interference from an human administrator or operator.

For the task of finding a person in the home environment, it is necessary that those locations should be selected that offer possibility for observing the maximum area around them. This effectively means that the robot has to navigate to less places while having the opportunity to find the person. This chapter focuses on development of such a strategy that can be used to determine these locations in the home environment by an autonomous mobile robot for finding the elderly person.

The determination of a location where the robot can navigate to find the human in the home environment cannot be achieved without knowing the environment. Therefore, a map of the environment is a pre-requisite for autonomous computation of these locations. There can be a variety of ways to generate different kinds of maps that may represent the
environment. One such possibility of obtaining a 2D grid map of the environment has been described in Section 3.1.3 where autonomous map generation using sensor system installed on ARTOS has been described.

In the following, a methodology for evaluating the locations for observing the human being from the generated map has been described.

5.1 Placement of an Observer for Area Coverage

The problem of placing an observing system, like guards or sensor systems, in the environment in an optimal way can be traced back to “Art Gallery Problem” in computational geometry. Victor Klee posed the problem of determining the minimum number of guards sufficient to fully cover the interior of an n-wall art gallery room in 1973. Since then, many variations to the problem and their solutions have been proposed. An in-depth theoretical analysis of the problem of maximizing camera coverage of an area, where the cameras have an unlimited field of view, is provided by [O’Rourke 87] and [Lee 86]. Their approaches, generally, consider corners and walls for the placement of cameras for maximum coverage. They also argued that the optimal placement of the sensor system in a polygon is an NP-Hard problem, leading researchers to focus on suboptimal solutions based on heuristics which solves the problem within adequate time and memory consumption at the cost of computing global optimum solution. However, [Baumgartner 10] has presented an exact solution and bounds to the general Art Gallery Problem. Their algorithm places an arbitrary number of guards randomly to cover the entire interior of the polygon and then computes a sequence of lower and upper bounds on the optimal number of guards until an optimal solution is reached.

Gerkey et al. in [Gerkey 06] has proposed a solution to the pursuit and evasion situation using a mobile robot with sensor system. The sensor system on the robot has a limited field of view but unlimited range. They have also reasoned that it is an NP-Hard problem to compute minimum number of robot’s locations for the task of searching in the environment. For observing a particular critical area, [Bodor 07] has proposed a methodology to use multiple camera system for the task. They have applied their algorithm to the problem of path observation in wide-area scenarios. The critical areas are observed by more than one camera system from different angles to avoid any occlusion that may occur. Clearly, their focus is not full area coverage. They have performed several experiments both in simulation and real world. In one of the experiments, they observed pedestrians in a courtyard. Figure 5.1(a) shows the result of their camera placement algorithm for observing the defined area which require 4 cameras to be installed for observation. This scenario describes use of static cameras for viewing. In another experiment, a mobile robot has been used to inspect a well defined region. The result after training the system was a trajectory for the robot to follow in order to have maximum observation of the required area, see Figure 5.1(b).

Kazazakis and Argyros [Kazazakis 02], have used divide and conquer policy to inspect the complete 2D workspace. The methodology assumes that the guards or autonomous mobile robots have 360 degrees field of view but have limited range in the sense that an object can be observed with sufficient detail if it is closer to the guard. Their algorithm decomposes the original space into polygons. These are divided into sub-polygons until
5.1. Placement of an Observer for Area Coverage

Figure 5.1: (a) Result of camera installation for observing pedestrians in a courtyard (b) Results of optimal view for mobile robot. From [Bodor 07] p. 279 and 284 respectively.

Figure 5.2: Results of placement of guards with limited range. Visible is the issue of redundant guards placed very close to each other at the right of the image. From [Kazazakis 02] p. 2847.
it becomes viewable by a single guard with limited range. The strategy works efficiently in cases where full coverage of the workspace is required. In some cases, their approach results in redundant guards in specific areas. Figure 5.2 shows result of their approach and also demonstrates areas were multiple guards have been placed very close to each other. Though, their approach is workable but they have not taken into account the limitations of a mobile robot while calculating the points as some of them are extremely close to the edge of the polygon.

Several researchers like [Zhao 09], [Gonzalez-Barbosa 09] have used grid map approach for placing the camera system at optimal locations. Monte-Carlo simulations have been used by [Zhao 09] to perform the experiments. The simulations ran on an environment of dimension $10m \times 10m$ and have a single obstacle in the environment. The experimental results show about 99% area coverage with 11 cameras to be placed in the environment. They conducted their experiments in simulation with limited focal length and limited field of view of the cameras.

[Gonzalez-Barbosa 09] defined the environment as an occupancy grid map and used camera model and environment model for the placement of the cameras. Besides using a directional camera, they have also used an omni-directional camera. Since it is a grid map, the location of an installed camera is defined according to the grid cells with the assumption that the cameras are installed in the center of the cell. Moreover, a cell is marked visible if it is viewable by at least one of the cameras in the environment. The results presented in the paper show nice placement of multiple types of camera in a very small grid of $6 \times 12$ cells with holes to represent the obstacles. Figure 5.3 shows the result of their approach.

A methodology for maximizing coverage with respect to a predefined “sampling rate” which ensures that an object in the environment will be imaged at a certain minimum resolution has been presented in [Hörster 06]. They work with triangular sensing range of the camera instead of a circular sensing range. The important consideration in their approach is to cover the most important areas rather than total area coverage. They have used linear programming for the optimal solution of the camera placement. Moreover, they have proposed a variety of heuristics for better placement of cameras with a variety of sensing range. Their results are quite good on a very simple environment that they have used for testing their algorithm.
5.1. Placement of an Observer for Area Coverage

Figure 5.4: Results of multiple robot covering the full area with identical sensor system. It can be clearly seen that some points are very close to each other as compared to others. Moreover, the path shows that the robots have moved very close to walls and safety of the robots has not been taken into consideration. From [Fazli 10] p. 5579.

An approach for full area coverage by multiple mobile robots has been presented by [Fazli 10]. The procedure requires representation of the environment as a simple polygon with polygonal obstacles and perfect sensor system. The results presented the full coverage of the environment with multiple robots having same features. Figure 5.4 shows the trajectory of multiple robots for observing the full area. It can be seen very easily that some selected locations are very close to each other as compared to other locations. Similarly some of the selected locations are very close to walls and thus safety of the robots has not been taken into account in this approach.

The solutions provided are very much workable in situations where full area coverage is required with no safety constraints related to the guard as the guard can be placed any where in the environment without any difficulty. These solutions do not facilitate a mobile robot which requires to maintain a minimum safe distance from obstacles in a cluttered environment in order to avoid any collision.

Similarly, the above mentioned strategies mostly select corners as key places for viewing the environment. In contrast, a home environment usually do not provide enough space to the mobile robot to reach a corner which is generally occupied with furniture and other objects.

Determining the key locations in the environment by using above mentioned methods may not be effective in case of an autonomous mobile robot as all these methods require a highly accurate description of the environment. A precise definition of the environment is usually not possible in mobile robots due to imprecision in sensor measurements and localization errors during navigation. The range of sensor systems installed on a mobile robot to accurately detect objects or humans is usually limited, which makes this problem
even more difficult than the classical Art Gallery Problem. Due to imprecision in sensor systems it is very difficult to define a closed polygonal structure of the environment when using mobile robots where above mentioned algorithms can be applied.

The situation changes tremendously if the task in hand is to find the human being in the home environment. The probability of finding the human being in the center of the room is much higher due to their daily life activities and it is not even necessary to look into every single corner of the environment. Finding locations for robot in home environment to search humans is also different from other scenarios as in this case it is important that robot should be able to identify those locations which offer maximum observability despite the fact that furniture will occlude the view of the robot. Therefore, a new methodology is required that can be used to determine key locations in the home environment considering limitations and constraints of both the environment and the autonomous mobile robots. The following section describes the developed methodology that has been validated both in simulation and real environments.

5.2 Locations for Finding Human

There can be many locations in a room from where a human can be observed. Some of them can be easily reached by a mobile robot and some would be very difficult for the robot to reach. It is also possible, that the robot may go to a location and then get stuck due to obstacles around it. Moreover, there can be certain locations in the environment from where the robot can only see walls or tables and nothing else. Therefore, such places need to be avoided. The most valuable places are those, which are easily reachable by a mobile robot and allow maximum possibility to observe the environment. In this thesis, these places are defined as follows:

**Definition 5.1** Locations in the environment from where a mobile robot can observe its surroundings to detect a person using its sensors are called View Points.

Autonomously determining View Points is a challenging task as it depends not only on the robot and its sensor systems but also on the environment. The number of obstacles and their placement in the environment will effect the number of View Points. In case, the central areas are free of obstacles, only a few locations will be required, as most of the area will be visible from these locations. Consequently, distributed obstacles will result in more View Points and may also decrease the total viewable area. The sensor systems installed on the robot also play an important role. In case the sensors have long range and can detect the objects from a distance, it will result in a fewer number of View Points. Thus, for short range sensors more View Points are required for maximum observability in the home environment. Another aspect of the process is to identify how to deal with the redundant View Points that covers the same area and are close enough that some of them can be easily ignored, a very prominent problem that can be seen in the results of [Kazazakis 02, Fazli 10].

The proposed methodology for autonomously generating View Points in a home environment requires a perception of the working environment in the form of a grid map. Since ARTOS already generates a grid map of the environment, as described in Section 3.1.3, therefore, it is wise to use this information. Besides grid map with marked obstacles, the
methodology also takes into account range of sensor system installed on the mobile robot and free space required by the robot for safe navigation to compute View Points. These parameters are defined as follows:

**Definition 5.2** The distance viewable by the sensors of the robot to detect an object or a human is called Sensing Range (SR) and is measured from kinematic center of the mobile robot.

**Definition 5.3** The distance between kinematic center of a mobile robot and its surrounding obstacles required to safely navigate in an environment is called Inner Circle (IC).

The Inner Circle is the distance that is required to be free from obstacles around a location to be considered in the process of View Point selection process. In this way it is ensured that a mobile robot has sufficient space for easy navigation at View Point. From the definition of Inner Circle, Safety Zone can be defined as

**Definition 5.4** The absolute minimum distance between kinematic center of a mobile robot and obstacles in its surroundings is called Safety Zone (SZ).

Any distance less than Safety Zone may result in a collision between the robot and the obstacles. The relationship between Safety Zone (SZ), Inner Circle (IC), and Sensing Range (SR), can be defined by Equation 5.1.

\[ SZ \leq IC \leq SR \] (5.1)

The strategy can be used to calculate View Points at multiple levels. These levels can be used to increase area coverage by introducing more View Points. Based on the experimental results usually two levels are sufficient to achieve adequate area coverage. The View Points in these two levels are defined as:

**Definition 5.5** The View Points determined at the first level offer maximum area coverage are called Primary View Points.

**Definition 5.6** The View Points determined at the second level to observe remaining area are called Secondary View Points.

The Primary View Points are characterized by offering maximum area for observing the environment. These are furthest from the obstacles and thus can be easily reachable by a mobile robot. The area observed by Primary View Points is exclusive and no two Primary View Points share the same covered area. This leads to some area in the environment that is not observed by any Primary View Point.

The Secondary View Points are locations to view areas that have not been covered by the Primary View Points. These locations are relatively close to the obstacles in the
environment and, therefore, offer less area coverage. The Secondary View Points can share area coverage among themselves and also with Primary View Points.

Primary View Points and Secondary View Points give a possibility to control area coverage and total number of View Points in the environment. The advantage of determining View Points in levels becomes more prominent in situations where prioritized area coverage is required.

Figure 5.5 shows an overview of the methodology for finding View Points in a home environment. The process starts with perception of the environment in form of an occupancy grid map with cells containing information about obstacles and free space in the environment. The neighboring cells of the obstacles are marked to indicate closeness to the obstacles. The process is also provided with the details about the robot which specifies space required by the robot for autonomous navigation. The final piece of information is sensing range of the sensor installed on the robot that will be used for detecting humans.

The goal of the algorithm is to maximize the area visibility with minimum number of locations in the home environment making it a typical optimization problem as selection of these locations or View Points will effect the observability. To address the problem a greedy approach has been adopted that selects a View Point as soon as suitable location is found in the environment. The use of greedy approach makes the developed solution as suboptimal solution which is acceptable in the scenario of searching an elderly person in the home environment as an optimal solution will be an NP-hard problem, described in Section 5.1.

Moreover, the developed approach for finding View Points is discrete in nature due to the fact that the environment is perceived in the form of a grid map. The grid map is a
5.2. Locations for Finding Human

collection of uniform sized cells aligned together to form a 2D map. A View Point selected in this grid map corresponds to the center of a cell. The cells are considered viewable from a View Point if and only if they are fully observable from that View Point. It is a strict constraint and as a consequence partially observed cells are treated as unobserved.

A View Point is characterized by its center $\vec{C}_{vp}$ representing the coordinates of the View Point, Safe Zone $SZ$, Inner Circle $IC$, and Sensing Range $SR$. Any free region in the home environment that is smaller than $SZ$ is not suitable for the robot due to closeness to the obstacles. From a mobile robot’s perspective, area within $IC$ should be free of obstacles for safe and better navigation.

The methodology described in Section 3.1.3 generates an occupancy grid map of dimension $m \times n$, where obstacles are marked as follows

$$\vec{M} = \begin{cases} 1 & \text{if there is an obstacle} \\ -1 & \text{else if it is free} \\ 0 & \text{else if it is not observed,} \end{cases}$$

(5.2)

where $\vec{M}$ is a particular cell in the map. Thus, a set of obstacles $O$ can be defined as

$$\{O\} = \{O_k : \vec{M}_k = 1\}.$$  

(5.3)

Any cell that does not have an obstacle can be a View Point and therefore need to be evaluated. To accomplish the task, a window of $IC \times IC$ is traversed over the grid map. A cell $\vec{M}_i$ can be candidate for being a View Point if the distance of the cell from any obstacle in the environment is greater than the Safe Zone ($SZ$) and there is no obstacle within the Inner Circle ($IC$) around the cell for easy robot navigation as described by Equation 5.4

$$||\vec{M}_i - O|| > IC,$$

(5.4)

A candidate View Point can be described by Equation 5.5

$$Cvp \in \{\vec{M}_i : \vec{M}_i = -1 \text{ and } ||\vec{M}_i - O|| > IC\}$$

(5.5)

The set cells that are viewable from a View Point can be described as

$$V_{ccvp} = \{\vec{M}_i : ||\vec{M}_i - Cvp|| < SZ \text{ and all cells between } Cvp \text{ and } \vec{M}_i \text{ are free.}\}.$$  

(5.6)

It is important to mention that as soon as an obstacle is observed during marking the viewable cells, all the cells behind that obstacle are simply ignored as the robot cannot see through the obstacles.

Once a candidate View Point is selected and number of cells that are visible from this candidate are known, it is possible that due to presence of obstacles like furniture in the
home environment it may not have optimal visibility and a little adjustment in selecting a View Point might give a better observability. The observability in this context refers to number of free cells that can be viewed from the candidate View Point.

Since an area has already been found where there can be a View Point which has more free cells in its surrounding. Therefore search for a better cell is performed only in the neighboring cells. In order to minimize the overall computation cost, the search is carried out only in one quadrant as all other quadrants have already been searched for the candidates in previous iteration. Finally, only the cell with maximum visibility is selected as the View Point in that area.

During the process of determining Primary View Points next View Point is evaluated after skipping all cells that are within the Sensing Range (SR) of the already identified View Point. This tremendously speeds up the whole process and also ensures maximum area coverage with minimum number of View Points.

The strict constraints in selecting Primary View Points result in many distributed areas where a robot can navigate in reduced speeds but these locations either do not provide maximum observability or close to some obstacles. In order to increase the area coverage and mark View Points in these distributed areas, Secondary View Points are calculated.

For evaluating Secondary View Points the value of IC is halved but it must remain more than or equal to SZ. The selection criteria includes that a candidate Secondary Point must be a free cell and is not being observed by any other View Point. Moreover, all the cells within IC of the candidate Secondary Point must not contain any obstacles but they can be observed by other View Points.

Algorithm 5.1: Grid based View Points algorithm for indoor mobile robots with the given Sensing Range, Inner Circle, and Safety Zone.

```plaintext
for no_of_level = 0 to max_levels do
    inner_circle = max_inner_circle / pow(2, no_of_level)
    if inner_circle < safe_zone then
        break
    end if
    List_of_View_Points = EvaluateViewPoints(sense_range, inner_circle, no_of_level)
end for
```

Algorithm 5.2: Algorithm for selecting a cell that has no obstacle in its radius of Inner Circle.

```plaintext
for i = min_x + inner_circle to max_x do
    for j = min_y + inner_circle to max_y do
        if grad_map[i][j] == free and IsInnerCircleFree (i, j, inner_circle) then
            Select a free cell in the map whose surrounding cells within a radius of Inner Circle are free.
            The selected cell is not in visibility of any other View Point.
            Optimize the selection of cell in the area for maximum coverage. (Algorithm 5.3)
        end if
    end for
end for
```
Algorithm 5.3: Algorithm for optimized selection of View Point around the surrounding of a candidate View Point cell that has maximum free cell within the radius of Sensing Range.

\[
\text{for } i = \text{Candidate}_x \text{ to Candidate}_x + \text{inner\_circle} \text{ do} \\
\hspace{1em} \text{for } j = \text{Candidate}_y \text{ to Candidate}_y + \text{inner\_circle} \text{ do} \\
\hspace{2em} \text{Find the View Point in one quadrant of candidate cell that has} \\
\hspace{2em} \text{maximum free cells within the radius of Sensing Range.} \\
\hspace{1em} \text{end for} \\
\text{end for}
\]

Algorithm 5.1 describes the process of determining View Points in levels. The levels are controlled by the values of Inner Circle and Safety Zone. By reducing the Inner Circle in subsequent levels result in View Points in the environment that a mobile robot can reach but with reduced speed. Algorithm 5.2 finds a free cell as a candidate View Point that is not being observed by any other View Point. Once it is known that all cells in its surrounding of Inner Circle are free, then a search is carried out to find the View Point. Algorithm 5.3 shows that the search is performed only in one quadrant to speed up the whole process. The View Point is selected based on the number of maximum free cells in a radius of Sensing Range. During this process, if an obstacle is detected, all cells behind the obstacles are marked unobservable from the current cell as sensor cannot see through the obstacles.

5.3 Experiments and Results

Several experiments have been performed both in real environment and various simulated environments to validate the developed strategy for generating View Points. The cells in the grid maps of these various environments cover an area of $10\text{cm} \times 10\text{cm}$. Thus each cell represents an area of $100\text{cm}^2$. Using these measurements, an environment like assisted living lab at IESE Fraunhofer, see Figure 3.8, which is $7\text{m} \times 9\text{m}$ can be mapped by a grid map with 6300 cells. Similarly a map of Robotics Research Lab with 36362 cells corresponds to an area of $363.62\text{m}^2$. In the following all discussion will be carried out in terms of cells.

Experiments in simulation of IESE

Before performing experiments in the real environment, the grid map based View Points determination methodology has been evaluated in the simulated environments. In order to generate the grid map of the simulated environment, shown in Figure 7.1, ARTOS has been navigated to different rooms. A cell in the grid map is marked occupied when the simulated laser range finder detects an obstacle, as described in Section 3.1.3. The cells marked in red are obstacles in the environment, neighboring cells of obstacles have been marked in orange, free space is represented as white colored cells and area unobserved by the laser scanner has been in gray color. In total there are 8025 cells out of which 928 cells are red, 1371 are orange, 3884 are white and 1842 cells are gray. Effectively, the robot has mapped an area of about $61.83\text{m}^2$ as red, orange and white cells. Since the simulation is closely to the reality, therefore, one can easily observe artifacts in generated grid map of the simulated environment, like obstacles that are partially marked or cells outside the environment has been marked as obstacles.
Figure 5.6: Experiments in simulation of IESE for autonomously determining View Points. (a) Evaluated Primary View Points in the environment marked as green cells with Inner Circle and Sensing Range as dark and light shades respectively. (b) Overview of area coverage (97.84%) with both Primary View Points and Secondary View Points (blue color). Only small distributed areas are not observed in the current scenario.

The approaches discussed in Section 5.1 consider a closed environments for determining locations. As it can be seen, it is not possible in the current situation and area is not bounded by walls and there are viewed space beyond obstacles as well. The developed methodology takes care of such situation by considering unobserved cells as the boundary for evaluation. Figure 5.6 shows the result of View Points obtained with Sensing Range, Inner Circle and Safety Zone set to 12, 8 and 4 respectively. With this configuration, there are 8 Primary View Points and 16 Secondary View Points that covers 2640 and 1160 cells respectively. Thus out of 3884 white cells 3880 are observable by the View Points which corresponds to 97.84% of area coverage and only small corners where it is difficult for robot to reach remained unobserved. These area are not critical as in case human is present in these corners some portion will still be visible from evaluated View Points.

As discussed in Section 2.3 that the location of furniture in the home environment may change. This implies that previously generated View Points will be not useful anymore and new View Points need to be determined in the environment. Changing locations of furniture is easier in simulation as compared to the real world, therefore, experiments have been conducted in simulation with changed location of furniture in the environment.

Figure 5.7(a) shows the simulated environment after changing location of different furniture. Basically furniture in two important rooms namely bedroom and living room have been changed. Figure 5.7(b) shows the result of determining View Points in the changed environment with the same values of Sensing Range, Inner Circle and Safety Zone that is 12, 8 and 4 respectively. As expected, the new locations of Primary View Points and Secondary View Points are different from the previous ones in the new map. There are
5.3. Experiments and Results

Figure 5.7: Experiments in simulated environment of IESE after changing location of different furniture in bedroom and living room. The Sensing Range, Inner Circle and Safety Zone remains the same.

4070 white cells in the map and with 8 Primary View Points and 17 Secondary View Points a total of 3952 cells are observable which is about 97.1% area coverage. An interesting Primary View Point has been obtained under the bed. This is due to the fact that the simulated bed is higher from the side and simulated laser scanner marks the cells under the bed as viewed and non-occupied cells.

Similarly, it is interesting to know the difference in placement of View Points if there is no furniture in the environment. Therefore, furniture in the simulated environment was removed and robot was navigated in the environment to build the grid map. Figure 5.8(a) shows the simulated environment without furniture. It consists of 6717 cells out of which 797 are red, 969 are orange, and 4951 are white cells. Figure 5.8(b) shows the result of View Points with same Sensing Range, Inner Circle, and Safety Zone as in previous experiments. There are 9 Primary View Points and 22 Secondary View Points which observe about 4889 cells (98.75% area).

Experiments in simulation with map of RRLAB

All the tests in simulated home environment give very nice results in terms of area coverage and determination of View Points. But in order to ensure that the developed methodology is also workable for larger environments, a simulation of Robotics Research Lab at University of Kaiserslautern was taken for determining View Points in the office environment. The robot was navigated in the simulated environment to generate map of the environment. A few artifacts were also introduced in the grid map during this process and can be seen in Figure 5.9.

The map consists of 3629 obstacles (red), 2986 neighbors to obstacles (orange), and 29747 free cells (white). In total it represents about 363.62 m$^2$. Figure 5.9 shows the result of determining View Points by changing Sensing Range to 10 while Inner Circle and
Figure 5.8: Experiments in simulated environment of IESE without any furniture. 9 Primary View Points and 22 Secondary View Points cover a total of 98.75% area and only 62 cells remain unobserved which are distributed in the environment.

Safety Zone are still 8 and 4 respectively. The mobile robotic platform can easily observe 96.57% of the lab with 77 Primary View Points and 113 Secondary View Points in the environment, proving the correctness and usability of the developed approach for various environments.

Experiments at IESE

For performing experiments in assisted living lab at IESE, Fraunhofer, the robot has been driven in the environment to generate a map of the area. Figure 5.10 shows the result of the generated grid map. The red cells represents the obstacles in the environment. The robot should avoid these cells in order to prevent any collision with the obstacles. The cells marked in orange are those which are very near to the obstacles. These cell are traversable cells but should be avoided in order to maintain a safe distance from the obstacles. Traversing these cells also cause the robot to slow down as the safety behaviors, described in Section 3.1.5, get activated and prevent the robot to move in its normal speed. Free cells have been marked in white color and the range sensors have not detected any obstacles at these locations. The gray cells represent the area that has not been explored by the robot and their occupancy is not confirmed by the range sensors.

The grid map also shows some artifacts that have been introduced during the process. These can be specially seen in the cells outside the living area. They have been generated mainly due to the inaccurate sensor measurements and localization errors. A few of these are also due to the fact that there were initially no obstacles at that place and afterwards due to introduction of a new obstacles, cells behind them were not accessible anymore but their occupancy has not been updated.

The experiment is based on the sensor systems and dimensional features of Artos. As mentioned earlier, ARTOS is equipped with laser range finder which has a range of about
5.3. Experiments and Results

Figure 5.9: Experiments in simulation of RRLAB. The map represents about $363.62m^2$ area. A total of 77 Primary View Points and 113 Secondary View Points in the environment covers about 96.57% area of the lab.

Figure 5.10: Grid map of the environment generated using range sensors of ARTOS in the real environment. Gray cells are unseen and white cells have been observed by the robot. Red cells represents the obstacles and Orange cells are neighbors to the obstacles. The artifacts introduced during mapping and erroneous sensor measurements are also visible in the map.
Figure 5.11: Experiments with ARTOS in IESE for determining View Points in the home environment. Inner Circle and Sensing Range are represented in darker shade and lighter shade respectively. (a) Primary View Points shown in Green have been determined at level 1 (b) Shows the complete observability from Primary View Points (Green) and Secondary View Points (Blue) after level 2.

Figure 5.12: Experiment in real home environment with Sensing Range set to 30 cells which corresponds to 300cm, equivalent to range of typical laser scanners used in indoor mobile robots.
40 cells (400 cm). The accuracy of sensor measurement decreases with the increase in distance from the laser scanner. Therefore, in this experiment, Sensing Range (SR) and Inner Circle (IC) are set to 12 cell and 8 cells respectively. The dimensions of the robot dictates the Safety Zone (SZ) to be 4 cells which is the minimum distance required by the Safety Behaviors. On ARTOS, several Safety Behaviors have been implemented which turn the robot away from the obstacles during autonomous navigation, see Mehdi 09 for more details.

At the first level, Figure 5.11a, View Points are selected based on the above mentioned criteria. After a suitable View Point is selected, it is optimized based on the best available location for maximizing the view. As can be seen in the figure, only a few View Points have been selected at the end of the first level. These Primary View Points offer maximum area to view with the given Sensing Range.

After completion of first level, some small distributed areas like corners or areas near obstacles remain unobserved. In order to achieve more observability, the second level is proceeded with IC reduced by half while other parameters remain unchanged. In this level, overlapping of area covered by different View Points is permissible. Figure 5.11b shows that after level 2, the combination of Primary View Points and Secondary View Points results in more than 95% of area coverage.

Several other experiments have been conducted in the home environment at IESE by varying Sensing Range to see the effect on View Points. Figure 5.12 shows that there is a significant decrease in the number of View Points when Sensing Range (SR) is set to 30 cells. Most of the area is covered by the optimally placed Primary View Points in the center of the rooms and rest of the region is covered by selecting location for Secondary View Points.

## 5.4 Discussion

The experiments conducted with the various environments in simulation and on real environment with the mobile robot ARTOS, show the suitability of the developed grid based algorithm for View Point determination and practicality of the approach. It clearly takes into account constraints required by a robotic platform to safely navigate in the environment.

The parameters of the algorithm can be adjusted according to the requirements of the environment and the robot. The Safety Zone, SZ, represents the minimum area required by a robot to navigate without colliding with obstacles and therefore can be adapted according to the geometry of the robot. The Sensing Range, SR, can be specified based on the range of sensors in which the sensors can detect the object of interest with acceptable accuracy. The Inner Circle, IC, can be used to control the number of View Points. Large value of IC will not fit in small areas and as a results there will be less number of Primary View Points. Therefore, IC needs to be adjusted according to the free areas in the environment. Though, IC can be any number between SZ and SR, initializing it with double the value of SZ gives sufficient number of Primary View Points and suitable area for robot to navigate.

The number of View Points can be controlled by limiting the number of levels and the required observable area of the environment. Surety of finding the elderly person in the
Figure 5.13: Area coverage in different environments with Primary View Points and Secondary View Points.

Figure 5.14: Time required to calculate View Points in different environments with varying Sensing Range.
home environment will increase by increasing the number of View Points, however it will also increase the computational effort for inspecting at various different View Points. Therefore, View Points at multiple levels have been prioritized based on the viewable regions. Figure 5.13 shows results of area coverage in different environments corresponding to Primary View Points and Secondary View Points. It can be easily seen that with Primary View Points the robot can easily observe more than 50% of all these environments. The results are much better in simulated environments (minimum 60% coverage with Primary View Points) where localization is much better and generated map includes less artifacts. As expected, the area coverage by Secondary View Points is related to the area observed by the Primary View Points. In totality, more than 90% of the area is observable by these View Points which is sufficient for detecting humans in the environment as the unobserved area is distributed in small chunks. Thus if human is at these unobserved areas, robot will still be able to partially see the person from the nearest View Point.

Comparing results of the grid map based View Point algorithm with already developed methodologies, it proved to produce better results than others. The Primary View Points and Secondary View Points keep the robot away from obstacles and provides good opportunity to view the environment. In terms of computational time, Figure 5.14 shows total time required by the algorithm to compute both Primary View Points and Secondary View Points in different environments. The worst case scenario of the developed approach can clearly seen as an environment which does not have many obstacles in it. In this case, maximum number of cells need to be evaluated for selection of View Points. The algorithm developed by [Zhao 09] takes minimum 1.90 seconds to calculate 10 locations for camera placement in the environment. The time increases up to 10.01 seconds when calculating the locations for 9 cameras. In both of these cased, time taken is significantly higher than the time required for determining View Points by the approach developed in this thesis. In contrast, [Kazazakis 02] has reported running time of less than 10 msec in general situations and up to 230 msec for the worst case scenario. Despite the fact that time required for determining location is extremely low, their methodology generates locations which are redundant and significant missing areas as has been described earlier and therefore specially not workable in elderly care scenario. The developed grid based algorithm for View Point determination generally takes under 500 msec to evaluate View Points in a typical home environment. The running time increases in case of small Sensing Range or when there are not many obstacles in the environment.

Nevertheless, methodologies described in Section 5.1 are more academic exercises and do not deal with real life situations like errors and inaccuracies induced in the perception of environment due to sensor system. The developed grid map based View Point algorithm has been demonstrated the capability of working in both the simulation and the real environment. Moreover, it also takes care of physical dimensions of a robot during calculation of View Points in the environment and produces result in fairly reasonable amount of time and sufficient observability of the environment.
5. View Points for Finding Human
6. Human Location Estimation Based on Daily Routine

Any assistive living setup can provide more facilities to a person at home when it knows the location of the inhabitant. These facilities can involve inquiring or reporting health of the person, or letting the person know if there is someone calling on the phone or some visitor is waiting outside for the door to be opened and many services like that. The elderly person can be anywhere in the home environment and therefore estimating the location of the person requires a comprehensive search in the living space of the person. It is also necessary that during the process of finding the person, his privacy may not be breached. This chapter focuses on the development of a search strategy that is based on daily routine of the person and expedite the search process by first investigating those locations where the inhabitant is most likely to be found. The process of searching is carried out using an autonomous mobile robot. In the coming section, an overview of already existing methodologies for locating the person in the environment using various approaches has been provided. Later in the chapter, the developed methodology for finding the human in an household environment has been discussed, followed by the experimental results performed in a real home environment with a real robot.

6.1 Approaches for Human Localization

In research community, there have been various strategies developed for monitoring and localizing the human being in the home environment. These works can be classified into following categories based on the approach used for human localization.

- Finding humans by installing sensors in the environment.
- Making sensors wearable by humans.
- Using indoor mobile robots to find humans.

In the following, a few of them are described in context of using them in scenario of elderly care situations.
6.1.1 Human Localization by Installing Sensors in the Environment

Georgia Tech in Atlanta has developed an Aware Home for conducting research in everyday activities of human being [Abowd 02b]. It is in the form of a home with two identical and independent living spaces, consisting of two bedrooms, two bathrooms, one office, kitchen, dinning room, living room and laundry room, see Figure 6.1.

For localization of human being, they have custom-designed RFID system that uses antennas in the form of floor mats distributed throughout the house, see Figure 6.1. Individuals wearing the passive RFID tags below the knee can be tracked by simply recording their movement over these mats. The system provides room-level occupancy information of known individuals in the home. Additionally, they have installed a grid of cameras to determine the pose and orientation of the person in the room. These optical sensors are installed in the ceiling at a height of 3 meters, with a Field of View (FOV) of approximately 120 degrees, see Figure 6.2. These cameras have been calibrated with respect to the location in the home. The human is detected by subtracting background and foreground images on each of the camera. The result of each foreground connected component in form of a blob is transferred to a central server which maintains the spatial and temporal information of these blobs that is accessible to the customers. The usage of blob information somehow ensures the privacy of the person.

For locating the person, they have also used open-air speaker ID system by installing an “always on” microphone system in the environment [Abowd 02a]. The microphones are constantly recording 5-seconds sound sample and a sound recognizer compare the features of audio sample with an already known feature set. In case of a match, the location of the microphone is considered as the location of the person.

Besides the above mentioned installations, they have also conducted experiments with fingerprint scanners on the door knobs to identify when the door is opened and hence localizing the human being [Abowd 02a]. Figure 6.3 shows the overall architecture for combining information from different sensors installed in the environment to localize humans in the home environment.

The approach developed is workable at homes to locate humans wearing different tags to be identified and localized in the environment. Asking elderly people to carry a special tag is not promising as they usually either they forget to follow instructions or they do not like to carry extra objects with them. Although, the persons in the room can be monitored with the camera system and room occupancy information can be retrieved, it requires a lot of installations in the apartment and calibration of the camera system with respect to the location in the home and thus it is not a desired plug-and-play solution. Specially in case, the person changes the living place, all the installations and calibrations needed to be performed again to make the system work again.

[Crandall 11] describe the use of Passive Infra Red (PIR) sensors in the home environment to estimate the location of a person in the environment. They have installed multiple PIR sensors along with simple door open/close switches and light control switches in the home environment, as can be seen in Figure 6.4. There are 44 PIR sensors with about uniform interval of 1.2 meters and at the height of about 3.0 meters on the ceiling facing downwards to the floor to detect the movement of the person. They have implemented
6.1. Approaches for Human Localization

Figure 6.1: (a) Floor Plan of Aware Home at Georgia Tech in Atlanta. Visible are the locations of RFID mats installed in the environment. (b) RFID Antennas under the floor mat at Aware Home to track human location in the home. From [Abowd 02b], p. 5.

Figure 6.2: View from a camera installed in the home to monitor the human. The human is detected by identifying the foreground and background images at Aware Home. Left image shows the actual scene and right image shows the segmented human in the environment. From [Abowd 02b], p. 6.
two algorithms for tracking and localizing the person. The first is a rule based algorithm that uses a set of simple rules combined with a graph of all possible routes between sensor locations to track the human. This algorithm has been named as Graph and Rule based Entity Detector (GR/ED). The other is based on Bayesian updates that take a corpus of training data annotated with the number of residents and build a probabilistic transition matrix which is used to update the world model. This algorithm is named as Bayesian Updating Graph based Entity Detector (BUG/ED).

Overall the accuracy reported using GR/ED is 72.2% and 85% when using BUG/ED where it can perform better when more than one occupant were in the home. Thus BUG/ED performed marginally well in comparison to GR/ED for complex scenarios and better localization of person over a longer period of time has been reported.

As reported by the authors, although full coverage of the environment was intended and therefore a dense grid of sensors was installed in the home, there were situations where human activity was not monitored correctly by one or more than one sensors resulting in less accurate results which could be improved by incorporating data from other sensors already installed in the environment. Although the results achieved are promising but in case of an elderly care scenario installation of sensors in the environment for localizing the human might not be desirable.

[Yu 06] has used Condensation algorithm to locate residents’ positions via fusion of multi-camera and sensory floor information. The Condensation algorithm is a kind of Bayesian filters and has the ability to handle multi-target tracking. By integrating information from
6.1. Approaches for Human Localization

In order to accurately determine the human location they have installed forty pressure sensors on the floor. These sensors are supplemented by installing four video cameras on the corner of the ceiling see Figure 6.5. Each camera covers a part of the region and these regions are partially overlapped.

Their system consists of three main components namely camera localization, sensory floor localization and condensation tracker. In order to detect the human being in the environment using camera, they perform background subtraction and apply human template matching techniques. The output of the sensor floor localization is the possible values of location of resident. They have reported that the result of fusion of data from cameras and floor sensors lead in better localization of the human being as compared to using information from only one type of sensors. They have claimed to achieve 30 cm accuracy in 78.8% cases by calibrating the cameras and sensory floor.

[Cucchiara 07] has also described a methodology by installing multiple camera system in the environment to detect people in the environment. They have installed single camera in adjacent rooms and track the movement from one room to other. They have proposed to use warping person’s silhouette to exchange visual information between partially overlapped cameras whenever a camera handover takes place. The Hidden Markov Model used in their approach maintains a belief of human even if there is a partial occlusion. Since they are using only one camera per room the observability is limited to field of view of the camera in the rooms. The results presented also demonstrate the limited applicability to the elderly care scenario where exact calibration of the camera system might be difficult to achieve.
6. Human Location Estimation Based on Daily Routine

![Figure 6.5: (a) Installation of pressure sensors in the floor and (b) four cameras on the walls for localization of human. The human is localized in the environment by combining the information from pressure sensors and four cameras installed. From [Yu 06] p. 3823 and 3826 respectively.](image)

[Hu 04] has rightly identified six key issues regarding problems that arise when using multiple camera systems for detecting humans. These issues are installation, calibration, object matching, automated switching between cameras, data fusion and occlusion handling. Therefore, although visual surveillance systems can be used, but an alternative solution would be much appreciated which is independent of these problems and readily usable.

There are significant technological, scientific and engineering challenges to overcome in order to create an environment that can identify the presence of its occupants. Nevertheless, installation of sensor system in the home environment is not only expensive, in terms of cost of installation, but also requires manipulation in the living environment for instance installation of wires and mounting of different sensors. Even more important, this setup may critically affects the privacy of the inhabitant. The results of study conducted by [Zagler 08] revealed that elderly people require that an ambient assisted home should do a lot of good things for them but there should not by any surveillance of the inhabitants.

In case of video streams, there is always a concern of privacy among the resident of the home. Some people are willing to sacrifice their privacy by such kind of intrusion in return for better care services, but many other do not. Thus installation of multiple cameras in the environment as mentioned in [Abowd 02b] and [Yu 06] might not be greatly acceptable by the elderly population due to privacy concerns.

In case of carried device or tag, a base station maintains the information of human location. These devices need to be carried all the time by the individuals to get localized, which is not feasible most of the times. Especially elderly people who sometimes forget or do not like to carry or wear any special instrument when at home, these solutions become unreliable in providing any significant results in estimating the location of the person. The solution for localizing humans provided by [Crandall 11] is much appreciated as they are not using any camera system to monitor the humans. But their solution heavily relies on installation of various PIR sensors, door switches and light switches to monitor the activity of the person.

Manipulation in the living space of the elderly person is a major concern which involves installation cost of sensors, wiring and maintenance cost. Moreover, these solutions are extremely personalized. For example, in situations where elderly person move from one
6.1. Approaches for Human Localization

(a) (b)

Figure 6.6: (a) User Badge (also called badge node or user node) that user has to carry for being localized in the environment. From [Tabar 06] p. 148. (b) Overview of the human localization system From [Keshavarz 06] p. 2. The network nodes, installed in the environment, communicate with the user node and based on signal strength the human is localized.

apartment to another apartment, all the installations and calibrations need to be performed again which not only is time consuming task but also inefficient in monetary terms.

6.1.2 Wearable Sensor Based Human Localization

[Aghajan 07] has developed a distributed vision based system for managing the accidents in the assisted living environment. They have installed a wireless sensor network having multiple sensing and event detection modules. Distributed vision-based analysis is used to detect posture of the person by merging features from different cameras images. The main focus of their work is detecting elderly people who suffers fall. A user wears a non-obtrusive identification badge, shown in Figure 6.6 for better estimation, although they claim that it is not necessary to wear the badge all the time. The badge node provides user-centric event sensing functions for example accelerometer signals, position sensing and voice communication with the care center [Tabar 06].

The approximate location of the user is determined by evaluating the signal strength measurements of the network in the home environment. The signal strength is measured with respect to the three static nodes installed in the room with known spatial coordinates. The triangulation thus formed determines the approximate location of the person carrying the badge.

In case the user is wearing the badge, the camera nodes are turned off. These are triggered when the user badge detects a sudden movement. The system can assess situations, anticipate problems, produce alerts, advise carers and provide explanations. An overview of their concept is shown in Figure 6.6.

The multi-camera images of the person at home are analyzed for posture before, during and after an alert for possible fall has been generated by the user badge. This analysis results in an assessment of the situation and refers to care-givers in case of an emergency. The
6. Human Location Estimation Based on Daily Routine

**Figure 6.7:** Mapping environment and tracking person using FreeDigiter [Schindler 06]. The two phase process starts with a person wearing FreeDigiters moves in the environment. The accelerometer keep tracks of the movement and proximity sensor detects doorways. The proximity sensor is also used to detect finger gestures performed by the person to mark doorways. The tracking phase using particle filter to localize the person in the rooms.

Use of images of the same scene from multiple cameras ensures that false alarm generation is reduced. It is also beneficial in case the user forgets to wear the badge. In that case, the human is timely monitored by the camera system for any possible accident.

They have developed a rule based reasoning for assessing the situation from the multiple camera images. The rules define the state of the human in terms of time [Aghajan 07]. They have used backward reasoning for more focused and efficient explanation of the scenario. The casual connection in between the features of particular situations and potential unsafe conditions for the occupant is established through the rules.

[Aghajan 07] have also recognized that capturing images raises issues of privacy and has ethical considerations, which may influence user acceptance. Therefore, they emphases on processing the images in the camera nodes of the wireless sensor network and transmitting the silhouetted images for reporting.

As mention above, the elderly people tend to forget very easily that they have to wear a special sensing device that can transmit vital information about their health, or position in the environment, therefore, any such solution that requires carrying a device cannot be used reliably to estimate the location of the person.

[Schindler 06] has presented a method for localizing in indoor environment using a wearable gesture interface. The ear-mounted device consists of an infrared proximity sensor and a dual axis accelerometer. The user walks in the home environment wearing the device and accelerometer keeps track of the footsteps taken while proximity sensor records the doorways. During this mapping phase, the proximity sensor also detects the finger gestures performed by the user to label the detected doorways. Figure 6.7 shows the wearable device used by the user to be able to get localized in the home environment.

The information from the device is transmitted to a desktop computer via Blue-tooth, where a topological map is built. After completion of topological map, the user may walk
6.1. Approaches for Human Localization

Figure 6.8: (a) A grid of passive RFID tags placed underneath the carpet at IESE Fraunhofer. From [Koch 07b] p. 274. (b) A shoe with an RFID reader to detect the RFID tags under the carpet and a bluetooth module for transmitting location of a person wearing the shoe in the environment wearing the device and the particle filter is employed to track the location of the user based on the generated map.

The solution provided by [Schindler 06] is unique as they initially map the environment by walking from room to room and marking doorways using finger gestures. Afterwards they localize using particle filter. As mentioned by the authors, sometimes the proximity sensor fail to recognize the doorways accurately. In their experiments, it also has detect a nearby human as a doorway. As a result, the topological graph was inaccurate and thus localization of human was not precise. Recovering from such failures is very hard for the system. Thus the system is not reliable and using such system for elderly care may cause problems in estimating the location of the person.

[Koch 07b] has mentioned an interesting approach to solve the problem of human localization in the home environment. They have installed a grid of passive RFID tags underneath the carpet in the testing apartment, see Figure 6.8a. Details about these passive RFID tags have already been described in Section 3.2. In order to localize the human using these standard 13.56 MHz tags, they have used RFID reader from FEIG Electronics with custom built modules for transmitting data via Bluetooth and mounted it on a shoe, see Figure 6.8b.

The sensing range of the RFID reader is about 10 cm. With this range the RFID reader is able to detect one or more RFID tags with the above mentioned grid density. In case only one tag is recognized, the human is precisely standing on that tag. Otherwise, the maximum deviation from the location can be 50 cm in either direction. 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.

The use of passive RFID tags and a shoe with RFID reader can determine the location of the person very accurately, in cases where human wear the shoe. In reality, this is not always the case. Even in normal scenarios many people do not like to wear shoes when at home. Therefore, ensuring that an elderly person may wear a specific shoe is a constraint that is very hard to realize. Moreover, if the elderly person changes the residence, the RFID tags need to be installed and assign specific locations at the new place.
6. Human Location Estimation Based on Daily Routine

6.1.3 Finding Humans using Robots

A companion mobile robot can also be used to find the human in the home environment \cite{Volkhardt11a, Volkhardt11b}. The methodology employs mission vision to detect the human being. They have claimed to detect people in standing in upright pose. In some situations the person is also detected seating in different poses. The methodology is based on the contextual color model of the respective place in the environment and a color model of the user’s appearance. The robot is optionally supported by infrared motion sensors that are installed in the smart home. These motion sensors provide a clue for a possible human location to start the search for the human in the home.

For tracking the person, they are using a multi-modal, multi-cue tracking framework based on Kalman Filter. The system incorporates a set of 3D position hypotheses of people along with their velocity, resulting in a six dimensional state space \( s = (x, y, z, v_x, v_y, v_z) \) for each hypothesis. When using the range-based detection module, a Gaussian is created at \((x, y)\) position and \(z\) is set to the height of a normal person. In case of using visual detection modules, the bounding box of the user is transformed into a 3D Gaussian by using the parameters of the calibrated camera and estimating the distance.

In order to detect the people resting at their favorite places, the system first learns the appearances of these places. Afterwards, the deviation from the already learned place with the current scene is analyzed to detect the person. It is important that the robot
6.1. Approaches for Human Localization

Figure 6.10: Results of experiment performed in simulation to search the human in an environment. The human trajectory is shown in blue color and the trajectory followed by the robot is in red color. The locations for searching the human have been manually defined and robot moves to these locations sequentially to find the person. From [Granata 11] p. 19.

orientation towards these places is pre-defined to have a minimum deviation in the normal situations of un-occupied place. The places are learned as multi-modal contextual color models, where color histogram is a 3 dimensional RGB (Red, Green, Blue) color space and the multi-dimensional discrete context distribution is represented by placement of the robot.

The color model of the human is similar to the color model of the place, where the color model of the user is learned with a limited variety of different clothes. Every time the user has different clothes, there is a need to learn the new model of the user. In order to recognize the user from the place, they have trained a single linear Support Vector Machine (SVM) on the data of multiple labeled positive and negative samples with empty and occupied places.

For traversing in the environment, they have utilized the information from motion detector sensors installed in the environment. The robot starts searching in the room where the motion detector has triggered at the last and traverse in different rooms based on which sensor triggered before the current one. Based on the developed methodology they have reported 70% success in finding the human being or identifying that no human is in the home. Figure 6.9 shows an overview of their approach and one of the results of finding the person.

The solution presented by [Volkhardt 11a] is more desirable where a robotic platform performs the task of finding the human being in the environment. The results are promising but the system needs to be trained every time when the user wears different clothes, hence might not be usable. Similarly, it will fail in scenarios where placement of
Similarly, [Granata 11] has also developed a strategy to find the human using a mobile robot platform. The robot develops the map of the environment using SLAM algorithm and locations in the environment are manually defined to find the person in the environment. The robot navigates to these locations based on time information of the last visit, that is the chosen point has been less recently visited. After reaching the destination the robot searches for the human in the vicinity. In case the robot does not find the human, it navigates to the next possible location, see Figure 6.10. The experiments performed with the assumptions that there is no slipping of wheels of the robot on the ground plane show the robot is able to find the human at the designated locations in an environment which contains only a table as obstacle. As mentioned by the authors, the experiments have been performed only in simulation and tests on real robot with real people in the real environment are yet to be performed.

The approaches used for searching the human using a robotic platform are promising. They have reported good results but they search the environment under very very controlled conditions. The search process starts from one pre-defined location in the environment and goes to the next location in a sequence and in this way it tries to find the human at these locations. The search process can be optimized if the robot considers the routine of the person when planning for visiting locations which has been completely missing in the above mentioned methodologies and will be addressed in greater detail in the next section.

6.2 Analysis of Life of Elderly People

In the previous sections, several scientific solutions to find a human in the home environment have been presented. Most of these approaches demand modifications in the home of the elderly person or require the user to wear or carry a specific instrument in order to be found in the environment. The more promising approaches with respect to their applicability in the real life scenarios uses mobile robotic platforms to search the human. These are a step closer towards a general solution of the human search problem. Although so far these solutions have either not been implemented on the real robot or require special configurations from the supervisors prior to be used in the elderly care scenarios. Such requirements not only make the proposed solution very local to only a specific environment but also raises the question of their usage in the real life situations.

For developing a robotic solution that is robust and generalize enough to be used in human search scenarios, an analysis of the situation is being presented. Undoubtedly, there are three key players in such a scenario where robot is finding an elderly person, namely:

- Home environment
- Robot
- Elderly person

The elderly person has spent all his life to decorate the home according to his needs and choices and this establishes feeling in the human being towards the home. A typical home
What I’d really like to do is stay in my current residence as long as possible? Among those age 45 and over (n = 2000)

- Strongly agree: 72%
- Somewhat agree: 12%
- Somewhat disagree: 7%
- Strongly disagree: 8%
- Don’t know/refused: 1%

Preferences if help is required? Among those age 45 and over (n=2000)

- Have help given at current home: 82%
- Move to a friend’s home: <0.5%
- Move to a relative’s home: 4%
- Move to a facility where care is provided: 9%
- Don’t know: 4%

Figure 6.11: (a) “What I’d really like to do is stay in my current residence as long as possible”. The chart reflects a sample of people age 45 and over. With increasing age, the percentage of those agreeing with the statement increased to 92% in the 65-74 age group and nearly 100% of those over 75 years of age (b) Though at the age of 45 years, 82% of the surveyed people prefer that assistance at old age should be provided at their own homes. Only 9% agreed to move to an elderly care facility for services at old age. Reproduced using data from Bayer 00
environment is usually cluttered with furniture, indoor plants, objects of daily use lying on the floor and things like that. Any modification done to the home by a third party is extremely undesirable whether it is for the betterment of the inhabitants or not. Moreover, the interior of the home does not remain the same. People like to change the position of furniture or objects at home after certain time. Thus inherently, the home environment is bound to change according to the need of the person. Any scientific or technical solution that intends to estimate the location of the human being in the environment should be robust enough to autonomously accommodate the changes in the environment. Moreover, the human finding system should be portable and work as a plug-and-play unit even in scenarios where an elderly person changes his/her place of residence.

The mobile robot is to be used on behalf of caregivers in the home environment to find the person in the home environment. Its main responsibility is to accompany and look after the condition of the old person at home. In order to perform its duties, it is required that the robot can navigate autonomously in the home environment. Given that there is not much space for robot in the environment due to objects placed in the home, the size of the robot becomes extremely important. Therefore, a robot of a size of a vacuum cleaner would perform better and may be more acceptable among the elderly people as they would know when they are being monitored or observed.

The elderly person is the one who requires caregiver services and monitoring of his/her health conditions. In one of the survey conducted by [Bayer 00], the results revealed that more than 70% of elderly people want to spend their lives at their home in their old age, see Figure 6.11. In order to comply with their desire and still to provide them with better health care services, remote presence or tele-presence becomes important. As indicated by the survey, a small number of elderly people also move to their friends or relatives, therefore, the tele-presence systems should be generic enough to cater such changes and hence a robotic solution is much more preferred than the installation of a surveillance system at home.
6.2. Analysis of Life of Elderly People

Figure 6.13: Survey shows that a person of age 65 years or above spend more than 5 hours of waking and non-work time alone even if a spouse or unmarried partner is present at home. This alone time increases to about 10 hours in case there is no spouse or unmarried partner. This data includes all days of the week and are averages for 2007 - 2011. Reproduced using data from http://www.bls.gov/tus/charts/older.htm

Figure 6.14: Number of hours spent in various leisure activities in a day by an elderly person. The most favorite leisure time activity of the day seems to be watching TV which takes on average of more then 2.5 hours of an employed and more than 4 hours of an unemployed person of age 65 years or above. This data includes all days of the week and are averages for 2007 - 2011. Travel for leisure is included in other leisure. Reproduced using data from http://www.bls.gov/tus/charts/older.htm
As people grow older, they start spending their day according to some unwritten schedule. This schedule varies from person to person as some like to spend more time in one activity and others in some other activity. Looking deeply into the lives of old people living at their own homes reveals interesting aspects of how they utilize their time. According to a survey conducted by Bureau of Labor Statistics\(^1\) of United States Department of Labor in 2011, an individual person of age from 65 years to 74 years spend on average more than 8 hours per day sleeping. This average hours of sleep increase to more than 9 hours per day in case of people older than 75 years, see Figure 6.12. Out of the remaining hours of the day, a person of age more than 65 years with no spouse or unmarried partner spends on average 10 hours of his waking time all alone at home. Of course, this time decreases in case there is a partner living in the home, see Figure 6.13. Going into more details, the time spent for leisure and sports activity is mostly consumed in watching television. The number of hours spent by un-employed person goes up to an average of 4.5 hours per day, see Figure 6.14.

From the above statistics, it becomes more evident that if a robotic platform has to find the elderly person, it should accommodate these numbers of human daily routine in its search strategy. These statistics also give a clue about how to devise the search strategy and where to start the search process at particular times of the day. More importantly, it should be kept in mind that not all the elderly people follow the same schedule. Some people may have an entirely different routine and therefore the most important and may be the most difficult concept to realize on a service robot is to make it adapt itself to the daily routine of an individual person.

### 6.3 Concept for Human Search using Daily Routine

As mentioned by [Zagler 08](http://www.bls.gov) elderly people want the services but do not want to be observed all the time. This implies that any robotic solution that has to be developed as a service robot, should not keep the human under its continuous surveillance and must give the person some time alone. This results in losing track of the person and the robot needs to initiate a search process to find the elderly person in the home environment.

Although, searching the person can be performed using tele-operated mobile robots, where the caregiver may remotely control the robot and navigate it to different rooms to find the person, but the task of controlling the robot itself will be cumbersome and will add an extra burden to the caregiver who just wanted to inquire about the health of the person at home besides intervening in the life of the person without his/her knowledge. Therefore, the robot should autonomously perform the task of finding the human being in the environment. It should autonomously navigate to different rooms of the apartment and find the person on its own.

Following are some of the strategies that can be derived from the methodologies used by humans for finding lost objects or persons (see Section 4.1):

- **Random search**

  In this strategy, a random location is selected and the robot navigates to that location to look for the human being. The benefit of this search strategy is that it is very
simple and can be implemented very easily. On the other hand, this strategy is highly unreliable and results cannot be regenerated for validation. This is similar to exhaustive search that humans perform when trying to find lost objects.

- **Closest location based search**
  The search starts from the location that is closest to the robot. In this way the search sequence is propagated from the start location of the robot in an exploring manner. This approach is beneficial in situations when prior knowledge about the presence of the person is not available. The strategy focuses on minimum robotic navigation to find the person and is subject to the environment.

- **Time based search**
  The robot navigates to that location which has been least recently searched for the human [Granata 11]. The benefit of the approach is that the robot traverse all the desired locations to find the human. On the other hand, it is also quite possible that the human is near to a location that has been just visited by the robot. In such situations, the robot will traverse all the environment before looking at place near to the last found location.

- **Last known location based search**
  The robot moves to the last known position of the human being and then traces back to where the human was seen previously. The benefit of the approach is that it can speed up the search by looking at the last seen location in cases when search is performed in short intervals as the person might be at the same place as before. In this strategy some knowledge of the environment and the person is required. This search depicts the memory search methods used by humans.

Random search, closest location based search, and time based search specially focus on the environment and the searching agent. These approaches do not consider human daily life attributes during process of finding the human. The last known location based search do consider the presence of human in the environment and accommodates it in the searching process. Moreover, the home environments usually do not provide an easy to navigate place to the mobile robots and searching randomly in such an environment may not be time efficient. Therefore, preference should be given to the places where it is more likely to find the human being at particular times. Therefore the basic concept is that the human daily routine should be taken into account for an accurate and efficient search. This proposed idea can be stated as:

**Search based on probability of human presence:** The robot moves to the places where it is most likely to find the person. This perception is developed over a period of time by maintaining probabilities of presence of the person at different locations.

The key benefit in using this strategy is that it can find the human being very quickly but it requires time to build the knowledge about the routine of the person. This search is more closely related to locus search (see Section 4.1) which has been used mostly by the humans to find the lost objects.
6. Human Location Estimation Based on Daily Routine

Figure 6.15: Concept of human search that incorporates human presence information for an autonomous mobile robot

Figure 6.15 provides an overview of the proposed concept where an autonomous mobile robot, ARTOS, navigates to different rooms in order to find the human. During this interaction with the environment it perceives the information of human presence and also avoid obstacles during navigation from one location to another besides maintaining state of its current location in the home environment. During the process, it learns the locations of presence of the person in the environment and update its belief with respect to time. The granularity of this belief system is dependent on the application and the underlying scenario.

As the human is independent in his/her movement in the environment, therefore, there is no assurance that a human is always found at one particular location at a given time of the day. It is only a probabilistic belief that the presence of the person is more likely but other places are also possible. Consequently, the search strategy should determine all possible locations in order to find the human.

It should also be kept in mind that the daily routine varies from person to person and thus the solution of searching the elderly person should be generalized enough to learn and adapt to these variations. In order to have a generic solution for searching the elderly person at home, the robot should able to learn on its own from the daily routine of the elderly person and pursue the search from the learned behavior.

6.3.1 Learning Daily Routine of an Elderly Person

On a robotic platform, the learning of human daily routine can be carried out in many different ways. In general three types of learning are more popular namely, supervised learning, unsupervised learning, and reinforcement learning [Russell 95]. The supervised learning requires a dataset in the form of input-output pair for learning positive and negative examples. In case of learning human routine, the input-output pair can indicate a location and time along with the human presence information. The positive and negative examples may indicate whether a human is present at that particular time at that place or
not respectively. Such a dataset of presence of persons at home at different times is very hard to develop as it is not necessary that the person exactly follows given input-output dataset. Moreover, the daily routine of the person may change over the period of time and therefore, using supervised learning algorithms to learn the places for searching the human is not feasible.

In case of unsupervised learning, a dataset of inputs is required and the system learns the output on its own. Once again a set of positive and negative samples is required to learn using unsupervised learning algorithms. Moreover once learned, variations in the life of the person are very hard to integrate to the developed system.

One of the main characteristic of both supervised and unsupervised learning is that they have a defined learning phase and after learning phase further variations are not possible [Russell 95].

For reinforcement learning the learner is not explicitly given the inputs and, thus, it has to interact with the environment to receive the information [Barto 04]. Based on this interaction the agent tries to maximize a numerical reward value which in this case can be the probability of the person at a given location. Thus, the learner discovers the sequence of actions that yield the maximum reward by trying all the possible actions. For the current problem of finding the human, this type of learning suits the best as no explicit desired output needs to be given and the learning process is a life long process where any future variations in the daily routine of the person can be easily accommodated. In the following a brief description of reinforcement learning is being presented for evaluating its usability for the task of learning human routine.

Reinforcement Learning for Learning Human Routine


> “Of several responses made to the same situation, those which are accompanied or closely followed by satisfaction to the animal will, other things being equal, be more firmly connected with the situation, so that, when it recurs, they will be more likely to recur; those which are accompanied or closely followed by discomfort to the animal will, other things being equal, have their connections with that situation weakened, so that, when it recurs, they will be less likely to occur. The greater the satisfaction or discomfort, the greater the strengthening or weakening of the bond.”

It is evident that in any situation where reinforcement learning has to be applied, the learner has to continually interact with its environment and the learning process does not rely on supervision [Sutton 98].

The main components of reinforcement learning, besides the agent and the environment, as described by [Sutton 98] are:

A policy: It describes an action that a learning agent performs when it is at a perceived state at a given time in the environment. It is similar to a set of stimulus-response rules or associations described in psychology. The policy is the most central part of the reinforcement learning as it describes the behavior of the learning agent.
A reward function: It describes the goal of the reinforcement learning problem and defines the intrinsic desirability of a state by assigning a number, a reward, to it. The objective of the learning agent is to maximize the total reward it receives by selecting states with larger rewards. In general, a reward function is unalterable by the agent. However, it serves as a basis for changing the policy. In short, a reward function is the defining feature of the problem faced by the agent. It is a feedback to the agent about its performance during the current policy and defines what are good or bad events for the agent. In a biological system, a reward function can be regarded as a pleasure or pain received after performing an action.

A value function: The reward function defines the good and bad for an agent in an immediate sense, a value function on the other hand describes what is good for the learner or the agent in the long term. The value of a state is the total reward a learner can expect to accumulate in future if it starts from that particular state. Thus, the value defines the long-term desirability of the state. Evaluating the values is much harder as compared to the reward. The reward is given directly by the environment, but the values are estimated and re-estimated from the sequence of observations made by the learner over a longer period of time. A reward can be seen as primary in nature, whereas values are secondary as prediction of rewards. A policy attempts to take those actions which takes the learner to states with highest value as these actions lead to greatest amount of reward in the long run.

A model of the environment: The last element of some reinforcement learning system is a model of the environment which mimics the behavior of it. The model might also predict the next state from a given set of states and actions.

The sole purpose of the learning agent is to learn a policy that may determine an optimal action that can be performed when in a particular state. Since the agent or learner is continually interacting with the environment, the right set of control actions for all situations are usually not known in advance and therefore it is very hard to come up with a training set, a set of inputs and a set of control actions for defining the optimal moves. As a consequence, the agent must learn from its own experiences. To achieve this, a performance measurement criterion is provided to the learner for evaluating the results of actions and is usually in the form of notion of maximizing the cumulative reward. The learner, thus, has the knowledge of possible actions that can be performed but a complete view of the environment is usually not known. It senses its state in the environment and performs a sequence of actions to reach a goal state. The feedback or reward for performing an action is delayed as it does not know immediately if it has performed a correct action or a wrong decision has been made. This dependency on sequence of actions and delayed rewards encourages the learner to explore other possible actions that may improve the overall performance of the system. From evaluating the performance measures, the learner improves its performance over a period of time.

In summary, the characteristic features of reinforcement learning, as described by [Mitchell 97, Sutton 98] are:

Delayed rewards: The task of the learning agent is to learn a mapping from the current state to perform an optimal action that will lead to an eventual goal state. The
training information is available in the form of rewards. As the agent executes its sequence of actions, it faces the problem of temporal credit assignment which is determining those actions in the sequence which are not to be taken into account for generating final reward. The sequential decision making and credit assignment problems, make reinforcement learning a very hard problem and it no longer remains a one shot decision problem.

**Exploration:** During the training process, the learning agent faces a trade-off in preferring to select exploration of unknown states and actions or exploit already learned states and actions that yield high rewards. In order to receive maximum reward, the agent has to exploit those actions that it had already explored and received maximum reward. But it has to explore unknown actions which may result in better overall yield. The agent cannot exclusively pursue exploration or exploitation and it must try a variety of actions to learn in a given scenario.

**Life-long learning:** Since the learner is already immersed in the environment, this setup raises the possibilities to use previously obtained knowledge to reduce complexity when learning new tasks. Thus the learning process of the agent keeps on going.

The reinforcement learning problems can be formally modeled using mathematical framework of Markov Decision Process (MDP). An introduction to MDP is presented next for better understanding.

**Markov Decision Process for Modeling Reinforcement Learning**

In Markov decision process (MDP), environment of a learning agent is modeled as sets of states and actions. At any state, the agent may choose any action from the set of actions possible at that state and as a result the learner moves from one state to another.

The states in MDP observe the Markov property, that has been described as:

\[
\text{“The future is independent of the past given the present”}.\]

In other words, a state \( s_t \) is Markov if and only if it satisfies

\[
P(s_{t+1}|s_t) = P(s_{t+1}|s_1, \ldots, s_t).
\]

(6.1)

This implies that a state captures all relevant information from the history. Once the state is known, the history is not required anymore and the state is a sufficient statistic of the future. Thus, the choice of performing any specific action is independent of any decision that has been made previously.

Formally, an MDP is a 4-tuple represented as

\[
MDP = (S, A, P_{sa}, R),
\]

(6.2)

where \( S \) is a set of states, \( A \) is a set of actions that can be performed at any state, and \( P_{sa} \) is state transition distributions representing the probability distribution of moving to a state \( s' \) when an action \( a \) has been performed at a state \( s \). Since \( P_{sa} \) is a probability distribution, it observes the two conditions of probability, that are,
\[
\sum_{s'} P_{s\pi}(s') = 1 \text{ and } P_{s\pi}(s') \geq 0.
\]  
(6.3)

Finally, \( R \) is a reward function that maps a state to a real number, \( R: S \rightarrow \mathbb{R} \). The learning agent receives the reward \( r_t \) when it visits the state \( s_t \). From the reward function a value function \( V \) can be derived which tries to maximize the overall reward of the learner over a longer period of time.

The goal of the learning agent is to select those actions for which the cumulative reward is maximized. To achieve the goal, the learner performs actions to move to new states and in return it receives rewards. If \( S \) is the set of states and \( A \) is the set of actions then selecting an action \( a \) at a state \( s \) can be represented by a policy \( \pi: S \rightarrow A \). Thus, at any discrete time \( t \) the policy of performing an action is

\[
\pi(s_t) = a_t.
\]  
(6.4)

Since the learner thrives to accumulate maximum reward, the value function \( V^\pi(s) \) can be described by \( V^\pi(s_t): S \rightarrow \mathbb{R} \) and is defined as

\[
V^\pi(s_t) = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots = \sum_{i=0}^{\infty} \gamma^i r_{t+i},
\]  
(6.5)

for an arbitrary policy \( \pi \) from an arbitrary initial state \( s_t \) with \( r_t \) defining the reward received. Here \( \gamma \) is a discount factor and it satisfies \( 0 \leq \gamma < 1 \). A value of \( \gamma \) close to 0 implies that only the immediate rewards are considered whereas setting it close to 1 indicates that future rewards are given more importance as compared to the immediate rewards. The \( \gamma \) also ensures that the infinite sum converges.

For a policy \( \pi(s) \) starting from state \( s \), \( V^\pi(s) \) is then the expected total payoff of starting in state \( s \) and executing \( \pi \). Thus,

\[
V^\pi(s) = E[R(s_1) + \gamma R(s_2) + \gamma^2 R(s_3) + \cdots | \pi, s_1 = s],
\]  
(6.6)

which can be simplified as

\[
V^\pi(s) = E[R(s_1) + \gamma (R(s_2) + \gamma R(s_3) + \cdots) | \pi, s_1 = s].
\]  
(6.7)

Writing the Equation 6.7 in recursive form yields

\[
V^\pi(s) = E[R(s_1) + \gamma (V^\pi(s_2)) | \pi, s_1 = s].
\]  
(6.8)

Solving the expectation gives

\[
V^\pi(s) = R(s) + \gamma \sum_{s'} P_{s\pi}(s') V^\pi(s'),
\]  
(6.9)

where \( P_{s\pi}(s) \) is the probability of executing policy \( \pi \) at a state \( s \) which is the same as performing an action \( a \) at a state \( s \) and the resultant state is \( s' \).
Equation 6.9 is also known as Bellmans equation and provides a way to solve the value function. It imposes a set of linear constraints on the value function. One such equation can be written for every state in the MDP and by solving the linear system of equations, the value function can be calculated.

The optimal value function defined as

$$V^*(s) = \max_\pi V^\pi(s),$$

(6.10)

describes the best possible expected sum of discounted rewards that can be expected by starting from state $s$ and executing maximum possible policy $\pi$.

The optimal value function can be solved by a version of Bellman's equation defined as

$$V^*(s) = R(s) + \max_a \gamma \sum_{s'} P_{sa}(s')V^*(s'),$$

(6.11)

stating that $V^*(s)$ can be solved by selecting actions that maximizes the total expected values. The optimal policy $\pi^*$ can be defined as

$$\pi^* = \arg \max_a \sum_{s'} P_{sa}(s')V^*(s'),$$

(6.12)

stating that the optimal policy can be obtained by selecting an action which results in maximum value function.

**Solving MDP for Generating Policy**

There are three fundamental classes of methods that are primarily used for solving an MDP [Sutton 98]. These are Monte Carlo methods, dynamic programming, and temporal difference learning.

Dynamic programming algorithms require a complete and accurate model of the environment. Monte Carlo methods are more suitable for step-by-step incremental computation and do not require a complete model. The temporal difference methods do not require any model at all and are fully incremental. Consequently these are more complex to analyze.

The two most popular dynamic programming methods for solving MDP are value iteration and policy iteration. The basic idea behind value iteration is that if an agent knows the true value of each state then decision for the next state would be simple that is always choose the action that maximizes the expected utility. The true value (or utility) of a state is the immediate reward for that state, plus the expected discounted reward if the agent acted optimally from that point on. Algorithm 6.1 describes the update of value at each state.

The value iteration algorithm works fine, but it has two weaknesses. Firstly, it can take a long time to converge in some situations, even when the underlying policy is not changing. Secondly, it is not directly determining the optimal policy for the given states and evaluates the values at each state instead. To overcome these issues, policy iteration is used that start with a random policy and compute each state’s utility given that policy, and then select a new optimal policy. Algorithm 6.2 describes the steps of policy iteration.
Algorithm 6.1: Value iteration method for updating values at each state

Initialize $V(s) = 0$ for all $s$
Choose any initial state value function $V_0$

for all $n \geq 0$
do
  for all $s$
do
    $V_{n+1}(s) \leftarrow \max_a R(s, a) + \gamma V_n(T(s, a))$
  end for
Until convergence
end for

Algorithm 6.2: Policy iteration method for solving MDP

initialize $\pi$ randomly
Choose any initial policy $\pi_0$

for all $n \geq 0$
do
  Compute $V^{\pi_n}$
  Choose $\pi_{n+1}$ greedy with respect to $V^{\pi_n}$
  Until $V^{\pi_{n+1}} \leftarrow V^{\pi_n}$
end for

A major drawback of the dynamic program algorithms is that they perform operations over all the states of the MDP, thus it can be extremely computational expensive for MDPs with large number of states.

Monte Carlo methods learn by directly interacting with environment and require only sequence of states, actions, and rewards. These algorithms learn from complete sample returns and therefore only defined for episodic tasks. Moreover, Monte Carlo methods sometimes have insufficient exploration issues. The system does not perform those actions that have never been selected in the past and thus some states might remain unexplored.

The temporal difference learning is a combination of Monte Carlo and dynamic programming. It does not require the model of the environment and learns from direct interaction with the environment. Moreover, it updates the estimates based on partially learned estimates with waiting for final outcome. Two of the most popular algorithms of temporal difference learning are SARSA (State-Action-Reward-State-Action) and Q-learning. The difference between SARSA and Q-learning is that the former one waits until an action is taken and then it updates the value from that action. In contrast, Q-learning updates the best or maximum value from the state it reaches. Thus Q-learning always gives a better estimate of the value from the current state.

From the above description, it becomes clear that in order to develop a life long learning process for a mobile robot to find an elderly person in the home environment, the scenario can be modeled as Markov decision process where policy may represent the possible locations to visit in order to search the person. The MDP then can be solved using temporal difference learning algorithms to have the benefits of both dynamic programming and Monte Carlo methods.
6.4 Modeling of Human Search as Markov Decision Process

An autonomous mobile robot usually has a set of sensors to observe the state of its environment and a set of actions it can perform to alter this state. The task of the robot is to perform sequence of actions or navigate to different states in the environment, observe the state or search the human with the sensor system and learn a control policy for future use. The reward in the scenario of finding the human is determined by the probability of finding the person from that state.

A desired control policy is one where robot navigates from one state to another in search of humans and either find the human or declare that the human is not present in the home. To achieve such a policy, the robot has to accumulate rewards associated with the states and maximize the cumulative reward for the sequence of actions. Besides rewards associated to the states, there is a cost for navigating to the states. This navigation cost can be in terms of energy, distance traveled, time taken or derived from the given scenario. Thus the task of the robot is to find the human as soon as possible to retain the maximum reward from a state. Nevertheless, in case the robot does not find the person at the first location/state, it has to traverse to other location/state in the policy for observing the rest of the environment.

In case of generating a strategy for navigating in the home environment, the robot has to interact with the environment to determine the location of the human. The robot does not receive a feedback unless it finds the human in the environment. The robot receives reward in the form of probabilistic belief about the presence of the person at different times [Mehdi 14].

For searching the elderly person in the environment, certain locations are taken as reference points or View Points (see Section 5.2 for determination of View Points). The View Points work as the destinations for the robotic navigation and search the human around these locations. The probabilities of presence of human being are calculated corresponding to these View Points. These probabilities are multi-model probabilities representing the presence of a person at a place at a particular time of the day. These probabilities are updated on the basis of successful search of the person at different time intervals throughout the day.

When the robot “feels” a need to find the human being, it computes the cost of reaching these destinations from its current location and uses the probabilities of presence of the person at these points at that particular time to evaluate a policy to drive to one of the View Points. The “feel” of the robot has been realized as behavior of Integrated Behavior Based Control architecture (iB2C) where a behavior is defined as $B = (f_r, f_a, F)$. The $f_r$ is Target Rating that defines the satisfaction level of a behavior, $f_a$ represents the Activity of a behavior and $F$ implements the Transfer Function of a behavior. Detailed information about iB2C can be found in Appendix B.

6.4.1 Behavior for Human Daily Routine based Search

The module of Human Search is implemented as one of the behavior that gets self-activated when the Target Rating, $f_r$, increases with the passage of time. The $f_r$ is computed by
\[ f_r = \frac{\text{Time Elapsed Since Last Search}}{\text{Maximum Wait Time}}, \]  
(6.13)

where, \textit{Time Elapsed Since Last Search} is the time elapsed since the robot has last seen the person. \textit{Maximum Wait Time} is the maximum time the robot can stay contented without observing the person. Certain actions are taken with the change in the values of \textit{Target Rating}, \( f_r \), are

- \textit{Target Rating} is less than \textit{lower threshold}: In this do nothing and just wait.
- \textit{Target Rating} is greater than or equal to \textit{lower threshold} but less than \textit{upper threshold}: In this case start calculating the \textit{Navigational Costs} between different \textit{View Points}.
- \textit{Target Rating} is greater than or equal to \textit{upper threshold}: In this case evaluate the \textit{Transfer} function.

The calculation of \textit{Navigational Costs} is deferred towards the end of the waiting time. This delay is necessary in order to accommodate any observable changes in the environment to be taken into consideration before evaluating the \textit{Transfer function} \[\text{Mehdi 11a}\].

The \textit{Navigational Costs} to the selected \textit{View Points} are calculated using A*-Algorithm \[\text{Hart 68, Russell 95}\] described as

\[ f(n) = g(n) + h(n), \]  
(6.14)

where \( g(n) \) is the path cost from the start node to node \( n \), \( h(n) \) is the estimated cost of the smallest path from \( n \) to the goal and \( f(n) \) is the estimated cost of the smallest cost from start node to goal node (see Section 3.1.5 for more details).

The A*-algorithm seems to be the most logical choice at this stage as there are few goal locations and the Euclidean distance based heuristic function results in an efficient cost calculating mechanism \[\text{Mehdi 10}\]. Since the main purpose of the strategy is to find the human being, therefore, priority is given to those places that have highest probability of presence of person. But in case when the difference of probabilities is insignificant then \textit{Navigational Costs} are considered as the decisive parameter.

The \textit{Transfer Function} \( F \) of human search is formulated using MDP to determine the navigation policy from the current location of the robot. Equation 6.2 defines the MDP as a four-tuple. Rewriting it in terms of parameters according to the human search scenario gives

\[ MDP = (S, A, T(S, A, S'), R(S, A)), \]  
(6.15)

where \( S \) is the current state, \( A \) is the action performed at \( S \), \( S' \) is the resulting \textit{View Point}, \( R(S, A) \) is the reward of performing \( A \) at \( S \) and \( T(S, A, S') \) is the probability of reaching \( S' \) when \( A \) is performed at \( S \). The value function \( U(S) \) for the state \( S \) is defined by Equation 6.16 and the policy, \( \pi(S) \), for an action \( A \) to be performed at state \( S \) is determined by Equation 6.17.
6.4. Modeling of Human Search as Markov Decision Process

\[ U(S) = \max_A \left( R(S, A) + \gamma \sum_{S'} T(S, A, S') U(S') \right). \] (6.16)

\[ \pi(S) = \arg \max_A \left( \sum_{S'} T(S, A, S') U(S') \right). \] (6.17)

In the current scenario, \( S \) is taken as the current position of the robot where it is not able to find the human being. \( A \) are the possible locations that can be reached from \( S \) to find the human being. The result of performing \( A \) at \( S \) is a new location \( S' \) where it is most likely that the robot may find the person. The choice of new location \( S' \) is dependent on the reward value and the Navigational Cost.

For the current implementation, the reward \( R(S, A) \) is the probability of finding the person, \( P(S') \), when \( A \) is performed at \( S \) to reach \( S' \). The \( P(S) \) is the probability of finding the human being at \( S \) at a particular time.

To incorporate the Navigational Cost along with the probabilities in determining the new state, \( T(S, A, S') \) is taken as a normalized value of \( P(S') \) times the inverse of Navigational Cost to reach \( S' \). Using Navigational Cost along with probabilities is specially beneficial when more than one location have almost the same probability. In such scenarios, the preference is given to those locations which have less Navigational Cost.

Equation 6.17 is used to determine the policy, \( \pi(S) \), for performing an action \( A \) at location \( S \). The evaluated set of actions or View Points to reach \( A \) is then navigated by the robot to find the human in the environment.

With all the parameters defined for reinforcement learning, an optimized policy is generated using Q-learning algorithm [Watkins 89]. The \( Q(s,a) \) in this case is a tabular structure which maintains the information about which View Points exists connected to each other and thus reachable. It also represents the values at different states after receiving reward. The policy is then a collection of View Points that the robot should visit in order to find the person. Algorithm 6.3 presents the Q-learning algorithm used for computing optimal policy for searching the human in the environment.

**Algorithm 6.3**: Q-Learning for generating optimal policy for searching the human in the home environment.

For each state-action pair \( (s,a) \), initialize the table entry \( Q(s, a) \) to zero

- Observe the current state \( s \)
- for all Do forever: do
  - Select an action \( a \) from the set of possible actions and execute it
  - Receive immediate reward \( r \) from the resulting state
  - Observe the new state \( s' \)
  - Update the table entry for \( Q(s,a) \) as follows
    - \( Q(s, a) \leftarrow r + \gamma \max_{a'} Q(s', a') \)
    - \( s \leftarrow s' \)
- end for

The algorithm guarantees that the approximation will converge to the true Q-function, but every state-action pair must be visited infinitely many times.
6.4.2 Behavior for Last Known Location based Search

Sometimes it is more feasible to find the elderly person at a place where he was last seen. Therefore, rather than navigating to different locations at home, a quick strategy would be to navigate to the location where the human was last seen. This behavior maintains the information of the human being in the environment. In case the time difference between two successive searches is not sufficient, the Target Rating of this behavior is at maximum, see Equation (6.18).

\[ f_r = \frac{1}{||\text{current time} - \text{last seen time}|| + 1}. \] (6.18)

The Transfer function for this behavior is very simple as it only maintains the location from where the person has been seen and thus just forwards the saved locations as destination.

6.4.3 Fusion of behaviors

The fusion of the behaviors for searching the human is carried out using Maximum Fusion (winner takes all) strategy. The behavior with the maximum activity is chosen as the winner behavior. The final activity, target rating and control values are calculated by

\[ \vec{u} = \vec{u}_s, \text{ where } s = \arg \max_c (a_c), \] \hspace{1cm} (6.19)
\[ a = \max_c (a_c), \] \hspace{1cm} (6.20)
\[ f_r = f_{rs}, \text{ where } s = \arg \max_c (a_c). \] \hspace{1cm} (6.21)

The maximum fusion results in selecting of a behavior with maximum activity value. In this way selection between last known search behavior and human routine based searched behavior can be easily made which is not possible with weighted average fusion or weighted sum fusion (see Appendix B for more details). As a result the robot navigates to a destination selected by the fusion of behaviors.

The search based on last known location is activated when the caregiver or operator intends to search the human in the environment and manually selects the option of searching. In this way, the time elapsed since the last autonomous search is smaller and the chances are higher that the human might still be at his previous location. In general, the autonomous search based on human daily routine is performed after every hour and, therefore, the fusion behavior mostly selects human daily routine based search.

6.5 Human Detection

The next step in regards to finding the human after determining a possible location in the home environment is to detect the human. The main focus of this thesis is to develop a strategy for robot navigation which may help in human search in the home environment. Therefore, human detection is not a major part of the work.

The human is detected in the environment using images obtained from the PTZ camera installed on the robot. The detection is realized by either finding the human face or the human in standing position. Since finding the human being is carried out using image
processing, limitations of image processing algorithms sometimes hinder the detection of the person in the real environment. Currently, a Haar cascade classifier [Viola 01] and History of Oriented Gradients (HOG) [Dalal 05] have been used to detect human being in the images captured using PTZ camera of the robot.

An OpenNI framework based human detection has been implemented for detecting pose of humans in the home environment [Fett 13]. The implementation is based on NITE library and Kinect camera. The NITE library provides the angles and position of different body parts of the person in-front of the camera. In some cases these are not valid and provides unrealistic values. In order to avoid such situations, different restrictions have been imposed. Based on the angle between vector from shoulder to the hips and vector from hips to the knees, two human postures namely standing and sitting can be easily identified. Similarly, by analyzing the location of different body parts, it is possible to detect the lying person. Some results of the developed system has been shown in Figure 6.16.

The human detection module is not fully integrated to ARTOS and therefore experiments in the real environment are performed without using human detection.

6.6 Experiments and Results

As described in Section 3.2 a real home like environment, Figure 3.8 has been established at IESE Fraunhofer Kaiserslautern, Germany, to evaluate the performance of the developed robotic solution for finding the human.

The test scenario is that the autonomous mobile robot, ARTOS, has been placed in the home and it has to find the human being at different times in the living environment. The human moves from one room to another according to specific set probabilities. The location of human in the room not specified and he can be anywhere according to his choice.
In the following experiments the learned probabilities of presence of human in different rooms at various times have been given to the robot. This has been done due to the fact that learning cannot be performed in real environment for longer period of time, however, in simulation it is more feasible. The autonomous learning of these probabilities by the robot in simulation has been explained in Chapter 7.

Initially, the robot is navigated in the environment to generate the map of the environment. Details on building map of the environment has been described in Section 3.1.3. Once the map is generated, View Points are calculated for determining the key locations in the environment to monitor and find the human. The generation of View Points is explained in Chapter 5. For human search experiments, Sensing Range of 10 cells has been used in the map of real environment (see Figure 6.17). View Points are given names based on their location in the environment for future reference. Since, ARTOS do not turn on spot autonomously rather follow a trajectory to take the turn, it is difficult for ARTOS to maneuver properly in kitchen. Therefore, the View Point determined in the kitchen has been excluded from searching the human. Once the map and the View Points are generated, ARTOS is ready to perform the search of human in the home environment.

The first step in the direction of finding the human is to determine the navigational cost from the current location to the possible View Points. Since the starting location can be anywhere in the home, validating the cost and the determined policy afterwards would be very difficult. Therefore, the starting location is also set to be one of the View Points. In this way, it becomes very easy to validate the policy required to find the human. The navigational cost from one View Point to the others is given in Figure 6.18.

The probabilities of presence of human viewable from these View Points are given in Figure 6.19. These probabilities clearly show the movement of human being in different rooms at different times of the day.

Given the above mentioned probabilities and distances between the View Points, the policy is calculated using Markov Decision Process for finding the human. Figure 6.20 shows the
6.6. Experiments and Results

![A* Algorithm based Navigational Costs between the View Points](image1)

**Figure 6.18:** A* Algorithm based Navigational Costs between the View Points

![Human presence probabilities in different rooms at different times of the day](image2)

**Figure 6.19:** Human presence probabilities in different rooms at different times of the day
Figure 6.20: MDP policy at 14:00 Hrs on four consecutive days for finding the human after 56 weeks of learning. The red color shows that the human has been found at these View Points. Finding the person at either current location of the robot or at “Location 1” of the policy clearly shows that the robot has learned the human routine and thus able to find the person by visiting least number of View Points in the home environment.
Figure 6.21: Human found by ARTOS at various places in the home environment at different times of the day. The images are from PTZ camera installed on the robot. Haar cascade classifier [Viola 01] to detect human face (red box in the images) and HOG [Dalal 05] to detect standing human (green box in the images) have been used to detect human in these images.
6. Human Location Estimation Based on Daily Routine

Figure 6.22: Elaboration of policy generation based on both probability of human presence and Navigational Cost. Size of the bubble represents the Navigational Cost. Both (a) and (b) shows the situation when location with higher probability (Red color) was not selected rather location with low Navigational Cost was selected (Green color) and human was found at the green colored location.
policies at 14:00 Hrs of four consecutive days after learning and searching for 56 weeks (see Chapter 7 for complete learning process of human routine.). The policy contains the current location of the robot along with 4 other locations to visit in a sequential manner. The red color shows that the human has been found at that particular View Point. It can be easily seen that the starting location of the robot is different in three of these cases and also the human has been found at three different locations. Finding the human at either the current location of the robot or at the “Location 1” of the policy clearly shows that the human routine has been learned over the period of time and MDP has used human presence probabilities to generate an effective policy.

In several experiments conducted at the living environment, ARTOS was able to determine the location of the human being at different times of the day. In some situations, it finds the human in the very first case of policy but in other situations it has to navigate to other destinations before it could find the human. The first attempt is the best case scenario where the robot determines a policy, follow it and find the human from the very first location where it has navigated. In case a human is detected, searching of human being is inhibited and the robot stays at that location for some specific period of time. If the human is not found, the process of searching the human is resumed. Figures 6.21 shows the result of finding the human at different times, at various locations and in different postures.

Figure 6.22 shows the Navigational Cost of reaching a View Point from current location of the robot and is represented by the size of the bubble. The smallest bubble shows the current location of the robot which does not have any Navigational Cost. This figure depicts two cases where MDP has generated a policy by giving preference to a View Point which is closer to the current location and thus has lower Navigational Cost instead of a View Point with slightly higher probability of presence of person. This selection of location closer to the current location has actually paid-off as the human was found at the nearest location with relatively low probability. Here it is evident that the result of the product of probability and inverse Navigational Cost has an in impact on decision making and the outcome is a destination where human has been successfully found.

6.7 Discussion

The successful results of experiments performed in real environment at IESE Fraunhofer prove the concept of using daily routine of a human to search the person in the home environment with an autonomous mobile robot. The routine of the individual may vary from time to time and to accommodate this change a life long learning process is required. In the current scenario it has been demonstrated by modeling the problem as a Markov decision process (MDP) and using reinforcement learning technique such as Q-learning to learn the routine of the person.

The search of human is carried out using the probabilistic analysis of presence of human being in the environment. The Markov decision process (MDP) is being used to generate the policy for navigating autonomously to different View Points in the environment to search the human being. The results obtained in the real environment depicts the importance of probability of presence of human being and the Navigational Cost to find the human being as these are the two deciding factors for calculating destinations. A
successful search results in updating probabilities of finding the human at that particular location, which is required in order to learn from the habits of an individual elderly person.

The generated policy incorporates all the possible locations that are needed to visit in order to find the human. This ensures that in case a human is not found at the first location, robot can navigate to other locations to find the person. In case all the places have been visited and the human has not been found, then the robot assumes that the human is not in the environment and resumes searching after a while.

The human detection using Haar cascade classifier and HOG is very much dependent on the lightening conditions in the environment and pose of the human. These algorithms also detects a picture of human as a real human. To eliminate such false alarm algorithms dealing with depth images [Saleh 13] can be integrated to the system.

Experiments performed in simulation provides more insight to the developed approach. These have been discussed in Chapter 7. The analysis of these experiments has also been presented after explanation of the simulation setup.
7. Experiments

This chapter focuses on the experiments performed to validate the implemented methodology of searching a human based on daily routine of the person. For evaluating the process of learning probabilities of human presence in different rooms in home environment, the robot has to navigate in the environment for several months. Moreover, to analyze the scenarios where algorithm needs further improvements it is necessary to regenerate the same experiments for optimizing the output.

To satisfy these requirements, a simulated environment has been created where the robot can maneuver and learning process can be carried out under various conditions. A simple simulation might not able to capture all aspects of a real home, therefore, a more complex simulation is required that mimics the real home like environment both in its static and dynamic natures. An additional benefit of using such a simulated environment for learning is that if the daily routine of an elderly person is already known then the robot can learn it in simulation and tremendously reduce the learning process in the real environment.

In the following an account of experimental setup has been provided along with details of simulated environment and realization of static and dynamic objects in the simulation.

This chapter will describe the experiments and their results. Firstly, the View Points will be evaluated for their area coverage. Secondly, the learning methodology will be evaluated which enables the robot to learn the elderly person’s routine in order to facilitate the search process. The evaluation criteria will be how much the learned routine correlates with the person’s original routine. Thirdly, the search technique itself will be evaluated. The evaluation criteria will be how accurately the methodology guided the robot to search the human to the right place at right hour of the day. Before going into details of these experiments, an account of experimental setup has been provided in the following.

7.1 Experimental Setup

The simulated environment for the experiments is based on SimVis3D framework, see Appendix C for more details. The complete 3D model of the environment and layout of

[^http://rrlib.cs.uni-kl.de/software/simvis3d]
the real apartment are shown in Figure 7.1. It consists of a home setup with furniture placed in different rooms, a simulated mobile robot that can be controlled to navigate from one room to another and a simulated human that is controlled to walk from one room to another at different times of the day. In simulation laser-scanner and Pan-Tilt camera are realized as sensor system of the robot. It is important to mention, that all the implementation on ARTOS works in simulation by only changing the hardware abstraction layer with the simulation layer [Mehdi 11b].

The more interesting component of simulation environment is simulation of human and has been discussed in the next section.

### 7.1.1 Simulation of Human

In real life scenarios, there are humans that influence the environment and they themselves are dynamic in nature i.e. move from one location to another. In order to generate a simulated environment that may be used for testing the search of human in the environment, it is mandatory that a simulation of a person is created that may move from one room to another like a real human and depicts the characteristics of an elderly person living in the home.

Modeling of human character or avatar is based on well known human modeling standard H-Anim. This standard describes specifications for defining interchangeable human figures.

7.1. Experimental Setup

Figure 7.2: Dynamic postures of simulated human character (a) Intermediate posture for falling human, (b) Human fall and (c) Sitting posture. These motions have been generated in Blender and have been imported to the simulation for realistic human movements during experiments in simulation. The selection of these postures in simulation is based on Markov chains.

Various versions of H-Anim have been published and several characters based on this standard are already available. The choice of this standard offers the possibility to use new avatars without changing already defined motions.

H-Anim standard defines three levels of articulation (LOA) for accommodating needs of various applications. LOA 0 is the minimum H-Anim humanoid with only HumanoidRoot and LOA 3 is full H-Anim Hierarchy for controlling all possible joints. In the current scenario, LOA 3 has been taken for modeling various human motions. The modeling has been performed using Blender which supports the most detailed LOA. The motions are modeled by specifying key frames in a time-line. At these key frames, positions of various body joints are defined using drag and drop and inverse kinematics tool of Blender. An export plug-in for simulation framework is used to interpolate the specified key frames and export it as a sequence of joint angles after modeling is completed.

The human motions are categorized into two categories namely simple movements and complex movements. Simple movements consists of those movements which have been generated using Blender. These movements are independent of each other and have a definite time of execution. Complex movements, on the other hand, consists of combination of simple movements and takes place one after the other. Examples of developed movements have been shown in Figure 7.2. Based on the motion files exported with Blender’s export script the connection between the avatars and different motions has to be realized. Each avatar in a scene is connected to a motion module which controls all activities that an avatar can perform. More details about simulation of a human character complying to H-Anim standards as discussed in [Schmitz 11, Hirth 11].

Till this stage the motions are independent of each other and each sequence ends in a finite time. For more humanly movements, the generated movements need to be combined together to perform a meaningful and desirable action. One approach would be to combine the movements randomly. This could result in a chaotic movement pattern like walking right after falling on the ground without getting up; which in real life includes first standing up after falling and then walking.

In order to avoid disorderly movements of simulated human character, Markov chain has been employed. A Markov chain consists of a set of states and set of probabilities describing the transition between states [Norris 98]. The states in Markov chain observe
the Markov property i.e. the next states are independent of previous states given only the current state (see Section 6.3.1 for more details about Markov property). Given a state space of $n$ states, represented as $S = \{s_1, s_2, \ldots, s_n\}$, the process starts in one of the states and moves successively from one state to another. This movement is usually called a step. The movement is based on probabilities associated to states, usually denoted by $p_{ij}$ and is independent of any probability before the current state. These probabilities $p_{ij}$ are called the transition probabilities and the process changes the state based on these probabilities.

Figure 7.3 shows the Markov chain for selecting different movements to form a sequence of motion for the simulated human character. The shapes of different human posture represents different states of the Markov chain. These states corresponds to standing, sitting, walking, and falling down. The edges of the directional graph are labeled with the state transition probabilities. The state transition probabilities from sitting, walking, and falling down to standing state is 1.0 indicate that if in order to switch among these movements, it is necessary that the character is in standing position. The use of Markov chain ensures that selection of different movements will not result in unrealistic movement and thus co-occurrences of movements are regulated.

In the simulation, a simulated human character walks to different rooms according to a pre-defined set of probabilities that replicates the presence of an elderly person in the home environment. These probabilities are unknown to the robot and will also act as the benchmark for evaluating the learned behavior of the robot.

Besides body movements, for performing experiments regarding searching of human in the environment, it is also required that the simulated character walks to different rooms. A random walk to different rooms will not convey the effect of a human in an environment. Therefore, once again, Markov chain is used to simulate the human behavior of moving from one room to another at different times of the day. In this scenario, rooms are treated as states of the state space and probabilities of human being present at different times of the day in different rooms are state transition probabilities. Using these probabilities make it possible to move the simulated character based on some pattern that represents the real human being and thus the movement to different rooms is not completely random.

Since a real human can be anywhere within the room, therefore, the destinations for the simulated character within the room are also kept random. This randomness within the
room also creates situations where the avatar is out of the sensing range of the simulated robot even if both are in the same room. It is a desired situation as in real life scenarios, it is possible that the real robot may not able to detect the human due to occlusions, lightening conditions and limited sensing range.

Currently, obstacle detection and avoidance is not incorporated for the simulated character and therefore sometimes unrealistic behavior like passing through furniture and walls is also observed. Such functionality needs to be worked on in future. Moreover, the motions are combined with each other only in a sequential manner. The parallel movements like waving hands during walking can also be realized using the developed framework of Markov chains.

### 7.1.2 Data Sets

For training in simulation, a data set is required that represents the location based daily routine of a person in the home environment. It must contain hourly information of human presence in different rooms and should consists of 24 hours for completeness. Currently, there are no comprehensive datasets available that maintain the information of an elderly person in various rooms at different times of the day. Some data sets like [Riboni 09] contain activities performed by an elderly person through out the day. These involve both outdoor and indoor activities. This data set does not maintain any hourly and room based information. Similarly, the data set available at [Krantz-Kent 07] records hourly activity information of a person but room information is not available. Moreover, this data set does not only related to the elderly person as it takes into account activities of person of age 15 or more. To obtain information about room occupancy by a person at different times of the day, the data sets need to modified. Therefore, the data sets used for training is based upon the [Riboni 09] and [Krantz-Kent 07] data sets but are also modified to generate room level information at various times of the day. The most important modification in the data set is to divide different tasks that can be performed in different rooms as it has not been mentioned in these data sets. As an example, sleeping is confined to bedroom only. Although, this activity can also be performed in TV room. Similarly, house keeping related tasks are distributed to TV room. For this, the TV room has been divided into three sub-rooms, namely TV-Kitchen, TV-Room and TV-Corridor (see Figure 6.17). These two modified data sets can be found in Appendix A. The probabilities in the data sets are used for moving human character in the simulated environment and are unknown to the robot.

In order to evaluate View Points, maps of different environments have been used with various Sensing Range. The values of Inner Circle and Safety Zone are kept constant to 8 cells (80 cm) and 4 cells (40 cm) respectively that are dictated by the dimensions of the robot. The experiments conducted for the evaluation of learning human routine are based on the map of the real home like apartment at IESE Fraunhofer. In these experiments Sensing Range of 10 cells have been used.

### 7.2 Evaluation of View Points

The View Points have proved to be beneficial for searching an elderly person in the home environment. They provide suitable locations to the robot from where it can view the person and also safely navigate from one View Point to another.
In order to make the experiments close to real world, map of the real environment has been used in simulation, (see Figure 7.4). In this experiment the View Points have been calculated using Sensing Range (SR), Inner Circle (IC) and Safety Zone (SZ) as 10 cells (100 cm), 8 cells (80 cm) and 4 cells (40 cm) respectively.

In real life home apartment, not all the areas are completely visible and reachable by a mobile robot. Therefore, the proposed methodology of View Points deals with the constraint imposed by the limited range of visibility by the sensor system of the robot. It also takes into account dimensions of the robot and generate View Points according to characteristics of an individual robot. It is important to mention, that the main purpose of the View Points is to provide suitable locations for maximum observability from certain locations in the environment and it is not designed for full area coverage of the environment.

A cell is marked viewable, if and only if it is completely within the range of the sensor system of the robot. Otherwise it is treated as a non-viewable from a View Point. This is a strict constraint and leaves cells at boundary of the Sensing Range as not being viewed from that View Point. Such strict constraint ensures a more reliable system with greater accuracy in finding the human within the Sensing Range.

The algorithm determines the View Points based on priority. Thus View Points that cover more area and providing better navigational space around them, i.e. Primary View Points, are given higher priority as compared to those View Points that are close to obstacles, i.e. Secondary View Points, and thus are given lesser priority. In the experiments performed, the Primary View Points cover up to 80% of the environment which is quite enough for human search scenario, therefore, Secondary View Points were not taken into account for searching the human. As mentioned earlier, ARTOS do not turn on spot autonomously rather follow a trajectory to take the turn. Due to this navigation limitation, it is difficult for ARTOS to maneuver properly in kitchen. Therefore the View Point determined in the kitchen has been excluded from searching the human.
The developed methodology is robust and does not depend on the starting location of the robot. Moreover, the map also does not need to be a closed shape. The methodology only generates View Points on the area which have marked as free in the map by the sensor system of the robot during mapping process and has been observed by the robot.

Generally, the number of View Points decreases with the increase in Sensing Range. Figure 7.5 shows the results for the above mentioned home environment scenario. The number of Primary View Points do not change much with the increase in Sensing Range (SR). It is due to the fact, that the environmental structure used in the experiments is divided by obstacles into specific regions and to cover such an environment, therefore, a certain minimum number of Primary View Points is always required. As can be expected, the Secondary View Points are effected to a great extent by extending the Sensing Range. This number is directly dependent on the obstacles in the environment which hinders the view of the robot at Primary View Points.

Results have also shown that the area covered increases with the increase in Sensing Range at the first level (Primary View Points) as can be seen in Figure 7.6. It is also evident from the Figure 7.6 that after second level, area coverage is more than 90% with most of the Sensing Ranges which shows the impressive performance of the developed methodology. It also provides liberty in selecting the number of levels depending upon the desired area coverage.

Moreover, it can also be clear seen that only 4 to 6 Primary View Points are sufficient to observe up to 80% of a typical home environment of around 60m² consisting of multiple rooms and cluttered with various furniture.

Various experiments have been performed with maps of other environments. Table 7.1 shows the free area in these environments along with number of View Points required to observe the free space. The time taken for computing all the View Points have also been provided. Environments with less number of obstacles and more free space results
in better observability. The obstacles block the field of view of the sensor and therefore more View Points are required to cover the area. The map of RRLAB shows the largest free space among all other maps and thus require significantly more time to compute all the View Points to cover the environment. Nevertheless, with all the determined View Points in the RRLAB, more than 95% area is visible from these View Points. Moreover, the area covered by the View Points is more than enough to find the human in a typical home environment. Further details about the results corresponding to these environments have been described in Chapter 5.

7.3 Evaluation of Learning Human Routine

The simulated environment provide the freedom to expedite the experiments by manipulating time. In the current scenario, one hour of the real world has been mapped to 2 minutes in simulation. In this way, learning human presence probabilities with simulated human moving from one room to another at different times of the day has been performed in about 13 days which would have taken about 56 weeks or an equivalent of about 9408 hours if performed in real world. This reduced time in simulation proved to be sufficient for the human character to move in different rooms and the robot to navigate to different rooms in pursuit of human. The times mentioned in the following text corresponds to the real world times for easy understanding though performed in simulation.

Since in the beginning, the robot does not have any information about the presence of the human being in the environment, therefore, the probability of finding the human is equally distributed over these View Points. For every search, a new policy to find the human is generated using MDP. The policy consists of an ordered set of View Points as destinations for searching the human. The robot follows this policy by navigating to these View Points in a sequential manner. When the human is found at a View Point, the probability of human presence is increased at that View Point for that particular hour of the day. With
7.3. Evaluation of Learning Human Routine

<table>
<thead>
<tr>
<th>Map of Location</th>
<th>Free area ((m^2))</th>
<th>Total number of View Points</th>
<th>Area coverage (%)</th>
<th>Time taken (\text{msec})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assisted living lab at IESE</td>
<td>30.55</td>
<td>24</td>
<td>88.38</td>
<td>331.47</td>
</tr>
<tr>
<td>Simulation of IESE</td>
<td>38.84</td>
<td>27</td>
<td>95.7</td>
<td>436.13</td>
</tr>
<tr>
<td>Simulation with changed furniture</td>
<td>40.70</td>
<td>31</td>
<td>94.32</td>
<td>435.61</td>
</tr>
<tr>
<td>Simulation without furniture</td>
<td>49.51</td>
<td>35</td>
<td>96.83</td>
<td>802.19</td>
</tr>
<tr>
<td>RRLAB simulation</td>
<td>297.47</td>
<td>190</td>
<td>96.57</td>
<td>4587.96</td>
</tr>
</tbody>
</table>

Table 7.1: Overview of View Point results in different environments with Sensing Range set to 10 cells (100cm). Due to large number of obstacles in the environment, area coverage is less in real environment (Assisted Living Lab) as there are less places that can ensure safety to the robot. Moreover, these obstacles block the field of view of sensors. Thus, less number of View Points are possible resulting in less area coverage. Still this area coverage is more than enough for finding human in home environments.

the passage of time, the robot learns the pattern of human presence in different rooms at different hours of the day.

To illustrate the discussion, the routine learned by the robot of human presence in the bedroom and TV room are being shown in the following. Figure 7.7(a) shows the probabilities learned after 1 week of searching and finding the human in the bedroom. Figure 7.7(b) shows the changes in human presence probabilities learned by the robot after 8 weeks. It can be easily seen that the robot has found human in bedroom several times and thus the belief of human presence has significantly changed from 1st week. Finally, the Figure 7.7(d) shows the probabilities of human presence learned by the robot after 56 weeks which is much closer to the reference routine used by human character for being present in the bedroom shown in Figure 7.7(e). Similar results are shown for TV room in Figure 7.8. Human routine learned from other View Points are being presented in the Appendix A.

The learning process can be evaluated by determining the correlation coefficient between the learned and the reference probabilities. Generally Pearson correlation coefficient is used which is a measure of the strength of linear dependence between two variables and its range is \([-1, +1]\). Values closer to +1 indicates a strong positive linear relationship and −1 refers to strong negative linear relationship. The value 0 implies that there is no linear correlation between the probabilities.

Table 7.2 shows the correlation coefficient between the learned probabilities with the desired probabilities in the period of 58 weeks. The data shows that initially the probabilities are weakly correlated indicating that the robot was unaware of the routine of the person at home. After learning for some weeks the correlation between the probabilities got stronger as indicated by values getting closer to +1. The gradual improvement in value
7. Experiments

Figure 7.7: Learning the presence of a person in bedroom. Initially a uniform probability of human presence in the home is assumed which is updated upon successfully finding the person in a room. These probabilities are updated for every hour of the day. (a) shows the learned probabilities after one week of searching in the home. (b) shows the structure of probabilities started forming in about 8 weeks. (c) the structure gets more refined and become more closer to the human presence probability. (d) final result after 56 weeks of searching and learning the probabilities (e) reference probability of human presence in the bedroom.

<table>
<thead>
<tr>
<th>Correlation</th>
<th>Bed Room</th>
<th>Corridor</th>
<th>TV Kitchen</th>
<th>TV Corridor</th>
<th>TV Room</th>
</tr>
</thead>
<tbody>
<tr>
<td>After 1 week</td>
<td>0.73</td>
<td>0.52</td>
<td>0.73</td>
<td>0.85</td>
<td>0.58</td>
</tr>
<tr>
<td>After 8 weeks</td>
<td>0.96</td>
<td>0.94</td>
<td>0.92</td>
<td>0.96</td>
<td>0.91</td>
</tr>
<tr>
<td>After 24 weeks</td>
<td>0.97</td>
<td>0.97</td>
<td>0.96</td>
<td>0.97</td>
<td>0.93</td>
</tr>
<tr>
<td>After 56 weeks</td>
<td>0.98</td>
<td>0.97</td>
<td>0.96</td>
<td>0.99</td>
<td>0.97</td>
</tr>
</tbody>
</table>

Table 7.2: Correlation of learned human routine with reference to original human presence routine in different rooms of the environment. The correlation gets better as the robot learns more about daily routine of the person after successfully searching the person in home environment. The progress in different weeks shows the correctness in learning process.
Figure 7.8: Learning the presence of a person in TV room. (a) shows the learned probabilities after one week of finding the person in TV room. (b) shows the structure of probabilities started forming in about 8 weeks. (c) the structure gets more refined and become more clear. (d) final result after 56 weeks of searching and learning the probabilities (e) reference probability of human presence in the TV room
of correlation is a clear indication that robot has learned the human routine. Thus, the results of correlation are evident to show that an autonomous mobile robot can learn the human routine by performing search operations in the home environment and support the hypothesis described in Section 1.2.

7.4 Evaluation of Navigation Policies for Human Search

The policy generated using MDP is the guideline for the robot to navigate to different View Points in the environment. It determines the next location for searching the human in the environment from current location of the robot. It is dependent on the probabilities of human presence and the costs to navigate to different View Points.

Initially, all the probabilities are the same therefore MDP generates a policy that is solely dependent on the cost information. This implies that the points that are close together will be visited first than the farthest locations in the environment. The generated policy ensures, that during the search process all the locations have been visited to find the human being. This is mandatory as the smallest probability of presence of person implies that it is possible that the human can be at that location. The robot learns the probabilities with the passage of time and thus resulting in a better policy generation accordingly. After the probabilities have been learned, the result of the MDP are more influenced by the probabilities rather than the cost information. However, in case, the difference between the highest probabilities is minimal, the cost information is used to break the tie and nearer View Point is navigated first.

The evaluation criteria of policy is the fact that the robot should able to find the human in the first location calculated by the policy and do not need to navigate through all the View Points in the environment. Detailed analysis of the situation has been provided in the next section.

7.5 Overall Performance of Developed Methodology

Total Success is generally defined as fraction of success cases among total number of cases. In the case of human search scenario, this definition needs to be modified to represent true success of the system. Therefore it has been defined as:

**Definition 7.1** Success of human search scenario is that an autonomous mobile robot successfully reaches the location from where it can observe the person given that the human is present in the home environment or in case human is not there then it successfully declares the absence of human.

Thus Total Success of the search process can be formulated as Equation (7.1)

\[
\text{Total Success} = \frac{\text{Total Success Cases}}{\text{Total Search Attempts}}, \quad (7.1)
\]
7.5. Overall Performance of Developed Methodology

where **Total Success Cases** includes cases when human was at home and the robot found the person and also cases when human was not at home and robot declared the absence of person.

In experiments carried out in simulated environment, a total of 9401 search attempts were made by Atos to find the human in the simulation environment. Each attempt was made after one hour. Thus, in total it accounts for 9401 hours which is equivalent to about 391 days. In other words, in simulation, the data for more than one year has been generated and analyzed which is sufficient to evaluate the learning performance of the complete system.

During these search attempts, the robot was successful in 9046 attempts to reach the location of the human. There were total 355 attempts when the robot did not find the human in the environment which is about 3.78%. Among the total 355 attempts when robot was not able to find the human, in 105 situations the human was not in the environment and the robot successfully declared after navigating to all **View Points** that human was not at home. Thus, the **Total Success** comes out to be 97.34%.

Although, **Total Success** is a good instrument to measure the success of the system, a better performance evaluation can be made by measuring the **Success Rate** of the overall process and analyzing the reliability of the developed strategy. Therefore, further analysis of the experimental results is given in the following subsections.

### 7.5.1 Success Rate

A more critical way to determine the success of the system is that those cases contribute more towards the success in which the person is found at first location while subsequent location findings contribute less towards the overall success. Moreover, in those cases where human is not at home, the belief can only be established after visiting all the locations in the environment to ensure human absence. Such cases also contribute to the success of the system.

Therefore, the **Success Rate** has been defined as

\[
\text{Success Rate} = \frac{(N_{fnh} + F_{acl} + \sum_{i=1}^{N-1} F_{c_{i}})_{i}}{T_{ns}},
\]  

(7.2)

where \(N_{fnh}\) represents Total number of cases when human was not found given that human was not at home, \(F_{acl}\) is Count of cases when human was found at current location, \(F_{c_{i}}\) is Count of human found cases in \(i\)-th attempt, \(T_{ns}\) is the Total number of searches, and \(N\) are the Total number of locations in the home environment from where the robot can observe the human.

Since, the developed methodology gives multiple successive locations based on human routine where it is most probable to find the person at that particular hour of the day; **Success Rate** should increase significantly if human is observable either from the current location of the robot or it has to navigate to minimum number of locations to reach the person. In case it has to navigate to all the locations before reaching the person then, though successful, the human routine was not learned properly for that scenario.

Based on the definition of **Success Rate** (Equation 7.2), results of the experiments are as follows. Among the total 9046 successful search attempts, in around 23.92% cases (2164
times) the robot was already at the View Point from where it can see the person and about 33.17% cases (3001 times) it has found the person at the first View Point of the evaluated policy. The percentages of human found in second, third and fourth View Point in the evaluated policy are 21.70% (1963 times), 15.82% (1431 times) and 5.38% (487 times) respectively, which represents that the robot did not find the human after traversing to the first View Point of the evaluated policy and it has to move to other View Points to find the person.

From the above mentioned numbers, the Success Rate of the current scenario comes out to be 0.7287 or 72.87% which represents that in some cases robot has navigated to other View Points as the person was not found at first View Point. On the other hand, since human is independent in its movement and the robot has learned only a probabilistic belief of presence of person therefore the performance of the system is significantly high as in most cases it has correctly evaluated the possible location of the person.

7.5.2 Reliability of Searching Human

Reliability generally defined as the ability of a system to perform its functionality in orderly manner in normal scenarios as well as in unexpected conditions. For evaluating the reliability of autonomously searching an elderly person in indoor environment by a robot, the results need to be examined carefully specially for unexpected situations. A normal scenario will be that the person is somewhere in the home and following the daily routine as always whereas an unexpected scenario will be when the human is not present in the home and the robot is trying to search the person. Therefore, to evaluate the reliability of the developed strategy, false positive rate and false negative rate needs to be computed. A smaller value of these rates will prove the strategy to be reliable under various circumstances.

**Definition 7.2** A false positive in the scenario of human search will be a situation when the robot indicates that a person in the home environment has been found although the person is not present at home.

Since, the focus of the work was on developing the strategy for finding the person and not on optimization of human detection algorithms, therefore, false positive does not apply to the current scenario as it will depict the shortcomings of vision based algorithms and searching strategy will not be evaluated. Thus false negative will be more significant in the current situation.

**Definition 7.3** A false negative in the human search scenario will be the case where the robot could not find the person in the environment although the person is at home.

Equation 7.3 has been used to compute the false negatives in the current scenario.

\[
\text{False negative rate} = \frac{\text{False negative case count}}{\text{Total count of human presence at home}} \tag{7.3}
\]

Among the attempts where the human was not found, around 105 times the simulated human was outside the home environment. Therefore, these 105 times are counted towards
the correct evaluation of the situation where the robot indicates that the person is not at home.

In the remaining 250 situations, the human was out of the range of sensors of the robot and therefore could not determine the presence of human. Thus false negative rate is 0.0269 or 2.69%. The results depict that the developed learning strategy worked very well for searching the human in the environment and the robot adapted itself according to the routine of the person for generating a policy to conduct the search.

Table 7.3 shows the overall results of developed methodology of searching a person in the home environment performed over two different data sets. The successful searches show that robot has been able to find the human and success rate describes that during these searches human routine has been taken into account.

The successful results at various stages of the search strategy shows the effectiveness and correctness of the proposed methodology for searching the human in the home environment. The determination of View Points paved the way for observing the environment in a meaningful manner. The learning component shows that a robot can learn the daily routine of a person in the home environment and the human search approach has combined all the components together to demonstrate the validity of the developed methodology.

<table>
<thead>
<tr>
<th></th>
<th>Total search attempts</th>
<th>Total success (%)</th>
<th>False negative (%)</th>
<th>Success rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Set 1</td>
<td>9401</td>
<td>97.34%</td>
<td>2.69%</td>
<td>72.87%</td>
</tr>
<tr>
<td>Data Set 2</td>
<td>9744</td>
<td>97.08%</td>
<td>2.94%</td>
<td>75.11%</td>
</tr>
</tbody>
</table>

Table 7.3: Results of performing search of human in the home environment using two different data sets of human presence probabilities. Total success represents that the robot was successful in finding the person and success rate depicts that the robot has learned the human daily routine effectively and used that information in searching the human.
8. Conclusion

The growing population of elderly people imposes a need to develop robotic solutions to provide care services to these people at their homes. To address the issue, this thesis has presented the work done in direction of serving an elderly person living alone in a home environment using an autonomous mobile robot. The state-of-the-art methodologies provide a set of predefined locations in the home environment to the mobile robot and human-robot interaction is carried out at these locations thus limiting the possibilities of services being offered.

In order to enhance the interaction, the autonomous mobile robot should pro-actively reach the elderly person. In this scenario, the complete home environment becomes a search space for the robot. The human can be anywhere in the search space and thus require an exhaustive search in the environment.

This thesis proposes a methodology for autonomous search of human in the environment that analyzes the environment to determine best viewing locations in the environment and use these locations to find the human. It autonomously develops a probabilistic belief of presence of person in the environment at different times of the day using sensor system installed on the mobile robot. For searching the human, it uses this belief to reach locations already identified as best locations for maximum visibility. This methodology has been evaluated using autonomous mobile robot, ARTOS. The robot is equipped with sensor systems to develop map of the surroundings and localize itself in the environment. These sensors are also used to avoid obstacles during navigation. For autonomous movements, paths are planned and optimized to reach the goal locations (see Chapter 3 for more details).

In the following a brief summary of accomplishments has been described.

**Determining View Points**

For searching human, an autonomous mobile robot has to navigate to all possible locations in the home environment. Visiting all locations is not feasible as some locations might not be accessible and it will take a long time to search at every location. Therefore, a subset is needed that represents the environment which will tremendously speed up the process of searching. This process efficiency is achievable at the expense of missing some locations
from being searched. Thus, the focus of the developed methodology is to identify some key locations, or *View Points*, that can maximize the area coverage with minimum number of search locations.

The autonomous determination of *View Points* is realized by considering the dimensions of the robot and range of the sensors installed on it. These *View Points* serve as basis for finding the human as they offer maximum visibility in the environment. Experiments performed in various environments have shown that with *Primary View Points* and *Secondary View Points*, an area coverage of more than 90% has been successfully achieved which is more than enough for human search scenario. The results in Chapter 5 in clearly show that using combination of *Primary View Points* and *Secondary View Points* not only give better coverage of the environment considering constraints regarding robotic movements but also the *View Points* generated at multiple levels provide control over the number of *View Points* in the environment according to the need of the scenario.

**Learning Human Routine**

From the study of lifestyle of elderly people, it has been seen that they usually have monotonic ways of spending their days and have well defined routines. Thus, the routine of an elderly person seems to be the best criteria to search the person in the home environment at different times of the day. Therefore, the basic idea of methodology developed for searching the person is to learn these routines over a period of time and to improve with every successful search result. The key benefit is that it can adapt to different daily routines of people. Moreover, it can also adapt to changing routines of an individual (see Chapter 6 for more details).

During the search process, the human routine is learned in terms of human presence probabilities corresponding to the already determined *View Points* at every hour of the day. The learning technique is based on reinforcement learning and has been realized using Markov decision process (MDP). Experimental results have shown that even after 1 week of searching and learning, correlation between the actual human routine and learned routine is increased to 0.5. This improves further after learning for 6 to 8 weeks and reaches up to 0.9. These results clearly show that the robot has successfully adapted itself according to the daily routine of the person.

**Searching Human**

The human search in the developed methodology is inspired by human behavior during searching lost objects. It uses the learned human routine to estimate locations where there are more chances to find the person in the home environment. This has been realized by Markov decision process (MDP) (see Chapter 6 for more details).

The Markov decision process (MDP) calculates a policy indicating the sequence of *View Points* to navigate in order to search the human in the environment. The policy is based on the learned probabilities of presence of the person that have been developed by successfully searching the human at every hour of the day at different *View Points* and cost to navigate from current location to the candidate *View Point*. The priority is given to that *View Point* which has higher probability as chances of finding the person are equally higher. In case, when the probabilities are equal or the difference between them is small, the location which is closest to the current location is selected.

Another criteria to find the human is to search the person where he was last seen as there are more chances that he is still there if the time interval between two searches is
small. To coordinate between search based on MDP and last known location, these have been implemented as behaviors of a behavior based network where both of these methods compete against each other and the final decision is taken on the basis of “winner takes all”. Similarly, other search behaviors can be easily implemented and integrated to the developed framework to enhance the overall search strategy.

As a consequence of a successful search, the probabilities are updated and these new probabilities are used for future searches. In case of an unsuccessful search, the next location in the policy is selected as destination. The experimental results in Chapter 7 show that using the developed methodology the robot is able to find the human in more than 95% of the cases. These promising results show the effectiveness and usability of the developed approach.

In conclusion, the thesis has presented a reliable methodology to search a person in the home environment using an autonomous mobile robot. The methodology incorporates the characteristics of the home environment to find locations for best viewing while considering the limitations of the robotic platform. It also establishes the concepts of learning human routine in the home environment and using this learned routine to successfully find the person in the environment. The promising results in real and simulated environments show the overall correctness and applicability of the proposed methodology.

8.1 Future Directions

The work presented in this thesis can be extended in many regards.

The developed human search methodology is based on the daily routine of the person, therefore, a possible extension can be to detect an unusual behavior in the daily life of the person. A deviant behavior may able to predict health issues related to the person which can be reported to caregivers. A similar methodology with initial results have been presented by [Franco 10] where they have tried to detect deviations in human movement data from motion sensors installed in the environment. A predictive measure can save the elderly person from emergency situations that may occur in future.

A reliable human detection technique is necessary for searching the human in the home environment. In a typical home setup, normally parts of human body are visible which makes human detection and posture estimation very difficult. It is also required for detecting emergency situations where human may not be in his normal posture and thus may not be easily recognizable.

The performance of the developed methodology can be enhanced by using a 3D map of the environment. These maps can be easily generated using point clouds. The View Points thus generated will incorporate the height of robot and furniture in the environment to find maximum viewable locations in the home environment which are also easily reachable by the robot.

The developed methodology can be a useful extension to any service robot that has to reach a person in order to serve him. Such a scenario has been demonstrated by [Graf 09b] where a human at a predefined location is served meal by the robot. Other similar scenarios can include reminding the person of taking medicine or an appointment with the doctor, or even working as a butler [Srinivasa 10, Reiser 13] in the home environment. Moreover,
for approaching the person using an autonomous mobile robot, results from [Torta 13] can be utilized in order to respect the personal space of the person.

Currently, the developed human search methodology is uni-model and it maintains information of one person only. This can be extended for searching multiple people according to their daily routine in one home. It can be achieved by extending the developed framework of MDP to a multi-model MDP that generates policies for multiple people. This enhancement will require a framework for recognizing people in the environment besides detecting humans.

The human search can also be triggered by an external event like a sound of falling. The localization of sound has been described in [Schmitz 11]. This will be helpful in emergency scenarios where the human may not be able to call the robot and thus the robot has to be proactive in finding the person by determining the origin of sound.

The developed methodology of location based routine learning can also facilitate the recognition of different activities performed by the person at home. The information for inferring activities can help in devising new strategies to facilitate the person in his daily life.

In conclusion, human daily routine is composed of three components, namely activity, location and time. In this thesis the main focus was to search human based on daily routine comprising of only location and time. The search process can greatly enhance in case all three components of daily routine are taken into account. Future researchers are directed to explore these components for searching humans in the home environment.
A. Human Probabilities and Learned Routine

In order to learn daily routine of a person in the home environment, reference probabilities of presence of the human are required. The data sets currently available focus on the activities performed by a person. These data sets do not relate the hourly presence of human in one particular room. As an example see data sets mentioned by [Riboni 09] and [Krantz-Kent 07]. Therefore, information from these data sets has been extracted and modified in order to generate two sets of data that provide hourly probability of presence of a person in different rooms of a typical home environment. Table A.1 and Table A.2 are two different data sets that are used in experiments to validate the learning of human routine by ARTOS. More discussion about the data sets can be found in Section 7.1.2.

Based on the Table A.1, the mobile robot, ARTOS, learned the daily routine of the person. Using this learned information it tries to find the person in different rooms at different times of the day. Figure A.1, Figure A.2, and Figure A.3 show the results of learned probabilities after successful search of human in the home environment. These graphs show that initially, the robot having no idea of presence of person, learns some routine of the person after 1 week. The variation in the graph is minimal as robot has learned only 7 days. The learning progresses with the passage of time and the perception of human presence gets better and better. These experiments have been performed for about 56 weeks. As can be seen in all the three figures that the final result of 56 weeks resembles very much to the human presence routine in the respective rooms. Thus proving the concept that learning human routine is possible for the mobile robot and can facilitate in searching the person in the home environment. More detailed discussion can be found in Section 7.3.
### Table A.1: Data Set 1: A probable daily routine of a person at home. The probabilities represent in which room it is more likely that the human is present.

<table>
<thead>
<tr>
<th>Bed Room</th>
<th>Corridor</th>
<th>TV-Kitchen</th>
<th>TV-Corridor</th>
<th>TV Room</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.235</td>
<td>0.210</td>
<td>0.009</td>
<td>0.419</td>
<td>0.127</td>
</tr>
<tr>
<td>0.253</td>
<td>0.104</td>
<td>0.014</td>
<td>0.458</td>
<td>0.171</td>
</tr>
<tr>
<td>0.305</td>
<td>0.126</td>
<td>0.179</td>
<td>0.103</td>
<td>0.287</td>
</tr>
<tr>
<td>0.105</td>
<td>0.101</td>
<td>0.392</td>
<td>0.201</td>
<td>0.201</td>
</tr>
<tr>
<td>0.146</td>
<td>0.168</td>
<td>0.170</td>
<td>0.118</td>
<td>0.398</td>
</tr>
<tr>
<td>0.404</td>
<td>0.033</td>
<td>0.024</td>
<td>0.089</td>
<td>0.450</td>
</tr>
<tr>
<td>0.282</td>
<td>0.025</td>
<td>0.234</td>
<td>0.216</td>
<td>0.243</td>
</tr>
<tr>
<td>0.284</td>
<td>0.277</td>
<td>0.224</td>
<td>0.101</td>
<td>0.114</td>
</tr>
<tr>
<td>0.024</td>
<td>0.004</td>
<td>0.458</td>
<td>0.149</td>
<td>0.365</td>
</tr>
<tr>
<td>0.132</td>
<td>0.002</td>
<td>0.375</td>
<td>0.137</td>
<td>0.354</td>
</tr>
<tr>
<td>0.008</td>
<td>0.133</td>
<td>0.392</td>
<td>0.125</td>
<td>0.342</td>
</tr>
<tr>
<td>0.025</td>
<td>0.212</td>
<td>0.204</td>
<td>0.220</td>
<td>0.339</td>
</tr>
<tr>
<td>0.176</td>
<td>0.183</td>
<td>0.181</td>
<td>0.125</td>
<td>0.335</td>
</tr>
<tr>
<td>0.263</td>
<td>0.034</td>
<td>0.119</td>
<td>0.267</td>
<td>0.317</td>
</tr>
<tr>
<td>0.018</td>
<td>0.057</td>
<td>0.339</td>
<td>0.272</td>
<td>0.314</td>
</tr>
<tr>
<td>0.016</td>
<td>0.184</td>
<td>0.269</td>
<td>0.131</td>
<td>0.400</td>
</tr>
<tr>
<td>0.141</td>
<td>0.002</td>
<td>0.443</td>
<td>0.028</td>
<td>0.386</td>
</tr>
<tr>
<td>0.298</td>
<td>0.102</td>
<td>0.338</td>
<td>0.015</td>
<td>0.247</td>
</tr>
<tr>
<td>0.296</td>
<td>0.115</td>
<td>0.435</td>
<td>0.001</td>
<td>0.153</td>
</tr>
<tr>
<td>0.211</td>
<td>0.116</td>
<td>0.337</td>
<td>0.212</td>
<td>0.124</td>
</tr>
<tr>
<td>0.394</td>
<td>0.046</td>
<td>0.176</td>
<td>0.162</td>
<td>0.222</td>
</tr>
<tr>
<td>0.440</td>
<td>0.002</td>
<td>0.182</td>
<td>0.201</td>
<td>0.175</td>
</tr>
<tr>
<td>0.393</td>
<td>0.111</td>
<td>0.135</td>
<td>0.215</td>
<td>0.146</td>
</tr>
<tr>
<td>0.423</td>
<td>0.010</td>
<td>0.159</td>
<td>0.119</td>
<td>0.289</td>
</tr>
<tr>
<td>Bed Room</td>
<td>Corridor</td>
<td>TV-Kitchen</td>
<td>TV-Corridor</td>
<td>TV Room</td>
</tr>
<tr>
<td>---------</td>
<td>----------</td>
<td>------------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td>0.85</td>
<td>0.03</td>
<td>0.03</td>
<td>0.02</td>
<td>0.07</td>
</tr>
<tr>
<td>0.92</td>
<td>0.02</td>
<td>0.02</td>
<td>0.01</td>
<td>0.04</td>
</tr>
<tr>
<td>0.95</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.02</td>
</tr>
<tr>
<td>0.96</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>0.93</td>
<td>0.03</td>
<td>0.02</td>
<td>0.01</td>
<td>0.02</td>
</tr>
<tr>
<td>0.87</td>
<td>0.05</td>
<td>0.03</td>
<td>0.02</td>
<td>0.03</td>
</tr>
<tr>
<td>0.69</td>
<td>0.13</td>
<td>0.09</td>
<td>0.03</td>
<td>0.05</td>
</tr>
<tr>
<td>0.49</td>
<td>0.18</td>
<td>0.16</td>
<td>0.07</td>
<td>0.10</td>
</tr>
<tr>
<td>0.35</td>
<td>0.19</td>
<td>0.21</td>
<td>0.11</td>
<td>0.15</td>
</tr>
<tr>
<td>0.27</td>
<td>0.19</td>
<td>0.23</td>
<td>0.14</td>
<td>0.18</td>
</tr>
<tr>
<td>0.22</td>
<td>0.19</td>
<td>0.22</td>
<td>0.16</td>
<td>0.21</td>
</tr>
<tr>
<td>0.20</td>
<td>0.18</td>
<td>0.24</td>
<td>0.16</td>
<td>0.23</td>
</tr>
<tr>
<td>0.17</td>
<td>0.16</td>
<td>0.34</td>
<td>0.14</td>
<td>0.21</td>
</tr>
<tr>
<td>0.18</td>
<td>0.16</td>
<td>0.28</td>
<td>0.15</td>
<td>0.23</td>
</tr>
<tr>
<td>0.19</td>
<td>0.17</td>
<td>0.23</td>
<td>0.16</td>
<td>0.25</td>
</tr>
<tr>
<td>0.19</td>
<td>0.17</td>
<td>0.22</td>
<td>0.16</td>
<td>0.26</td>
</tr>
<tr>
<td>0.18</td>
<td>0.17</td>
<td>0.23</td>
<td>0.15</td>
<td>0.27</td>
</tr>
<tr>
<td>0.15</td>
<td>0.15</td>
<td>0.29</td>
<td>0.13</td>
<td>0.27</td>
</tr>
<tr>
<td>0.13</td>
<td>0.13</td>
<td>0.34</td>
<td>0.11</td>
<td>0.29</td>
</tr>
<tr>
<td>0.13</td>
<td>0.13</td>
<td>0.30</td>
<td>0.11</td>
<td>0.35</td>
</tr>
<tr>
<td>0.14</td>
<td>0.13</td>
<td>0.21</td>
<td>0.10</td>
<td>0.42</td>
</tr>
<tr>
<td>0.22</td>
<td>0.12</td>
<td>0.14</td>
<td>0.08</td>
<td>0.43</td>
</tr>
<tr>
<td>0.43</td>
<td>0.11</td>
<td>0.09</td>
<td>0.06</td>
<td>0.32</td>
</tr>
<tr>
<td>0.68</td>
<td>0.07</td>
<td>0.05</td>
<td>0.04</td>
<td>0.16</td>
</tr>
</tbody>
</table>

**Table A.2:** Data Set 2. The probabilities represent the daily human routine. These values have been derived from information available at [http://www.bls.gov/tus/tables/a3_0711.pdf](http://www.bls.gov/tus/tables/a3_0711.pdf)
Figure A.1: Progress in learning human routine based on finding the person at different times of the day in Tv_Kitchen. The results of 56 week (shown in (d)) are highly correlated to the actual human routine (shown in (e)).
Figure A.2: Progress in learning human routine based on finding the person at different times of the day in TV_Corridor. The results of 56 week (shown in (d)) are highly correlated to the actual human routine (shown in (e)).
Figure A.3: Progress in learning human routine based on finding the person at different times of the day in Corridor. The results of 56 week (shown in (d)) are highly correlated to the actual human routine (shown in (e)).
B. Implementation

This chapter explains architectures and frameworks used in developing the autonomous mobile robot for searching humans in the home environment. On the very basic, Modular Controller Architecture (MCA2) has been employed as the controller architecture. On top of it, behavior based control has been realized using Integrated Behavior Based Controller Architecture (iB2C). In the following a short overview of these control architectures and frameworks is provided.

B.1 Modular Controller Architecture

The control system of Artos is based on MCA2-KL\textsuperscript{1} architecture. Originally the development of MCA2 started at FZI Karlsruhe [Scholl 02]. Later on an independent branch was established in Kaiserslautern. The MCA2-KL introduced several improvements and therefore has been used in the work in hand.

Modular Controller Architecture (MCA2) is a software framework based on C++ with real-time capabilities for controlling autonomous robots. The methods are realized by creating simple modules which have standardized interfaces. Figure B.1a shows the basic structure of the module in MCA2. Each module has five communication interfaces that receive and transmit data. Four of such interfaces, namely Sensor Input, Sensor Output, Controller Input and Controller Output, are used to connect it with other modules using edges and the fifth interface can be used to exchange internal parameters of the module during the execution of the software. Figure B.1b shows the interfaces of one of the module. The exchange of communication between different modules is realized via data transporting edges. These edges copies values from output interface of one module to the input interface of the other module. This exchange is carried out only when the data at the source is modified. Several modules are combined together into groups, which then act like a module and can be connected to other modules in the same manner. Figure B.2 shows the formation of localization group by combining individual modules.

\footnotetext[1]{Modular Controller Architecture - Kaiserslautern Branch [http://rrlib.cs.uni-kl.de/]}
Several tools have been developed to monitor modules and the exchange of data between them and facilitating the use of MCA2 for robotic solutions. Figure B.3 shows one of such tools called “MCABrowser” that can be used to observe the connectivity of different modules and communication of information between them. The values can be easily tracked and different parameters can be easily adjusted without requiring to restart the program on robot.

Another tool for monitoring is “MCAGui” which is more graphical in nature and is able to visualize the information. Several plugins and widgets have been developed that can translate the data into meaningful and user understandable visualizations. Figure B.4 shows the interface of the MCAGui with different widgets displaying information for easy understanding.

### B.2 Integrated Behavior Based Control Architecture (iB2C)

Many components of ARTOS are being developed as behaviors of the behavior based control architecture iB2C\(^2\) [Proetzsch 10]. A behavior in iB2C is a three tuple as defined in Equation B.1

\[
B = (f_r, f_a, F)
\]  

\(^2\)Integrated Behavior-Based Control
Figure B.3: Overview of Mapping in MCABrowser
In the following a brief description of the components of the iB2C is provided.

**Activation** The activation of a behavior is composed of the stimulation $s$ and the inhibition $i$:

$$i = s (1 - i)$$

**Activity** Activity $a$ indicates the level of activity of the behavior. The value ranges $[0,1]$ representing the amount of influence of a behavior where $a = 1$ implies highest impact intended and $a = 0$ indicates an inactive behavior.

**Inhibition** Inhibition $i$ is used to control the deactivation or inhibition of the behavior. Its value ranges $[0,1]$, where $i = 1$ refers to full inhibition which can be gradually decreased to $i = 0$ meaning no inhibition. Each behavior can be inhibited by $k$ other behaviors with its input $i$. The Inhibition has the inverse effect of Stimulation $s$ and reduces the relevance of the inhibited behavior.

**Input Vector** A behavior receives the data in the form of input vector, $e$, to execute a particular task. This input can be sensory data or information form other behaviors.
Output Vector The computed value of a behavior is transmitted via output vector $u$. This output can be used to control the actuators of the robot or can be passed to other behaviors for further processing.

Stimulation Stimulation $s$ is used to control the gradual activity of the behavior. Its value ranges $[0,1]$, where $s = 0$ means no stimulation and $s = 1$ signifies a fully stimulated behavior. The values in between 0 and 1 refers to partially stimulated behavior. Each behavior has an input stimulation, that indicates the intended relevance of the behavior. Stimulation can be used to adjust the relevance of the competing behaviors or to enable higher-level behaviors to use the functionality of lower level behaviors by explicitly stimulating them.

Target Rating Target rating $r$ indicates the assessment of the current situation. The behavior signal target rating represents the satisfaction of a behavior with the current situation with respect to its goals. The value ranges $[0,1]$, where $r = 0$ indicates that the behavior is contented, while $r = 1$ shows maximum dissatisfaction level.

Transfer Function Transfer function $F(e, s, i) = u$, implements the functionality of the behavior and computes the output of the behavior. $F$ provides the intelligence of a behavior, calculating actions depending on input values and internal representations. There is no limit on $F$, and it could be reactive response to certain input values, state machines, output of sophisticated algorithms etc.

Figure B.5 shows the evaluation of a behavior in MCAbrowser. Several of these behaviors can be combined together to form a behavior based group. Figure B.6 shows the combination of simple behaviors to form a group of behavior.

The fusion behavior is a special kind of behavior that fuses the data from different behaviors to generate the final outcome of the group. There are three variants of fusion function, namely maximum fusion, weighted average fusion and weighted sum fusion [Berns 09b, Proetzsch 10].

For maximum fusion, transfer function $F$ is defined as:

$$\bar{u} = \bar{u}_s \text{ where } s = \arg \max_c (a_c) \quad (B.2)$$
Activity and target rating are calculated using following equations:

\[ a = \max_c (a_c) \quad \quad r = r_s \text{ where } s = \arg \max_c (a_c) \tag{B.3} \]

In case of weighted average fusion, transfer function \( F \) is defined as:

\[ \vec{u} = \frac{\sum_{j=0}^{p-1} a_j \cdot \vec{u}_j}{\sum_{k=0}^{p-1} a_k} \tag{B.4} \]

Activity and target rating are defined as:

\[ a = \frac{\sum_{j=0}^{p-1} a_j^2}{\sum_{k=0}^{p-1} a_k} \cdot \iota \quad \quad r = \frac{\sum_{j=0}^{p-1} a_j \cdot r_j}{\sum_{k=0}^{p-1} a_k} \tag{B.5} \]

In case of weighted sum fusion, transfer function \( F \) is defined as:

\[ \vec{u} = \sum_{j=0}^{p-1} \frac{a_j \cdot \vec{u}_j}{\max_c (a_c)} \tag{B.6} \]

Activity and target rating are defined as:

\[ a = \min \left( 1, \sum_{j=0}^{p-1} \frac{a_j^2}{\max_c (a_c)} \right) \cdot \iota \quad \quad r = \frac{\sum_{j=0}^{p-1} a_j \cdot r_j}{\sum_{k=0}^{p-1} a_k} \tag{B.7} \]

Being modular in nature, MCA2 is a natural choice for implementing the behaviors of behavior based control architecture where behaviors are also divided into smaller behaviors to fulfill a major task.
C. Simulation and Visualization Environment

SimVis3D [Braun 07, Wettach 10] is being used for simulation and visualization of various robots in Robotics Research Lab. It is an open source modular framework based on widely used 3D rendering library Coin3D that rely on OpenGL for accelerated rendering. It is compatible to Open Inventor and is capable of generating complex simulation and visualization for robots and their environments. Using various components, a user can easily create custom scenes for various different meaningful scenarios and visualize and simulate a variety of environmental situations. Main components of SimVis3D framework have been shown in Figure C.1.

The SimVis3D framework consists of three major components namely, visualization module, simulation module, and physics engine. The visualization module is mainly responsible for visualizing spatial information like 3D environments, robots, static objects and human characters. The simulation module simulates sensors and actuators and generates data for them. A variety of actuators like stepper motors, servo motors etc. can be simulated by the simulation module. Similarly, distance sensors like laser scanners, ultrasonic sensors, PMD cameras, tactile sensors, etc. can easily be simulated. Vision sensors and acoustics (see [Schmitz 11] for acoustics simulation) have also been included in the simulation module. Using multiple cameras in the simulated environment, different views can easily be generated. All these sensors and actuators can be parameterized to achieve desired properties as of a real sensor. The physics engine module is based on Newton dynamics engine, though other engine can also be integrated, which is responsible for generating effect of interaction between different objects based on laws of physics.

The 3D replica model of the assisted living facility at IESE, Fraunhofer has been developed using a 3D modeling tool namely Blender. The 3D simulation of the environment is shown in Figure 7.1. During the modeling process, it was ensured that the dimensions of different rooms and eventually the complete simulation matches the real apartment.

[2] Photonic Mixer Device
A variety of furniture has been designed and added in the simulation to make it more close to reality. All these models and objects are exported as “wrl” files for importing in SimVis3D. These separate components are combined together in the XML description file which contains the mounting pose and location of different objects such that these are at their appropriate places in the 3D scene. The model of IESE is mounted as “ROOT” and named as “LAB” to indicate that all objects will be mounted to the “LAB”, see Listing C.1. This XML file is input to SimVis3D to visualize the scene.

The robot has been visualized with its chassis, wheels, pan-tilt camera and laser scanner. According to the scene description in Listing C.1, it has been introduced as “ARTOS” object in the visualized “LAB” environment. The pan-tilt camera and laser scanner are mounted to “ARTOS” with mounting location specified as “position” and “pose_offset” respectively. The ranges, field of view and other parameters have also been stated according to the characteristics of the real sensors. Figure C.2a shows the view from the camera and Figure C.2b shows the range of laser scanner mounted on the robot. A 3D pose element “artos_pose” is attached to the robot to control the movement and rotation in the simulated environment. As in real ARTOS, the same MCA2-KL is used to control the simulated robot which is accomplished by only making changes in the hardware abstraction layer.

The modeling of human motions is based on well established human modeling standard H-Anim\(^4\). One of the characteristics of this standard is that modeling of a movement is independent of a character or an avatar and therefore once a motion is modeled, it can be used for a variety of characters. The movements of body parts to generate a motion has been developed using Blender by specifying key frames in a time-line. At each key frame, position of the character is specified. Afterwards these key frames, along with the interpolation performed by the inverse kinematics tool integrated in Blender, are exported.

\(^4\)http://www.h-anim.org/Models/H-Anim1.1/
Figure C.2: (a) View from the simulated camera in simulation where human has been detected. (b) The red lines emitting from ARTOS show the range and field of view of simulated laser scanner.

as a sequence of joint angles. Combination of different movements to replicate real human like motion has been discussed in detail in Section 7.1.1.

The H-Anim avatar is included in the simulation by attaching the model to the “LAB” as can be seen in the Listing C.1 and the exported sequences of joint angles are loaded at the start of the simulation.

Listing C.1: A snippet of XML description for the scene in elderly care scenario

```
<content unit="meters"/>

<!-- iese-room -->

<part file="artos/vis_obj/iese.wrl" name="LAB" attached_to="ROOT"
    pose_offset="0 0 0 0 0 0" />
<element name="lab_switch" type="Switch" switch_mode="-3" attached_to="LAB"/>
<element name="lab_mount" type="3d Pose Tzyx" position="0 0 0"
    orientation="0 0 0" angle_type="deg" attached_to="LAB"/>

<part file="artos/vis_obj/empty.iv" name="HALL_MOUNT" attached_to="LAB"
    pose_offset="0 0 0 0 0 0" />
<part file="artos/vis_obj/empty.iv" name="BATHROOM_MOUNT" attached_to="LAB"
    pose_offset="0 2.3 0 0 0 0" />
<part file="artos/vis_obj/empty.iv" name="BEDROOM_MOUNT" attached_to="LAB"
    pose_offset="0 4.25 0 0 0 0" />
<part file="artos/vis_obj/empty.iv" name="LIVINGROOM_MOUNT" attached_to="LAB"
    pose_offset="3 0 0 0 0 0" />
<part file="artos/vis_obj/empty.iv" name="KITCHEN_MOUNT" attached_to="LAB"
    pose_offset="3.1 6.3 0 0 0 0" />

<!-- hall -->

<part file="artos/vis_obj/halltable.wrl" name="HALLTABLE" attached_to="HALL_MOUNT"
    pose_offset="2.2 0.8 0.01 0 0 0" />
<element name="halltable_mount" type="3d Pose Tzyx" position="0 0 0"
    orientation="0 0 0" angle_type="rad" attached_to="HALLTABLE"/>

<part file="artos/vis_obj/hallbookshelf.wrl" name="BOOKSHELF"
    attached_to="HALL_MOUNT" pose_offset="0.5 1.5 0 0 0 -20" />
```
<part file="artos/vis_obj/ktlittlecabinet.wrl" name="KTCABINET3" attached_to="KITCHEN_MOUNT" pose_offset="2.7 2.6 0 0 0 0" />
<element name="KtCabinet3_mount" type="3d Pose Tzyx" position="0 0 0"
orientation="0 0 0" angle_type="rad" attached_to="KTCABINET3"/>

<part file="artos/vis_obj/kttable.wrl" name="DINNERTABLE" attached_to="KITCHEN_MOUNT" pose_offset="2.8 0.1 0 0 0 0" />
<element name="dinnertable_mount" type="3d Pose Tzyx" position="0 0 0"
orientation="0 0 0" angle_type="rad" attached_to="DINNERTABLE"/>

<part file="artos/vis_obj/ktoven.wrl" name="OVEN" attached_to="KITCHEN_MOUNT" pose_offset="0.2 2.6 0 0 0 0" />
<element name="oven_mount" type="3d Pose Tzyx" position="0 0 0"
orientation="0 0 0" angle_type="rad" attached_to="OVEN"/>

<part file="artos/vis_obj/ktwoven.wrl" name="MWOVEN" attached_to="KITCHEN_MOUNT" pose_offset="0.2 2.6 0.9 0 0 0" />
<element name="mwoven_mount" type="3d Pose Tzyx" position="0 0 0"
orientation="0 0 0" angle_type="rad" attached_to="MWOVEN"/>

<part file="artos/vis_obj/ktwash.wrl" name="KTWASH" attached_to="KITCHEN_MOUNT" pose_offset="1.5 2.6 0 0 0 0" />
<element name="ktwash_mount" type="3d Pose Tzyx" position="0 0 0"
orientation="0 0 0" angle_type="rad" attached_to="KTWASH"/>

<!-- living room -->
<part file="artos/vis_obj/lvcompchair.wrl" name="COMPCHAIR" attached_to="LIVINGROOM_MOUNT" pose_offset="3 5 0 0 0 180" />
<element name="compchair_mount" type="3d Pose Tzyx" position="0 0 0"
orientation="0 0 0" angle_type="rad" attached_to="COMPCHAIR"/>

<part file="artos/vis_obj/TVtable.wrl" name="TVTABLE" attached_to="LIVINGROOM_MOUNT" pose_offset="2 0.75 0 0 0 0" />
<element name="tvtable_mount" type="3d Pose Tzyx" position="0 0 0"
orientation="0 0 0" angle_type="rad" attached_to="TVTABLE"/>

<part file="artos/vis_obj/lvlittleshelf.wrl" name="LVLITTLECABINET" attached_to="LIVINGROOM_MOUNT" pose_offset="0.05 6.15 0 0 0 40" />
<element name="lvlittlecabinet_mount" type="3d Pose Tzyx" position="0 0 0"
orientation="0 0 0" angle_type="rad" attached_to="LVLITTLECABINET"/>

<part file="artos/vis_obj/lv wideshelf.wrl" name="LVWIDESHELF" attached_to="LIVINGROOM_MOUNT" pose_offset="0.5 0.7 0 0 0 -25" />
<element name="lwideshelf_mount" type="3d Pose Tzyx" position="0 0 0"
orientation="0 0 0" angle_type="rad" attached_to="LVWIDESHELF"/>

<part file="artos/vis_obj/lvttable.wrl" name="LVTABLE" attached_to="LIVINGROOM_MOUNT" pose_offset="2.3 0 0 0 90" />
<element name="lvtable_mount" type="3d Pose Tzyx" position="0 0 0"
orientation="0 0 0" angle_type="rad" attached_to="LVTABLE"/>

<part file="artos/vis_obj/tv.wrl" name="TV" attached_to="LIVINGROOM_MOUNT" pose_offset="2 0.4 0 0 0 180" />
<element name="tv_mount" type="3d Pose Tzyx" position="0 0 0"
orientation="0 0 0" angle_type="rad" attached_to="TV"/>

<part file="artos/vis_obj/sofa.wrl" name="SOFA_1" attached_to="LIVINGROOM_MOUNT" pose_offset="2.7 2.2 0 0 0 0" />
<element name="sofa_mount" type="3d Pose Tzx" position="0 0 0" orientation="0 0 0" angle_type="rad" attached_to="SOFA_1"/>

<part file="artos/vis_obj/sofa.wrl" name="SOFA_2" attached_to="LIVINGROOM_MOUNT" pose_offset="3 0 0 0 0 0" />
<element name="sofa_mount" type="3d Pose Tzx" position="0 0 0" orientation="0 0 0" angle_type="rad" attached_to="SOFA_2"/>

<part file="artos/vis_obj/sofa.wrl" name="SOFA_3" attached_to="LIVINGROOM_MOUNT" pose_offset="3.8 0 0 0 0 5" />
<element name="sofa_mount" type="3d Pose Tzx" position="0 0 0" orientation="0 0 0" angle_type="rad" attached_to="SOFA_3"/>

<part file="artos/vis_obj/lvlowcabinet.wrl" name="LVLOWCABINET" attached_to="LIVINGROOM_MOUNT" pose_offset="-0.1 3.3 0 0 0 0" />
<element name="lvlowcabinet_mount" type="3d Pose Tzx" position="0 0 0" orientation="0 0 0" angle_type="rad" attached_to="LVLOWCABINET"/>

<part file="artos/vis_obj/lvdesk.wrl" name="DESK" attached_to="LIVINGROOM_MOUNT" pose_offset="2.7 5.55 0 0 0 180" />
<element name="desk_mount" type="3d Pose Tzx" position="0 0 0" orientation="0 0 0" angle_type="rad" attached_to="DESK"/>

<!-- H-ANIM MODEL -->
<part file="hanim/new_models/yt/yt_002b.wrl" name="myModel" attached_to="LAB" pose_offset="1 1 0 90 0 90" />

<!-- artos -->
<part file="artos/vis_obj/artos.iv" name="ARTOS" attached_to="LAB" pose_offset="0 0 0 0 0 0" />
<element name="artos_pose" type="3d Pose Tzx" position="1 1 0" orientation="0 0 0" angle_type="rad" attached_to="ARTOS"/>

<!-- laser scanners -->
<part file="artos/vis_obj/empty.iv" name="Another_Scanner_Mount" attached_to="ARTOS" pose_offset="0.39 0.0 0.06 0 0 0" type="Point Set" number_of_points="682" point_size="100" color="0 255 10 25" visible="1" />
<distance_sensor name="Another_scanner" type="URG" max_distance="4.094" scan_angle_range="240" angular_resolution="0.352423" sensor_offset="0.0 0.0 0.0 0 0 0" image_width="248" image_height="5" near_distance="0.01" far_distance="200." scan_mode="depth_buffer" attached_to="Another_Scanner_Mount" />

<!-- cameras -->
<camera name="livingroom_cam" type="perspective" vfov="45" near_limit="0.1" attached_to="LIVINGROOM_MOUNT"/>
<element name="livingroom_cam_mount" type="3d Pose Tzyx" position="0 0 0.1" orientation="0 0 0" angle_type="rad" attached_to="livingroom_cam"/>

camera name="ARTOS_CAM" type="perspective" near_limit="0.1" vfov="45"
attached_to="ARTOS"/>
<element name="ARTOS_PTZ_CAM" type="3d Pose Tzyx" position="-0.2 0.0 0.3"
orientation="90 0 -90" angle_type="deg" attached_to="ARTOS_CAM"/>
<! -- PMD camera - Kinect Camera -->
<part file="artos/vis_obj/empty.iv" name="PMD_MOUNT" attached_to="ARTOS"
pose_offset="0.25 0.0 0.25 0 0 0"/>
<distance_sensor name="Kinect_camera" type="pmd" scan_mode="depth_buffer"
max_distance="3.5" image_width="64" image_height="48" vfov="43"
near_distance="0.01" far_distance="2.0" attached_to="PMD_MOUNT"
sensor_offset="0.0 0.0 0.0 0 0 0"/>
</content>
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Population trend in Germany - from pyramid to mushroom</td>
<td>2</td>
</tr>
<tr>
<td>1.2</td>
<td>Elderly population doubles from 7 to 14 percent in different countries</td>
<td>3</td>
</tr>
<tr>
<td>2.1</td>
<td>Number of robots in use by 2008</td>
<td>12</td>
</tr>
<tr>
<td>2.2</td>
<td>Number of service robots for personal and domestic use</td>
<td>12</td>
</tr>
<tr>
<td>2.3</td>
<td>Examples of robotic vacuum cleaners in domestic use</td>
<td>13</td>
</tr>
<tr>
<td>2.4</td>
<td>Motion patterns of robotic vacuum cleaners</td>
<td>14</td>
</tr>
<tr>
<td>2.5</td>
<td>Transportation robots in indoor environments</td>
<td>15</td>
</tr>
<tr>
<td>2.6</td>
<td>Examples of telepresence robots in use</td>
<td>17</td>
</tr>
<tr>
<td>2.7</td>
<td>Futuristic view of elderly care robots</td>
<td>18</td>
</tr>
<tr>
<td>2.8</td>
<td>Elderly care robots, Pearl and SCITOS G3</td>
<td>19</td>
</tr>
<tr>
<td>2.9</td>
<td>Series of Care-O-bots designed for performing household tasks</td>
<td>20</td>
</tr>
<tr>
<td>2.10</td>
<td>Kitchen helper robots, PR-2 and Rosie</td>
<td>21</td>
</tr>
<tr>
<td>2.11</td>
<td>Elderly care robots Kompaı and uBot used for telepresence</td>
<td>22</td>
</tr>
<tr>
<td>2.12</td>
<td>Overall system architecture of SCITOS</td>
<td>23</td>
</tr>
<tr>
<td>2.13</td>
<td>Basic Modules of system architecture of Care-O-bot</td>
<td>23</td>
</tr>
<tr>
<td>3.1</td>
<td>Autonomous Robot for Transport and Service, (ARTOS)</td>
<td>30</td>
</tr>
<tr>
<td>3.2</td>
<td>CAD model of ARTOS</td>
<td>31</td>
</tr>
<tr>
<td>3.3</td>
<td>Placement of ultrasonic sensors on ARTOS</td>
<td>33</td>
</tr>
<tr>
<td>3.4</td>
<td>RFID reader module by Feig Electronics installed under ARTOS</td>
<td>34</td>
</tr>
<tr>
<td>3.5</td>
<td>Map building on ARTOS</td>
<td>38</td>
</tr>
<tr>
<td>3.6</td>
<td>Localization on ARTOS</td>
<td>40</td>
</tr>
<tr>
<td>3.7</td>
<td>Navigation in ARTOS</td>
<td>46</td>
</tr>
<tr>
<td>3.8</td>
<td>Floor plan of IESE Fraunhofer marked with sensors installation</td>
<td>47</td>
</tr>
<tr>
<td>3.9</td>
<td>Apartment setup at IESE Fraunhofer for performing research activities in</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>range of elderly care</td>
<td></td>
</tr>
<tr>
<td>3.10</td>
<td>Passive RFID tags by Euro I.D.</td>
<td>49</td>
</tr>
</tbody>
</table>
3.11 Simulated Environment with human character and simulated ARTOS 50
4.1 Information processing performed by a human brain 52
4.2 Design concept for searching human in home environment using mobile robot 55
5.1 Observing only critical regions 59
5.2 Placement of guards with limited range 59
5.3 Camera placement for full coverage 60
5.4 Full area coverage using multiple robots 61
5.5 Overview of methodology for finding View Points 64
5.6 Experiments in IESE simulation for autonomously determining View Points 68
5.7 Experiments in simulated environment of IESE after changing furniture 69
5.8 Experiments in simulated environment of IESE without any furniture 70
5.9 Experiments in simulation of RRLAB 71
5.10 Grid map of home environment at IESE as generated by ARTOS 71
5.11 View Points determined in the home environment 72
5.12 Experiment in home environment with extended Sensing Range 72
5.13 Area coverage in different environments with Primary View Points and Secondary View Points 74
5.14 Time required to calculate View Points in different environments with varying Sensing Range 74
6.1 Aware Home at Georgia Tech in Atlanta with RFID mats installed in the environment 79
6.2 View from camera installed at Aware Home at Georgia Tech in Atlanta 79
6.3 Overall architecture for localizing humans in the home environment at Aware Home 80
6.4 PIR sensors, door switches and light control switches installed at Center for Advanced Studies in Adaptive Systems (CASAS) to detect human 81
6.5 Installation of pressure sensors and camera system to localize human at home 82
6.6 Human localization using user badge and network nodes in the environment 83
6.7 Wearable FreeDigitizer used to generate map of the environment and localize the person at home 84
6.8 Human localization using passive RFID tags installed under the carpet 85
6.9 Using a robot to find human in the home environment 86
6.10 Using a robot to find human in simulation 87
6.11 Preferences of people wanting to stay at home and care services provided at their homes 89
6.12 Number of hours an elderly person spends in various activities . . . . . . . 90
6.13 Number of hours spent with people by a person of age 65 years or above . . . . 91
6.14 Leisure time activities of an elderly person ................................. 91
6.15 Concept of human search using an autonomous mobile robot .................. 94
6.16 Human pose detection ............................................................. 105
6.17 Names of View Points in the environment for searching human ................. 106
6.18 $A^*$ Algorithm based Navigational Costs between the View Points ......... 107
6.19 Human presence probabilities ....................................................... 107
6.20 MDP policy at 14:00 Hrs on four consecutive days for finding the human .. 108
6.21 Human found by ARTOS at different places in the home environment ...... 109
6.22 Elaboration of policy generation .................................................. 110
7.1 Floor plan of assisted living lab and simulated environment .................... 114
7.2 Dynamic postures of simulated human character ............................... 115
7.3 Markov chain for selection of human movements ................................ 116
7.4 Primary View Points determined in the map of IESE Fraunhofer .............. 118
7.5 Effect of varying Sensing Range on the number of View Points .............. 119
7.6 Effect of varying Sensing Range on total area coverage ....................... 120
7.7 Learning the presence of a person in bedroom .................................. 122
7.8 Learning the presence of a person in TV room .................................. 123
A.1 Human routine learned for TV_Kitchen ....................................... 136
A.2 Human routine learned for TV_Corridor ..................................... 137
A.3 Human routine learned for Corridor .......................................... 138
B.1 (a) Human Probability module in MCA2 (b) Interfaces of Point Approacher Module .................. 140
B.2 Several modules combined together to form a group .......................... 140
B.3 Overview of Mapping in MCABrowser ........................................ 141
B.4 MCAGui with various widgets ..................................................... 142
B.5 (a) Graphical illustration of a behavior (b) Signals of behaviors .............. 142
B.6 Behavior Based Group ............................................................... 143
C.1 Overview of SimVis3D ................................................................. 146
C.2 Sensors on ARTOS in simulation ................................................ 147
Bibliography


