DEVELOPING A HOME SERVICE ROBOT PLATFORM FOR SMART HOMES

By

HA MANH DO

Bachelor of Science in Electronics and
Telecommunications
Hanoi University of Science and Technology
Hanoi, Vietnam
1999

Submitted to the Faculty of the
Graduate College of
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May, 2015

DEVELOPING A HOME SERVICE ROBOT PLATFORM FOR SMART HOMES

Thesis Approved:

Dr. Weihua Sheng

Thesis Adviser

Dr. Qi Cheng

Committee Member

Dr. Damon Chandler

Committee Member

ACKNOWLEDGMENTS

First and foremost, I would like to express my most sincere thanks to my adviser, Dr. Weihua Sheng, for his guidance throughout my study. I appreciate his guidance, inspiration, motivation, and encouragement. He has taught me not only the technical knowledge, but also passion for research. I also thank the members of my committee, Dr. Qi Cheng and Dr. Damon Chandler, for their comments and questions to improve this thesis.

Furthermore, I would like to thank my labmates. Ye Gu wrote the code for the gesture recognition through the kinect sensor for my experiments. Jeremy and Paul designed and built the telepresence robot. The discussions between me and other labmates also inspired me a lot. Most of all, I want to express my deepest thanks to my parents. I cannot thank them enough for their everlasting love, support, and understanding; these are the most precious gifts in my life. Finally, I am also grateful to my wife and my daughters, whose love, understanding, patience, consideration, and support make my life full of happiness and joy.

Ha Manh Do[1]

[1] Acknowledgements reflect the views of the author and are not endorsed by committee members or Oklahoma State University

Name: Ha Manh Do

Date of Degree: May, 2015

Title of Study: DEVELOPING A HOME SERVICE ROBOT PLATFORM FOR SMART HOMES

Major Field: Electrical Engineering

Abstract: The purpose of this work is to develop a testbed for a smart home environment integrated with a home service robot (ASH Testbed) as well as to build home service robot platforms. The architecture of ASH Testbed was proposed and implemented based on ROS (Robot Operating System). In addition, two robot platforms, ASCCHomeBots, were developed using an iRobot Create base and a Pioneer base. They are equipped with capabilities such as mapping, autonomous navigation. They are also equipped with the natural human interfaces including hand-gesture recognition using a RGB-D camera, online speech recognition through cloud computing services provided by Google, and local speech recognition based on PocketSphinx. Furthermore, the Pioneer based ASCCHomeBot was developed along with an open audition system. This allows the robot to serve the elderly living alone at home. We successfully implemented the software for this system that realizes robot services and audition services for high level applications such as telepresence video conference, sound source position estimation, multiple source speech recognition, and human assisted sound classification. Our experimental results validated the proposed framework and the effectiveness of the developed robots as well as the proposed testbed.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

In this thesis, a home service robot platform is developed and integrated into smart homes to assist the elderly who live independently in their own residence. In this chapter, the motivation, the related works, the objectives, and the outline of this thesis are presented as an overall introduction.

1.1 Motivation

Mobile robots have already come into human environments in recent years. They have been used in many places such as homes, offices, hospitals, battlefields, and emergency-response sites. The home service robots are expected to live with humans and do chores around their homes. They can provide simple services, such as vacuuming the floor. Also, they can do more complex tasks, such as serving drinks, or even entertaining guests by dancing or playing instruments.

The elderly population around the world is increasing. The number of people 60 years old and above had increased to almost 810 million in 2012 and is forecast to reach 2 billion by 2050 [2] [3]. Elderly people are an important asset to society. The life experience and wisdom they have gained over the years make them a vital social resource [4]. However, along with these benefits there are added challenges. Providing olders with age-friendly physical and social environments helps them live an independent life and also improves their active participation, maximizing their contribution to the society [3]. Although services such as adult day care, long term care, nursing homes, hospice care, and home care can provide elders with all the supports of the health, nutritional, social support, and daily living needs of adults, the feeling of independence is lost. Elders would prefer to stay in the comfort of their home where they feel more confident than moving to any expensive adult care or healthcare facility. Hence if old adults are able to complete self-care activities on their own, encouraging them in their efforts in maintaining independence can provide them with a sense of accomplishment and ability to enjoy independence longer [5]. The best way to support them is to provide a physical environment that promotes active ageing through the use of innovative technologies, such as smart homes [3].

The most recent survey on smart homes was presented by Alam et al. [6]. A smart home environment is defined as a ubiquitous computing application that is able to provide users with context-aware automated or assistive services in the form of ambient intelligence, remote home control, or home automation. Moreover, smart homes provide comfort, healthcare, and security services to their inhabitants. In a robot-integrated smart home environment, the mobile robots can utilize the smart home sensor networks as their own sensors, therefore they could be smarter and could effectively assist and collaborate with humans. The complexity could shift from the mobile robot into the environment and into the model knowledge [7]. Therefore, the robot-integrated smart home environment would be the perfect use of technology to achieve the goal of caring for the elderly at their own home. The goal of this thesis is to develop an open home service robot platform that not only is equipped with some basic services and applications but also is integrated into a smart home environment to support further research in elderly care.

1.2 Related Work

In recent years, assistance for the elderly living alone at home has been receiving growing interest. There are already some commercial assistive social robots for elderly care, such as Aibo, Pearl, Care-o-bot I, Care-o-bot II, Care-o-bot III, Homie, iCat, Paro, Huggable, etc. [8]. Some robots, for example Pearl and Care-o-bots, can recognize words, synthesize speech, work as autonomous guiding robots or telepresence robots, and remind people about routine activities such as eating, drinking, taking medicine, and using the bathroom, but they do not have the auditory learning capability which enables the robots to understand both voice and environmental sounds. Most previous researches have focused on human-robot interaction and navigation in the development of personal service robots for elderly care. Johnny [9], a research and development platform robot for domestic environments, was equipped with multi-modal human robot interaction including speech and gesture, SLAM (Simultaneous Localization and Mapping) and navigation, object detection, and sound localization. The European FP7 CompanionAble project [10] aimed at developing a mobile home robot companion that has friendly GUI, SLAM, different task-oriented navigation behaviours, speech recognition, dialogue manager, and human detection and tracking through smart-home infrared sensors. In 2009, The Honda humanoid ASIMO robot was controlled to a limited degree by the operator's thoughts. The experimental system combines EEG with near-infrared (NIR) spectroscopy and the operator wears a helmet featuring NIR and EEG (Electroencephalography) sensors which monitor and decode electrical brainwaves and cerebral blood flow [11]. The personal robot platform PR2 was programmed to help a severely disabled man [12]. The HealthBot robot [13] can interact with a user via the touch screen and synthesized speech, navigate and avoid obstacles, remind about schedules, detect falls, and assist in taking blood pressure and blood oxygen saturation (SPO2) measurements.

Recently, Robotics and cloud computing have been integrated to develop healthcare systems. The architecture of ROSCHAS [12] (Robotics and Cloud-assisted Healthcare System) was designed to enable the system to provide pervasive healthcare services and especially the mental healthcare for empty nesters. However, many challenges need to be addressed, especially on high bandwidth and energy efficient

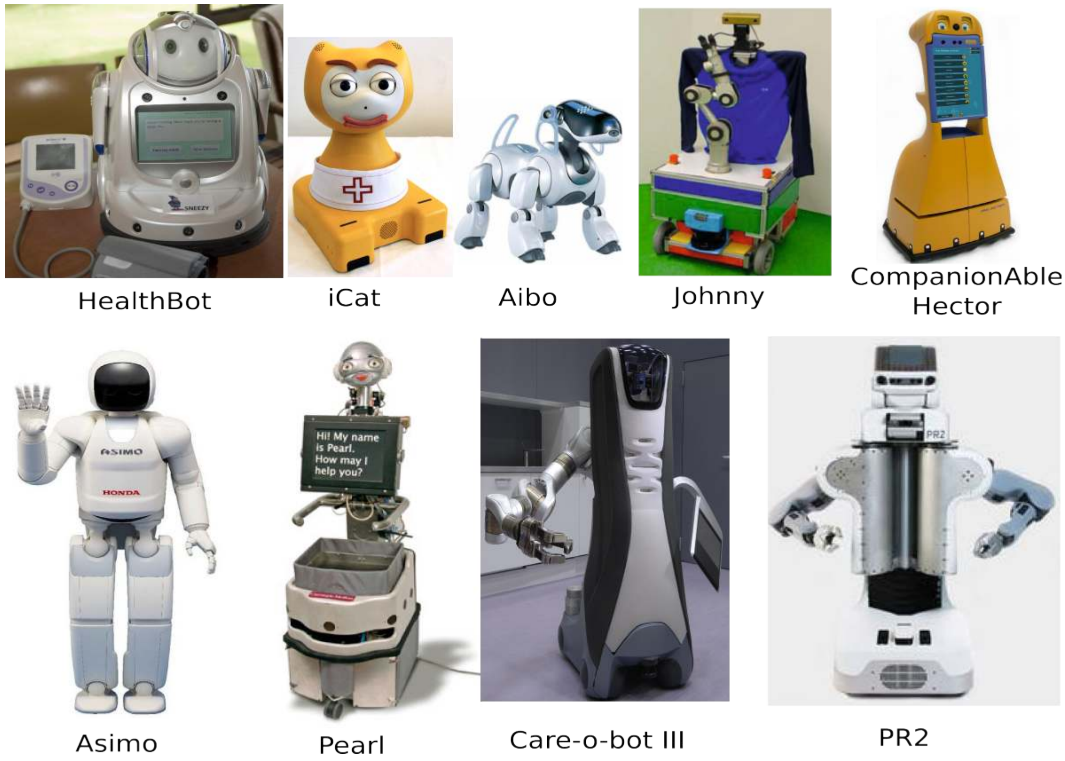


Figure 1.1: Existing assistive social robots [Source: Internet]

communication protocols, low cost robot, interoperability between robot and cloud platform, and physiological sensor networks, etc.

A smart home along with a domestic robot can be used to take care of the elderly [10] [14]. Georgia Institute of Technology has constructed Aware Home Research Initiative (AHRI) that includes a three-story 5,040 square-foot house to address the social challenges the elderly face at home. In this research facility, the human's location is identified using RFID tags worn below the knee. This system provides room level occupancy accuracy. In order to improve accuracy and to recognize the human activity, a series of unobtrusive cameras were installed on the first floor of the Aware Home. By using location, gestures, and interaction with other objects, the behavior of the human is recognized [15]. Another similar work is Gator Tech Smart House [16] in The University of Florida that helps aged and disabled people. Gator Tech Smart House is fitted with various smart devices to help day to day activities of the occupant. Ultrasonic transceivers are placed on the ceiling corners of each room and the user has to wear a vest with an ultrasonic tag. The location of the resident is detected using triangulation in detecting occupant's movements, location and orientation. Smart floors embedding pressure sensors are used to identify and track the location of all house occupants. They are also used to detect if occupants fall. Smart cameras are used for video surveillance and motion detection.

Smart home environments offer a better quality of life by employing automated appliance control and assistive services. Integration of mobile robots into smart home environments brings more benefit to our lives, specially for elderly care. It also helps to shift complexity from the mobile robots into the environment. The technologies which are present in smart homes can be used to improve mobile robots in terms of costs, performance and safety. This work aims to set up a small testbed for doing experiments of researches on Robot-integrated Smart Home Environment. The testbed has the following features:

- Simulate a smart home environment that includes physical spaces, sound sources, and sensor networks, etc.
- Be integrated with the different kinds of domestic robots (home service robots, surveillance robots, healthcare robots, etc.)
- Have open software infrastructure ready for high-level researches.

Moreover, future home service robots for the elderly who live independently in their own residence should play as a tool to serve the human, an avatar to represent the remote caregiver, and a social companion to collaborate and interact with the human. The robots could mainly overcome challenges in performing home tasks, co-operating with the remote caregivers or doctors to take care of elderly adults, who may be non-experts in technology, communicating with them in a natural and intuitive way, as well as understanding their intentions and activities. Therefore, those robots first have the capability to interact with humans through natural human interface, understand the home environments through both visual and auditory sensing, collaborate with remote caregivers in performing tasks. Such a human-aware capability frees the robot to do its daily routine work, while being able to attend the human more proactively and effectively. Therefore it is important to develop an open home service robot platform that is equipped with natural human interface, audition system, and human-robot collaboration capability. Such a platform can provide basic services and application to develop high-level applications for home service robots that can effectively serve the elderly at home.

1.3 Objectives

The objective of this thesis is to implement a testbed for smart home environment and particularly to develop an open and flexible home service robot platform for further researches in future robot-integrated smart homes. It is desirable to equip the robot platform with some basic services such as 2D SLAM, autonomous navigation, and other related functionalities. This robot platform will be integrated into the smart home environment through connection to the environmental sensor network, the body sensor network, as well as cloud servers. The robot platform is also capable of natural human interfaces such as gesture recognition, voice recognition, and sound event recognition. Such a platform can benefit the robotics research community due to its open architecture in both hardware and software.

1.4 Outlines

This thesis is organized as follow:

- This chapter presents the motivation, related works, and the objective of this work.

- Chapter II covers the development of a testbed for robot-integrated smart homes and the hardware and software implementation of the home service robots.

- Chapter III demonstrates the natural human interfaces for the home service robot.

- Chapter IV explains the audition system developed for the robot.

- Chapter V gives conclusion and potential future work.

CHAPTER 2

SMART HOME TESTBED INTEGRATED WITH A HOME SERVICE ROBOT

This chapter presents the development of a testbed for robot-integrated smart homes. The testbed includes a smart home environment, remote caregivers, cloud servers, and an experiment environment. An open layered architecture is proposed for software implementation of the testbed. This chapter also describers the hardware and software implementation of the home service robots that are integrated in the smart home environment.

2.1 Introduction

In order to develop and test our proposed home service robot, we need set up a home environment. Modifying a real apartment can cost a significant amount of money and time. Hence, a small-scale smart home testbed is preferred for initial study and reliability tests in a laboratory environment before conducting the experiment in real apartments. This smart home consists of a living room, a bedroom, a kitchen and a bathroom. The floor plan design and 3D view of the home are shown in Fig. 2.1. Furniture is set up in different rooms, such as a chair in the living room and a bed in the bedroom. The tesbed combines the small apartment with a network of an environmental sensors, a home service robot, a gateway, a mobile device, a sound simulation system, as well as an indoor localization system that provides the ground truth of the robot, the human, and other target objects. The design and implementation of this testbed are described in the following sections.

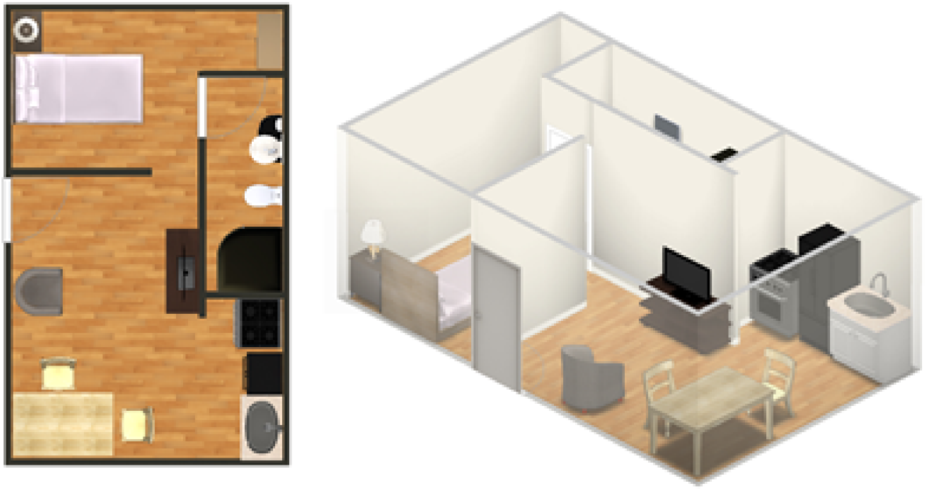


Figure 2.1: The floor plan and 3D view of the smart home

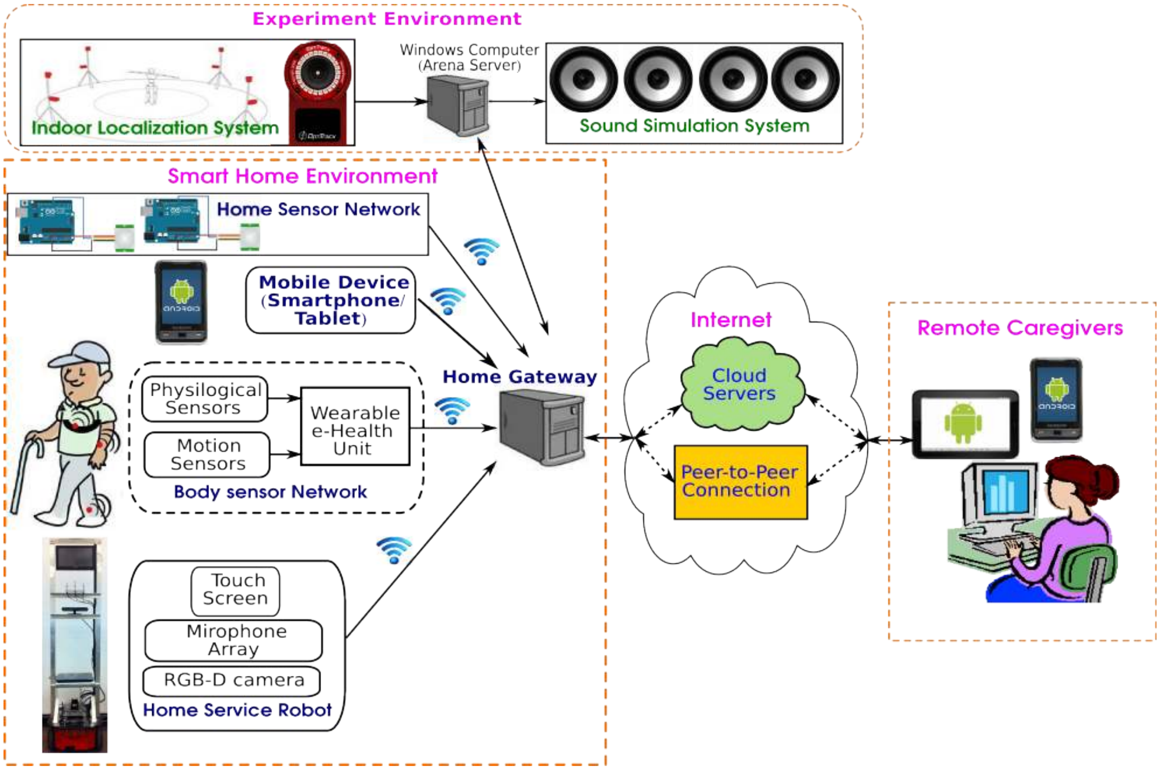


Figure 2.2: Structure of the ASCC Smart Home Testbed

2.2 The ASCC Smart Home (ASH) Testbed

The overall architecture of the ASCC Smart Home testbed is shown in Fig. 2.2. This testbed consists of four parts: a Smart Home environment, a remote caregiver,

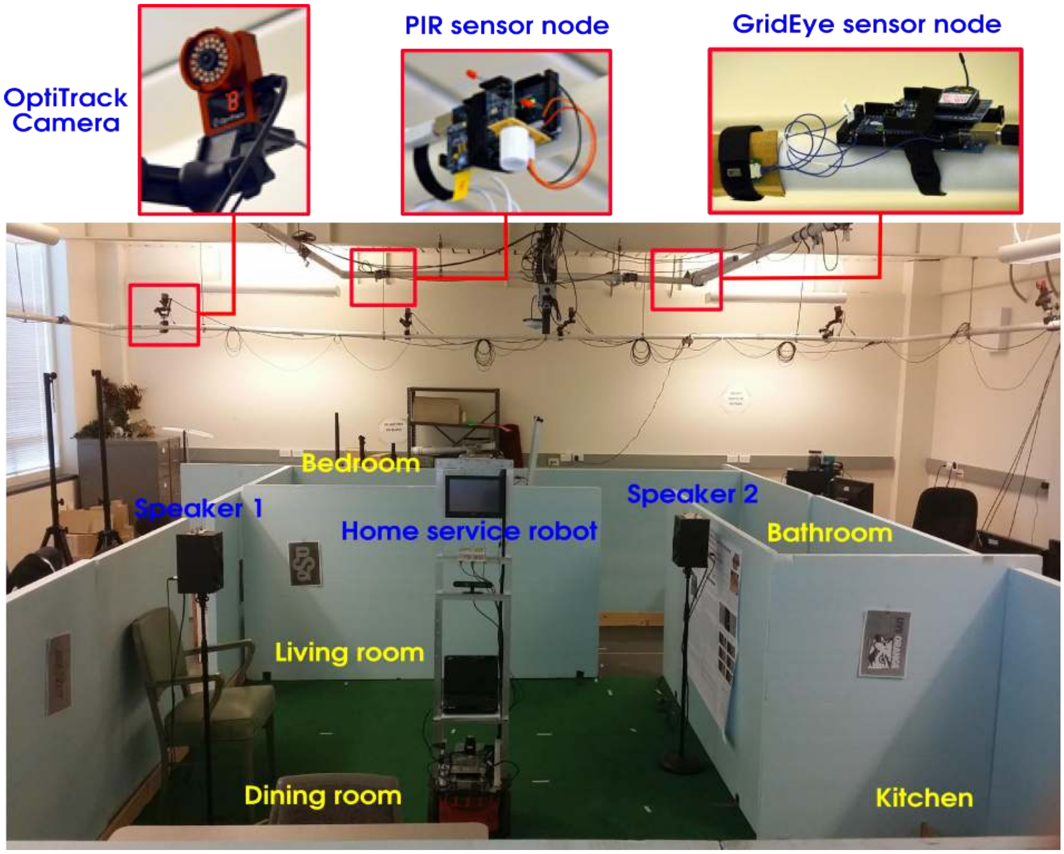


Figure 2.3: ASCC Smart Home testbed

cloud servers, and an experiment environment that provides ground truth and sound simulation. The Smart Home environment consists of five parts: a small apartment equipped with environmental sensors, a home service robot, a human subject with wearable body sensors, a mobile device such as a smartphone, and a home gateway. The Smart Home can be connected to the cloud server through the home gateway. The remote caregiver can provide health care to the elderly as well collaborate with the home service robot to take care of the elderly. The experiment environment includes a sound simulation system that generates various sounds related to human activities at home and the OptiTrack motion capture system that provides ground truth of locations. The Smart Home environment has been developed in our lab and it is shown in Fig. 2.3. The size of the Smart Home is about 16 feet by 22 feet. The components of the Smart Home environment are explained in the following

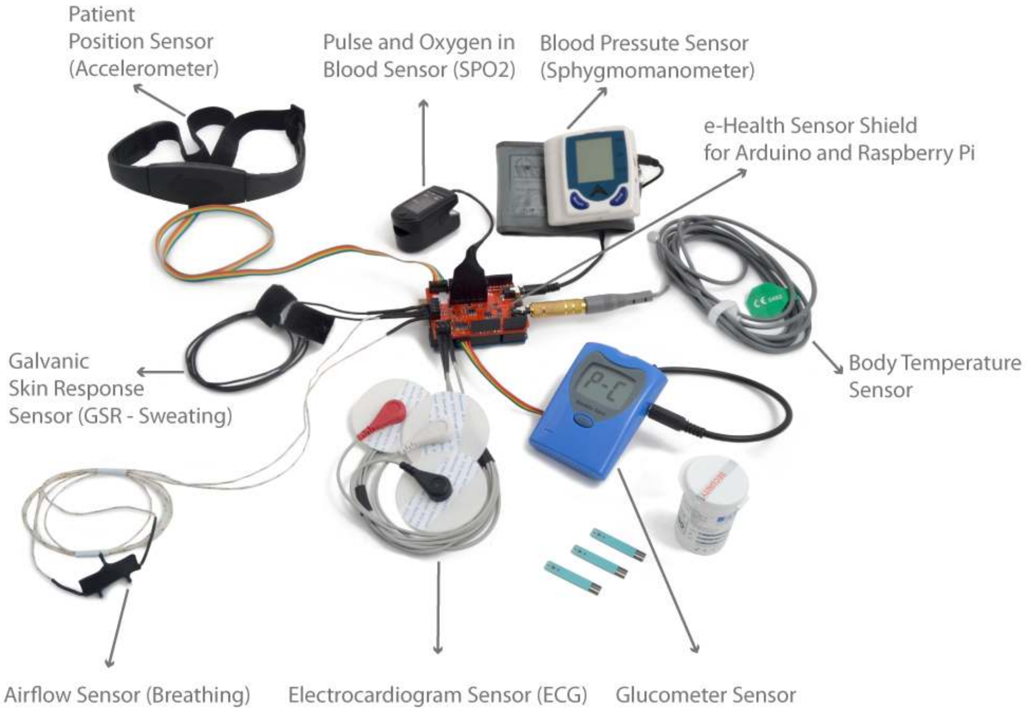


Figure 2.4: A prototype of body sensor network [1]

subsections.

2.2.1 Home service robots

Several home service robots are developed. They are based on the Pioneer robot base [17] and the iRobot Create base [18], respectively. Each robot is capable of holding other attachments which feature a laser rangefinder (LRF) for 2D navigation, an RGB-D camera for 3D perception and gesture-based applications, a microphone array to build the audition system, and a touch screen. The robot can be used as a telepresence robot, an elderly healthcare robot, or a companion. The hardware and software implementation of those robots will be presented later.

2.2.2 Body sensor network

The human body sensor network consists of physiological sensors, motion sensors, and a wearable e-Health unit [1]. The elderly carries this network for self-health check up through mobile devices or for remote healthcare provided by a doctor. A prototype of a body sensor network is shown in Fig. 2.4. The wearable e-Health unit, or the e-Health Sensor Shield, allows to develop biometric and medical applications by using 9 different sensors: pulse, blood pressure (sphygmomanometer), body temperature, oxygen in blood (SPO2), airflow (breathing), electrocardiogram (ECG), glucometer, galvanic skin response (GSR - sweating), and motion (accelerometer).

2.2.3 Mobile devices

A mobile device such as a smart phone or a tablet is used as a user interface to control the home service robot. It can also be used to collect the data from the body sensor network. Furthermore, the caregiver can use other mobile devices to remotely control the robot, remotely connect to the body sensor network, as well communicate with the elderly at their homes.

2.2.4 Home sensor network

The home sensor network consists of PIR sensors and GridEye sensors connected through XBee protocol. The installation of the sensors is shown in Fig. 2.3. The GridEye sensor node can provide 64 pixel temperature data in its field of view. The PIR sensor node can provide binary motion information in its field of view by detecting the IR radiation emitted by the target. The sensor nodes were strategically placed at different places inside the apartment. The GridEye sensor nodes were placed in larger rooms such as the living room and the bedroom, which enables better tracking performance. While PIR sensors were placed in the kitchen and the bathroom. Data from these nodes are transmitted through XBee protocol to the home gateway.

2.2.5 Home gateway

The home gateway is a local hub for data collection and processing in the Smart Home. It also enables the communication with the cloud server and the remote caregivers. The home gateway receives sensor data from the robot, the body sensor network and the home sensor network. Data processing that requires less computational power and more realtimeness can be done locally on the home gateway. However, if the data processing requires more powerful computation, such as visual and audio understanding, human health diagnosis, anomaly detection, etc. it is more desirable to outsource the processing to the cloud servers.

2.2.6 Indoor localization system

The indoor localization system is used to provide the ground truth of the robot, the human, and other target objects. We adopted an OptiTrack motion capture system from Natural Point Inc. [19]. This system consists of twelve OptiTrack V100:R2 cameras that can capture images within the range of 18 to 433 inches. The cameras were placed around the testbed to be able to cover the whole testbed area at the heigh from 4 feet to 6 feet. Each camera includes 26 Infrared (IR) LEDs which can emit an IR beam. The IR beam is reflected back to the cameras by the silver markers so that the system can localize the markers accurately. Each group of markers in a specific shape can be recognized in the system, be represented by a rigid body, be tracked at more than 95% accuracy, as well as be localized at millimeter accuracy in real-time.

2.2.7 Sound simulation system

The sound simulation was developed to simulate various sound events like those in a typical apartment. Multiple sound cards were installed in a desktop computer to connect to multiple speakers. The sound events in the bathroom, kitchen, living room,

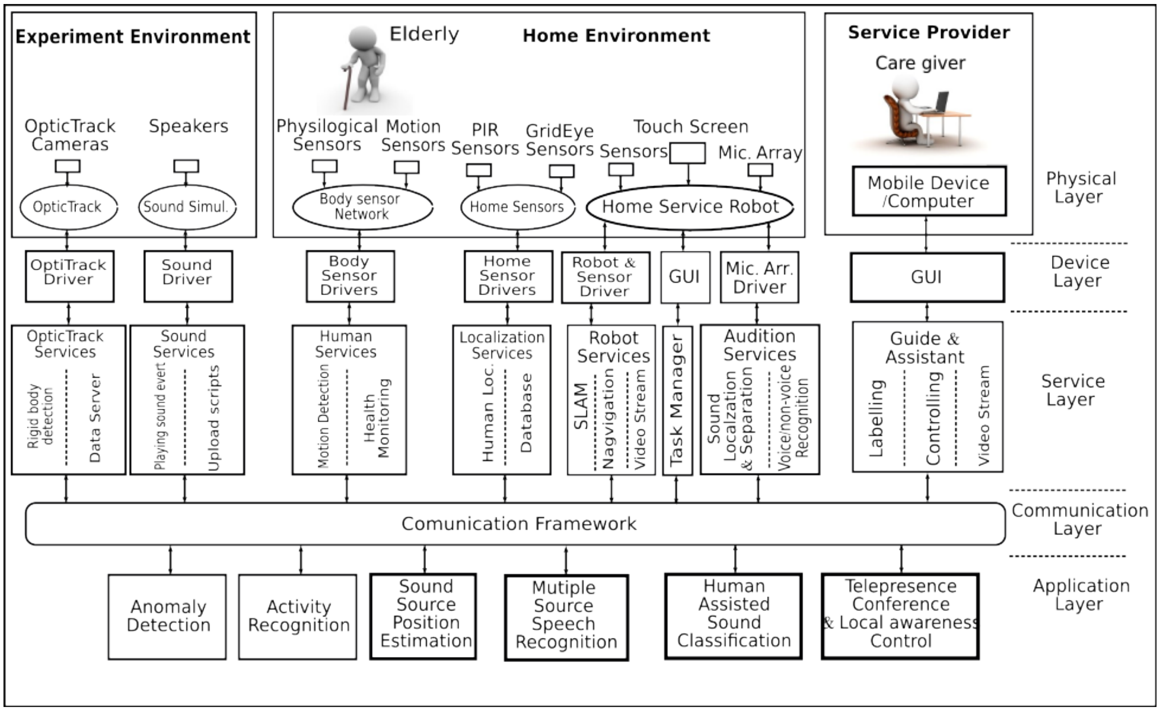


Figure 2.5: Layered architecture of ASH

and bedroom were recorded or collected from the Internet to build a sound library (SoundLib) for development of the robot audition system which will be discussed later.

2.3 Layered architecture of the ASCC Smart Home

In order to achieve modularity, extensibility, customizability, and reusability, the ASH testbed software was developed based on a layered architecture shown in Fig. 2.5. This architecture, extended beyond the collaborative architecture proposed in [20], allows remote caregivers and a home service robot to collaborate in serving the elderly. The architecture consists of five layers including the physical layer, device layer, service layer, communication layer, and application layer.

The physical layer is the hardware of ASH which includes hardware of the home service robot, human body sensor network, home sensor network, OptiTrack system, and sound simulation system, as well mobile devices or computers used by the caregivers. The robot hardware includes various sensors such as an RGB-D camera, a laser range finder (LRF), encoders, and a microphone array.

The hardware is handled by the device layer that provides the device drivers for the sensors and actuators on the robot, and the graphical user interface (GUI) for the caregiver and the elderly, and the drivers for other systems and sensor networks in ASH.

The service layer contains basic services provided by the robot, the audition system, the caregiver, the home sensor network and the body sensor network. The robot services include the SLAM(Simultaneous Localization and Mapping) and navigation. The audition services include sound localization, sound separation, and voice/non-voice recognition. The caregiver services include remote control, audio stream, and sound labelling. The home sensor network provides indoor localization service. The body sensor network service includes health monitoring, motion detection, etc.

The application layer contains high level functions such as anomaly detection, activity recognition, telepresence conference, local awareness control, sound source position estimation, multiple source speech recognition, human-assisted sound classification. Human-assisted sound classification provides a new way to enable the caregiver and the robot to work together to detect and classify sounds.

Based on the Robot Operating System (ROS) network [21], the communication layer provides a seamless connection between the different modules in the service layer and the application layer.

2.4 Home service robots

The ASCC Home Service robot is built with preexisting mobile platforms, which include a Pioneer mobile robot base and an iRobot Create robot base. Besides basic features such as SLAM and autonomous navigation based on 2D/3D maps, the robots allow a remote caregiver to achieve telepresence through audio and video communi-

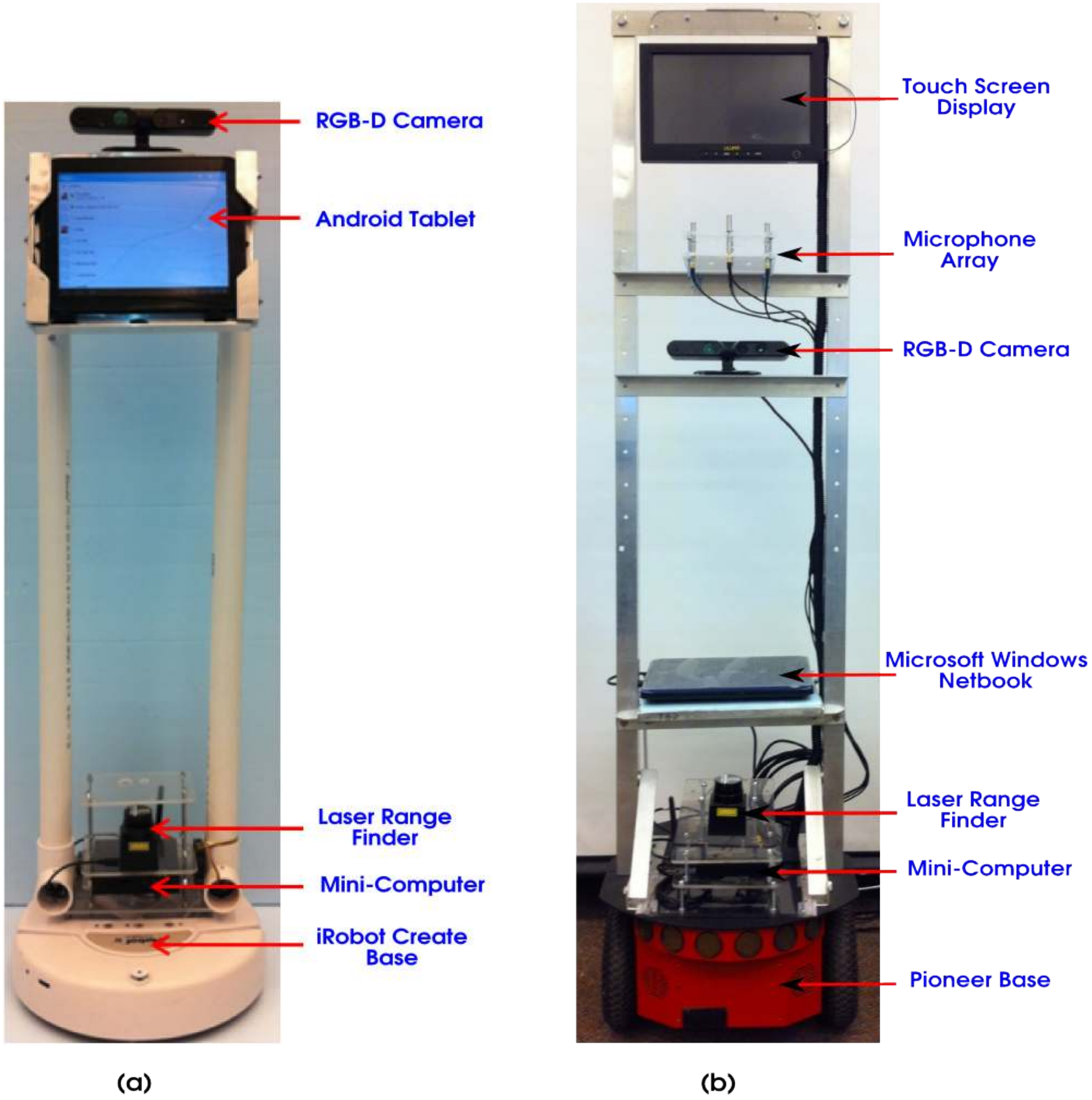


Figure 2.6: ASCC Home Service Robots

cation with the elderly. The robots also have the capability to collaborate with the remote caregiver to recognize sound events. Moreover, the robots robots are used to develop natural human-robot interaction and applications of collaborative sound event recognition.

2.4.1 Hardware platform

iRobot Create based home service robot

As shown in Fig. 2.6-(a), the home service robot, named ASCCHomeBot1, is built on an iRobot Create base with approximately 1.20m-long PVC pipes holding up a tablet used for video communication and online speech recognition. Mounted on it are an RGB and Depth (RGB-D) camera, a laser range finder (LRF) and a FitPC2 minicomputer [22]. It also has batteries onboard to power these devices. The iRobot Create and FitPC2 are powered by an Advanced Power System 14.4V/3000mAh NiMH battery and Powerizer 9.6v/4200mAh batteries respectively. The lifetime of the batteries was evaluated in both continuously fully-functional operation mode and standby mode. The FitPC2's battery can last for approximately 2 hours in the operation mode and more than 10 hours in the standby mode and the iRobot Create's battery can last for more than 5 hours in the operation mode and more than a week in the standby mode. The RGB-D camera mounted on top of the robot is an ASUS Xtion PRO LIVE [23] which enables to develop 3D modeling applications, gesture-based applications, etc. In addition, it is able to capture video with 24-bit true color and a resolution of 640x480 at 30 frames per second. The LRF is a Hokuyo URG-04LX-UG01 [24] which is used to build the 2D map of the environment around the robot. It is a low-power LRF with a wide range up to 5600mm x 240^0, and an accuracy of ±30mm. The FitPC2 minicomputer running ROS on Ubuntu 10.04 is used as the main computing system on the robot. It has a dual-core 1.6 GHz Intel Atom processor, and 1 GB of memory. It controls the robot motion and collects sensory data from the camera, the laser range finder, as well as the robot odometry. It also communicates with the remote mobile devices for tele-operation.

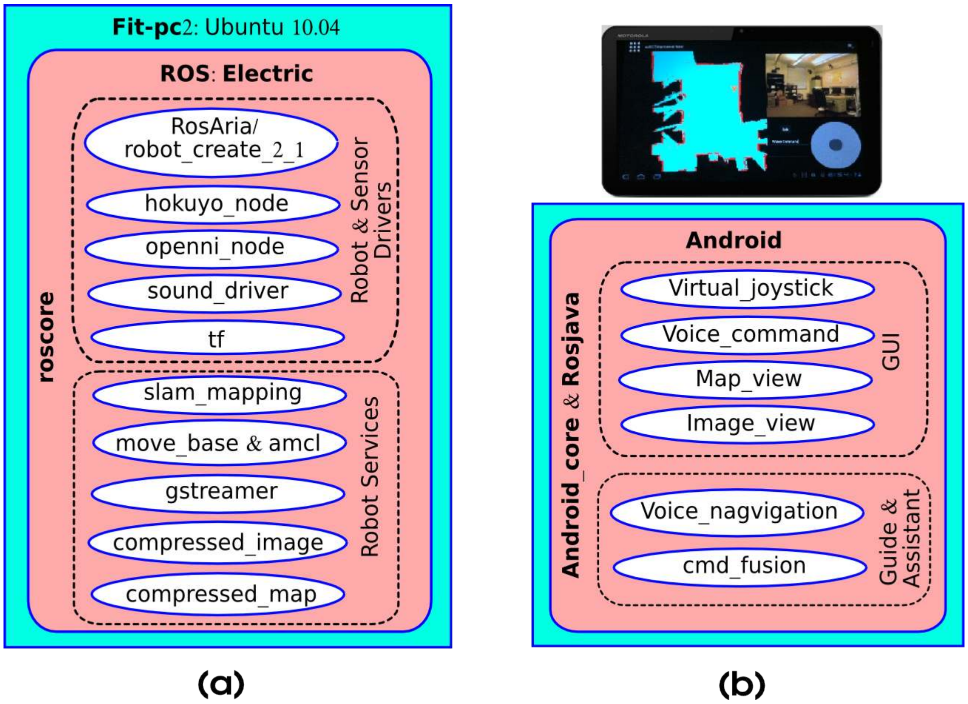


Figure 2.7: Software platform for ASCC Home Service Robots

Pioneer based home service robot

As shown in Fig. 2.6-(b), the home service robot, named ASCCHomeBot2, was built on a Pioneer P3-DX base with approximately a 1.5m-long aluminium frame holding up a touch screen monitor used for video communication and graphical user interface. Mounted on it are a RGB and Depth (RGB-D) camera, a laser range finder (LRF), a FitPC2 minicomputer that are the same as those on the ASCCHomeBot1. In addition, ASCCHomeBot2 was equipped with an auditory system built on a microphone array and a Microsoft Windows netbook.

2.4.2 Software platform

Software platform for the robots

The software for the robots was developed based on ROS. ROS Electric was installed on Ubuntu 10.04 in the FitPC2 mini computer to run the robot software. ROS, a Linux based software framework, is an open-source, meta-operation system for robots [21]. It uses the concept of packages, nodes, topics, messages, and services and provides services similar to real operation systems, including hardware abstraction, low-level device control, implementation of commonly-used functionalities, message-passing between processes, and package management. The distributed computing feature of it can also facilitate multi-agent applications in a wireless network. In ROS, a program can be divided into different nodes which can be distributed to different computers in the same network. The driver of one component of the hardware can be treated as a node, while a data processing method can be made as one node as well. The information transferred between nodes is called a message. A node which sends messages on a topic is called a publisher and the receiving node called a subscriber has to subscribe the topic to receive that message. Currently, in ROS, all the nodes share one "ROS master". Nodes connect to other nodes directly. The ROS master only provides lookup information, much like a DNS server.

The software platform for the robots is shown in Fig. 2.7-(a). For the most basic functions in the driver layer of the robot, we utilized existing packages from ROS repositories [21] for interfacing with hardware, processing, and computing, such as *irobot_create_2_1* for driving the iRobot Create or *RosAria* for driving the Pioneer, *hukoyo_node* for interfacing the hokuyo LRF, *openni_node* and *openni_tracker* for talking to the ASUS Xtion PRO LIVE camera, and *tf* for coordinate transformations. In the service layer, three main services including SLAM, navigation, and data streaming were developed based on existing ROS packages which consist of *slam_mapping* (the implementation of grid-based Simultaneous Localization and Mapping (SLAM) using Rao-Blackwellized particle filters [25]) for simultaneously creating an occupancy grid map and localizing the robot, *move_base* for motion planning and autonomous navigation using the particle filter based localization method [26] and *amcl* for the adaptive (or KLD-sampling) Monte Carlo localization [27].

Software for Mobile Devices

We used mobile devices to control the home service robots, both locally by the elderly and remotely by the caregiver. Android tablets are low cost and portable computing platforms that can be used for this purpose. Two Motorola XOOM tablets [28] are used. The first is used on the robot as the local user interface, and the second is used at the remote end as the remote user interface. The tablet at the remote user end runs Android applications implemented with the Android SDK [29] and the *rosjava_core* library [30] to control the robot, subscribe to the video stream, and recognize voice commands, etc.

The software on the Android tablet combines two key libraries. The first library is the Android SDK. It is a free development kit which provides APIs and development tools for the Android operating system. We create an Android application to run our software. This way it can be easily integrated into any Android tablet, and could eventually be downloaded from the Android marketplace.

The second library that we used is *rosjava*, which is a complete implementation of ROS in pure Java. It also has Android support. To use *rosjava* with Android, the *android_core* package must also be included as part of the development kit. The combination of these two allows for complete integration with both ROS nodes and the Android SDK.

We also developed a graphical user interface (GUI) as shown in Fig 2.8 for robot control. Android has a series of prebuilt views that we can take advantage of, including an *ImageView* for the video, and *MapView* for displaying the 2D map, and *JoystickView* for controlling the robot, etc.

2.5 Situation awareness control based on natural human interface

The robot gateway could provide different interfaces to control the robots such as virtual joystick on GUI node, speech recognition, gesture recognition, etc. The com-

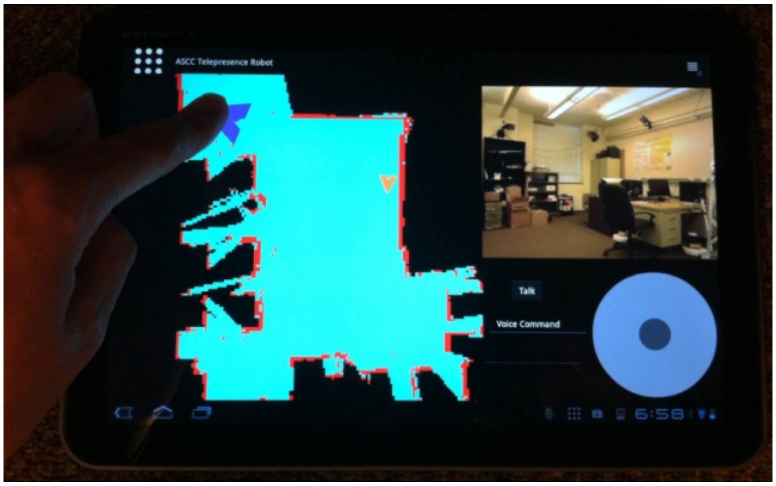


Figure 2.8: The user interface of rosjava-based Android software

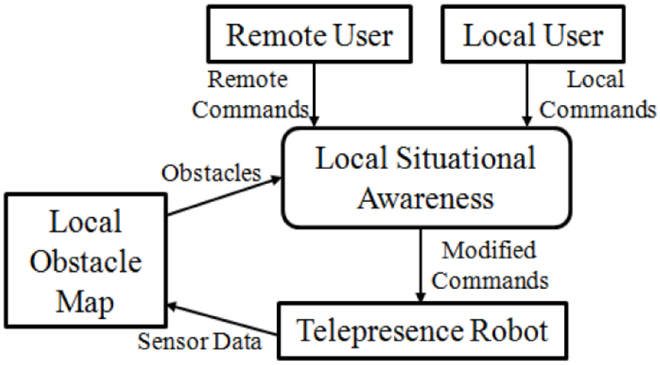


Figure 2.9: Local situational awareness control software architecture

mands from those should be fused to generate only one command to control the robot. This avoids the conflict between the interfaces, makes the controlling more natural and easier, as well as keeps interaction between the human and the robot more stable and reliable. Currently, this node has been developed by using a simple way though setting different priority levels on commands. The voice commands have higher priority than gesture commands. The further work should improve this node.

Furthermore, to help the remote user and the local user control the robot in a seamless way at the same time, the local situational awareness control (LSAC) software architecture as shown in Fig. 2.9 will be developed. It aims to find a velocity command (modified command) that integrates the commands from both the remote user and the local user. The remote commands consist of virtual joystick commands and speech commands. The local commands consist of speech commands and gesture commands. Basically, each command tries to control the robot according to specified direction and speed. The LSAC control algorithm will evaluate each command based on the probability of collision with the obstacles in the local obstacle map, which is derived from the sensor data. The command that results in the minimum probability of collision will be selected as the modified commands. There are also other ways to combine the multiple commands. We will investigate them in the future. This function is important because it enables the elderly and the remote care givers to tell the robot where to go and it will automatically move to that location in home environments. This frees up the user's time because they no longer have to drive the robot around all the time. Also, along with local situational awareness control, assisted navigation should be integrated into the robot. This assistance allows the user to drive the robot in complex environments more easily. Seamless integration of arm gesture recognition, speech recognition, and autonomous navigation could make the robot more reliable, smarter, more natural and social in behaviors, as well as easier to use.

CHAPTER 3

NATURAL HUMAN INTERFACE FOR HOME SERVICE ROBOT

This chapter describes the development of a natural human interface for ASCCHomeBots. Our goal is to add some natural human robot interface (HRI) features to the robot. It can recognize local user's hand gestures and recognize voice commands of both local and remote users.

3.1 Introduction

3.1.1 Motivation

People expect to interact with robots by natural interfaces just as they interact with other humans. Hence, natural human robot interaction has great values in developing home service robots [31]. Desai et al. [32] recommended a set of guidelines for designing personal service robots. Besides providing audio and video for communication and navigation, user interface for remote controlling, and suitable physical features, the robots should be easy to use and also be capable of autonomous navigation. They also should be designed with natural human interfaces. For example, it is important to have the capability of recognizing human's arm gestures and voice commands. These capabilities make the robots behave more socially when interacting with humans.

Besides voice-based interaction, the most effective way of communication and interaction between humans, gesture-based interaction also has wide applications including Human Machine Interaction (HMI), Human Robot Interaction (HRI) and Social Assistive Robotics (SAR). Since this technology has the potential to change the way users interact with computers or robots by eliminating input devices such as joysticks, mouse and keyboards. Moreover, the social anthropologist Edward T. Hall claims that 60% of all our communications are nonverbal [33], because gestures are widely used for expressing emotions and conveying information. Therefore we chose to develop natural human interfaces including speech recognition and gesture recognition into our robots. These technologies make home service robots effective in interacting with both the elderly and remote caregivers.

3.1.2 Related Works

In the last few years, natural human interfaces have been receiving growing interest in researches of human-robot interaction. Natural multimodal human-robot interaction using speech, head pose, and pointing gesture was integrated on a mobile robot platform for real-time human-robot interaction in a kitchen scenario [34]. However, the performance of this system seems not so high with low speech recognition accuracy less than 70% and the limited number of pointing gestures. Some researchers [35] [36] have been developing natural interfaces for tour-guide robots that have interaction functions different from those of home service robots. Moreover, cloud robotics has been receiving growing interest. Several research projects have been conducted to connect robots to the web and provide cloud services for robot control purposes. These projects include building the web-service infrastructure for robotics [37], and implementing a cloud computing framework for service robots based on ROS. As a result, the cloud can substantially improve the implementation speed of SLAM [38]. The European RoboEarth project [39] develops World Wide Web services for robots. This project provides a large cloud-enabled database where robots can share information about objects, environments, and tasks. Another example is the Google Cloud Robotics project [40] that creates robot-friendly cloud services for Android platforms. Besides developing local speech recognition, we will also develop natural human robot interaction through cloud-enabled speech recognition. Particularly, we

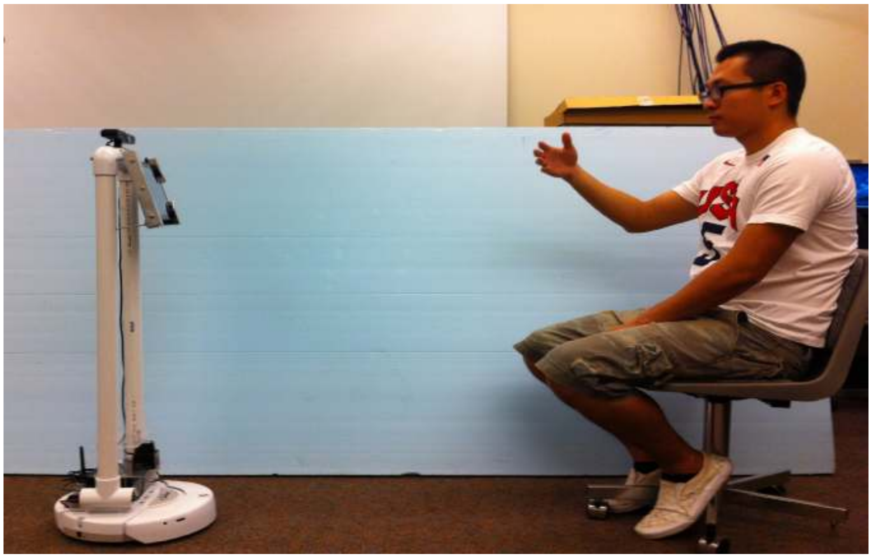


Figure 3.1: Control the robot using hand gesture

will utilize Google's speech to text service to issue commands to the robot.

Popular approaches for gesture recognition are based on visual information. Most appearance-based algorithms have used 2D images or videos for direct interpretation [41]. However, the skeleton-based gesture recognition algorithms using 3D information could benefit from identifying key elements of the body parts such as palm position or joint angles. In this work, we perform temporal human gesture recognition using an RGB-D camera. Through using this sensor, the features of the human gestures can be extracted based on depth images. Therefore, the features are not sensitive to changing of lighting condition and ordinary image noise.

In addition, we designed and implemented the *hmm* node to recognize arm gestures based on Hidden Markov Models (HMMs) [42], *voice_cmds* node to recognize voice based on both CMU Sphinx and Google cloud computing, and *cmd_fusion* node to efficiently handle the robot's motion from a virtual joystick, voice commands, and arm gesture commands.

3.2 Gesture Recognition using RGB-D camera

Hand gestures can be used to control the robot as shown in Fig. 3.1. The ASUS Xtion PRO LIVE camera gives us an RGB image as well as the depth at each pixel. Using the *openni_tracker* node, a human subject can be tracked as a rigid skeleton with fifteen joints. Gesture features are extracted from the following four joint angles: left elbow yaw and roll, left shoulder yaw and pitch. Our gesture recognition system adopts HMM for modeling the dynamics of the gestures. The detailed recognition algorithm can be found in our previous work [43], which has the following advantages:

- Easy to train: the user can simply train the system by recording the gesture to be detected.
- Person independent: the system can be trained by one person and used by others with acceptable performance.
- Orientation and distance independent: the system can recognize gestures even if the trained and recorded gestures do not have the same orientations or distance with respect to the sensor.
- Flexible speed: the system is able to recognize gestures if they are performed at a different speed compared to the training data.

We also evaluated the human interface, which includes both hand gesture recognition and voice recognition. Five arm gestures are defined in our experiments: *1: move forward, 2: turn right, 3: move back, 4: turn left, 5: stop, 6: unknown.* The real-time recognition results are shown in Fig 3.2. It is shown that when the user does multiple repetitions of the same gesture (Gesture 1) consecutively without any pause, all the gestures can be detected. The user can change the gesture from 1 to 5 successively and consecutively switch the gesture from 5 to 3 to 2 to 1 and then ends up with the initial position. The overall results show that none of the gestures misses detection

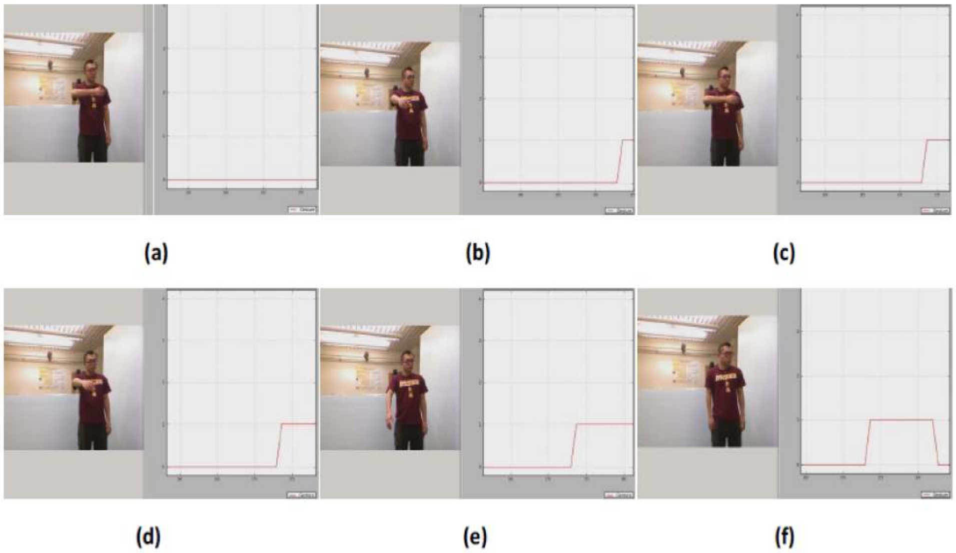


Figure 3.2: Online recognition results. The user keeps doing gesture 1 without any pause. (images are from the sensor view) [41]

and no false alarm occurs. Except for the minor delay in detection, the recognition performance is robust. The processing time of the recognition is trivial compared to that of the data collection; therefore it won't cause data missing problems while executing the recognition algorithm. A sliding window of size 20 with 50% overlap is used. A majority voting rule is applied to the results of three consecutive windows to generate one gesture recognition decision. We were able to achieve around 85% accuracy for the subject who provided training data and 73% for the non-trainer, resulting in Table 3.1 and Table 3.2.

3.3 Speech Recognition

Speech Recognition, also known as Automatic Speech Recognition (ASR), is the process to find the most probable sequence of words W given the speech signal X by means of an algorithm. By applying Bayes's theorem on conditional probabilities, the problem can be written in the following form [44]:

Table 3.1: Gesture Recognition with Non-Trainer

Ground	Gestire recognized						Test
Truth	**1**	**2**	**3**	**4**	**5**	**6**	Accuracy
1	17	0	0	0	0	4	0.8095
2	0	19	0	0	0	7	0.7307
3	0	0	15	0	0	7	0.7241
4	0	0	0	21	0	8	0.7241
5	0	0	0	0	14	5	0.7368

Table 3.2: Gesture Recognition with Trainer

Ground	Gestire recognized						Test
Truth	**1**	**2**	**3**	**4**	**5**	**6**	Accuracy
1	34	0	0	0	0	7	0.8262
2	0	36	0	0	0	8	0.8182
3	0	0	35	0	0	6	0.8573
4	0	0	0	42	0	8	0.8400
5	0	0	0	0	45	5	0.9000

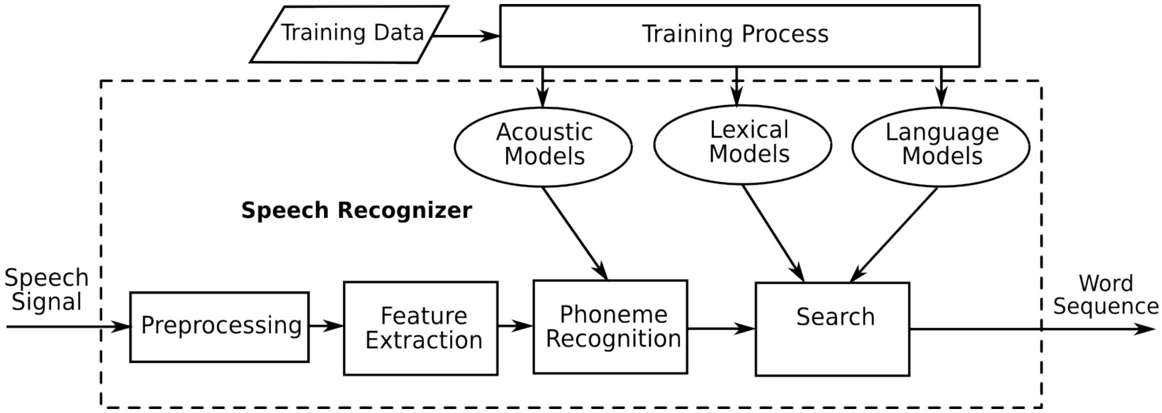


Figure 3.3: Basic model of Automatic Speech Recognition (ASR) [43]

$$\widehat{W} = arg\,max_{\mathbf{W}}\{P(W|X)\} = arg\,max_{\mathbf{W}}\{P(X|W)P(W)\} \tag{3.1}$$

The likelihood P(X|W) is determined by an acoustic model (AM) based on an HMM and the prior P(W) is determined by a lexicon model and a language model (LM) based on graph search [44].

The basic model of ASR is shown in Fig. 3.3. The phonetic recognition process starts by first preprocessing and transforming the speech signal into feature vectors. The preprocessing process reduces noise, emphasizes higher frequencies, segments speech signal into adjacent frames with the length within the range of 20 to 40 msec. The feature vectors such as common MFCC (Mel-Frequency Cepstrum Coefficient) feature vectors, LPC (Linear Predictive Encoding), PLP (Perceptual Linear Prediction coefficient extraction), and statistical parameters (first-order and second-order derivatives) are computed for each frame. Then, phoneme recognition and graph search processes are applied [45].

English words are comprised of a sequence of phoneme. Every phoneme including mono-phonemes, diphthong-phonemes, and triphthong-phonemes is modelled with a Hidden Markov Model. The AM provides a mapping between a phoneme and an HMM. The HMM-based phoneme recognition can provide an acoustic description of the signal and transform into a set of phoneme hypotheses. The lexicon model provides pronunciations or dictionaries for words. The pronunciations break words into sequences of sub-word units found in the AM. The language model provides word-level language structure that is include graph-driven grammars of stochastic N-Gram models. The graph-driven grammar represents a directed word graph where each node represents a single word and each arc represents the probability of a word transition taking place. The stochastic N-Gram models provide probabilities for words given the observation of the previous n-1 words [45].

Such a speech recognizer has been well researched and provided by open libraries or cloud servers.

3.3.1 Online speech recognition

The voice command recognition is implemented by using speech recognition services provided through Google cloud server. The models are trained and the speech recognizer runs on cloud server. Communicating with mobile devices through speech recognition has been attracting great attention recently. In other words, users can use voice to place calls, dictate text messages, start or terminate an application, set alarms etc. on their mobile devices. This idea was popularized by Apple's Siri on their iOS devices. Android tablets are becoming ever more popular [3]. They are becoming faster, more efficient, and easier to use with time.

Android, as one of the vastly used and highly maintained platforms for mobile devices, has also come a long way in implementing voice recognition capabilities. Moreover, Android devices are becoming ever more popular, faster, more efficient, and easier to use [46]. Furthermore, latest Android devices are now capable of performing enormous tasks through speech to text techniques. This voice interface operates by connecting to Google's speech recognizer cloud service which converts the spoken word into text and afterwards the text is downloaded by the device and translated to a command or simply a set of keywords to search on the web.

For issuing commands to the robot, we utilize Google's reliable and efficient speech to text service and convert voice into ROS commands. The converted text is then mapped to a valid command. Additionally, by offloading the processing to Google's cloud computing services, we do not need very sophisticated computing platforms on the remote user's end.

The speech recognition interface is capable of detecting specific keywords and acting upon them. For instance, if the user tells the robot to *go forward* the command is processed and since it's a move forward command, the robot will initiate a forward

Table 3.3: Accuracy of Online Speech Recognition

Distance	Accuracy
3 Feet	97%
4 Feet	87%
5 Feet	20%

Table 3.4: Accuracy of the local Speech Recognition

Distance	Accuracy
3 Feet	97%
4 Feet	91%
5 Feet	62%

movement. However if the user sends the command *Don't go forward!*, the robot will check the current state of the transition, and it will stop provided that it has been moving forward. To state more clearly, let's assume that the robot is rotating clockwise and the user sends the command *Don't go forward*. The speech recognition module will look up its current state (rotating) and since it is not moving forward, it will simply ignore the newly issued command and will keep its rotational movement. In contrast if the robot indeed has been moving forward and either of the command *stop* or *Don't go forward* is issued, the robot will halt its movement.

To evaluate the accuracy of the speech recognition on the robot, we increase the distance between the speaker and the tablet and observe the accuracy of the recognition process, which results in Table 3.4. The results were satisfactory provided that the distance of the speaker from the tablet is not more than 3 feet. From 3 feet onwards, the accuracy drops significantly and at 5 feet the tablet is almost unable to recognize the words due the low sensitivity of internal microphones of mobile devices. Moreover, the latency of online speech recognition depends on the speed of the network. In our experiments, although the latency of online speech recognition is large at around 1 second, it can recognize a wide range of words, phrases, and sentences that were trained by the cloud computing providers.

3.3.2 Local speech recognition

We also implemented local voice recognition based on Pocketsphinx library [47]. Currently, the recognizer requires a language model, a lexicon model, and a language model. These can be automatically built from a corpus of sentences using the Online Sphinx Knowledge Base Tool [47]. The *voice_navigation* node controls a mobile robot using commands such as *move forward, turn left, turn right, or stop.* It is easy to create a new vocabulary or corpus as it is referred to in PocketSphinx. First, we need create a simple corpus file with one word or phrase per line. The list of phrases is as follows: *stop, panic, shut down, slow down, speed up, ahead, back, go back, rotate left, rotate right, turn left, left, turn right, right, quarter speed, quarter, half, half speed, full speed, full, pause speech, continue speech.* Before we can use this corpus with PocketSphinx, we need to compile it into special dictionary and pronunciation files. This can be done using the online CMU language model (lm) tool located at: http://www.speech.cs.cmu.edu/tools/lmtool-new.html. The online tool returns the files that define vocabulary as a lexicon model and a language model that PocketSphinx can understand.

The *voice_navigation* node was developed using python to recognize speech commands and map those commands to control the robot. Another feature of this node is that it will respond to the two special commands *pause speech* and *continue speech.* If you are voice controlling your robot, but you would like to say something to another person without the robot interpreting your words as movement commands, just say *pause speech.* When you want to continue controlling the robot, say *continue speech.*

Local speech recognition can produce results very fast. The latency is less than

0.5 seconds. In addition, to evaluate the accuracy of the local speech recognition on the robot, we increase the distance between the speaker and the microphone and observe the accuracy of the recognition process, which results in Table 3.4.

CHAPTER 4

DEVELOP THE AUDITION SYSTEM FOR THE HOME SERVICE ROBOT

This chapter designs and implements an audition system for a home service robot. The hardware of the audition system was set up using microphones. We successfully implemented the software for this system that realizes audition services for high level applications such as sound source position estimation, multiple source speech recognition, and human assisted sound classification.

4.1 Introduction

One important communication channel in human daily life is sound including voice and non-voice. Therefore it is desirable to equip home service robots with sound processing capability. The robot needs to know where the sounds are coming from even when multiple sound sources exist. This can help the robots respond to human commands and events more accurately and quickly. For example, a service robot for elderly care needs to be able to respond to the request for help by quickly localizing where the request voice comes from in the house. Moreover, it is very important for the home service robots to be able to understand human speech even with the existence of other sound sources. It is also critical that the robots have the capability to understand the sounds generated by human's daily activities, such as using toilet, washing hands, cooking, etc. Such a human-aware capability frees the robot to do its daily routine work, while being able to attend the human more proactively and effectively. Therefore, the home service robots need to be equipped with such auditory capability to better service the elderly. Such a capability can also be extended to other applications such as surveillance, reconnaissance, search and rescue etc.

Humans have strong auditory capability which enables them to not only understand the voice and the environmental sounds, but also use their listening and repeating skills to sort through the incoming information [48]. The home service robots should also have such auditory ability. As discussed in the previous chapter, speech recognition has been well researched and there are even open source software available, such as Pocketsphinx [47] and Julius [49]. However, speech recognition in a multiple sound sources environment is still challenging. On the other hand, environmental sound event recognition is even harder, due to the diversity of the sounds associated with the same event. For example, even the same event of an elderly falling on the floor can create different sounds, depending on where the fall occurs. This makes it extremely hard to preprogram the robot with a small set of training data from an existing database. This example tells us that such a robot should gradually learn the auditory events in its unique environment, and whenever possible, get assistance from humans who can provide guidance on the auditory learning process.

We propose that by putting a human in the loop of the auditory learning process, a robot can better understand and adapt to its environment more quickly. Using a microphone array, the robot is able to separate multiple sound sources. Then the robot classifies the separated sounds into voice and non-voice. The voice sound can be recognized locally. The non-voice sound can be sent to a human, for example, the care giver, for recognition and labelling. Since only non-voice sound is sent to outside, the privacy of the elderly can be protected. With more and more labelled environmental sound data, the robot can train its sound recognition algorithm to achieve better accuracy.

This chapter proposes and implements an open hardware/software platform for the above described auditory learning for a home service robot.

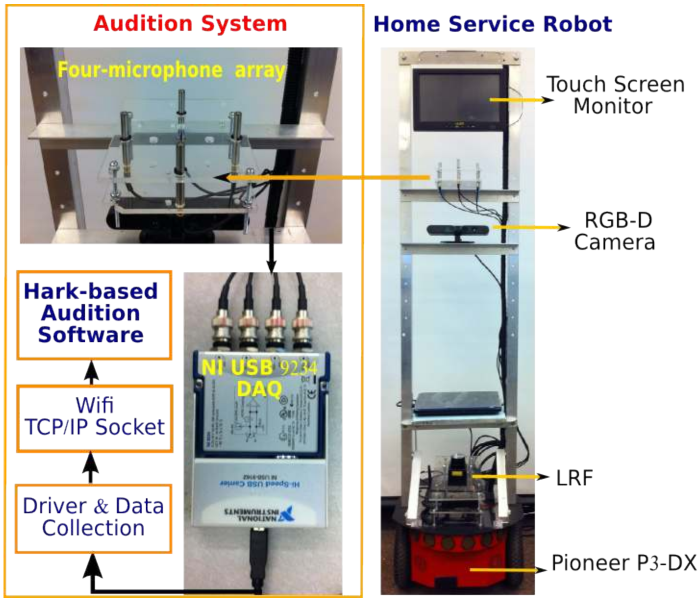


Figure 4.1: Audition System for home service robot

4.2 Hardware and Software of the Audition System

Audition Software

The hardware of audition system as shown in Fig. 4.1 was built on 4 G.R.A.S IEPE (Integrated Electronic Piezoelectric) microphones and an NI USB-9234 DAQ (Data Acquisition). This set of microphones has high-sensitivity at 50mV/Pa, a wide frequency range up to 20 kHz, and a large dynamic range topping at around 135 dB. The DAQ is a USB-based four-channel C Series dynamic signal acquisition module for high-accuracy audio frequency measurements from IEPE and non-IEPE sensors. It can deliver 102 dB of dynamic range, incorporate programmable AC/DC coupling and IEPE signal conditioning for accelerometers and microphones, as well as digitize signals at rates up to 51.2 kHz per channel with built-in antialiasing filters that automatically adjust to the sampling rate. The DAQ is connected to a Windows 7 netbook running the DAQ driver.

The audition software in the device layer and service layer was based on HARK [50]. Developed at Kyoto University, HARK, first released in Apr. 2008, is an open-source

Algorithm 1: Audio Data Collection

```
1. Initialization phase:
- Initialize: Sensitivity, Frame length, Sampling rate, Number of channels,
cut-off frequencies, FIR filter type.
2. Design FIR filter (Highpass/Lowpass/Bandpass):
- Compute filter coefficients using the window method
- Compute FFT of filter coefficients
3. Create a channel and a data acquisition section for NI USB9234:
- Create a task: taskHandle
- Create channels that use microphones and add the channels to the task
4. Set up the TCP/IP socket connections to HARK program:
- Create a client socket to control the data recovering proceeds through control
messages
- Create a client socket for sending audio data out.
- Wait for the recording request.
5. Recording, filtering, sending data out until receiving the
stop-recording message from AudioStreamFromMic:
- Read the buffer inside NI USB-9234 board and write to data buffer
- Do the filter in frequency domain
- Send the filtered data to AudioStreamFromMic via data socket
6. Stop the task and close socket connections.
```

robot audition software consisting of acoustic signal processing module, sound source localization module, sound source separation module, and automatic speech recognition module for various microphone array configurations. In the hardware layer, the DAQ is connected to a netbook running Windows 7 and NI-DAQmx9.7 software including its driver. NI-DAQmx9.7 provides APIs to develop the program to interactively create channels and tasks, write to or read data from DAQs. Developed in

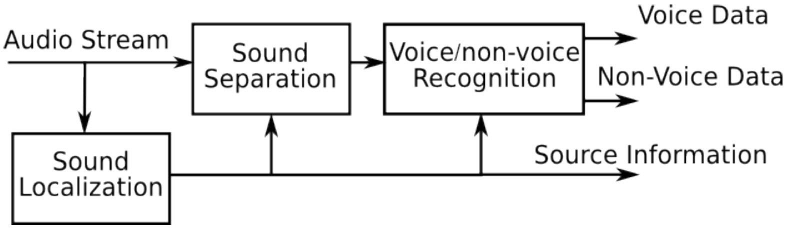


Figure 4.2: Audition Services

Visual C++ based on NI-DAQmx C Library, the data collection program captures the data from the microphone array and sends it to the HARK program running on another Ubuntu computer through WiFi TCP/IP socket. This Ubuntu computer is installed with HARK and HARK-ROS to develop audition software in the service layer. As shown in Algorithm 1, the data collection module reads data from the microphone array, establishes the WiFi TCP/IP socket to communicate with the *AudioStreamFromMic* block in HARK, filters them out, and sends them to that block for further processing.

4.3 Audition services

As shown in Fig. 4.2, audition services perform sound localization, separation, and voice/non-voice recognition from the four-channel audio stream coming from the data collection module. Those services are developed based on HARK.

4.3.1 Sound Localization

Direction-of-arrival (DOA) in the horizontal plane is estimated by the Multiple Signal Classification (MUSIC) method [51], which has shown the best performance. This method localizes sound sources based on source positions and impulse responses (transfer function) of microphones. The transfer function generally varies depending on the shape of the room and the positional relations between microphones and sound sources [52]. Therefore it is difficult to estimate it. However, when ignoring acoustic reflection and diffraction, in the case that the relative position of microphones and sound sources is known, the transfer function is limited only to the direct sound $H_D(k_i)$ calculated by the following Equation [52]:

$$H_{Dm,n}(k_i) = exp(\frac{-j2\pi\omega_i}{c} r_{m,n}) \tag{4.1}$$

where c is the speed of sound; ω_i is the frequency in the frequency bin k_i; $r_{m,n}$ is the difference between the distance from the microphone m to the sound source n and the distance from the reference point of the coordinate system to the sound source n.

Sound localization implemented the GEVD (Generalized EigenValue Decomposition) method [53] that is based on MUSIC, but a noise correlation matrix is additionally used in order to suppress environmental noises. More details about sound localization can be found in [53] [54].

4.3.2 Sound Separation

The sound that is emitted from N sound sources is affected by the transfer function $H(k_i)$ in space and observed through M microphones as expressed by Equation 4.2.

$$X(k_i) = H(k_i)S(k_i) + N(k_i) \tag{4.2}$$

where $S(k_i)$ is the sound source complex spectrum corresponding to the frequency bin k_i; $N(k_i)$ is the additive noise that gets into each microphone.

The matrix of a complex spectrum of separated sound $Y(k_i)$ is obtained from the following equation:

$$Y(k_i) = W(k_i)X(k_i) \tag{4.3}$$

The separation matrix $W(k_i)$ is estimated by Geometric-Constrained High-order Source Separation (GHDSS) [55], which has the highest total performance in various acoustic environments. With the source direction from the sound localization, the separated sound $Y(k_i)$ is likely close to its sound source $S(k_i)$.

4.3.3 Voice/Non-voice Recognition

The separated sounds are recorded into files and classified into voice and non-voice by the voice/non-voice recognition (VNR) algorithm. Recently, the support vector machine (SVM) algorithm has proved highly successful in a number of binary classification problems. SVM discriminates the data by creating boundaries between classes rather than estimating class conditional densities, it may need considerably less data to perform accurate classification. The boundary is the optimal decision hyperplane that has the largest distance to the nearest training data points of any class. These points are called support vectors. In order to apply SVM, the kernel function is applied to transform non-linear and high-dimensional feature vectors into simpler feature vectors that can be classified by the optimal decision hyperplane using linear discriminant functions. The kernel function most popularly used in SVM for audio applications is the Gaussian radial basis function (RBF) as follows:

$$K(x_i, x_j) = exp(-\gamma \|x_i - x_j\|^2) \tag{4.4}$$

where γ is a control parameter estimated from the variance of the distribution function of training data. RBF-VSM aims to construct the decision function for the data point x based on N support vectors $\{\mathrm{x}_k\}_{k=1}^N$ and labels $\{\mathrm{y}_k\}_{k=1}^N$ as follows:

$$y(x) = sign[\sum_{k=1}^{N} \alpha_k y_k K(x_k, x) + b] \tag{4.5}$$

where α_k is the weight assigned to the support vector x_k, b is a constant bias.

The above decision function can be trained based on least-square method [56] implemented in the Matlab statistics toolbox.

The radial basis function (RBF)-based SVM using Mel Frequency Cepstral Coefficients (MFCC) feature was implemented for VNR based on Voice Active Detection proposed in [57]. As shown in Fig. 4.3, VNR includes the training process and the recognition process. The audio training data consist of voice and non-voice segments

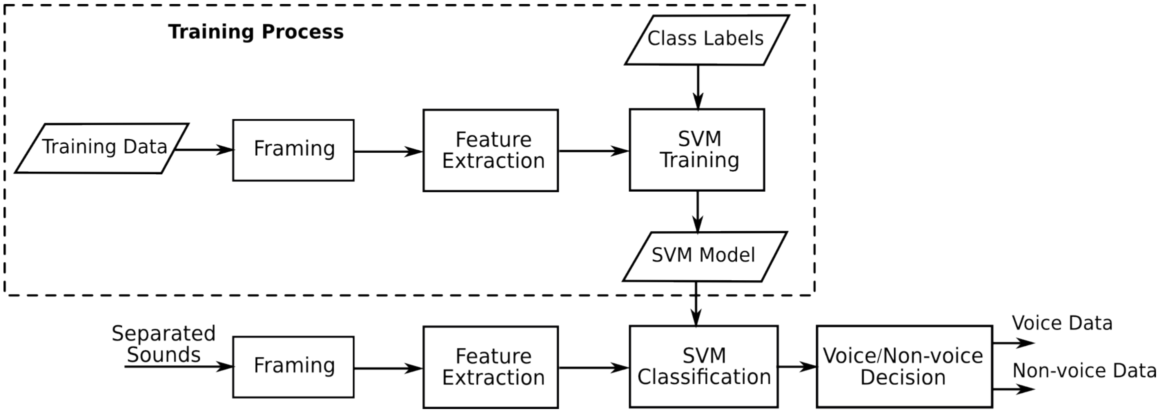


Figure 4.3: Voice/Non-voice recognition based on Voice Active Detection proposed in [54]

that are labelled (-1) and (1), respectively. In the training process, audio training data are decomposed into frames. The 36-MFCC feature vectors are computed for the frames based on the MFCC algorithm from the Auditory Toolbox by Slaney [58]. The RBF-SVM is trained using these feature vectors and their class labels. This trained SVM model can classify frames of separated sounds into voice or non-voice. Voice/Non-voice decision can classify separated sounds into voice data or non-voice data based on the rates of voice and non-voice frames in each sound. The sound is classified into voice data if its voice frames dominate, otherwise it is classified into non-voice data.

4.4 Auditory Applications

4.4.1 Sound source position estimation

The sound direction can be estimated using the above sound localization method. Using only one stationary microphone array, it is hard to estimate the sound source position. However, the home service robot can move around and therefore it is possible to use triangulation to localize the sound. Fig. 4.4 shows an example of using triangulation to estimate the positions of two sound sources. If the robot can mea-

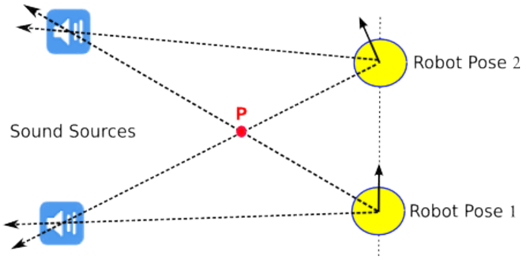


Figure 4.4: Sound position estimated by triangulation

Algorithm 2: Sound Source Position Estimation

```
1. Measure direction data in N steps:
for step = 1 to N do
    - Do sound localization in $T$ seconds
    - Compute average and Root-Mean-Square error (RMSE) of the direction data
    - Remove data with large errors
    - Generate the goal point based on the direction data and the map
    - Navigate the robot to the goal point
2. Calculate intersection points:
- Calculate intersection points by triangulation between the N groups of direction.
- Remove the intersection points outside the map.
3. Random and select data:
for $n = n_0$ to Number of Intersection Points do
    - Pick up n random points in intersection point groups
    - Calculate RMSE for each of n-point subgroups
    - Search n-point subgroups to find the one with the least RMSE
4. Calculate sound positions from average of the subgroups with the least RMSE.
```

sure the sound direction at two different positions on the 2D map that was created by SLAM, the sound position can be estimated by calculating the intersection of two sound-direction vectors. This method may create a undesired intersection point like point P as shown in Fig. 4.4. However, this point moves when the robot measures at another position. Therefore it can be eliminated since we assume that the sound sources are stationary. With multiple steps, the robot can improve the accuracy of position estimation using the RANdom SAmple Consensus (RANSAC) algorithm [59]. The sound source position estimation algorithm is shown in Algorithm 2.

4.4.2 Multiple source speech recognition

The voice sounds that are separated and classified in the service layer are sent to the recognizer which is implemented based on the Pocketphinx [60] and has been presented in chapter 3. Similarly, common commands such as *move forward, turn left, turn right, go back, stop, come here, etc* were tested to drive the robot around using voice commands in the background of another sound such as music, TV sound.

4.4.3 Human assisted sound recognition

Many non-voice sounds relate to human activities at home, such as taking shower, flushing a toilet, soaping hands, washing hands, or brushing teeth in the bathroom; using a microwave oven, boiling water, or frying pan in the kitchen; or sorting dishes or pouring water into a cup in the dinning room. Recognizing these sounds can help robot understand human activities. However, due to the lack of sufficient training samples in the individual home environment, it is very hard to achieve satisfactory non-voice sound recognition. Therefore we propose to let the robot and the human care giver collaborate to recognize it. Basically, the robot sends the segment of non-voice sound to the care giver, who then recognizes it and labels it through a user interface. Such an interface can be on a computer, or a mobile device such as a tablet or smart phone.

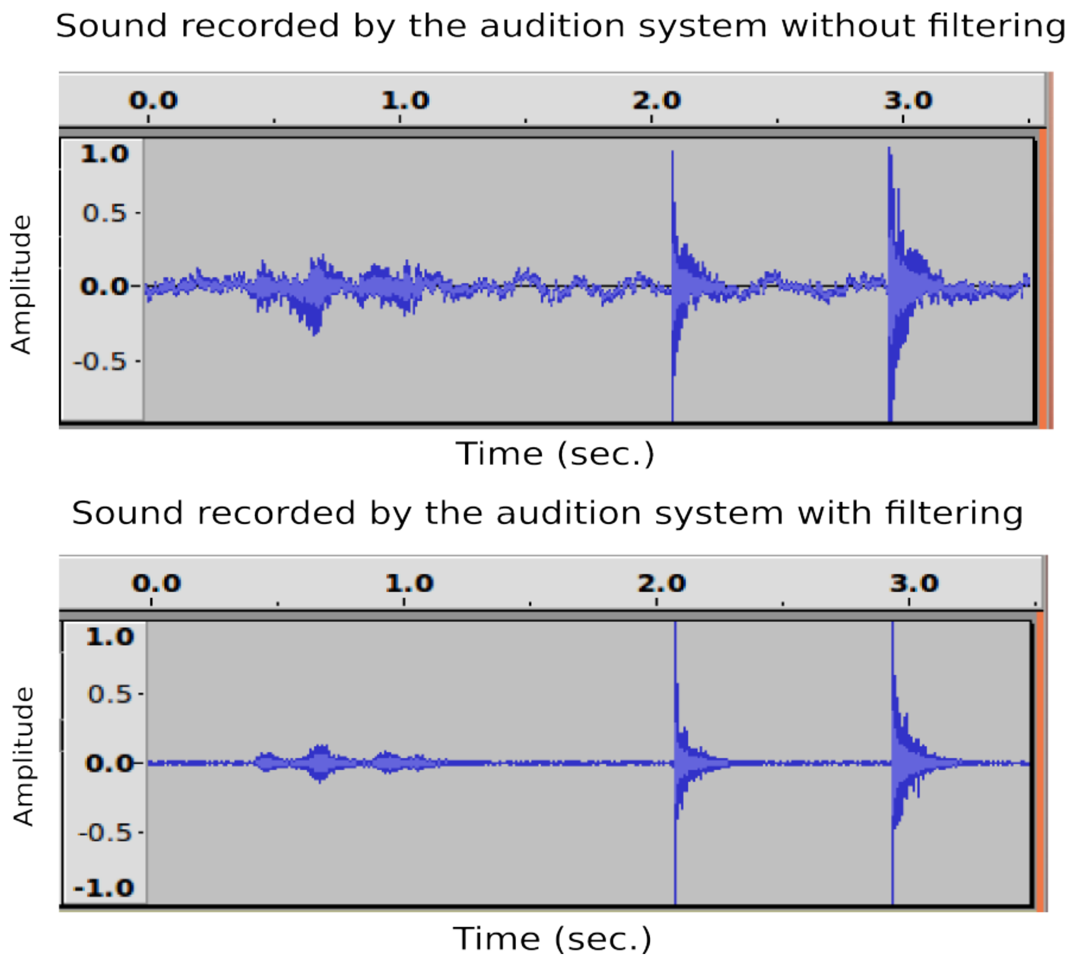
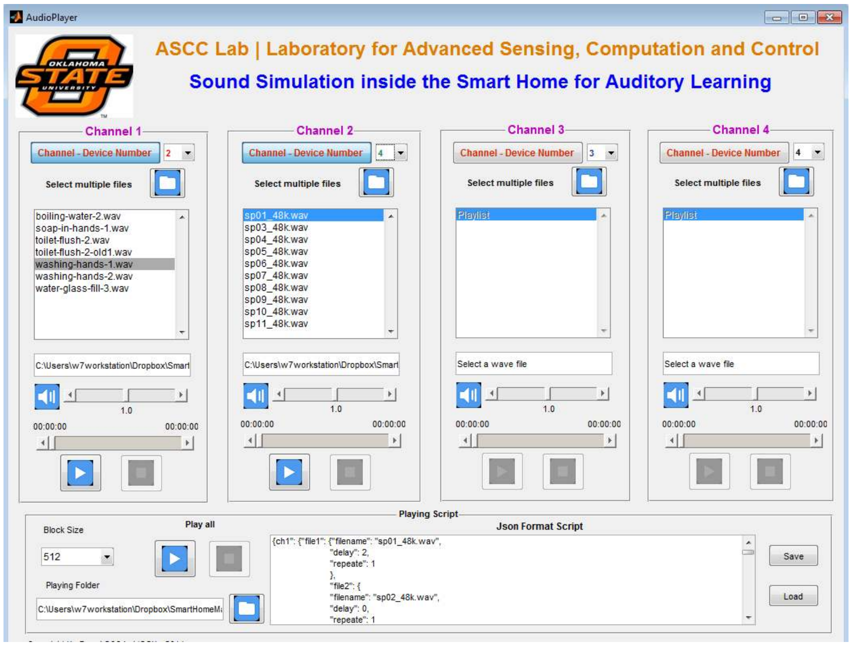


Figure 4.5: Sound recorded by the audition system

4.5 Experiments and Results

4.5.1 Audio Data Collection

On the data collection module, the microphones were tested at different sensitivity levels from 50mV/Pa to 200mV/Pa. Different FIR filters (HPF - High Pass Filter, LPF - Low Pass Filter, and BPF - Band Pass Filter) also were tested at different cut-off frequencies. The system can work in real-time at a highest sampling rate up to 51.2 Khz. However, the sampling rate is fixed at 17066 Hz to be compatible with HARK. The other parameters can be set beyond background noise level and sound source power level. In our Smart Home testbed, the sensitivity is set at 100mV/Pa


AudioPlayer
ASCC Lab | Laboratory for Advanced Sensing, Computation and Control
Sound Simulation inside the Smart Home for Auditory Learning
Channel 1
Channel 2
Channel 3
Channel 4
Select multiple files
Playing Script
Play all
Json Format Script
Save
Load

Channel 1

Channel - Device Number: 2

boiling-water-2.wav
soap-in-hands-1.wav
toilet-flush-2.wav
toilet-flush-2-old1.wav
washing-hands-1.wav
washing-hands-2.wav
water-glass-fill-3.wav

C:\Users\w7workstation\Dropbox\Smart

1.0

00:00:00 00:00:00

Channel 2

Channel - Device Number: 4

sp01_48k.wav
sp03_48k.wav
sp04_48k.wav
sp05_48k.wav
sp06_48k.wav
sp07_48k.wav
sp08_48k.wav
sp09_48k.wav
sp10_48k.wav
sp11_48k.wav

C:\Users\w7workstation\Dropbox\Smart

1.0

00:00:00 00:00:00

Channel 3

Channel - Device Number: 3

Playlist

Select a wave file

1.0

00:00:00 00:00:00

Channel 4

Channel - Device Number: 4

Playlist

Select a wave file

1.0

00:00:00 00:00:00

Block Size: 512

Playing Folder: C:\Users\w7workstation\Dropbox\SmartHomeM

```
{ch1": {"file1": {"filename": "sp01_48k.wav",
                  "delay": 2,
                  "repeate": 1
                  },
                  "file2": {
                  "filename": "sp02_48k.wav",
                  "delay": 0,
                  "repeate": 1
```

Figure 4.6: Sound simulation inside the smart home

and BPF with frequency range from 50 Hz to 8 kHz is utilized. The audition system working at this configuration can filter out the background noise and pick up the sound signal at low power as shown in Fig. 4.5.

4.5.2 Sound Simulation System

We developed a system to simulate the multiple sound events in the smart home testbed. The various sounds in the bathroom, kitchen room, living room, bedroom were recorded or collected from the Internet. Our sound library (SoundLib) has 50 sound event files, 50 speech files including 30 clean files of 30 sentences in NOIZEOUS database [61]. A GUI shown in Fig. 4.6 was developed to play multiple sound event files at the same time on different speakers. Currently, it can play different sounds simultaneously in four channels. For example, it can play both the TV sound in the living room and the shower sound in the bathroom, or play a sequence of multiple sound events related to the cooking activity in the kitchen. The script or schedule for playing sound events was written in JSON (JavaScript Object Notation) format.

The following script plays in the speaker 1 the five speech sentences from sp11_48k.wav to sp15_48k.wav and in the speaker 2 the washing hands sound from wahsing-hands-1.wav:

```
{ ”ch1”:{
    ”file1”:{”filename”:”sp11_48k.wav”, ”delay”:0, ”repeat”:1},
    ”file2”:{”filename”:”sp12_48k.wav”, ”delay”:0, ”repeat”:1},
    ”file3”:{”filename”:”sp13_48k.wav”, ”delay”:0, ”repeat”:1},
    ”file4”:{”filename”:”sp14_48k.wav”, ”delay”:0, ”repeat”:1},
    ”file5”:{”filename”:”sp15_48k.wav”, ”delay”:0, ”repeat”:1},
    },
    ”ch2”:{”file1”:{”filename”: ”washing-hands-1.wav”,”delay”: 0, ”repeat”:1
    },
    ”ch3”: {},
    ”ch4”: {}}
```

4.5.3 Sound Localization and Separation

Sound localization and separation was tested using the sound simulation system and OptiTrack system. In the experiments, we used transfer functions that were estimated by geometrical calculation based on the microphone array configuration as shown in Fig. 4.7.

The sound sources can be localized in 360° at reasonable accuracy. As shown in Table 4.1, the Mean Absolute Error (MAE) of sound source localization is less than 1.6° at 0.5 m and less than 3.5° at 3 m away from the robot.

As shown in Fig. 4.8, two voice and non-voice sources can be separated. In this experiment, the speaker 1 and the speaker 2 placed at -45° and 45°, respectably, played the five speech sentences from sp11_48k.wav to sp15_48k.wav and the washing-hands sound, respectively. The system is able to localize and track both sources correctly

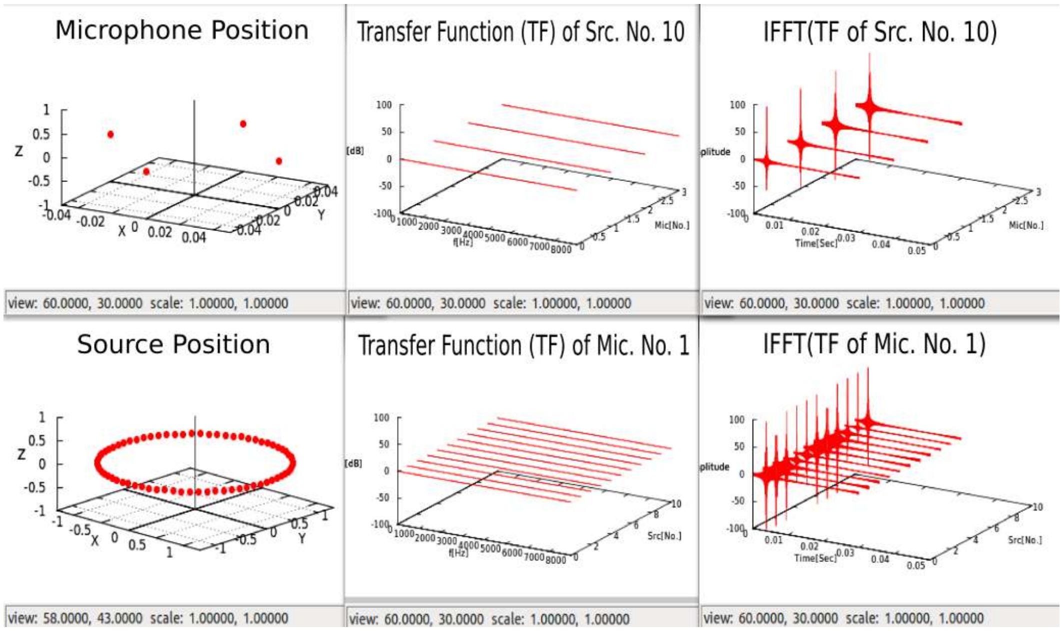

Microphone Position
Transfer Function (TF) of Src. No. 10
IFFT(TF of Src. No. 10)
Source Position
Transfer Function (TF) of Mic. No. 1
IFFT(TF of Mic. No. 1)
Z
X
Y
[dB]
f[Hz]
Mic[No.]
Src[No.]
plitude
Time[Sec]
view: 60.0000, 30.0000 scale: 1.00000, 1.00000
view: 58.0000, 43.0000 scale: 1.00000, 1.00000

Figure 4.7: Graphic representation of the microphones position, source location, and transfer functions

Table 4.1: Mean Absolute Error (MAE) of sound localization

Direction	Distance			
	0.5m	1m	2m	3m
0°	1.2°	1.8°	2.6°	3.4°
±45°	1.5°	1.6°	2.2°	3.2°
±90°	1.3°	1.7°	2.4°	3.3°
±135°	1.4°	1.7°	2.3°	3.2°

and then separate them into two sounds that have the similar waveforms as their original sounds. Although each sound is still interfered by the other, the separated sounds can work with voice/non-voice and multiple-source speech recognition.

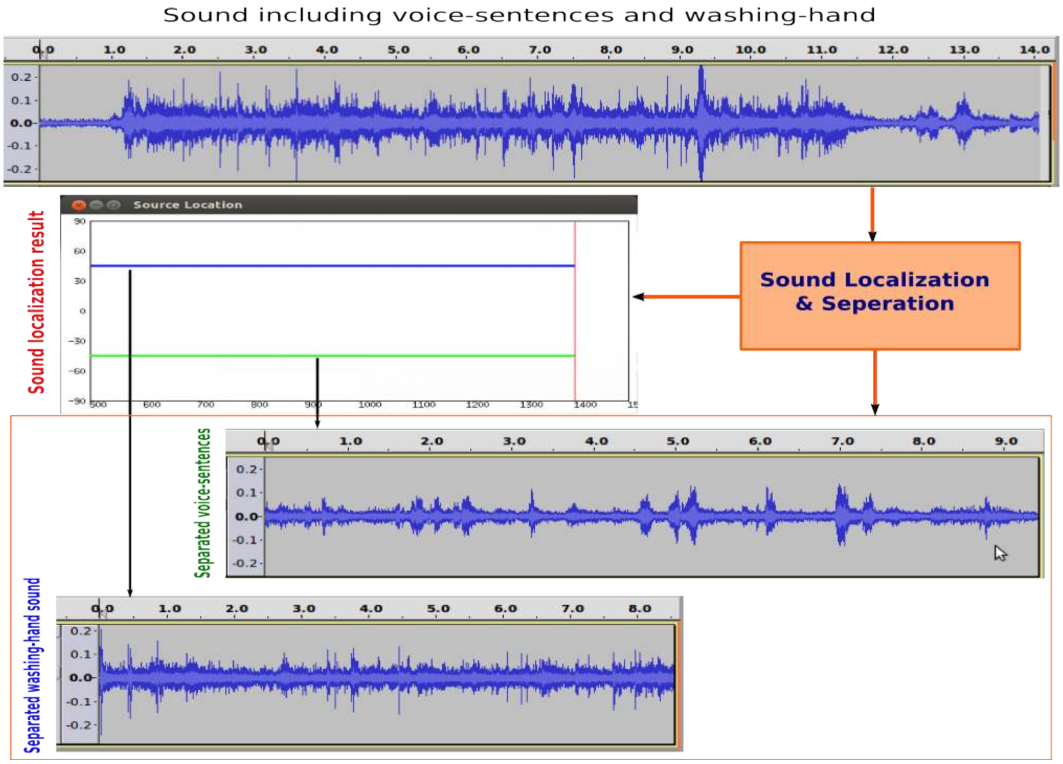

Sound including voice-sentences and washing-hand
Source Location
Sound localization result
Sound Localization & Seperation
Separated voice-sentences
Separated washing-hand sound

Figure 4.8: Results of sound localization and separation

4.5.4 Voice/Non-voice Recognition

Event sound and voice sound in our Soundlib are randomly divided into training and testing data. The input signal is segmented with the frame length of 512 samples, extracted into 36-MFCC feature vectors. Those MFCC vectors are used for SVM. We labelled non-voice for all frames in event sounds and voice for all frames in voice sounds. The event sounds and voice sounds in testing data were divided into voice/non-voice pairs that were simultaneously played by two speakers. The robot successful separated each pair into two different sounds. Those sounds were used to test our SVM-based VNR algorithm. The percentage of correctly recognized frames of 95% voice and non-voice sounds is from 70% to 80%. Therefore, when the thresholds of voice/non-voice decisions are set below 70%, the voice/non-voice recognition results of the robot can reach the accuracy of 95% for the whole separated sound segment. As shown in Fig. 4.9 and Fig. 4.10, more than 75% frames of the sepa-

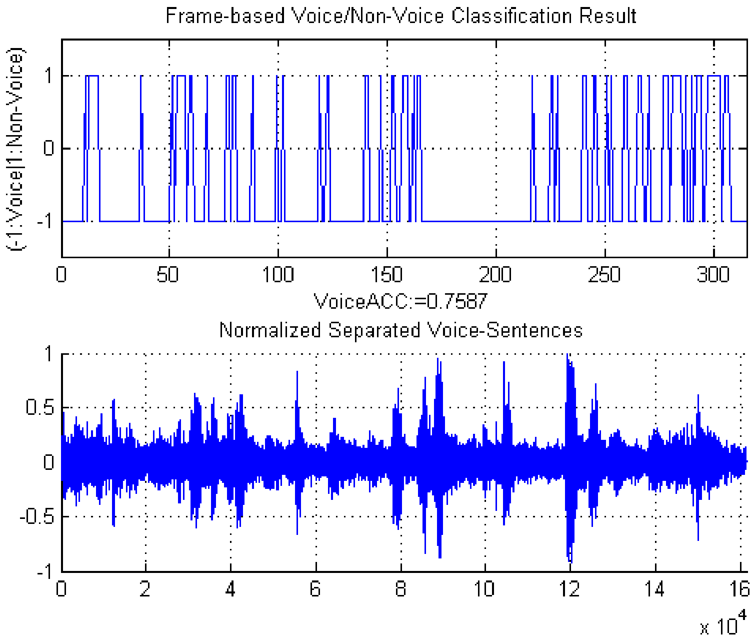


Figure 4.9: Voice/Non-voice Recognition of Separated voice-sentences

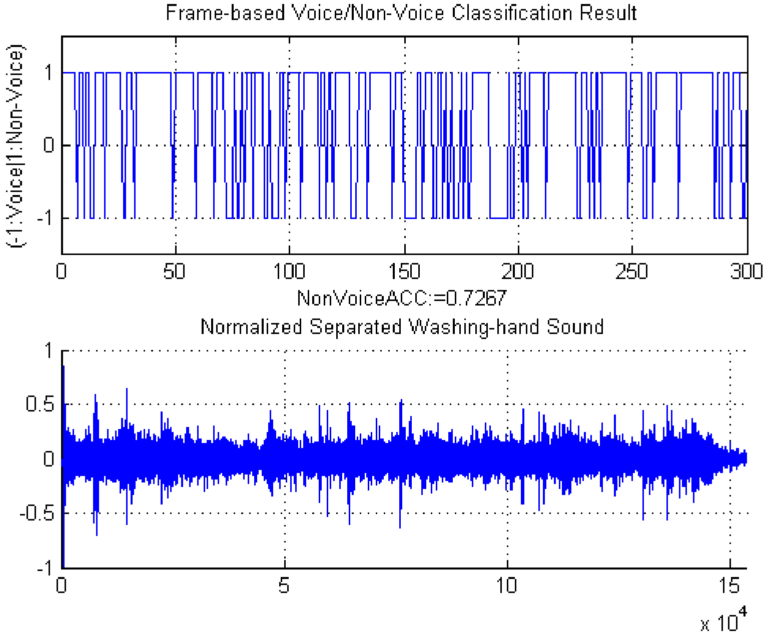


Figure 4.10: Voice/Non-voice Recognition of Separated washing-hand sound

rated voice-sentence sound are voice, and more than 72% frames of the separated washing-hand sound are non-voice.

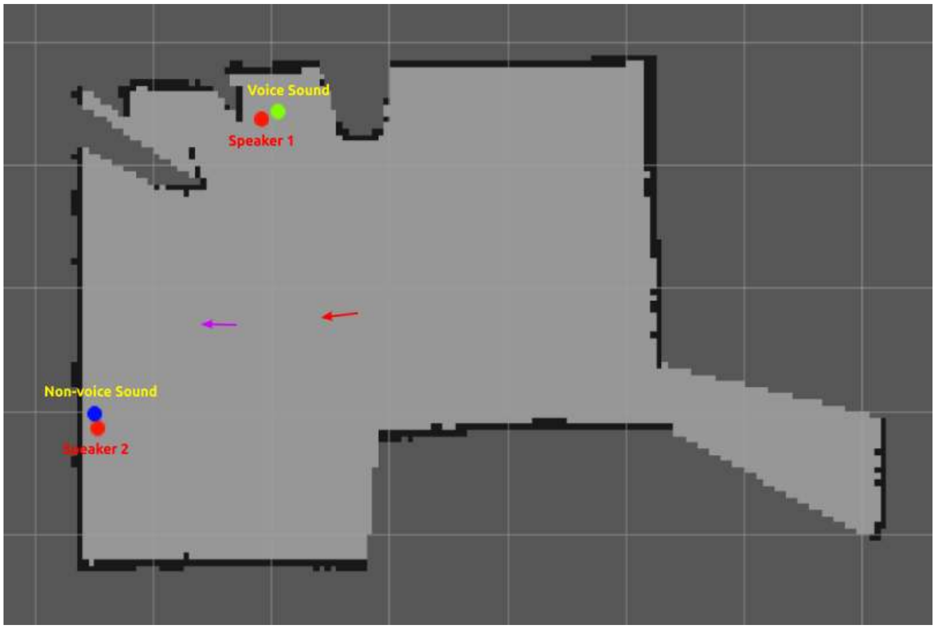


Figure 4.11: Sound position estimation by triangulation

Table 4.2: Mean Absolute Error (MAE) of Sound Source Positions

Direction	Distance		
	1m	2.0m	3.0m
0°	213mm	313mm	417mm
30°	146mm	257mm	341mm
45°	123mm	225mm	278mm
90°	102mm	141mm	195mm
150°	228mm	309mm	397mm
180°	291mm	366mm	423mm

4.5.5 Sound Source Position Estimation

Two speakers in the living room and the kitchen simultaneously played voice sound and non-voice sound continuously. Fig. 4.11 presents the results of sound positions estimated by the robot using triangulation. The ground truth positions of the two speakers are tracked by OptiTrack and are represented by the red dots in the 2D

Table 4.3: Accuracy of Speech Recognition without Separation

Degree	0.5m	1.0m	1.5m	2.0m	2.5m	3.0m
0°	21%	15%	5%	0%	0%	0%
45°	15%	0%	0%	0%	0%	0%
90°	21%	15%	5%	0%	0%	0%
135°	26%	18%	5%	0%	0%	0%
180°	27%	21%	5%	0%	0%	0%

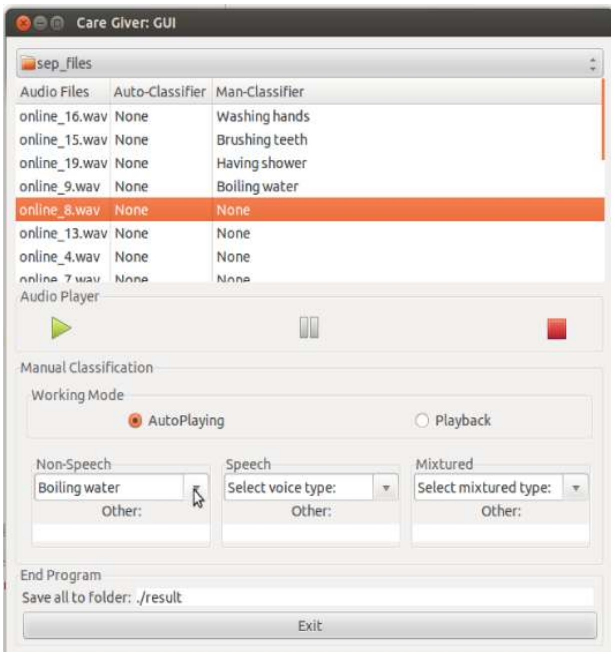
Table 4.4: Accuracy of Speech Recognition with Separation

Degree	0.5m	1.0m	1.5m	2.0m	2.5m	3.0m
0°	81%	76%	52%	40%	20%	0%
45°	25%	16%	7%	0%	0%	0%
90°	81%	76%	52%	40%	20%	0%
135°	79%	69%	48%	37%	28%	0%
180°	82%	64%	44%	33%	15%	0%

map that was created by the robot. The red arrow and the cyan arrow represent the robot poses in the 2D map at the begin and the end of the triangulation step. The estimated positions are presented in the map by the blue dot for the non-voice source and the green dot for voice-source. The error of estimated positions depends on the initial position of each sound source as shown in Table 4.2.

4.5.6 Multiple Source Speech Recognition

In our experiment, the speech recognizer produced poor results of recognizing speech played from a speaker because of accumulated distortion and noise. However it worked relatively well with the real human voice. We tested some common human voice



Audio Files	Auto-Classifier	Man-Classifier
online_16.wav	None	Washing hands
online_15.wav	None	Brushing teeth
online_19.wav	None	Having shower
online_9.wav	None	Boiling water
online_8.wav	None	None
online_13.wav	None	None
online_4.wav	None	None
online_7.wav	None	None

Working Mode: ● AutoPlaying ○ Playback

Non-Speech: Boiling water
Other:

Speech: Select voice type:
Other:

Mixtured: Select mixtured type:
Other:

Save all to folder: ./result

Figure 4.12: The user interface for human labelling of non-voice sound

commands such as *move forward, turn left, turn right, go back, stop, come here, slow down, speed up, help me* at different locations with background sound of the TV or music played by the speaker that was placed at 3.0m and 45° from the robot. The accuracy of speech recognition without separation and with separation is shown in Table 4.3 and Table 4.4, respectively. Without separation the accuracy is only around 20%, although the human is very close to the robot. The robot can not recognize the commands of the human at more than 1.0m away from the robot. The result is better if sound separation is utilized. The accuracy is around 80% at 0.5 m and around 40% at 2 m. However, the speech recognizer produced poor results when the human is at 45°, the same direction as the speaker. In this direction, the human voice could not be separated from background sound since two sound source were at the same directions.

4.5.7 Human Assisted Sound Classification

In our experiment, we were able to successfully assist the robot in labelling non-voice sounds from another lab room. The GUI on a computer as shown in Fig. 4.12 was developed based on TCP/IP socket peer-to-peer connection between the robot and the care giver. Each of separated non-voice sounds was sent to the care giver in five-second segments that were saved into .wav files for offline playing. They also autonomously played on the GUI and the care giver selected the label by clicking on the combobox or input the new labels. The labelling results were sent back to the robot using JSON format files that can be used as further training data set. The separated non-voice sounds were sent to the GUI with the latency less than 0.5 seconds. Therefore, this application can be improved for human-robot collaboration in exploring abnormal sounds in home environments.

CHAPTER 5

CONCLUSIONS AND FUTURE WORKS

5.1 Conclusions

In this work we have proposed a robot-integrated smart home environment testbed called the ASCC Smart Home Testbed and developed two home service robot platforms. The testbed consists of four components: the robot-integrated smart home environment, the caregivers, the cloud servers, and the experiment environment. The open layered architecture was proposed for the development of the software platform. Furthermore, this thesis proves that it is possible to create an affordable home service robot platform that can be controlled with the natural human interfaces including both voice recognition and gesture recognition. Using ROS repositories, Android API, rosjava core, and Google cloud computing services, the robot platform was equipped with local and remote user interface, gesture recognition, speech recognition, etc. Moreover, the open platform of audition system of the home service robot was developed. In addition, three functionalities of the audition system were implemented: sound source position estimation based on triangulation, multiple speech recognition based on the voice/non-voice recognizer and speech recognition, and human assisted non-voice sound classification. We tested and evaluated the above three functionalities. We proved that human and robot can collaborate to facilitate non-voice classification. Overall, the testbed and the home robot platforms have the potential to be deployed into the real world and have impact on developing social intelligence for robot companions.

5.2 Future Works

In the future, the testbed will be fully implemented and the home service robots will be improved in 3D mapping and autonomous navigation. Additionally, cooperation and collaboration between the robot and the remote caregiver is a challenging task due to the limited information provided to the remote caregiver, such as the limited field of view, and the communication delay and unreliable connection. Future researches will address these problems by equipping the robots with capabilities of local situational awareness control (LSAC) that integrates autonomous navigation, assisted driving, obstacle avoidance, combination of commands from both the remote caregiver and the local human for collision-free navigation. The future work will also develop algorithms for the robot to understand and predict the human activities and intentions through sound events and speech which will make the robot more efficient in serving the elderly at their homes.

BIBLIOGRAPHY

[1] C. Hacks, "e-health sensor platform for arduino and raspberry pi [biometric / medical applications]," *http://www.cooking-hacks.com/documentation/tutorials/ehealth-v1-biometric-sensor-platform-arduino-raspberry-pi-medical/*, 2014.

[2] World Health Organization, "10 facts on ageing and the life course," in *http://www.who.int/features/factfiles/ageing/ageing*$_f$*acts/en/, Oct.*2014.

[3] UNFPA (United Nations Population Fund), "Ageing in the twenty-first century: A celebration and a challenge," in *https://www.unfpa.org/public/home/publications/pid/11584*, Oct. 2014.

[4] D. C. KALLUR, "Human Localization and Activity Recognition using Distributed Motion Sensors," Master's thesis, Oklahoma State University, 2014.

[5] J. Secker, R. Hill, L. Villeneau, and S. Parkman, "Promoting independence: but promoting what and how?," *Ageing and Society*, vol. 23, no. 03, pp. 375–391, 2003.

[6] M. Alam, M. Reaz, and M. Ali, "A Review of Smart Homes—Past, Present, and Future," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1190–1203, 2012.

[7] M. Arndt and K. Berns, "Mobile Robots in Smart Environments: The Current Situation," *Autonomous Mobile Systems*, pp. 39–47, 2012.

[8] J. Broekens, M. Heerink, and H. Rosendal, "Assistive social robots in elderly care: a review," *Gerontechnology*, vol. 8, 2009.

[9] T. Breuer, G. R. Giorgana Macedo, R. Hartanto, N. Hochgeschwender, D. Holz, F. Hegger, Z. Jin, C. Müller, J. Paulus, M. Reckhaus, J. A. Álvarez Ruiz, P. G. Plöger, and G. K. Kraetzschmar, "Johnny: An Autonomous Service Robot for Domestic Environments," *Journal of Intelligent & Robotic Systems*, vol. 66, pp. 245–272, July 2011.

[10] H.-M. Gross, C. Schroeter, S. Mueller, M. Volkhardt, E. Einhorn, a. Bley, T. Langner, M. Merten, C. Huijnen, H. V. D. Heuvel, and a. V. Berlo, "Further progress towards a home robot companion for people with mild cognitive impairment," *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 637–644, Oct. 2012.

[11] R. Bogue, "Robots in healthcare," *Industrial Robot: An International Journal*, vol. 38, no. 3, pp. 218–223, 2011.

[12] M. Chen, Y. Ma, S. Ullah, W. Cai, and E. Song, "Rochas: robotics and cloud-assisted healthcare system for empty nester," in *Proceedings of the 8th International Conference on Body Area Networks*, pp. 217–220, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2013.

[13] C. Jayawardena, I. H. Kuo, U. Unger, a. Igic, R. Wong, C. I. Watson, R. Q. Stafford, E. Broadbent, P. Tiwari, J. Warren, J. Sohn, and B. a. MacDonald, "Deployment of a service robot to help older people," *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5990–5995, Oct. 2010.

[14] B. T. Horowitz, "Cybercare: Will robots help the elderly to live at home for longer?," *Scientific American*, 2010.

[15] G. D. Abowd, A. F. Bobick, I. A. Essa, E. D. Mynatt, and W. A. Rogers, “The aware home: A living laboratory for technologies for successful aging,” in *Proceedings of the AAAI-02 Workshop "Automation as Caregiver"*, pp. 1–7, 2002.

[16] University of Florida, “Gato-tech smart house,” in *http://www.icta.ufl.edu/gatortech/*, Sep. 2014.

[17] Pioneer P3-DX, “http://www.mobilerobots.com/researchrobots/pioneerp3dx.aspx,” Sep. 2014.

[18] iRobot Create, “http://store.irobot.com/shop/index.jsp?categoryid=3311368,” Sep. 2014.

[19] NaturalPoint, Inc, “Motion Capture Systems - OptiTrack,” in *https://www.naturalpoint.com/optitrack/*, Sep. 2014.

[20] H. M. Do, C. Mouser, M. Liu, and W. Sheng, “Human-robot collaboration in a mobile visual sensor network,” in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 2203–2208, May 2014.

[21] Willowgarage Inc., “ROS wiki,” in *http://www.ros.org/wiki/*, Sep. 2014.

[22] CompuLab, “Fit-PC2,” in *http://www.fit-pc.com/web/*, Sep. 2014.

[23] ASUS, “Xion Pro Live,” in *http://www.asus.com/*, Sep. 2014.

[24] Hokuyo, “Hokuyo laser,” in *http://www.hokuyo-aut.jp/*, Sep. 2014.

[25] W. B. G. Grisetti, C. Stachniss, “Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters,” *Robotics, IEEE Transactions on*, vol. 23, no. 1, pp. 34–46, 2007.

[26] D. Fox, "Adapting the Sample Size in Particle Filters Through KLD- Sampling," *International Journal of Robotics Research*, vol. 22, 2003.

[27] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo localization for mobile robots," *Artificial Intelligence*, vol. 128, pp. 99–141, May 2001.

[28] Motorola, "Motorola Xoom," in *Webpage: http://www.motorola.com/Consumers/US-EN/Consumer-Product-and-Services/Tablets/ci.MOTOROLA-XOOM-with-WiFi-US-EN.altanchor*, Sep. 2014.

[29] Google, "AndroidSDK," in *Webpage: http://developer.android.com/sdk/index.html*, Sep. 2014.

[30] Google, "Rojava core," in *Webpage: http://code.google.com/p/RosJava/*.

[31] H. Do, C. Mouser, Y. Gu, W. Sheng, S. Honarvar, and T. Chen, "An open platform telepresence robot with natural human interface," in *Cyber Technology in Automation, Control and Intelligent Systems (CYBER), 2013 IEEE 3rd Annual International Conference on*, pp. 81–86, May 2013.

[32] M. Desai, K. M. Tsui, H. A. Yanco and C. Uhlik, "Essential Features of Telepresence Robots," in *Technologies for Practical Robot Applications (TePRA), 2011 IEEE Conference on*, pp. 12–20, 2011.

[33] G. Walker, *Telepresence: the future of telephony.* Springer, 2000.

[34] R. Stiefelhagen, C. Fugen, R. Gieselmann, H. Holzapfel, K. Nickel, and A. Waibel, "Natural human-robot interaction using speech, head pose and gestures," in *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 3, pp. 2422–2427 vol.3, Sept 2004.

[35] D. Rodriguez-Losada, F. Matia, R. Galan, M. Hernando, J. M. Montero, and J. M. Lucas, “Urbano, an interactive mobile tour-guide robot,” *Advances in Service Robotics*, pp. 229–252, 2008.

[36] V. Alvarez-Santos, R. Iglesias, X. M. Pardo, C. V. Regueiro, and A. Canedo-Rodriguez, “Gesture-based interaction with voice feedback for a tour-guide robot,” *Journal of Visual Communication and Image Representation*, vol. 25, no. 2, pp. 499–509, 2014.

[37] K. Bong Keun Kim, Miyazaki, M., Ohba, K., Hirai, S., Tanie, “Services Based Robot Control Platform for Ubiquitous Functions,” in *Proceedings of IEEE Robotics and Automation*, pp. 691– 696, 2005.

[38] R. Arumugam, V. R. Enti, K. Baskaran, and a. S. Kumar, “DAvinCi: A cloud computing framework for service robots,” in *2010 IEEE International Conference on Robotics and Automation*, pp. 3084–3089, Ieee, May 2010.

[39] Google, “RoboEarth,” in *http://www.roboearth.org/*, Nov. 2014.

[40] Google, “Google I/O 2011,” in *http://www.google.com/events/io/2011/sessions/cloud-robotics.html*, Nov. 2014.

[41] V. I. Pavlovic, R. Sharma, and T. S. Huang, “Visual interpretation of hand gestures for human-computer interaction: A review,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 7, pp. 677–695, 1997.

[42] D. O. T. Jr, “Hidden Markov Models for gesture recognition,” Master’s thesis, Stanford University, 1995.

[43] Y. Gu, H. Do, J. Evert, and W. Sheng, “Human Gesture Recognition through a Kinect Sensor,” in *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1379–1384, 2012.

[44] M. Gales and S. Young, “The application of hidden markov models in speech recognition,” *Foundations and Trends in Signal Processing*, vol. 1, no. 3, pp. 195–304, 2008.

[45] V. Zue, J. Glass, M. Phillips, and S. Seneff, “The mit summit speech recognition system: A progress report,” in *Proceedings of the workshop on Speech and Natural Language*, pp. 179–189, Association for Computational Linguistics, 1989.

[46] H. M. Do, C. Mouser, and W. Sheng, “Building a telepresence robot based on an open-source robot operating system and android,” in *Third Conference on Theoretical and Applied Computer Science (TACS 2012)*, February 2012.

[47] D. Huggins-Daines, M. Kumar, A. Chan, A. Black, M. Ravishankar, and A. Rudnicky, “Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices,” in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, 2006.

[48] A. Vincent and D. Ross, “Learning Style Awareness: A Basis For Developing Teaching and Learning Strategies,” *Journal of Research on Computing in Education*, vol. 33, no. 5, pp. 1–11, 2001.

[49] A. Lee and T. Kawahara, “Recent development of open-source speech recognition engine julius,” in *Proceedings: APSIPA ASC 2009: Asia-Pacific Signal and Information Processing Association, 2009 Annual Summit and Conference*, pp. 131–137, 2009.

[50] K. Nakadai, T. Takahashi, H. G. Okuno, H. Nakajima, Y. Hasegawa, and H. Tsujino, “Design and Implementation of Robot Audition System ’HARK’ Open Source Software for Listening to Three Simultaneous Speakers,” *Advanced Robotics*, vol. 24, pp. 739–761, Jan. 2010.

[51] R. Schmidt, “Multiple emitter location and signal parameter estimation,” *Antennas and Propagation, IEEE Transactions on*, vol. 34, pp. 276–280, Mar 1986.

[52] H. G. Okuno, K. Nakadai, T. Takahashi, R. Takeda, K. Nakamura, T. Mizumoto, T. Yoshida, A. Lim, T. Otsuka, K. Nagira, T. Itohara, and Y. Bando, “HARK Document Version 2.1.0,” tech. rep., Kyoto University, 2014.

[53] K. Nakamura, K. Nakadai, F. Asano, Y. Hasegawa, and H. Tsujino, “Intelligent sound source localization for dynamic environments,” *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 664–669, Oct. 2009.

[54] F. Asano, M. Goto, K. Itou, and H. Asoh, “Real-time sound source localization and separation system and its application to automatic speech recognition.,” in *INTERSPEECH*, pp. 1013–1016, 2001.

[55] H. Nakajima and K. Nakadai, “Blind source separation with parameter-free adaptive step-size method for robot audition,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, pp. 1476–1485, Aug. 2010.

[56] J. A. Suykens and J. Vandewalle, “Least squares support vector machine classifiers,” *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[57] W. S. Yue-Xian Zou, W. Q. Zheng and H. Liu, “Improved Voice Activity Detection Based on Support Vector Machine with High Separable Speech Feature Vectors,” in *19 th International Conference on Digital Signal Processing*, 2014.

[58] M. Slaney, “Auditory toolbox,” *Interval Research Corporation, Tech. Rep*, vol. 10, 1998.

[59] Y. Sasaki, S. Kagami, and H. Mizoguchi, “Multiple Sound Source Mapping for a Mobile Robot by Self-motion Triangulation,” *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 380–385, Oct. 2006.

[60] D. Huggins-Daines, M. Kumar, A. Chan, A. W. Black, M. Ravishankar, and A. I. Rudnicky, “Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices,” in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 1, pp. I–I, IEEE, 2006.

[61] P. Hu, Y. and Loizou, “Subjective evaluation and comparison of speech enhancement algorithms,” *Speech Communication*, vol. 49, pp. 588–601, 2007.

VITA

Ma Manh Do

Candidate for the Degree of

Master of Science

Thesis: DEVELOPING A HOME SERVICE ROBOT PLATFORM FOR SMART HOMES

Major Field: Electrical Engineering

Biographical:

Personal Data: Born in Hai Duong, Vietnam on February 08, 1976.

Education:

Bachelor of Science from Hanoi University of Science and Technology, Hanoi, Vietnam, 1999, in Electronics and Telecommunications.

Completed the requirements for the degree of Master of Arts/Science with a major in Electrical Engineering in Oklahoma State University in December 2014.

Experience:

Lecturer in Posts and Telecommunications Institute of Technology from 1999 to 2011.