Shared User-Computer Control of a Robotic Wheelchair System

by

Holly A. Yanco

B.A., Wellesley College (1991)
M.S., Massachusetts Institute of Technology (1994)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2000

Signature of Author . . . .
Department of Electrical Engineering and Computer Science
September 5, 2000

Certified by . . . .
Rodney A. Brooks
Professor
Thesis Supervisor

Accepted by . . . .
Arthur C. Smith
Chairman, Departmental Committee on Graduate Students

Shared User-Computer Control of a Robotic Wheelchair System

by

Holly A. Yanco

Submitted to the Department of Electrical Engineering and Computer Science
on September 5, 2000, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

An assistive robotic wheelchair system should allow its user to travel more efficiently and with greater ease. The robotic wheelchair system described in this thesis, *Wheelesley*, performs semi-autonomous navigation for its user, taking high-level directional commands and performing the low-level navigation required to avoid obstacles and stay on a safe path. The system consists of a standard electric wheelchair with an on-board computer for motor control, a vision system running on an off-the-shelf notebook computer, sensors, and a graphical user interface running on a tray mounted notebook computer.

Most other research on robotic wheelchairs only addresses indoor navigation. The Wheelesley system can travel both indoors and outdoors using specialized navigation modes; there is a control algorithm for indoor navigation and a control algorithm for outdoor navigation. User tests have been conducted for both control modes to compare robotic assisted control against manual control. Robotic assisted control requires 71% less effort for indoor navigation and 88% less effort for outdoor navigation. In addition, the total time needed to travel between two points is reduced since less time is spent waiting to scan to the desired commands.

The system switches automatically between navigation modes through the use of a novel indoor/outdoor detector. The detector is comprised of an ultrasonic transducer, three light-to-voltage optical sensors and a thermistor. A decision tree learned by C4.5 using data collected in a variety of indoor and outdoor conditions classified a test set correctly 98.3% of the time.

The system can be easily customized for the access method(s) required by each user. This thesis describes customization of the user interface for two different access methods: eye tracking, an uncommon access method for a wheelchair, and single switch scanning, which is considered the driving method of last resort on standard powered wheelchairs. The wheelchair system and its interface was evaluated by a group of physical therapists.

Thesis Supervisor: Rodney A. Brooks
Title: Professor

Shared User-Computer Control of a Robotic Wheelchair System

by

Holly A. Yanco

Submitted to the Department of Electrical Engineering and Computer Science
on September 5, 2000, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

An assistive robotic wheelchair system should allow its user to travel more efficiently and with greater ease. The robotic wheelchair system described in this thesis, *Wheelesley*, performs semi-autonomous navigation for its user, taking high-level directional commands and performing the low-level navigation required to avoid obstacles and stay on a safe path. The system consists of a standard electric wheelchair with an on-board computer for motor control, a vision system running on an off-the-shelf notebook computer, sensors, and a graphical user interface running on a tray mounted notebook computer.

Most other research on robotic wheelchairs only addresses indoor navigation. The Wheelesley system can travel both indoors and outdoors using specialized navigation modes; there is a control algorithm for indoor navigation and a control algorithm for outdoor navigation. User tests have been conducted for both control modes to compare robotic assisted control against manual control. Robotic assisted control requires 71% less effort for indoor navigation and 88% less effort for outdoor navigation. In addition, the total time needed to travel between two points is reduced since less time is spent waiting to scan to the desired commands.

The system switches automatically between navigation modes through the use of a novel indoor/outdoor detector. The detector is comprised of an ultrasonic transducer, three light-to-voltage optical sensors and a thermistor. A decision tree learned by C4.5 using data collected in a variety of indoor and outdoor conditions classified a test set correctly 98.3% of the time.

The system can be easily customized for the access method(s) required by each user. This thesis describes customization of the user interface for two different access methods: eye tracking, an uncommon access method for a wheelchair, and single switch scanning, which is considered the driving method of last resort on standard

powered wheelchairs. The wheelchair system and its interface was evaluated by a group of physical therapists.

Acknowledgments

My thesis committee encouraged me to pursue directions that improved this research. I thank my advisor, Rod Brooks, for all of his support and guidance and for making so much time for me in his busy schedule. Eric Grimson has been a mentor to me since my first day of graduate school (don't ask when that was). Roz Picard has questioned me repeatedly on many aspects of the thesis, which helped me to express my ideas more clearly. All three of them contributed greatly to this research.

Customization of the user interface for eye tracking and single switch scanning was the result of a fruitful collaboration with Jim Gips of Boston College. The indoor user tests were also the result of this collaboration.

Without the support of many people, including those listed below, this thesis would not exist. Thanks to Dave Baggett, Kim Beck, Steve Burns, Jill Fekete, Polina Golland, Dave Miller, Radhika Nagpal, Nancy Ross, Marika Santagata, Randy Sargent, Brian Scassellati, Gideon Stein, my students at Wellesley (especially Alison Peacock and Harriet Wintermute), Eduardo Torres-Jara, Carlin Vieri, Mike Wessler, Ron Wiken, and the people who trusted me enough to participate in the user tests. Special thanks to Ben's grandparents for many months of babysitting.

Finally, I'd like to dedicate this thesis to Oded, who was there at the beginning of this project, and Ben, who joined us near its end.

This research was funded by the Office of Naval Research under contract number N00014-95-1-0600, the National Science Foundation under grant number CDA-

9505200 and a faculty research grant from Wellesley College.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In 1993, approximately 1.5 million people in the United States used wheelchairs and 6.6% of this population used powered wheelchairs [Jones and Sanford, 1996]. Another survey estimates that there are 1.5 to 2 million full-time wheelchair users in the United States [Cooper, 1999]. A study in the United Kingdom found that 5.1% of their sample group of wheelchair users were using powered wheelchairs [Kettle *et al.*, 1992]. The top three US manufacturers produce 2,100 wheelchairs each week[1] [Ashmore, 2000].

The number of wheelchair users is growing faster than the general population [Cooper, 1999]. Reduced infant mortality has resulted in an increase in congenital disabilities; for example, the incidence of cerebral palsy in the United States was 1.9 per 1000 in 1960 and had increased to 2.3 per 1000 in 1986. Additionally, there are fewer deaths from trauma, but the incidence rate remains the same. Finally, people are living longer, resulting in increases in age related disabilities caused by arthritis,

[1] Invacare, the US's largest wheelchair manufacturer has $1B/yr in sales and is a Fortune 500 company.

hip fractures, stroke (and a decline in stroke mortality), and diabetes mellitis which can lead to lower extremity amputation [Jones and Sanford, 1996].

A study in the United Kingdom with 215 people[2] who required mobility assistance included 171 people who used wheelchairs. In their sample, 35.6% reported using both a manual and a powered wheelchair,[3] but only 1.8% used a powered wheelchair alone. Of the larger group of powered wheelchair users, 6.3% were using an alternative access method (chin, arm or tongue control) instead of the standard joystick. This study also found that 16 of their subjects (7.4% of the total group) needed special controls to drive a powered wheelchair: 10 people had a powered wheelchair but could no longer use it or had great difficulty using it, 1 person used to have a powered wheelchair but their condition deteriorated, and 5 people had never had a powered wheelchair [Lachmann *et al.*, 1993].

While the Lachmann study finds that 7.4% of their study group has difficulty using a powered wheelchair, their study group corresponds to less than 1 per 1000 in the general population. Cooper estimates that there are 1.5 to 2.0 million wheelchair users in the United States, approximating 7 per 1000 in the general population. To increase the number of people in the Lachmann study to that of the general population, we will dilute their population with wheelchair users who we will assume have no difficulty using a wheelchair (either powered or manual). This results in an estimate that 1% of people requiring mobility assistance have difficulty driving a powered wheelchair,

[2] The response rate of 215 people in this study corresponds to slightly less than 1 per 1000 of the general population living in the study's geographical area. This rate is much lower than the approximate rate of 7 per 1000 in the United States [Jones and Sanford, 1996, Cooper, 1999].

[3] Many of the subjects lived in a group situation where a wheelchair would be shared.

which is likely to be an underassessment. With a population in the United States of 1.5 to 2.0 million wheelchair users, we then estimate that there are at least 15,000 to 20,000 people unable to drive a powered wheelchair who must rely upon a caregiver to move them around the world.

Borgolte [Borgolte *et al.*, 1998] states that "[a] conservative estimate indicates that over 2 million people with severe special needs within the European Community could benefit from an individually configurable intelligent wheelchair."

The goal of the research described in this thesis is to provide people unable to drive a standard powered wheelchair with a mobility device that does not require the assistance of a caregiver. These users may or may not have driven a powered wheelchair in the past. The wheelchair system developed, called *Wheelesley*, uses acoustic and vision based navigation to provide navigation assistance (e.g., obstacle avoidance and path centering). It also includes a graphical user interface that can be customized for the user's abilities and required access method.

1.1 Problem statement

The target community for a robotic wheelchair system consists of people who are unable to drive a powered wheelchair using a standard joystick. This group includes people with cerebral palsy, stroke patients who omit stimuli from one side[4] and quadriplegics. The users vary in ability and access methods used to drive the

[4] Cooper [Cooper, 1999] states that some type of collision avoidance could be useful for stroke patients who omit stimuli from one side, but it is less necessary for stroke patients with no evidence of unilateral neglect since they have fewer collisions.

wheelchair. Some people can move a joystick, but are unable to make fine movement corrections using the joystick. Other people are able to click one or more switches using their head or other body part. Some of our potential users are unable to control a powered wheelchair with any of the available access devices and must rely upon a caregiver to move them through the world. The wide variety of user abilities in our target community requires that the system be adaptable for many types of access devices.

While members of the target community have different abilities, we assume that all users will have some common qualities. We expect that any potential user can give high-level commands to the wheelchair through some access method and a customized user interface. We assume that the user of the wheelchair is able to see, although later versions of the system may be developed for the visually impaired. We also assume that a potential user has the cognitive ability to learn to how to operate the system and to continue to successfully operate the system once out of a training environment.

The system must be able to navigate in indoor and outdoor environments. A survey of powered and manual wheelchair users found that 56.6% used their wheelchair only outside, 33.3% used their wheelchair both inside and outside, and 10% used their wheelchair only inside [Kettle *et al.*, 1992]. New locations must be navigated as well as known locations; the 90% of subjects using their chair outdoors are likely to encounter new locations. The system can not make driving more difficult by adding requirements for selecting options for the system; driving a robotic system should be be easier for its user than driving a standard powered wheelchair.

1.2 Approach

This research developed a robotic wheelchair system that provides navigation assistance in indoor and outdoor environments, allowing its user to drive more easily and efficiently. A robotic wheelchair should be a semi-autonomous system, taking advantage of the intelligence of the chair's user by allowing the user to plan the general route while taking over lower level control such as obstacle avoidance and centering on a path.

There are two basic requirements for any robotic wheelchair system. First and foremost, a robotic wheelchair must navigate safely for long periods of time. Any failures must be graceful to prevent harm from coming to the user. Second, in order for such a system to be useful, it must interact effectively with the user. Outside of these two requirements, desirable features may include outdoor as well as indoor navigation, automatic mode selection based upon the current environment and task to reduce the cognitive overhead of the user, and easily adaptable user interfaces.

The Wheelesley system takes over low-level navigation control for the user, allowing the user to give higher level directional commands such as "forward" or "right." Most people take low-level control for granted when walking or driving. For example, when walking down a busy corridor, people are not usually aware of all of the small changes they make to avoid people and other obstacles. However, for users in our target community, low-level control requires just as much effort as high-level control. For example, it may be easy for a disabled person to gesture in the direction of a doorway, but difficult for that person to do the fine navigation required to direct the wheelchair through a doorway that is barely wider than the wheelchair. The robot carries out each command from the user by using its sensors and control code to navigate safely.

The Wheelesley robotic wheelchair system is a general purpose navigation assistant in environments that are accessible for the disabled (e.g., ramps and doorways of sufficient width to allow a wheelchair to pass). The system uses reactive navigation, meaning that the wheelchair will modify its path on the fly as it encounters obstacles. One of the advantages of this strategy is that users are not limited to one particular location by the need for maps or environment modifications.

The focus on a wheelchair introduces a new view to robotics research. Instead of concentrating on the difficult problem of autonomous navigation in many different and possibly previously unseen environments, the semi-autonomous navigation system allows the user to help the robot when it can not interpret a situation. The need for traveling indoors and outdoors resulted in a system that can switch automatically between the two environments. While some robots can work indoors or outdoors, no other system is specifically designed to interact in both environments by using different control modes. A novel indoor/outdoor detector was designed and implemented for this application. The detector measures ultraviolet light, infrared light, unfiltered light, temperature and distance to the nearest overhanging object (such as a ceiling or tree branch) using an ultrasonic ranging sensor. Data collected with the detector was used to learn a decision tree for determining the current environment.

The inclusion of an easily customizable interface makes the system's use with disabled users in a real world situation more possible. Almost all prior work on

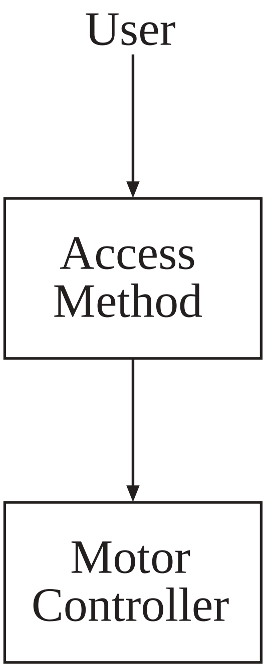


Figure 1-1: Block diagram of a standard powered wheelchair. The wheelchair's user gives movement commands to the wheelchair through an access method (e.g., a joystick, switch array, or a sip-and-puff system). The signals from the access method are translated into motor commands that are passed to the motor controller.

robotic wheelchair systems has not addressed the issue of customizing the system for a client. A person who is unable to drive with standard joystick control will mostly likely not be aided by a robotic wheelchair whose input method is a joystick.

Figure 1-1 shows a block diagram for a standard powered wheelchair. The user interacts with the wheelchair using an access method such as a joystick or sip-and-puff system. The commands given through the access method are passed to the wheelchair controller as motor commands consisting of a direction component and a speed component.[5]

Figure 1-2 shows a block diagram of the Wheelesley system. The user interacts

[5] Under some circumstances, the speed may be preset by the wheelchair provider and not determined by the user's control of the access method.

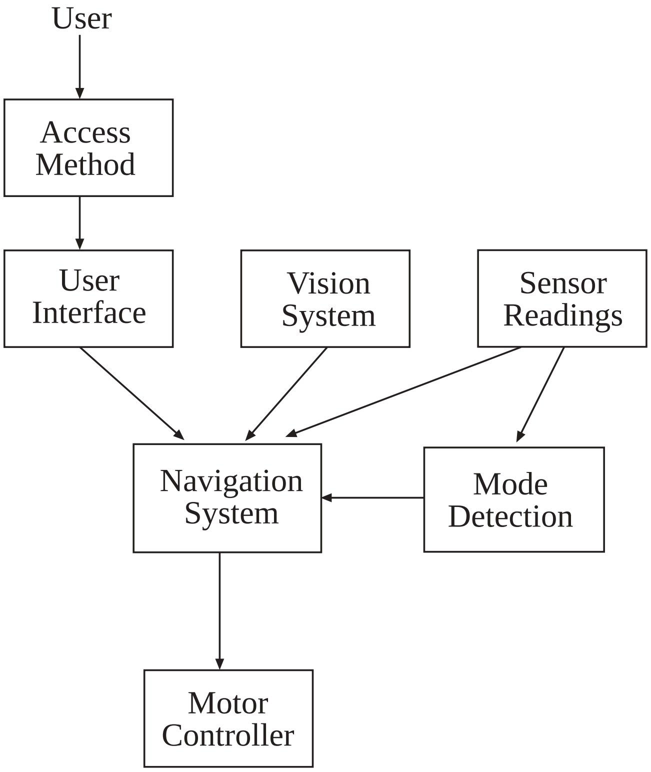


Figure 1-2: Block diagram of the Wheelesley system. The user gives commands to the user interface using an access method. The command from the user interface is passed to the navigation system along with sensor readings and information from the vision system. Sensor readings are also used for mode detection, which determines the proper navigation code to use for the current environment. The navigation system computes the correct motor commands and passes them to the motor control.

with the system using an access method to give movement commands via the user interface. Commands from the user interface are passed via a serial line to the robot's on-board processor which is running the navigation system. In addition to this directional command, the navigation system needs information about the state of the world in order to travel safely; this information is gathered using sensors and a vision system. A mode detector is used to determine whether the system should be executing navigation routines for indoor or outdoor navigation. Finally, the selected navigation routine uses the information gathered by the sensors about the state of the world to send the correct motor commands to the wheelchair controller.

1.3 Research contributions

This research resulted in a complete robotic wheelchair system, *Wheelesley.* Research contributions include the following:

- The system works indoors and outdoors, automatically switching its control algorithm for the current environment. The use of different control mechanisms for different environments allows the system to exploit the current environment.
- User tests of the indoor navigation system showed an improvement over standard control of 71% for user effort measured by the number of commands issued to navigate a test course. Assisted navigation also resulted in a 25% improvement in the amount of time required to navigate the course.
- The outdoor navigation system reduced user effort by 88% and reduced the time needed to navigate the test course by 27%.
- The development of a indoor/outdoor mode detector to allow the chair to select the proper mode for the current environment. Using a light-to-voltage sensor with an ultraviolet bandpass filter, an ultrasonic transducer pointing up, and a temperature sensor, the decision tree that was learned was able to classify the test set correctly 98.3% of the time.
- A wheelchair system must be customized for each user. The Wheelesley system has a user interface that can be quickly and easily customized for users and

their required access methods. The interface was customized in less than one hour for two different access methods: eye tracking and single switch scanning.

1.4 Outline of the thesis

Chapter 2 presents a brief overview of assistive technology with a detailed discussion of the related work in robotic wheelchairs. In Chapter 3, the hardware used in this research is described. This includes the robotic wheelchair, mode detector and vision system. Chapter 4 discusses general navigation system issues. The indoor navigation system and indoor user tests are discussed in Chapter 5 and the outdoor navigation system, including user tests, is presented in Chapter 6. The development of the mode detector is presented in Chapter 7. The design of the user interface and its use with access methods are discussed in Chapter 8. Finally, we present our conclusions and suggestions for further research in Chapter 9.

Chapter 2

Assistive Robotics

> *A person's control of his/her personal space is an important component of human dignity and the quality of life* [Engelhardt *et al.*, 1983].

Robotics technology has been applied to and developed for many different applications to assist people with disabilities. This broad field of robotics is usually called *assistive robotics* or *rehabilitation robotics.* Robots have been built to assist people with personal care, to provide vocational assistance, to retrieve items and to provide safe travel.

Several earlier overviews of the field have been written. Dario, Guglielmelli and Allotta [Dario *et al.*, 1994] discuss the use of robotics in medicine, covering topics from robot guided surgery to robotic arms to mobile robots for delivering items in hospitals. Dallaway, Jackson and Timmers [Dallaway *et al.*, 1995] present the state of research in Europe. Harwin, Rahman and Foulds [Harwin *et al.*, 1995] review rehabilitation robotics with an emphasis on systems developed in North America.

2.1 Manipulation assistance

Robotic arms fitted with some type of gripper can be used to help people eat, assist with personal hygiene, fetch items in a home or office environment, and open doorknobs. The arms can be mounted on wheelchairs, attached to mobile robots, on a mobile base, or fixed to one location as part of a workstation. An overview of rehabilitation research investigating robotic arms and systems can be found in [Mahoney, 1997].

Robotic arms are mounted on wheelchairs to provide a person with assistance everywhere they travel. Tasks to be accomplished with the arm include eating and drinking, personal hygiene, opening doors, pushing elevator call buttons, and picking up items. Arms mounted on wheelchairs must not interfere with normal use of the wheelchair by increasing its size too much or causing the chair's balance to become unstable. The Manus arm [Kwee *et al.*, 1989, Rosier *et al.*, 1991] is a five degree of freedom arm on rotating and telescoping base unit that is now available commercially. The Wessex robot [Hillman *et al.*, 1999, Hagan *et al.*, 1997] is a wheelchair mounted arm that is currently under development. This six degree of freedom arm is mounted at the rear of the wheelchair in a fixed position.

While arms mounted on wheelchairs usually require that the user be seated in the wheelchair in order to use the arm, robotic arms mounted on mobile robots can provide assistance away from the user as well as in the user's presence. WALKY [Bolmsjö *et al.*, 1995] is a mobile robot with an arm designed to assist its user in laboratory environments to conduct microscope work, blood group determination and culture analysis. It is designed to work with its user's own input device(s). The MoVAR project [Van der Loos, 1995] also integrated a robotic arm and mobile robot base for vocational assistance. MOVAID [Dallaway *et al.*, 1995] was designed for home use, to provide assistance with food preparation and house cleaning.

Handy 1 [Topping, 1999] is mounted to a (non-robotic) wheeled base. The robotic arm was designed to assist its users with personal hygiene and eating. Different trays can be attached in front of the arm to allow specific tasks to be accomplished: a tray for eating and drinking, a tray for washing, shaving and teeth cleaning, and a tray for applying make-up. A tray for art called the Artbox is currently being prototyped to allow the robot to assist with recreation in addition to necessary care.

Alternatively, robotic arms may be fixed to one location as part of a workstation. Workstations can be used for vocational assistance, where its user can perform his work duties with the assistance of the arm and its supporting interface. Workstations can also be used for eating, reading and personal hygiene.

The RAID workstation [Bolmsjö *et al.*, 1995, Dallaway *et al.*, 1995] was designed to work in a vocational environment. The robot arm was mounted on a track and could access materials like books and disks on a bookshelf and printouts from the printer. The end effector could also turn pages for the user. The EPI-RAID workstation [Dallaway *et al.*, 1995] is an extension of the work on the RAID workstation, requiring less complexity in programming tasks. The EPI-RAID workstation would also be useful in a home environment, which has more variation than an office workspace.

The DeVAR workstation [Van der Loos, 1995] was also designed for vocational assistance. It included a robotic arm for manipulation, telephone control and environmental control. ProVAR [Wagner *et al.*, 1999] introduces a user interface that is easier to use.

The MUSIIC system [Kazi *et al.*, 1998] allows for telemanipulation of objects using speech and gesture. For example, a user can point to a straw and say "That's a straw" followed by "Insert the straw into that" while pointing at a cup. This system allows the environment to change and include new objects.

2.2 Electronic travel aids for the blind

Electronic travel aids for the blind take many forms. All attempt to provide the user with assistance to compensate for the user's blindness. Some take a passive role, suggesting a safe travel direction through sound and allowing the user to walk as he wishes. Some systems are more active, guiding their users on a path as the user holds on to the system's handle.

The guide dog robot MELDOG [Tachi and Komoriya, 1991, Tachi *et al.*, 1985] was developed to emulate the assistance that a guide dog provides to a blind person. The robot was built to execute the commands of the user while providing intelligent disobedience if the command would cause the user to be put into harm's way. In initial prototypes, the robot required painted lines as landmarks, but later development utilized natural landmarks. Ultrasonic sensors were used to detect these natural landmarks as well as to provide obstacle avoidance. The robot communicated with its user through electrodes placed on the user's skin. This electrocutaneous communication was determine to be preferable to an audible warning to allow the person to use his hearing to help guide his travel without noise from the system.

HITOMI [Mori *et al.*, 1995, Mori *et al.*, 1998] was designed as a travel aid for people who lost their sight later in life and have trouble remembering routes. This system uses a powered wheelchair as its base. The user walks behind the wheelchair, holding on to its handles. The system uses vision, ultrasonic sensors, tactile sensors and GPS to guide its user in outdoor environments. HITOMI acts like a guide dog, taking its user safely across streets and down sidewalks.

PAM-AID [Lacey *et al.*, 1998, Lacey, 1999] was designed as a mobility aid for the elderly blind who need support while walking. In addition to providing physical support, the robot also provides obstacle avoidance using ultrasonic sensors, infrared detectors and bumpers. The user commands the robot using a joystick and a single switch. The robot was designed for indoor environments such as hospitals or nursing homes where its users would be bed-ridden without a caregiver or the robot for guidance and support.

Electronic travel aids do not need to be robotic. The Navbelt system [Shoval *et al.*, 1998], also meant for the blind, is comprised of a portable computer carried as a backpack, a belt with 8 ultrasonic sensors worn in a manner similar to a fanny pack, and stereo headphones. It uses the same algorithms for sonar firing and direction computation as the NavChair system described below [Levine *et al.*, 1999]. The travel direction computed is converted to tones played in the user's headphones. The system was tested indoors and outdoors, but fails to detect steps, holes, edges of sidewalks and overhanging objects. Plans for future work include the addition of sonars to detect these missed objects.

2.3 Robotic wheelchairs

Research in the field of robotic wheelchairs seeks to address issues such as safe navigation, splitting control efficiently between the user and the robot, and creating systems that will be usable by the target population. The focus is not on improving the mechanical design of the standard powered wheelchair. Thus robotic wheelchairs are usually built with standard powered wheelchairs for their bases.[1] Field [1999] presents a literature review covering many aspects of powered mobility and Cooper [1999] discusses issues for engineering both powered and manual wheelchairs.

This section will begin with an overview of other research on robotic wheelchairs and will conclude with a discussion of this other work in relation to the work presented in this thesis.

2.3.1 Description of research on robotic wheelchairs

An early system provided collision avoidance [Scott and Munro, 1985]. The wheelchair was driven using a joystick and provided collision avoidance using three ultrasonic sensors (one pointing forward, one to the left and one to the right). The chair would slow down if an obstacle was less than one foot away on either side or less than six feet away in the front. The chair would hit an obstacle at a maximum speed of 1/4 foot per second, allowing a user to pull up to a desk.

The Ultrasonic Head Controlled Wheelchair [Jaffe, 1981, Jaffe, 1983, Jaffe *et al.*, 1990] uses two ultrasonic sensors to measure forward/backward and left/right compo-

[1] A few projects have custom built bases (e.g., [Borgolte *et al.*, 1995] and [Tahboub and Asada, 1999].)

nents of the motion of a user's head. This information can be used to drive a powered wheelchair with no navigation assistance or can be used with the assistive mode that was developed. In assistive mode, using information from ultrasonic sensors, the chair slows or stops if an obstacle gets too close, can follow a slow moving person at a fixed distance, can follow walls, and provide a cruise control where the system maintains the last set speed. The user can switch between the system's modes by moving his head to the rear and then to one of the four quadrants.

The OMNI project [Borgolte *et al.*, 1995, Hoyer *et al.*, 1997, Bühler *et al.*, 1997, Borgolte *et al.*, 1998] uses a custom-designed omnidirectional wheelchair as its base. The chair can rotate around its center point, allowing it to move in tighter spaces than a standard powered wheelchair base. The system uses ultrasonic and infrared sensors to provide assisted control through obstacle avoidance, wall following and door passage. The project also includes a custom user interface that can be simplified for a row/column scanning mode.

Another custom designed omnidirectional wheelchair was built in the Mechanical Engineering department at MIT [Tahboub and Asada, 1999]. A behavior-based architecture for semi-autonomous control has been designed for the wheelchair, but has not yet been implemented. Plans call for the user to drive the system using a joystick. The system will use ultrasonic sensors to provide semi-autonomous control and autonomous control where the chair could follow a guide or wander randomly. Tahboub likens driving a semi-autonomous wheelchair to horseback riding. A horse will follow its rider's commands, but not if they put the horse in danger.

A system built by Connell [Connell and Viola, 1990] also follows a horseback riding analogy. The user would sit on a chair on a mobile robot base. A joystick was used for driving the system. A bank of toggle switches were used to turn on or off the ability of the robot to perform some tasks autonomously. These behaviors included obstacle avoidance, hallway traversal, turning at doors and following other moving objects. The robot was equipped with ultrasonic, infrared and bump sensors. The overhead involved with selecting behaviors and toggling switches would most likely be prohibitive for our target group.

An autonomous robotic wheelchair was developed at Arizona State University [Madarasz *et al.*, 1991]. The purpose of the system was to transport its user to a specified room in a building using a map of the environments and planning. It used a scanning Polaroid ultrasonic range finder for obstacle avoidance and a digital camera. The system used only a restricted amount of vision processing to locate and verify known objects such as room numbers, look at elevator lights and keep the wheelchair centered in the hallway.

TinMan II wheelchair [Miller and Slack, 1995] uses infrared, bump and ultrasonic sensors to provide semi-autonomous control. The hardware used in the Wheelesley project is similar to TinMan II, built by the same people at the KISS Institute for Practical Robotics. Our system uses the same hardware and sensor polling routines as their system. In their semi-autonomous control mode, the user can drive using a joystick with obstacle avoidance that will override the user's commands. In addition, the chair can be driven by pushing one button to turn while avoiding obstacles and by pushing another button to move forward while avoiding obstacles.

The goal of the TAO project [Gomi and Griffith, 1998] is to develop a robotic module for navigation that can be interfaced with standard wheelchairs. The behavior-based navigation module has been put on two different commercially available wheelchairs. The system uses computer vision and infrared sensors to navigate in its environment. It is primarily an indoor system, although it has been tested outdoors in limited situations such as a snowy sidewalk with 1 meter high walls of snow on either side. The TAO wheelchairs navigate in an autonomous mode, randomly wandering in an unstructured environment or performing landmark-based navigation. The user can override the robotic control by touching the joystick. In joystick mode, no assistance is provided.

Like the TAO Project, Alanen [Alanen *et al.*, 1996] has developed an add-on system with sensors to be placed on the wheelchair by its manufacturer. The system is semi-autonomous, providing obstacle avoidance using 12 ultrasonic sensors: 8 placed in the front of the wheelchair and 4 in the back. Like NavChair (described below), its collision avoidance is based upon the virtual force field (VFF) method [Borenstein and Koren, 1991]. The wheelchair slows down when there are obstacles, but still moves toward the obstacle, allowing the user to pull up to tables and open doors. The system was initially developed using a simulator [Ojala *et al.*, 1991].

The Hephaestus project [Simpson *et al.*, 1999] also aims to build a navigation assistant that can be added to any powered wheelchair. The system would be installed between the wheelchair's joystick and motor controller. The first prototype has been tried with one powered wheelchair base to date. The navigation assistance provided is based on the NavChair system described below. The system uses 16 ultrasonic sensors; 13 are mounted on the front and sides of a standard wheelchair tray and 3 are pointed backwards. Up to 24 switches can be placed anywhere on the wheelchair as bump sensors.

The NavChair [Simpson *et al.*, 1998] navigates in indoor office environments using a ring of sonar sensors mounted on the wheelchair tray. The height of the sensors prevents the system from being used outdoors since it can not detect curbs. The system has three operating modes: general obstacle avoidance, door passage and automatic wall following. The system can select a mode automatically based on the environment or the environment and location [Simpson and Levine, 1999]. (Automatic mode selection is discussed further in Chapter 7.) People who are unable to drive a standard powered wheelchair have been able to drive the NavChair using sensor guidance with either a joystick or voice commands as an access method.

Vocomotion [Amori, 1992] is another voice controlled wheelchair. The system provides no driving assistance. However, it will fill in non-specified information required to execute a user's command; for example, if the speed is not specified, it will continue to travel at the set speed.

Senario [Katevas *et al.*, 1997, Beattie and Bishop, 1998] can be operated in a semi-autonomous or fully autonomous mode. The system informs the user of risk and takes corrective measures. The user can override in semi-autonomous mode. The wheelchair will stop moving if an emergency situation is detected. The system uses 13 ultrasonic sensors, split into navigation sensors and protection sensors. Two encoders provide a rough orientation estimate. Two infrared range finders mounted at 192 cm (above the user's head) are also used for calculating positioning information. The user can command the system using voice control or the joystick.

The UMIDAM Project [Mazo *et al.*, 1995] developed a wheelchair navigation system using ultrasonic and infrared sensors. They include sensors pointing downward to detect stairs. The wheelchair can be commanded by voice in semi-autonomous or autonomous mode. The autonomous mode provides obstacle avoidance and/or wall following with the speed controlled by voice. In semi-autonomous mode, the voice commands are executed using the sensors to provide safe navigation (if the user has not elected to turn off the sensors). Face tracking can also be used to control the wheelchair [Bergasa *et al.*, 1999].

A deictic navigation system has been developed for shared control of a robotic wheelchair [Crisman and Cleary, 1998]. This system navigates relative to landmarks using a vision-based system. The user of the wheelchair tells the robot where to go by clicking on a landmark in the screen image from the robot's camera and by setting parameters for motion, where the target should be at the end of the motion, what the distance between the robot and the target at the end of the motion and the desired speed in a computer window. The robot then extracts the region around the mouse click to determine to which landmark the user wishes to travel. It then uses the parameters to plan and execute the route to the landmark. Deictic navigation can be very useful for a disabled person, but a complicated menu might be difficult to control with many of the standard access methods. However, it could be adapted for a scanning system, perhaps in a row-column scanning pattern. The long-term goal of the project is a "gopher" robot; the wheelchair would have an attached arm and could operate with or without a person in the wheelchair. The robotic arm and hand is currently being developed unattached to the wheelchair.

Wakaumi [Wakaumi *et al.*, 1992] developed a robotic wheelchair that drove along a magnetic ferrite marker lane. A magnetic lane is preferable to a painted line due to its ability to continue to work in the presence of dirt on the line. The chair travels along the path when commanded with two infrared sensors in front of the wheelchair for obstacle detection. This type of system is useful for a nursing home environment to allow people to drive around without the need for being pushed by a caregiver.

A wheelchair developed at Notre Dame [Yoder *et al.*, 1996] provides task-level supervisory control; the user can select the nominal speed, stop and select a new destination or stop and take over control. The system is taught "reference paths" during set up that are stored in memory. Visual cues from two cameras are used to correct errors that normally occur in dead reckoning systems. The system does not include obstacle avoidance. If a trash can is put in the wheelchair's path, the user needs to take over control to maneuver around the trash can and can then pass control back to the system. The philosophy is in direct opposition to the one used in the Wheelesley system; we believe that fine navigation control required to navigate around small obstacles is more difficult for our target group than traversing a known route.

PSUBOT [Perkowski and Stanton, 1991, Stanton *et al.*, 1991] was designed to navigate indoors between rooms of a known building. The system uses vision and ultrasonic sensors to navigate. Commands are given to the robot using voice recognition. The robot is taught where landmarks are in images in a learning mode and then navigates autonomously using this information.

Wang [Wang *et al.*, 1997] designed a wheelchair system that uses ceiling lights as landmarks to self-localize. The system performs autonomous navigation, taking commands of the form "travel from node x to node y." One camera and a slit laser are used to detect obstacles and a second camera is used to recognize landmarks.

The VAHM project [Bourhis and Agostini, 1998b, Bourhis and Agostini, 1998a] operates in an assisted manual mode and an automatic mode. In assisted manual mode, the system can provide obstacle avoidance and wall following. In automatic mode, the system performs globally planned paths with obstacle avoidance of non-modeled objects. Their philosophy is that the person supervises the robot in automatic mode, overriding robotic commands that are unwanted, and the robot supervises the person in assisted manual mode, overriding commands that put the user in danger. The project has developed a user interface for single switch scanning.

A system developed at Carnegie Mellon University [Radhakrishnan and Nourbakhsh, 1999] uses a vision system with a 360 degree field of view to localize on a topological map. The system currently works only in an indoor environment, but there are plans to implement the system for outdoor environments. The navigation system is running on a wheelchair platform built by KIPR [Miller and Slack, 1995], but no access considerations are included at this time.

The Intelligent Wheelchair Project [Gribble *et al.*, 1998] also uses a base built by KIPR [Miller and Slack, 1995], as does this research. The research on this system is addressing spatial knowledge representation and reasoning. The structure of the environment is learned over time through local observations. The system uses stereo color vision, in addition to ultrasonic and infrared sensors.

A system developed at Osaka University [Adachi *et al.*, 1998, Kuno *et al.*, 1999a, Kuno *et al.*, 1999b] controls the wheelchair's direction by observing the direction of the user's face, under the assumption that a person will look where he wants to go. They use ultrasonic sonars to choose the sensitivity of the program to a user's head movement. For close obstacles, they smooth over a large number of frames, making the system ignore small head movements since the user may look at the close obstacles. For long readings from the sonars, smoothing is done over a smaller number of frames to allow for finer control in open spaces.

The CALL Centre Smart Wheelchair [Nisbet *et al.*, 1996] is a configurable system that can be modified for each individual user. It uses a standard powered wheelchair base, but the joystick is removed and replaced with a "Smart Controller." Bumpers detect collisions; several behaviors can be selected to correct the bump for the user. A line following behavior can follow reflective tape on the floor. Several access methods can be selected. Several case studies are discussed where the system starts out doing most of the work through bump behaviors and line following, then users learn to take on more of the task themselves. In some cases, the users can end up driving a conventional powered wheelchair, when they could were not successful learning to drive it without assistance from this project.

2.3.2 Discussion

Travel restrictions due to limited access to new environments

Some of the previous research on robotic wheelchairs has resulted in systems that are restricted to a particular location or set of locations, either due to the use of environmental modifications or the requirement for knowledge of the environment. Wakaumi [Wakaumi *et al.*, 1992] requires an environmental modification in the form of a magnetic track painted on the floor. Systems like this are useful in settings like nursing homes and hospitals, but are not useful for people who need to travel in many different places.

Some systems require knowledge of the environment, which may be provided to the system or learned by the system. A topological map is used for navigation in [Radhakrishnan and Nourbakhsh, 1999]. Wang [Wang *et al.*, 1997] developed a system that uses ceiling lights as landmarks for navigation, requiring no environmental modifications; however, the system does require a map that includes these landmarks. The VAHM project requires maps in its automatic mode. The wheelchair system developed by Madarasz [Madarasz *et al.*, 1991] also uses a map of a single known building. Other projects train the wheelchair on certain paths that can be repeated later (e.g., [Stanton *et al.*, 1991] and [Yoder *et al.*, 1996]).

The Wheelesley system uses no maps. The goal of the project is to help a user navigate safely through known and unknown environments semi-autonomously, not to take over navigation in an autonomous or even a supervisory capacity.

Indoor and outdoor navigation

Many robotic wheelchairs were designed only for indoor environments. However, a survey of powered and manual wheelchair users found that 57% used their wheelchair only outside and 33% used their wheelchair both inside and outside [Kettle *et al.*, 1992]. Even considering that some of the target population may be institutionalized which may increase the number of people using their systems indoors only, a large number of users still need a system that will work outdoors.

Systems described above that work or plan to work in both indoor and outdoor environments are the TAO project (in limited outdoor situations to date), the Intelligent Wheelchair Project (research plans include outdoor environments), and the Carnegie Mellon University wheelchair (plans to develop outdoor navigation).

Our research developed assisted navigation for both indoor and outdoor environments.

Levels of autonomy

Robotic wheelchair systems can either have semi-autonomous control or autonomous control. Semi-autonomous, or shared, control is used for the Deictic wheelchair [Crisman and Cleary, 1998], the OMNI project [Borgolte *et al.*, 1995, Hoyer *et al.*, 1997, Bühler *et al.*, 1997, Borgolte *et al.*, 1998], UMIDAM [Mazo *et al.*, 1995], and this research. Autonomous, or supervisory, control is used for many systems. The TAO Project [Gomi and Griffith, 1998] uses random walk or landmark-based navigation. In Wang [Wang *et al.*, 1997], the chair plans a path between the current position and the desired end position. The wheelchair developed at Arizona State University [Madarasz *et al.*, 1991], travels from one room to another in a known building, as was PSUBOT [Perkowski and Stanton, 1991, Stanton *et al.*, 1991]. The UMIDAM Project [Mazo *et al.*, 1995] provides both an autonomous and a semi-autonomous mode.

We have selected semi-autonomous navigation to allow the user to participate in moving through the world. Semi-autonomous navigation does not require planning or maps, which allows our system to be used in new environments without a training period for acquiring a map.

Methods for interaction

Very few projects have addressed the need for a customizable user interface. The OMNI project [Bühler *et al.*, 1997] developed a user interface for use with a joystick that has been customized for row/column scanning with a single switch. The VAHM project [Bourhis and Pino, 1996, Bourhis and Agostini, 1998b] developed an interface for single switch scanning. The Wheelesley project has also developed a customizable user interface (see Chapter 8).

Many robotic wheelchairs include only a joystick interface (e.g., [Scott and Munro, 1985], [Yoder *et al.*, 1996], [Tahboub and Asada, 1999], and [Simpson *et al.*, 1999]) or a joystick with one or more buttons/switches (e.g., [Connell and Viola, 1990] and [Miller and Slack, 1995]). Some research projects have investigated using other access methods to make the systems useful to people unable to drive using a joystick. These alternative access methods include voice control (e.g., [Stanton *et al.*, 1991], [Amori, 1992], [Mazo *et al.*, 1995], [Simpson and Levine, 1997], and [Katevas *et al.*, 1997, Beattie and Bishop, 1998]), eye tracking (see Section 8.2.1), single switch scanning (e.g., [Bourhis and Pino, 1996], [Bühler *et al.*, 1997] and Section 8.2.2), ultrasonic head control (e.g., [Jaffe, 1981, Jaffe, 1983] and [Ford and Sheredos, 1995]), and face tracking (e.g., [Adachi *et al.*, 1998] and [Bergasa *et al.*, 1999]). A further discussion of this topic can be found in Section 8.1.

Chapter 3

Hardware

3.1 Robotic wheelchair

The robotic wheelchair used in this research (Figure 3-1) was built by the KISS Institute for Practical Robotics [Miller and Slack, 1995]. The base is a Vector Mobility powered wheelchair. The drive wheels are centered on either side of the base, allowing the chair to turn in place. There are two front casters and a rear caster with spring suspension. Power is supplied by two 12 volt wheelchair batteries. The wheelchair has a tray that slides over the arms of the wheelchair; a support spans from each side of the front of the tray to the wheelchair base for additional stability when driving.

The wheelchair is 65 cm (25.5 inches) wide and 127 cm (50 inches) long, measured with the rear caster extended fully behind the chair. The front of the tray is 76 cm (30 inches) from the ground, and the base of the seat is 51 cm (20 inches) above the ground.

For sensing the environment, the robot has 12 SUNX proximity sensors (infrared), 4 ultrasonic range sensors, 2 shaft (wheel) encoders and 2 Hall effect sensors. (See

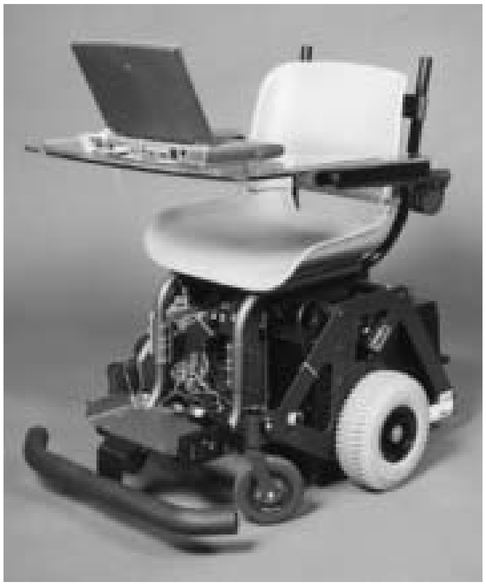


Figure 3-1: Wheelesley, the robotic wheelchair system.

[Everett, 1995] for descriptions of many sensor types, including the ones used in this work). The infrared and sonar sensors are placed around the perimeter of the wheelchair, with most pointing towards the front half of the chair. The Hall effect sensors are mounted on the front bumper of the wheelchair.[1] A diagram of the locations of the sensors is presented in Figure 3.1.

The robot has an onboard 68332 processor that is used to process sensor information and control the robot. The onboard processor is programmed using the ARC Development System written by Newton Research Labs. ARC is a multitasking system that provides a remote debugging environment for downloading and testing software on the host computer. A serial connection is used for communication between the host computer and onboard processor [Wright *et al.*, 1996]. The robot wheelchair

[1] In the early work on the system, the bumper was instrumented with two switches instead of two Hall effect sensors. These switches often were sheared off by bumper hits, which is why a non-contact sensing method was then used.

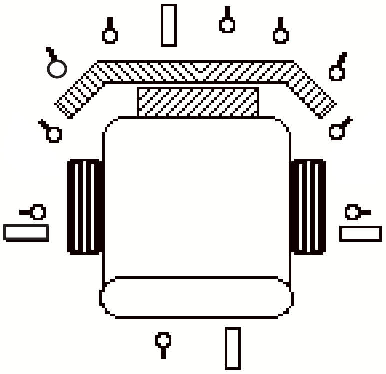


Figure 3-2: Placement of the sensors on the wheelchair. Infrared detectors are shown as circles with short lines pointing in the direction that the detectors are pointed. Ultrasonic sensors are shown as rectangles. For illustrative purposes, the sensors are drawn outside of the wheelchair's boundaries; most of the sensors are mounted on the perimeter of the wheelchair. The two Hall effect sensors are not shown in this figure; they are mounted on the front bumper, where the bumper is attached to the footrest.

came with software for polling the sensors and writing the values to variables.[2]

Additional sensors, including light-to-voltage optical sensors and a thermistor, have been added to the wheelchair system to detect whether the chair is in an indoor or outdoor environment in order to automatically switch between indoor and outdoor navigation modes. Data from these sensors is gathered with a National Instruments DAQ-Card 700 data acquisition card in one of the two PCMCIA slots of a Dell Computer Latitude notebook computer with a Pentium 166 MHz MMX processor. The software for the detector was written using Microsoft Visual C++ and National Instrument's Ni-Daq software. See Chapter 7 for a complete discussion of this custom built detector.

A Macintosh Powerbook 520c is used to run the robot's graphical user interface. The interface was designed to be easily customized for each individual user and his

[2] The robot also came with software for obstacle avoidance, but it was not used.

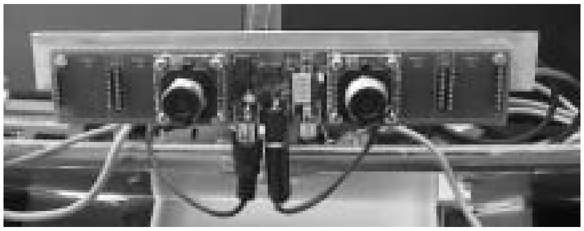


Figure 3-3: The STH-V1 Stereo Head from Videre Designs

access method, which is described in Chapter 8.

The Powerbook, Latitude notebook and the robot's processor are connected via a serial ring. Each computer has only one serial port. To connect the three devices, we built a connector that connected the send line to the receive line of the next computer in the ring. Messages were tagged with the name of the intended recipient and the computers used these to determine whether to capture the data or to pass it on.

3.2 Vision hardware

The SRI Small Vision System [Konolige, 1997, Konolige, 1998] is an implementation of the Stereo Engine algorithm. The SVS consists of library functions that can be called to perform stereo computation and other visual processing related to stereo computation. Any stereo images can be processed by the SVS; we are using the STH-V1 Stereo Head from Videre Designs.

The STH-V1 Stereo Head interlaces two video signals onto a single video stream so images can be grabbed with a single frame grabber. The head is 19 cm (7.5 inches) long by 3.2 cm (1.25 inches) wide. It is mounted on the front of the wheelchair's tray, pointing down at an approximately 40 degree angle. The baseline between the

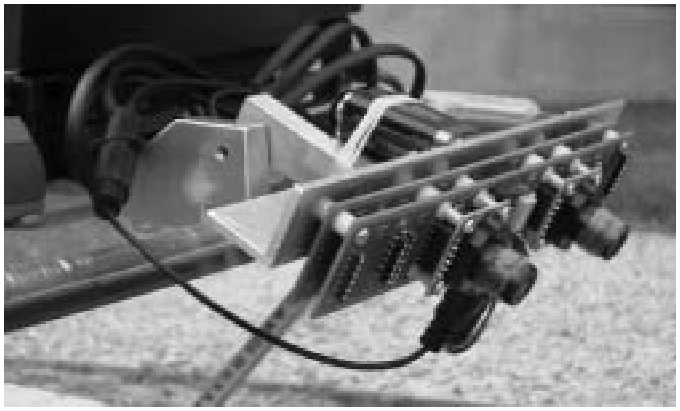


Figure 3-4: The Stereo Head mounted on the robotic wheelchair

cameras is 69 mm (2.7 inches). The focal length of each camera lens is 6.5 mm. See Figure 3-3 for a photograph of the stereo head and Figure 3-4 for a photograph of the stereo head's mount on the wheelchair.

The frame grabber used is a MRT VideoPort Professional, which is inserted in a PCMCIA slot of the Dell Latitude notebook computer with a Pentium 166 MHz MMX processor.

The stereo head and SVS can process 15 frames per second without the additional processing described in Section 6.2; this rate is due to the slow speed of the MRT VideoPort frame grabber. Without processing, the frame rate remains 15 frames per second. As a comparison, SVS can process at 26 frames per second on the same computer using a saved video stream. With the additional vision processing described in Section 6.2, the frame rate drops to 5.5 frames per second.

Chapter 4

General Navigation Principles

> *A basic goal in rehabilitation robotics is to be able to use the robot for a task that is done only once.* [Bolmsjö *et al.*, 1995]

This research developed a reactive navigation system for indoor and outdoor navigation to allow the wheelchair to successfully navigate in new environments. In this chapter, issues common to both indoor navigation and outdoor navigation are discussed first. The indoor navigation system and results of user tests are presented next. Finally, the outdoor navigation system and user tests validating it are discussed.

4.1 High-level and low-level control

There are two types of control when driving a wheelchair: low-level and high-level. Low-level control involves avoiding obstacles and keeping the chair centered in a hallway or on a sidewalk. High-level control involves directing the wheelchair to a desired location. For a powered wheelchair user who has good control of the joystick, these two types of control can be easily managed at the same time. The user can avoid obstacles on the path by moving the joystick a small amount to make the proper adjustment, while operating the chair to carry out the plan of arriving at the desired destination. This is analogous to driving a car; people make many small adjustments to keep the car in the proper lane and to avoid obstacles like potholes and other cars, while following their desired route to their destination.

When a powered wheelchair user does not have perfect control of a joystick or can not control a joystick at all, low-level control does not blend with high-level control as unconsciously. It is not possible to make small adjustments easily. For a user driving using an alternative access method (see Section 8.1), low-level control adjustments require as much effort as high-level commands. A robotic wheelchair can assist such a person by taking over low-level control, requiring the user to use the access method only to give high-level directional commands like *right* or *left* at an intersection.

In the Wheelesley system, the user gives the high-level commands (*forward*, *left*, *right*, *back*, and *stop*) through the graphical user interface (see Chapter 8). The system carries out the user's command using common sense constraints such as obstacle avoidance. The robot's low-level control acts to keep the wheelchair and its user safe; this low-level control is described in Chapters 5 (indoor navigation) and 6 (outdoor navigation). For example, if the user instructs the chair to go forward, the robot will carry out the command by taking over control until another command is issued. While executing the high-level "forward" command, the chair will prevent the user from running into walls or other obstacles. If the chair is completely blocked in front, it will stop and wait for another command from the user. If it is drifting to the right, it will correct itself and move to the left. This navigation method allows people

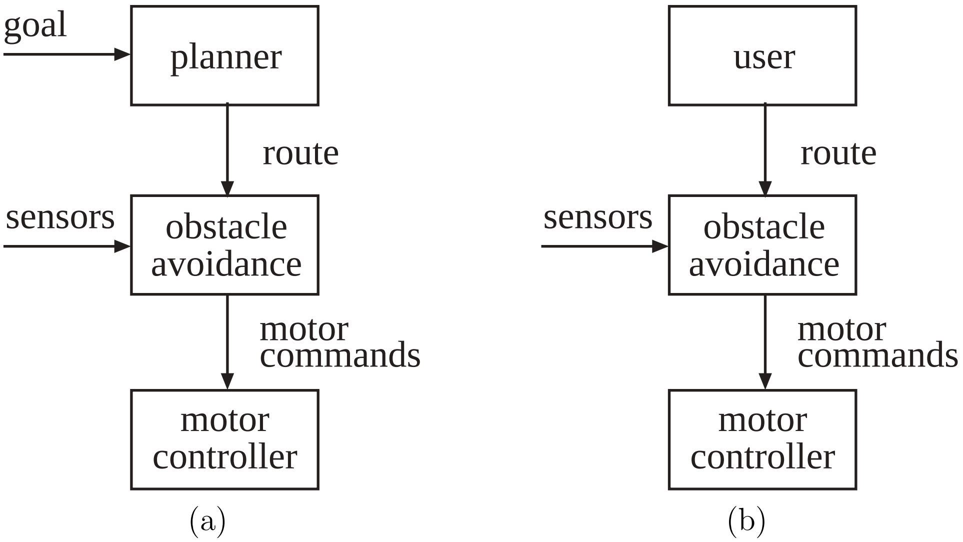


Figure 4-1: (a) Diagram of a typical planning-reaction architecture for a mobile robot. (b) Diagram of the architecture for the Wheelesley system. In the Wheelesley architecture, the user functions as the planning system while the robot takes over low-level control to maintain a safe path locally.

who have trouble with fine motor control but who have the ability to issue high-level commands to control a powered wheelchair.

This split between high-level and low-level control can be compared to the split between planning and reacting in some mobile robots. Figure 4-1a shows a diagram of a robot architecture using a planner. Figure 4-1b shows a diagram of the navigation architecture for Wheelesley. In the Wheelesley system, the user functions as the planner, determining how to get from point A to point B, while allowing the navigation system to reactively avoid obstacles and maintain a safe path.

4.2 Varying the navigation code for different environments

The Wheelesley system incorporates two navigation modes: indoor and outdoor. The navigation code was broken into these two modules to allow the system to take advantage of knowledge of the current environment. When traveling in an indoor office environment, assumptions about the regularity of corridors can be made (e.g., [Horswill, 1994]). For example, indoor environments usually have flat and regular floors, walls perpendicular to the floors, and few sudden drops. Simplifying assumptions can also be made about outdoor environments when using vision systems (e.g., [Lorigo *et al.*, 1997]). One simplifying assumption is the ground plane constraint, which assumes that the ground is flat and that all objects are on the ground. We can also assume that the wheelchair's user will only be traveling in wheelchair friendly environments.[1] Another assumption is that the edges of ramps and straight sidewalks form an upside-down V which approaches a vanishing point at the top of the image.

Indoor navigation relies upon bump, infrared and ultrasonic sensors (see Chapter 5) while outdoor navigation uses a vision system for its primary sensing mode (discussed in Chapter 6). The correct mode is selected automatically by the system using our indoor/outdoor sensor (see Chapter 7).

[1] If such an environment truly exists.... The goal of this thesis is to assist users in their travels through the world, but no robotic system can solve the difficulties of wheelchair travel in environments that are not compliant with the Americans with Disabilities Act.

Chapter 5

Indoor navigation

5.1 Sensor clustering

As discussed in Section 3.1, Wheelesley is equipped with infrared sensors, sonar sensors, and two bump sensors on the front bumper. (In early work, the bump sensors were contact switches; in later work, two Hall effect sensors were used.[1]) Placement of the sensors is shown in Figure 5-1.

Indoor navigation relies on the infrared sensors, sonar sensors and bump sensors. The infrared sensors give binary readings that indicate if something is about one foot from the sensor. As soon as an infrared sensor signals that it is blocked, the robot immediately corrects to avoid the obstacle. These close reading sensors function to avoid obstacles not anticipated by the sonar sensors. The sonar sensors return distance information. The sonar readings are smoothed over a short window of readings to diminish the effect of any noisy readings; the smoothed value is used to determine

[1] A voltage is generated across a current in the presence of a magnetic field; this is called the Hall effect. A digital Hall effect switch is an open circuit in the absence of a magnetic field and turns on when the voltage goes over some threshold value. [Everett, 1995]

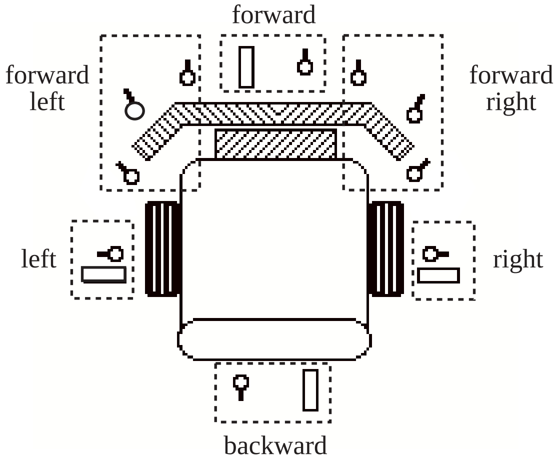


Figure 5-1: Placement of the sensors on the wheelchair and their clusterings. The sensors are clustered into six groups: forward, forward right, right, backward, left and forward left. Infrared sensors are drawn as circles with a line pointed in the direction that the sensor is pointed. Ultrasonic sensors are represented by rectangles. The forward group includes two Hall effect switches mounted on the front bumper, which are not illustrated in this figure.

if there are obstacles too close to the wheelchair. The switches/Hall effect sensors are mounted on the wheelchair's bumper and are used as a last resort. If an obstacle was missed by the infrared and sonar sensors while traveling forward, the bumper will hit the obstacle. Empirically, bumper hits are very infrequent (only one bumper hit in over ten hours of user testing).

The robot is able to traverse long hallways without requiring user corrections. The system uses infrared and sonar sensors pointed out to each side at a angle perpendicular to the forward movement. The system stays centered in the hallway by keeping sensor readings on each side of the chair equal. Temporal smoothing of sonar values keeps the wheelchair from losing its centering when an open doorway appears on one side of the robot. While moving down the hallway in this manner, the chair will also avoid obstacles in the path. Obstacle avoidance takes priority over the hallway centering. In designing the system, centering was chosen over wall following to keep the chair in a better position for turning to avoid obstacles.

Dervish, designed at Stanford to navigate office environments, used only sonar sensors to extract information about the world [Nourbakhsh, 1998]. The robot uses clusters of sonars, which consist of multiple sonar sensors with overlapping coverage, to detect obstacles in the world. This redundancy results in the more reliable detection of obstacles.

Wheelesley also uses clusters of sensors to improve reliability of object detection. However, unlike Dervish, we use sonar sensors in conjunction with infrared sensors. The sensor readings are clustered according to their location on the robot. The groups of sensors are left, forward-left, forward, forward-right, right and backward; these groups are shown in Figure 5-1.

5.2 Navigation algorithm

The inputs to the indoor navigation algorithm are the values of the infrared sensors, sonar sensors and bump sensors. Additionally, the user command given through the user interface is an input to the algorithm. The outputs of the algorithm are motor commands for the left and right motors.

The system uses reactive navigation because the user must be able to successfully navigate novel environments immediately with the absence of a complete map. Because there is an intelligent human giving high-level navigation commands to the wheelchair robot, the common limitation of a reactive navigation system (lack of planning) is alleviated. The system concentrates on what a reactive system can do well by carrying out the user's commands while keeping the user safe, leaving the planning that a reactive system typically omits to the user. If a user of the system desires a greater level of robot autonomy, maps of commonly traveled environments such as the home and the office could be incorporated. Path planning for indoor robotics has been studied extensively (see [Kortenkamp *et al.*, 1998] for examples) and could be implemented on the robotic wheelchair base.

For traveling forward, if the chair detects no obstacles, it will drive forward at the maximum speed allowed (set through the user interface). While traveling forward, the chair keeps itself centered in a hallway by checking the group of side sensors on the right and the group of side sensors on the left. If one group is reading closer than the other, the chair will turn slightly towards the direction of the farther group. When the readings of both groups are approximately equal, the chair continues going forward while continuing to poll the side sensors for centering.

If any of the forward groups of sensors trigger while traveling forward, the chair will correct to maintain a safe path. If the path is completely blocked in front but is clear to the right, the chair will turn to the right. If the chair is blocked in front and to the right, the chair will turn left. We decided to always select right first in a blocked situation, which can be used to avoid additional commands. The user can let the robot make right turns when coming to the end of a corridor, if a right turn is desired. If the user prefers to turn left, he will stop the forward movement and select the left arrow. This behavior allows the user to issue fewer commands to travel on his selected route.

If the chair is completely blocked in front, on the right and on the left, it will back up, using its rear facing sensors to detect obstacles. Since the chair just came from a clear path behind the robot, it is highly likely that the path is still clear. In user tests, the chair very rarely encounters a situation where it needs to back up. In the presence of malicious obstacles (such as people trying to break the system), this situation could occur more often. However, most people do not try to block the path of a wheelchair user; in fact, it's most often the opposite. Most people make it a point to avoid the path of an oncoming wheelchair. However, the group of physical therapists who evaluated the system (see Section 8.3) crowded the chair to try to push the system to its limits; in this case, the system did need to back up a few times.

5.3 User tests

Single switch scanning is the access method of last resort for powered wheelchairs, primarily because drift is a significant problem. To correct a drift to the left or the right, the user must stop going forward, wait for the scanning device to get to the arrow for the direction of choice, click to turn the chair, stop turning, wait to scan to forward and then click to move forward again. Robotic assisted control can improve the ease and speed of driving using single switch scanning. Under robotic control, sensors are used to correct the drift problem and to avoid obstacles, thus taking over the low-level control. The user is only required to give commands to change direction, i.e. issue high-level commands.

For these experiments, the user interface consists of four large arrows and a stop button (Figure 8-5). The user interface was designed to look and function like a standard single switch scanning device. The interface scans to the forward arrow, the right arrow, the left arrow and the back arrow until the user selects a command by hitting a switch. The interface pauses at each possible selection for one second. Since all test subjects are able-bodied, the commands are latched.[2] To stop driving or turning, the user hits the switch again. After the stop command is given, scanning starts again on the forward arrow.

5.3.1 Design of the user tests

An experiment to test the performance of subjects under robotic assisted control and under standard manual control was designed to determine if robotic assistance improved driving performance using single switch scanning as an access method. Fourteen able-bodied subjects (7 men and 7 women), ranging in age from 18 to 43, were tested. All subjects were familiar with using computers and none had driven the wheelchair before.

At the beginning of a session, the subject was shown the wheelchair. Sensors that are used in robotic assisted control were pointed out and explained briefly. Safety

[2] Under robotic control, Wheelesley can be set to operate in a latched or a non-latched mode. In the rehabilitation community, latching is defined as executing a command given through the access device until the user either stops the wheelchair or issues another movement command. A standard wheelchair driven with a joystick operates in a non-latched mode; when the user lets go of the joystick, the wheelchair stops. In the case of a joystick, a latched mode would not make sense. However, in the case of an access method like sip-and-puff, where a user issues a command by blowing or sucking, a latched mode allows the user to stop blowing or sucking while the chair continues to move. To stop or change directions, the user issues a new command by blowing or sucking. Rehabilitation providers determine whether a system should be latched based upon the access method and the abilities of the user. If the user is unable to issue commands quickly, a non-latched system is required.

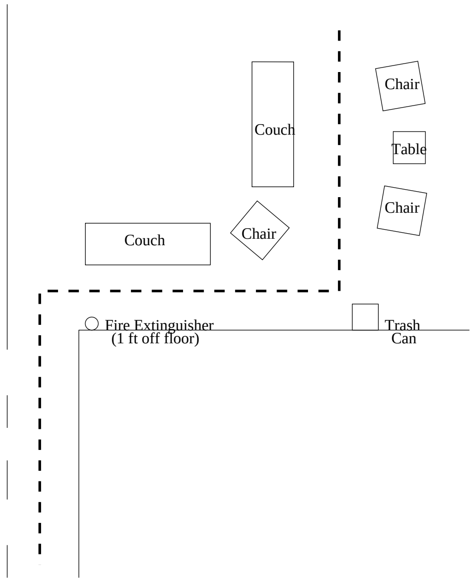


Figure 5-2: A diagram of the test course. Subjects drove the course four times in each direction, alternating driving methods after a round trip of the course. The course is 20 meters (65.7 feet) long. Obstacles are couches, chairs, a table, a trash can, and a fire extinguisher mounted on the wall 30 cm (11.8 inches) above the floor. The three doors in the hallway could be open or closed, determined by the office occupants.

measures, such as the power button, were discussed. Then the two driving methods were explained to the subject. After this introduction, the subject was seated in the wheelchair and the user interface was connected to the wheelchair. The single switch scanning interface was explained to the subject, who then practiced using the interface with the motors turned off.

Once the subject was comfortable with the interface, the session entered a practice phase in which the subject first tried robotic assisted control and then standard manual control. The subject practiced both methods until he expressed an understanding of each control method; subjects usually spent about two minutes trying each method. All practice was done off of the test course, so that the subject was not able to learn anything that would assist him during the test phase.

The course was designed to include obstacles (several couches and chairs, a fire extinguisher mounted to the wall 30 cm (11.8 inches) above the ground, a trash can, and a table) and turns to the left and to the right. A diagram of the course is given in Figure 5-2.

The test phase consisted of four up-and-back traversals of the test course, alternating between the two control methods. Half of the subjects started with robotic assisted control and the other half started with standard manual control. Each up-and-back traversal consists of two parts: running the course from the couch area to the hallway and then the return trip. The turn in the middle of the course is not counted as part of the run, as turning completely around in the middle of the hallway is not a normal driving occurrence. The total session time for each subject was approximately 45 minutes.

Most data collection was done by the computer which was running the user interface. The researcher only recorded the number of scrapes made by the chair. At the completion of the test, the user was asked to rank standard manual control and robotic assisted control on a scale from 1 (worst) to 10 (best).

5.3.2 Results

There were four experimental performance measures collected by the computer: (1) the number of clicks required to navigate the course, (2) the amount of time spent scanning to get to the necessary commands, (3) the amount of time spent moving or executing the given commands, and (4) the total amount of time spent on the course

	Trial	Manual		Robotic	
Num Clicks	1	90.2	(16.3)	25.6	(4.9)
	2	77.1	(9.8)	22.0	(3.3)
Scanning Time (seconds)	1	93.6	(20.3)	30.9	(8.3)
	2	81.1	(13.0)	25.2	(8.6)
Moving Time (seconds)	1	311.6	(36.4)	268.2	(21.5)
	2	316.6	(36.2)	277.1	(28.4)
Total Time (seconds)	1	405.1	(42.1)	299.1	(18.4)
	2	397.7	(43.7)	302.3	(32.5)

Table 5.1: Results of the experiments: the number of clicks, amount of time spent scanning for commands, amount of time moving and total time to complete the course. The first number for each method is the mean and the number in parentheses is the standard deviation.

(scanning time plus moving time). Results are summarized in Table 5.3.2. A full listing of the data can be found in Appendix C.

Data for each experimental measure was analyzed using an ANOVA test. The differences between robotic control and manual control were highly significant with $p < .0001$ for all measures. On average, robotic control saved 60 clicks over manual control, which is a 71% improvement. Total time for robotic assisted control was 101 seconds shorter than manual control on average, which is a 25% improvement.

The differences between the two trials were significant for clicks ($p = .003$) and for time spent scanning ($p = .015$). There was not a significant difference between trials for moving time or total time.

The only performance measure not collected on the computer was the count of the number of scrapes. A scrape was recorded when the chair brushed along a wall or piece of furniture. Bumps with the bumper were also counted as scrapes. No subject hit a wall or an obstacle with great force. The average number of scrapes per run under manual control is 0.25. The average number of scrapes under robotic control is 0.18. These numbers are not significantly different.

Finally, the subjects were asked to evaluate the two driving methods by giving a score from 1 (worst) to 10 (best). The average score for standard manual control was 3.5. The average score for robotic assisted control was 8.7. These scores are highly significant with $p < .0001$. No test subject preferred manual control over robotic control.

5.3.3 Discussion

Subjects drove more efficiently and preferred to drive with robotic assisted control. Robotic control automatically adjusts for drift where manual control does not. When traveling down a long hallway under robotic control, a user can click on forward at the beginning of the corridor and does not need to do anything more until he wishes to stop or turn. Under manual control, the user must make many adjustments to compensate for drift.

The total time taken on a test run is a sum of the scanning time and the command execution time. Both scanning time and execution time improved from manual control to robotic control. As would be expected, if fewer clicks are issued, the scanning time required is shorter. Estimating that forward is clicked 50% of the time, left is clicked 25% of the time and right is clicked 25% of the time,[3] with a scan time of one second and an estimated reaction time of one half second in our able bodied subjects, each click would require an average of 1.25 seconds. As Table 5.3.2 shows, the scanning

[3] Empirically, backwards commands are issued very infrequently.

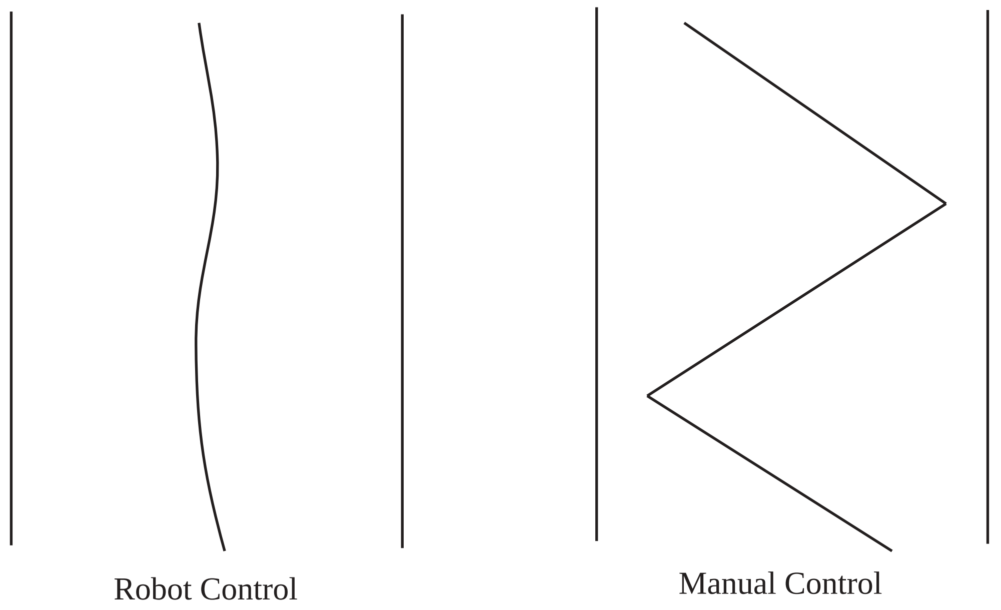


Figure 5-3: Sketches of representative paths taken by the robotic wheelchair during user tests. On the left, the path under robotic control is smoother and shorter. The path on the right shows the tendency of subjects to push the wheelchair as close as possible towards the wall before stopping and turning. This behavior occured because subjects wanted to minimize the amount of clicking and scanning required to traverse the course. The shorter path taken under robotic control resulted in faster time for course traversal.

time is approximately 1.25 times the number of clicks.

Although the wheelchair's preset speed was the same for both control types[4], subjects traversed the course in less time under robotic control than under manual control; the average driving time under robotic control was 87% of the average driving time under manual control. Driving the course took longer under manual control due to an uncoached attempt to minimize clicks on the part of the subjects. In order to issue the fewest clicks, subjects would drive as close to the wall as possible, stopping the wheelchair just before a collision. Under robotic control, the wheelchair would not allow the user to get as close to the walls. Figure 5-3 shows a sketch of a sample path

[4] The speed was preset to approximately 4 m/min.

down a corridor for each control method. The path under robot control is shorter than the path under manual control. Therefore, the faster driving times for robotic control occur due to a shorter path being taken.

Each user executed two trials for each control method. Learning played a significant role between trials when counting clicks and scanning time. As the user became more comfortable with the system, he was able to judge more effectively when it was necessary to make adjustments to the current course. There was no significant effect of learning on moving time and total time between trials; since the speed is held constant throughout the experiment, the user can not significantly reduce the amount of time required to travel the course between trials of the same control method.

A single subject ran the course 10 times in manual mode to determine how learning could affect the number of clicks and scanning time. The subject was this researcher; a naive user is not required to test for optimal performance. Over the 10 runs, the average number of clicks in a test run was 71.4 with a standard deviation of 9.5. Over the last 5 runs, the average number of clicks was 68 with a standard deviation of 4. Scanning time averages 73.7 seconds (standard deviation 12.0) over all 10 runs and 68 seconds (standard deviation 5.3) over the last 5 runs. Optimal performance for this course in manual mode will not approach the average performance in robotic mode.

5.3.4 Discussion of comparable user tests

The NavChair system was tested in an indoor environment using voice control as the access method [Simpson and Levine, 1997]. Six able bodied subjects navigated through three different scenarios (room traversal, door passage and wall following), four times with navigation assistance and then four times with no navigation assistance. For each scenario, it took longer to navigate using navigation assistance, primarily because the chair slows down as it gets closer to obstacles. However, no collisions occured with navigation assistance, while there were occasional collisions with no navigation assistance. Test subjects prefered driving with navigation assistance.

The VAHM Project was also tested in an indoor environment using single switch scanning as the access method [Bourhis and Pino, 1996]. Four able bodied users familiar with computers drove the wheelchair through a course simulating a kitchen and living room environment in manual mode and in an assisted mode which provided obstacle avoidance. Assisted mode resulted in a 13% improvement in the number of actions required on the interface screen (which we called the number of clicks required). Execution time also improved.an average of 7.7%. The tests were executed at three scanning rates: 0.8, 2.5 and 4.5 seconds. There was a 2.5% improvement in execution time for the 0.8 second rate,[5] a 7.7% improvement for the 2.5 second rate, and a 13% improvement for the 4.5 second rate. One would expect to see a more dramatic improvement in total execution time for longer scanning times since fewer clicks result in a greater time savings.

Indoor user tests of Tin Man II used a joystick and buttons for the access method [Miller and Slack, 1995]. The test course was 50 meters long and included a hallway, a doorway and two rooms. Subjects were told to attempt to minimize their joystick movements in both the manual and assisted tests. Manual mode required 50% more

[5] This is the closest to our 1 second scanning rate.

joystick moves than the obstacle avoidance mode. The time required to traverse the course was less than 10% longer in assisted mode than in manual mode.

Chapter 6

Outdoor Navigation

While indoor navigation can be successful with infrared and sonar sensors, outdoor navigation in our system can not rely on these available sensors. The walls that appear in indoor environments are missing in outdoor environments. There are only four sonar sensors on the wheelchair, one pointed in each compass direction, which provide information that is too sparse for outdoor navigation. Infrared sensors are likely to fail outdoors due to the infrared light in sunlight. To navigate successfully outdoors, we use a vision system to provide driving assistance.

The wheelchair must provide obstacle avoidance and path following in an outdoor environment, just as it provides in the indoor environment. The system uses the same philosophy of taking over low-level control in outdoor environments, just as is done in indoor environments. The robot continues to take high-level directional commands from its user and executes them while keeping the user safe.

The outdoor system does not attempt to navigate autonomously. We have always assumed that the chair is being driven by a sighted user who can assist in the high-level navigation. Due to this assumption, we did not develop an outdoor system that can automatically cross streets. The user can issue a forward command when it is safe to cross the street.

6.1 Related work

Many groups have researched the problem of outdoor navigation. In some cases, the outdoor navigation is off road in unstructured environments (e.g. [Stentz and Hebert, 1995], [Lorigo *et al.*, 1997], and [Baten *et al.*, 1998]). In our application, the wheelchair will be driving on paved surfaces (or at least a very well packed dirt surface of a road) with structure such as curbs, sidewalks and ramps. The structure of our problem, most notably following sidewalks, is similar to the problem of driving a motor vehicle autonomously.

Motor vehicles that can drive European autobahns autonomously have been developed by Dickmanns and his colleagues. In early work [Schiehlen and Dickmanns, 1994, Maurer *et al.*, 1996], bifocal vision using one camera with wide angle lens and one camera with a mild telephoto lens was used to drive the vehicle. Two sets of these cameras were used: one pointed forward and one backwards. Saccading vision, which is similar to human vision, was implemented; the wide angle lens was used to pick up things of interest and the telephoto lens was used for a more detailed look.

The Navlab at Carnegie-Mellon University [Thorpe *et al.*, 1988] uses color vision for road following and 3-D vision, which uses a scanning laser range finder, for obstacle detection and terrain following. For road following, colors are classified into road and non-road. Texture is also used, but color is weighted more heavily. The geometry of the expected road (locally flat and straight, with parallel sides) is used to vote for the best road position based upon the road/non-road point classifications. Finally, the color model is updated to allow the system to adapt to changing road conditions. This method for road detection can not be used in our system since we are using black and white cameras.

Another system using color vision is Martin Marietta's Alvin (autonomous land vehicle) [Turk *et al.*, 1988]. One panning color camera is used to capture images. Color distributions are used to determine the road pixels. Then the road boundary is obtained and the 3-D edge points of the road are calculated.

The CMU UGV (unmanned ground vehicle) uses a neural net for road detection and a range sensor for obstacle avoidance. ALVINN, the neural network, has a single hidden layer. The input to the neural network is video images taken while a person was driving and the output used to train the network is the person's current steering at the time of the video image. The system is trained in about 3 minutes and can be trained for a wide variety of road conditions. In our application, we are more likely to see a wider variety of conditions than could be learned in this manner. Our users would not be able to retrain the system for different sidewalk and ramp conditions.

In [Charkari and Mori, 1993], the least square method is used to find the lines marking the road boundaries from a set of edge points. Since the slope a_k at time t_k is likely to be close to the slope a_{k+1} at t_{k+1}, a_{k+1} and b_{k+1} are selected to be the closest values to a_k and b_k. This eliminates most problems of edges of shadows and dirt. In our application, least squared fit has difficulties on ramps, where there are several lines present.

Madarasz [Madarasz *et al.*, 1991] used vision to detect the edges of an indoor corridor. To find the edges of the corridor, the image was searched for dark converging edges by scanning the lower portion of the image from center to the edge in both directions, looking for a transition to a dark region. These transitions were interpreted as junctions between the wall and the floor. To avoid problems with obstacles and doorways, only one wall was tracked at a time. If there were problems locating the current wall, the system switched to track the other wall. This type of scanning would not work in outdoor situations due to the lack of constant lighting conditions and shadows falling in the center of the sidewalk.

A method for detecting vehicles was developed by [Charkari and Mori, 1993, Charkari and Mori, 1995] to calculate the distance between the robot and the vehicle, the velocity of the vehicle and the vehicle's width. They use the shadow underneath the vehicle as a "sign pattern" or indicator. The intensity of the shadow under a car is fairly constant throughout the day, which other shadows and sunny surfaces vary depending upon lighting conditions. The system is able to detect vehicles under all weather (fine to cloudy) and lighting conditions. While our system is not currently attempting to detect and avoid vehicles for road crossings, this method could be applied to our work in the future to further assist users in outdoor environments.

6.2 Vision system

As described in Section 3.2, we are using the STH-V1 Stereo Head from Videre Designs. It is mounted on the front of the wheelchair's tray, pointing down at an approximately 40 degree angle. The bottom of an image taken by one of the cameras is .7 meters (2.25 feet) from the wheelchair's bumper and the top of the image is 8.2 meters (27 feet) from the wheelchair's bumper.[1] The baseline between the cameras is 69 mm (2.7 inches). The focal length of each lens is 6.5 mm.

We have simplified vision processing task by making assumptions about the environment. When traveling in outdoor environments, the wheelchair system must stay on a path, which is most likely to be a sidewalk or crosswalk. Sidewalks and crosswalks both have long lines heading towards a vanishing point, although they can also be curved.

There are two vision processing modules whose outputs are combined to provide driving assistance: obstacle detection and local path detection. The SRI Small Vision System (SVS) is used for stereo processing.

6.2.1 Stereo processing with the SRI Small Vision System

The SRI Small Vision System (SVS) [Konolige, 1997] uses the area correlation method to compute stereo matches on a pair of images. In the area correlation method, patches of the images are matched instead of features. The system first corrects images for camera distortions, then applies a Laplacian of a Gaussian filter to transform the image into a normalized form. Next, a variable disparity search is executed using a 16, 24 or 32 pixel window (set by the user). The results are filtered using an interest operator and a left/right check. The algorithm has been optimized to use the SIMD

[1] The measurement for the top of the image is for distinguishable points on the ground plane. The pixels at the very top of the image may be farther away.

instructions available on MMX processors.

6.2.2 Obstacle detection

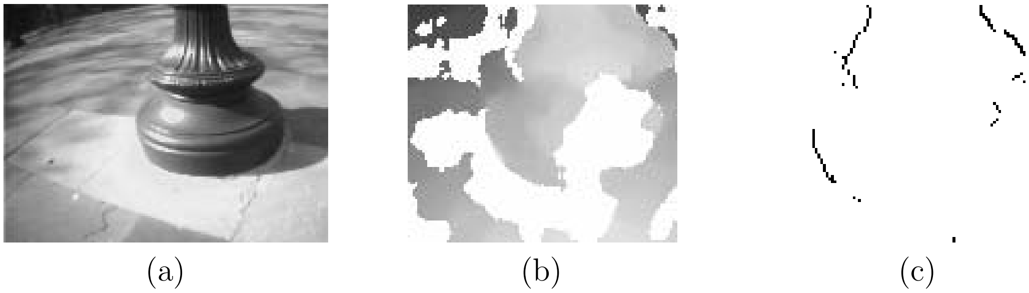
To detect obstacles, the disparity image computed by the SVS is scanned horizontally and vertically, looking for changes in disparity over a specified threshold. This change in disparity is likely to indicate that an obstacle is present.

We look for change in disparity between adjacent points. However, since the disparity map is not fully populated, we consider points on either side of one or more uncomputed pixels to be adjacent. This method results in more edge points being found, since there are often uncomputed points around the edge of an obstacle boundary. Many edges of obstacles are missed when changes are not computed over the non-mapped regions.

The obstacle boundary point is assigned to the larger disparity, which corresponds to the closer point, assuming that the closer point is the obstacle. The direction of the high disparity is also marked, to avoid the risk of driving into a large obstacle where the edges were marked but the center looked clear.

When scanning horizontally, we can assume that all points on the ground plane have the same disparity on a scan line. Any points that differ from this value are likely to be obstacles.

Vertical scanning takes a slightly different approach. As one scans from top to bottom, the disparities should increase smoothly since points at the bottom of the image are closer than points at the top of the image. To catch large disparity changes



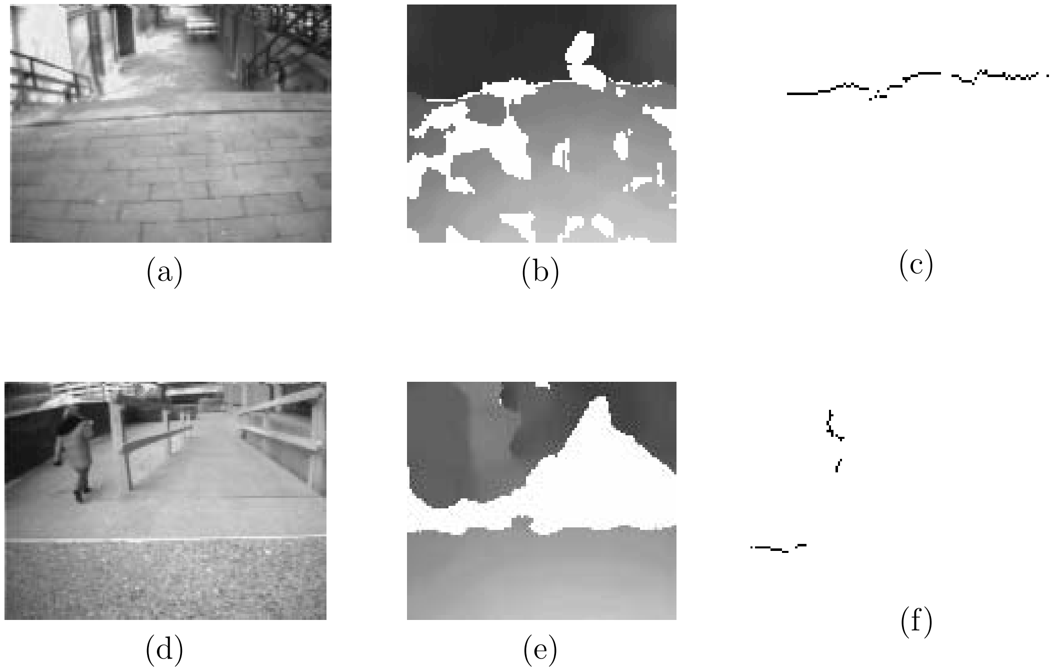
(a) (b) (c)

Figure 6-1: Results of the obstacle detection vision module for an image showing the bottom of a light pole in an open courtyard. (a) Left image of the stereo pair. (b) Disparity map computed by SVS. Brighter points are closer to the wheelchair and darker points are farther away. White points are do not have a valid disparity in the map. (c) Computed obstacle boundaries.

across gaps, a 15 pixel window is checked for disparities. If there is a large unmapped region, the top and bottom of it will not be compared. This may cause some points to be missed, but the boundary between obstacles would be likely to have disparities matched whereas a long stretch of sidewalk would be more likely to be blank in the disparity map. Additionally, a larger threshold is used for marking disparity changes in the vertical direction.

Figure 6-1 shows the results of our obstacle detection code run on an image where the chair is approaching a light pole in an open courtyard. Figure 6-1a shows the left image of the stereo pair. Figure 6-1b shows the disparity map computed by the SVS. Figure 6-1c shows the obstacle boundary points extracted by the system.

The vertical scanning for disparity changes allows us to find the drops associated with open stairways. Figure 6-2 shows the results of processing on images where there are open stairways in front of the wheelchair. In Figure 6-2a, the wheelchair was driving up to an open stairway that was much wider than the wheelchair. The


(a) (b) (c)

(d) (e) (f)

Figure 6-2: Results of obstacle detection processing for open stairways. Figures (a) and (d) show the left image of the stereo pair. Figures (b) and (e) show the disparity map computed by SVS. Bright points are closer to the wheelchair. White points have an uncomputed value in the disparity map. Figures (c) and (f) show the results of the obstacle detection module.

vertical scanning code finds the drop in the video sequence from which this image is taken, starting out from when the stairway is about 15 feet away (near the top of image) to when it is right in front of wheelchair. Figure 6-2d shows a situation where the wheelchair was driving up to a ramp on the right of the image with an open staircase on the left hand side. In the obstacle map, a line is drawn right across the top of the stairs.

6.2.3 Local path detection

We make the assumption that the current path (sidewalk, crosswalk) is locally straight. To find the edges of this path, the system finds the edge points in the image using the intensity gradient and filtering, then fits a line to the edge points on the left half of the image and a line to the right half of the image. The left image of the stereo pair is used to locate the current path; it differs from the right images by a few pixels at the edges, but not enough to affect the outcome of the vision routines. Several filtering steps are taken to eliminate noise. Figure 6-3 shows a video frame and the results of each progressive processing step.

The following steps are computed to calculate these two lines. First, the image is median filtered (Figure 6-3b). A five pixel horizontal window, centered on the pixel being computed, is used to filter the image. This eliminates some noise caused by "textured" sidewalks (e.g., concrete made with rocks ranging in color from white to gray to black).

After median filtering, edge points are computed using an intensity gradient (Fig-

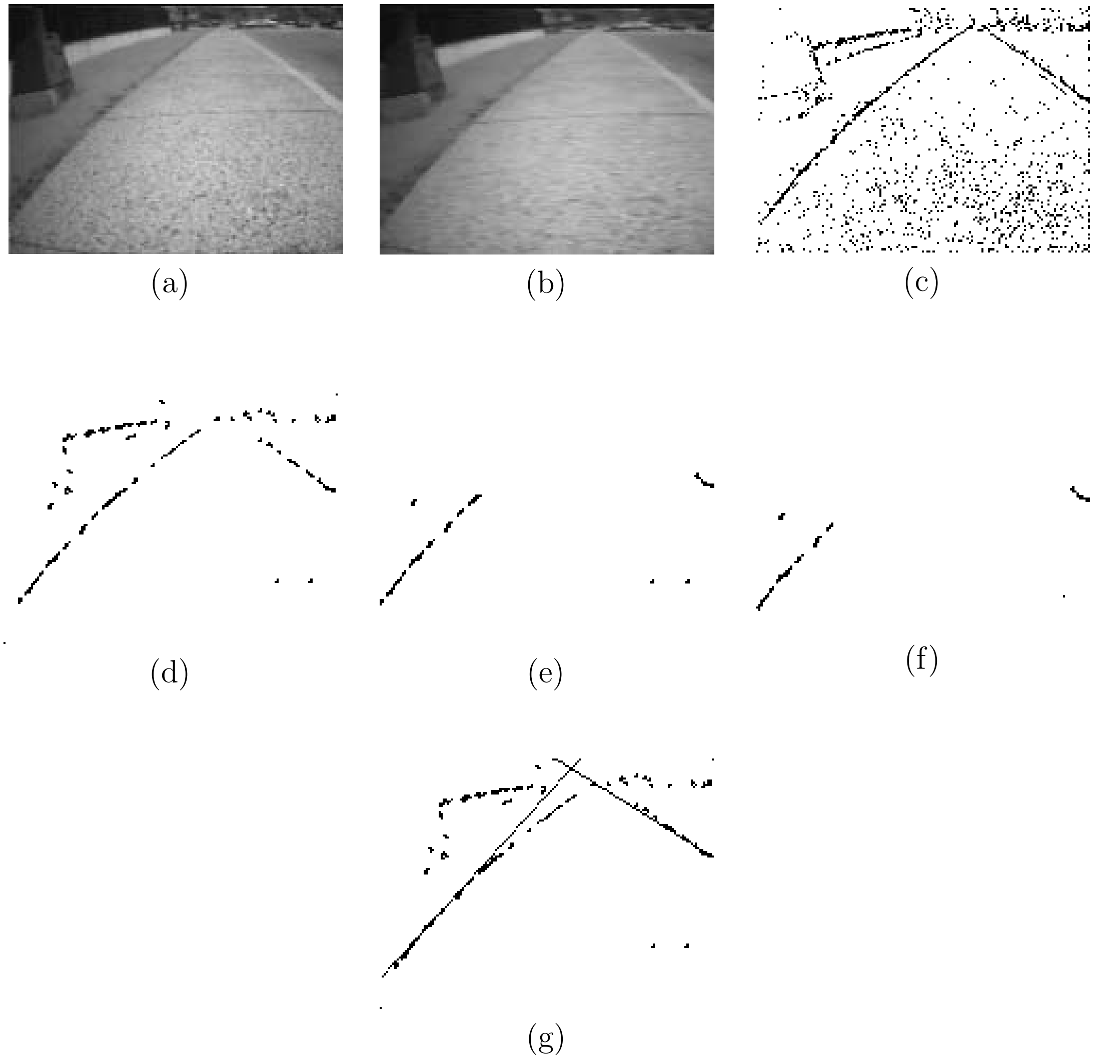

(a) (b) (c) (d) (e) (f) (g)

Figure 6-3: Results of each processing step for local path detection. (a) The original image. (b) Median filtering with a 5 pixel window centered on the pixel being computed. (c) Edges found by computing the intensity gradient. (d) Edge points remaining after filtering out points with fewer than two neighbors, then scanning the points again to remove any points with fewer than one neighbor. (e) Edge points minus the points in the horizon region. We are only trying to find lines for the local path. (f) Points minus the 25% of the points with the largest average distance from the other edge points. (g) The set of points from (d) with the left and right lines computed using the least absolute deviation method drawn on the image.

ure 6-3c). A point is compared to its neighbor to the right and its neighbor below it. If the intensity values in both directions vary more than a specified threshold, the point is marked as an edge point.

The edge points found are further filtered to eliminate outliers. Edge points forming the edge of a sidewalk or obstacle are likely to have more neighboring edge points than a noisy point. We eliminate any points with fewer than two neighbors, then pass over the points again and filter out any points with fewer than one neighbor (Figure 6-3d). We deselect points at the top of the image (Figure 6-3e) since we are looking for the edges of the local path only. This also eliminates any horizontal line near the horizon of the image that would skew the line. Finally, one quarter of the points with the greatest average distance from the other points in the set are eliminated (Figure 6-3f).

Lines are then fit to the left side and right side of the processed edge points using the least absolute deviation method (one description can be found in [Press *et al.*, 1992]); the result is shown in Figure 6-3g. The least squares method was too sensitive to noise points that remained after filtering, skewing the line to fit outlying points. By trying to minimize the absolute deviation of points from the line, we fit lines more effectively in noisy situations.

While the filtering steps eliminate noisy points caused by differing sidewalk textures, it does not eliminate the edges of shadows. If we were using color cameras, processing could be done to eliminate shadows. However, in the current system with grey scale images, we usually test in situations without bright sun creating dark shadows. When we have tested in situations with shadows, the edge of the shadow can be read as the edge of the sidewalk, causing the wheelchair to overestimate or underestimate the location of the curbs. Shadow filtering on color images would eliminate this problem.

6.3 Navigation algorithm

The results of the two vision modules (obstacle detection and local path detection) described above are combined to provide assistive navigation in outdoor environments.

6.3.1 Sidewalk following

The left and right lines computed by the local path detector are used to follow sidewalks. After each frame is processed, the system computes the percentage of points that fall within 1.5 pixels of the line. A line with a higher percentage of points on it should be the better fit for the data, implying that it has found the most salient edge. In our images, this edge is usually the sidewalk. In addition to selecting based upon this criterion, we also require that the set of edge points contains at least 8 points. For smaller numbers, it is likely that only noise points are included in the edge set.

The system judges whether it should drive straight, left or right based upon the slope and intercept of the selected line. Many videos of driving on sidewalks were taken to determine the normal range of slopes and intercepts for driving straight. More videos were taken with what would be considered poor driving, including going to the edges of curbs. The observed data was used to define normal ranges to the system. The current values are then passed through the function to determine whether a turn is warranted. If either the slope or intercept is out of range, a right or left turn is made to move the sidewalk edge closer or farther away from the chair. Conditions are also included to handle cases where the left edge has passed into the right region or the reverse.

If a computed line has a slope of 0 (or is close to 0), it is ignored. These lines are more likely to be the division between two concrete blocks of the sidewalk than the edge of a curb, given that the wheelchair is started pointing up a sidewalk rather than at the edge. The algorithm will drive the chair down the sidewalk in a manner that does not allow it to turn toward the curb fast enough for it to approach a 0 slope.

In the case where a valid line is not computed on either the left or the right side of the image, the robot continues to drive forward at a decreased speed. A valid edge is usually detected within a few frames, allowing the robot to continuing driving at its normal speed.

6.3.2 Obstacle avoidance

For obstacle avoidance, we make the following assumptions:

Assumption 1: Obstacles in the middle of the sidewalk are likely to be moving (people, animals, bikes). Unlike a hallway where people might leave boxes, it is unlikely that people will leave obstacles in the middle of the sidewalk. Stationary obstacles may appear at the edges of sidewalks.

Assumption 2: Most sidewalks are fairly narrow. That is, there is not much room to maneuver around an obstacle in the center of the sidewalk.

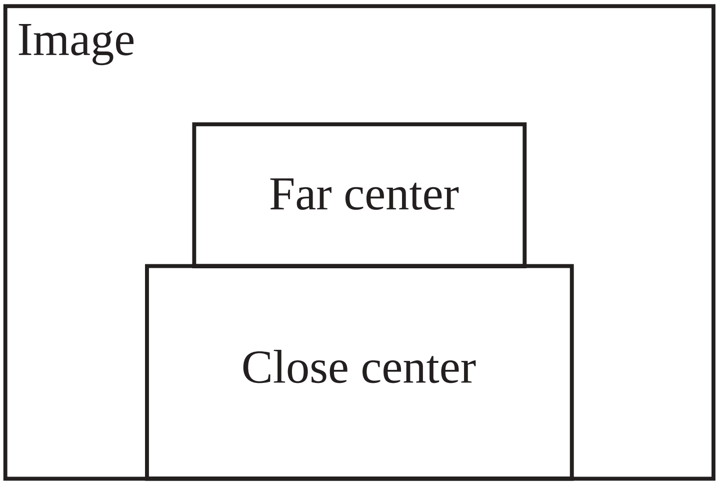


Figure 6-4: The regions used for obstacle avoidance. The far center region is approximately 5-10 feet from the wheelchair and the close center region is approximately 2-5 feet from the wheelchair. If an obstacle is detected in the far center region, the chair slows down. If an obstacle is detected in the close center region, the chair stops.

Assumption 3: People will walk faster than the chair is traveling.

Given these assumptions, the behavior for dealing with obstacles is to slow if they appear in the far center region and to stop if they get too close to the chair in the center region. The close center region is approximately 2-5 feet from the wheelchair and the far center region is approximately 5-10 feet from the wheelchair. The regions are shown in Figure 6-4.

Once the chair is stopped, one of the following situations will occur. If the obstacle is moving, it will clear and the chair can continue. If the obstacle is stationary, the user can direct the chair around the obstacle by turning to the left or right.

Obstacles that appear at the far left and far right of the image are usually avoided by the system's sidewalk following behavior. A light pole or fire hydrant near the edge of the curb would cause the edge of the sidewalk to look closer, causing the wheelchair to steer around the obstacle. If the wheelchair were to turn toward an obstacle on the edge of the image, the obstacle avoidance behavior described above would be activated.

Obstacle avoidance takes precedence over sidewalk following. The chair will not move until the obstacle is cleared, either by it moving away or by the user turning the chair away from the obstacle.

6.4 User tests

As with the indoor navigation system, an experiment to test the performance of subjects under robotic assisted control and under standard manual control was undertaken to determine if robotic assistance improved the driving performance using single switch scanning as an access method. Data was collected with two able-bodied subjects: a 29 year old woman and a 31 year old woman. Both were familiar with using computers. The second subject had participated in the indoor user tests while the first subject had never driven the wheelchair before.

The outdoor user tests followed the same protocol for the indoor user tests described in Section 5.3.1. Each subject ran the test course four times, twice under each control method. The test course was a sidewalk on MIT's main campus that includes a curve to the left and to the right. Obstacles were not placed on the sidewalk; people walking on the sidewalks were natural obstacles that the navigation system (robotic control) or subject (manual control) needed to avoid. The test course is 35 meters (96 feet) long. A diagram of the course is shown in Figure 6.4.

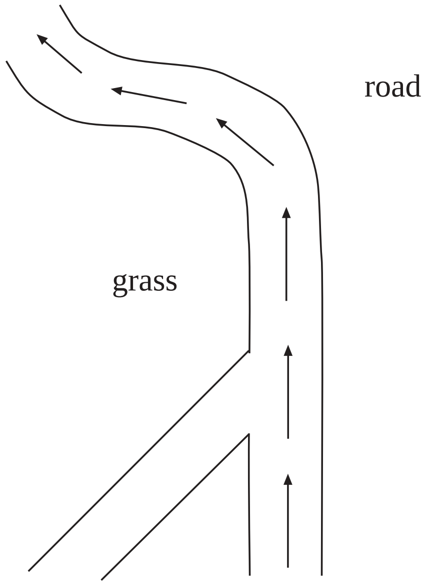


Figure 6-5: The outdoor test course, which is 35 meters (96 feet) long. Subjects started from the bottom of the diagram. No stationary obstacles were on the course, but pedestrians were natural obstacles throughout the tests.

6.4.1 Results and discussion

Results of the outdoor tests are shown in 6.4.1 and 6.4.1. Robotic assisted control resulted in an 88% reduction in driving effort, as measured by the number of clicks required to drive the test course. As discussed in Section 5.3.3, fewer commands issued means that less time is spent scanning to commands that need to be issued. This resulted in 22% less time time needed to traverse the test course under robotic assisted control than manual control.

For three of the four robotic assisted control test runs made, the user issued only two clicks: one to start going forward and one to stop the wheelchair at the end of the course. The other robotic assisted run took 12 clicks. Subject 2 was very nervous about robotic control after her practice run. During the practice run, the wheelchair got very close to the curb where the sidewalk bends to the left. The subject tried to

	Manual		Robotic	
	First run	Second Run	First run	Second run
Number of Clicks	56	32	2	2
Num Forward Clicks	13	7	1	1
Num Right Clicks	7	4	0	0
Num Left Clicks	8	5	0	0
Num Back Clicks	0	0	0	0
Num Stop Clicks	28	16	1	1
Scanning Time (sec)	116.7	42.9	0	0
Moving Time (sec)	155.1	147.1	124.9	140.1*
Total Time (sec)	271.8	190.0	124.9	140.1

Table 6.1: Results of the outdoor experiment with Subject 1: the number of clicks, amount of time spent scanning for commands, amount of time moving and total time to complete the course. *The researcher stood in front of the chair as an obstacle for approximately 10 seconds during this run. The chair slowed as it approached, then stopped when it was too close to the researcher. It resumed moving once the researcher moved out of the way.

	Manual		Robotic	
	First run	Second Run	First run	Second run
Number of Clicks	38	26	12*	2
Num Forward Clicks	10	7	4	1
Num Right Clicks	4	0	0	0
Num Left Clicks	5	6	2	0
Num Back Clicks	0	0	0	0
Num Stop Clicks	19	13	6	1
Scanning Time (sec)	44.6	32.0	36.3	0
Moving Time (sec)	148.1	144.4	147.6	151.7
Total Time (sec)	192.7	176.4	183.9	151.7

Table 6.2: Results of the outdoor experiment with Subject 2: the number of clicks, amount of time spent scanning for commands, amount of time moving and total time to complete the course. *The subject unintentionally hit stop immediately after starting the run. Removing this would result in a total of 10 clicks.

stop the chair with one click, but ended up clicking multiple times. In this confusion, she managed to stop the chair, but then issued a “right” command. Since we have assumed that the user will only issue a turn command when in a safe location to turn in place, no obstacle or path checking occurs during these turns. The chair started to turn on the edge of the curb, taking the right front caster off the curb. After this near accident, the subject was very leery of the robotic assisted control mode during her first test run. Two of the clicks are due to her testing the stop command at the start of the run. The additional clicks were issued in the curved part of the sidewalk when the subject was unwilling to trust the system. In her second run, she was more comfortable with the control method and drove the test course using only two clicks.

The situation described above demonstrated that the system will need to include error checking during right and left turns before the system can be tested with disabled users. Sonar sensors pointing down toward the sidewalk could be used to detect drops around the wheelchair before executing a turn.

6.4.2 Discussion of comparable user tests

No quantitative results for outdoor trials of other systems were found in the literature, mostly due to the fact that most systems are not designed for both indoor and outdoor use. A description of an outdoor trial of the TAO Project can be found in [Gomi and Griffith, 1998]. In this trial, the wheelchair ran outdoors on a sidewalk between two walls of snow. The walls of snow could be considered simulations of indoor walls. No trials were run on sidewalks under their usual conditions.

Chapter 7

Mode Selection

The Wheelesley system has been designed with two navigation modes: indoor and outdoor. This split between the two environments allows the navigation code to exploit common features of each environment. (See Chapter 4 for a complete discussion of the navigation modes.) However, this division requires that the correct mode be selected for the current environment. While there is a person sitting in the chair who will know whether he is indoors or outdoors, a user of our system probably will not be able to toggle an additional switch or click a button with ease. For a person using single switch scanning, a mode selection would need to be added to the scanning pattern. With a scanning pattern of "forward, left, right, backward and mode select", the need to select between indoor and outdoor navigation mode is made as important as giving the current driving command to the robot. Yet the mode will need to be selected infrequently, only as often as the user moves between indoors and outdoors. The system developed for our target population must make driving easier rather than add to the overhead of driving.

We are not aware of any other research that has addressed this problem. Most research in mobile robotics is concerned either with indoor navigation or with outdoor navigation, but not both.

The issue of detecting indoor vs. outdoor images has been addressed in the area of video image retrieval. Texture orientation can be used to classify pictures into country and city/suburb classes [Gorkani and Picard, 1994]. The system finds the dominant orientation(s) of images and uses the observation that man made structures have more vertical orientation or a combination of vertical and horizontal orientation (perhaps skewed due to the camera angle) than natural scenes.

A combination of color and texture features can be used to determine indoor and outdoor classification of digitized photographs [Szummer and Picard, 1998]. The addition of texture features improves accuracy over methods that use only color information. Another method for classifying images into indoor and outdoor categories is spatial color distribution, which measures intensity and illumination changes [Vailaya, 2000]. First and second order moments in LUV color space are used as features. There is more variation in spatial color moments in outdoor images than indoor images. The classification success of this method is comparable to Szummer's color and texture model described above.

A similar type of vision processing might work for our application. However, our need for real time vision processing for outdoor navigation has kept us from imposing additional requirements upon the vision system. With the MRT VideoPort Pro our only choice as a frame grabber for a PCMCIA slot, we are already limited to 15 frames per second with no vision processing. With vision processing, our frame rate is 5.5 frames per second. Additional vision processing could slow this rate below where it could be useful for safe navigation.[1]

The wheelchair system automatically selects the indoor or outdoor mode using an indoor/outdoor detector developed for this research. The detector uses information about the properties of light, presence or absence of a ceiling, and temperature data to determine the current state of the robot. Use of these sensors adds no overhead to our vision processing for outdoor navigation.

It is important to have a seamless transition between the indoor and outdoor modes. The user should not have to stop moving at a building's door to allow the system to select the proper mode. While the robot uses multiple navigation modes to take advantage of regularities in environments, these modes should be transparent to the user. The user should always interact in the same way with the system and does not need to know which navigation mode is currently running (or, in fact, that there are different methods for navigation and obstacle avoidance).

The NavChair system also provides an automatic mode selection for its user, but for indoor navigation only [Simpson and Levine, 1999]. The navigation modes for indoor tasks are door passage, general obstacle avoidance and automatic wall following. The mode is selected based upon the environment or a combination of the environment and location. The environment is determined using the robot's ultrasonic sensors. A topological map is used to determine the chair's location; the experimenter sends a signal to notify the chair when it has moved from one location

[1] With the vision system running at the same time as the mode detector, which uses the second PCMCIA slot on the notebook computer, the frame rate drops to 9-10 frames per second with no vision processing. While the detector prototype uses a data acquisition card, the next version of the detector could be implemented on a small microprocessor that interfaced directly with the robot, eliminating this decrease in speed.

to another. To select the proper mode based upon the environment alone, the system used the following rules: If the environment is a door, the chair enters door passage mode; if there is an obstacle in front, general obstacle avoidance is selected; if there is a wall to the left or to the right, automatic wall following is triggered; if none of these conditions are met, the chair will operate in general obstacle avoidance mode. The selection process for environment and location uses Bayesian network to choose the proper mode.

7.1 Design of the indoor/outdoor detector's hardware

To determine whether the chair is indoors or outdoors, we use multiple sensing modalities to decrease the chance that an erroneous reading on one sensor type will cause an incorrect classification. The detector's inputs are light characteristics, temperature, and distance between the wheelchair and any ceiling or object above the chair. A photograph of the detector's hardware is shown in Figure 7-1.

Data acquisition and signal processing occur on a Dell Latitude notebook computer with a Pentium 166 MHz MMX processor. Data acquisition is accomplished through the use of a DAQCard-700 from National Instruments. The data acquisition board is installed in one of the PCMCIA slots of the notebook computer. The card has 8 differential analog inputs with analog-to-digital conversion,[2] 8 digital input

[2] We have configured the card for differential analog input. If the card is configured for single-ended analog input, there are 16 analog inputs.

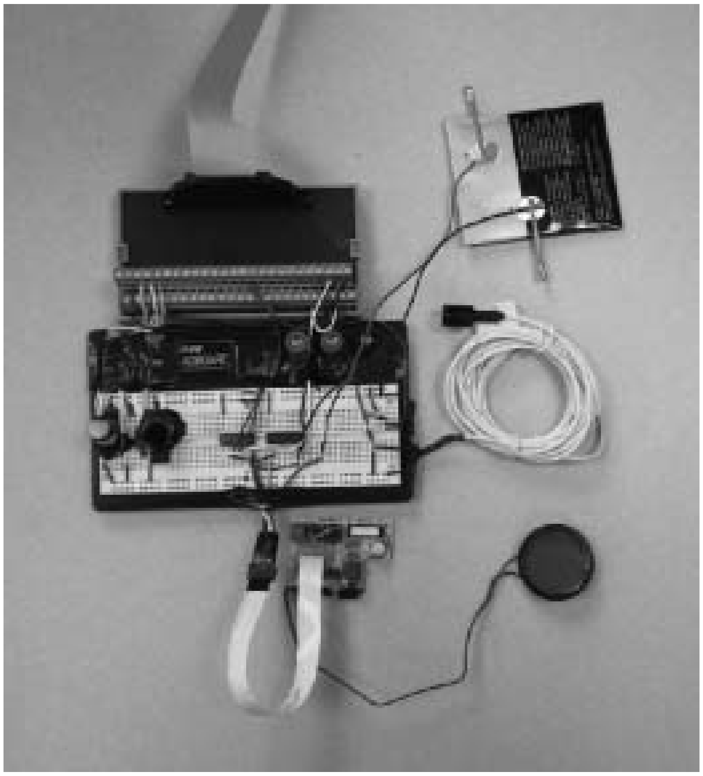


Figure 7-1: Picture of the indoor/outdoor detector hardware. On the left side of the breadboard is the circuitry for the light-to-voltage sensors. Two of the three sensors have filters over them. In the center of the breadboard is the circuitry for the Polaroid ultrasonic ranging module. The Polaroid ranging board is just under the breadboard and the ultrasonic transducer is to the right of the ranging board. In the upper right corner of the photo is the Polaroid Polapulse battery that is used to provide power to the ranging board. On the right of the breadboard is the circuitry for the thermistor probe. The probe is to the right of the breadboard. Above the breadboard is the connector block for the National Instruments DAQ-700 data acquisition card.

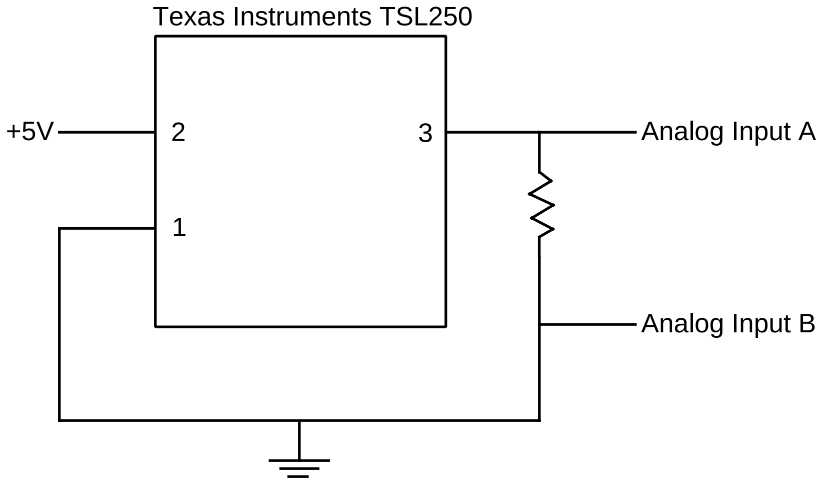


Figure 7-2: Schematic diagram of the light detection portion of the indoor/outdoor detector. The circuit for one photodiode is shown. Three were used in the detector: one was covered with an ultraviolet bandpass filter, one was covered with an infrared bandpass filter, and one was left unfiltered.

lines, 8 digital output lines, a 1 MHz clock, and three internal counters. The card also provides a 5 volt supply and a ground signal (both digital and analog). We used National Instruments' CB-50 connector block with a PR50-50F ribbon cable between the data acquisition card and the connector block. National Instruments provides its Ni-Daq software with its data acquisition cards. The software includes the drivers for the card and functions that can be called from Microsoft Visual C++.

For determining the characteristics of the light in the chair's environments, three Texas Instruments' TSL250 light-to-voltage optical sensors are used. The integrated circuit has a photodiode and a transimpedance amplifier. The output voltage is directly proportional to the light intensity on the photodiode. Figure 7-2 shows a diagram of the circuit used to detect light characteristics.

Outdoor environments usually contain more light in the ultraviolet spectrum than man-made light sources.[3] Fluorescent lighting emits little infrared radiation with incandescent lights emitting more. We would expect to find more infrared radiation in an outdoor environment than an indoor environment [Wyszecki and Stiles, 1982].

The photodiode is more responsive to visible light than ultraviolet and infrared light. Unfiltered visible light saturated the photodiode in both indoor and outdoor environments. In order to get useful voltage readings, filtering was required.

To filter the light, we used two 12 mm diameter bandpass filters sold by Edmund Optics: one for ultraviolet light and one for infrared light. The ultraviolet bandpass filter (U-360) passes 70% of ultraviolet light between 300 and 400 nm and a small amount (10%) of infrared radiation between 700 and 800 nm. The infrared bandpass filter (RT-830) passes up to 85% of light waves between 700 and 1200 nm, with the peak of the curve at 800 nm. The filters are mounted above two of the optical sensors. A third optical sensor was left unfiltered.

Depending on the season and the part of the world the chair is being used in, there are likely to be temperature differences between indoor and outdoor environments. Air temperature was measured using an ON-406-PP thermistor probe from Omega. The temperature sensor was one leg of a Wheatstone bridge. See Figure 7-3 for a diagram of the circuit used to measure temperature.

To sense the presence or absence of an object above the wheelchair, which would most likely be a ceiling, we used a Polaroid ultrasonic ranging system with a range of .5 to 35 feet[Pol, 1995]. The ranging board required two slight modifications for

[3] Sunlamps can generate higher levels of ultraviolet light, but would be uncommon in normal indoor environments.

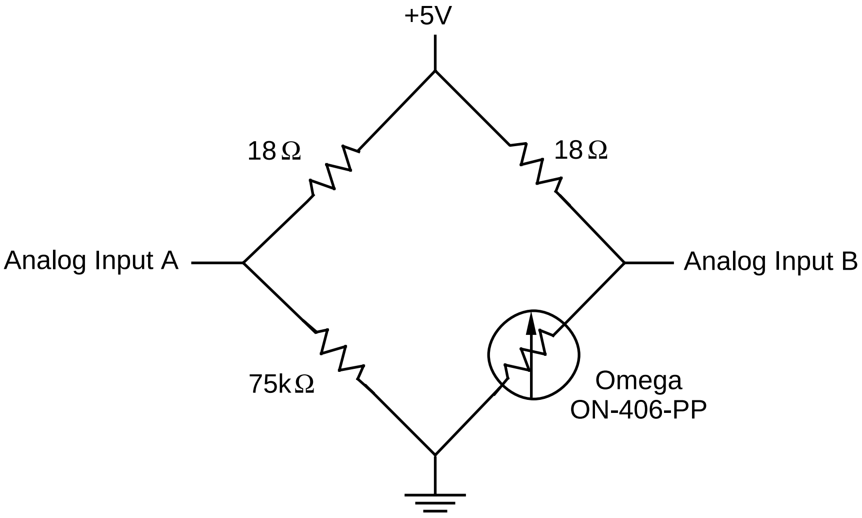


Figure 7-3: Schematic diagram of the temperature measurement portion of the indoor/outdoor detector.

our circuit. First, a 4.7K resistor was added between the processed echo line and V_{cc}. Second, BINH was grounded on the ultrasonic ranging board rather than on the detector's breadboard to prevent any possible interference. Power was provided to the Polaroid board with a 6 volt Polaroid Polapulse battery.

The DAQ-Card 700 was used to generate the INIT pulse for the ultrasonic ranging system and to count the elapsed time between sending the ping and hearing it return. See Figure 7-4 for a diagram of the ultrasonic ranging circuitry and software.

7.2 Learning the correct mode selection

Most people travel within a single environment during their everyday movements. Some people do travel long distances from home, but the norm will be one environment. To allow the system to adapt to a user's common environment, we have the system learn a decision tree in order to select the proper mapping of sensor values to

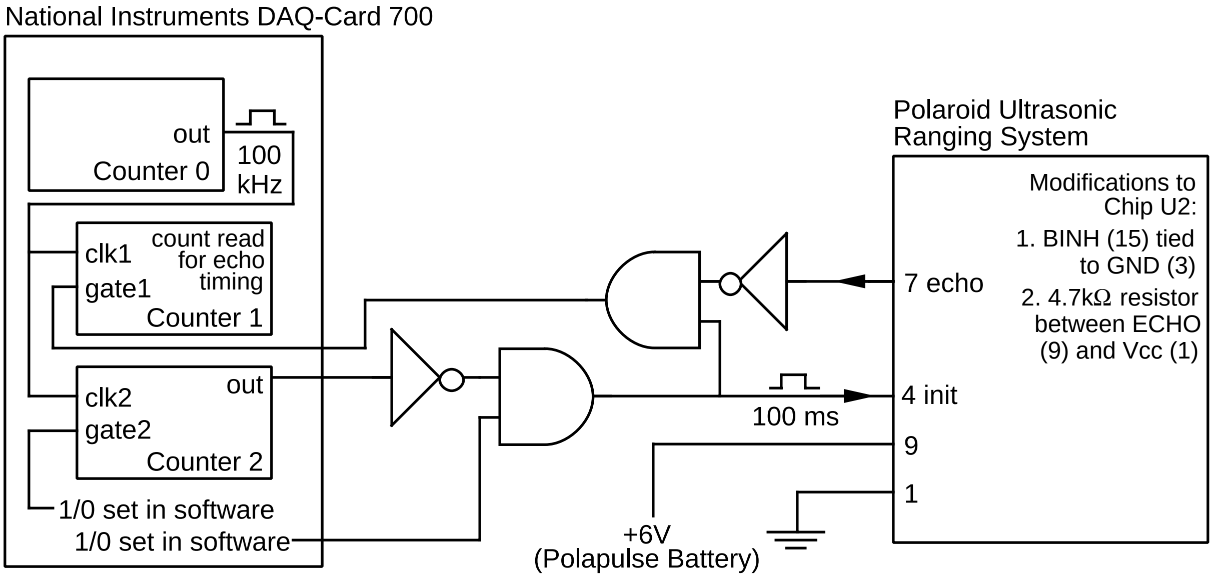


Figure 7-4: Schematic diagram of the ultrasonic ranging portion of the indoor/outdoor detector.

mode selection.

A decision tree is a succinct representation of a mapping from feature values to classes. Some examples of features and their values are: "temp=80", "height=2.5", and "uv=1.32". The class (or label) given to each feature vector is "indoor" or "outdoor". Each node in the tree represents a decision boundary. Each leaf returns a classification. The mapping from a feature vector to a class is obtained by following a path through the tree's decision nodes to a leaf. Decision nodes normally involve a single feature. For example, given the decision node "temp < 68", any feature vectors with a temperature value of less than 68 will follow the left child of the node, and any feature vectors with a temperature value of greater than (or equal to) 68 will follow the right child. A decision node "splits" the data. Geometrically, a decision tree divides the feature space into axis-parallel hyper-rectangles, with a class being associated with each hyper-rectangle.

It is possible to learn a decision tree from a set of training data which will hopefully generate correct classification on unseen data. Popular algorithms such as CART [Breiman *et al.*, 1984] and C4.5 [Quinlan, 1993] build a decision tree from the top down by greedily selecting the “best” feature to split on until only leaves remain. There are many ways of measuring which is the best feature, but in general they involve finding which split minimizes the entropy of the subsets of points to the left and right of the split. The full tree tends to overfit the data, so a pruning stage is usually performed to cut back some decisions that might not have been warranted by the data.

To train the decision tree, we collected data in indoor and outdoor environments. Indoor lighting conditions included offices and hallways with fluorescent lighting and a home with incandescent and energy saving lighting; outdoor lighting conditions included sunny and overcast days. Indoor ceiling conditions included ceilings of normal height and lobbies with very high ceilings; most outdoor conditions have nothing above the wheelchair, but occasionally there were buildings or trees overhanging the wheelchair. Indoor temperature conditions varied from the cold of a computer room to the warmth of an unventilated elevator lobby to an office on the sunny side of the building before the air conditioning was turned on; outdoor temperatures were in the range of temperatures found during spring in New England (40° to 85°). Sensor readings were stored every 5 seconds during data collection.

Two data sets were used for training. In the first, the outdoor temperature was usually lower than the indoor temperature, except for some situations such as unventilated lobbies. This smaller set had 352 indoor data vectors and 341 outdoor data vectors collected. The second, larger data set includes the data from the first set and adds data taken when the temperature indoors and outdoors was approximately the same and data taken when it was warmer outdoors than indoors. This set includes 547 indoor data vectors and 647 outdoor data vectors.

C4.5 was used to create the decision tree. Two thirds of the collected data was randomly selected for the training set, with the remaining third reserved for testing. A data vector is comprised of the temperature reading (*temp*), the reading from a light-to-voltage sensor covered by an ultraviolet bandpass filter (*uv*), the reading from a light-to-voltage sensor covered by an infrared bandpass filter (*ir*), the reading from an unfiltered light-to-voltage sensor (*light*), the value of an ultrasonic transducer pointing up (*sonar*) and the label for the vector (*indoor* or *outdoor*).

The default values of C4.5 are used when learning decision trees. For splitting, any test must have at least two outcomes with at least two cases under each outcome. The confidence level, CF, for pruning is 25%. With N training cases and E cases classified incorrectly by the leaf covering the N training cases, if the predicted error rate $U_{CF}(E, N)$ for the subtree exceeds the predicted error rate for replacement by a leaf, the subtree is pruned and replaced by the leaf.

Error rates for the decision trees given various attributes from the two data sets are summarized in Table 7.2. The error rates on the test set are smaller for the small data set mainly due to the fact that the indoor temperature was always higher than the outdoor temperature in that data set. Notice that the error is 2.8% using temperature as the only criterion for the small set while the error increases to 21.0% for temperature on the large data set. In the following discussion, we will only address

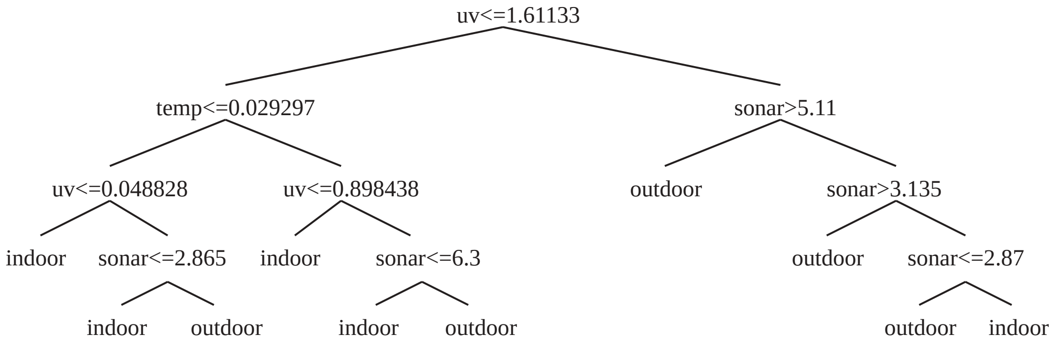


Figure 7-5: The decision tree built for the indoor/outdoor detector when given all five features in the large data set. If the node is true for a given data vector, the left branch is selected. This tree has an error rate of 1.7% on the test set.

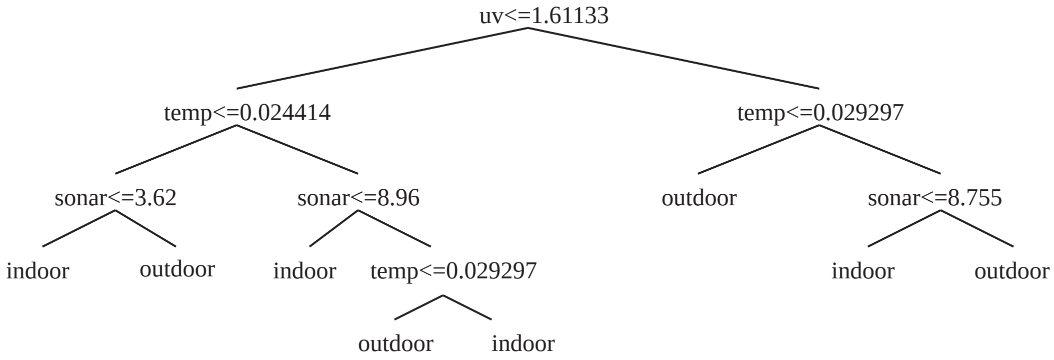


Figure 7-6: The decision tree built for the indoor/outdoor detector when given all five features in the small data set. If the node is true for a given data vector, the left branch is selected. This tree has an error rate of 0% on the test set, mostly due to the fact that the outdoor temperature was always lower than the indoor temperature in the small data set.

	Small data set	Large data set
All attributes	0.0%	1.7%
uv only	4.2%	3.2%
sonar only	10.7%	11.7%
temp only	2.8%	21.0%
sonar and *temp*	0.9%	7.9%
uv and *sonar*	4.2%	2.9%
temp and *uv*	2.3%	2.0%
uv, *sonar* and *temp*	0.0%	1.7%

Table 7.1: Error rates for decision trees learned for the indoor/outdoor detector. Error rates for the small data set are smaller since the temperature range in the small set was narrow enough to allow the decision tree to make a good split based upon temperature alone. The large data set included a wider range of indoor and outdoor temperatures.

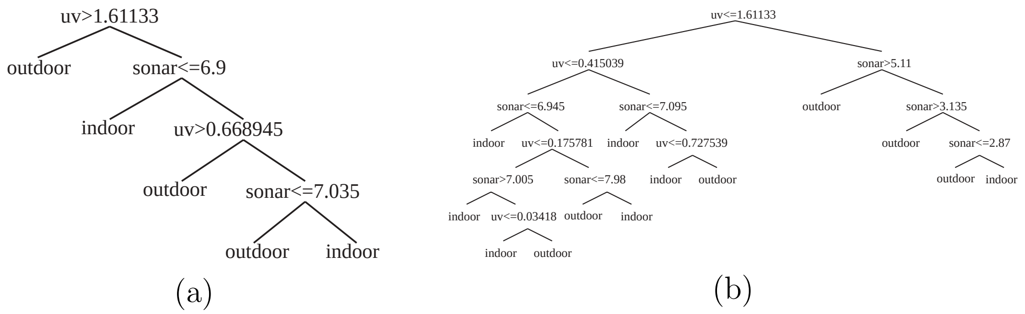


Figure 7-7: (a) The decision tree created using only uv and sonar as features from the small data set. This tree has a 4.2% error rate on the test set. (b) The decision tree created using only uv and sonar as features from the large data set. This tree has a 2.9% error rate on the test set.

the errors in the large data set, which includes all of the points from the small set.

Given all five features, C4.5 builds a tree that uses only *uv*, *temp* and *sonar*. The tree, shown in Figure 7-5, has a 1.7% error rate on the test set. For comparison, the tree built using the small data set is shown is Figure 7-6, which has a 0% error rate due to the clear split between indoor and outdoor temperatures.

If C4.5 is given *uv* as the only feature, a split at 1.61133 results in a 3.2% error on the test set. Classification failures occur when the wheelchair is under an overhang in an outdoor environment. Temporal smoothing will not be able to solve this problem if the wheelchair is under the overhang for a prolonged period of time (e.g., a long row of dense trees or Building 39 on the MIT Campus). If *uv* is combined with *sonar*, the resulting tree (Figure 7-8a) has a 2.9% error rate on the test set. If *uv* is combined with *temp*, the resulting tree (Figure 7-8b) has a 2.0% error rate on the test set. Given *uv*, *temp* and *sonar* as features, while ignoring *ir* and *light*, C4.5 builds the same tree given in Figure 7-5 with an error rate of 1.7% on the test set.

The use of multiple sensor types results in a more robust indoor/outdoor detection

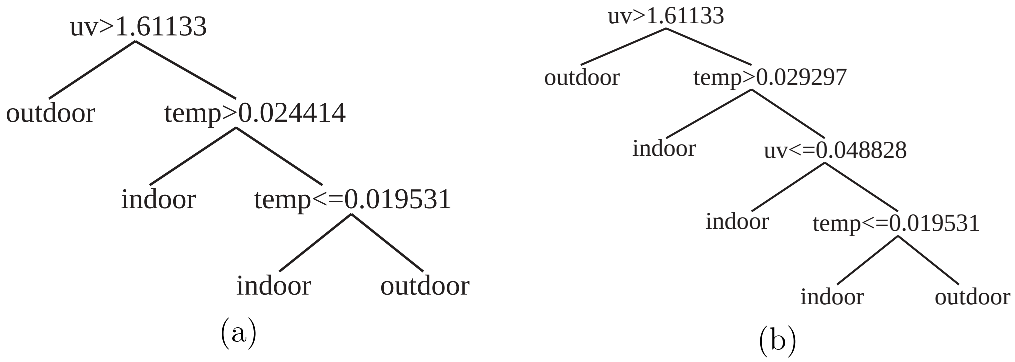


Figure 7-8: (a) The decision tree created using only uv and temp as features from the small data set. This tree has a 2.3% error rate on the test set. (b) The decision tree created using only uv and temp as features from the large data set. This tree has a 2.0% error rate on the test set.

than can be accomplished using only one sensor type. While the error rate of 3% for ultraviolet light alone might be considered small for other applications, the robotic wheelchair system's need for a rapid and smooth transition between navigation modes requires the addition of the ultrasonic transducer and thermistor to reduce the error rate.

Chapter 8

User Interface, Access Methods, and Evaluation by Physical Therapists

A robotic wheelchair system can not be only a navigation system. While it is important to develop a system that will keep its user from harm and assist in navigation, the system will be useless if it can not be adapted for the abilities of its intended users. A robotic system that can only be controlled by a joystick will be of no use to people unable to use a joystick. The system must be able to be adapted for many different access methods, which allow users to control a powered wheelchair in ways other than the standard joystick. These access methods include switches, sip-and-puff systems, voice control, retainer-like plates in the mouth, and eye tracking. The Wheelesley system solves the adaptation problem through the addition of a general user interface that can be customized for each user and their access method.

When a powered wheelchair is prescribed for a person, a professional wheelchair provider, such as a physical therapist, builds a wheelchair for its user by selecting the proper seat, access method and power train [Enders and Hall, 1990, Warren, 1990]. A seat type is selected and padding for the seat is created to fit the user properly. A headrest may be installed. An access method for controlling the wheelchair is selected to suit the user's abilities. Parameters are set for the wheelchair controller using a plug in pad. With all of this customization occurring, the addition of a customizable interface is not prohibitive for the system's installation and use. The customization required of the physical therapist (or other professional) would be to select the interface for the user's access method. Additional customization could come in the form of different screen layouts, which would be part of the standard Wheelesley system.

In this chapter, we describe access methods that are used to control powered wheelchairs. Then the design of the user interface is discussed. Finally, we present the customization of the user interface for two different access methods (single switch scanning and eye tracking).

8.1 Access methods

In the rehabilitation community, access methods (sometimes called input devices) are devices used to enable people to drive wheelchairs or control computers. Many different access methods for powered wheelchairs are currently used. The default access method is a joystick. If a user has sufficient control with a joystick, no additional assistance is necessary. These users would not be candidates for a robotic wheelchair since they are able to drive without assistance. If a person has some control of a joystick, but not very fine control, joystick movement can be limited through the addition of a plate which restricts the joystick to primary directions. Users in this group might be aided by a robotic system. If they push the joystick forward, the fine control could be taken over by the robotic system.

If a user is unable to use a joystick, there are other access devices which can be employed. A switch or group of switches can be used to control the wheelchair. If a user has the ability to use multiple switches, different switches can be linked to each navigation command. The multiple switches can be on the wheelchair tray, mounted around the user's head or placed anywhere that the user will be able to reliably hit them.

If the user has only one switch site, the wheelchair must be controlled using single switch scanning. In this mode, a panel of lights scans through four directional commands (forward, left, right and back). The user clicks the switch when the desired command is lit. If the user is traveling forward and drifts left, he must stop, wait for the light to scan to right, turn the chair to the right, stop, wait for the light to scan to forward, and then select forward again. This mode of driving is very slow and difficult; it is the method of last resort. Obviously, a robotic wheelchair system could help this group of users. Very few people are given single switch scanning as an access method with a standard powered wheelchair since it makes driving very difficult. Adaptive Switch Laboratories sells 50-100 single switch scanning systems each year[1][Ashmore, 2000].

[1] ASL sells a total of 1500 systems per year for the "seriously involved": people who would have

Another access method for wheelchairs is a sip and puff system. With this method, the user controls the wheelchair with blowing or sucking on a tube. If the user can control the air well enough, soft and hard sips or puffs can be linked to control commands. This is analogous to the multiple switch system above.

The access methods described above are currently in use in the rehabilitation community. In addition to these methods, researchers are developing new access methods. One such method is an ultrasonic head controller [Ford and Sheredos, 1995]. Two ultrasonic sensors are placed on the left and right side of the user's head; the user moves his head to control the direction of the wheelchair. Another novel access method is eye tracking. In Section 8.2.1, we discuss the use of an eye tracking system called EagleEyes as an access method.

Most research on robotic wheelchairs has not focused on the issue of access methods. Many of the current systems are driven using a joystick (e.g., [Scott and Munro, 1985], [Miller and Slack, 1995], [Yoder *et al.*, 1996], [Gomi and Griffith, 1998], [Simpson *et al.*, 1999] and [Tahboub and Asada, 1999]). While some users in the target community can use joysticks, there are many others who can not.

Voice control can be used an access method; however, problems can arise if a failure to recognize a voice command could cause the user to be unable to travel safely. Simpson and Levine [Simpson and Levine, 1997, Simpson *et al.*, 1998] studied voice control as an access method for the NavChair system, using a commercial voice recognition system. Navigation assistance provided by NavChair would prevent the user from hitting obstacles due to an incorrectly interpreted voice command. The

to be pushed in a manual wheelchair if not for their ASL system [Ashmore, 2000].

Vocomotion system [Amori, 1992] was designed for a non-robotic powered wheelchair addresses concerns about safe driving using voice control. Using a commercial voice recognition system, Vocomotion converts a speech command to its equivalent joystick output. The system offers the user several different ways to give the same command and requires only that a user's speech be consistent rather than be clearly understood by other people. Any unrecognized command will halt the wheelchair and the wheelchair stops automatically if no new command is given in a predetermined amount of time. All of these safety features make driving with voice control more feasible, especially if they were paired with a robotic wheelchair. However, while voice control could be useful for some people, some members of our target community are non-verbal. Other systems using voice control are PSUBOT [Stanton *et al.*, 1991] and the UMIDAM project [Mazo *et al.*, 1995].

The work in this thesis investigated eye control as a novel control method (Section 8.2.1) and single switch scanning (Section 8.2.2). Single switch scanning has also been investigated on the OMNI wheelchair with a row/column scanning mode for their user interface [Bühler *et al.*, 1997] and the VAHM project [Bourhis and Pino, 1996].

8.2 Graphical user interface

The graphical user interface is built on a Macintosh Powerbook using Hypercard and can be easily customized for various access methods (see Section 8.1 for a discussion of access methods). In this prototype, Hypercard was selected for its ability to quickly

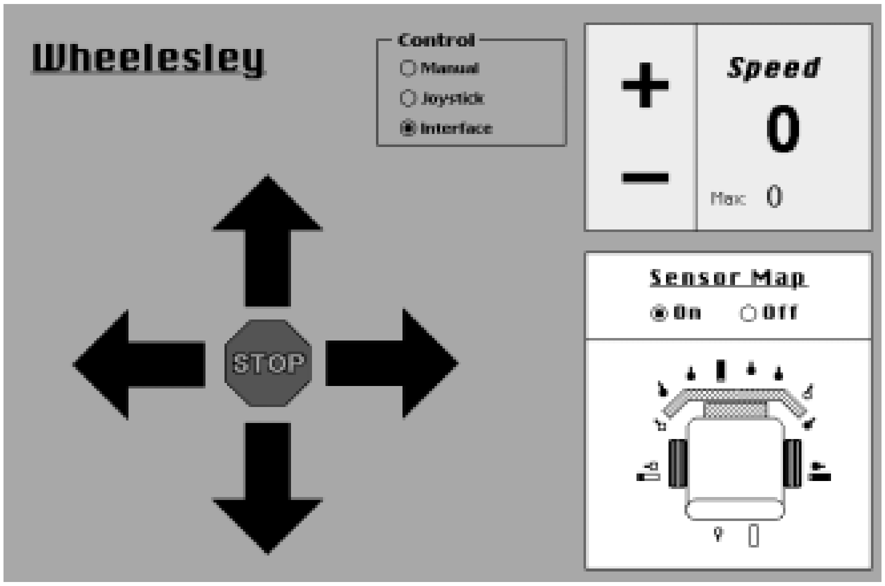


Figure 8-1: Screen shot of the full user interface.

and easily update the graphic look of the interface. This allowed for many different interfaces to be designed and tested, both in initial development and during the adaption for various access methods. Commands given to the user interface are passed to the control program running on the robot through the Macintosh Powerbook's serial port.

To demonstrate the ease of adaptability, the interface has been customized for two access methods. The first is an eye tracking device called EagleEyes [Gips, 1998] (Section 8.2.1). The second is a single switch scanning device (Section 8.2.2).

The full user interface is shown in Figure 8-1. (See Appendix A for a description of an earlier version of the interface.) The full interface is too detailed for use with most access methods; customization for each access method simplifies the interface appropriately to allow use with the the access device.

There are three control modes that the user can select via the user interface:

manual, joystick, and interface (see the top center of Figure 8-1). In manual mode, the joystick commands are passed directly to the motor controller with no sensor mediation. This allows the system to act like a standard powered wheelchair. In joystick mode, the user's joystick commands are carried out using low-level control on the robot to avoid obstacles. The navigation arrows on the user interface are hidden since they are not being used to control the robot in joystick mode. This assistive navigation mode would be most useful for people who have some control of the joystick, but not very fine control. A user in this group may have a plate installed on his joystick to prevent him from moving the joystick in any direction other than a primary one. In interface mode, the arrows on the interface screen are used to direct the robot. The navigation command portion of the interface used in interface control consists of four directional arrows and a stop button. All customization of the user interface for access methods (other than a joystick) selects the interface mode.

The standard speed of the robot is controlled by clicking on the plus and minus buttons in the upper right corner of the screen. The robot may move at a slower pace than the standard speed when the current task requires a slower speed to be carried out safely. The actual speed of the robot is displayed by the robot under the speed control buttons.

The sensor map shows a representation of the wheelchair and the location of the sensors. Obstacles detected by the sensors are displayed on this sensor map. This is intended to provide a user who is unable to move his head with a picture of the obstacles in the world around him. In Figure 8-1, the sensor map is shown in the lower right corner. The rectangular bars represent the sonar sensors. The bar fills proportionally to indicate the distance of obstacles, filling more as obstacles get closer. An empty rectangle indicates that no object has been detected. The infrared sensors are represented by circles with a line pointed out in the sensor's direction of detection. For these binary detectors, an empty circle indicates that no obstacle has been detected and a full circle indicates that an obstacle has been detected.

At its most basic level, the user interface consists of directional arrows to move the wheelchair and a method for stopping, which may be a stop button. We have found that simpler interfaces are more appropriate for use with our target user communities. For example, with a single switch scanning system, each button of the user interface must be highlighted in sequence, pausing while highlighted to allow the user to select that button if desired. When there are simply four direction buttons, the user need not wait too long if they miss the desired command on the first scan. However, if additional capabilities and choices are added to the interface, each scan will take longer. Even more importantly, each button is afforded the same level of importance. However, movement commands are of more importance than a mode selection which might only be issued when the chair is powered on.

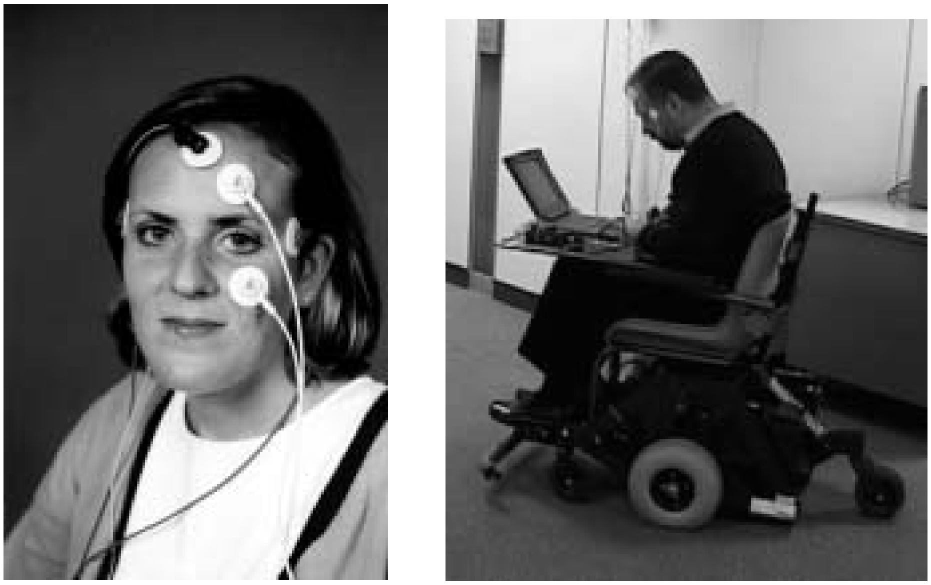
When simplifying the user interface, the other capabilities are usually hidden from the user in the main screen. However, they can be brought to the foreground by pressing the shift key, allowing for the control mode, speed, and other modes to be selected from the user interface. In practice, the control mode would remain the same from day to day. The user's normal speed preference would also usually be static. In fact, most of these selections for standard powered wheelchairs are made by a physical therapist using a programming module and are not changed later by the user. Our interface would allow a physical therapist to work with the wheelchair's user to find the proper settings, just as is currently done with setting for a standard powered wheelchair's controller.

8.2.1 Customizing for EagleEyes

We have investigated eye tracking as a novel method for controlling a wheelchair. EagleEyes [Gips *et al.*, 1996, Gips, 1998] is a technology that allows a person to control a computer through five electrodes placed on the head (Figure 8-2a). Electrodes are placed above and below one eye and to the left and right of the eyes. A fifth electrode is placed on the user's forehead or ear to serve as a ground. The electrodes measure the EOG (electro-oculographic potential), which corresponds to the angle of the eyes in the head. The leads from these electrodes are connected to two differential electrophysiological amplifiers. The amplifier outputs are connected to a signal acquisition system for the Macintosh.

Custom software interprets the two signals and translates them into cursor coordinates on the computer screen. The software can be set to interpret the user's movement two different ways: the user moves his eyes while keeping his head steady or the user moves his head while keeping his eyes steady. This is set according to the preference and natural ability of the user.

The difference between the voltages of the electrodes above and below the eye is used to control the vertical position of the cursor. The voltage difference of the electrodes to the left and right of the eyes controls the horizontal position of the



(a) (b)

Figure 8-2: A novel access method for a powered wheelchair is EagleEyes, an eye tracking device. (a) Placement of the electrodes for EagleEyes. The electrodes above and below the left eye are used to calculate the y coordinate and the electrodes on the temples are used to calculate the x coordinate. The electrode on the forehead is used for ground. (b) The robotic wheelchair system being driven using EagleEyes.

cursor. If the user holds the cursor in a small region for a short period of time, the software issues a mouse click.

The user interface was quickly customized for use with EagleEyes [Yanco and Gips, 1997]. The screen (Figure 8-3) was redesigned to accommodate the needs of the EagleEyes system. Large buttons are easier to use with an electrode system than small ones. The interface has four large direction arrows and four large stop buttons. Four stop buttons are provided so that the user will be near a stop button regardless of the current cursor position. To move, the user moves the cursor to the appropriate arrow through eye and head movement and dwells on the arrow to issue a mouse click. The robot travels in the commanded direction, avoiding obstacles and staying

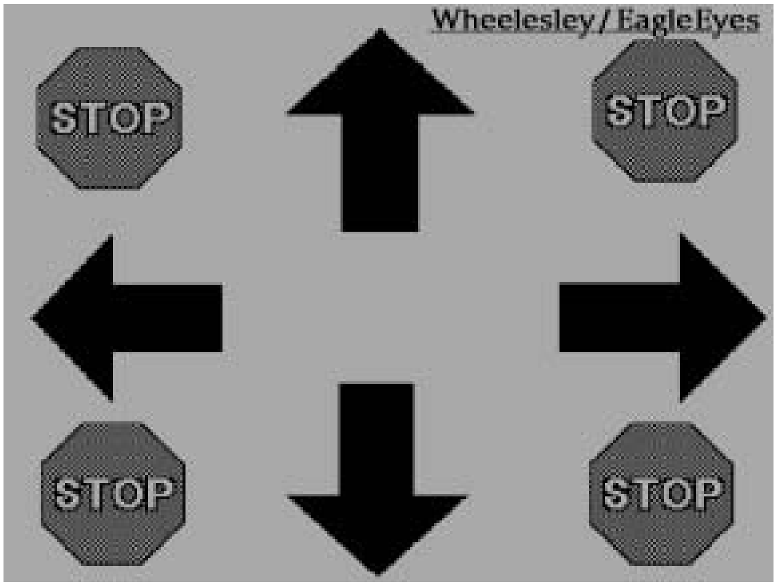


Figure 8-3: The customized interface for use with EagleEyes. Four stop buttons are provided so the user can look to any corner to stop. Layout of the arrows and stop buttons can be quickly changed to accommodate a user's abilities and preferences.

centered in the hallway, until a new directional command or a stop command is issued.

The robotic wheelchair has been successfully controlled by three able-bodied subjects using EagleEyes. (See Figure 8-2b for a photo of the two systems being used together.)

There is currently a “Midas Touch”-like problem with this access method; there is no way for the computer to differentiate between the cursor moving because the user wants to issue a command and the cursor moving because the user is looking around the environment. The able-bodied subjects solved this problem by fixing their gaze either on the arrow for the current direction or on part of the unused portion of the screen. Other users may not be as proficient with EagleEyes and might look at other command buttons accidentally. This problem could be solved by using a voluntary blink as a mouse click or by using a voluntary blink to switch in and out of using

EagleEyes to control the wheelchair. A voluntary blink can be measured since it lasts longer than an involuntary blink. For users able to hit a switch reliably, a switch could be used to toggle the modes or could be used as a mouse click.

With an eye tracker as a control method, an experienced user may not need to have a computer screen in front of him on a tray. Once the user learned how to issue commands on the screen, the user could move his head and eyes in a similar manner to issue commands with the screen removed. This would make the robotic wheelchair look more like a standard wheelchair, which is desired by many potential users.

8.2.2 Customizing for single switch scanning

Single switch scanning is the access method of last resort for standard powered wheelchairs. A single switch scanning system consists of a switch and a control panel with four lights for four directions (forward, left, right and back); see figure 8-4 for a photograph of a commercially available single switch scanning system. When using this method, the control panel scans through the four commands. The user clicks the single switch when the control panel shows the desired direction. Usually, these systems are not “latched” for forward. This means that person must hold down the switch as long as he wishes to go forward. Latching the system would mean the wheelchair would start going forward when the switch was pressed and would continue going forward until the switch is pressed again. This is considered too dangerous for a standard powered wheelchair configuration since the wheelchair would continue to drive if the user was unable to press the switch to stop it.

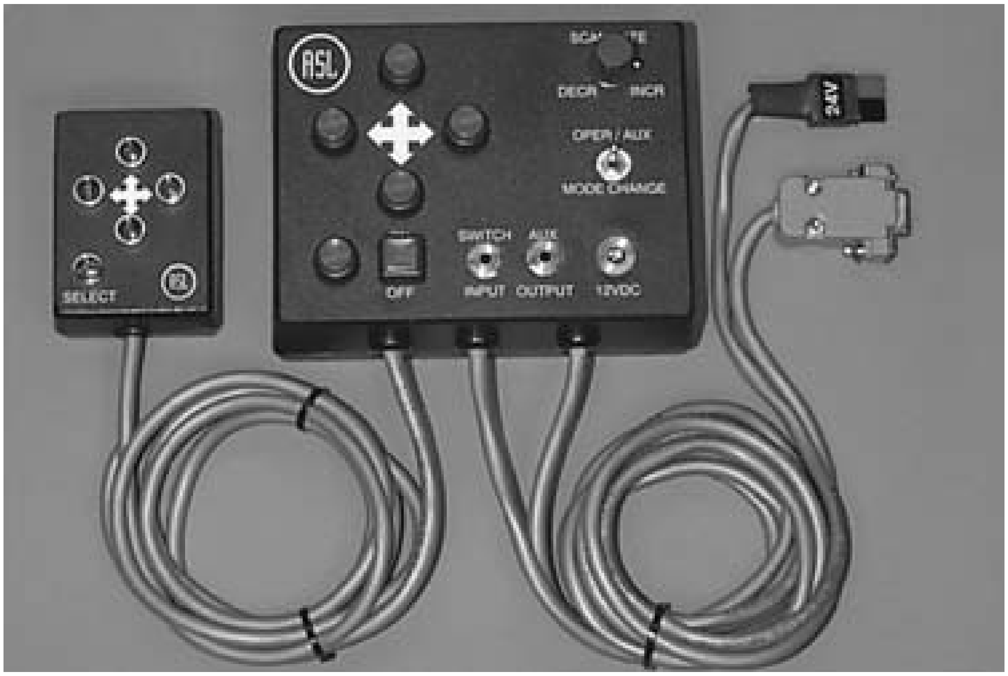


Figure 8-4: A commercially available single switch scanning interface. Photo courtesy Adaptive Switch Laboratories.

This method is very difficult to use for standard powered wheelchairs, primarily because drift is a significant problem. To correct a drift to the left of the right, the user must stop going forward, wait for the scanning device to get to the arrow for the direction of choice, click to turn the chair, stop turning, wait to scan to forward and then click to move forward again. Robotic assisted control can improve driving under this access method by correcting drift automatically and avoiding obstacles. Additionally, the system can be latched due to the safety provided by robotic control.

Customization for this access method took less than 1 hour. The screen has four arrows and one stop button (see Figure 8-5). The system scans in the same pattern as commercially available single switch scanning systems (forward, right, left, back). The stop button is only on the screen so that it can be highlighted when the chair is stopped; the system does not scan to that button. The chair can be stopped at any

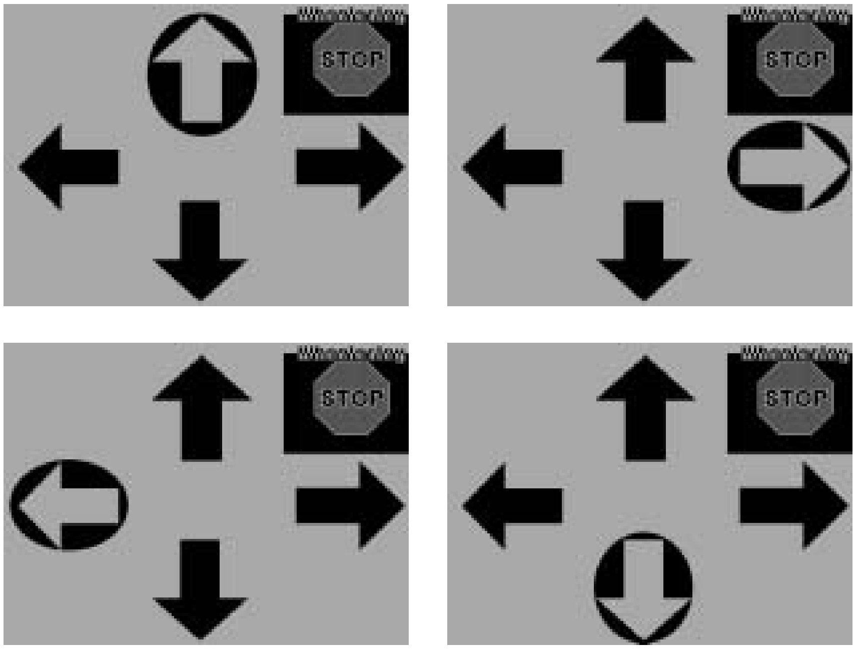


Figure 8-5: The customized interface for single switch scanning. The interface scans through the four directions in the following order: forward, right, left and back. To start moving, the user clicks a switch when the interface is highlighting the arrow corresponding to the desired direction. To stop moving once a command has been issued, the user clicks the switch. Scanning resumes on the forward arrow once the chair stops.

time by hitting the switch while the chair is moving. After the chair is stopped, the system resumes scanning on the forward arrow, since forward is the most likely next command. Left and right are the two next most likely commands. Empirically, back commands are rarely issued. See Figure 8-6 for a picture of the system being driven with single switch scanning.

User tests with fifteen able-bodied subjects determined that an obstacle course can be completed in less time and with less effort with navigation assistance than without[Yanco and Gips, 1998]. Users traversed the obstacle course in 25% less time with 71% fewer clicks with robotic assisted control. (See Section 5.3 for details.)

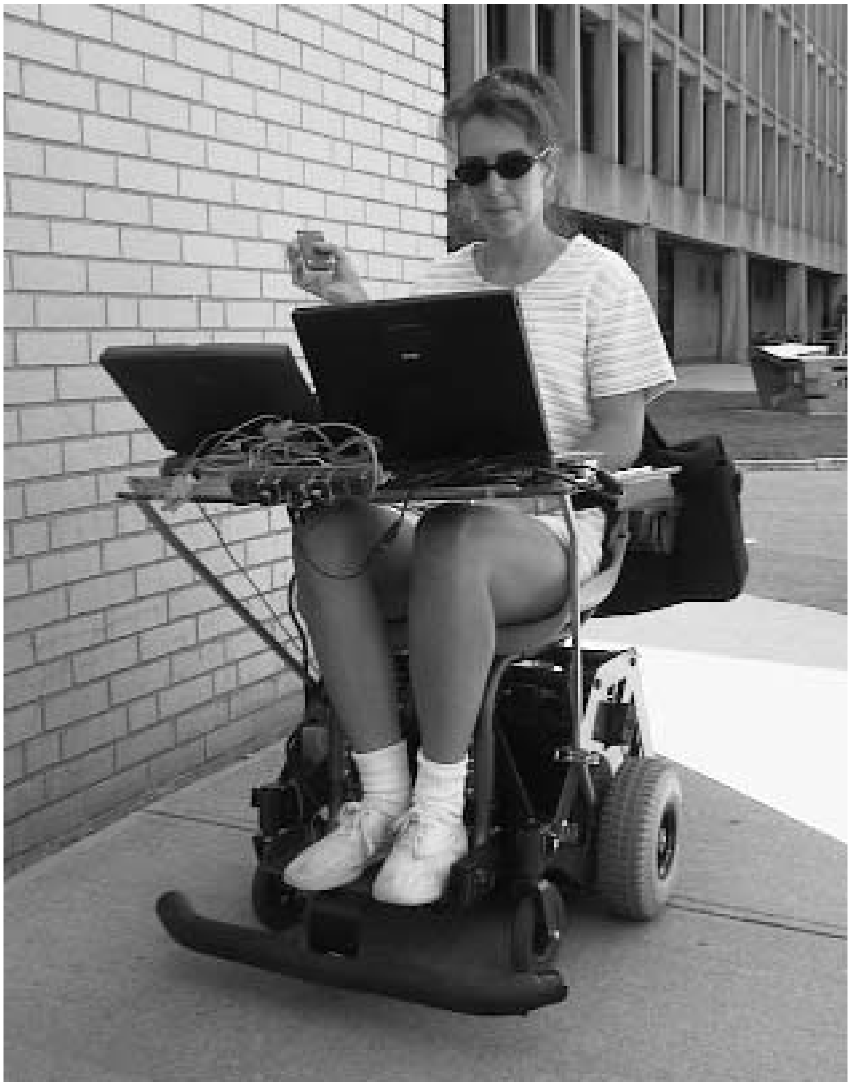


Figure 8-6: The robotic wheelchair system being driven with single switch scanning.

8.3 System evaluation by Physical Therapists

The robotic wheelchair system was brought to the Spaulding Rehabilitation Hospital in Boston in July 2000 for feedback from physical therapists who are experienced with providing wheelchair systems and teaching people to use them. Twelve physical therapists attended the session.

After a presentation about the system and a question period, the therapists tried the wheelchair indoors.[2] During the demo period, the chair was first run without a user to allow the physical therapists to get a feel for the behavior of the chair. They crowded the chair to see how it would behave, and it turned to avoid all of the feet, finding an opening between the groups of people.

Questionnaires were distributed at the presentation. (See Appendix B for a copy of the questionnaire.) In addition to the answers provided on the questionnaires, the following sections also include observations made by the physical therapists during the presentation and demonstration.

The physical therapists seemed very interested in the technology and thought well enough of it to offer their patients as test subjects for future tests. One therapist wrote that the system is a "promising tool in decreasing work of driving for those with very limited access."

[2] On the day of the presentation, it rained several inches in Boston, eliminating the possibility of outdoor runs. However, the outdoor system was later evaluated by one physical therapist who traveled to MIT to test the outdoor navigation.

8.3.1 Target patient population

The physical therapists indicated that this type of system would be useful for many groups of patients, including those with the following:

- Initial onset of Guillian-Barré syndrome
- Multiple sclerosis
- Cerebral palsy
- Spinal cord injury
- Brain injury

The patients who would benefit have “limited use of available systems due to poor endurance issues and subtle attention defects or those who lack fine motor control for joystick access” while “eye gaze control allows access for those with little to no head control.” People with spinal cord injury or brain injury with fine motor defects would have some of the work of driving taken over by the system’s small corrections made to maintain a safe path.

In addition to being an everyday system for some users, the system could be useful for training people with brain injury who have decreased attention. The system would decrease the likelihood of impact with obstacles during training. One therapist wrote that there are “benefits [to using the robotic wheelchair system] as a training device: no gouged walls and a decreased risk of injury to bystanders as a patient is learning to operate a power wheelchair.”

8.3.2 Sensor-guided driving for reverse

Driving backwards occurs much more commonly in daily use of a wheelchair than we (and, judging from other research published in this field, every other robotic wheelchair researcher) thought. In our indoor user tests (14 people, about 14 hours of runs), a backwards command was issued only once. However, the physical therapists said that their users are pulling up to sinks and desks, needing to back out of tight spaces, backing into vans and busses, and sometimes even backing up once in a crosswalk if the light changes.

In the current version of the wheelchair system, it would be difficult to address these concerns. Additional sensors would need to be added to the back of the wheelchair. Currently, there is one infrared sensor and one ultrasonic sensor. To allow for backing up to behave as well as driving forward, the number of sensors on the rear of the chair would need to be similar to the number on the front of the wheelchair (seven infrared and one sonar). The outdoor system would require an additional stereo head pointing backwards. A panning head might be used, but it probably would not be able to turn quickly enough when the chair switched from backward driving to forward driving.

8.3.3 Mounting of sensors and vision system

The group thought the infrared and ultrasonic sensors were mounted well, in that they didn't really show much. However, they had two issues with the mounting of the vision system on the wheelchair's tray. First, the camera system extends past the front bumper, making the bumper fairly useless. The mount could be changed to go under the tray, keeping it within the bumper's footprint. However, this remount would conflict with their second comment. They said that many of their users would not want a lap tray. So an alternative mount for the vision system would need to be found, perhaps on a swing out bar around knee height. They also asked if the computer would be necessary in the final version, which seemed to be a problem due mostly to its placement on the lap tray.

8.3.4 The powered wheelchair

If the system is to be tested with disabled people in the future, the system should be implemented on an Invacare wheelchair, which has a tiltable seat and other features needed for their users. The disabled user would just transfer his seat cushion to the Wheelesley prototype. The physical therapists offered to participate in user tests, if a new system is built on an Invacare chair.

One of the therapists noted on the questionnaire that the system would need to be able to be added to any power wheelchair. In addition to the Invacare system mentioned above, there are only a few other major manufacturers of powered wheelchairs. For research purposes, implementation on one major platform should be enough to gain a patient population for user tests.

8.3.5 System demo

At first, no one was willing to ride in the chair. So it was first run with no user riding it through the halls, with people clicking the switch for the interface as they walked near the chair. After this initial demo, several people wanted to try. With and without the rider, the PTs really pushed the system, standing close individually and in groups, trying to crowd the wheelchair to see its behavior.

The chair did very well in these very crowded situations, avoiding people and finding an opening. It went down to the end of a corridor and maneuvered its way back by turning and backing up until it had turned 180 degrees. One problem during the demo was that without a rider, the chair turned more quickly with no weight in the chair and hit a doorway opening due to this overturning; since the system was designed to drive with a user, code was never written to account for the lack of a user's weight. The vision system mount also scraped the wall a couple of times since it overhangs the bumper and even extends beyond the sensing range of the infrared sensors.

8.3.6 Interface and access methods

During the presentation, the physical therapists asked about various switch arrays that could be used. While only single switch scanning had been implemented, it would be trivial to map additional switches to particular commands.

They asked if the system could work with a joystick as the interface. They said that some users can operate a joystick, but have cognitive impairments preventing them from driving without hitting things. These users could be aided by the addition of obstacle avoidance while using the joystick.

The system can be controlled with a joystick, but this was never the main focus of the research. To drive with a joystick, the joystick readings are mapped to the nearest driving command, effectively with an X cutting across the joystick, mapping -45 degrees to 45 degrees to forward, 45 degrees to 135 degrees to right, 135 degrees to 225 degrees to backward, and 225 degrees to 315 degrees to left. However, to deliver the system for the user group described above, the the joystick code should be rewritten to allow for more fine control (if the user has it – some users have a plate installed on the joystick that only allows them to push in one of four directions).

The group liked the idea that a person could drive with an eye tracker, especially with the removal of the computer once the person learned where to look to issue commands. However, they were concerned about the cosmetic issues of electrodes on the user's head. A camera based eye tracker could solve the electrode problem, but would add computers and a camera, most likely requiring the lap tray that they'd like to see removed.

On the questionnaires, the physical therapists indicated that their patients are likely to use single switch scanning, multiple switch arrays, sip and puff, and a joystick as access methods for a robotic wheelchair system.

Chapter 9

Conclusions

This research project resulted in an assistive robotic wheelchair system that can potentially be integrated into the everyday life of disabled people who are currently unable to drive a standard powered wheelchair. The physical therapists who evaluated the system were very excited by the possibility of such a system, even offering their patients for future user tests. They saw a potential for the system being used as a training device for new drivers as well as a regular system for some people.

One potential barrier to introduction is cost despite the fact that we had used many off the shelf components. In our development system, we had two notebook computers running parts of the system in addition to the robot's processor; obviously, we could reduce this to one computer in a deliverable system. In fact, it is likely that given the continuation of speed increases for processors, we could reduce the system to use only an onboard processor, eliminating any laptop computers. This would also allow the wheelchair's tray to be eliminated, since the physical therapists pointed out that many users would not want to have a tray.

However, cost may be not as much of a problem as one would think. Any child under 21 or still in school can get a $20K system very easily due to equal opportunity laws. However, it is more difficult for the elderly to get such systems due to Medicare [Ashmore, 2000]. Most powered wheelchairs are purchased by third parties such as insurance companies and the government [Cooper, 1999].

Before the system were tested on disabled users, we would want to modify the system to account for the comments of the physical therapist and our observations during the development phase. The system would need to include a suite of sensors and cameras pointing backwards to match the sensors devoted to forward travel. Our system was developed with the belief that most travel would occur in the forward direction; we learned from the physical therapists that actual wheelchair users back up frequently when pulling away from desks and sinks, getting into a van, or moving in tight spaces.

The vision system would need to be upgraded from gray scale cameras to color cameras. Color processing would allow us to eliminate shadows. The updated cameras would also need to have automatic gain control (AGC) to allow the system to function on very sunny days. The current cameras often saturate in very bright sunlight.

Using the Hough transform instead of fitting lines using least absolute deviation would probably increase the robustness of the vision system. Navlab [Thorpe *et al.*, 1988] classifies points as road or non-road using a color classification algorithm, then uses the Hough transform to find the most likely vanishing point and orientation for a road. In our system, we have detected edge points that we are trying to fit to a sidewalk boundary. Using the Hough transform, each edge point would vote for the possible lines that it could belong to. The line with the highest score would be selected as the correct sidewalk boundary. Lane boundaries of a road have been detected from edge points using the Hough transform by [Yu and Jain, 1997].

Additional error checking should be added before the system is delivered to a user. In the current outdoor system, forward driving is guided with the vision system. Since the right and left turns issued through the interface are turns in place, we have assumed that the user will be turning from a safe position. However, this assumption should be replaced by sonar sensors pointing down to detect drop offs to avoid turning when very close to the curb.

The vision system is confused by guard rails along the side of a ramp. The edges of the rails are usually much stronger edges than the edge of the ramp. Additionally, the vertical poles supporting the railings cause the computed line to skew upwards, away from the actual line forming the edge of the ramp. Adding a module to look for guard rails using a combination of the Hough transform and stereo vision would allow us to find the railings and then to use them for navigation.

Other potential areas for improving the system include adding maps for frequently traveled locations to further reduce the effort required to travel in the home or office. Additional feedback could be provided by the system to inform the user of its current intentions. This feedback could be scaled back as the user started to trust the system over time.

Despite the need for continuing work before the system could be used, this research resulted in many steps towards the goal of a deliverable system.

- The robotic wheelchair system provides assistive navigation in novel indoor and outdoor environments by following the local path (hallway or sidewalk) and avoiding obstacles. User tests show that driving with robotic assistance requires 77% less effort in indoor tests and 88% less effort in outdoor tests. People can arrive more quickly at their destinations without needing to wait to scan to a large number of commands; test subjects completed the indoor course in 22% less time and the outdoor course in 25% less time.

- We developed a customizable user interface that is intended to be used with a wide array of interface devices. We have demonstrated its customizing for single switch scanning and eye tracking in this work.

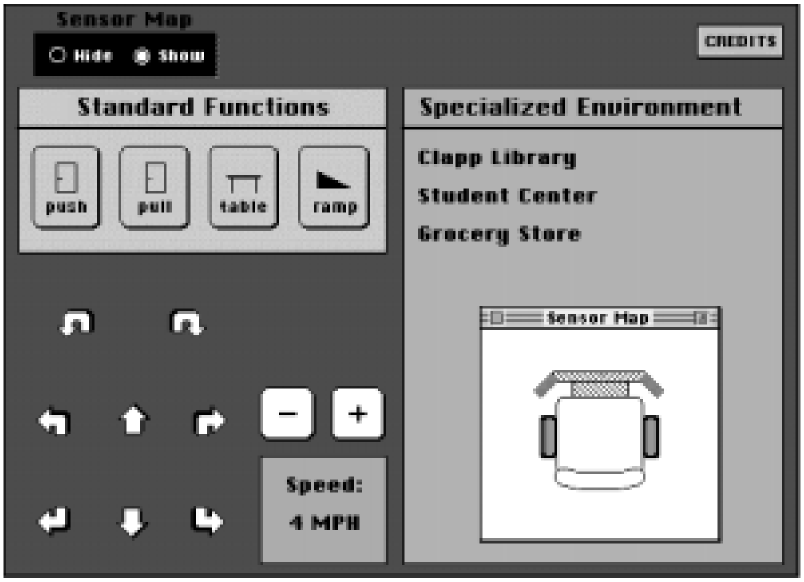
- The indoor and outdoor navigation modules can be selected automatically using an indoor/outdoor detector that we developed. Using the temperature, the measurement between the wheelchair and the nearest overhanging object, and the amount of ultraviolet light present, we can correctly classify the environment 98.3% of the time.

Appendix A

Preliminary Design of the User Interface

In the initial stages of the Wheelesley project, a graphical user interface for controlling the robot was designed [Yanco *et al.*, 1995]. This preliminary interface was designed to include all of the perceived needs the system's user might have. However, the interface was designed before the robot's navigation system was fully developed and many of the features were found to be impractical. Additionally, the preliminary interface described below was too complex to be used with the access methods of our users. We present the preliminary design here to document the full design process and to discuss reasons changes were made to the original design.

In this preliminary design, we envisioned a point-and-click user interface. The primary user interface screen is shown in Figure A-1. The fine motor control required to use this particular interface would most likely indicate that the user would not require the Wheelesley system. The final user interface described in Chapter 8 can also be used as a point-and-click interface, but the simpler design allows for easy



Sensor Map ○ Hide ● Show

Speed: 4 MPH

Figure A-1: Screen shot of the preliminary graphical user interface's main screen.

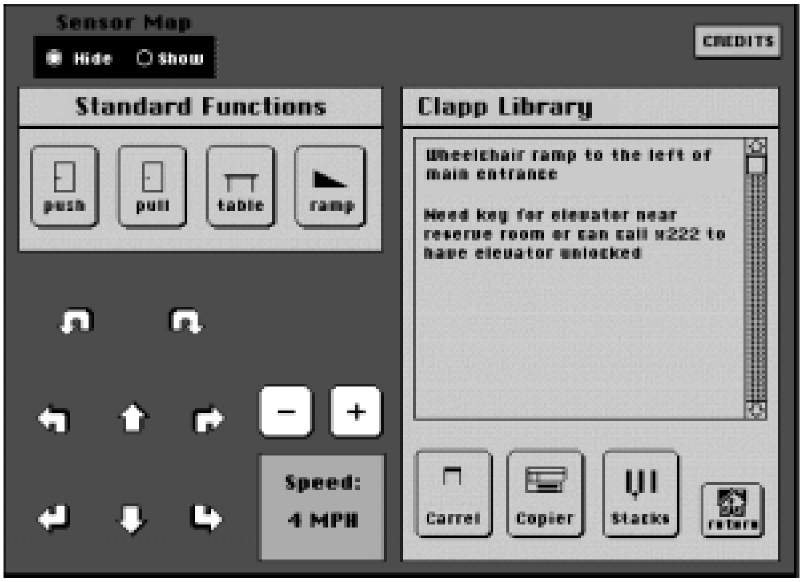
customization for users requiring access methods other than a joystick or mouse.

For movement control, six arrow buttons were included on the lower left of the interface screen, corresponding to forward, forward to the right, backward to the right, backward, backward to the left, and forward to the left. There were also two buttons that allow the user to turn in place to the left or to the right; these buttons were directly above the movement buttons. During the development of the navigation system, we realized that the system did not require all of these driving possibilities. It was enough to give the user four directional arrows and the system would take care of the fine navigation control required to avoid obstacles. Right and left turns are done in place; this eliminated the need for buttons allowing the user to turn in place.

As with the current interface, the user could control the standard speed of the robot by clicking on the plus and minus buttons to the right of the direction buttons. The robot may move at a slower pace than the user requests when the current task requires a slower speed in order for the task to be carried out safely. The actual speed of the robot is displayed by the robot under the speed control buttons. In our customized interfaces for eye tracking and single switch scanning, we chose to eliminate the speed control from the screen. As discussed in Chapter 8, the maximum speed of a standard powered wheelchair is usually set by a professional wheelchair provider, as it could be for the Wheelesley system.

Above the movement buttons were the "standard function" buttons. The tasks envisioned included moving through doors (the task is broken down to doors that open away from the wheelchair and doors that open toward the wheelchair), pulling up to a table, and moving on ramps. As development on the navigation system progressed, we realized that while the robot might break these tasks down in order to navigate properly, we should not require the user to differentiate these possibilities for us and eliminated this from the design of our interface. The system should simplify the driving task, not complicate it.

The user could choose to hide or to show the sensor map. The sensor map shows a representation of the wheelchair. Obstacles detected by the sensors would be displayed on this sensor map. This is intended to provide a user who is unable to move his head with a picture of the obstacles in the world around him. In Figure A-1, the sensor map is shown but no obstacles have been detected by the robot. The sensor map is also included in our final user interface. It requires no interaction from the user and thus does not add to the overhead of using the system. In both of our customized interfaces, we eliminated the sensor map for simplicity. Our users for both customized interfaces were able bodied and were able to see the environment around them. If this



Wheelchair ramp to the left of main entrance

Need key for elevator near reserve room or can call x222 to have elevator unlocked

Figure A-2: Screen shot of the specialized environment screen for Clapp Library for the preliminary user interface.

feature were desired by a user, it would be very easy to include it on a customized screen. However, a button to hide or show the sensor would not be included; this feature would be added or removed by a professional wheelchair provider.

To the right of the standard functions and movement buttons was the specialized environment area. In this region, the user could tailor the system to their needs. In the system in this example, there were three specialized environments. When the user clicked on “Clapp Library”, the right side of the screen would change to the specific information for Clapp Library[1] (see Figure A-2). There was a region for the user to make notes about the location of ramps, elevator locations and details about moving around the particular environment. The specialized environment area was to be used to customize the robot to move more quickly around places the user travels frequently and also to record information such as the location of ramps in infrequently traveled

[1] Clapp Library is a building on the Wellesley College campus.

locations. This was judged to be impractical. If we were to customize for a particular location such as the user's house or office, most of the customization would come from the addition of maps for the navigation system. We might need to add to the interface to allow the user to specify travel from point A to point B, but we would not include the ability to add notes to the interface. In theory, the environments that were important enough to require customization would be the most familiar to the system's user.

The specialized environment area also included some buttons for tasks that would be specific to the environment. In the library, we envisioned customized routines for pulling up to a study carrel and the copy machine. Notice that these two routines would be similar to the standard "table" routine, but in these two cases the user would enter in the specific heights of the objects to allow the robot to pull up more effectively. The "stacks" button would perform the same task as moving down a corridor, but with a higher tolerance for tight spaces. To return to the primary interface screen, the user would click on the "return" button in the lower right hand corner of the specialized environment region. As discussed above, we decided that while the robot may use different navigation processes for different situations, we would not require the user to select these subroutines for us.

In our initial paper about this project, we stated that "this interface is not intended to be a final product, but a step towards an effective user interface for a robotic wheelchair system [Yanco *et al.*, 1995]." The further development of the navigation system helped us to reexamine our ideas for the user interface, as described above. Work with access methods required a refined interface for ease of customization and use. All of these factors resulted in the final interface described in Chapter 8.

Appendix B

Questionnaire for System Evaluation by Physical Therapists

1. What type of patients do you think will benefit from this system?

2. What are those patients currently using (generally speaking)?

3. Are your patients more likely to need indoor navigation, outdoor navigation or both?

4. What features of the system would be the most useful for your patients?

5. What do you think should be changed or added to make the system more useful for patients?

6. Which of the following access methods are likely to be used by your patients in this type of system?

   - single switch scanning

   - multiple switch arrays

- sip and puff
- joystick
- joystick with a plate added
- chin joystick
- mouth plate
- other:

7. What do you think should be changed or added to make your job easier?
8. Please list any issues that might prevent this system from being useful.
9. If this system were to be made commercially available in 5 years, would you be likely to introduce it to patients? If not, why?
10. Other comments:
11. Please circle your job title: PT OT Speech Therapist Physiatrist/MD Nurse Other:

Appendix C

Data from Indoor User Tests

First Robotic Trial						
	Type of	Number	Time (seconds)			Number
Subject	First Trial	of Clicks	Scanning	Command	Total	of Hits
A	Robotic	20	19.45	304.35	323.80	2
B		28	34.15	244.53	278.68	0
C		32	45.88	258.28	304.17	0
D		36	34.23	271.25	305.48	0
E		22	27.62	282.28	309.90	0
F		22	27.57	258.73	286.30	0
G		24	43.43	218.03	261.47	0
H	Manual	26	30.67	269.80	300.47	0
I		24	23.13	253.87	277.00	0
J		26	37.65	270.92	308.57	1
K		22	33.17	292.40	325.57	0
L		24	21.53	286.85	308.38	0
M		20	19.38	267.77	287.15	0
N		32	34.60	275.60	310.20	1
All Subjects						
Mean		25.57	30.89	268.19	299.08	0.29
Std Dev		4.85	8.33	21.45	18.44	0.61
Median		24.00	31.92	270.36	304.83	0.00
Subjects with Robotic as First Trial						
Mean		26.29	33.19	262.50	295.69	0.29
Std Dev		5.94	9.31	27.54	21.25	0.76
Median		24.00	34.15	258.73	304.17	0.00
Subjects with Manual as First Trial						
Mean		24.86	28.59	273.89	302.48	0.29
Std Dev		3.80	7.16	12.76	16.09	0.49
Median		24.00	30.67	270.92	308.38	0.00

Table C.1: Data for first robotic trial in the single switch scanning experiments. These tests are discussed in Section 5.3.

Second Robotic Trial						
	Type of	Number	Time (seconds)			Number
Subject	First Trial	of Clicks	Scanning	Command	Total	of Hits
A	Robotic	30	26.08	313.22	339.30	0
B		22	22.33	262.30	284.63	0
C		20	18.37	262.78	281.15	0
D		20	18.38	285.35	303.73	0
E		20	17.45	284.12	301.57	0
F		20	18.75	264.03	282.78	0
G		20	17.55	223.05	240.60	0
H	Manual	20	20.63	282.15	302.78	0
I		24	31.93	248.58	280.52	0
J		24	32.55	246.77	279.32	0
K		20	30.65	310.21	340.86	0
L		20	19.58	301.70	321.28	0
M		28	47.00	322.70	369.70	0
N		20	31.37	272.95	304.32	0
All Subjects						
Mean		22.00	25.19	277.14	302.32	0.00
Std Dev		3.33	8.58	28.41	32.48	0.00
Median		20.00	21.48	277.55	302.18	0.00
Subjects with Robotic as First Trial						
Mean		21.71	19.85	270.69	290.54	0.00
Std Dev		3.73	3.21	27.84	29.86	0.00
Median		20.00	18.38	264.03	284.63	0.00
Subjects with Manual as First Trial						
Mean		22.29	30.53	283.58	314.11	0.00
Std Dev		3.15	9.09	29.62	32.71	0.00
Median		20.00	31.37	282.15	304.32	0.00

Table C.2: Data for second robotic trial in the single switch scanning experiments. These tests are discussed in Section 5.3.

First Manual Trial						
	Type of	Number	Time (seconds)			Number
Subject	First Trial	of Clicks	Scanning	Command	Total	of Hits
A	Robotic	85	82.22	396.65	478.87	0
B		92	93.68	303.02	396.70	0
C		92	90.37	314.32	404.68	0
D		98	117.87	335.77	453.63	0
E		60	54.98	307.57	362.55	0
F		74	69.45	300.65	370.10	0
G		76	79.80	249.28	329.08	1
H	Manual	72	80.12	286.07	366.18	0
I		92	112.45	282.63	395.08	2
J		124	134.63	288.13	422.77	0
K		108	102.12	357.05	459.17	0
L		102	99.08	340.55	439.63	0
M		94	98.80	307.08	405.88	0
N		94	94.27	293.35	387.62	0
All Subjects						
Mean		90.21	93.56	311.58	405.14	0.21
Std Dev		16.25	20.28	36.39	42.09	0.58
Median		92.00	93.97	305.05	400.69	0.00
Subjects with Robotic as First Trial						
Mean		82.43	84.05	315.32	399.37	0.14
Std Dev		13.21	19.81	44.39	52.36	0.38
Median		85.00	82.22	307.57	396.70	0.00
Subjects with Manual as First Trial						
Mean		98.00	103.07	307.84	410.90	0.29
Std Dev		16.00	16.95	29.44	31.92	0.76
Median		94.00	99.08	293.35	405.88	0.00

Table C.3: Data for first manual trial in the single switch scanning experiments. These tests are discussed in Section 5.3.

Second Manual Trial						
	Type of	Number	Time (seconds)			Number
Subject	First Trial	of Clicks	Scanning	Command	Total	of Hits
A	Robotic	76	69.42	381.78	451.20	1
B		78	73.75	294.07	367.82	0
C		80	82.00	301.57	383.57	0
D		78	107.00	331.42	438.42	0
E		64	74.87	315.42	390.28	0
F		68	68.22	308.02	376.23	2
G		64	60.80	257.13	317.93	1
H	Manual	86	90.38	319.00	409.38	0
I		70	69.62	282.92	352.53	0
J		80	84.80	275.83	360.63	0
K		88	89.13	367.68	456.82	0
L		100	100.58	369.22	469.80	0
M		72	86.72	317.18	403.90	0
N		76	78.60	311.02	389.62	0
All Subjects						
Mean		77.14	81.13	316.59	397.72	0.29
Std Dev		9.79	12.99	36.19	43.71	0.61
Median		77.00	80.30	313.22	389.95	0.00
Subjects with Robotic as First Trial						
Mean		72.57	76.58	312.77	389.35	0.57
Std Dev		7.00	14.92	38.10	44.73	0.79
Median		76.00	73.75	308.02	383.57	0.00
Subjects with Manual as First Trial						
Mean		81.71	85.69	320.41	406.10	0.00
Std Dev		10.48	9.71	36.77	44.45	0.00
Median		80.00	86.72	317.18	403.90	0.00

Table C.4: Data for second manual trial in the single switch scanning experiments. These tests are discussed in Section 5.3.

	Manual	Robotic	
Subject	Ranking	Ranking	Difference
A	5	8	3
B	4	8	4
C	2	8	6
D	3	8	5
E	3	7	4
F	6	10	4
G	8	8	0
H	2	8	6
I	1	9	8
J	6	9	3
K	2	9	7
L	2	10	8
M	4	10	6
N	1	10	9
All Subjects			
Mean	3.50	8.71	5.21
Std Dev	2.10	0.99	2.42
Median	3.00	8.50	5.50
Robotic First			
Mean	4.43	8.14	3.71
Std Dev	2.07	0.90	1.89
Median	4.00	8.00	4.00
Manual First			
Mean	2.57	9.29	6.71
Std Dev	1.81	0.76	1.98
Median	2.00	9.00	7.00

Table C.5: Rankings for manual control and robotic control by single switch scanning test subjects in indoor user tests.

	Number	Time (seconds)				
Run	of Clicks	Scanning	Commands	Forward	Right	Left
1	96	104.9	170.9	146	12.7	12
2	66	74.2	160.1	143.4	8.7	8
3	66	70.9	151.1	134.9	8.6	7.7
4	76	78.2	154.2	136.8	10.5	6.8
5	70	68.5	159.5	142.4	10.9	6.2
6	68	70.4	158.2	142.2	9.8	6.4
7	64	61.7	160.6	144.2	9.8	6.6
8	64	62.9	165.3	147.7	10.6	6.8
9	72	71.5	169	150.5	11.3	7.2
10	72	73.5	172.1	152.1	11.6	8.5

Table C.6: Data for a single user over 10 manual runs with single switch scanning. The tests were run to attempt to optimize a person's performance with no robotic assistance. The sum of scanning time and command execution time is the total amount of time spent on the run. The sum of the time spent on forward, left and right commands is equal to the command execution time. No backward commands were issued during the tests. The data is discussed in Section 5.3.

Bibliography

[Adachi *et al.*, 1998] Yoshihisa Adachi, Yoshinori Kuno, Nobutaka Shimada, and Yoshiaki Shirai. Intelligent wheelchair using visual information on human faces. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, pages 354–359, 1998.

[Alanen *et al.*, 1996] Jarmo Alanen, Olavi Karasti, Pentti Mattila, Virpi Säntti, and Ari Virtanen. A drive assistanct for a wheelchair. In *Proceedings of the IEEE Conference on Systems, Man and Cybernetics*, pages 376–381, 1996.

[Amori, 1992] Richard D. Amori. *Vocomotion* – an intelligent voice-control system for powered wheelchairs. In *Proceedings of the Rehabilitation Engineering Society of North America 1992 Annual Conference*, pages 421–423. RESNA Press, 1992.

[Ashmore, 2000] Rucker Ashmore, May 2000. Personal communication.

[Baten *et al.*, 1998] Stefan Baten, Michael Lützeler, Ernst D. Dickmanns, Robert Mandelbaum, and Peter J. Burt. Techniques for autonomous, off-road navigation. *IEEE Intelligent Systems*, 13(6):57–65, November/December 1998.

[Beattie and Bishop, 1998] P.D. Beattie and J.M. Bishop. Self-localisation in the Senario autonomous wheelchair. *Journal of Intelligent and Robotic Systems*, 22:255–267, 1998.

[Bergasa *et al.*, 1999] L.M. Bergasa, M. Mazo, A. Gardel, J.C. Garcia, A. Ortu no, and A.E. Mendez. Guidance of a wheelchair for handicapped people by face tracking. In *Proceedings of the 7th IEEE Conference on Emerging Technologies and Factory Automation*, volume 1, pages 105–111, 1999.

[Bolmsjö *et al.*, 1995] Gunnar Bolmsjö, Håkan Neveryd, and Håkan Eftring. Robotics in rehabilitation. *IEEE Transactions on Rehabilitation Engineering*, 3(1):77–83, March 1995.

[Borenstein and Koren, 1991] J. Borenstein and Y. Koren. Potential field methods and their inherent limitations for mobile robot navigation. In *Proceedings of the 1991 IEEE Conference on Robotics and Automation*, pages 1398–1404, April 1991.

[Borgolte *et al.*, 1995] U. Borgolte, R. Joelper, H. Hoyer, H. Heck, W. Humann, J. Nedza, I. Craig, R. Valleggi, and A.M. Sabatini. Intelligent control of a semi-autonomous omnidirectional wheelchair. In *Proceedings of the 3rd International Symposium on Intelligent Robotic Systems '95*, pages 113–120, 1995.

[Borgolte *et al.*, 1998] Ulrich Borgolte, Helmut Hoyer, Christian Bühler, Helmut Heck, and Ralf Hoelper. Architectural concepts of a semi-autonomous wheelchair. *Journal of Intelligent and Robotic Systems*, 22:233–253, 1998.

[Bourhis and Agostini, 1998a] G. Bourhis and Y. Agostini. Man-machine cooperation for the control of an intelligent powered wheelchair. *Journal of Intelligent and Robotic Systems*, 22:269–287, 1998.

[Bourhis and Agostini, 1998b] G. Bourhis and Y. Agostini. The Vahm robotized wheelchair: system architecture and human-machine interaction. *Journal of Intelligent and Robotic Systems*, 22(1):39–50, May 1998.

[Bourhis and Pino, 1996] G. Bourhis and P. Pino. Mobile robotics and mobility assistance for people with motor impairments: rational justification for the VAHM project. *IEEE Transactions on Rehabilitation Engineering*, 4(1):7–12, March 1996.

[Breiman *et al.*, 1984] Leo Breiman, Jerome Friedman, Charles J. Stone, and R.A. Olshen. *Classification and Regression Trees.* Wadsworth International Group, 1984.

[Bühler *et al.*, 1997] Christian Bühler, Helmut Heck, and Wolfram Jumann. User-driven human-machine interface configuration for a wheelchair with complex functionality. In *Advancement of Assistive Technology*, pages 375–380. IOS Press, 1997.

[Charkari and Mori, 1993] N. Moghadam Charkari and Hideo Mori. A new approach for real time moving vehicle detection. In *Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 273–278, 1993.

[Charkari and Mori, 1995] N. Moghadam Charkari and Hideo Mori. Visual vehicle detection and tracking based on the 'sign pattern'. *Advanced Robotics*, 9(4):367–382, 1995.

[Connell and Viola, 1990] Jonathan Connell and Paul Viola. Cooperative control of a semi-autonomous mobile robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1118–1121, 1990.

[Cooper, 1999] Rory A. Cooper. Engineering manual and electric powered wheelchairs. *Critical Reviews in Biomedical Engineering*, 27(1&2):27–73, 1999.

[Crisman and Cleary, 1998] Jill D. Crisman and Michael E. Cleary. Progress on the deictically controlled wheelchair. In Mittal et al. [Mittal *et al.*, 1998], pages 137–149.

[Dallaway *et al.*, 1995] John L. Dallaway, Robin D. Jackson, and Paul H.A. Timmers. Rehabilitation robotics in europe. *IEEE Transactions on Rehabilitation Engineering*, 3(1):35–45, March 1995.

[Dario *et al.*, 1994] Paolo Dario, Eugenio Guglielmelli, and Benedetto Allotta. Robotics in medicine. In *Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*, volume 2, pages 739–752, 1994.

[Enders and Hall, 1990] Alexandra Enders and Marian Hall, editors. *Assistive Technology Sourcebook*, pages 295–307. RESNA Press, 1990.

[Engelhardt *et al.*, 1983] K. G. Engelhardt, Roger Wicke, Gregory L. Goodrich, and Larry J. Leifer. Evaluation of a robotic aid: From theory to application using an interactive model. In *Proceedings of the 6th Annual Conference on Rehabilitation Engineering*, pages 279–281, 1983.

[Everett, 1995] H.R. Everett. *Sensors for Mobile Robots: Theory and Application.* A.K. Peters, 1995.

[Field, 1999] Debbie Field. Powered mobility: A literature review illustrating the importance of a multifaceted approach. *Assistive Technology*, 11:20–33, 1999.

[Ford and Sheredos, 1995] J.M. Ford and S.J. Sheredos. Ultrasonic head controller for power wheelchairs. *Journal of Rehabilitation Research and Development*, 32(3):280–284, 1995.

[Gips *et al.*, 1996] "James Gips, Philip DiMattia, Francis X. Curran, and Peter Olivieri". Using EagleEyes – an electrodes based device for controlling the computer with your eyes – to help people with special needs. In "Joachim Klaus, Eduard Auff, Willibald Kremser, and Wolfgang Zagler", editors, *Interdisciplinary Aspects on Computers Helping People with Special Needs*, pages 77–83. R. Oldenbourg, 1996.

[Gips, 1998] James Gips. On building intelligence into EagleEyes. In Mittal et al. [Mittal *et al.*, 1998], pages 50–58.

[Gomi and Griffith, 1998] Takashi Gomi and Ann Griffith. Developing intelligent wheelchairs for the handicapped. In Mittal et al. [Mittal *et al.*, 1998], pages 150–178.

[Gorkani and Picard, 1994] Monika M. Gorkani and Rosalind W. Picard. Texture orientation for sorting photos "at a glance". In *Proceedings of the IEEE Conference on Pattern Recognition*, volume 1, pages 459–464, 1994. Also available as MIT Media Lab Tech Report No. 292.

[Gribble *et al.*, 1998] William S. Gribble, Robert L. Browning, Micheal Hewett, Emilio Remolina, and Benjamin J. Kuipers. Integrating vision and spatial reasoning for assistive navigation. In Mittal et al. [Mittal *et al.*, 1998], pages 179–193.

[Hagan *et al.*, 1997] Karen Hagan, Sean Hagan, Mike Hillman, and Jill Jepson. Design of a wheelchair mounted robot. In *Proceedings of ICORR '97: International Conference on Rehabilitation Robotics*, 1997.

[Harwin *et al.*, 1995] William S. Harwin, Tariq Rahman, and Richard A. Foulds. A review of design issues in rehabilitation robotics with reference to North American research. *IEEE Transactions on Rehabilitation Engineering*, 3(1):3–13, March 1995.

[Hillman *et al.*, 1999] Michael Hillman, Karen Hagan, Sean Hagan, Jill Jepson, and Roger Orpwood. A wheelchair mounted assistive robot. In *Proceedings of ICORR '99: International Conference on Rehabilitation Robotics*, pages 86–91, 1999.

[Horswill, 1994] Ian Horswill. Specialization of perceptual processes. Technical Report 1511, MIT Artificial Intelligence Laboratory, 1994.

[Hoyer *et al.*, 1997] Helmut Hoyer, Ulrich Borgolte, and Ralf Hoelper. An omnidirectional wheelchair with enhanced comfort features. In *Proceedings of ICORR '97: International Conference on Rehabilitation Robotics*, 1997.

[Jaffe *et al.*, 1990] David L. Jaffe, Husher L. Harris, and Steve K. Leung. Ultrasonic head controlled wheelchair/interface: A case study in development and technology transfer. In *Proceedings of the RESNA 13th Annual Conference*, pages 23–24. RESNA Press, 1990.

[Jaffe, 1981] David L. Jaffe. Smart wheelchair. In *Proceedings of the 4th Annual Conference on Rehabilitation Engineering*, pages 91–93, 1981.

[Jaffe, 1983] David L. Jaffe. Ultrasonic head control unit. In *Proceedings of the 6th Annual Conference on Rehabilitation Engineering*, pages 242–243, 1983.

[Jones and Sanford, 1996] Michael L. Jones and Jon A. Sanford. People with mobility impairments in the United States today and in 2010. *Assistive Technology*, 8:43–53, 1996.

[Katevas *et al.*, 1997] N.I. Katevas, N.M. Sgouros, S.G. Tzafestas, G. Papakonstantinou, P. Beattie, J.M. Bishop, P. Tsanakas, and D. Koutsouris. The autonomous mobile robot SENARIO: a sensor-aided intelligent navigation system for powered wheelchairs. *IEEE Robotics and Automation Magazine*, pages 60–70, December 1997.

[Kazi *et al.*, 1998] Zunaid Kazi, Shoupu Chen, Matthew Beitler, Daniel Chester, and Richard Foulds. Speech and gesture mediated intelligent teleoperation. In Mittal et al. [Mittal *et al.*, 1998], pages 194–210.

[Kettle *et al.*, 1992] Melvyn Kettle, Corinne Rowley, and M. Anne Chamberlain. A national survey of wheelchair users. *Clinical Rehabilitation*, 6:67–73, 1992.

[Konolige, 1997] Kurt Konolige. Small vision systems: Hardware and implementation. In *Eighth International Symposium on Robotics Research*, pages 203–212. Springer Verlag, 1997.

[Konolige, 1998] Kurt Konolige. *SRI Small Vision System: User's Manual*. SRI International, software version 1.2 edition, August 1998. Available at www.ai.sri.com/∼konolige.

[Kortenkamp *et al.*, 1998] David Kortenkamp, R. Peter Bonasso, and Robin Murphy, editors. *Artificial intelligence and mobile robots: case studies of successful robot systems.* MIT/AAAI Press, 1998.

[Kuno *et al.*, 1999a] Yoshinori Kuno, Saturo Nakanishi, Teruhisa Murashima, Nobutaka Shimada, and Yoshiaki Shirai. Robotic wheelchair with three control modes. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, pages 2590–2595, 1999.

[Kuno *et al.*, 1999b] Yoshinori Kuno, Saturo Nakanishi, Teruhisa Murashima, Nobutaka Shimada, and Yoshiaki Shirai. Intelligent wheelchair based on the integration of human and environment observations. In *Proceedings of the 1999 International Conference on Information Intelligence and Systems*, pages 342–349, 1999.

[Kwee *et al.*, 1989] H.H. Kwee, J.J. Duimel, J.J. Smits, A.A. Tuinhof de Moed, and J.A. van Woerden. The Manus wheelchair-borne manipulator: System review and first results. In *IARP Proceedings of the 2nd Workshop and Medical and Healthcare Robotics*, pages 385–395, 1989.

[Lacey *et al.*, 1998] Gerard Lacey, Shane MacNamara, and Kenneth M. Dawson-Howe. Personal adaptive mobility aid for the infirm and elderly blind. In Mittal et al. [Mittal *et al.*, 1998], pages 211–220.

[Lacey, 1999] Gerard Lacey. *Personal Adaptive Mobility Aid for the Frail Visually Impaired.* PhD thesis, University of Dublin, Trinity College, April 1999.

[Lachmann *et al.*, 1993] S. M. Lachmann, Eve Greenfield, and Alan Wrench. Assessment of need for special seating and/or electronic control systems for wheelchairs among people with severe physical disabilities. *Clinical Rehabilitation*, 7:151–156, 1993.

[Levine *et al.*, 1999] Simon P. Levine, David A. Bell, Lincoln A. Jaros, Richard C. Simpson, Yoram Koren, and Johann Borenstein. The NavChair assistive wheelchair navigation system. *IEEE Transactions on Rehabilitation Engineering*, 7(4):443–451, December 1999.

[Lorigo *et al.*, 1997] Liana M. Lorigo, Rodney A. Brooks, and W. Eric L. Grimson. Visually-guided obstacle avoidance in unstructured environments. In *IEEE-RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 373–379, 1997.

[Madarasz *et al.*, 1991] Richard L. Madarasz, Loren C. Heiny, Robert F. Cromp, and Neal M. Mazur. The design of an autonomous vehicle for the disabled. In *Autonomous Mobile Robots: Control, Planning, and Architecture*, pages 351–359. IEEE Computer Society Press, 1991. Originally appeared in *IEEE Journal of Robotics and Automation*, Volume RA-2, Number 3, September 1986, pages 117–126.

[Mahoney, 1997] Richard M. Mahoney. Robotic products for rehabilitation: Status and strategy. In *Proceedings of ICORR '97: International Conference on Rehabilitation Robotics*, 1997.

[Maurer *et al.*, 1996] M. Maurer, R. Behringer, S. Fürst, F. Thomanek, and E.D. Dickmanns. A compact vision system for road vehicle guidance. In *Proceedings of the 13th International Conference on Pattern Recognition*, volume 3, pages 313–317, 1996.

[Mazo *et al.*, 1995] Manuel Mazo, Francisco J. Rodríguez, José L. Lázaro, Jesús Ure na, Juan C. García, Enrique Santiso, Pedro Revenga, and J. Jesús García. Wheelchair for physically disabled people with voice, ultrasonic and infrared sensor control. *Autonomous Robots*, 2:203–224, 1995.

[Miller and Slack, 1995] David P. Miller and Marc G. Slack. Design and testing of a low-cost robotic wheelchair prototype. *Autonomous Robots*, 2:77–88, 1995.

[Mittal *et al.*, 1998] Vibhu Mittal, Holly A. Yanco, John Aronis, and Richard C. Simpson, editors. *Lecture Notes in Artificial Intelligence: Assistive Technology and Artificial Intelligence*. Springer-Verlag, 1998.

[Mori *et al.*, 1995] Hideo Mori, N. Moghaddam Charkari, and Shinji Kotani. Danger estimation of vehicles at intersection. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 781–787, 1995.

[Mori *et al.*, 1998] Hideo Mori, Shinji Kotani, and Noriaki Kiyohiro. HITOMI: Design and development of a robotic travel aid. In Mittal et al. [Mittal *et al.*, 1998], pages 221–234.

[Nisbet *et al.*, 1996] Paul Nisbet, John Craig, Phil Odor, and Stuart Aitken. 'smart' wheelchairs for mobility training. *Technology and Disability*, 5:49–62, 1996.

[Nourbakhsh, 1998] Illah Nourbakhsh. Dervish: An office-navigating robot. In Kortenkamp et al. [Kortenkamp *et al.*, 1998], pages 73–90.

[Ojala *et al.*, 1991] Jari Ojala, Kenji Inoue, Ken Sasaki, and Masaharu Takano. Development of an intelligent wheelchair using computer graphics animation and simulation. *Computer Graphics Forum*, 10:285–295, 1991.

[Perkowski and Stanton, 1991] Marek A. Perkowski and Kevin Stanton. Robotics for the handicapped. In *Northcon Conference Record*, pages 278–284, 1991.

[Pol, 1995] Polaroid Corporation. *Ultrasonic Ranging System: Description, operation and use information for conducting tests and experiments with Polaroid's Ultrasonic Ranging System components*, May 1995.

[Press *et al.*, 1992] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing; Second Edition*. Cambridge University Press, 1992.

[Quinlan, 1993] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaurmann Publishers, 1993.

[Radhakrishnan and Nourbakhsh, 1999] Dandapani Radhakrishnan and Illah R. Nourbakhsh. Topological robot localization by training a vision-based transition detector. In *Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 468–473, 1999.

[Rosier *et al.*, 1991] J.C. Rosier, J.A. van Woerden, L.W. van der Kolk, B.J. Driessen, H.H. Kwee, J.J. Duimel, J.J. Smits, A.A. Tuinhof de Moed, G. Honderd, and

P.M. Bruyn. Rehabilitation robotics: The MANUS concept. In *Fifth International Conference on Advanced Robotics: Robots in Unstructured Environments*, volume 1, pages 893–898, 1991.

[Schiehlen and Dickmanns, 1994] J. Schiehlen and E.D. Dickmanns. A camera platform for intelligent vehicles. In *Proceedings of the Intelligent Vehicles '94 Symposium*, pages 393–398, 1994.

[Scott and Munro, 1985] Jeffrey Brent Scott and James L. Munro. Collision-avoidance system for a motorized wheelchair. In *Proceedings of the RESNA 8th Annual Conference*, pages 51–52. RESNA Press, 1985.

[Shoval *et al.*, 1998] Shraga Shoval, Johann Borenstein, and Yoram Koren. The Navbelt – a computerized travel aid for the blind based on mobile robotics technology. *IEEE Transactions on Biomedical Engineering*, 45(11):1376–1386, November 1998.

[Simpson and Levine, 1997] Richard C. Simpson and Simon P. Levine. Development and evaluation of voice control for a smart wheelchair. In Stephen Sprigle, editor, *Proceedings of the Rehabilitation Engineering Society of North America 1997 Annual Conference*, pages 417–419. RESNA Press, 1997.

[Simpson and Levine, 1999] Richard C. Simpson and Simon P. Levine. Automatic adaptation in the NavChair assistive wheelchair navigation system. *IEEE Transactions on Rehabilitation Engineering*, 7(4):452–463, December 1999.

[Simpson *et al.*, 1998] Richard C. Simpson, Simon P. Levine, David A. Bell, Lincoln A. Jaros, Yoram Koren, and Johann Borenstein. NavChair: an assistive wheelchair navigation system with automatic adaptation. In Mittal et al. [Mittal *et al.*, 1998], pages 235–255.

[Simpson *et al.*, 1999] Richard Simpson, Daniel Poirot, and Mary Francis Baxter. Evaluation of the Hephaestus smart wheelchair system. In *Proceedings of ICORR '99: International Conference on Rehabilitation Robotics*, pages 99–105, 1999.

[Stanton *et al.*, 1991] Kevin B. Stanton, Paul R. Sherman, Mark L. Rohwedder, Christopher P. Fleskes, David R. Gray, Dinh T. Minh, Cecilia Espinoza, Dieudonne Mayui, Mohammad Ishaque, and Marek A. Perkowski. PSUBOT – a voice-controlled wheelchair for the handicapped. In *Proceedings of the 33rd Midwest Symposium on Circuits and Systems*, volume 2, pages 669–672, 1991.

[Stentz and Hebert, 1995] Anthony Stentz and Martial Hebert. A complete navigation system for goal acquisition in unknown environments. *Autonomous Robots*, 2:127–145, 1995.

[Szummer and Picard, 1998] Martin Szummer and Rosalind W. Picard. Indoor-outdoor image classification. In *Proceedings of the IEEE International Workshop on Content-based Access of Image and Video Databases*, 1998. Also available as MIT Media Lab Tech Report No. 445.

[Tachi and Komoriya, 1991] Susumu Tachi and Kiyoshi Komoriya. Guide dog robot. In *Autonomous Mobile Robots: Control, Planning, and Architecture*, pages 360–367. IEEE Computer Society Press, 1991. Originally appeared in *International Symposium on Robotics Research*, Volume 2, pages 333-340, MIT Press, 1984.

[Tachi *et al.*, 1985] Susumu Tachi, Kazuo Tanie, Kiyoshi Komoriya, and Minoru Abe. Electrocutaneous communication in a guide dog robot (MELDOG). *IEEE Transactions on Biomedical Engineering*, BME-32(7):461–469, July 1985.

[Tahboub and Asada, 1999] Karim A. Tahboub and Harry H. Asada. A semi-autonomous control architecture applied to robotic wheelchairs. In *Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 906–911, 1999.

[Thorpe *et al.*, 1988] Charles Thorpe, Martial H. Hebert, Takeo Kanade, and Steven A. Shafer. Vision and navigation for the Carnegie-Mellon Navlab. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(3):362–372, May 1988.

[Topping, 1999] Mike Topping. The development of Handy 1, a robotic system to assist the severely disabled. In *Proceedings of ICORR '99: International Conference on Rehabilitation Robotics*, pages 244–249, 1999.

[Turk *et al.*, 1988] Matthew A. Turk, David G. Morgenthaler, Keith D. Gremban, and Martin Marra. VITS – a vision system for autonomous land vehicle navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(3):342–361, May 1988.

[Vailaya, 2000] Aditya Vailaya. *Semantic Classification in Image Databases.* PhD thesis, Michigan State University, 2000.

[Van der Loos, 1995] H.F. Machiel Van der Loos. VA/Stanford rehabilitation robotics research and development program: Lessons learned in the application of robotics technology to the field of rehabilitation. *IEEE Transactions on Rehabilitation Engineering*, 3(1):46–55, March 1995.

[Wagner *et al.*, 1999] J.J. Wagner, H.F.M. Van der Loos, N. Smaby, K. Chang, and C. Burgar. ProVAR assistive robot interface. In *Proceedings of ICORR '99: International Conference on Rehabilitation Robotics*, pages 250–254, 1999.

[Wakaumi *et al.*, 1992] Hiroo Wakaumi, Koichi Nakamura, and Takayoshi Matsumura. Development of an automated wheelchair guided by a magnetic ferrite marker lane. *Journal of Rehabilitation Research and Development*, 29(1):27–34, Winter 1992.

[Wang *et al.*, 1997] Hongbo Wang, Takakazu Ishimatsu, and Jamal Tariq Mian. Self-localization for an electric wheelchair. *JSME International Journal, Series C - Mechanical Systems Machine Elements and Manufacturing*, 40(3):433–438, 1997.

[Warren, 1990] C. Gerald Warren. Powered mobility and its implications. *Journal of Rehabilitation Research and Development: Clinical Supplement*, pages 74–85, 1990.

[Wright *et al.*, 1996] Anne Wright, Randy Sargent, and Carl Witty. *ARC Development System User's Guide*. Newton Research Labs, 1.0.1 edition, 1996. Available at http://www.newtonlabs.com/arc/index.html.

[Wyszecki and Stiles, 1982] Günter Wyszecki and W. S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae, Second Edition*. John Wiley & Sons, 1982.

[Yanco and Gips, 1997] Holly A. Yanco and James Gips. Preliminary investigation of a semi-autonomous robotic wheelchair directed through electrodes. In Stephen Sprigle, editor, *Proceedings of the Rehabilitation Engineering Society of North America 1997 Annual Conference*, pages 414–416. RESNA Press, 1997.

[Yanco and Gips, 1998] Holly A. Yanco and James Gips. Driver performance using single switch scanning with a powered wheelchair: robotic assisted control versus traditional control. In Stephen Springle, editor, *Proceedings of the Rehabilitation Engineering Society of North America 1998 Annual Conference*, pages 298–300. RESNA Press, 1998.

[Yanco *et al.*, 1995] Holly A. Yanco, Anna Hazel, Alison Peacock, Suzanna Smith, and Harriet Wintermute. Initial report on Wheelesley: a robotic wheelchair system. In *Proceedings of the IJCAI-95 Workshop on Developing AI Applications for the Disabled*, Montreal, Canada, August 1995.

[Yoder *et al.*, 1996] John-David Yoder, Eric T. Baumgartner, and Steven B. Skaar. Initial results in the development of a guidance system for a powered wheelchair. *IEEE Transactions on Rehabilitation Engineering*, 4(3):143–151, September 1996.

[Yu and Jain, 1997] Bin Yu and Anil K. Jain. Lane boundary detection using a multiresolution hough transform. In *Proceedings of the International Conference* *on Image Processing*, volume 2, pages 748–751, 1997.