7-15-2014

A Brain Robot Interface for Autonomous Activities of Daily Living Tasks

Don Indika Upashantha Pathirage
University of South Florida, indika@mail.usf.edu

A Brain Robot Interface for Autonomous Activities of Daily Living Tasks

by

Indika Upashantha Pathirage

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Computer Engineering
Department of Computer Science and Engineering
College of Engineering
University of South Florida

Co-Major Professor: Sudeep Sarkar, Ph.D.
Co-Major Professor: Redwan Alqasemi, Ph.D.
Rajiv Dubey, Ph.D.

Date of Approval:
July 15, 2014

Keywords: Brain Computer Interface, Object Selection, Rehabilitation Robotics, Assistive Robotics, Task Level Control

DEDICATION

I would like to dedicate this thesis to my parents, family and friends, and all the teachers who taught me.

ACKNOWLEDGMENTS

I would like to thank Dr. Sudeep Sarkar, Dr. Redwan Alqasemi and Dr. Rajiv Dubey for their guidance and support throughout this project. I would specially like to acknowledge Dr. Alqasemi for being supportive and patient with me during this challenging project. I would like to thank Elijah Klay for his initial work on the grid creation algorithm described in section 4.3. I would like to thank Anthony Murphy from the USF Department of Psychology for his support in getting me up to speed on BCI systems. I would also like to thank all the members of the CARRT lab at USF, Garret, Karan, Mustafa, Kester, Daniel, Andoni and Paul for their support.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

There have been substantial improvements in the area of rehabilitation robotics in the recent past. However, these advances are inaccessible to a large number of people with disabilities who are in most need of such assistance. This group includes people who are in a severely paralyzed state, that they are completely "locked-in" in their own bodies. Such persons usually retain full cognitive abilities, but have no voluntary muscle control. For these persons, a Brain Computer Interface (BCI) is often the only way to communicate with the outside world and/or control an assistive device. One major drawback to BCI devices is their low information transfer rate, which can take as long as 30 seconds to select a single command. This can result in mental fatigue to the user, specially if it necessary to make multiple selections over the BCI to complete a single task. Therefore, P300 based BCI control is not efficient for controlling a assistive robotic device such as a robotic arm.

To address this shortcoming, a novel vision based Brain Robot Interface (BRI) is presented in this thesis. This visual user interface allows for selecting an object from an unstructured environment and then performing an action on the selected object using a robotic arm mounted to a power wheelchair. As issuing commands through BCI is slow, this system was designed to allow a user to perform a complete task via a BCI using an autonomous robotic system while issuing as few commands as possible. Furthermore, the new visual interface allows the user to perform the task without losing concentration on the stimuli or the task. In our interface, a scene image is captured by a camera mounted on the wheelchair, from which, a dynamically sized non-uniform stimulus grid is created using edge information. Dynamically sized grids improve object selection efficiency. Oddball paradigm and P300 Event Related Potentials (ERP) are used to select stimuli, where the stimuli being each cell in the grid. Once selected, object segmentation and matching is used to identify the object. Then the user, using BRI, chooses an action to be performed on the object by the wheelchair mounted robotic arm (WMRA). Tests on 8 healthy human subjects validated the functionality of the system. An average accuracy of 85.56% was achieved for stimuli selection over

all subjects. With the proposed system, it took the users an average of 5 commands to perform a task on an object. The system will eventually be useful for completely paralyzed or locked-in patients for performing activities of daily living (ADL) tasks.

CHAPTER 1

INTRODUCTION

1.1 Motivation

People with disabilities have to face many difficulties in their daily life. According to the 2010 US census data [1], 19.2 million Americans suffer from some form of disability. Rehabilitation Robotics technology aims to help improve their daily lives by helping them regain control of some parts of their lives that are impacted by disability. A set of common, every day tasks, performance of which is required for personal self-care and independent living, are called Activities of Daily Living (ADL) [2].

In most cases of disability, the individuals retain some form of voluntary motor control that can be used to control an assistive device. However, in extreme cases, the patients can lose the ability to perform voluntary muscle movements. Amyotrophic lateral sclerosis (ALS) is perhaps one of the most prominent of these conditions [3]. ALS is a progressive neuro-degenerative disease that ultimately leaves the patient in a locked-in state [3]. The patient is essentially locked inside their own body, unable to move any voluntary muscles while being completely conscious and aware of the surroundings. Other conditions such as brain-stem stroke and other extended brain lesions, Guillain-Barre syndrome, some rare cases of Parkinsons disease, multiple sclerosis, and severe cerebral palsy can also leave a patient unable to voluntarily control their muscles.

In extreme debilitating conditions such as the above, often the only possible way for the patient to communicate with the outside world is through a Brain Computer Interface (BCI). A BCI is a user input device, which captures data of the electrical activity of the user's brain and then analyzes these signals allowing the user to convey decisions to a computer without the use of any muscles [4]. In this thesis, we define a Brain Robot Interface (BRI) to be a BCI whose primary objective is to control a robotic system (more specifically, a robotic manipulator). As this work

describes a BCI system with a primary objective of robotic control, we may use the terms BCI and BRI interchangeably throughout the rest of this work. The use of BCI devices is not limited to completely paralyzed individuals. Any person can use a BCI to control an assistive device if they choose to do so.

1.2 Scope of Work

The goal of this research is to create a system that will enable a user to perform an ADL task via the Brain Robot Interface using the WMRA system more efficiently. In this regard, efficiency is defined as performing the task by issuing minimum number of commands. Minimizing the number of commands that need to be issued via the BRI is necessary as BCI systems are low symbol rate devices. For example, using the P300 paradigm, it takes roughly up to 30 seconds to select a single command. Therefore, by reducing the number of commands needed to perform an action, the time taken for the task and user fatigue are reduced.

To accomplish this goal, the user of the system will only issue high level task based commands, and the system will perform the low level tasks autonomously. The user will first select an object and then select an action to be performed on the selected object as shown in figure 1.1. Therefore, the scope of this work is to design and implement such a Brain Robot Interface (Visual User Interface) that is optimized for issuing high level commands necessary for performing Activities of Daily Living Tasks using a robotic manipulator.

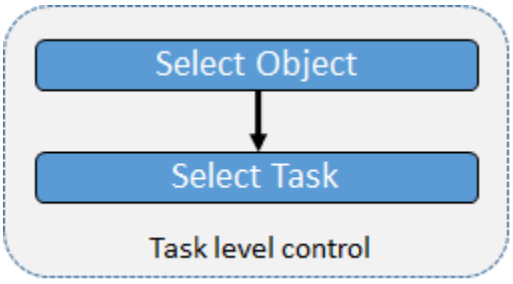


Figure 1.1: Task level control

As a proof of concept, the work described in this thesis implements the ability to execute three simple ADL tasks using the wheelchair mounted robotic arm via commands issued through our AGEOS Brain Robot Interface.

1.3 Contributions of this Work

In this thesis, we present a method for performing ADL tasks through a Brain Robot Interface (BRI) by issuing task level(high level) commands.

1. First, the major contribution in this work is a novel vision based visual interface, hereafter called AGEOS (Adaptive Grid based Efficient Object Selection) BRI. This system allows for selecting novel objects, segments and identifies the selected objects, and then allows the user to select a custom action (depending on the selected object) to be performed on the object.

2. Second, we also present an algorithm for stimulus presentation for the AGEOS interface to be used with the P300 paradigm. The algorithm was adopted from the the P300 row-colomn based presentation algorithm, but includes changes to account for the lack of predefined rows and columns and the neighbor associativity issues arising from such a configuration.

3. Finally, we also present a proof of concept implementation of our Brain Robot Interface system that is capable of performing simple ADL tasks via the wheelchair mounted robotic arm.

1.4 Thesis Outline

The rest of this thesis is organized as follows: Chapter 2 contains background information about Brain Computer Interfaces and the Wheelchair Mounted Robotic Arm used in this work. The second path of Chapter 2 is the literature review. Chapter 3 presents an overview of the entire system, and a brief introduction of each of the major components. The rest of the chapters describe the components in more detail. In Chapter 4, our novel adaptive grid based object selection method is described. Chapter 5 goes over the details of object segmentation and object identification. In Chapter 6, stimulus presentation and P300 response calculation are described. Chapter 7 describes how the user selected task is performed by the wheelchair mounted robotic arm system. Testing of our system and results are discussed in Chapter 8. Part of the work described in this thesis has been published in [5].

CHAPTER 2

BACKGROUND AND PREVIOUS WORK

2.1 Brain Computer Interfaces

A Brain Computer Interface (BCI) is a system that attempts to create a communication pathway from the brain to an external device. BCI research began in the early 70's and has advanced significantly since then. The general principle behind BCI systems is to capture data of the electrical activity of the user's brain. Then through signal processing techniques deduce the user's intentions, which are then used as an input to control an external device. The most common use of BCI systems is in assisting persons with disabilities, and augmenting human cognitive and sensory-motor functions.

2.1.1 Common Issues with BCIs

BCI systems in general have several disadvantages. One of them is the low signal to noise ratio in EEG signals. BCIs try to infer user intention by analyzing the electrical field resulting from neuron activity in the brain. This is measured by placing electrodes on the user's scalp. However, the amplitude of these EEG signals are usually very low compared to other noise signals present. Noise includes electrical noise, mains power-line noise, and noise resulting from neuro-muscular activity in the user's body.

Due to this reason, it is often necessary to apply some noise reducing techniques, such as averaging, when dealing with EEG signals. This is specially true in the case of P300 paradigm, which we will be using for our BCI interface described in this thesis. This results in a slow information transfer rate from the user to the computer (the system being controlled).

2.1.2 Invasive and Non Invasive BCIs

Based on how the electrical activity of the brain is captured, there are primarily two types of BCI systems [6].

1. Invasive BCIs: Also known as Electro Cortico Graphs (ECoG), invasive brain computer interfaces use electrodes placed under the patient's skull, directly on the gray matter of the brain to record the electrical signal resulting from neural activity of the brain. Usually, a surgical incision in the skull is made and then the electrodes are placed on the cerebral cortex of the brain. Signals from an ECoG is usually of higher quality and have less noise artifacts because the electrodes are in direct contact with brain tissue. Therefore, ECoG based BCI systems usually have higher accuracy compared to other BCI systems. However, a major disadvantage is that as time progresses, scar tissue tends to form around the electrodes and the body treats them as foreign objects. This, in turn, reduces the sensitivity of the electrodes and therefore the quality of the recorded electrical signals [7]. ECoG systems have been successfully used. However because of the risks associated with brain surgery, invasive BCI's are less desirable.

2. Non-invasive BCIs: These systems, also known as Electroencephelography's (EEG's) record the brain's electrical activity via non-invasive electrodes placed on the patients scalp. The electrodes are placed on the skin and no incisions are required. Using a very sensitive bio-signal amplifier, electric potential readings are taken from the surface of the skin, amplified, and digitized. The electric potential is usually measured with respect to a reference electrode. Electrode placement positions on the scalp are standardized according to the international 10-20 system [8].

2.1.3 Electrode Types

Non invasive EEG based brain signal acquisition systems can be further divided based on the type of electrodes used

1. Ag/Cl Wet Electrodes: These electrodes require a EEG gel between the electrode and the skin to make proper electrical contact. Skin preparation is also necessary to get good quality EEG readings.

2. Dry Active Electrodes: Active electrodes usually contain a high gain op amp on the electrode it self, and amplifies the measured EEG signal before sending it to the main amplifier.

2.1.4 BCI Paradigms

BCI systems can be divided into several categories depending on the modality they use to determine user intention.

1. SSVEP: Visually Evoked Potentials are electrical potential changes in the brain due to an external visual stimulus. These can be observed in the sonsory cortex of the brain in response to an external visual stimulus. A Steady State Visually Evoked Potential (SSVEP) is a periodic component of the same frequency as an external visual stimulus blinking at a constant frequency. The strongest SSVEP response is observable for frequencies in the range of 5-20 Hz. In this method, the subject will be shown several visual stimuli that will be blinking at different frequencies. By observing the frequency of the strongest SSVEP response, it is possible to deduce the visual stimulus the subject was focusing on.

2. P300: P300 is a part of Event Related Potential (ERP) in EEG. An ERP is the electrical potential change that occurred as a result of a sensory, motor or cognitive function in the brain. It is thought to be a product of brain activity resulting from evaluating and categorizing a stimulus. The oddball paradigm is usually used to evoke P300 response. In this method, the subject is presented with common (probable) stimuli and rare (improbable) stimuli. The presentation of improbable stimuli causes a P300 response. However, the P300 is attention dependent in that for it to occur, the subject needs to pay attention.

2.2 Wheelchair Mounted Robotic Arm

Wheelchair-mounted robotic arms (WMRA) can improve the users ability to perform ADLs and reduce dependency on caregivers. Despite the vast potential for these devices, they have had little commercial success, due to general difficulty of operation and low payload.

The first use of a robotic arm mounted to a wheelchair was in the work of Mason [9] at the Veterans Administration Prosthetics Center in New York in the 1970s. This manipulator had four degrees-of-freedom, and featured a novel telescoping design which allowed it to reach the ceiling or the floor.

The University of South Florida's WMRA [10] is a 7 DoF robotic arm mounted on a powered wheelchair. The system, as a whole, has nine degrees-of-freedom, with coordinated control of both the arm and wheelchair base.

Figure 2.1: WMRA II system with Kinect camera and gripper camera

The USF WMRA shown in figure 2.1 was designed to have better ability to manipulate objects, greater payload, and easier control than current commercially available alternatives. It goes above and beyond other 6 DoF systems, adding a 7th DoF to incorporate redundancy. This allows for a greater freedom of movement during task execution. Using a gallon jug of milk (weighing approximately 4 kg) as a baseline, the USF WMRA was designed to accommodate a payload of 6 kg at full horizontal extension. Through the use of encoders and specially designed control software, the user can choose to use Cartesian control, or joint control.

The WMRA system has two main control modes:

1. Teleoperation: In Teleoperation mode, the WMRA arm will follow the user's directional movements. The user will issue directional commands to the WMRA using a variety of input devices including the OMNI haptic device, Joystick, etc.

2. Autonomous motion: In autonomous mode, only the goal pose is provided to the WMRA system. A trajectory of way-points is generated from the current position to the goal pose. Then for each waypoint, inverse kinematics is performed using the resolved rate algorithm. Thus, the arm is able to reach the desired goal pose autonomously.

2.3 Previous Work

Most of the previous research in BCI controlled assistive robotics have been focused on wheelchair control and navigation.

Pfurtscheller et al.[11] implemented EEG based control of a functional electrical stimulation (FES) device to grasp an object. The tetraplegic patient used imagination of foot movements to effect sequential grasp phases. Millan et al. [12] and Tonin et al. [13] in their works described asynchronous and spontaneous control of a wheelchair using BCI. In [12], the subjects were trained to voluntarily modulate EEG signals by executing three mental tasks that corresponded to three directional movements of the wheelchair. Machine learning was used to train models that produced the correspondence between thoughts and wheelchair actions.

Rebsamen et al. [14] and Iturrate et al. [15] have demonstrated the use of P300 oddball paradigm to control the movement of a wheelchair. Bell et al. [16] developed an interface in which two pre-segmented solid colored objects were presented to the user as stimuli. The p300 response

was used to identify the object of interest. However, the objects have to be pre-segmented before BCI stimuli presentation and they did not demonstrate how arbitrary objects could be selected using the BCI.

Escolano et al. [17] used the P300 oddball paradigm to control a tele-presence robot. A live image taken from a camera was overlaid with a polar grid. Selecting an intersection point on this grid would determine the target location for the mobile robot to traverse to. A subset of these images (with obstacle locations removed) were presented to the user via the P300 paradigm. However, this system was focused towards robot navigation and did not include any capabilities for performing ADL's or manipulating objects.

Perrin et al. used a different BCI paradigm to control and navigate a robot. In his method, an autonomous navigation system proposes the most likely destination to the user based on sensory data. Then the user makes a decision to either reject or accept the proposed destination. Then error-related EEG signals measured from the user's brain are analyzed using a Bayesian network in order to fuse the result of the BCI and the navigation systems proposition. The determined action will then be carried out by the autonomous navigation system.

Previous work in our lab, [18] and [19], consisted of controlling the movement of a robotic arm using a P300 based BCI and a visual interface. The visual interface comprised of a matrix of linear, rotational and directional symbols that were flashed on a screen. P300 oddball paradigm enabled the user selection of the directional command. This user interface can be seen in figure 2.2.

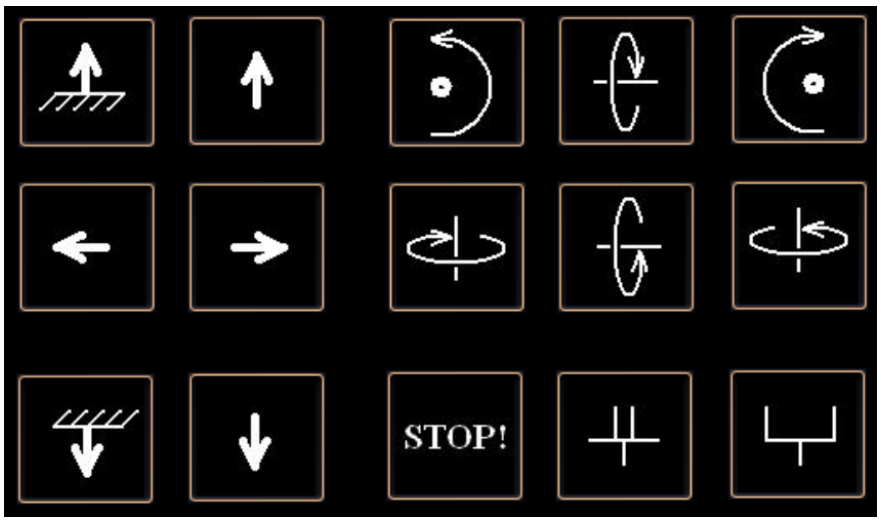


Figure 2.2: P300 BCI Cartesian control for WMRA [19].

The drawback of this system was that it took a long series of directional commands to perform a simple task such as picking up a cup. This, coupled with the slow symbol rate of the BCI P300 system, made it very tedious for the subject to perform an ADL task. Moreover, user's attention was divided in between the flashing screen and the robot arm location which led to poor accuracy on the P300 BCI system. Gabriel et al. [20] implemented a similar system for controlling the movement of a wheelchair.

CHAPTER 3

SYSTEM OVERVIEW

As previously mentioned, it is not practical to use a Cartesian directional control method for a Brain Robot Interface to perform an ADL task. Even to perform a simple task, it requires the user to issue multiple commands through the BCI/BRI, causing user fatigue and increasing the time taken to complete an ADL task. Therefore, it is necessary to have a more efficient method that minimizes the user fatigue as well as the time required to perform an ADL task when using a Brain Computer Interface.

Our novel vision-based brain robot interface described in this thesis accomplishes this goal by minimizing the number of commands the user needs to issue via the BRI. This is achieved by allowing the human user to issue high level task related commands while the the low level tasks are carried out autonomously whenever possible.

This chapter details the system overview of our approach. The system is divided into five main components as can be seen in figure 3.1.

1. User Interaction: This component covers the graphical user interface that is used to interact with the system. The goal of our research is to improve the user experience and efficiency. Overview of User Interaction in section 3.1 describes the interface from a user's prospective. It describes steps involved for a user to navigate the interface in order to select an object and then perform a task on it.

2. Object Selection: This component includes the novel Adaptive Grid based Efficient Object Selection (AGEOS). This allows the user to select a screen region belonging to an object in a visual scene image through the P300 BCI paradigm, which uses discrete stimuli.

3. Object Segmentation and Identification: Once a screen location on the object has been selected, this component segments the object from the background and then attempts to

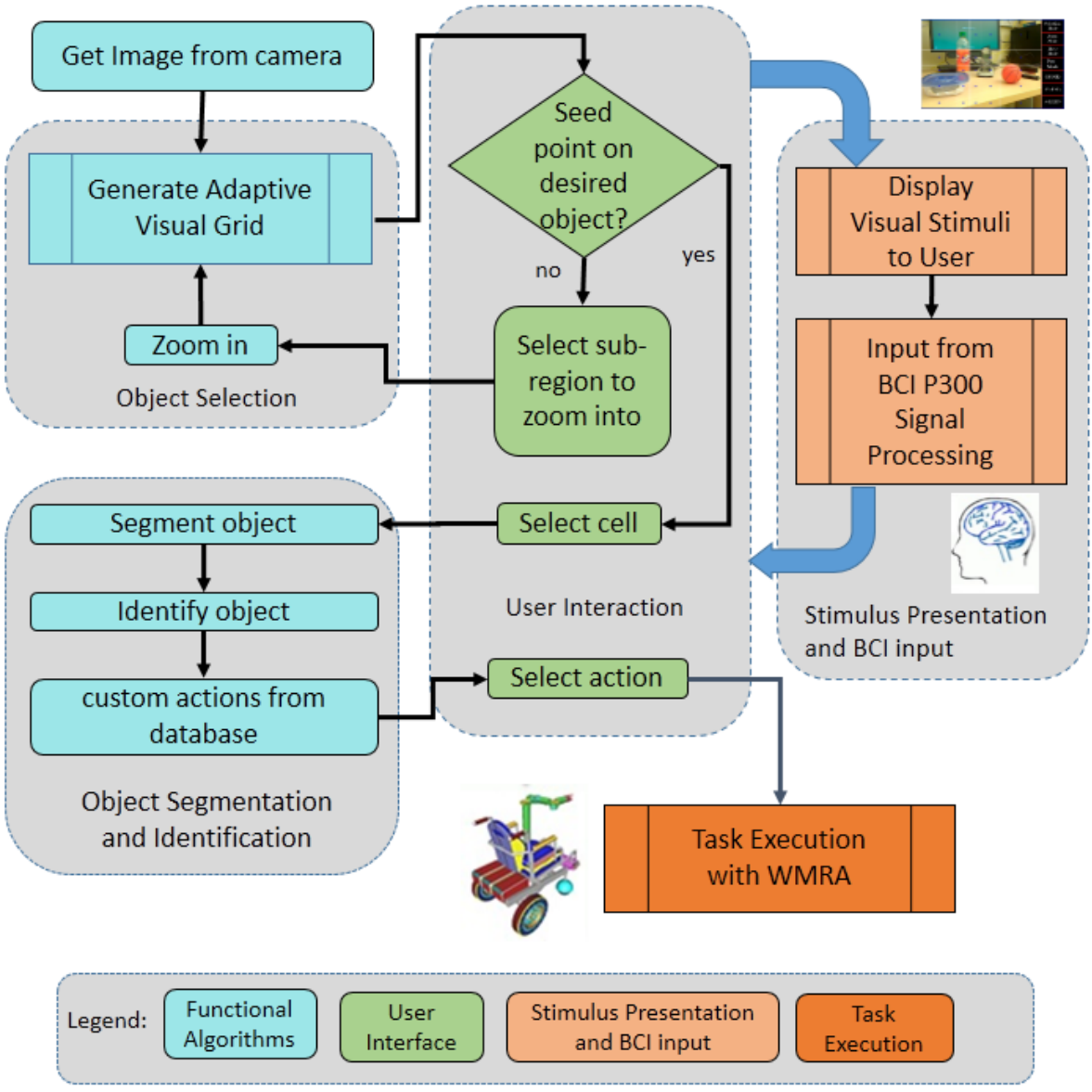


Figure 3.1: System overview

identify it from a database of known objects. It then displays to the user a list of custom actions that can be performed on the selected object.

4. Stimulus Presentation and BCI Input: This module presents the visual stimuli to the user and then calculates the selected input based on P300 signal processing. This is then sent to the WMRA system as an input.

5. Task Execution: Once an action is selected to be performed on an object, this component works with the robotic system to physically execute the selected task.

It is important to note the arrows between the 'User Interaction' component and the 'Stimulus Presentation' component in figure 3.1. When user input is required, it is obtained through the 'Stimulus Presentation and BCI Input' module.

3.1 Overview of User Interaction

The user is tasked with making high level decisions while the software system automates the low level tasks such as arm movement and control. The main end use of our system is performing ADL tasks, which usually consists of manipulating an object in the user's environment. Therefore, performing an ADL task can be separated into two high level abstract steps:

1. Select the object that the user wishes to interact with
2. Select an action to be performed on the previously selected object

Hence, task level user interaction with our system is composed of the two steps mentioned above. The first step would be identifying and selecting an object in the user's environment, which the user intends to manipulate. For example, let us take the ADL task of drinking water from a cup that is on a table. In order to perform this task, the first step would entail selecting the cup. Then the user needs to choose an action to be performed on the previously selected object, which is drinking.

3.2 Tools and Software Used

This section describes the software libraries and other tools that are used in this work.

- OpenCV: is an open source, comprehensive library of computer vision functions.
- BCI2000: is an open source brain computer interface software system. This software system includes capabilities for EEG signal acquisition and signal processing for several different BCI paradigms. Our work utilizes the p300 signal processing functionality of BCI2000 for stimulus classification. Classification consists of determining whether presenting stimulus to the user evoked a P300 response in the user's brain.

CHAPTER 4

ADAPTIVE VISUAL GRID FOR EFFICIENT OBJECT SELECTION

4.1 Introduction

In the context of this section, an object is defined to be any physical item in the user's surroundings that he/she wishes to interact as part of an Activity of Daily Living (ADL) task. A clear example of an ADL activity would be picking up a cup from a table. In this case, the object would be the cup. Another example of an ADL activity could be opening a door, in which case the door and door handle will be considered as objects.

One of the major contributions of the work presented in this thesis is the ability to select an arbitrary object from a scene image using a P300 paradigm based BCI. The P300 paradigm works by presenting to the user a set of discrete stimuli (for example by highlighting screen regions corresponding to the visual stimuli) and then having the user select one of them. Therefore the object selection mechanism is required to use discrete stimuli.

Object selection can be accomplished in two ways. In the first method, an automated image segmentation algorithm would first try to detect all the likely objects in the scene image. Then the detected and segmented objects would be shown to the user. The user would then be able to select an object from these options. However, image segmentation is a non-trivial problem and therefore, the algorithms are unlikely to select, detect, and segment all the objects present in the image. Hence, this would prevent the user from selecting any objects the segmentation algorithm missed. Previous work by Escolano et al. [17] used this approach. objects were pre-segmented based on color information and these segmented images were presented to the user as visual stimuli for the P300 paradigm. However, this approach will only work for pre-known objects that the automated algorithm is able to identify and segment.

In the second method, the user would be presented with an image and would select an object by selecting a pixel that is part of the desired object. Then an image segmentation algorithm would be used to segment and subsequently identify the object.

Due to the drawbacks present in the first method, our interface uses the second method for object selection. Selecting an object from an image can be accomplished by selecting a pixel in the image that is part of the object and then segmenting that object from the background. In a classical graphical user Interface with conventional input devices such as a pointing device mouse, the most common approach to selecting an object from a scene image would be to click on the object of interest in the image. A pointing device can provide a unique x and y value representing a pixel on the screen, and the x and y values can be indispensably and continuously varied. However, due to the requirement of having discrete stimuli, it is difficult to pick an arbitrary pixel location on the screen. This issue is somewhat simplified when we consider the fact that the user only needs to select "any" pixel that belongs to the object of interest. Due to the discrete nature of the P300 based BCIs, users can only pick from a set of given discrete options. Therefore, it is not possible nor necessary to have a method that would enable the user to select arbitrary pixel positions using a discrete stimulus based BCI system.

4.2 Stimulus Grid and Stimulus Cells

As a solution to the problem mentioned above, a grid based design was proposed and implemented. A scene image is taken from a camera mounted on the robot platform. This image is then divided into grid of m x n cells. The resulting grid is referred to as the "Stimulus Grid" hereafter and each of the cells in the aforementioned stimulus grid will be referred to as a "Stimulus Cell". These "Stimulus cells" are used for stimulus presentation as described later in chapter 6. This stimulus grid is then overlaid on the scene image as shown in 4.1.

For our tests, a grid matrix of size 5x5 was used. A bigger grid size causes longer presentation times and increases fatigue on the user, whereas a smaller grid size causes multiple zooms, increasing the time to select an object.

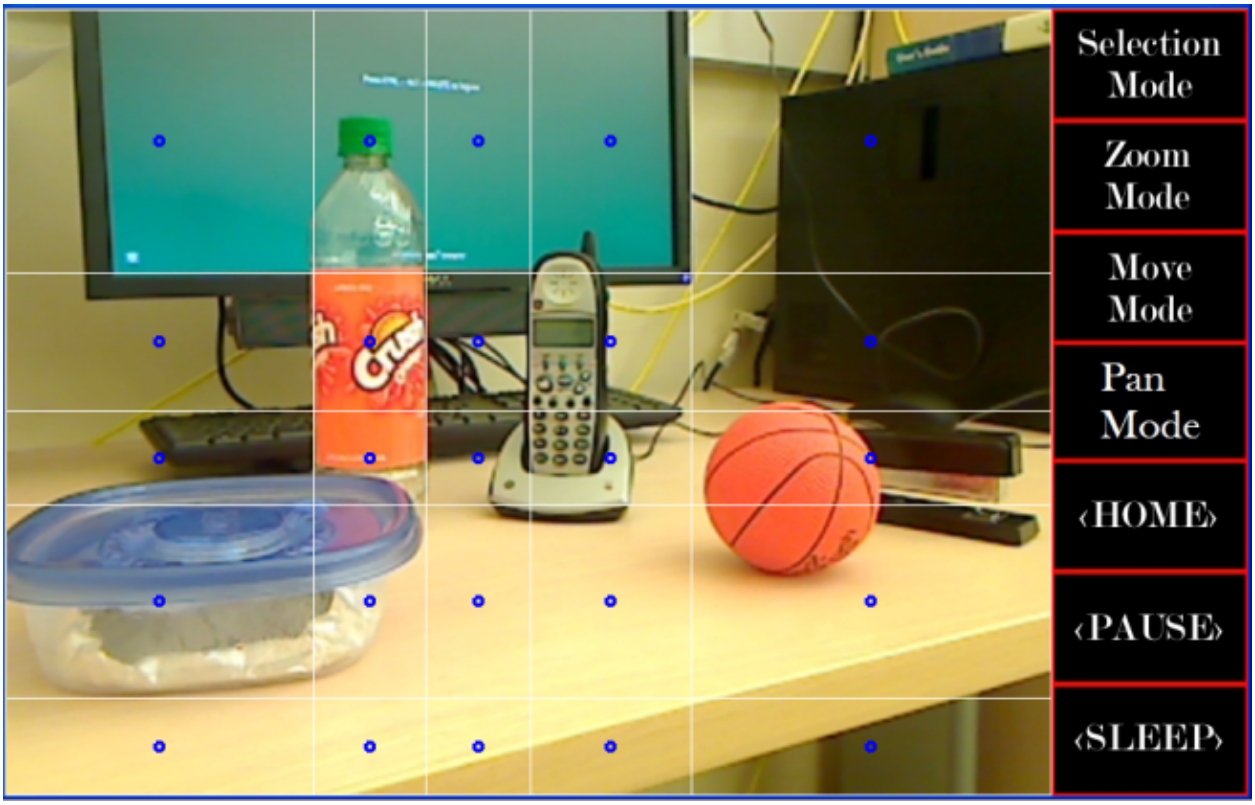


Figure 4.1: Stimulus grid overlaid over the scene image.

4.2.1 Seed Points

At the center of each stimulus cell in this grid is a Seed Point, which is displayed to the user as a small dot (blue dots in 4.1). Selecting a stimulus cell is treated as if the user selected the seed point pixel. This seed point is used as a seed for segmenting the object of interest from the image as described later in Chapter 5. Thus, the user conveys his intention to select an object by selecting a grid element whose seed point is on the object of interest.

4.2.2 Zooming

The ability to zoom into a sub-region of the image is a key feature in our interface. Sometimes, a seed point may not be visible on the desired object as shown in the currently drawn stimulus grid. In this case, a zoom operation will allow the user to recursively zoom into the sub-region of the image that contains the desired object. The zoom operation will continue until a seed point is placed on the desired object. Utilizing this method, it is possible to select any arbitrary pixel

location. However, since it is only necessary to have the seed point somewhere on the desired object, the number of zoom operations necessary is much less in practice. As can be seen in figure 4.2, there are no seed points on the pink colored object (water bottle). Therefore, a zoom operation is necessary to select that object. Figure 4.3 shows the sub-region of the image containing the bottle after it is zoomed into, and another visual grid drawn over the zoomed in area. As can be seen in the same figure, several seed points (shown in blue) are available on the pink water bottle. Selecting any one of these seed points on the water bottle is sufficient to select it.

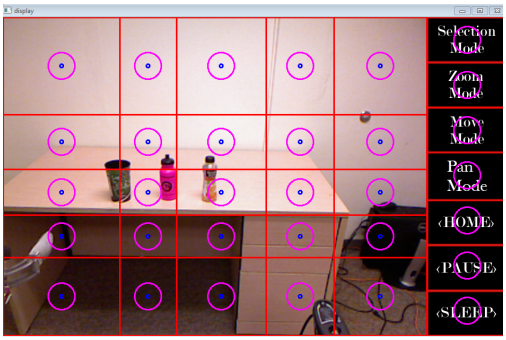


Figure 4.2: Visual grid before zooming

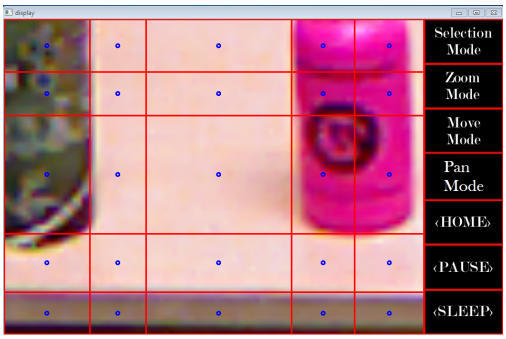


Figure 4.3: Visual grid after zooming into a subregion.

4.3 Adaptive Visual Grid Creation and Cell Sizes

In real-world environments, an object can be located anywhere and the user must be able to select that object. Using a uniform grid would require multiple zoom operations to select an object that might be at any arbitrary location on the image. This would require the user to issue multiple commands through the BCI, which makes the selection process slower and adds more fatigue to

the user. A non-uniform dynamically-sized grid was used to improve the efficiency of selecting an object with the least amount of commands.

Our non-uniform grid construction algorithm uses features obtained from edge detection as a heuristic to draw grid lines along object boundaries. This increases the likelihood of a seed point falling on an object and thus reduces the amount of zoom operations required. The image processing pipeline used for object selection compromises of four main steps as shown in figure 4.4.

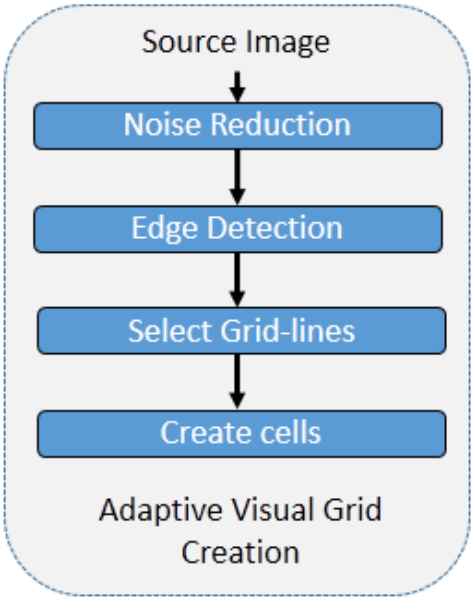


Figure 4.4: Adaptive Visual Grid creation.

1. Noise Reduction
2. Edge Detection
3. Identify horizontal and vertical lines with most edges
4. Draw grid using selected lines

Each of these steps in the algorithm will be explained in more detail next.

4.3.1 Noise Reduction

Any images taken from a sensor can have noise present in the image due to various reasons. These could be sensor noise due to poor illumination , high temperature or other electronic noise

from circuitry. Therefore, a Gaussian smoothing based noise filter was applied to all images as the first image processing step. The output from noise reduction can be seen in figure 4.5.

Figure 4.5: Output after noise filtering. The image on the left shows the source image and the image on the left shows the output after noise filtering.

4.3.2 Edge Detection

In our work, edge detection is used as a heuristic for object boundaries. Edge detection is a method of determining where the brightness in an image changes sharply in neighboring pixels. This change in brightness can have a correlation with object boundaries present in the scene.

4.3.2.1 Background

Brightness or luminescence gradient for an image can be calculated by taking the difference of brightness levels in neighboring pixels. Usually, the horizontal and vertical image gradients are calculated separately, then combined to create an overall gradient magnitude and direction image. For an image I with j rows and k columns, the horizontal gradient is defined by :

$$G_x(i,j) = I(i,j+1) - I(i,j-1) \tag{4.1}$$

where G_x is the matrix containing the horizontal image gradient. Similarly for the vertical Image Gradient, G_y, of the same Image I :

$$G_y(i,j) = I(i+1,j) - I(i-1,j) \tag{4.2}$$

However, using just one row or column to calculate the gradient can be susceptible to noise and sudden brightness changes. For the horizontal gradient calculation, the row above and below of the current row is used. Similarly for the vertical gradient calculation, the column above and below is used. This results in a 2 Dimensional convolution matrix as shown in figure 4.6.

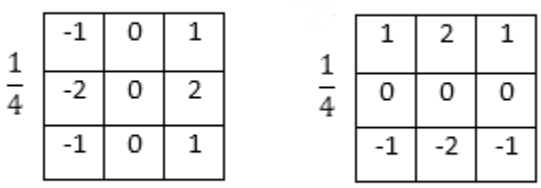


Figure 4.6: Horizontal and vertical Sobel operator convolution matrix.

Then the two horizontal and vertical gradient matrices are combined to produce a matrix containing gradient magnitude and direction.

Gradient Magnitude $G(i, j)$ is given by:

$$G(i, j) = \sqrt{G_x(i, j)^2 * G_y(i, j)^2} \tag{4.3}$$

Similarly, gradient direction at location i, j , $\theta(i, j)$ is given by:

$$\theta(i, j) \tag{4.4}$$

4.3.2.2 Canny Edge Detector

As the above mentioned Sobel operator has some draw backs, such as thick edges and discontinuities, we use the canny edge detector [21]. The canny edge detector improves upon the previously mentioned method by performing several additional steps. First, it calculates diagonal edges in addition to the horizontal and vertical edges.

Second, it performs an edge thinning non max suppression on the previous output [21]. Then the resulting image is threshold-ed using hysteresis with a min and a max value. The output from the canny edge detector is shown in figure 4.7.

Figure 4.7: Output of edge detection step.

4.3.3 Selecting Lines

The output from the edge detection stage is an image I, of m rows and n columns, where:

$$I(m,n) = \begin{cases} 255 & \text{if } I(m,n) \text{ is an edge pixel,} \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

For each row m in this image, the "row sum" is defined to be:

$$R_m = \sum_{i=1}^{n} I(m,i) \tag{4.6}$$

Similarly, for each column C_n, the "column sum" is defined as:

$$C_n = \sum_{i=1}^{m} I(i,n) \tag{4.7}$$

which creates two sets, R_{sums} and C_{sums}, where:

$$R_{sums} = \{R_1, R_2, R_3, ..., R_m\} \tag{4.8}$$

$$C_{sums} = \{C_1, C_2, C_3, ..., R_n\} \tag{4.9}$$

After this point, in order to create an A rows by B columns stimulus grid, the highest ranking $A-1$ items from R_{sums} and the highest ranking $B-1$ items from C_{sums} need to be selected. However, to prevent grid lines being drawn too closely to each other, a minimum separation S is maintained between them. To do this, R_{sums} and C_{sums} are first sorted in descending order. Then, for each set, the first item is added to the selected set. Thereafter, the next item is added to the selected set, only if it is not within a predefined distance S away from any of the other items already in the selected set. This is repeated until the selected set contains $A-1$ and $B-1$ (for the selected row set and the selected column set respectively).

After the vertical and horizontal grid lines are selected, the area separated by them become visual stimulus cells. Visual stimuli cells are explained in more detail in chapter 6. Figure 4.8 shows the selected grid lines overlaid on a the output from the edge detection stage.

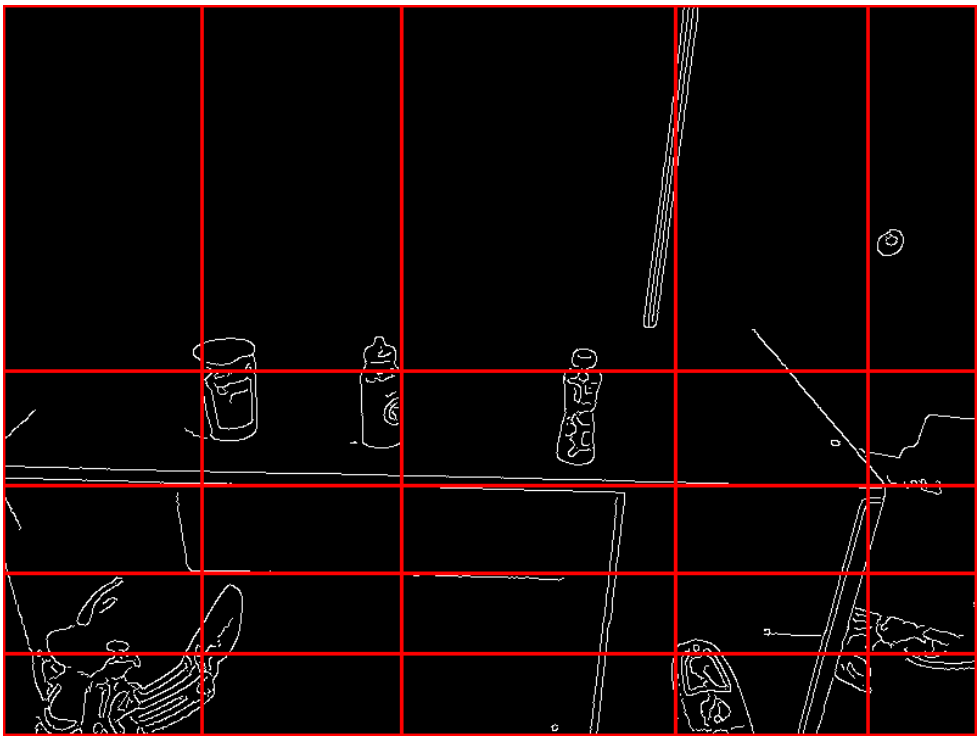


Figure 4.8: Grid selection overlaid on edge image.

CHAPTER 5

OBJECT SEGMENTATION AND IDENTIFICATION

Object segmentation and Identification component of our system is shown in figure 5.1. Once an object is identified, a list of actions that can be performed on the selected object is displayed to the user.

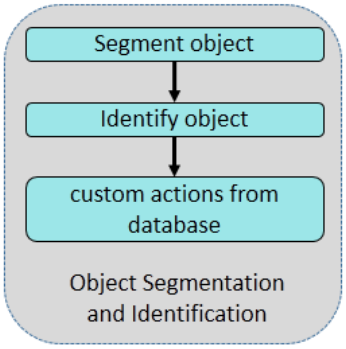


Figure 5.1: Segmentation and identification flowchart.

5.1 Object Segmentation

Object segmentation deals with separating the portion of the image that belong to the object of interest from other regions of the scene image. In this work, object segmentation serves two purposes:

- First, it serves as a pre-processing step for object identification.
- Second, When 3D data is available, it provides the physical location of the object which is used later in the process by the robotic system to manipulate the object.

We receive a seed point (which is a pixel that is part of the selected object) from the previous object selection step. In this section, we will consider two cases for segmentation. The first case

is when a 2D camera is used as the sensor, and therefore, only 2D image data is available. The second case is when 3D depth data is available from a sensor such as a Microsoft Kinect.

5.1.1 Segmentation with 2D Image Data

Humans rely on stereo vision and other high level visual cures to infer depth, combining information from the left and right eyes. Stereo vision works by using two cameras that are separated by a base line distance d. There will be a disparity between the image points formed by the same real world object in the two images. However, object segmentation using 2D image data from a monocular camera is a non trivial problem and is actively researched. One reason for this is, when a real world 3D scene is projected into the image plane of the camera, depth information about those points are lost.

5.1.1.1 Color Based Segmentation

A relatively simple method of segmenting is color based segmentation. As explained in the previous sub-section, the seed point is necessary to segment the object of interest from background regions in the scene image. Using this seed point, a region growing algorithm can be used to segment the object the user selected from the rest of the image. a color based flood-filling algorithm is used to segment the object.

5.1.1.2 Flood Filling

During flood-filling, edge information is used as a boundary mask for the flood-fill algorithm. In flood filling, a seed pixel is selected as the starting point. It is added to the set of selected pixels, then all of it's neighbors are added to a set of candidate points. Then for each point in the candidate set, if the color of the point is within a certain threshold of the seed point, then it is added to the selected set and all its neighbors are added to the candidate set. This process is repeated until the candidate set is empty. Figure 5.2 shows a red cup segmented using the color based flood filling method.

Figure 5.2: A red cup displayed to the user after the segmentation step.

5.1.2 Segmentation with RGB + Depth Data

When a 3D point cloud of the scene is available from a depth sensing camera, the additional geometric information can be used to simplify the segmentation problem greatly. In our current work we only focus on segmenting objects that are located on a large horizontal surface, such as a table. The algorithm for segmenting objects on a table top is as follows:

1. Filter: Point cloud data is filtered with respect to distance from the camera. All points further than 3 meters from the camera are removed. This reduces the amount of data points that need to be processed for the next steps. It further aids by removing walls and other large planer elements that are far away.

2. Find table plane: finding the table plane consists of several steps. First, surface normals are calculated for all points. Then, a random sample consensus method(RANSAC) is used to fit these normals into a plane model. The planer model is represented by the equation $ax + by + cz + d = 0$. The RANSAC algorithm iteratively calculates the parameters of the model $(a, b, candd)$. At the end of RANSAC algorithm, the parameters for the plane of the

table will be represented by the model parameters that fit the inliers. This method assumes that there is a table close to the camera in the scene.

3. Convex hull: The convex hull for the set of all points in the inliers of the previously computed planner model is calculated next. The convex hull is the minimum polygon that includes all the inlier points and the set of all lines between any two points in the inliers.

4. Extract polygonal prism: Using the convex hull previously calculated, a polygonal prism is extracted around the table up to a predefined distance d away from the surface of the table.

5. Remove table plane: The points that are part of the table surface are removed in this step.

6. Euclidean cluster extraction: The remaining points now represent all the objects that were on the table. However, these are still represented as a set of free floating x, y, z points. Therefore, it is necessary to group these into the different clusters (and hence objects) that they represent. This is done with euclidean cluster extraction. According to R. Rusu [22], the algorithm for euclidian cluster extraction is as follows:

First, the coordinates $P_s = [x, y, z]$ corresponding to the seed pixel are extracted from the RGBD image. Then the following procedure is used to check if the seed point is within a δErr distance away from any point in any of the extracted clusters. For each euclidean cluster extracted,

$$E_i = \{p_1, p_2, p_3, ...p_n\} \tag{5.1}$$

where $p_1...p_n$ is all the points that belong to the cluster E_i. Then the euclidean distances from P_s to p_i is calculated by:

$$f(p_i) = \sqrt{(P_s[x] - p_i[x])^2 + (P_s[y] - p_i[y])^2 + (P_s[z] - p_i[z])^2} \tag{5.2}$$

$$D_{min} = min\left(\{(p_1), f(p_2), f(p_3) ..., f(p_n)\}\right) \tag{5.3}$$

If $D_{min} < \delta Err$, then it is the correct cluster. Otherwise, repeat the process with other point clusters.

5.2 Object Identification

In the scope of this work, object identification refers to matching a query image (from the segmented scene image) with a database of training images and to find if there is a match. The

purpose of object identification is to customize the available task options based on the selected object.

Object identification usually depends on unique features that can be used to find a positive match, and should be uniqe enough to have a low probability of false positive matches. Success of object matching also depends on the training data set of images.

5.2.1 Color Based Identification

As a simple method, color based object identification was used during the development of our system. In our color based identification method, the average color for the segmented image portion is calculated (the segmentation can be done using color based flood filling method mentioned previously in section 5.1.1.1). If the set of points that belong to the segmented region is:

$$S = \{p_1, p_2, p_3, ..., p_n\} \tag{5.4}$$

where each point p_i is a tuple $[r, g, b]$ representing red,green and blue values for that pixel. Then for each color, the average Avg value will be calculated as:

$$Avg[r] = \frac{1}{n}\sum_{i=1}^{n} p_i[r] \tag{5.5}$$

$$Avg[g] = \frac{1}{n}\sum_{i=1}^{n} p_i[g] \tag{5.6}$$

$$Avg[b] = \frac{1}{n}\sum_{i=1}^{n} p_i[b] \tag{5.7}$$

This color is then compared to the pre-computed average colors for several objects that were stored in a database. Euclidean distance in RGB color space was used to compare the two colors, if the compared color is C_1 :

$$Dist_{rgb} = \sqrt{(Avg[r] - C_1[r])^2 + (Avg[g] - C_1[g])^2 + (Avg[b] - C_1[b])^2} \tag{5.8}$$

Then:

$$match = \begin{cases} TRUE & Dist_{rgb} < THRESHOLD \\ FALSE & \text{otherwise} \end{cases} \tag{5.9}$$

if the distance is less than a predefined threshold, the colors are determined to be a match.

We found that this method is very susceptible to lighting changes and that it is sometimes necessary to change or adapt the sample color RGB values according to the lighting conditions at the scene. Furthermore, it also limits the number of objects that can be identified, as only one object of each color can be stored in the database.

Due to these shortcomings with color based identification of objects, our current approach uses image feature based identification as described in the next section.

5.2.2 Image Feature Based Identification

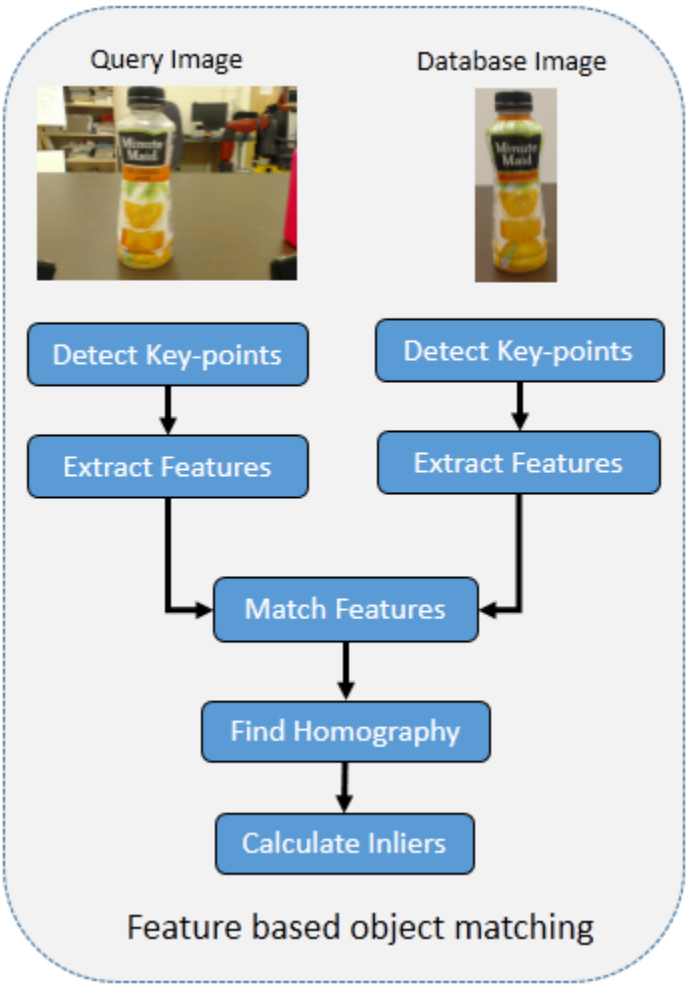


Figure 5.3: Image feature matching.

A feature point is a point in a image that can be easily differentiated from its neighbors. Good feature points should be invariant under lighting changes, scale changes and rotation changes etc.

Therefore, calculating image features and then finding a suitable match can be used to identify objects. As shown in figure 5.3, the algorithm used in this work consists of the following steps:

1. Key Point Detection: In this step, suitable key points from the image are calculated. We use SIFT keypoint detector for keypoint calculation. However, other key point detectors such as SURF [23] and ORB [24] can be used as well. SIFT algorithm was proposed by D.Lowe [25]. SIFT key points are invariant to scale and orientation changes, therefore, are very stable.

2. Feature Extraction: In this step, for every key point extracted in the previous step, a feature descriptor is extracted from the local neighborhood. We used SIFT features for this step as well. SIFT creates a 128 dimensional feature vector.

3. Feature Matching: The previous two steps are done to both the query image as well as the training image. Therefore, we now have two images: a query image and a training image with key points, and the corresponding feature vectors for the key points. The next step is matching feature vectors between the two images. We found several different approaches in the literature to finding matching features. In the first method, for every feature i in the query image, we find the best match from the training set. Then once all the matches are found, the minimum distance for all the matches is calculated. Then any match with a distance that is $3\times$ the minimum distance is discarded as a bad match. The distance is the euclidean distance in the vector space of the appropriate feature vector. In case of sift feature vectors, this is a 128 dimensional vector space. In the second method, for query image feature descriptor i, the two best matches in the training image, $match[i][1]$ and $match[i][2]$ are found. Then Lowe's ratio test [25] is used to determine whether it is a good match or not.

$$match = \begin{cases} ACCEPT & match[i][1] < (0.8 \times match[i][2]) \\ REJECT & \text{otherwise} \end{cases} \tag{5.10}$$

This test is based on the idea that good matches will be unique matches. The ratio between the best match and the second best match is used as a uniqueness indicator.

4. Homography Calculation: The remaining good matches from the previous step is used to check if there is an affine transform of the matched points that would transform one image to the other.A RANSAC based method is used to find the 3×3 affine transformation matrix from one image to the other.

5. Inlier Count: Based on the amount of the inlier points that fit the found affine transformation, a decision is made whether there is a positive match or not. In our work, we use several factors to determine a match: $\frac{inlierCount}{goodMatches} > 0.4$ and $inlierCount > 20$. The value of 20 was empirically determined.

5.2.3 Identification Based on RGBD Data

Object identification accuracy can be improved by using additional depth data provided by an RGBD camera such as the Microsoft Kinect. Because the Kinect camera is mounted approximately 2 meters away from the objects in our current system, the resolution of RGB data at this distance is insufficient for reliable object identification.

5.3 Custom Task Selection

Once the object is identified, a list of tasks (actions) that can be performed on that object will be displayed to the user. This list is dependent on the object and will include custom tasks. For example, in the three objects shown below, the disposable Minute Maid juice bottle has the option to "TRASH" as can be seen in figure 5.4, which will throw it in a trash can in a pre-known location. However, the USF cup, which should not be trashed, does not have such a menu item as can be seen in figure 5.5.

Figure 5.4: Actions that can be performed on Minute Maid bottle.

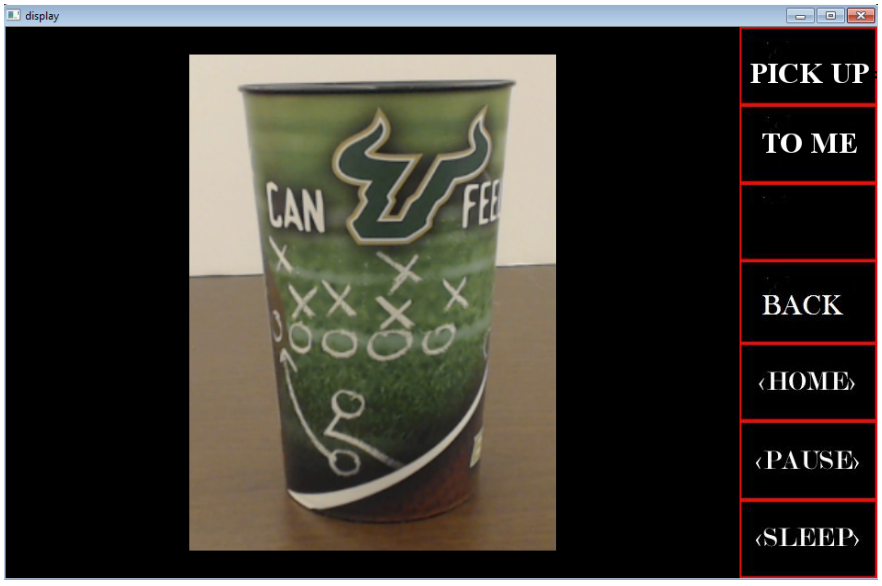


Figure 5.5: Actions that can be performed on USF cup.

CHAPTER 6

STIMULUS PRESENTATION

Stimulus presentation deals with the aspects of the program on how individual stimuli are presented to the user in order to maximize the visually evoked potentials. Also considered are the signal processing and classification techniques that are performed on the EEG data for P300 response detection.

6.1 Visual Stimulus

A visual stimulus is a region on the display screen, that has the potential to invoke a Visually Evoked Potential (VEP), when presented to the user (by stimulus intensification). In our user interface, a visual stimulus is created for each cell in the visual grid described in Chapter 4.

6.2 Modified Randomized Row-Column Based Stimuli Grouping

In traditional P300 based BCI systems, stimuli are presented in groups to reduce presentation time. Usually, a row-column method is used, where stimuli are arranged in a 2D matrix where each row and each column is presented as a group. This row-column paradigm (RCP) was introduced by Farwell and Donchin [26]. For a stimulus grid of (m x n) dimensions, the cells can be displayed in $(m + n)$ groups instead of $(m \times n)$ individual elements, thus reducing the time required for stimulus presentation. The selected stimulus is at the intersection of the row and the column with the highest P300 classification scores.

This basic row-column method works well for programs such as the P300 speller [26], where the elements can be neatly arranged in rows and columns on the display. However, our interface consists of a visual image (stimulus) grid whose dimensions (number of grids as well as grid size) change dynamically, and a fixed size menu. Therefore, we implemented a modified randomized

version of the row-column paradigm. The algorithm (as shown in figure 6.1) consists of five main steps:

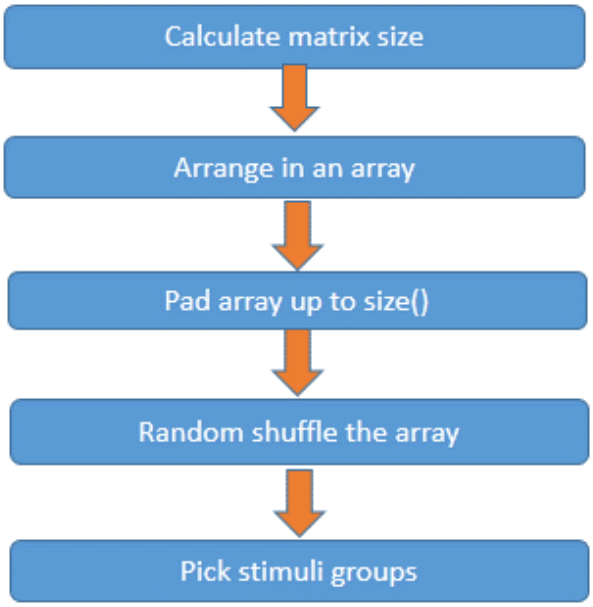


Figure 6.1: Algorithm for generating random stimulus sequences

1. Calculate stimulus matrix size: Depending on the number of active stimuli that is to be presented, a table look-up is used to determine the optimum stimulus matrix size ($m \times n$).

2. Arrange: Arrange the active stimuli in an array.

3. Add Padding: Pad the array with "dummy" visual stimuli to fill it up to ($m \times n$) size. A dummy visual stimuli doesn't have a corresponding display screen region, therefor, it won't be displayed during stimulus presentation. Padding is necessary for the next step of randomization.

4. Random Shuffle: Random shuffle the stimulus array.

5. Pick Groups: Select groups of stimuli. m row groups and n column groups are chosen by indexing into the 1 dimensional array by treating it as a a 2D array. A total of $m + n$ stimulus groups will be chosen.

The result of the above step is that each group of stimuli, when presented to the user, will appear to contain a random group of stimuli. This is a desirable outcome as the p300 response

is usually stronger if the stimulus presentation order appears to be random. A set of such stimuli groups used for presentation is called a stimuli sequence.

6.3 Stimulus Intensification Techniques

In traditional BCI applications such as the P300 speller, the letters and symbols are white against a black background or vice versa. This intensifies the letters, making them stand out. In our application, we have a color scene image as part of the stimuli. Therefore, a single color cannot be used as a stimuli intensification method since there could be an object in the scene image with the same color. This would have a detrimental effect on the P300 event related potential. To overcome this limitation, several other intensification techniques were implemented.

1. Intensify the border of the stimulus cell:

   In this case, only the border of the stimulus was intensified. When adjacent cells that share the same border were intensified, users were confused. Therefore, this modulation did not produce optimal results.

2. Inverting the color of the stimulus cell:

   This method has the benefit of working regardless of the colors present in the stimulus cell as the contrasting colors make them stand out against each other. Preliminary pilot tests showed that this method resulted in better accuracy. The stimulus cells were of different sizes, due to our dynamic grid sizing technique. Because of varying sizes, intensification of larger cells by inverting their color were distracting to users who were trying to focus on smaller neighboring cells. Therefore, to remedy this, a small uniform area around the seed point of each cell was intensified using color inversion.

CHAPTER 7

TASK EXECUTION USING THE WHEELCHAIR MOUNTED ROBOTIC ARM

Task execution consists of manipulating physical objects using a robotic manipulator. We used the Wheelchair Mounted Robotic Arm as described in section 2.2 on page 7 as the robotic manipulator. It is a custom designed 7 Degree of Freedom robotic arm attached to an off the shelf powered wheelchair.

Several steps are involved in performing an object manipulation task. First, the the object locations have to be determined in a suitable coordinate frame. Then if the task is complex, it is broken down into a combination of arm motions. Thereafter, for each arm motion, trajectory planning algorithm is executed to calculate the trajectory for the robotic arm. The output from this stage is a sequence of joint angle positions which are then sent to the robot's motor controller.

7.1 Obtaining Object Location

The location of each object in the WMRA's coordinate frame is required in order for the robot arm to manipulate it. Obtaining object location can be different depending on the sensors used.

If a 2D imaging camera is used, then it is not possible to obtain the object's physical x,y,z coordinates with just the 2D RGB image. Therefore, in our system, if only a 2D camera is in use, the object locations need to be measured beforehand and stored in a database. These locations will later be retrieved from the database when needed for task execution. In such a case, the operation of the system will be limited to a fixed environment (both the wheelchair and the objects). However, a 3D depth camera can be used to obtain the object locations dynamically.

7.1.1 Obtaining Object Location Using a Depth Sensor

The Kinect sensor outputs a point cloud consisting of a set of data points. Each point has a x,y,z 3D location and an RGB value for luminance intensity. When a user selects an object as

described in chapter 4, the object location can be obtained from the segmented point cloud (the subset of points that belong to the selected object) as described in section 5.1.2. Then further parameters about the object, such as its width and height, can be calculated using super quadratic fitting parameter estimation [27].

7.1.2 Transformation from Kinect Frame to WMRA Frame

The 3D point cloud obtained from Kinect is centered on the Kinect's coordinate frame whose origin lies at the image sensor. Object location measurements obtained are in Kinect frame initially. Therefore, it is necessary to convert these points into WMRA coordinate frame in order for the robotic arm to manipulate the objects.

To perform this calculation, we need to find the transformation matrix ${}^{W}_{K}T$ which is the rigid body transform from Kinect frame to WMRA frame. The first step is to obtain two point sets $\{{}^{K}P_i\}$ which represent points in Kinect frame, and $\{{}^{W}P_i\}$ points in wheelchair frame. ${}^{K}P_i$ and ${}^{W}P_i$ represent the same physical point. The coordinate frames can be seen in figure 7.1.

To measure these points, a marker was placed on the robotic arm's gripper. Then the coordinate frame of that marker was obtained using the WMRA's forward kinematics, and the same frame is obtained again using Kinect sensor data.

First, the rigid body transform from Kinect frame to WMRA frame ${}^{W}_{K}T$ is divided into a rotation ${}^{W}_{K}R$ and a translation ${}^{W}_{K}P$ component according to [28].

$$ {}^{W}P_i = {}^{W}_{K}R\,{}^{K}P_i + {}^{W}_{K}P + N_i \tag{7.1} $$

where N_i is measurement noise. The problem then becomes to find ${}^{W}_{K}R$ and ${}^{W}_{K}P$ such that the error is minimized.

$$ E^2 = \sum_{i=1}^{N} \left|\left| {}^{W}P_i - \left({}^{W}_{K}R * {}^{K}P_i + {}^{W}_{K}P\right)\right|\right|^2 \tag{7.2} $$

The solution is obtained using a least square fitting algorithm based on SVD [28].

Once the transformation matrix ${}^{W}_{K}T$ is found, then the object point measurements obtained in Kinect frame denoted ${}^{W}P_{Object}$ can be transformed into WMRA frame coordinates ${}^{K}P_{Object}$ using:

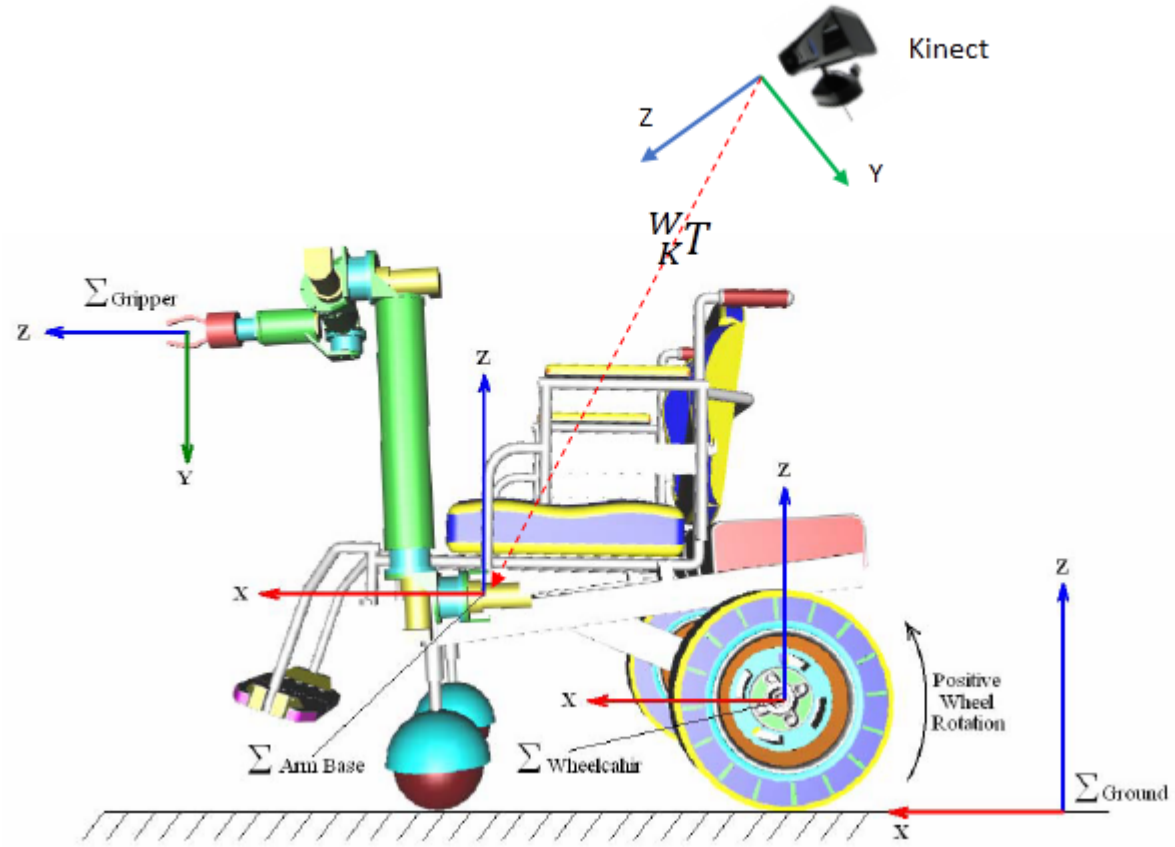


Figure 7.1: WMRA and Kinect coordinate frames

$$^{W}P_{Object} = {}^{W}_{K}T * {}^{K}P_{Object} \tag{7.3}$$

7.2 Tasks

Each complex task is divided into several motion primitives. For example, the "pick-up" action is divided into the following steps:

- Go to pre-grasp pose: This is a pose a few centimeters away from the object and the orientation is pointing towards the object (gripper frame z axis). In this pose, the arm is ready to grasp the object.
- Open gripper: Opens the gripper in preparation to grasp the object.

- Move forward: This step moves the gripper forward into the object with the gripper open.
- Close gripper: Closes the gripper firmly.
- Lift object: This step gently lifts the objects off the table.

7.3 Task Trajectory Planning

Once the object location in WMRA frame is acquired, the task action is divided into a set of smaller sub-actions. For each of these sub-actions, the start and end poses for the end effector are calculated. Thereafter, autonomous motion planning of the WMRA system is used to execute the task.

CHAPTER 8

TESTING AND RESULTS

8.1 Initial Validation of the User Interface

The system was developed in stages. Once the initial BRI user interface was developed, it was tested to validate functionality[1]. The main focus of the initial tests is to evaluate the ability of our Brain Robot Interface (BRI) system to function as a brain computer interface. The tests consisted of asking test participants to select an object using our new interface and then having them select an action to perform on the selected object. However, there were some limitations:

1. There was only one action ("PICK UP") available for all objects.
2. Object locations were fixed and pre-known.
3. Object segmentation and identification were color based.

Once the initial testing discussed in this section was successfully completed, the system was improved to remove the above mentioned limitations. Then, the complete Brain Robot Interface system was tested as described later in section 8.2.

8.1.1 Internal Review Board (IRB) Approval

We obtained approval from the University of South Florida's Internal Review Board to conduct human subject testing. This study was approved under IRB #Pro00005223. Each participant signed an informed consent form.

[1]Portions of this section were published in 2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. Permission is included in Appendix A

8.1.2 Test Equipment and Setup

The test-bed consisted of a 9 degree of freedom (DOF) wheelchair mounted robotic arm (WMRA) system interfaced to the BCI-2000 system . The WMRA comprises of a 7 DOF arm and a 2DOF wheelchair to which the arm is attached. The arm is developed at the Rehabilitation Robotics Lab at the University of South Florida (USF) and made up of an assembly of electric motors and harmonic drives. The WMRA system has optimized Cartesian control based on singularity avoidance and other optimization functions.

A variety of input control devices can be interfaced with the WMRA for autonomous and teleportation motion, including the Brain Robot Interface. The Brain Robot Interface system comprises of an 8-channel electrode cap, a g-tec USBAmp-8 biological signal amplifier and an open source software module called BCI2000 [29]. EEG signals captured by the sensor array on the EEG cap are sent to the amplifier. Then the EEG data is digitized and sent to a computer running the BCI2000 software system for filtering, signal processing and P300 detection. A monocular lens RGB Logitech webcam mounted on the WMRA captures the scene of the environment, which is overlaid with the adaptive stimulus grid and displayed on a monitor. The vision program, arm control program and BCI2000 communicate via UDP sockets. They were run on the same PC. Figure 8.1 shows the actual testbed. The environment used for testing, with multiple objects, is shown in figure 8.3.

8.1.3 Training the P300 Classifier

To improve the accuracy, the Brain Robot Interface system needs to be calibrated for each user. Training the Brain Robot Interface system involves determining participant specific linear classifier weights. Using the P300 speller application (shown in figure 8.2), which comes with the BCI-2000 [14] software system, training data was collected for each participant in the form of raw EEG signals and stimulus timing information. Each participant was asked to spell two 5-letter words in two runs of the speller and each sequence was repeated five times. BCI2000 offline analysis tool was used to perform Step-Wise Linear Discriminant Analysis (SWLDA) on this data and generate individualized feature weights, for the P300 classifier. These were later loaded for each participant, as customized classifier weights to be used in our new vision based Brain Robot Interface.

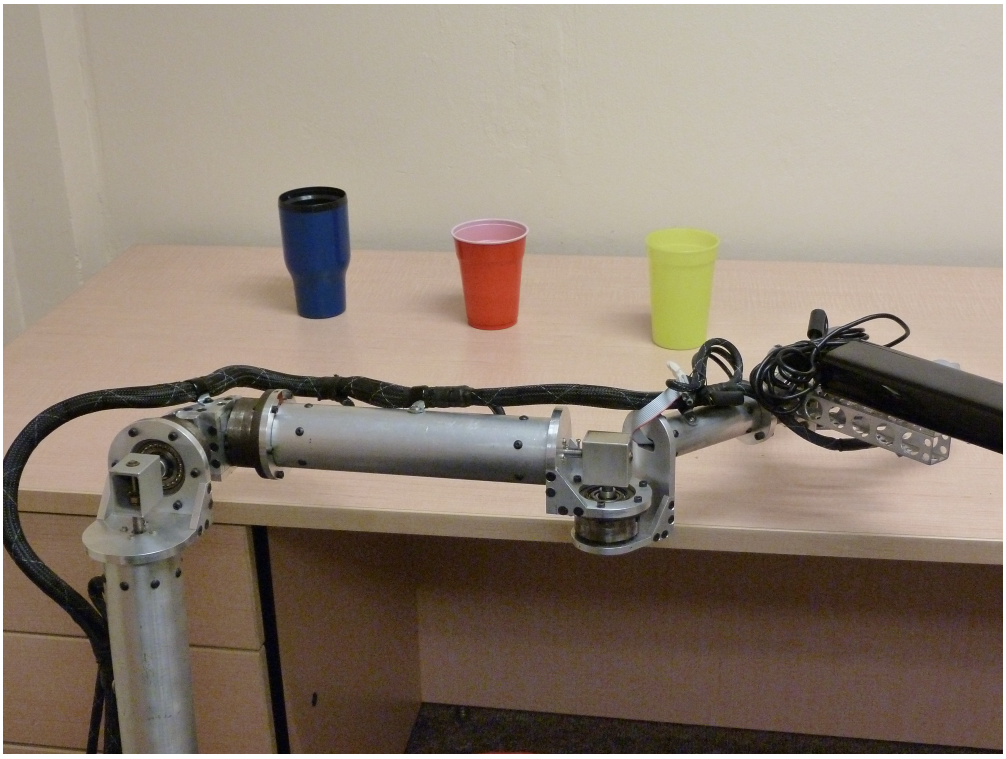


Figure 8.1: Test environment setup

8.1.4 Experimental Methodology

The following parameters were used for stimulus presentation and p300 classification. Stimulus on time was set to 250ms and the inter-stimulus-interval was randomized between 50ms-100ms. Epoch length was set to 500ms.

We tested the system on six able-bodied subjects. There were 5 males and 1 female aged 24 to 40 years. As metrics to validate the system, we determined the average accuracy of the intention detection per subject and the average number of commands it takes per subject to grasp an object. The accuracy was measured as the percentage of times the system detected the correct cell that the user intended from the flashing grid. None of the subjects had earlier experience using or testing on a BCI/BRI system. The users gained familiarity with the BCI P300 system during the training stage described earlier, which lasts 15-20 minutes. After the training stage, they are asked to spell two more words to determine their accuracy on the P300 speller as shown in figure 8.2. This accuracy will be used as a baseline to compare the accuracy of our system. The environment consisted of three cups: blue, yellow, and red colored, and these were kept on a table. Subjects

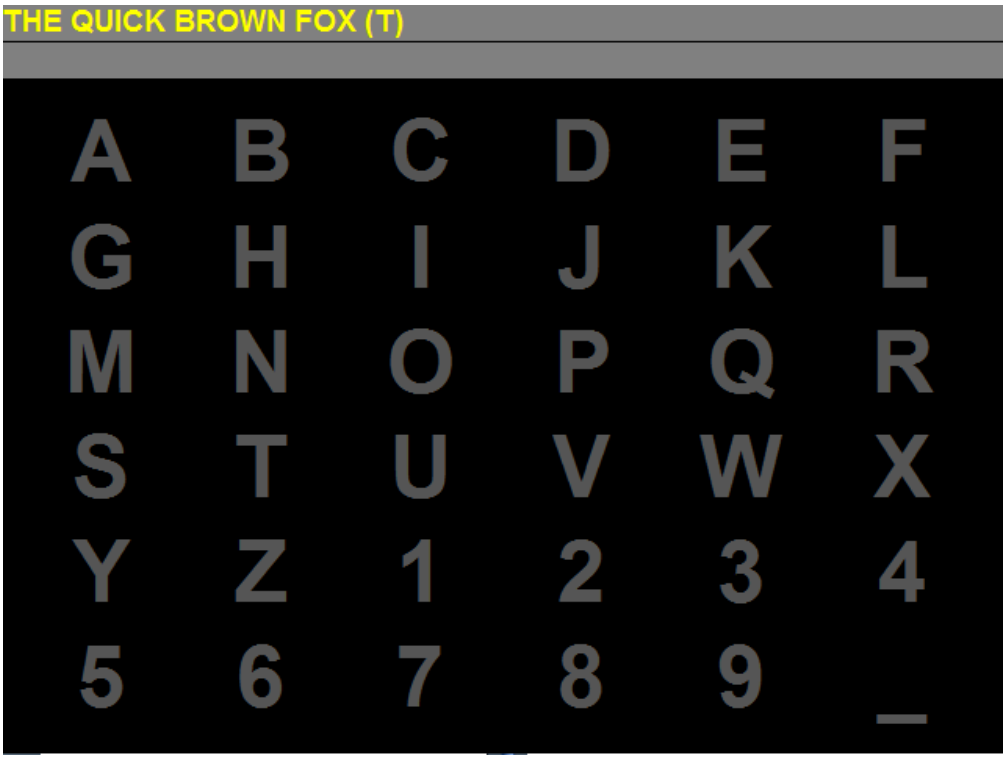


Figure 8.2: Screenshot of P300 speller

were asked to pick up all three cups using the Brain Robot interface. Picking up a cup involved the user issuing multiple commands (selections) via the Brain Robot Interface. Then data were recorded on whether the system detected the intended stimulus correctly or not, and how many commands had to be issued to pick up each cup. From this data, the accuracy of the system for each user and the average number of commands issued to pick up an object were calculated.

8.1.5 Results and Discussion

We measured two key parameters in our testing. These were:

1. Number of commands to perform a task: This is the key metric for the main objective of our system. The number of commands is measures as the number of selections the user need to choose (via the BRI) to perform a complete task.

2. Accuracy: This was defined as the ability of the system to correctly identify the visual stimulus that the user was focusing on. In other words, this is the ability of the system to correctly identify the user intention. This is a key metric as inaccurate selections can

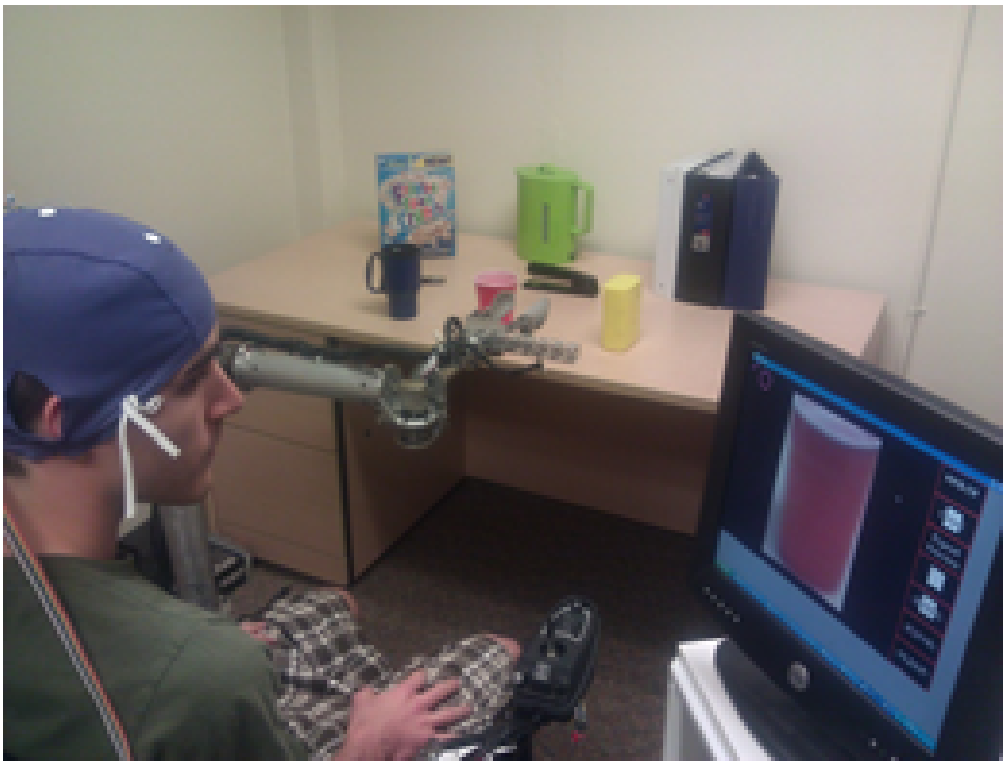


Figure 8.3: Selecting an object through the interface.

drastically reduce the user experience and increase the number of commands that need to be issued to perform a task.

Accuracy results obtained for each subject is shown in figure 8.4. The average accuracy for each subject is the average number of correct stimuli detected by the system in grasping all three cups. We obtained 100% accuracy for three out of six subjects. All the subjects were able to pick all the three cups except subject 4, who failed to pick up the yellow cup within 10 selections. We believe that excessive fatigue was the cause for this. The accuracy average over all the subjects was found to be 85.56%. With the P300 speller application, we found that the users reached an average accuracy of 93.33%. Therefore, the accuracy of our system is comparable to the accuracy of the P300 speller. Additionally, we observed that subjects who achieved good accuracy levels in the p300 speller achieved better accuracy in our system as well.

During testing, we observed that there was a drop in the accuracy that occurred with all the subjects after they had tested with the first two objects. We hypothesized that this drop may be occurring due to the users experiencing fatigue. This may have been because the subjects had

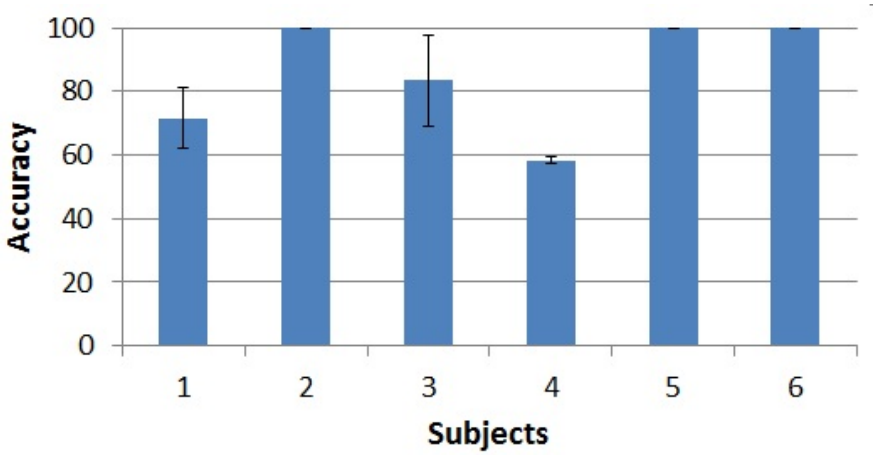


Figure 8.4: Accuracy

to maintain high concentration for long periods of time. Some subjects verbally mentioned that they were fatigued. This is a drawback of P300 based BCI applications since it demands the user to actively attend to the presentation sequences for considerable amounts of time. We gave the participants a fifteen minute break between the second and third cup trials and then resumed the testing. We observed that after resting, the participants were able to achieve accuracy similar to that obtained during the testing for the first two objects. In our testing, we also observed that accuracy dropped when the flashing sequence was such that two neighboring cells flashed simultaneously, one of them being the cell of interest. We believe that this flashing diverted the attention of the participants from the cell of interest and resulted in wrong cell selection by the P300 classifier. The electrode impedances were an important factor in accuracy as well. We observed that when the subjects had high impedance values at the electrode interface, the accuracy was low. When the impedance was around 20 k, good accuracy was achieved. The data we obtained during the testing was collected after calibrating all these factors for consistent readings.

In our BRI system, the user needs to issue a series of commands, in a specific order, to perform a complete task. There is a certain accuracy associated with the selection of each stimulus. The total accuracy of the task is thus the product of the accuracy of each selection. This value decreases as the number of sequences in a task increases.

In conclusion, accuracy depends on a number of factors, such as mental fatigue level of each participant, their ability to concentrate, and external auditory and visual disturbances from the

surrounding environment. The amount of training for each participant, ideal numbers of sequences that lead to the correct detection, and the stimulus intensification method also affect the accuracy of the system.

The average number of commands each user issued to perform the grasping operation is shown in figure 8.5. The average for all subjects is 5 commands. In the ideal situation, our interface would only require two commands from the user. This is the case when a seed point would be present on the desired object in the initial grid. However, the median number of commands was 4, which was the case when a single zoom operation was necessary to obtain a seed point on the object.

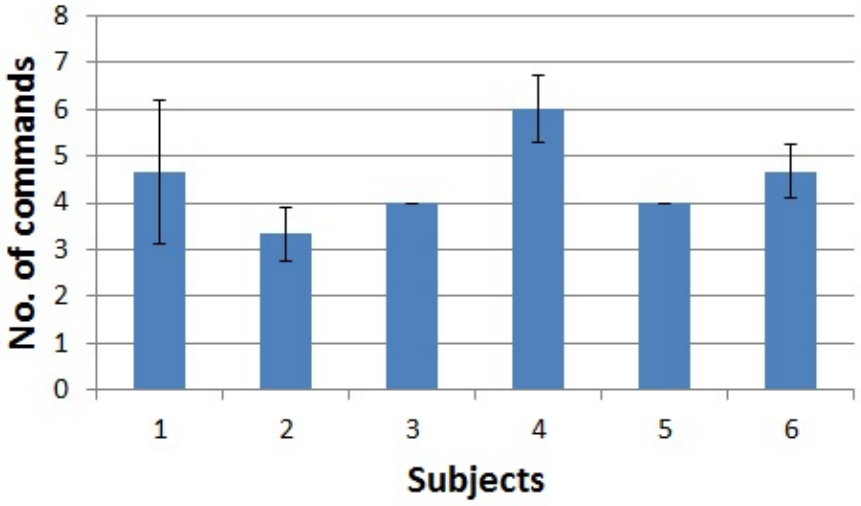


Figure 8.5: Average number of commands for each subject.

The adaptive grid performed well in enabling object selection with minimum amount of zooming. During our tests, only one instance was observed when the participant had to zoom twice in-order to get the seed point on the desired object.

8.2 Testing of the Complete System with Dynamic Object Locations

After the testing mentioned in the previous section, we implemented the ability to obtain object locations dynamically using data from a Kinect RGBD sensor. Furthermore, object segmentation was improved and currently uses 3D point cloud data as explained in section 5.1.2. Also, object matching is now performed using image features as described in section 5.2.2.

This complete BRI system, which includes our grid based object selection, and the WMRA robotic system, was tested as described in this section.

8.2.1 Test Equipment and Setup

The test setup, as shown in figure 8.6, was similar to the setup described in section 8.1.2, except for the changes described next. A Microsoft Kinect RGB+Depth sensor was mounted on the WMRA. It captures a scene image of the environment, which is overlaid with the adaptive stimulus grid and displayed on a monitor. The WMRA control program was run as a separate node and inter-process communication between our visual interface and the WMRA control program was done through network sockets.

The manipulable objects in the environment consisted of three objects. A disposable beverage bottle, a reusable water bottle and a USF cup. Additionally, a trash can in a pre-known location is used as well for the "trash" action described later. We believe having the trash bin in a pre-known location is a reasonable assumption. Figure 8.6 shows the actual testbed objects. The environment used for testing, with multiple objects, is shown in figure 8.6. P300 classifier feature vectors for each subject was obtained as described in section 8.1.3. This study was approved under IRB #*Pro*00005223 as described in section 8.1.1.

Figure 8.6: Test setup with dynamically located objects.

8.2.2 Experimental Methodology

The following parameters were used for stimulus presentation and p300 classification. Stimulus on time was set to 250ms and the inter-stimulus-interval was randomized between 50ms-100ms. Epoch length was set to 500ms.

We tested the system on two able-bodied subjects, both were males. None of the subjects had earlier experience using or testing on a BCI system. They were first introduced to the P300 Speller application, and then a training data was collected in copy spelling mode. Each subject spelled five words ("HELLO", "WORLD", "SOUTH", "FLORIDA", "ROBOTICS") in copy spelling mode. A classifier matrix for each user was calculated using this data as explained in section 8.1.3. This stage took 60 -90 minutes per participant. Then each user was asked to spell the two words ("HELLO", "WORLD") in offline mode to determine their accuracy with the P300 speller. Since the P300 speller [26] is a well known program, we use this accuracy as a baseline to compare to the accuracy we get with our interface.

The rest of the test was conducted on a different day to reduce user fatigue. Each subject was then asked to perform two actions on each of the three objects, for a total of six tasks per participant. This process took around 60-90 minutes per subject. During these tasks, the following parameters were measured:

1. Number of commands to perform a task: This is the key metric for the main objective of our system. The number of commands is measured as the number of selections the user needed to choose (via the BRI) to perform a complete task. For example, a typical series of commands issued to pick up an object with our interface would be:

   (a) select zoom mode

   (b) select image grid cell to zoom into

   (c) select image grid cell containing suitable seed point

   (d) select pickup action

   In the optimum scenario, a seed point will be available on the object on the initially drawn grid. In this case, no zoom operation would be needed, and therefore, object selection would only require selecting the cell with the seed point, hence just one command. Then another

command would be needed to select an action to be performed on the object. Therefore, in the optimum case, our interface would be able to perform a complete task with two BRI/BCI commands.

2. Accuracy: This was defined as the ability of the system to correctly identify the visual stimulus that the user was focusing on. In other words, this is the ability of the system to correctly identify the user's intention. This is a key metric as inaccurate selections can drastically reduce the user experience and increase the number of commands that need to issued to perform a task.

3. Time taken: The time taken to complete a task was measured. This parameter, however, is directly dependent on the the number of commands issued and the time the WMRA system took to perform the task autonomously (which varied between different tasks depending on object locations).

8.2.3 Results and Discussion

The accuracy for each task performed by each of the subjects is shown in figure 8.7. Accuracy for each task was calculated as:

$$accuracy = \frac{correctly\ identified\ stimuli}{number\ of\ commands\ issued\ for\ the\ task} \tag{8.1}$$

In cases of P300 classifier wrongly identifying the intended stimulus, the subject had to issue additional commands to correct, or had to start from the beginning. Both subject 01 and subject 02 reached 100% accuracy for three of the six tasks. The average accuracy for subject 01 was 88.57% and the average accuracy for subject 02 was 88.37%. These are comparable to the accuracy obtained with earlier tests with the P300 speller.

The number of commands needed to perform an action was measured as well. The number of commands consisted of the commands needed to select an object (which depended on how many zoom operations were required), and the commands to select an action (this was always one, except in the case of a wrong stimulus selection). The number of commands that was issued to perform each task is presented in figure 8.8. The average number of commands to perform an ADL task was 4.83 (rounds up to 5) for subject 01 and 4.16 (rounds up to 5) for subject 02. The minimum

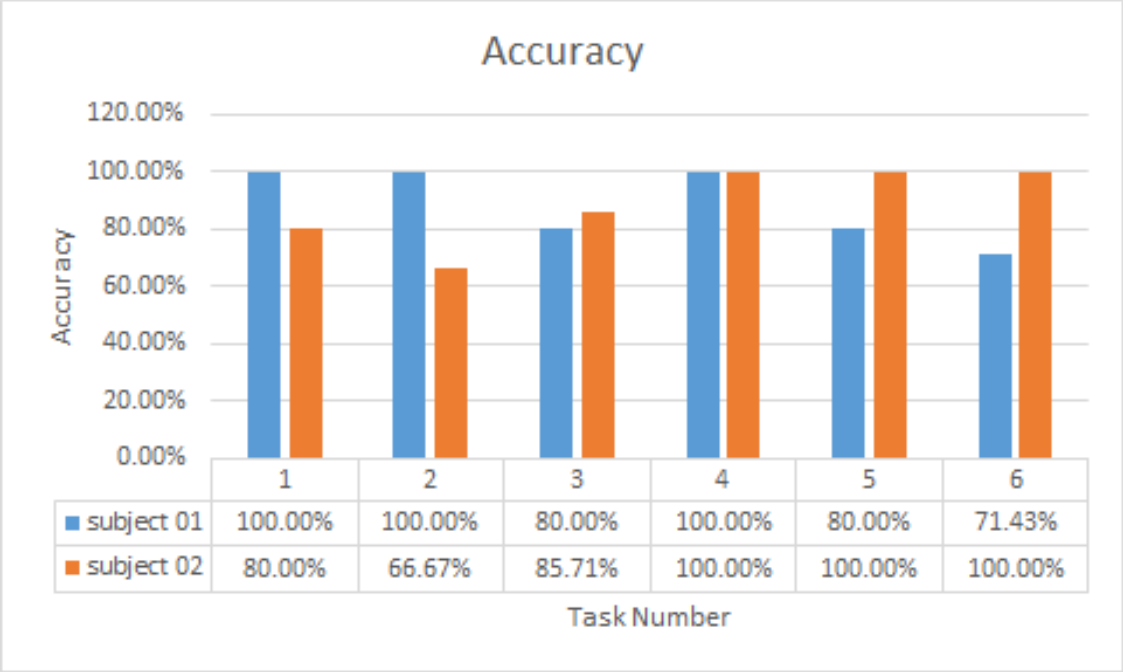


Figure 8.7: Accuracy for each task

number of commands needed was 2, which was the case when a seed point was present in the initial visual stimulus grid, and therefore, no zoom operation was needed. The user only needed to select the appropriate cell and then select an action, which results in two selections. This situation can be seen in task 5 for subject 02 in figure 8.8. When the P300 classifier fails to identify the correct stimulus, the user had to perform additional commands to either correct the wrong selection or start the selection process from the beginning.

The other aspect we measured was the time it took to issue all the commands necessary to perform an ADL task. Selecting a command from the object selection stage took 20 seconds with the P300 stimulus parameters used in test setup. Selecting an action from the menu took only 12 seconds because there were only 7 stimuli (options) to select from. Figure 8.9 shows the time taken to perform each task by each person. These times do not include the amount of time the robot took to perform the action. The time the robot took is fixed between trials and does not depend on the user, but depends only on the motion speed of the robot, and the task it is performing. More complex tasks, such as "BRING TO ME", which require the arm to travel a greater distance, require more time, and simple tasks, such as "PICK UP", require less time.

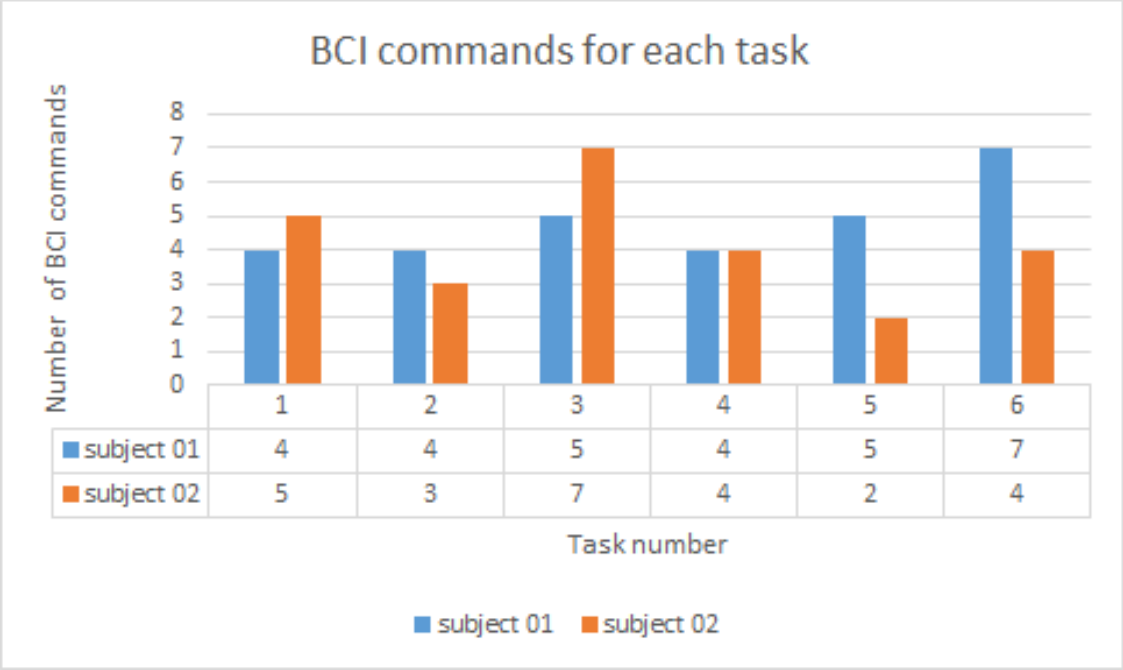


Figure 8.8: Number of commands for each task

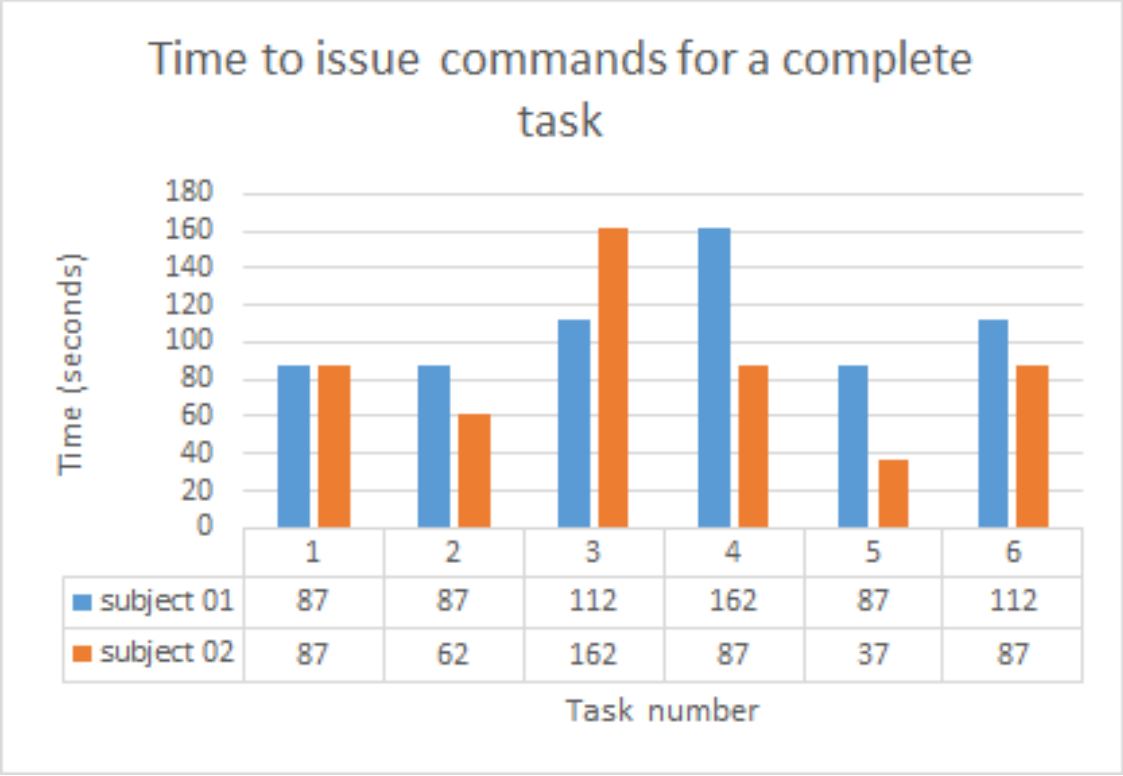


Figure 8.9: Time taken to issue commands for each task

Average time for subject 01 was 107.8 seconds while the average time for subject 02 was 87 seconds. This time could be further improved by reducing the number of times the stimulus was presented to the user. For example, in our test setup, each stimulus group was presented 7 times and the average response was taken. However, reducing this number might have a negative effect on the ability of the system to accurately identify the intended stimulus due to noise and other disturbances.

The WMRA system took around 60 seconds to perform the "BRING TO ME" task autonomously. Compared to this time, the average time taken to issue the command was 97.4 seconds.

CHAPTER 9

CONCLUSIONS AND RECOMMENDATIONS

In this work, we have demonstrated a novel vision-based non-invasive Brain Robot Interface optimized for performing activities of daily living tasks. Our interface is designed to reduce the cognitive load on the user and the mental fatigue it causes when using a traditional P300 BCI. We approach this problem by trying to minimize the number of BCI/BRI selections a user need to do in order to perform an action. To this end, our interface allows the user to perform a task by issuing task level (high level) commands as opposed to low level directional commands.

The main component of our Brain Robot Interface is the adaptive visual stimuli grid, that allowed for efficient selection of objects using the P300 BCI paradigm, which require discrete stimuli. The Adaptive Grid for Efficient Object Selection (AGEOS) takes a scene image of the user's environment and enables the user to select any object visible on the scene image. This leads to intuitive control for the user, as the user is directly focusing on the object. Furthermore, our interface allows a user to select an object with less number of commands issued via the brain computer interface, when compared to the (low level) directional control methods. This reduces the cognitive load on the user, which in turn reduces mental fatigue. To the best of our knowledge, there are no other existing BCI systems that are capable of selecting objects from an environment using the P300 paradigm.

We also presented a method of displaying custom actions based on the selected object. To achieve this, it is necessary to segment the selected object and then identify the selected object. Therefore, we also presented methods of object segmentation and identification in this thesis. The presented methods for object segmentation and identification include algorithms that work with 2 dimensional RGB image data as well as 3 dimensional RGBD data. As discussed in Chapter 8, we demonstrated the functionality of both the 2D and 3D algorithms successfully. Furthermore,

as object location is obtained dynamically from the scene, our interface does not depend on static object positions.

Moreover, we also presented a stimulus presentation method derived from the P300 row column paradigm that is suitable for our interface. We then tested the stimulus presentation with test subjects and compared the accuracy of our system to the accuracy we obtained from the P300 speller application. As described in Chapter 8, the accuracy of our system is comparable to the P300 speller.

Finally, we tested our complete Brain Robot Interface system with human subjects, and demonstrated the functionality of the complete system. The test participants successfully completed several ADL tasks. The average number of commands issued to perform a complete ADL task was 5. The average time taken to issue all the commands was 100 seconds. We believe that we achieved the goal of our work, which was to reduce the cognitive load on the user and increase the efficiency of performing ADL tasks.

There are several areas where our system can be further improved. The current P300 classification used in our system works in asynchronous mode. If the user pauses the stimulus presentation, it is not possible to resume again via the BCI/BRI. Therefore, implementing an asynchronous approach would improve the user experience. The adaptive grid creation algorithm can also be improved. Also, in future work, more action items can be made available for the user to choose from. If a large number of actions can be performed on an object, machine learning approaches can be used to detect user preferences from previous selections, and then display the most relevant actions. We will look into implementing finer manipulation and contact determination algorithms to enable users to, say, drink from a cup as an example.

LIST OF REFERENCES

[1] US Census Bureau. Disability Among the Working Age Population : 2008 and 2009. (September), 2010.

[2] S Katz and C A Akpom. A measure of primary sociobiological functions. *International journal of health services : planning, administration, evaluation*, 6(3):493–508, 1976.

[3] LP Rowland and NA Shneider. Amyotrophic lateral sclerosis. *New England Journal of Medicine*, 344(22):1688–1700, 2001.

[4] Eric W Sellers and Emanuel Donchin. A P300-based brain-computer interface: initial tests by ALS patients. *Clinical neurophysiology : official journal of the International Federation of Clinical Neurophysiology*, 117(3):538–48, March 2006.

[5] Indika Pathirage, Karan Khokar, Elijah Klay, Redwan Alqasemi, and Rajiv Dubey. A vision based P300 Brain Computer Interface for grasping using a wheelchair-mounted robotic arm. *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 188–193, July 2013.

[6] Niels Birbaumer. Breaking the silence: brain-computer interfaces (BCI) for communication and motor control. *Psychophysiology*, 43(6):517–32, November 2006.

[7] Vadim S Polikov, Patrick a Tresco, and William M Reichert. Response of brain tissue to chronically implanted neural electrodes. *Journal of neuroscience methods*, 148(1):1–18, October 2005.

[8] H Jasper. Report of the committee on methods of clinical examination in electroencephalography. *Electroenceph. Clin. Neurophysiol.*, pages 370–375, 1958.

[9] C P Mason and E Peizer. Medical manipulator for quadriplegic. In *Proc Intl Conf. on Telemanipulators for the Physically Handicapped*, 1978.

[10] Redwan Alqasemi and Rajiv Dubey. Combined mobility and manipulation control of a newly developed 9-DOF wheelchair-mounted robotic arm system. *Robotics and Automation, 2007 IEEE*, (April):10–14, 2007.

[11] Jonathan R Wolpaw, Niels Birbaumer, Dennis J McFarland, Gert Pfurtscheller, and Theresa M Vaughan. Brain-computer interfaces for communication and control. *Clinical neurophysiology : official journal of the International Federation of Clinical Neurophysiology*, 113(6):767–91, July 2002.

[12] J J Del R Millan, F Galan, D Vanhooydonck, E Lew, J Philips, and M Nuttin. Asynchronous non-invasive brain-actuated control of an intelligent wheelchair. *Conference Proceedings of the International Conference of IEEE Engineering in Medicine and Biology Society*, 2009(9):3361–3364, 2009.

[13] L Tonin, R Leeb, M Tavella, S Perdikis, and J R del Millan. The role of shared-control in BCI-based telepresence. *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, pages 1462–1466, October 2010.

[14] Brice Rebsamen, Etienne Burdet, Cuntai Guan, Haihong Zhang, Chee Leong Teo, Qiang Zeng, Christian Laugier, and Marcelo H Ang Jr. Controlling a Wheelchair Indoors Using Thought. *IEEE Intelligent Systems*, 22(2):18–24, 2007.

[15] I Iturrate and JM Antelis. A noninvasive brain-actuated wheelchair based on a P300 neurophysiological protocol and automated navigation. *IEEE Transactions on Robotics*, 25(3):614–627, 2009.

[16] Christian J. Bell, Pradeep Shenoy, Rawichote Chalodhorn, and Rajesh P.N. Rao. An Image-based Brain-Computer Interface Using the P3 Response. In *2007 3rd International IEEE/EMBS Conference on Neural Engineering*, pages 318–321. Ieee, May 2007.

[17] Carlos Escolano, Javier Mauricio Antelis, and Javier Minguez. A Telepresence Mobile Robot Controlled With a Noninvasive Brain-Computer Interface. *IEEE transactions on systems man and cybernetics Part B Cybernetics a publication of the IEEE Systems Man and Cybernetics Society*, 42(99):1–12, 2011.

[18] M Palankar. Control of a 9-DoF wheelchair-mounted robotic arm system using a P300 brain computer interface: Initial experiments. In *2008 IEEE International Conference on Robotics and Biomimetics*, pages 348–353. Ieee, 2009.

[19] RM Alqasemi. *Maximizing manipulation capabilities of persons with disabilities using a smart 9-degree-of-freedom wheelchair-mounted robotic arm system.* Dissertation, University of South Florida, 2007.

[20] Gabriel Pires, Miguel Castelo-Branco, and Urbano Nunes. Visual P300-based BCI to steer a wheelchair: A Bayesian approach. In *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 2008, pages 658–61, 2008.

[21] J Canny. A computational approach to edge detection. *IEEE transactions on pattern analysis and machine intelligence*, 8(6):679–98, June 1986.

[22] Radu Bogdan Rusu. *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments.* PhD thesis, der Technischen Universität München, 2009.

[23] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF : Speeded Up Robust Features. pages 404–417, 2006.

[24] Ethan Rublee and V Rabaud. ORB: an efficient alternative to SIFT or SURF. *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564–2571, 2011.

[25] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

[26] L a Farwell and E Donchin. Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and clinical neurophysiology*, 70(6):510–23, December 1988.

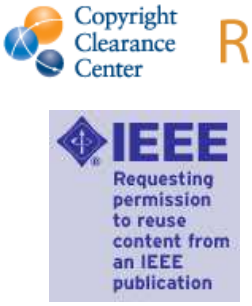
[27] Kester Duncan and Sudeep Sarkar. Multi-scale superquadric fitting for efficient shape and pose recovery of unknown objects. *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 2013.

[28] K S Arun, T S Huang, and S D Blostein. Least-Squares Fitting of Two 3-D Point Sets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-9(5):698–700, September 1987.

[29] Gerwin Schalk, Dennis J McFarland, Thilo Hinterberger, Niels Birbaumer, and Jonathan R Wolpaw. BCI2000: a general-purpose brain-computer interface (BCI) system. *IEEE transactions on bio-medical engineering*, 51(6):1034–43, July 2004.

APPENDICES

Appendix A Copyright Permissions



Copyright Clearance Center RightsLink®

Home | Create Account | Help

Title:	A vision based P300 Brain Computer Interface for grasping using a wheelchair-mounted robotic arm
Conference Proceedings:	Advanced Intelligent Mechatronics (AIM), 2013 IEEE/ASME International Conference on
Author:	Pathirage, I.; Khokar, K.; Klay, E.; Alqasemi, R.; Dubey, R.
Publisher:	IEEE
Date:	9-12 July 2013

Copyright © 2013, IEEE

User ID
Password
Enable Auto Login
LOGIN
Forgot Password/User ID?

If you're a copyright.com user, you can login to RightsLink using your copyright.com credentials. Already **a RightsLink user** or want to learn more?

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK | CLOSE WINDOW