Alternative Pick Device for Tetraplegics Using a Kinect

Thesis by Andrew Ruggeri, under the guidance of Douglas Lyon Ph.D.

Fairfield University
School of Engineering
Masters of Electrical & Computer Engineering
May 2016

Table of Contents

1 Introduction

1.1 Abstract

We describe the design and construction of a new tetraplegic computer input device based on the Microsoft Kinect. The system is unconventional in that it does not require the dexterity of hand and arm movements needed for a conventional mouse, yet acts like a standard mouse to the computer. Outlined within this text are two possible approaches which could be used to accomplish this goal. Ultimately, only the better of the two approaches was chosen and is detailed in depth. The system used by the chosen approach uses the visual and audio information from the Kinect to track the user's head movement as well as, listen for verbal cues. Using this information, the system can act as a pick device by moving as the user's head moves and trigging 'click's upon verbal command.

1.2 Problem Statement

We are given a person who are afflicted by tetraplegia, also known as quadriplegia, suffer from complete or partial paralysis of their torso and limbs and, that common computer user input is dependent upon the use of keyboard and mouse. These two devices are hand and arm centric, causing computer usage to be problematic for a person with paralysis in their upper limbs. The goal is to create a pick device subject to being controlled by no part of the body bellow the neck.

1.3 Approach

This section addresses two different approaches on how to implement a human to device interface for accomplishing the given problem statement. There are two approaches mentioned in the following subsections, both of the approaches are focused around using various existing standards and off-the-shelf electronics. This idea helps stimulate openness and modulation. Section 1.3.1, outlines a wireless approach using Bluetooth as the interface to the target device. The following section, 1.3.2, describes a completely wired system in which the target device is connected to with USB. Each approach has its own strengths and drawback which are discussed in their relative sections bellow. For the final system's implementation Approach 2 was chosen due to hardware limitation concerns of Approach 1.

1.3.1 Approach 1: Wireless

This approach consists of using embedded computers and Bluetooth as the primary blocks in the system configuration However, this approach was only considered and never fully implemented. In this approach Kinect is connected via Universal Serial Bus (USB) to an intermediate microcontroller, an ODROID-C1. This intermediate microcontroller communicates with the Kinect, which provides the processed tracking and audio data. The spatial information is filter to only focus on head and neck movements which are correlated to the 'x' and 'y' movements of the mouse. Mouse click events are triggered by audio keywords which can be user set via settings file. The mouse movement and events are set out via USB On-The-Go (OTG) set as a client and enumerated as a Human Interface Device (HID). This allows for driverless cross platform use on the target computer. Along with this a Bluetooth module is attached via serial communication as well and is identified as a HID device. This method involves addition parts such as the ODROID-C1 and the Bluetooth module. The additional parts cause an increase in cost and complexity to the project however, these additional parts have the advantage of increasing the adaptability of the project with a greater variety of target devices (for example tablets which feature Bluetooth but no USB host input).

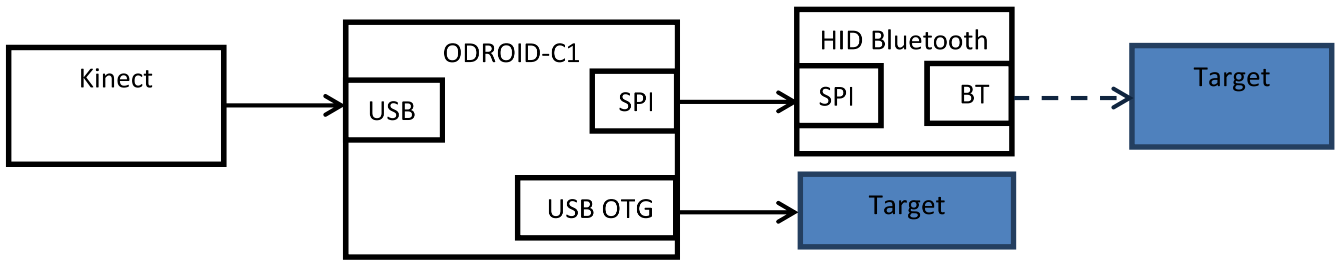

Figure 1-1: Option 1

1.3.2 Approach 2: Selected Approach, Wired

In this approach outlined in this section, the Kinect is directly attached to the target hardware through a chain of devices. This is the most straight forward and simple approach as well as being the most cost effective. The Kinect is connected to a host computer which preforms the tracking. Basic information about mouse movements and commands are then sent to a microcontroller. The microcontroller is connected to the target devices via USB which is enumerated as a HID mouse. The target device only receives a stream on HID mouse actions which it will then preform as if system as a tradition mouse. The target device via USB to the target's host USB port. On the target device software interacts with the kinetic depth and audio sensors and interprets them to mouse movements and pick functions. This was the chosen approach, as it offered a few key advantages. The first advantage is no new computer device is needed unlike approach 1 which required an embedded computer. Secondly, since the target device is connected to a HID mouse, the target does not require any device drivers to be installed. Lastly, the main advantage is there are no performance concerns in this approach compared to those uses an embedded solution with significantly more limited resources. However, this method of interfacing with the target device does limit the adaptability of the device as it is limited to targets with USB interfaces rather than Approach 1 which uses Bluetooth.

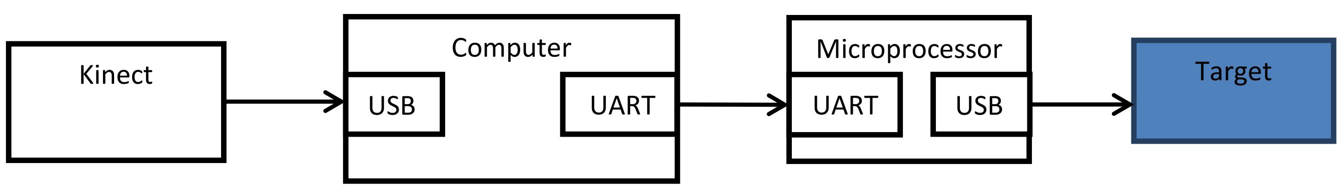

Figure 1-2:Approach 2

1.4 Motivation

Each year there it is estimated that 12,000 people [1] will have an accident which leaves them with spine and nerve damage. Currently there are a total of 250,000 people currently living with spine damage [1]. Quadriplegics must undergo a complete rearrangement of their life style in regards to the way they interact with the world. Even though there is a limited target audience for such, specialty equipment such as this could be sold at a premium price. Both the approaches require low cost equipment to be used. Approach option 1 has the highest cost but could be sold for upwards of 200USD (which would be about more than four times the material cost), which is not unreasonable as similar device range from 500USD [2] to 1000USD [3]. A device such as this aims to help empower such people and promote independence and greater sense of normality. This idea or parts of this idea, could be reused or refocused to help other groups of people who are mobility impaired in other ways. The work here could be taken and embellished upon to encompass Keyboard HID as well providing a full hands-free interface which leads way to numerous uses for all types of people.

1.5 Marketability

Alternative pick devices are a marketable commodity, this can be shown by looking at two vital points for selling a product. The first point being is there a market for the product. This means are there people of who would be interested in product. The second point is, is there a need or demand for the product.

To address the first point let's look at the key demographic who would make up the people most interested in an alternative pick-device. The target market for a device such as this, would be those who suffer from tetraplegia. As discussed in the prior section, it is estimated that 12,000 people yearly will have an accident which leaves them with spine and nerve damage [1]. Further estimates indicate that there are about a quarter of a million people currently inflicted with some form of spine damage [1]. Both these statistics show that there is indeed a good sized demographic suitable for this device

Digital computing devices become more and more part of our daily lives, this can be seen looking at sales trends of devices such as tablets, laptops, and cell phones. Each year sees increases in sales of these devices [4] indicating a growing popularity and dependence on these devices. People who suffer from spinal injuries will undoubtedly want to interact with these devices as well. This plays into the second point of demand, as an alternative pick-device would help people interact with their computing devices.

A head-movement base pick-device has the two key aspects for a viable product, it has a large target demographic and along with that the demographic in question has a need for the device. Furthermore if we look at what's available currently we can see there are already several companies that exist solely based on these devices [2, 5]. This validate that there is a market for such a device.

There is an important distinction which sets this device far ahead of its competitors. Most other devices in this area have three major draw backs: cost, ease of use, and invasiveness for the user. Other devices cost between 500USD to 1000USD [2, 3], by using commercial and open source productions the cost of this device could be significantly lower. The second point of conveyance stems from the tracking method other products use as they require the user to place some type of tracking 'marker' on their body. The Kinect used in this device does not require the user to adhere anything to their body.

1.6 Ethics

A device such as this is deeply electronic and software based thus, it falls under the IEEE's code of ethics. The device itself does not pose any direct safety concerns as, all pieces are consumer-grade electronics and have undergone intensive safety testing by their OEMs (Original Equipment Manufacturer). The primary focus of this project is on improving the lives of anyone who may have impaired or limited dexterity in their limbs and is looking for a way to interact with electronic devices. A focus such as this hits on several key IEEE ethic points as it is improving the lives by developing and enhancing several technologies.

The IEEE's code of ethics was chosen over the Biomedical Engineering Society's BMES's code of ethics as this is not a medical device. To understand why this is not a medical device, a definition for what constitutes a medical device is needed for this the Federal Food and Drug Administration's (FDA) definition will be used. The FDA states their definition as: "an instrument, apparatus, implement, machine, contrivance, implant, in vitro reagent, or other similar or related article, including a component part, or accessory" [6]. This is a pretty broad definition; they further continue to narrow the scope of what signifies a medical device by limiting to three areas.

> "[A device which is] recognized in the official National Formulary, or the United States Pharmacopoeia, or any supplement to them"[6]
>
> "[A device which is] intended for use in the diagnosis of disease or other conditions, or in the cure, mitigation, treatment, or prevention of disease, in man or other animals" [6]
>
> "[A device which is] intended to affect the structure or any function of the body of man or other animals, and which does not achieve its primary intended purposes through chemical action within or on the body of man or other animals and which is not dependent upon being metabolized for the achievement of any of its primary intended purposes." [6]

The first point is easily ruled out as this device is not pharmacological. Point number two is not relevant as, this device cannot diagnose nor can using it improve one's neurological condition. Lastly this device does not alter the structure of the body nor does it affect the functioning of the human body. Quod erat demonstrandum, this is not a medical device as it does not meet the FDA's outlined description.

Practical usage of the device can also be a concern when compared to a tradition pick device such as a mouse. The Kinect has several limitations such as the depth values returned from the sensor are non-linear in their correlation to measured depth [7, 8]. This means depending on the user's distance from the Kinect, the cursor's sensitivity can vary. Further depth sensor limitations are that the Kinect must be placed more than 2 feet away from the user to allow a meaning measurement to be taken [7, 9]. Furthermore, typical traditional mouse latency is around 20ms, the Kinect being used with the official Microsoft drivers has a latency of almost 5x that at 106ms [10]. This limit the user's interaction with the computer as any interactions that require real-time interfacing may not be possible. Further ethical concerns arise if the user is dependent on the functionality of this device. Absolute reliability and functionality of this device, like all devices, cannot be guaranteed. This put greater strain on the quality control this device should be subject to, thus to ensure this device is of high reliability.

Similarly, to all devices there are concerns on how the device will be used and operated. A device such as the Kinect is able to capture and process a wide range of sensor data. In the context of this feasibility test of using the Kinect to monitor head and neck movements the user is aware of being monitored and of the data being collected. However, the application of this device could be used in such a way in which someone could be unknowingly monitored. The use of standards such as Bluetooth and USB and readily available API's contribute to the openness of this project. It allows an easy way for people to adapt this project to a new use whether that be virtuous or one of a more concerning nature. The use of using a HID protocol for interaction with the computer also poses a potential problem as vulnerabilities and exploitations of HID have been used in the past.

As one approach mentioned the use of a Bluetooth interface, the device should adhere to the regulations governing wireless communications. FCC regulations will be abided by in this project as they are set of standard for the US (where this project is taking place). If this project is to be used outside of the U.S. local government wireless regulations should be noted before you. Failure of observing to such regulations can cause interference with other wireless device or disciplinary actions.

1.7 Verification

Testing and verification will ultimately come down to a usage test, in which the functionality of the device is compared to the functionality of a tradition pick device. Can the test device preform the same role, such as movement and selection via pick or 'click'. Further verification can be supplied by visual inspection of the sensor’s output to a graphic computer window. The raw depth data as well as the processed depth data can be reviewed to ensure proper tracking and analysis of movements. The mouse data being sent to the computer can be view in a number of different ways as well. Mouse data being sent via Bluetooth can be view by a program such as Wireshark [11, 12]. Data being sent directly through USB can be view by debugging the software and monitoring what is being sent or by debugging on the host target and reviewing what is being received.

2 Milestones

2.1 Schedule

The project has been broken up into several parts threaded together in a way that one will lead into the next. The two main long term task are research and the overall thesis as they both are evolving over the duration of the project. Building up the software needs to start with first interfacing and then controlling the Kinect. Once basic communication with the Kinect has been setup the next part will be to start working with the data coming from the Kinect and using it in a useful manner, extra time is factor in for minor adjustments and tweaking as well. After movement tracking and processing is nearing completion audio processing comes underway and this too has some add buffer time for adjustments. The final part is having all that data integrated as a pick device. During software development as well as after there is an ongoing task of testing and validation.

The table below shows the task, in the rows, and duration in weeks, in the columns. The header on the column shows the starting date of all cells in that column. For example, if we look at timing for task 'Movement Tracking' it is the second row and starts in the January 25th column. The bar is 3 cells in length meaning it is predicted to take 3 weeks and end on February 14th (on day before the start of the next week).

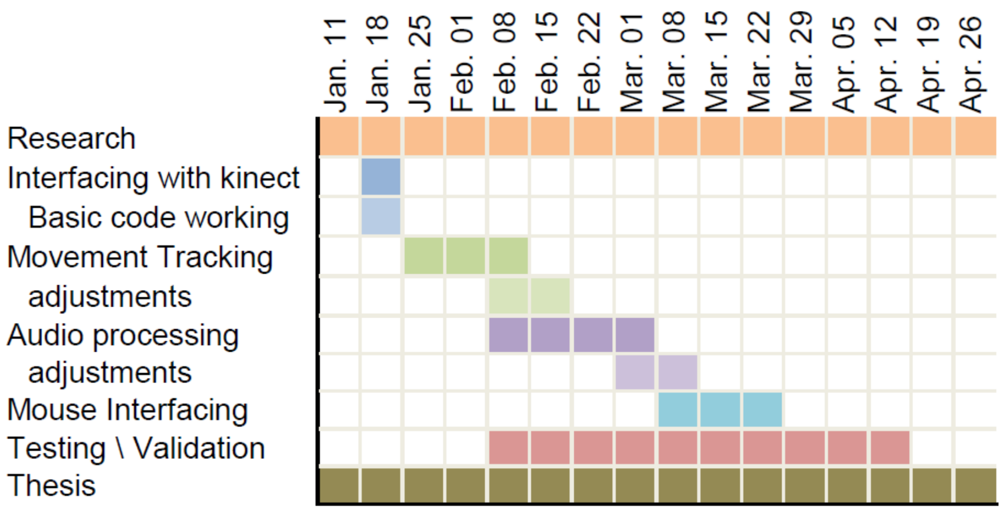

Table 2-1: Schedule

2.2 Adherence

As with all long term projects, there is naturally some deviance from the original perceived schedule. This project was susceptible to schedule slips as well. The main area where a schedule change was most predominate was in the area of 'Mouse Interfacing'. Although this was originally listed as being only a three-week long task, this was stretched to being about five weeks with small amounts of incremental fixes. The reason for this increase was due to technical problems which, cause a change of approach. The original idea was to use a Bluetooth module which could emulate a mouse HID. Unfortunately, the module's lack sufficient documentation which made debugging challenging so a fully wired connection was chosen in its place.

2.3 Milestones

Major milestones that I have set are the completion of major tasks, as one would normally expect.

- Interfacing with the Kinect and getting things up and running. Although this is a short task in terms of duration, it marks the start of the rest of the software portion of the project.

- Completion of movement tracking is half of the data processing task to create a pick device.
- Completion of audio processing is the other half of the processing task and these two parts complete all the necessary processing that is needed to emulate a pick device.
- Pushing the data to a HID mouse interface will be one of the final goal posts of the software side of this project.
- Completion of research and the thesis are both huge mile stones as they ultimately signify the end of the project.

3 Historical Overview

There are several aspects which are incorporated into this project which have progressed significantly and have been well review over the years. The project is focused around two parts, the first being the over idea of creating an alternative pick device. Seeking alternative computer input methods for the disabled is a topic with many iterations. The second part of this project is the use of the Microsoft Kinect which has been the subject of numerous scholarly and non-scholarly works.

3.1 Taxonomy

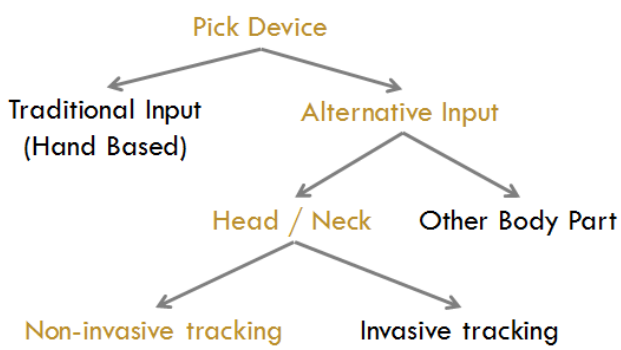

Figure 3-1: Taxonomy

The figure above ('Taxonomy 1') is the hierarchy of how, in regards to this project, the genre of pick devices was decomposed. This idea is also known as 'taxonomy', and is a key way to survey a genre and to see what other contributions exist in that area. Located at the top of this taxonomy is 'pick device' which is at its roots the device which is being created in this project. The children of 'pick device' have been split into two groups the tradition style of hand based input, and alternative input which consist of all non-hand based input methods. Moving down the input method is broken down by the body part which is controlling the pick device. Here there are two categories of "head and Neck" and "Other Body Part", for simplicity sake the other group is used but in reality there would be numerous groups (one for each body part which could be used to control a pick device). Final level groups the tracking method used. An invasive tracking method is determined by, if the device requires the user to place, adhere, or wear any special device on their body to assist with the tracking. For example, a device that requires the user to wear an accelerometer on their head is considered invasive. The categories pertaining to this project are highlighted in orange.

3.2 Alternative Methods

Finding ways to help to help people who suffer from quadriplegia interact with computers and other device is not a new pursuit. There are is a variety of devices currently on the market which encompass an abundance of different approaches to their user interface. Since most quadriplegics still retain a full or near-full range of head and neck motion in comparison to that of someone fully abled [13, 14], it is logical to use head motion as a source of input. Tracking head a user's head movements is also not a new idea, however past iterations of this have had a few shortcomings. A common setup of a head tracking device requires that the user adhere some form of a tracking dot on their head. A camera attached to the target computer monitors this dot's movement [2, 3]. Other implementations require the user to wear a measurement apparatus on their head which is tethered to a computer [15]. One less invasive technique is to monitor the user's eye motion or eye focus, which is used for mouse movement. Eye tracking seems to like the most natural way to interact when using a pointing or pick device as a user naturally looks at the area of interest [16]. However, eye tracking is interesting plagued with problems such as pupil dilation, blinking, or the user changing his or hers field of view [17]. A test was conducted between head tracking and eye tracking based input methods to monitor their performance. In this setup performance was being measured as number of correct mouse actions divided by the total time to complete those actions. The results showed that "head mouse outperforming the eye mouse by 1.28 times" [18] . Initiating a pick or a 'click' event has been dealt with in numerous ways some of which are non-solutions for a quadriplegic such as the use of a foot pedal [2, 3]. A common solution is dwell-time, which is allowing the user to hover over the point they which to initialize a 'click' event [2, 3, 19]. Methods such as monitoring mouth cheek movement [15], teeth presses [20], or inhale and exhale breaths [5], have also been utilized. Methods such as these however, require sensors to be placed in or around the user's mouth or head. Additionally some of these methods are prevalent to trigger false click events such as the puff [20]. Taking cues visual cues to trigger pick events such as eye blinks [21], is a good alternative as it does not require devices placed on the user.

3.3 Microsoft Kinect

In 2010 Microsoft released their first version of the Kinect, which is a motion, depth, and audio sensor. Later that year open source cross-platform drivers were created [22, 23] followed by Microsoft releasing official drivers and a SDK for Windows [24]. The Kinect hardware as well as its concept itself has been the subject of multiple papers and project. The precision of the depth sensors has been test in multiple ways, such as comparing it against known lab grade sensors which showed an measurement difference primarily of ± 2cm [8]. The depth sensors have also been tested for object (person) recognition [25] as well for motion tracking and gesture recognition [26, 27]. Several gestures tested were focus around basic head motions, such as nodding ‘yes’ (vertically) and ‘no’ (horizontally). The Results shown were that the Kinect using depth sensing alone works very at detecting and differentiating head movements [26]. For greater accuracy, noise reduction, or reduction of depth blind spots it has been suggested and show that multiple Kinects could be used in unison [28].

3.4 Future State

The goal of this project relies and improves upon the work that has been done in the before mentioned topics. Current pick device input solutions require the user to wear a device or to place something on themselves which can be tracked more easily (such as a reflective dot). This method of motion sensing is intrusive and uncomfortable for the user. Furthermore, the devices being used are limited to one target (i.e. the users home PC). The approach of this projects uses the Microsoft Kinect as it provides depth accurate depth information as opposed to current solutions which use visible light. Using depth information has the key advantage of not requiring a user to wear a sensor to track motion nor wear something which will be track by a webcam. The project here also looks at extending the usability by providing a standard HID interface allowing the user to interface with various types of computer devices.

4 Execution

The systems execution can best be described by following the path the data takes as it enters and exits the system. Since the data flows in a single linear direction within the system it makes everything easy to describe and follow. Beyond this the system, as shown in figure 4-1, is only composed of a few key elements which pass data along.

4.1 Setup

There are two elements which are needed for setup: there is the software, and there is the hardware. This section illustrates the physical placement and connections of all parts in the system as well as how to run the software.

4.1.1 Hardware

Ideally for setup, the user is place approximately 1.5 meters linearly away from the Kinect. This distance was chosen as the sensor depth value does not follow a linear correlation with the actual distance [7]. This distance is also the middle of the sensor’s “sweet Spot” which ranges from 0.8m to 2.5m as dictated by Microsoft [9]. The graph following graph, Kinect Depth, shows the relationship between the sensors returned value and the actual measure distance to the sensor (in feet). The area where the graph’s slope has the highest rate of change is the region were which will provide the best differentiations in distances. The user should be positioned such that the Kinect is in level with, and in front of the center of the users face.

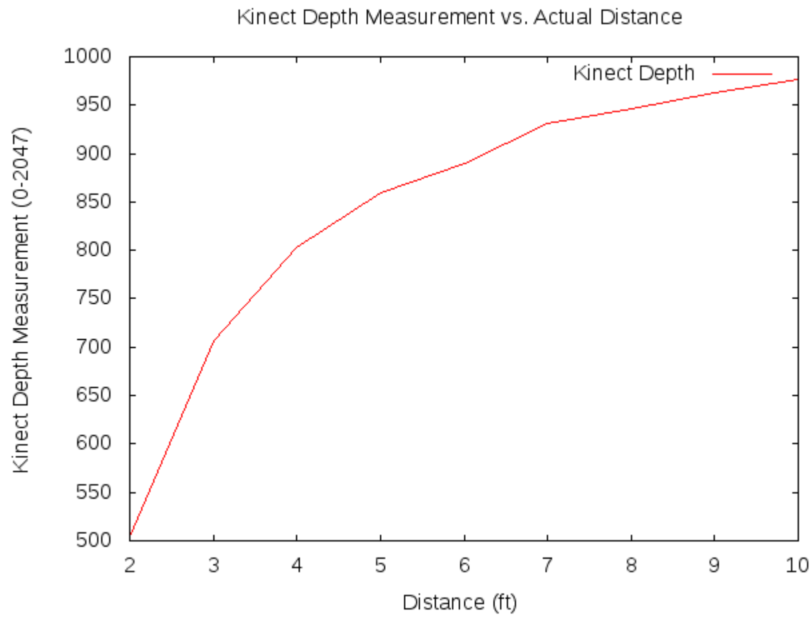

Figure 4-1:Kinect Depth Sensor

There are various connections which must be setup to ensure data flows to all the correct places. Figure 4-2 shows the connectivity between all the devices within the system. Next to each object in the figure there is a letter in a blue circle, these letter correspond to rows in table 4-1. The connections between each object are color coded and also link to rows in table 4-1. Object 'A' is the Microsoft Kinect, the purple cone projecting out of it is intended to represent the field of view of the Kinect's sensors. Placed directly in the field of the view is the user marked as 'E'. The Kinect is shown as being connected to object 'B', the host computer by a grey arrow. This connection is a simple USB 2.0 interface which operates at high-speed to ensure sufficient bandwidth. The Host computer is where the tracking program is run and it interfaces with object 'C', the Atmel 32u4 development board. The connection between the Host and the development board is UART. It should be noted that this strictly a one-way UART connection as the 32u4 only receives data and the 'clear-to-send' and 'clear-to-receive' signals are left unconnected. The 32u4 development board used is also connected to object 'D' which is the target device. The term device is chosen as any device which features a USB port and generic HID mouse drives is suitable.

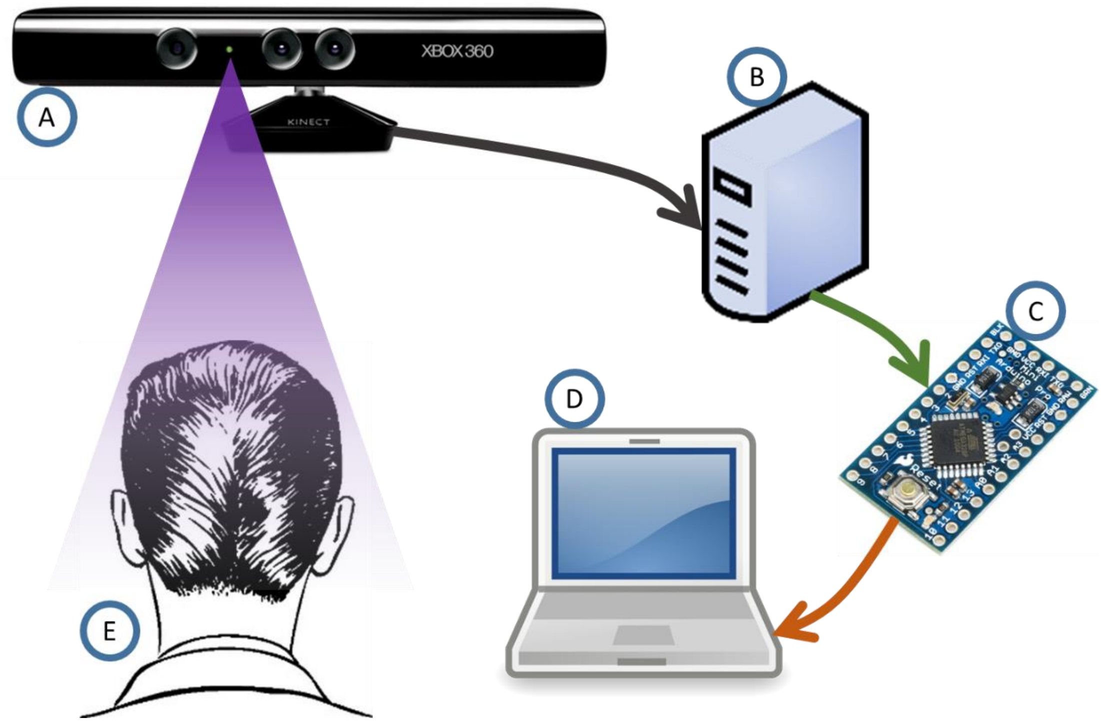

Figure 4-2:Device Layout

Letter | Connection Color	Description
A	Kinect
B	Host Computer
C	ATMEL 23u4 Development Board
D	Target Device or Target Computer
E	User
Grey	USB 2.0
Green	UART
Orange	USB (HID MOUSE)

Table 4-1:Object Info

4.1.2 Software

The Tracking software, which is discussed in greater depth in later subsections, is running on a what is known as the host computer. The program can be run on any PC with windows 7 or later. The program is started similarly to other typical applications. The Software which runs on the 32u4 is written to run on that particular processor, as it makes use of device specific functionality.

4.2 Data Flow

Figure 4-1 shows the interconnectivity of all parts within the system, and of more relevance to this section it also shows the flow of data between the parts by use of arrows. The Kinect collects visible light (in the form of RGB and grey scale image streams at 640x480 pixels), IR depth data (as a 640x480 array data stream), and audio data. These three streams are sent via USB 2.0 from the Kinect to the host computer. A program on the host computer, uses both the visible light and depth data to track the users head. The audio input is monitored for certain verbal commands, these verbal commands can be found in verbal commands section. The tracking information as well as any audio cues are processed and condensed into a single stream of commands. The stream of commands is sent to the Atmel 32u4 microprocessor through UART and an FTDI chip. The microprocessor reads the command stream and outputs HID mouse actions corresponding to the actions it read. The 32u4 has the ability to provide HID mouse actions over a native USB interface. On one side the 32u4 it takes in the command stream from the Host PC and on outside the 32u4 send out mouse actions. Lastly we have the final destination which is the target PC. The target PC is connected to the 32u4 but, only sees it as a simple mouse. This means any mouse movement or clicks act exactly the same on the target PC as if it was any other mouse.

4.3 Application

The tracking application is a windows application and is started just like any other application. There are three key parts which make up the tracking program: Head tracking, verbal cue recognition, and creating a command stream. To accomplish these tasks, the program makes use of Windows and Kinect drivers and APIs. These APIs and drivers provide both basic and high level support for the tracking program to function.

4.3.1 Face Tracking & Mouse Movement

By default, once the program has started, it will begin listening for verbal commands and trying to find a face to track. Tracking and mouse movement is done using the both the visual data and depth data streams from the Kinect. The program will look at both data streams and try to find some area (using the depth data) and visual data blob which line up and are likely to be a face. Once a suitable blob has been found it is bound in a rectangle and the center point of the bounding box is calculated. This first tracked center point is saved and marked as the origin point, as it's the point where the head was first discover. As the head moves the bounding box around is updated, as well as the center point for the new bounding box. The distance between the origin point and the current box's center point is used to calculate the mouse movement. Along with this the certain filters are applied to better tune the mouse movement. For example, a user might have limited lateral movement to the right side so the filter can be adjusted to compensate for this. Figure 4 shows the bounding box in pink, the origin point in blue and the current center point in mint green.

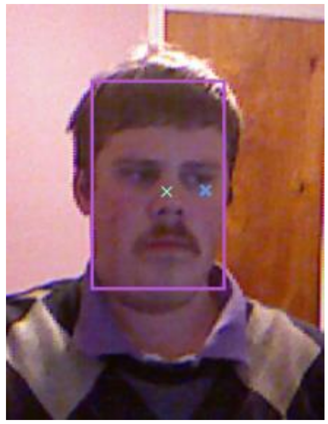

Figure 4-3: Bounding Box

The filtered difference movement information is then parsed down and sent over serial in what is called the command stream.

4.3.2 Verbal Commands

A predefined list of individual key words is loaded into the tracking application which is run on the host computer. The tracking application monitors the audio stream provided by the Kinect, if the audio stream contains sounds which matches a word from the verbal command list an event is triggered. In this program words are only matched, meaning any words not on the verbal command list are disregarded as noise. Within the event, a switched based on the matched word is used to set values and trigger certain actions. For example, the word ‘right’ or ‘left’ will not tiger an action but change the context of which mouse button an action will be carried out on. A word such as ‘click’ will invoke a function transmit a click command for the current button. Table 2 shows the commands and their resulting outcome.

Verbal Command	Resulting Action
Right	Changes the mouse context to correspond with the ‘right’ mouse button
Left	Changes the mouse context to correspond with the ‘left’ mouse button
Click	Invokes a click for the current mouse button context.
Down	Invokes a button down (but NOT up) button action for the current button context.
Release	Invokes a button up button action for the current button context
Halt	Stops the program from issuing any mouse actions (movement included)

Begin	Resume the programs functionality. On program start the active, until told to halt.

Table 4-2: Verbal Commands

5 Testing

5.1 Preface

This section further describes the steps and develops the ideas from the *Introduction – Verification* section. Section assumes that the setup used is in line with setup described in *Execution – setup.* Additional information about the testing environment used is detailed in the environment subsection.

5.2 Method

Each test will be done twice by the same user. The first iteration will be a control test, which will use a standard hand based 'mouse'. The second iteration the user will use the Kinect mouse. The test procedure will be a set of actions the user must perform on the computer which involve heavy usage of a pick device. The actions outline common 'pick' device oriented actions, that involve a mix of click, drag, and drop operations. The test performed in this section will measure the time difference between the two pick interface methods. By looking at the difference in completion time between the methods, we can get a better understand of how well suited the Face Mouse is for everyday computer usage. If common actions are found to take a significant amount of completion time compared to a regular mouse a new approach might need to be considered. conversely, if this system can perform the same actions in an equal or lesser time than the tradition mouse, this means the system is a good alternative pick method.

The user must perform the same set of actions twice once as a control and then again the Kinect based input. The time it takes to complete each set of actions will be measure. Each set of actions follows the same pattern: it starts by the test user starting a timer program on the computer, followed by some other graphical task, and is concluded by the used stopping the timer. The sets of actions are as follows:

5.2.1 Set 1

1) Start Timer
2) Open Opera Web browser from task tray
3) Iterate through each tab (5) by clicking on the tab
4) Minimizing the application
5) Stopping timer

5.2.2 Set 2

1) Start Timer
2) Open File Explorer from task tray
3) Change directory to home (~)
4) Copy "Copy1.txt" from home to desktop via drag-and-drop
5) Copy "Copy2.txt" from home to desktop via 'right click'
6) Stop Timer

Each set will be run twice by each for each interface type, mean the set will be run a total of four times. The sets were chosen to minimize outside factors as the all programs will already be running and opened, to reduce the difference in an application's start-up time. Along with that all actions are purely pick device driven so the total time is more reliant on variations of pick devices rather than other input methods which could have been used during the actions. The final values we are interested in are the time differences between input methods. This means, that the tabs opened in the browser or content of the Copy files is inconsequential, as long as they are keep the same across the tests.

5.3 System Configuration

There are several parts in this system which can vary, such as the Atmel 32u4 development board used and the target device. In concept any development which allows access to both of the 32u4's serial ports will be acceptable for use in this system. For this present system, an Arduino Pro Micro clone was used as it came with a presoldered USB header attached as well as all pins being exposed. The target device, is meant to vary as to allow the user to use which ever device they need to interface with. For testing, a laptop running Linux kernel 3.18 and the KDE windowing system was chosen. However, the test sets outlined in the earlier subsections can easily be run on a wide range of hardware and operating system combinations.

6 Closing

6.1 Conclusion

Overall the system was able to successfully complete all the required goals set out for it. It is able to track in a non-invasive manner a user's head movements and relay them to a target computing device as mouse movements. Beyond this the user can initiate a variety of 'click' actions, in this case it is done with verbal commands. Lastly the whole system uses standard off-the-shelf parts and is almost completely universal, as it can be connected to nearly any computer and run without any drivers being installed.

6.2 Positive Results

The system detailed above not only shows that it is possible for users who are afflicted with tetraplegia to interact with computers but, that is also possible to do so without the use of any invasive setup. The system's implementation is unique in that it works simply and easily for the user. Setup works with nearly any target device that features a USB port. That is due to the system's use of the standard HID USB interface no special device driver need to be installed. Every conceivable action which can be performed by a conventional arm and hand controlled mouse can be completed with this system. These actions extend beyond basic right and left clicks but include complex action sets as well. Actions such as drag-and-drop operations or the ability to have either the left or right button down while the other mouse button is toggled is also possible. This allows the system to be fully adaptable to nearly any operating system, as it can properly perform any needed button and cursor operation. Beyond this the system's verbal commands are a simple and intuitive way to control a pick device clicks, while still be advanced enough to perform complex tasks.

6.3 Negative Results

Albeit the system working as it should, there are still a few areas which are lacking and did not perform as well as I would have wished. The first and foremost problem that I see, would be the reliance on platform and operating system dependent software. Despite initial development being done with the open source Linux drivers and API, the final iteration was completed with C# and the Microsoft Kinect SDK. While the technical reasoning behind the switch made sense at the time, this is a draw back for long term development. The current code base limits the possible platforms to primarily x86 based processors, which can run full versions of the Windows operating system and C# runtime environment. Additionally, when using the device, the cursor movement is not a smooth and fluid as one would want and expect. Although through iterations it has gotten significantly better, it still leaves more to be desired. The cursor jitter comes from the method of tracking, which bound the tracked area in a box and finds it's centroid. Since the user's head is capable of pitch, roll, and yaw movements, this can effect what the tracking program 'sees' as being the center of the user's head.

6.4 Further Work and Improvements

For future refinements and iterations upon this system, there are several short term and longer term items which should be addressed. Calibration based on the dexterity of the user head motion is something that can and should be addressed in the short term. The functionality to account for head movement differences is already possible. However, these values are hard set in the tracking program and cannot be easily changed by the user. The cursor jitter mentioned earlier, has been mitigated by use of a low pass filter. Although mitigated, it is still persistent and further refinement and development of filtering a user's head movements is needed. Although the ability to connect the system to the target device via USB seems advantageous, there exists one large draw back; the target must feature a USB port. If the system were to be able to use multiple interfaces, either Bluetooth or USB, to connect to the target device a significantly greater range of devices would be accessible. With the increase in popularity of tables and smartphones, the usefulness of being able to interact with such devices seems more and more apparent. Having a Bluetooth interface would allow this system to fulfill that need. Additionally, I would want to see the system be implemented such that it is able to run using 100% open source software packages. This would cause a lot of rework to be done for the audio portion of the code however, the added flexibility of having a platform independent system are well worth it. There would be certain marketability constraints which arise from open licensing certain software libraries contain, however an altruistic approach of having this device also be open would solve that. The main benefit of moving away from the office Windows Kinect SDK, would be the ability to explore approach 1 in more depth. Running on an embed system with is possible with the current setup and approach, however the options are extremely limited. A more open solution UNIX based solution would greatly increase the number of hardware options, and help keep the decrease the cost of the whole system.

References

[1] Anonymous (October 24). *Spinal Cord Injury: Hope Through Research*.

[2] SmartNav. (). *SmartNav 4:AT Overview*. Available: http://www.naturalpoint.com/smartnav/products/4-at/.

[3] Orin Instruments. (November 12). *Origin Instruments Corporation: Price List*. Available: http://shop.orin.com/PriceList_111214.pdf.

[4] M. Graham. (November 28). *10 trends to Define Marketing in 2013- #3 Smartphones and Tablets Pass PCs*. Available: http://www.sourcelink.com/blog/matt-graham/2012/11/28/10-trends-to-define-marketing-in-2013--3-smartphones-and-tablets-pass-pcs.

[5] Origin Instruments Corporation. (). *Sip and Puff Switch Solutions*. Available: http://www.orin.com/access/sip_puff/index.html.

[6] FDA. (). *Is The Product A Medical Device?*. Available: fda.gov.

[7] Crock Nathan. (). *Kinect Depth vs. Actual Distance*. Available: http://mathnathan.com/2011/02/depthvsdistance/.

[8] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of kinect depth data for indoor mapping applications," *Sensors,* vol. 12, pp. 1437-1454, 2012.

[9] Anonymous (). *Human Interface Guidelines (HIG)*. Available: http://go.microsoft.com/fwlink/?LinkID=403900&clcid=0x409.

[10] M. A. Livingston, J. Sebastian, Z. Ai and J. W. Decker, "Performance measurements for the microsoft kinect skeleton," in *Virtual Reality Short Papers and Posters (VRW), 2012 IEEE,* 2012, pp. 119-120.

[11] E. Chai, B. Deardorff and C. Wu, "Hacking Bluetooth," 2012.

[12] G. Harris. (October 27). *Bluetooth capture setup*. Available: http://wiki.wireshark.org/CaptureSetup/Bluetooth.

[13] M. Ouerfelli, V. Kumar and W. S. Harwin, "Kinematic modeling of head-neck movements," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions On,* vol. 29, pp. 604-615, 1999.

[14] S. K. Salisbury, N. L. Choy and J. Nitz, "Shoulder pain, range of motion, and functional motor skills after acute tetraplegia," *Arch. Phys. Med. Rehabil.,* vol. 84, pp. 1480-1485, 2003.

[15] Y. Chen, W. Chen, T. Kuo and J. Lai, "A head movement image (HMI)-controlled computer mouse for people with disabilitiesAnalysis of a time-out protocol and its applications in a single server environment," *Disability & Rehabilitation,* vol. 25, pp. 163-167, 2003.

[16] R. J. Jacob, "The use of eye movements in human-computer interaction techniques: what you look at is what you get," *ACM Transactions on Information Systems (TOIS),* vol. 9, pp. 152-169, 1991.

[17] R. J. Jacob and K. S. Karn, "Eye tracking in human-computer interaction and usability research: Ready to deliver the promises," *Mind,* vol. 2, pp. 4, 2003.

[18] R. Bates and H. O. Istance, "Why are eye mice unpopular? A detailed comparison of head and eye controlled assistive technology pointing devices," *Universal Access in the Information Society,* vol. 2, pp. 280-290, 2003.

[19] J. Gips, M. Betke and P. Fleming, "The camera mouse: Preliminary investigation of automated visual tracking for computer access," in *In Proc. Conf. on Rehabilitation Engineering and Assistive Technology Society of North America,* 2000, pp. 98-100.

[20] T. Simpson, C. Broughton, M. J. Gauthier and A. Prochazka, "Tooth-click control of a hands-free computer interface," *Biomedical Engineering, IEEE Transactions On,* vol. 55, pp. 2050-2056, 2008.

[21] M. Chau and M. Betke, "Real time eye tracking and blink detection with usb cameras," *Real Time Eye Tracking and Blink Detection with Usb Cameras,* 2005.

[22] JAVIER MARTÍN. (November 11). *Me gusta trastear*. Available: http://tecnologia.elpais.com/tecnologia/2010/11/11/actualidad/1289469664_850215.html.

[23] Anonymous (November 10). *WE HAVE A WINNER*. Available: http://blog.adafruit.com/2010/11/10/we-have-a-winner-open-kinect-drivers-released-winner-will-use-3k-for-more-hacking-plus-an-additional-2k-goes-to-the-eff/.

[24] R. Knies. (February 21). *Academics, Enthusiasts to Get Kinect SDK*. Available: http://research.microsoft.com/en-us/news/features/kinectforwindowssdk-022111.aspx.

[25] L. Xia, C. Chen and J. K. Aggarwal, "Human detection using depth information by kinect," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference On,* 2011, pp. 15-22.

[26] K. K. Biswas and S. K. Basu, "Gesture recognition using microsoft kinect®," in *Automation, Robotics and Applications (ICARA), 2011 5th International Conference On,* 2011, pp. 100-103.

[27] Z. Ren, J. Meng, J. Yuan and Z. Zhang, "Robust hand gesture recognition with kinect sensor," in *Proceedings of the 19th ACM International Conference on Multimedia,* 2011, pp. 759-760.

[28] Y. Tai and I. K. Park, "Accurate and real-time depth video acquisition using Kinect–stereo camera fusion," *Optical Engineering,* vol. 53, pp. 043110-043110, 2014.