Assisted Touchscreen Interaction for Users with Motor Impairments

Maggie M. Casale
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA 56267
casal033@morris.umn.edu

ABSTRACT

With the high popularity of touchscreen devices such as smartphones and tablets, it is important to find ways to limit the digital divide for people with motor impairments. Users with motor impairments interact with touchscreen devices differently than an average user, and "users with motor impairments" is a broad category. These users can have varied disabilities ranging from minor hand tremors to missing limbs. This means solving common problems among these users can be difficult since solutions may need to be specific to the user's abilities. This paper explores how motor-impaired users are interacting with touchscreens and how this interaction research can improve current assistive technology adoption rates. Finally, we will give an overview of three new assistive programs: Touch Guard, Octopus Launcher, and DOWELL.

Keywords

interaction design, motor impairments, assistive technology, accessibility, mobile device, touchscreen, smartphone

1. INTRODUCTION

Research has shown that people with motor impairments interact with touchscreen devices differently than a typical user. These users often desire to use the same tools as everyone else, and their inability to do so is contributing to the digital divide. The digital divide is the gap in economic or social status between those who can and cannot readily access information with computers. To aid this touchscreen interaction motor-impaired users rely on *assistive technology* (AT), defined as:

> Any item, piece of equipment, or product system that is used to increase, maintain, or improve functional capabilities of individuals with disabilities [13].

There are several commercial AT products, but there are also many challenges when matching a product to a user. With the numerous options on the market it is difficult to choose the "best" product for a motor-impaired user. As an AT developer, there are also difficulties designing a general product for a motor-impaired user. With the combination of these difficulties and more, AT adoption rates are poor and only about 35% of purchased AT are used [8]. By looking at research in motor-impaired touchscreen interaction, we can limit the digital divide and improve AT adoption rates.

This paper will explore research in touchscreen interaction for motor-impaired users. In Section 2, we will give a brief description of motor impairments and explain current AT being used to help aid motor-impaired users in computer and touchscreen interaction. In Section 3, we will discuss research in motor-impaired touchscreen interaction and how this research can help increase AT adoption rates. In Section 4, we will talk about new AT programs being developed.

2. BACKGROUND

2.1 Motor Impairments

To discuss more detail about research in the field of AT for those with motor impairments, we first need to understand what motor impairments are. Motor impairments are:

> The partial or total loss of function of a body part, usually a limb or limbs. This may result in muscle weakness, poor stamina, lack of muscle control, or total paralysis [11].

There are several impairments that can affect a user's abilities, and these impairments can affect a user's abilities in several different ways. We will discuss three examples of these: tremors, cerebral palsy, and paralysis.

Tremors are an involuntary quivering movement, often in the user's hands. This is the most minimal example of a motor impairment affecting a user's abilities, but is still included in the category. Research done by Zhong et al's [14] has shown users with tremors struggle selecting small targets, such as a item or button on a touchscreen that a user is attempting to select. Users with tremors can have issues with other gestures, such as sliding or multi-finger gestures.

Another motor impairment is cerebral palsy. This is a condition that refers to any one of a number of neurological disorders that appear in infancy or early childhood and permanently affect body movement and muscle coordination but do not worsen over time. Common characteristics of cerebral palsy include muscle tightness or spasm, involuntary movement, and impaired speech. Severe cases can lead to paralysis. Users with cerebral palsy are able to use a mouse, but may have difficulty controlling movements. Similarly, in touchscreen interaction a user may struggle with continuous gestures such as dragging or scrolling accurately. [12]

The final motor impairment to discuss is the loss or damage of limbs. This is better known as *quadriplegia*. Quadriplegia, is paralysis caused by illness or injury that results in the partial or total loss of use of all limbs and the torso [12]. There are more precise terms for quadriplegia depending on the amount of absent limb abilities, but we will be using quadriplegia as a

UMM CSci Senior Seminar Conference, December 2015 Morris, MN.

general term. Users with quadriplegia often need assistance in basic interaction with a touchscreen device since one or more limbs are unable to interact.

2.2 Assistive Technologies

To enable a motor-impaired user to interact with touchscreens we use assistive input devices. This layout is similar to an impaired user interacting with a computer, where the user needs to be able to click and move the cursor on the screen without the typical mouse or touch interaction.

Options for the click interaction include: button switches, the Sip & Puff, using dwell time, and more. The button switch is the simplest, using a button to click. The Sip & Puff is a puff-entry device. To click, a user "puffs" into the straw-like device; a user can also right-click by "sipping" from the Sip & Puff [13]. Dwell time is typically built into assistive devices that act as cursors. To click, the user simply dwells on a target for a specified amount of time.

For the cursor interaction, some devices use a switch interface such as button switches and the Sip & Puff. The switch interface was initially produced as hardware, where a user would connect their switch to the hardware, and connect the hardware to their computing device. Currently, this interface is often a built-in or downloaded program that connects a switch to a computing device with a single cord or Bluetooth connection. Once the switch is connected with the switch interface, interaction can begin. The switch interface works by slowly scanning across a user's screen, highlighting each clickable target as it scans. To stop the highlight movement, a user clicks their switch once. The user clicks again to select the target. If no selection is made, the highlighting will continue to move. With two or more switches, the user can have more control of this highlighting movement.

An alternative to switch interfaces are *human interface devices* (HIDs), these are assistive input devices that give the user a similar cursor interaction as a mouse would. There are a variety of HIDs on the market including wheelchair, mouth or thumb joysticks; head arrays, which detect head position typically for wheelchair movement; eye recognition; voice recognition; and gesture recognition. We will not focus on recognition devices in this paper in favor of more cost-effective designs, but it is worth noting that they do exist. While some HID have built-in dwell time click, others can be combined with other inputs used for the click interaction such as the Sip & Puff.

With the growing need for assisted interaction with touchscreens, popular systems like iOS and Android have developed features which aid in some accessibility aspects. Both include features for connecting a user's switch to the device with built-in switch interfaces, an assistive menu (called *AssistiveTouch* in iOS and *Assistant Menu* in Android), and allows their users to customize touch settings. An assistive menu is an icon that always sits on the user's screen, this pops open to a mini window when selected. This menu helps a user to control their device by aiding in performing gestures like a pinch or multi-finger swipe, and also gives quick access to applications, hardware buttons, or other menus. The customizable touch settings allow users to adjust aspects like dwell times, where the system should only initiate a tap if the user is touching the screen longer than x seconds. This helps to avoid accidental selection. [5, 1]

3. TOUCHSCREEN INTERACTION

In this section, we will discuss two studies that take a closer look at touchscreen interaction for users with motor impairment disabilities. The first is Anthony et al.'s [4] observational study of YouTube videos. In this study, the researchers collected and analyzed 187 noncommercial videos uploaded to YouTube that depicted a person with a physical disability interacting with a mainstream mobile touchscreen device. They then coded the videos along a range of dimensions to characterize the interaction, the challenges encountered, and the adaptations being adopted in daily use. This gives a better understanding of how users interact with assistive technologies at home or in other personal spaces.

The other study is Carrington et al.'s [6] multi-case study. The goal of this study was to explore participants' preferences for wearable and *chairable* input and output devices for interacting with mobile devices. In the study, chairable is explained as:

> Technology that takes advantage of the (currently underutilized) space on and around a wheelchair, much as wearable computing leverages on-body and worn mobile technology.

They also wanted to identify possible design configurations for wearable and chairable devices. From studies like these, we can help researchers better understand AT adoption rates for motor-impaired users, which will be the last aspect we will discuss in this section. More specifically, we will look at factors that affect adoption rates and possible solutions to increase these adoption rates.

3.1 YouTube Study

In Anthony et al.'s [4] study, observations were categorized by the body part interacting with the device. We will focus on the finger interaction since this is the typical user interaction method. The observations from Anthony et al.'s study led to three design implications or findings.

3.1.1 Accidental Selections & Range of Motion

When using a touchscreen device it was common for a user to hold or touch the touchscreen too long, causing unintentional selecting on the device. Touching too long occurred when a user was moving too slow in the gesture, like a tap, and it would be recognized as a holding down gesture, such as a long tap. Some users were unaware they were even touching the device screen. Other gestures, like dragging or sliding, seemed to have similar issues as well. For example, the user was slow in the dragging or sliding gesture, it could be recognized as a long tap. To limit this, the researchers suggest supporting "constant touch habituation". This means that if a user is holding down on the device screen accidentally (longer than x seconds), the system should not change, move, or select anything.

An issue with dragging or sliding was the user's range of motion. If a user had more extensive motor-impairments, it was more difficult to complete the full motion on a larger display, like an iPad, than on a smaller display, such as an iPhone. Users adapted to this by changing the device's orientation (horizontal or vertical) depending on the motion of the drag or slide gesture needed. Some users were not able to complete this gesture of dragging or sliding at all, so a third party was needed to complete it.

3.1.2 Multi-finger Gestures

There were difficulties with multi-finger gestures such as pinching or 3-finger gestures, especially for users who used a mouth stick to interact with the touchscreen. Though iPads support single finger interaction with AssistiveTouch (discussed in Section 2.2), no videos were found of users using this feature. In response to a survey sent out to makers of these videos by the researchers, many participants were not aware of AssistiveTouch. Those who were aware, did not find the need for the feature constantly and preferred to struggle with a gesture than go though the process to turn AssistiveTouch on. The researchers suggest that additional interaction research be done with AssistiveTouch.

3.1.3 Edges or Barriers

The researchers observed users whose motor-skills were extensively impaired, so these users were focused on selecting targets. There was often a physical template placed over their device to help guide their finger(s) to specific areas for target selection. The researchers explain that most touchscreen interfaces are being created flat with no borders or edges along the touchscreen. This is a comparison of a current touchscreen device such as an iPad, with older touchscreen devices such as a PDA or cell phone that had a physically raised edge around the touchscreen. This edge acts as a barrier for users to navigate selections easier. By following this edge, the user moves their finger(s) to the appropriate area with fewer accidental selections. Unfortunately, the majority of these barriers were flimsy since they were made from materials found at home (paper or cardboard) and would not last long-term. The researchers concluded that the idea of a commercial template or barrier was interesting, but would need to be standardized for popular applications and touchscreen devices.

3.2 Wearable and Chairable Study

3.2.1 Chairable Procedure and Observations

In Carrington et al.'s [6] multi-case study, the researchers explored preferences of users in a wheelchair for chairable input and output devices for interacting with mobile devices. Input devices were HIDs used for interacting with a mobile device, while output devices were ways of displaying this interaction other than the mobile device.

They did this in multiple steps, starting by having motor-impaired participants use DIY (Do-It-Yourself) techniques to create their own input devices. The researchers then worked with 30 physical therapists, occupational therapists, and rehabilitation technicians in focus groups to configure designs for a phone, a game controller, modular input modes, and modular output modes. These designs were discussed, and the options were narrowed. The last step involved motor-impaired participants, who were asked preferences of each design based on various aspects.

In the the survey of participants, one question asked is: "What would the overall ideal design be of your wheelchair?". User's preferences varied but had some common aspects. All users, regardless of their abilities, placed inputs and outputs adjacent to each other in locations around their armrests and joystick area. All users chose to add multiple output options, often including a head mounted display (Google Glass) in combination with a projector which would project onto local surfaces (lap, table, etc.). Additionally, all users added integrated controls such as buttons, switches, or touch pads into at least one area of their wheelchair.

3.2.2 Chairable Design Implications

From Carrington et al.'s study three design implications were developed: maintaining the wheelchair's form, different controls for different regions, and familiarity. Starting with maintaining the wheelchair's form, it was found important to not obstruct the wheelchair's normal navigation. Additionally, it was found that some participants were uncomfortable altering their chair's silhouette, thinking it would become bulky and create more of a social barrier. The researchers explain that this is easily preventable by not expanding upon the wheelchair just anywhere, but by using discrete areas for storing hardware like underneath or behind the chair's seat.

The design implication of different controls for different regions was a priority to support a user's range of abilities. With this, researchers noted comments from participants saying "controls should match the area within the user's range of motion and the body part that will actuate it. Controls near the user's fingertips can be small, while controls near the user's shoulders must be larger."

The last design implication was familiarity. If a participant was familiar with a technology, they would be willing to interact with it. However, if a participant was not familiar with a technology (e.g, Google Glass), they may be skeptical or hesitant to use it. Some participants were excited about new input and output devices, while most favored simpler HID, they were already familiar with. This is an interesting design implication, considering the variety of new assistive technologies being developed. When designing for an older generation of motor-impaired users, this technology may not be familiar to the user, so the interaction with the technology should be familiar. From this, it is also important to note that examining the "walk up and use"-ability of a device may aid a user's perception of it as discussed in a similar study [9]. Essentially the "walk up and use"-ability, is the idea of how easy is it for a user to walk up to a device and be able to use it effectively or efficiently.

3.3 Assistive Technology Adoption

3.3.1 Heuristics

Only about 35% of all purchased AT is used after being bought. This means 65% is being thrown away or left on a shelf [8]. Naftali and Findlater [9] touch upon this explaining that the AT is not being adopted because it is not portable, it is costly, it is limited, or there is not a need for it. Other factors of AT adoption include: the amount of physical effort, motivation, time, cognitive effort, and social weight it takes to complete a task with the device [7].

To combat the poor adoption rates, we take these factors and design implications to formulate heuristics about the AT. Deibel [7] has created such a heuristic by expanding on a similar heuristic for an augmentative and alternative communication device (a type of AT). The heuristic determines the probability of an AT being adopted given a set of factors, this can be seen in Figure 1. These factors include: *Device Necessity* - How much a user needs the device to complete a task, *Task Motivation* - The amount of motivation it takes to complete a task with the device, *Time* - The amount of time it takes to complete a task with the device, *Physical Effort* - The amount of physical effort it takes to complete a task with the device, *Cognitive Effort* - The amount of cognitive effort it takes to complete a task, and *Social Weight* - The amount of social weight of using the device to complete a task. The social weight factor we've seen before in Section 3.2.2, where users may already feel social weight from their own wheelchair. Adding to this social weight would not make preferable AT. This heuristic also includes a context aspect,this is for factoring in the environment where the AT will be used. A person may be more likely to use an AT at home rather than on the go, or in public spaces. Essentially the idea is: the more time, physical effort, cognitive effort, and social weight a device takes to use, the higher the user's device necessity and task motivation must be to use it.

3.3.2 Lowering Costs

One aspect the heuristic does not include is the cost of an AT. AT is often expensive when initially developed since the technology being integrated is new or complicated. However some AT have been on the market for a long period, and costs haven't lowered on the user's end. Depending on the AT,

Figure 1: Heuristic Model of AT Adoption [7]

$$\text{P(usage|context)} = \frac{\text{DeviceNecessity(context)} \cdot \text{TaskMotivation(context)}}{\text{Time(context)} + \text{PhysicalEffort(context)} + \text{CognitiveEffort(context)} + \text{SocialWeight(context)}}$$

some can range between $1000 - $4000. Schüler et al.'s [10] research has been working on this. Based in Brazil, this team from Ciência e Tecnologia of Rio Grande do Sul and Cearã, have been developing low-cost versions of usually costly AT. This research is still in early stages, but already models completed are made of used parts and are significantly lower in costs. Their most relevant AT for touchscreen interaction is the button mouse, an AT which has several buttons switches on one device. Normally costing up to $300, their model only costs $20. Schüler et al. are not alone, as there are also efforts being made for open source code for wheelchair and interface designs [6]. This can significantly help designers and developers access current designs and modify them. This can be done either by modifying the AT design features, or building the AT in a more efficient way (such as incorporating used parts).

3.3.3 Inclusive & DIY Design

Additional aspects of AT adoption being further researched are inclusive and DIY design. Inclusive design usually involves a professional working with potential users to create an AT suited for a specific range of disabilities. Carrington et al.'s [6] multi-case study discussed in Section 3.2 includes an example of this. DIY design is done by the user with or without third party assistance and is specifically created for the user who designed it.

Hurst and Tobias [8] worked to empower users to DIY by creating, modifying, or building their own AT. They also included another factor of AT adoption: Changes in user needs and priorities. This can be important when considering a user whose disability may change over time.

This study involved impaired users creating their own AT with assistance if desired. These users then took a survey about the process and their outcomes. The results from the survey show participants are interested in customizing their AT and had many achievable ideas for modifications. Some were nervous about modifications to their wheelchair impacting its operation, but were okay with modifications if they were involved in the modification. Being involved was important to participants and some found it gave them additional independence. Overall the findings are promising for future DIY development, and could improve AT adoption.

4. ASSISTIVE TECHNOLOGY RESEARCH

This section introduces three new technologies being researched. These solutions are generally focused on selecting and interacting with touchscreens on smartphones. The first is a program that enhances a user's touch selections. This is for users capable of some touch interaction with smartphones. The next solution is a program which allows the user's wheelchair joystick to control the cursor interaction with their smartphone. To click, this program uses dwell-time, referred to as dwell-click. The final solution is similar to the prior in using dwell-click, however for the cursor interaction any HID can be used.

4.1 Touch Guard: Enhanced Area Touch

Zhong et al.'s study [14] explored enhancing touch areas on smartphones with Touch Guard. This enhancement allows a user to attempt to select an area on their smartphone device, and the program will then clarify this selection for the user.

4.1.1 The System

Selection is controlled with Touch Guard, an Android accessibility service designed for use with any application or in combination with other accessibility services. To do so, Touch Guard does not reconfigure an application, but overlays a transparent full-screen to intercept touch events. Once a user attempts to select a target, the touch event is intercepted and clarification can begin. It does this in one of two ways, either with magnification or listing targets.

In Magnification mode, Touch Guard magnifies the active area of the current screen, not including the status bar or system soft buttons. This magnification can be adjusted between 1 to 3 times with a default of 1.5. It is then easier for users to select from the larger targets in the magnified view. When selecting in the magnified view, the target can be confirmed upon a single target selection. If multiple targets are selected the system will choose the closest target to the touch point.

In Target List mode, Touch Guard will take the titles or descriptive text about icons in the active area and list them in a full screen view for easier selection. This list appears with a slight transparency over the previous screen to aid the user in visual mapping of what they were initially trying to select.

4.1.2 Findings

In Zhong et al.'s research, they compare results from the two clarification techniques and the typical touch interaction. The researchers looked at acquisition times, or how long it took to complete a task such as selecting a specific application. These times were slower for both magnification (4.7 seconds) and target listing (3.7 seconds), compared to typical touch selection (1.7 seconds). The researchers believe this was because the participants are accustomed to their selection not being recognized, so they will continuously tap until selected. Touch Guard does not allow this repeated interaction.

The researchers also looked at error rates. These were the percentage of trials in which at least one accidental application activation occurs. Magnification had the highest error rate with 25.8%, typical touch interaction was next with 20.2% and lastly the listed targets had 7.0%. Most errors occurred when targets were small and arranged close together.

From these findings, the target list seemed to be the best option for clarification. Magnification was initially hypothesized to be the more intuitive solution, but after testing several issues arose. One was that when a user would go into magnification mode, important aspects were cut out of the screen and limited how much could be viewed in the magnified view. Also when in this view, the interaction was not as familiar. A user could not scroll around in the magnified view. When surveying participants about their preferences, several preferred the target list. Participants expressed that the interaction seemed "normal" and noticed a reduction in errors.

4.1.3 Design Implications

From this study two design implications were derived. First, it is essential for developers to follow best practices for accessibility. When developing Touch Guard, the developers discovered several applications that use custom widgets. This prevented them from accessing descriptions about the applications for the target listing. This situation is problematic for other programs needing this information like screen readers for blind users.

Additionally, it was also important to support different orientations of device. The researchers frequently observed participants rotating their devices to limit fatigue. We also see this in the observational YouTube study from Subsection 3.1.1, where users would change the device's orientation to complete a wide range gesture easier.

The other design implication is when using text instead of graphic elements, the original presentation of the text must be considered. When taking a graphic icon and translating it into text as a listed target, this text needs to be clear to the user. If ambiguous to the user, they may get frustrated with the assistance. An example of this from the study was in the Gmail application. A participant thought they were looking at a summary of a single email rather than separate grouped conversations inside of Gmail.

4.2 Octopus Launcher and DOWELL

This subsection looks at two new smartphone frameworks developed over the past two years. Since these AT are newly developed, they do not include usability studies yet. They do discuss the system, explain how a user interacts with the new frameworks, and suggest future improvements.

4.2.1 Octopus Launcher

In *Use Octopus Launcher Like Your Hands* [2], the researchers chose to develop their software, Octopus Launcher, as a interaction framework rather than improving upon existing applications. They chose to do this since application information can be private or not easily accessible through programming. To be able to interact with the smartphone without touch interaction they chose to implement a wheelchair joystick HID for the cursor interaction with an Evolution Box. The Evolution Box allows the user to toggle between the joystick's wheelchair control and Octopus Launcher control with a switch. This device contains a Bluetooth module, a joystick link, and a few other features which aid in the smartphone and joystick interaction. The researchers are working to make this joystick modular, so it can be exchanged with a joystick from a user's wheelchair or any other joystick variations. This was chosen to save costs on the user's end, and give the user a sense of familiarity with the interaction.

Initially the Octopus Launcher is displayed on the user's smartphone as a joystick with icons of eight commonly used applications surrounding it, such as menu, settings, email, etc (Seen in Figure 2). The eight options are mapped according to the eight possible directions that a joystick can move. To select an application, a user simply moves the joystick in the direction of the application to highlight it. Once highlighted, the user dwells for two seconds to initiate a selection of a target. This highlighting aspect was found to be important to help aid the user in mapping their joystick movement to what they were selecting on the screen. Inside of the application, the user can navigate using the joystick as the cursor interaction (Seen in Figure 2). If movement stops, after a slight delay, a sub menu appears with five interaction options; these can include going back, click, menu, scroll, or home. To exit the submenu, the user moves backwards with the joystick (this is what the other three positions are left for in the submenu).

Figure 2: Octopus Launcher: Launch Screen (left) & Phone Dialing Screen (right)

4.2.2 DOWELL

The Octopus Launcher was successful in aiding selection control, but researchers in *DOWELL* [3] found a more interchangeable solution. DOWELL (pronounced "doo"-"wel") is another software solution which allows any HID to connect to a smartphone using an inexpensive ($3) USB-OTG cable (seen in Figure 3). This HID can be a user's joystick (as described in section 4.2.1), but can also be an adaptive mouse, keyboards, or recognition software. For users who are already familiar with these devices, this software is easier to learn and use than other systems which may incorporate new assistive devices or controls. DOWELL incorporates Guardrail UI, which is similar to Touch Guard from Subsection 4.1, in assisting selection interaction but through a HID interaction versus touch interaction. Unlike Touch Guard, Guardrail UI incorporates additional control options along the edges of a smartphone screen to aid in selection versus incorporating a translucent overlay with new controls to limit selection.

To interact with a smartphone, the user uses their HID as a cursor. To "tap" with the cursor the user simply dwells on the target for two seconds. This dwell time can be adjusted to the user's preference. Other gestures such as long tap, scroll, and swipe can be changed by accessing the top menu. The menu appears when a user dwells at the top of their screen, but is otherwise closed to allow more screen space. This menu displays the other gestures as listed above, additional gestures are also in this menu on the next page such as zoom in/out, drag & drop, and no action. Since gestures are recognized with dwell interaction, there needs to be areas for a user to dwell when resting. These are located on the left and right sides of the screen. A user can also use these areas to cancel a dwell selection by moving left or right. It was not specified if there are certain areas along the side to rest or cancel. Besides the top menu for gestures, there is also a bottom menu for hardware keys. Like the top, a user dwells on the bottom of the screen to open the menu. This menu contains hardware keys such as menu, home, and back. There are additional hardware keys on the next page of this menu such as power, volume, and brightness.

These gestures are one step and two-step gestures. The one step gestures include: tap, long tap, zoom-in, and zoom-out. These motions are simply dwell activated and do not need a

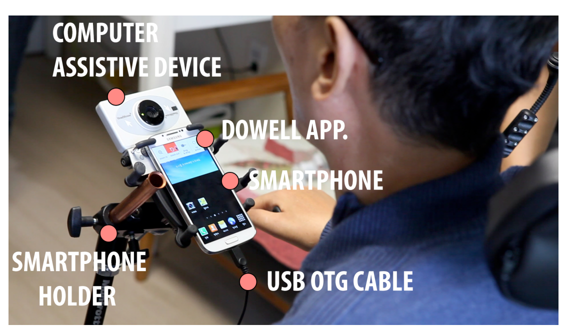

Figure 3: DOWELL System

specified direction or degree. The two-step gestures include only scroll and swipe. To scroll, a user selects a starting reference point using dwell and "scrolls" by moving their cursor in the specified direction. The speed of this scroll is indicated by the distance from the reference point on the screen. Similar to scroll, swipe needs to select a reference point using dwell. Once this reference point is selected so the user can swipe in any direction using dwell time to select a direction. All of these gestures have unique cursors help a user know what gesture they are currently using.

4.2.3 Future Work

Both the Octopus Launcher and DOWELL are relatively new assistive solutions for those who cannot interact with touchscreen smartphones due to motor impairments. Findings from the Octopus Launcher [2] show user preferences, while features in DOWELL [3] show potentially new concepts or ideas. One user preference is to use existing HID for control such as a powered wheelchair's joystick. In DOWELL, we see this as being a possibility. This interchangeability can save users who have these devices additional costs.

Another preference by users was for the ability to use "the same apps", meaning they can perform the same information activities as ordinary users do such as interact with Facebook, Email, or other common applications. We see this in both applications. In the Octopus Launcher, the researchers choose to develop a framework instead of enhancing individual applications to aid in the interaction and allow this interaction with existing applications. DOWELL has similar solution to this, but keeps the existing layout on the smartphone with additional menus. Comparing the two in this area, we see more benefits with DOWELL since its menus are hidden away when not in use. This allows for more screen space to be viewed when the menus are closed.

A final preference by users from Octopus Launcher, is that it is important to minimize errors, because correcting these errors can be costly. This is also seen in both solutions. With the Octopus Launcher, a user can cancel a selection by simply going "back" with their joystick while in the pop-up menu or can select to go back a page within the pop-up menu. DOWELL also has a quick fix for canceling a selection by simply moving to the left or right with their assistive device while in a dwell selection or can select to go back a page within the bottom menu.

5. CONCLUSIONS

In this paper we have further explored research in touchscreen interaction for motor-impaired users. We explained what motor impairments are and what current AT are being used to help aid motor-impaired users in computer and touchscreen interaction. These included switches, HIDS, and software. We discussed research in motor-impaired touchscreen interaction and how this research can help increase AT adoption rates. Finally, we talked about new AT programs being developed. These included TouchGaurd, Octopus Launcher, and DOWELL. The new AT described in this paper were cost effective solutions, but they are by no means the only solutions being researched today.

Acknowledgments

I want to thank Chris Dallager Coordinator of Disability Services at UMM and Rene Seykora at Device and Software Support at UMM for helping explore my research in assistive technologies.

6. REFERENCES

[1] A. Accessibility. Android accessibility features, 2015. `https://support.google.com/accessibility,[Online;accessed27-October-2015]`.

[2] H. Ahn, Y. Kim, and B. Suh. Use Octopus Launcher like your hands: Joystick-based smartphone control solution for motor impaired people in electric wheelchairs. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*, pages 1225–1230. ACM, 2014.

[3] H. Ahn, J. Yoon, G. Chung, K. Kim, J. Ma, H. Choi, D. Jung, and J. Lee. DOWELL: Dwell-time based smartphone control solution for people with upper limb disabilities. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, pages 887–892. ACM, 2015.

[4] L. Anthony, Y. Kim, and L. Findlater. Analyzing user-generated YouTube videos to understand touchscreen use by people with motor impairments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 1223–1232, New York, NY, USA, 2013. ACM.

[5] Apple. Accessibility: We make it easy to push all the right buttons., 2015. `http://www.apple.com/accessibility/ios/#motor-skills,Online;accessed27-October-2015]`.

[6] P. Carrington, A. Hurst, and S. K. Kane. Wearables and Chairables: Inclusive design of mobile input and output techniques for power wheelchair users. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3103–3112. ACM, 2014.

[7] K. Deibel. A convenient heuristic model for understanding assistive technology adoption. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*, page 32. ACM, 2013.

[8] A. Hurst and J. Tobias. Empowering individuals with do-it-yourself assistive technology. In *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*, pages 11–18. ACM, 2011.

[9] M. Naftali and L. Findlater. Accessibility in context: Understanding the truly mobile experience of smartphone users with motor impairments. In *Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility*, pages 209–216. ACM, 2014.

[10] E. Schüler, B. P. Salton, A. P. Sonza, A. R. Façanha, R. Cainelli, J. Gatto, L. Kunzler, and M. da CC Araújo. Production of low cost assistive technology. In *Proceedings of the Fifth International Conference on Management of Emergent Digital EcoSystems*, pages 297–301. ACM, 2013.

[11] I. N. Society. Motor impairment, 2014. `http://www.neuromodulation.com/motor-impairment,[Online;accessed22-October-2015]`.

[12] WebAIM. Motor disabilities, 2015. `http://webaim.org/articles/motor/motordisabilities,[Online;accessed27-October-2015]`.

[13] Wikipedia. Assistive technology — wikipedia, the free encyclopedia, 2015. `https://en.wikipedia.org/w/index.php?title=Assistive_technology&oldid=684333558,[Online;accessed27-October-2015]`.

[14] Y. Zhong, A. Weber, C. Burkhardt, P. Weaver, and J. P. Bigham. Enhancing Android accessibility for users with hand tremor by reducing fine pointing and steady tapping. In *Proceedings of the 12th Web for All Conference*, page 29. ACM, 2015.