Developing Steady Clicks: A Method of Cursor Assistance for People with Motor Impairments

Shari Trewin, Simeon Keates
IBM T.J. Watson Research Center
P.O. Box 704, Yorktown Heights
NY 10598, USA
+1 914 784 7616/7565
{trewin, lsk}@us.ibm.com

Karyn Moffatt
University of British Columbia
201-2366 Main Mall
Vancouver, BC V6T 1Z4
+1 604 827 3988
kmoffatt@cs.ubc.ca

ABSTRACT

Slipping while clicking and accidental clicks are a source of errors for mouse users with motor impairments. The Steady Clicks assistance feature suppresses these errors by freezing the cursor during mouse clicks, preventing overlapping button presses and suppressing clicks made while the mouse is moving at a high velocity. Evaluation with eleven target users found that Steady Clicks enabled participants to select targets using significantly fewer attempts. Overall task performance times were significantly improved for the five participants with the highest slip rates. Blocking of overlapping and high velocity clicks also shows promise as an error filter. Nine participants preferred Steady Clicks to the unassisted condition. If used in conjunction with existing techniques for cursor positioning, all of the major sources of clicking errors observed in empirical studies would be addressed, enabling faster and more effective mouse use for those who currently struggle with the standard mouse.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *input devices and strategies*

General Terms

Performance, Experimentation, Human Factors.

Keywords

Mouse, clicking, clicking errors, target acquisition, pointing and selection tasks, disability, user input.

1. INTRODUCTION

Accurate pointing and clicking with a computer mouse can be a challenge for some users. Though hardware solutions to meet the needs of motor impaired individuals do exist, many people still prefer to use a standard mouse. They may share a computer with others who use a mouse, or have used a mouse before acquiring a disability. They may find alternatives too expensive, or simply find that the mouse is easier to understand than other devices.

ASSETS'06, October 22–25, 2006, Portland, Oregon, USA.

As a result, software that supports mouse users in pointing and clicking tasks is a valuable accessibility tool. While a number of researchers have investigated techniques to support easier and more accurate pointing, little work has addressed problems with clicking. This paper describes clicking problems observed in an empirical study of older adults and people with Parkinson's Disease. It presents Steady Clicks – a clicking assistance technique intended to tackle some of the problems observed. Results from an evaluation of Steady Clicks with the target population are presented and discussed.

2. MOUSE CLICKING PROBLEMS

Studies of the effects of age and disability on mouse use have identified that advanced age and disabilities make mouse use and movement increasingly inaccurate [3, 10, 11]. For example, Trewin and Pain [11] found that 14 of 20 participants with motor disabilities had error rates greater than 10% in a point-and-click task. Participants had difficulty positioning the cursor over small targets, and keeping the cursor over the target while clicking. Many of the participants also clicked the mouse button unintentionally before reaching the target.

In an exploratory study, Keates, Trewin and Paradise [7] examined pointing and clicking performance for young adults, adults, older adults, and a group of adults with Parkinson's disease. The results of this study indicated that older adults, and adults with Parkinson's disease, used very different movement strategies to the adults and young adults. They showed lower peak velocities and used many more sub-movements, with multiple pauses during the movement. These pauses were associated with movement around and through the target.

210 incorrect clicks were observed in the study, classified as follows:

- Near misses: the mouse down position was within 50% of the target radius (110)
- Not-so-near misses: the mouse down position was between 50% and 100% of the target radius (35)
- Slips: the mouse button was pressed when the cursor was on the target but the cursor slipped off the target before the button was released (32)
- Accidental clicks: unintentional clicks, defined as clicks made at a distance > 200% of the target radius, or cases where the user presses down a button, and then presses another button before releasing the first button (9 + 5)
- Middle button press: the user pressed the wrong button (2)

A further 17 incorrect clicks were unclear. The majority of incorrect clicks were made by older adults (112). Perhaps surprisingly, young adults made as many click errors as individuals with Parkinson's disease – 35 compared with 34. All 32 of the slips, and 13 of the accidental clicks were made by the older adult and Parkinsons' groups. In interviews [9], five of the six study participants with Parkinson's disease reported that accidental clicks were a problem for them. Two reported click slips to be a problem.

No visual feedback was provided when the cursor was over the target, and the border of the targets presented was not sensitive. The older adults in particular had difficulty seeing when the cursor was in the target. This contributed to the high number of near misses observed in the data.

3. SUPPORT FOR TARGET SELECTION

A number of support techniques to aid cursor positioning have been proposed, including dynamic target expansion [8], area cursors [5, 12], 'sticky' targets [12] or the use of crossing actions instead of clicking to make selections [1].

Dynamically expanding targets as the cursor approaches has been shown to improve performance for non-disabled users [4, 8]. However, the technique assumes that the user is moving accurately towards the target. Targets near to the cursor are expanded, while surrounding targets must either move out of the way or be occluded. For users with motor impairments, or older users who make many more sub-movements around the target than younger adults, this is likely to frequently cause the user's intended target to either move or be occluded and it is not clear that any performance gain could be derived.

An alternative to target expansion is to expand the area covered by the cursor itself [12], or to use a 'bubble cursor' that dynamically adjusts the cursor area [5] such that the cursor remains large, but only covers one target at a time. Enlarging the cursor aids target selection in a similar way to making the target bigger, without requiring that every target be increased in size. Area cursors have been shown to be helpful for older adults [12]. Bubble cursors have been demonstrated to be helpful in a 2D selection task with multiple distractor targets, with young, non-disabled people [5].

Another support technique is to make targets 'sticky' by adjusting the ratio between mouse movement and cursor movement when the cursor moves over them slowly The amount of stickiness can be constant [12] or adjusted according to the semantic importance of the target [2]. This technique has been demonstrated to help older adults acquire small targets [12]. However, these experimental results relate to an artificial task in which only one distractor target needed to be crossed in order to reach the correct target. In [2], the task was a one-dimensional pointing task with no distractor targets. In real world tasks with many other targets on the screen, users who use many sub-movements in pointing may frequently find the cursor sticking on unwanted targets.

Another approach is to dispense with button clicking altogether and use target crossing actions, perhaps in combination with button pressing, to make selections [1]. One comparison [4] found that goal crossing was faster but had higher error rates than sticky targets and was unpopular with (non-disabled) users. It seems likely that the advanced motor demands of goal crossing would make it more difficult than clicking for many individuals with motor impairments.

All of these techniques are designed to help a user get a cursor onto a target. If their demonstrated benefits can be shown to extend to real applications with multiple small targets in close proximity (e.g. toolbar buttons), and to users with motor impairments, they may significantly reduce near misses and not-so-near misses. Target expansion, area cursors and sticky icons may also reduce slip errors by helping to keep the cursor within the target while clicking, at least for targets that are not able to be dragged. However, none of these techniques tackle accidental clicks or wrong button presses.

This paper proposes a general assistance technique that tackles slips and accidental clicks. It could be used in conjunction with these existing techniques.

4. THE STEADY CLICKS FEATURE

Steady Clicks is designed to help in situations where people successfully click down on a target but *slip* before releasing the mouse button, *accidentally click* buttons while en-route to the target, or click while trying to press a different button. Our software helps prevent slips by freezing the cursor at the button down location until either the button is released — causing a *steadied click* to occur, or the mouse is moved beyond the *freeze threshold* — returning it to normal operation. The *freeze threshold* defines the maximum Euclidean distance (currently 100 pixels) the cursor can travel away from the button down location. Within the region defined by this threshold, the cursor remains frozen and any button up events that occur do so at the button down location. While frozen, the cursor jitters slightly to provide feedback to the user that freezing is occurring. As soon as the cursor moves beyond this region, the freezing is broken and the cursor jumps (initiating a drag if appropriate) to where it would be, had freezing not occurred. This provides assistance that is unobtrusive, with minimal observable difference from standard cursor operation.

Steady Clicks identifies *accidental clicks* based on two criteria: velocity and button status. It ignores all clicks that occur while the mouse is moving above the *velocity threshold* or while another mouse button is pressed. Velocity is calculated using the naïve algorithm: Euclidean distance moved since previous mouse event divided by time elapsed since that event. The *velocity threshold* (currently 0.25 pixels per millisecond) was derived from the clicking data from the earlier exploratory study. Although more accurate velocity calculations are possible, the approach chosen has the advantage that it does not need data from the future *and is computationally quick to perform*. This allows decisions to be made at the time the click event is received, producing a more responsive behavior.

5. EVALUATION

An evaluation was performed to compare target selection using Steady Clicks to target selection without assistance. The evaluation was done by mouse users who experience the specific issues that Steady Clicks is designed to tackle: accidental clicks and slipping while clicking.

The following hypotheses were used:

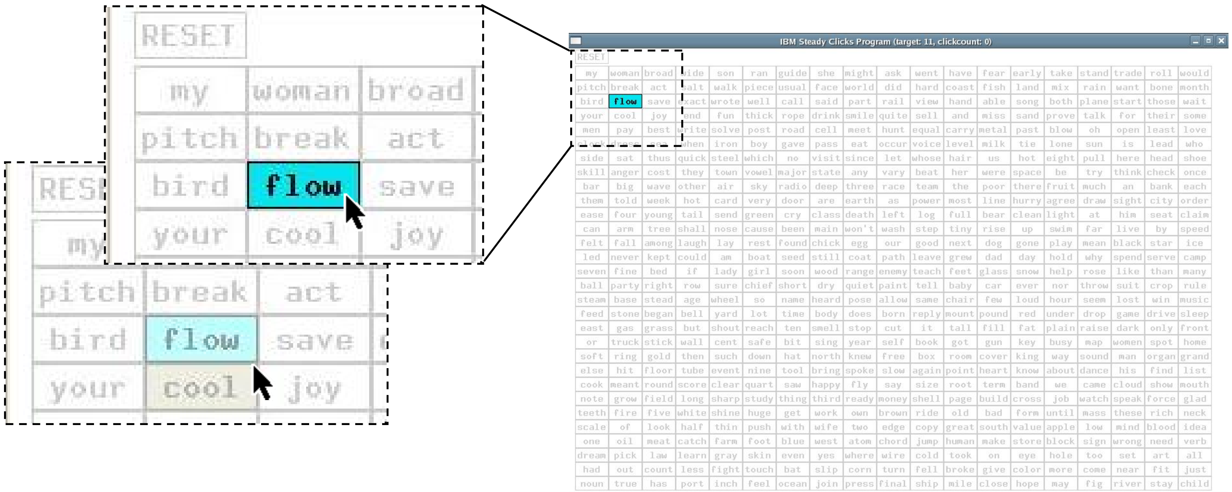

Figure 1: Screen shot of the evaluation task showing how the target underneath the cursor is highlighted when it is the desired target (upper magnification, cursor on 'flow') and when it is not (lower magnification, cursor on 'cool').

1. Participants will be able to click more accurately with Steady Clicks
2. Participants will prefer clicking with Steady Clicks as they won't have to concentrate as much on clicking

5.1 Participants

Eleven individuals with motor impairments (5 women and 6 men) participated in the evaluation, summarized in Table 1. They were aged between 27 and 80 with average and median ages of 49 and 44 respectively. Two of the participants had Parkinson's disease, two had cerebral palsy, three had impairments resulting from a stroke, one had multiple sclerosis, another had spinal damage resulting from a gunshot accident, and two had impaired manual dexterity caused by unspecified neuromuscular conditions.

Four participants used a computer daily, five used a computer several times a week, and two used a computer once a week or less often. Four had more than 10 years of computer experience. The remaining 7 had 0.5-5 years of experience. All were familiar with the standard mouse, and used that as an input device.

All gave informed consent and were paid $50 for participating. Potential participants were screened in advance to identify individuals for whom the Steady Clicks feature would be most relevant. Where possible this screening was done by observing them using the mouse and asking them to use a clicking program that measured how much they slipped the mouse while clicking. A total of 18 individuals were screened in this way, and those who slipped the most while clicking, or made any accidental click during the screening, were included in the study. For participants E1 and E3 this screening was not possible, and they were included based on a telephone interview.

5.2 Materials

During the evaluation session, participants completed a vision test, a semi-structured interview, a clicking task performed both with and without the Steady Clicks feature, and a dragging task.

All computer tasks were presented on an IBM T41p ThinkPad running the Linux Fedora Core 3 operating system, and using an IBM 3 button optical mouse with scroll wheel.

Participant	Gender	Age	Disability	Computer Experience
E1	M	77	Parkinson's	50 years
E2	F	27	Cerebral Palsy	11 years
E3	M	80	Parkinson's	30-40 years
E4	M	32	Cerebral Palsy	5 years
E5	M	50	Multiple Sclerosis	2 years
E6	M	61	Neuromuscular condition	None post-disability
E7	F	54	Spinal injury	6 months
E8	F	36	Stroke	2 years
E9	F	38	Stroke	1 year
E10	F	44	Stroke	1 year
E11	M	44	Neuromuscular condition	35 years

Table 1: Participants in the evaluation.

5.2.1 Visual acuity test

To ensure poor visual acuity would not affect task performance, a short vision test was designed. Existing tests were considered, but as the study was performed offsite, the setup, calibration and test time required would have significantly reduced the time available for data gathering. Moreover, precise vision data were not considered necessary. The test consisted of a series of short phrases presented to the participant in succession. As the test progressed, the font size of the phrases got smaller. The phrases were taken from popular nursery rhymes, with a few words selectively changed in each phrase to help ensure that the participant was reading and not reciting from memory. The smallest font size used in the test was 9 point; in the study the smallest font-size used was 18 pt.

5.2.2 Interview

A semi-structured interview was used to:

- gather information about each participant's background, computer experience and web browsing behavior;
- record the participants' subjective impressions and preferences during the clicking and dragging tasks;
- record observations made by the researchers during the session.

5.2.3 Clicking task

The clicking task interface consisted of a 19-column by 30-row grid of rectangles. Each rectangle was 52 pixels wide by 22 pixels tall, and had a two to five character word printed on it in 18pt font. A screen shot of the interface is shown in Figure 1.

For each trial, a single rectangle was selected to be the target and highlighted in blue; participants were instructed to click on the blue target. Once the blue target was successfully clicked on, the system automatically advanced to the next trial: the blue highlighting was removed from the previous target, and a new target was selected and highlighted. At all times, the target currently under the cursor was indicated with a strong dark border and gray shading. Blue highlighted targets changed to a brighter blue color when the cursor was over them, as shown in Figure 1, upper magnification.

The clicking task was designed to make it clear where participants were supposed to be clicking, and when they were over the correct target, to reduce the opportunity for errors in which the participant intentionally clicked in the wrong location.

Trials were grouped into sets of 37 targets. Each set of targets was dynamically generated to include a balanced number of short, medium and long movements presented at different angles from the previous target (the presumed starting cursor position). Each set of trials presented a different set of targets and used a fresh word grid. The participants did not know what the next target position or the target words would be.

In order to obtain more natural clicking data, participants were asked to remember the words of the targets they clicked on. The intention was to engage the participants, so that they did not focus entirely on clicking, as they had to devote some of their attention to trying to remember the words. At the end of each set of 37 trials, participants were presented with a list of nine words, informed that three of the words had been targets in the last set, and asked to try to identify those three. A different set of nine words was used each time. Additional realism was introduced by ensuring the user was aware of the errors they were making, and by making the penalty for errors parallel the penalty inherent in real-world tasks (e.g. web browsing). Therefore, the following penalties were associated with incorrect actions:

Drags or left button clicks outside the blue target: Caused the task to halt — restarting required clicking on a 'RESET' button in the upper left corner of the screen (shown in the magnified section of Figure 1). This was considered analogous to using the back button in a web browser.

Right or center mouse button clicks: Caused a window to pop-up—clicking again hid the window. This was considered analogous to triggering a popup menu.

5.2.4 Dragging task

One possible disadvantage of Steady Clicks is the potential for it to interfere with dragging. A brief dragging task was included in order to get a qualitative impression of participants' initial reactions. This task employed the same rectangle grid as the clicking task. For each trial, two rectangles were highlighted and verbal instructions were given to drag one of the highlighted rectangles on top of the other, and then to press the reset button. The task consisted of five drags including drags shorter and longer than the freeze threshold.

5.3 Method

A repeated measures design was used, with data from each participant being collected during a single 90-minute session. Each session began with a description of the session, request for consent, and the vision test. This was followed by two sets of 37 trials for each type of clicking (with or without Steady Clicks). The presentation order of the clicking types was counter-balanced to offset learning and fatigue affects. When starting a new condition, participants were able to first practice on a few targets.

Each set of clicking tasks was separated by a short segment of the semi-structured interview, to reduce fatigue effects, and to allow participants to provide feedback while the task was still fresh in their minds. The interviews were conducted verbally with responses being recorded by the experimenters. Again, this was done to minimize the physical effort demanded of the participants. After completing these tasks, the dragging task was presented, and a final interview gathered user feedback on dragging while using Steady Clicks.

When using Steady Clicks, participants were told "this program will keep the mouse steady while you are clicking and ignore any clicks you make while moving the mouse or clicking other mouse buttons". This information was provided to enable the participants to adjust their clicking strategies if they chose. This is also the most likely real world scenario – an individual already familiar with clicking with a mouse would then try using Steady Clicks, with some idea of what behavior to expect. When not using Steady Clicks, participants were told "this program operates as a regular mouse".

All of the participants were able to comfortably see text at the 18pt font size used in the experiment, but several had cognitive difficulty in reading. For these participants, we read out the words on the targets as they clicked on them, to reduce the cognitive load to a more comfortable level.

5.4 Data collected

In addition to the interview data, the clicking task program recorded time-stamped log files detailing each participant's cursor movements and mouse button presses. The log files also detailed the actions taken by the Steady Clicks feature, the positions of the targets being clicked on, all movements to and clicks on the 'RESET' button, and clicks made in order to clear the popup window.

From the clicking task, with practice sessions excluded, a total of 1554 trials were recorded, including 740 trials without Steady Clicks and 814 trials with Steady Clicks. The difference is due to two participants (E5 and E11) who found the 'without Steady Clicks' condition so tiring that they could not complete all the trials, and an additional seven trials for E2 that were lost due to a program crash. Both E5 and E11 used the 'without Steady Clicks' condition first, then went on to successfully complete two sessions using Steady Clicks.

For data analysis, the time spent in each trial was split into three states:

- '**Target acquisition'** state in which the participant was being presented with a target to click on
- '**Reset'** state in which the participant needed to press 'reset'
- '**Popup**' state in which the popup window was displayed, and the participant needed to click to clear the popup before proceeding with the trial.

6. RESULTS

6.1 Overall times

Table 2 gives an overview of the times taken, in seconds, with (W) and without (WO) Steady Clicks, showing each participant's average time taken per trial, and time spent in target acquisition, reset and popup states per trial. There was no significant effect of the order in which conditions were presented for average time per trial (p=0.12), or time spent in target acquisition state (p=0.15).

	Average time per trial (sec)							
	Overall		Target acquisition		Reset		Popup	
	WO	W	WO	W	WO	W	WO	W
E1	3.3	3.2	3.3	3.2	0	0	0.0	0.0
E2*	7.7	5.0	6.0	4.9	1.7	0.1	0.04	0.0
E3	2.8	3.3	2.7	3.1	0.1	0.2	0.0	0.04
E4	2.3	2.0	2.1	2.0	0.2	0.0	0.1	0.0
E5*	55.2	10.9	43.0	9.6	10.2	0.9	2.0	0.3
E6	11.4	7.7	9.0	7.1	2.1	0.5	0.3	0.0
E7*	4.6	2.2	3.9	2.2	0.7	0.0	0.0	0.0
E8	6.6	4.8	4.6	4.3	2.0	0.5	0.04	0.05
E9*	6.1	3.2	4.1	3.0	2.0	0.2	0.0	0.0
E10	9.0	6.1	4.7	4.6	4.3	1.5	0.0	0.0
E11*	21.0	5.3	12.6	4.8	8.3	0.5	0.1	0.0
Ave	11.8	4.9	8.7	4.4	2.9	0.4	0.2	0.04
Med	6.6	4.8	4.6	4.3	2.0	0.2	0.04	0.0

Table 2: Breakdown of average time per trial, overall and in each state, without (WO) and with (W) Steady Clicks. Participants marked with '*' are those for whom Steady Clicks produced a significant difference in total time.

10 of the participants had lower times when using Steady Clicks, but overall there was no significant difference in the average time per trial with and without Steady Clicks (single factor ANOVA, F=1.41, df=1,20, p=0.15). There was also no significant difference for time spent in target acquisition state (F=1.41, df=1,20, p=0.25) or popup state (F=1.19, df=1,20, p=0.29). However, there was a significant difference at the 5% level in time spent in reset state (F=5.69, df=1,20, p=0.027).

Looking at the participants individually, there were five users for whom the Steady Clicks condition did show a significant difference (at the 1% level) in the total times observed. These were E2 (p=0.003), E5 (p=0.005), E7 (p<0.001), E9 (p<0.001) and E11 (p<0.001), marked with '*' in Table 2. E10 also showed significance at the 5% level (p=0.028).

6.2 Clicking Activity

Figure 2 shows the frequency of each type of error and intervention averaged per 100 target activation tasks across all users and in each condition, i.e.. without and with Steady Clicks. When Steady Clicks was not active, the 224 observed misses in the WO condition (indicated by the first columns) were predominantly cases where the target was dragged (124 instances, 3rd columns) or a wrong target was dragged (55 cases, 4th columns). Clicks on the wrong target accounted for only 28 cases (2nd columns).

When Steady Clicks was active, almost all of these drags were suppressed (only a very long drag would break out of the freeze threshold, which happened 3 times). The 88 misses consisted of 51 clicks on the wrong target and 34 other clicks that were either blocked by Steady Clicks (30 cases) or were not on any target.

Because there were fewer incorrect clicks and drags in the Steady Clicks condition, only 6.6% of trials involved a reset operation, as opposed to 31.9% without Steady Clicks. These resets were also performed with fewer mouse clicks per reset: 1.26 instead of 1.73.

Table 3 gives an overview of the clicks made while in target acquisition state during each trial, showing the average number of incidences per trial for each participant without (WO) and with (W) Steady Clicks.

Clicks made while in target acquisition state that were blocked by Steady Clicks or were not on any target are included in the mouse down count but not in the other columns. The average number of mouse down events per trial ranged from 1 for E1, who made no errors, up to 4.89 for E11. Overall, the participants made on average 1.99 clicks per trial. In Table 3, participants are listed in ascending order by the rate at which they dragged the target when not using Steady Clicks. It also shows which condition was presented to each participant first. It happened that the three participants with the least difficulty (including the two who had not been screened using the software tool) were all in the Steady Clicks first condition, and that three of the four with the greatest difficulty used the Steady Clicks second condition. Despite this, there was no significant effect of the order in which conditions were presented for mouse downs per trial (p=0.14), due to the small number of participants.

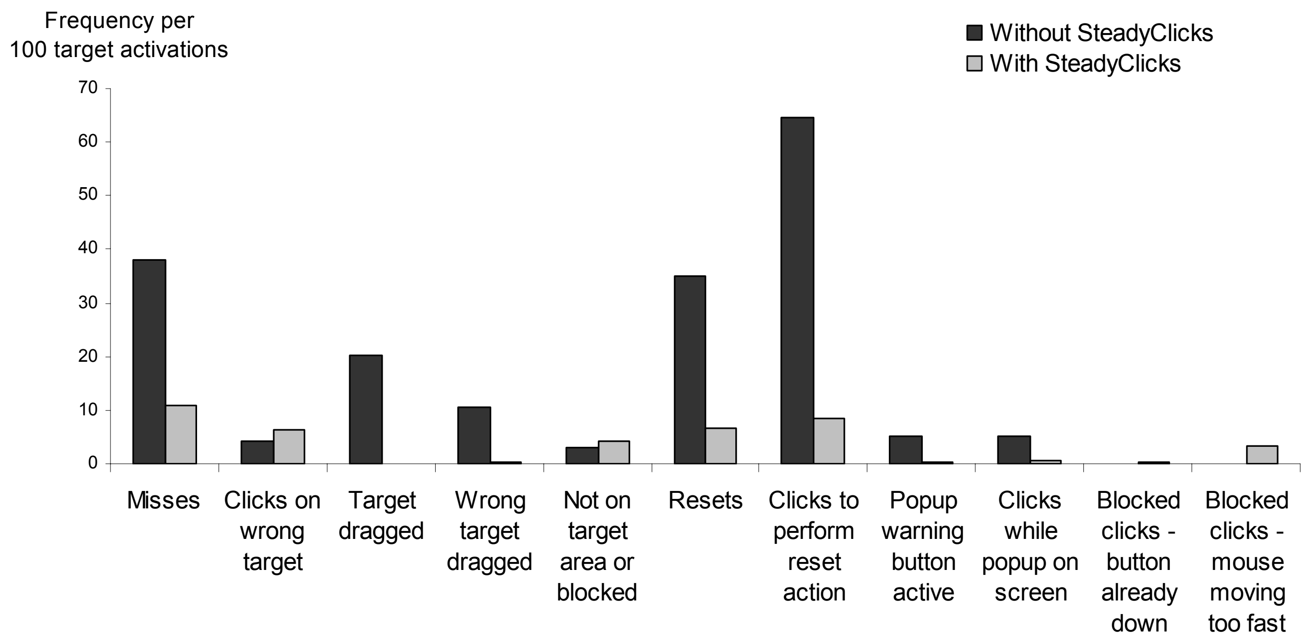

Figure 2: Frequency of errors and interventions per 100 target activations

	Cond-ition (interface) used first	Averages per trial					
		Mouse downs		Drags on target		Clicks or drags off target	
		WO	W	WO	W	WO	W
E1	W	1.00	1.01	0.00	0.00	0.00	0.00
E3	W	1.04	1.08	0.01	0.00	0.00	0.01
E4	W	1.19	1.00	0.01	0.00	0.00	0.00
E7	WO	1.35	1.01	0.11	0.00	0.05	0.00
E2	WO	1.55	1.04	0.13	0.00	0.04	0.01
E8	W	1.62	1.35	0.14	0.00	0.12	0.08
E6	WO	1.84	1.20	0.14	0.00	0.09	0.09
E10	W	2.81	1.88	0.23	0.00	0.35	0.31
E5	WO	2.59	1.24	0.27	0.00	0.16	0.08
E9	WO	2.05	1.14	0.34	0.00	0.12	0.04
E11	WO	4.89	1.28	0.75	0.00	0.43	0.09
Ave		1.99	1.20	0.19	0.00	0.13	0.07

Table 3: Clicks made in target acquisition state without (WO) and with (W) Steady Clicks.

Not surprisingly, there was a highly significant overall effect on the number of times the target was dragged instead of clicked (F=8.99, df=1,20, p=0.007), since small drags were suppressed by Steady Clicks. There was also a significant difference at the 5% level in the number of mouse downs per trial (F=5.15, df=1,20, p=0.035), and number of times the wrong target was dragged (F=6.55, df=1,20, p=0.019), but not for the number of times the wrong target was clicked (F=3.98, df=1,20, p=0.06).

6.3 Wrong mouse button presses

Without Steady Clicks, participants pressed a wrong mouse button 27 times, causing a warning popup to appear. Of these, 13 were a result of two button presses that overlapped in time. Two of these had the wrong (middle or right) button going down first, while the other 11 had the intended (left) button going down first. For E5 and E11, there was a sequence of many errors in a row, where the user's finger had moved onto a position between two buttons (E11) or on the wrong button (E5) and they did not realize. Their first wrong button press caused a popup to appear, and their next press of any button cleared it. Sometimes this was the overlapping button press, so for example E11 produced and cleared the popup in a single action, and then did the same thing again, several times in a row. E5 kept pressing the scroll wheel, which caused a popup to appear and clear repeatedly. E5 and E11 account for 15 of the wrong mouse button presses, including 7 of the 13 examples of overlapping button presses.

When Steady Clicks was active, 6 wrong mouse button presses were observed. Of these, 50% were successfully blocked by Steady Clicks. Overlapping button presses occurred 3 times, with 2 of these having the left button going down first and being blocked by Steady Clicks. Steady Clicks blocked a third wrong button press that occurred at high velocity.

6.4 High velocity clicks

28 clicks were blocked because the mouse velocity was too high prior to the mouse down event. Only 17 of these blocked clicks were genuine incorrect clicks, and these clicks were made by E10, E8 and E5. A further 11 clicks were incorrectly blocked by Steady Clicks. Nine of these errors occurred because the velocity calculation is based on the time period between two consecutive mouse events, and there were instances where this was only 4 msec or less. At such small time differences, even 1 pixel of

movement caused the calculated velocity to exceed the threshold. This is an artifact of the operating system event reporting mechanism. The velocity calculation should be smoothed over a longer time period, or else movements of only 1 pixel should not be considered high velocity movements. Of these nine blocked clicks, 7 were actually over the target. Two further clicks were wrongly blocked due to a program bug (neither of these was on the target). Overall, the click blocking feature suppressed 17 errors but introduced a further 7 errors.

6.5 Participants' experiences

There were no significant differences between conditions in the number of participants who noticed the mouse slipping while clicking (8 without and 7 with Steady Clicks), or who noticed accidental clicks (5 without and 6 with Steady Clicks). Five participants (E2, E4, E7, E9, E10) thought Steady Clicks had helped prevent them from slipping, and three (E8, E9, E10) thought it had filtered out unwanted clicks. No participant reported deliberate clicks being filtered out. One participant (E9) felt that Steady Clicks interfered with moving the cursor.

Eight participants felt that they worked faster with Steady Clicks, two (E8 and E10) reported the 'without' condition to be fastest, and E3 reported no difference between the conditions. Seven participants felt that they had fewer errors while using Steady Clicks, one (E3) reported fewer errors without, and three (E1, E2, E7) reported no difference between the conditions. Overall, nine preferred Steady Clicks, one (E8) preferred without and one (E3) had no preference.

One participant (E5) described their clicking strategy without Steady Clicks as being to put the cursor in the corner of the target to compensate for upward slippage. He explained that he had to keep looking at his finger to see where it was positioned on the mouse. His index finger had a tendency to move to the right, sometimes causing a middle button click. He said that he could not predict, from looking at the screen, whether a click would work or not, and commented that it "takes a lot of time, so gets frustrating, because you want to be moving on. I can't feel where my finger is, so I get into position but then find my finger is not over the button". When using Steady Clicks, he commented "This is a lot easier because I'm thinking that the mouse isn't going to move. I feel a little confident [*sic*] in the program that I can move faster." He described that he was no longer looking down at his finger as much because he was working faster, so there was less finger slippage. He said "Now I'm not even going in the corner anymore. I'm just getting in the spot and moving on." He was initially not sure how much this was due to practice, and how much to the Steady Clicks feature. By the end of the clicking task his eyes were tiring. He expressed a clear preference for Steady Clicks, but commented that sometimes he became overconfident.

E11 reported a similar strategy of aiming at the top right of each target. She felt that this reduced slippage. She said "I have a habit of holding the button down too long. If I hold it down for half a second instead of 1 second it doesn't move so much." When using Steady Clicks she commented that it "… didn't seem to slip as much as it did earlier". E7 commented that there are "… *some [mice] that when you move it jiggles and goes to another place. This one doesn't. This one stays where you want to put it.*"

6.6 Dragging task

Participants E6 and E11 did not do the dragging task. E6 had never done dragging before. E11 ran out of time in his session.

From observation of the other participants performing long and short drags with Steady Clicks active, it was clear that Steady Clicks did interfere with dragging. Participants would start to drag, and then stop and start again because nothing seemed to have happened. However, after several attempts, without instruction, all succeeded in dragging the targets. Two participants commented that the delay before the drag would start did make dragging more difficult, but that this would not necessarily stop them from using the utility, if it was helpful in normal clicking. Three commented that they do not normally do dragging anyway. Surprisingly, 4 participants did not notice any difference at all when dragging with Steady Clicks, versus normal dragging.

7. DISCUSSION

When using Steady Clicks, participants had significantly fewer mouse downs per trial, and spent significantly less time in the reset state trying to correct errors. Time spent in the reset state of this task is effectively a magnification of the extent of an individual's problem, since it is caused every time an error is made, and compounded by further errors while clicking the RESET button. Highly significant time savings were observed for 5 individuals. The benefit of Steady Clicks for these individuals is due to the slip blocking effect. Those with the highest rates of slipping while clicking had the greatest benefit.

The click errors observed break down very differently to those in our previous study [7]. Without Steady Clicks, 55% of click errors were entirely due to slipping (versus 15% previously), and 37% were misses (versus 69% previously). This is partly a reflection of the difference in participant population — this study specifically sought out participants for whom slipping was a problem. It is also partly a result of the task itself. Several participants commented that the strong visual confirmation of which target the cursor was over was very helpful, and made clicking easier than the clicking they normally do in other applications. This may have significantly reduced the number of errors that were misses. Furthermore, the targets used in the current study were larger than the smallest targets used in [7], for which the majority of misses occurred.

These participants did not make many accidental clicks — most qualified for the study based on their click slipping behavior. As a result, blocking of high velocity and overlapping clicks was less helpful for these individuals. Performance could be improved by modifications to the algorithm used. The individuals themselves were aware that they made accidental clicks, and two correctly observed that Steady Clicks had filtered out unwanted clicks for them. None reported noticing Steady Clicks' mistakes.

Both practice and fatigue played a role in these results. Practice effects were inevitable, since the participants performed the tasks on an unfamiliar machine, with a mouse provided for them, and were using a program they had never used before. Fatigue effects were also inevitable for some users, because a 90-minute session performing a difficult task was a significant undertaking. Some were unable to complete the task. Our experimental design attempted to control for these effects, but the small number of participants reduces the effectiveness of this counterbalanced within-subjects approach when individual variability is so high. Using separate sessions for the different participants may have helped, but was not within our budget.

Our preliminary look at dragging suggests that although Steady Clicks has a negative impact on dragging, for many users this may not be a big issue. Some users rarely do dragging. Others did not notice the effect, perhaps because they dragged quite quickly and broke the threshold before noticing the freeze. It is encouraging that users were able to complete the dragging tasks without help. Stronger visual feedback of the Steady Clicks freezing effect may be helpful.

The evaluation task required the user to take action to correct all accidental and slipped clicks, and this action was the same for every error. In real user interface tasks, accidental and slipped clicks have highly variable consequences. One slip may remain within the tolerance of the button being clicked, so that the click is successful, while another causes a folder to be dragged into another folder, or the focus to move to a different window. It is very difficult and time consuming to recover from such errors, because the user is often not sure what action they performed, or even aware that they have taken any action. The most important benefit of Steady Clicks for everyday use would be to prevent this kind of "I think I just did something but I don't know what" problem.

8. CONCLUSIONS

Our previous studies showed that slipping while clicking and accidental clicks are a source of errors for older adults and individuals with disabilities when using a mouse. The Steady Clicks assistance feature suppresses these errors by freezing the cursor during clicks, preventing overlapping button presses, and suppressing clicks made while moving at a high velocity.

Steady Clicks was evaluated with a group of individuals for whom slipping while clicking was the primary source of selection errors. Participants were able to select targets using significantly fewer attempts when using Steady Clicks. Overall task performance times were significantly improved for the five participants with the highest slip rates. Accidental click rates were low in both conditions. The evaluation suggested that blocking overlapping and high velocity clicks could be an effective error filter. Some tweaking of the algorithm used is required. Most participants preferred Steady Clicks to the unassisted condition. One reported that he was able to change his clicking strategy when using Steady Clicks, which allowed him to work more quickly.

The effect of Steady Clicks on dragging is potentially disruptive and more work is required to examine ways to reduce this disruption. These results suggest that some users would be willing to accept this disruption in return for easier clicking.

Steady Clicks could be used in conjunction with existing techniques for cursor positioning such as area and bubble cursors. Together these would tackle all of the major sources of clicking errors observed in our studies of older adults and people with motor impairments. A combined approach would enable faster and more effective mouse use for those who currently struggle with the standard mouse.

9. ACKNOWLEDGMENTS

Our thanks to all the individuals who participated in the evaluation, and staff at the Beth Abraham Adult Day Rehabilitation Center and Cerebral Palsy of Westchester who assisted in contacting individuals and provided venues for evaluation sessions, and the Multiple Sclerosis Society, who advertised the study among their local support groups.

10. REFERENCES

[1] Accot, J. and Zhai, S. (2002) More than dotting the i's – Foundations for crossing-based interfaces. In *Proceedings of CHI 2002*, Minnesota, USA, April 2002, pp 73–80.

[2] Blanch, R, Guiard, Y., and Beaudouin-Lafon, M. (2004) Semantic pointing: Improving target acquisition with control-display ratio adaptation. In *Proceedings of CHI '04*, Vienna, Austria, pp 519–526.

[3] Brownlow, N., Shein, F., Thomas, D., Milner, M., and Parnes, P. (1989). Direct manipulation: Problems with pointing devices. In Resna '89: Proceedings of the 12th Annual Conference,. Washington DC: Resna Press, pp 246–247.

[4] Cockburn, A. and Firth, A., Improving the Acquisition of Small Targets. in *People and Computers XVII* (Proceedings of the 2003 British Computer Society Conf. on Human-Computer Interaction.), (Bath, England, 2003), 181–196.

[5] Grossman, T. and Balakrishnan, R. (2005) The bubble cursor: Enhancing target acquisition by dynamic resizing of the cursor's acivation area. In *Proceedings of CHI '05*, Portland, OR, USA, pp 281–290.

[6] Keates, S., & Trewin, S. (2005). Effects of Age and Parkinson's Disease on Cursor Positioning Using a Mouse . In *ASSETS '05 Proceedings of the Seventh International ACM Conference on Assistive Technologies*. 68–75.

[7] Keates, S., Trewin, S., & Paradise, J. (2005). Using Pointing Devices: Quantifying differences across user groups. In *Proceedings of the 3rd International Conference on Universal Access in Human-Computer Interaction.*

[8] McGuffin and Balakrishnan (2002) Acquisition of expanding targets. *Proceedings of CHI 2002*, Minnesota, USA, pp 57–64

[9] Paradise, J., Trewin, S., & Keates, S. (2005). Using Pointing Devices: Difficulties Encountered and Strategies Employed. In *Proceedings of the 3rd International Conference on Universal Access in Human-Computer Interaction.*

[10] Riviere, C., and Thakor, N. (1996). Effects of age and disability on tracking tasks with a computer mouse: Accuracy and linearity. *Journal of Rehabilitation Research and Development*, 33, pp 6–15.

[11] Trewin, S., and Pain, H. (1999). Keyboard and mouse errors due to motor disabilities. *International Journal of Human-Computer Studies*, 50, pp 109–144.

[12] Worden, A., Walker, N., Bharat, A., and Hudson, S. (1997) Making computers easier for older adults to use: Area cursors and sticky icons. In *Proceedings of CHI '97*, Atlanta GA, pp 266–271.