Avestia Publishing
Journal of Virtual Worlds and Human Computer Interaction
Volume 3, Year 2015
Journal ISSN: 2368-6103
DOI: 10.11159/vwhci.2015.001

Examining Dynamic Control-Display Gain Adjustments to Assist Mouse-Based Pointing for Youths with Cerebral Palsy

Amur Al Manji[1], Claire Davies[1],
[1]University of Auckland, Department of Mechanical Engineering,
Auckland, New Zealand.
aalm759@aucklanduni.ac.nz; c.davies@auckland.ac.nz
Robert Amor[2]
[2]University of Auckland, Department of Computer Science,
Auckland, New Zealand.
trebor@cs.auckland.ac.nz

Abstract- *Youths with cerebral palsy encounter difficulties in target acquisition using mouse-based "point-and-click" tasks. The performance of "point-and-click computer tasks" by youths with cerebral palsy still needs improvement to allow easy target acquisition. The purpose of this study was to evaluate an algorithm intended to enhance the performance of point-and-click computer tasks for youths with cerebral palsy using a standard mouse. The curvature index–based algorithm for dynamic adjustment of control-display gain showed experimentally better performance during primary sub-movement, but worse performance during secondary correction sub-movements (higher number of sub-movements and longer movement time) for both typically developed youths and youths with cerebral palsy. It also showed better performance of average speed and maximum speed compared with Windows default settings. Furthermore, the average movement time, error rate, and overshoot rate for both typically developed youths and youths with cerebral palsy are higher using the curvature index–based algorithm.*

Keywords: Control-display gain, Curvature Index, Point and Click, Target Agonistic Algorithm, Cerebral Palsy

1. Introduction

Point-and-click tasks are required to access computer files and resources. The performance of these tasks is affected by many factors, including fatigue, accuracy, position, and hand-eye coordination. For people with upper limb impairments as a result of cerebral palsy (CP), motor control difficulties must be overcome or effectively reduced. CP is a common cause of motor dysfunction affecting children and adults [1] and is an umbrella term for a group of disorders of movement and/or posture, which include spasticity, dyskinesia, ataxia [2]. The prevalence is approximately 3.1 per 1,000 births [1].

Three major components are required to build a sophisticated computer system: interaction methods, interaction devices, and interface design [3]. A variety of interaction methods and devices have been developed to improve physical access to computers. These range from accessibility options within the operating system to enhanced input devices, such as joysticks, trackballs, touchscreens, gesture recognition, and speech recognition software [4]. However, the use of assistive technology for better computer access encounters barriers. Barriers vary from insufficient funding, lack of staff training, and negative attitudes at school [5], whereas the main barrier in the home environment is cost [6]. According to the study of Davies et al. [7] which surveyed youths with upper limb impairments as a result of CP, it was determined that most prefer the typical mouse to access computers, either by hand for cursor movement with finger-clicking for target selection, or by enhanced computer access methods, such as foot control with toe-clicking. In addition, the study mentioned (1) that only a small group of youths with CP use assistive technologies, although a large

Date Received: 2014-09-21
Date Accepted: 2014-11-24
Date Published: 2015-02-03

group is aware of existing technologies, and (2) that a number of youths with CP have used assistive devices and technologies in the past. The study suggested that there are several reasons why youths with CP return to using a standard or modified mouse, which include the lack of provided assistive technology in an education setting [8] and fewer restrictions to access locations (including a friend's home and the public library, where the existence of assistive technology to access computers is rare). Therefore, the most pragmatic decision is to use a standard mouse at all locations.

We are proposing an algorithm to enhance the performance of "point-and-click" computer tasks using a standard mouse for youths with CP.

2. Related Work

The HCI interface development for point-and-click computer tasks focuses on visual- and/or motor space–categorizing algorithms categorised as target-aware and target-agonistic. The target-aware algorithm is based on the dimensions and location of targets, but target-agonistic dimensions are based on path evaluation measures and allow for dynamic adjustment of control-display (C-D) gain. Some target-agonistic algorithms include the Angle Mouse [9], PointAssist [10, 11], and Dynamic Mouse Speed [12], and they are used to enhance the performance of point-and-click computer tasks.

Angle Mouse [9] uses angular deviation to dynamically adjust C-D gain to enhance point-and-click tasks. In a study of 16 participants including 8 motion-impaired users, Angle Mouse improved the throughput of point-and-click tasks by 10.3% over the Windows default (C-D gain is static and was set to 10) for motion-impaired users. However, the Angle Mouse algorithm did not improve the throughput over the Windows default for typically developed (TD) users. Typically developed refers to those individuals with no impairment associated with the perception, cognition and motor processes.

PointAssist uses C-D gain to control the precision (accuracy) of point-and-click computer tasks. PointAssist enhances accuracy measures by analyzing the features of sub-movements to detect when the user experiences pointing difficulty. PointAssist triggers a precision mode that slows the speed of the cursor to enhance the accuracy measures [10].

Hourcade et al. [10] assessed the efficiency of PointAssist with 20 older adults with ages ranging from 66 to 88 years. The study showed that the elderly gain better accuracy performance by using the PointAssist algorithm than the "enhance pointer precision" mode in Windows XP.

Tang and Lee [12] proposed a dynamic mouse speed (DMS) design, whereby mouse cursor speed (C-D gain) changes dynamically once they exceeds 5 predetermined speed thresholds. The DMS algorithm is based on the number of traveled pixels within the interval between mouse interrupts. If the traveled pixels' value is greater than a predetermined threshold, the system calls a particular function to change the pointer speed. The pointer speed is the C-D gain, which ranges from 1 (slowest) to 20 (fastest). Based on 2 experiments, with 11 participants in each, the DMS setting surpassed the Windows built-in settings significantly for larger distances. However, the DMS performance did not function well for all directions, especially for short distances where the performance of Windows built-in settings exceeded the DMS performance.

This prior research has shown that C-D gain (a transfer function that translates displacement of the physical movement of the mouse into speed of the cursor on the screen and can be changed under the "pointer speed" in the mouse setting) has been controlled to increase the performance of cursor movement during pointing tasks. However, any increment of C-D will generally give a speed advantage over accuracy (speed accuracy trade-off); therefore, providing a dynamic algorithm that balances the speed-accuracy trade-off will certainly promote the performance of pointing tasks. If the dynamic algorithm does not balance the speed-accuracy trade-off, it will worsen the performance of both speed and accuracy of pointing tasks. This study seeks to address these issues in the development of an algorithm that better balances speed-accuracy trade-offs.

3. The Proposed Algorithm

We have evaluated the movement of individuals with CP in two studies. The first study showed that for individuals with CP, it is more important to focus on the development of user interfaces and algorithms to increase speed because they already appear to have sophisticated accuracy [13]. The second study examined both system (width and amplitude) and human effects (errors, slips off the target, etc) that contribute to the speed and accuracy of cursor movement of youths with CP and determined that the most significant human effect is the curvature index [14]. The curvature index is the ratio of the total distance traveled to the straight-line distance between the start and end points. A value of 1 indicates that the cursor path follows a straight line toward the target, and a larger value shows increasing deviations [15].

The movement deviation caused by a spasm can be measured by the curvature index; a higher-value curvature index causes an increase in movement time (MT).

As a result of these prior studies, our proposed algorithm is based on a differential curvature index value (CI-1), as a CI value of 1 means that there is no deviation from the start-end movement line (see figure 1). Higher CI values indicates less accuracy therefore, C-D gain should be decreased to enhance the accuracy and lower CI values indicates higher accuracy , therefore increasing the C-D gain will provide faster cursor travel. The proposed formula of C-D is

$$\begin{Bmatrix} C-D = 14-((CI-1)*4), CI <= 3 \\ C-D = 6, CI > 3 \end{Bmatrix} \quad (1)$$

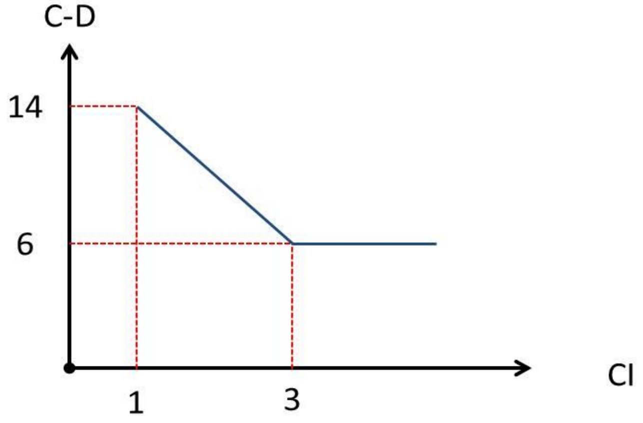


Figure 1. CI and C-D gain relationship.

The lowest value is set to 6 because the curvature index becomes high, resulting in a negative value of C-D (C-D must be greater than or equal to 1). The 6, 14 values are experimentally calibrated. The C-D gain is calculated based on a sequential set of 5 and 10 samples with a conditional distance of 16 pixels.

4. Empirical Validation and Examination of Proposed Algorithm

It was important to examine and validate the algorithm by carrying out a controlled experiment using standard discrete pointing tasks. This section covers three subsections of the experiments, including subjects, apparatus, and experimental design.

4.1. Subjects

A.1. Twelve TD subjects (8 males, 4 females, aged 18–32 years, all right handed) conducted the experiment.

A.2. Six subjects (aged 13–22 years) with CP (MACS I and II) conducted the "point-and-click" tasks.

4.2. Apparatus

The experiment was conducted on an HP laptop machine with 2.2 GHz AMD Turion™ X2 Ultra Dual-Core Mobile ZM-82 with 3GB of RAM running 32-bit Windows 7 Enterprise. The pointer speed (C-D gain) was set to 10 as normal. The screen used was at a resolution of 1280 × 800 pixels at a refresh rate of 60 Hz. The pointing device type was a standard USB optical mouse (manufactured by Dell). The resolution of the input device was at 96 pixels per inch (DPI).

4.3. Experimental Design

A multidirectional discrete pointing task was undertaken to evaluate the cursor movement between two targets with different sizes and center-to-center amplitudes. The software was developed in C# with a sampling interval of 15 milliseconds. The trial is defined as the cursor traveling from a home square located at the center of the screen to a target square located somewhere with a pre-specified movement amplitude and width. The trial starts once the user clicks on the home square and ends when the user clicks on the target square. Then the user returns to the home square to initiate a new trial. The experiment type is "with prior knowledge" of the target, which means that the target is visible to the participant before the trial begins. Table 1 summarizes the experiment setup.

Table 1. Summary of experiment configuration.

Target direction	4 directions (0°, 90°, 180°, and 270°)
Number of trials	16 trials (4 in each direction)
A*W combinations	9 combinations (3 amplitudes [120, 240, 360 pixels] with target size of 20, 30, and 40 pixels
Experiment type	Default (C-D:10) CI-based algorithm (2 window sizes: 5 samples and 10 samples)

5. Results and Analysis

The movement toward the target during the point-and click computer tasks is composed of sub-movement components: rapid primary movement and secondary correction movements.

The main speed and accuracy performance measures are MT and error rate (ER). In addition, we compare the performance of algorithms based on average speed (AS), maximum speed (MS), and MT of primary sub-movement and secondary sub-movement.

5.1. Experimental Design

The fixed factors are experiment type (ET; default, CI-based (5 samples) [CI-based(5)] and CI-based (10 samples) [CI_based(10)]), amplitude (A; 120, 240, and 360 pixels), and width (W; 20, 30, and 40 pixels). Therefore, there are seven interactions of fixed factors, which are ET, A, W, ET*A, ET*W, A*W, and ET*A*W. Subjects are selected as random factors. Independent tests were conducted for the following dependent variables:

Movement time (MT: milliseconds): The time interval from a mouse click on the home square to a mouse click inside the target square [15, 16]. The unit is milliseconds.

Average speed (AS: pixels/milliseconds): The average value of speed samples within a given trial. It is expected that the degree of impairment has an influence on AS. The unit is pixels/millisecond.

Maximum speed (MS: pixels/milliseconds): The maximum value of speed samples within a given trial. It is expected that the degree of impairment has an influence on MS. The unit is pixels/millisecond.

Primary sub-movement time (PST: milliseconds): The time interval between initial cursor movement and first cursor pause. The unit is milliseconds [17].

Primary sub-movement distance (PSD: pixels): The distance traveled between initial cursor movement and first cursor pause. The unit is pixels [17].

Secondary sub-movement time (SST: milliseconds): The total time of secondary sub-movements from the first primary sub-movement pause until the target click. The unit is milliseconds [17].

Secondary sub-movement distance (SSD: pixels): The total distance traveled by secondary sub-movements from the first primary sub-movement pause until the target click. The unit is pixels [17].

Error rate (ER: %): The percentage of erroneous clicks within the A*W combination [16].

Number of sub-movements (NS): The number of discrete movements separated by pauses, defined as a zero cursor speed. A high NS indicate more movement correction, resulting in an increase in MT. Hwang et al. [18, 19] found that TD users completed 90% of point-and-click trials in less than seven sub-movements, whereas 90% of trials of motion-impaired users required more than seven sub-movements. In our experiment, the definition of a pause was conservatively set to a cursor speed of zero because it is difficult to specifically define a sub-movement [20].

Overshoot rate (OSR: %): The percentage of trials that involved overshoot movement within the A*W combination [9].

The selected path evaluation measures provide clear insights and information on cursor movement and performance of point-and-click tasks by the CI-based algorithm and Windows default.

An analysis of variance (ANOVA) was conducted with the previously mentioned performance measures. For TD youths, the 3-way interaction effect of ET*A*W was not significant except MS ($F_{8,3853} = 2.4$, $P < 0.02$). The 2-way A*W was significant for AS ($F_{4,3853} = 14.6$, $P < 0.001$), MS ($F_{4,3853} = 2.8$, $P<0.03$), PST ($F_{4,3853} = 4.8$, $P<0.00$), PSD ($F_{4,3853} = 2.4$, $P<0.05$), SST ($F_{4,3853} = 2.9$, $P<0.02$) and SSD ($F_{4,3853} = 2.6$, $P < 0.03$). The 2-way ET*W was significant for AS ($P < 0.001$), MS ($F_{4,3853} = 6.9$, $P<0.00$), PST ($F_{4,3853} = 3.4$, $P<0.01$), PSD ($F_{4,3853} = 3.9$, $P<0.00$), SST ($F_{4,3853} = 2.4$, $P<0.05$), SSD ($F_{4,3853} = 5.1$, $P < 0.001$), NS ($F_{4,3853} = 3.3$, $P<0.01$) and OSR ($F_{4,3853} = 10.8$, $P<0.001$). The 2-way ET*A was only significant for PST ($F_{4,3853} = 3.2$, $P < 0.01$) and OSR ($F_{4,3853} = 6.6$, $P<0.00$). All main effects were significant for ET, A, and W levels, respectively, except OSR and ER for A level.

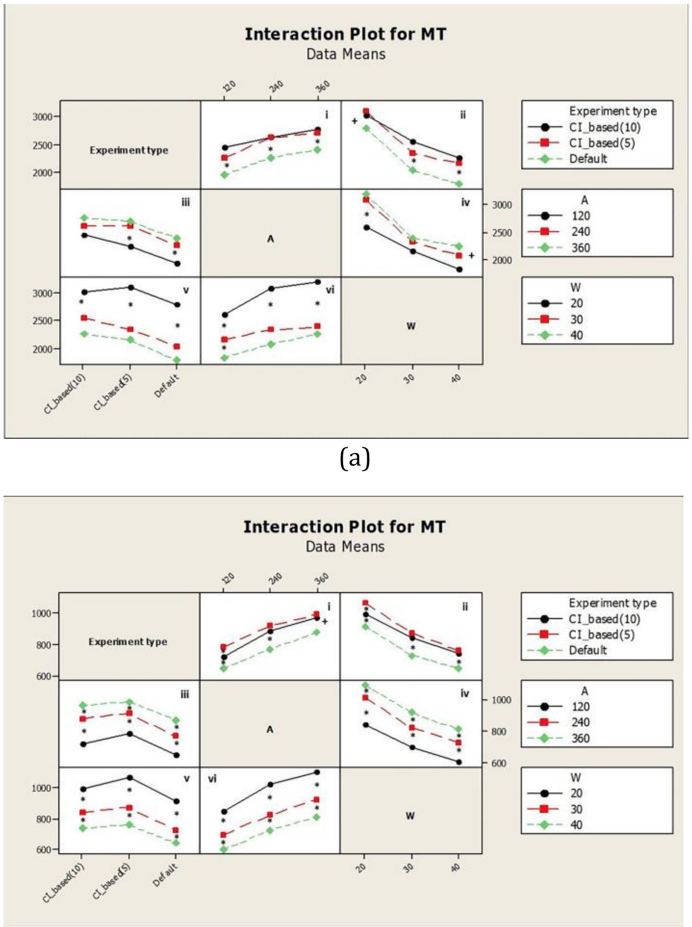
For youths with CP, the 3-way interaction effect of ET*A*W was not significant. The 2-way A*W was significant for AS ($F_{4,2410} =6.3$, $P < 0.001$), SSD ($F_{4,2410} =2.3$, $P < 0.05$) and OSR ($F_{4,2410} =2.6$, $P < 0.04$). The 2-way ET*W was significant for MS ($F_{4,2410} =4.8$, $P < 0.001$), SSD ($F_{4,2410} =2.8$, $P < 0.03$), ER ($F_{4,2410} =4.9$, $P < 0.001$). The 2-way ET*A was only significant for MS ($F_{4,2410} =2.5$, $P < 0.04$). All main effects were significant for ET, A, and W levels respectively, except OSR for A level and MS & PST for W level.

5.2. Movement Time (MT)

Average MTs for each experiment type for both youths with CP and TD youths are shown in Table 2 and Figure 2.

Youths with CP and TD youths displayed a significant increase in average MT with the proposed algorithm over the Windows default regardless of different target widths and movement amplitudes. There is no significant difference between performances of CI_based(5) and CI_based(10).

The MT increased by 12.4% (TD) and 18.5% (youths with CP) using CI_based(10) and 17.7% (TD) and 14.8% (youths with CP) using CI_based(5).



(a)

(b)

Figure 2. Interaction plot of MT by experiment type, amplitude, and width levels; youths with CP top), TD youths (bottom).* indicates significant difference between the two closest points vertically while + while indicates significant difference between the two furthest points vertically.

5.3. Average Speed (AS) and Maximum Speed (MS)

Means of AS for each experiment for both youths with CP and TD youths are shown in Table 3.

Table 2. Average MT.

Experiment Type	MT (ms) Average (SE)	
	CP	TD
Default	2205(41)	762(8)
CI-based(10)	2613(56)	856(9)
CI-based(5)	2531(48)	897(9)

Table 3. AS and MS.

Experiment Type	AS(pixels/ms) Average (SE)		MS(pixels/ms) Average (SE)	
	CP	TD	CP	TD
Default	0.16 (0.002)	0.35 (0.0034)	1.1 (0.018)	1.6 (0.018)
CI_based (10)	0.18 (0.003)	0.35 (0.0035)	1.47 (0.033)	1.81 (0.024)
CI_based (5)	0.18 (0.003)	0.37 (0.0036)	1.59 (0.031)	2.01 (0.026)

Youths with CP travel with an AS that is significantly faster with the proposed algorithm (0.18 pixels/ms) than with the Windows default (0.16 pixels/ms) regardless of different target widths and movement amplitudes (the proposed algorithm increased AS by a rate of 12.5%). The interaction between amplitude and width ($P < 0.05$) has significant effects on AS, resulting in a range from a maximum AS of 0.24 pixels/ms at a 360 pixels amplitude and a 40 pixels width to a minimum AS of 0.11 pixels/ms at a 120 pixels amplitude and a 20 pixels width.

TD youths travel slightly faster with the CI_based(5) (0.37 pixels/ms) than Windows default (0.35 pixels/ms) and CI_based(10) (0.35 pixels/ms, AS increased by a rate of 5.7%). The interaction between amplitude and either experiment type or width ($P < 0.05$) has a significant effect on AS.

Youths with CP have a significantly faster speed with the proposed algorithm over the Windows default (1.1 pixels/ms) regardless of different target widths and movement amplitudes. There is a significant difference between performances of window size (5 samples vs. 10 samples). The average value of MS traveled by TD youths at the Windows default mode is equal to the average MS traveled by youths with CP using the CI-based algorithm.

For youths with CP, there is a significant interaction effect between the experiment type and amplitude with respect to MS. CI_based results in a significantly higher MS at 240- and 360-pixel amplitudes than Windows default. The MS is significantly higher as the amplitude level increases within each experiment type.

The interaction of experiment type and width shows that the CI-based algorithm produces higher MS than Windows default across different widths.

For TD youths, MS is affected by the interaction of experiment type, amplitude, and width. The CI-based algorithm (both window sizes) was significantly faster than Windows default at each amplitude level and each width level, respectively.

5.4. Primary Sub-Movement Time (PST) and Secondary Sub-Movement Time (SST)

The average time of primary sub-movement and secondary sub-movements are shown in Table 4.

The proposed algorithm significantly assisted youths with CP and TD youths in traveling faster toward the target before the first pause (less PST) regardless of different target widths and movement amplitudes. There is no significant difference between performance of CI_based(5) and CI_based(10). In addition, the width has no significant effect on PST.

For TD youths, the PST using Windows default is significantly higher than the CI-based algorithm at each amplitude level or width level, respectively (significant interaction between experiment type and either amplitude or width).

Youths with CP require significantly more time with additional sub-movements to reach the target using the proposed algorithm compared with Windows default regardless of different target widths and movement amplitudes. There is no significant difference between the performance of CI_based(5) and CI_based(10). The traveled SST by the Windows default across different amplitudes and widths is generally less than that of the proposed CI-based algorithm, but this is not significant.

The primary sub-movement performance has been enhanced by the CI-based algorithm, but the performance of secondary sub-movements negates the effect when compared with the Windows default system. It decreased the PST by 10.8% (youths with CP) and 12.4% (TD) using the CI-based algorithm with CI_based(10), and 14.7% (youths with CP) and 15.2% (TD) using CI_based(5). The SST increased by 27.9% (youths with CP) and 36.7% (TD) using CI_based(10), while with CI_based(5), the SST increased by 22.3% (youths with CP) and 50.1% (TD).

Table 4. PST and SST.

Experiment Type	PST (ms) Average (SE)		SST (ms) Average (SE)	
	CP	TD	CP	TD
Default	526 (11)	378 (5)	1666 (41)	384 (8)
CI_based(10)	469 (13)	331 (4)	2151 (55)	525 (9)
CI_based(5)	448 (9)	320 (4)	2082 (47)	577 (9)

5.5. Primary Sub-movement Distance (PSD) and Secondary Sub-movement Distance (SSD)

Youths with CP travel significantly further in the primary sub-movement to reach the target using the proposed algorithm with CI_based(5) compared with Windows default and CI_based(10) regardless of different target widths and movement amplitudes.

TD youths travel further in the primary sub-movement to reach the target using CI_based(5) than when using CI_based(10) and Windows default. The significant interaction between experiment type and width on the primary sub-movement indicates the significant difference of PSD between the CI-based algorithm with the 5-sample window size and the other two types at each width level.

Youths with CP travel further to reach the target using the proposed algorithm compared with Windows default regardless of different target widths and movement amplitudes.

For youths with CP and TD youths, the interaction between width and either experiment type or amplitude showed that SSD by the Windows default across different widths is generally less than by the proposed CI-based algorithm.

5.6. Erroneous Clicks and Error Rate (ER)

The average ER for both TD youths and youths with CP are shown in Table 5.

TD youths score the least ER using Windows default (6%), whereas their performance became worse using the CI-based algorithm (16.13% for windows size of 10 samples and 18.36% for windows size of 5 samples). For youths with CP, the ER is lower using Windows default (9.4%) and higher using the CI-based algorithm (13.21% and 10.38%). Within each trial, erroneous clicks occur up to 6 times with youths with CP and three times for those that are TD.

For youths with CP, the interaction between experiment type and width has a significant effect on ER. The Windows default and CI_based(5) sample window size produce significantly lower ERs at widths of 30 pixels and 40 pixels in comparison with a width of 20 pixels.

Table 5. ER.

Experiment Type	ER (%) Average (SE)	
	CP	TD
Default	9.4 (0.9)	6 (0.6)
CI_based(10)	13.2 (1.3)	16.1 (1)
CI_based(5)	10.4 (1)	18.4 (1)

5.7. Number of Sub-movements (NS) and Overshoot Rate (OSR)

The average NS and OSR for both TD youths and youths with CP are shown in Table 6.

For youths with CP, the NS (including primary sub-movement) is significantly increased by using CI_based(10) and CI_based(5) compared with Windows default. In addition, for TD youths, the interaction between experiment type and width had a significant effect on NS. The Windows default produces significantly lower sub-movement correction than the CI-based algorithm at each width level; the NS at the 20-pixel width for each experiment level is significantly higher than at the 30- and 40-pixel widths. The CI-based algorithm with the 5-sample window size at the 20-pixel width level produces the highest NS corrections.

Youths with CP tend to overshoot during point-and-click computer tasks more often than TD youths. The OSR of youths with CP by the Windows default mode is 40% compared with 15% for TD youths. In addition, the OSR rose considerably for both, but for youths with CP, OSR rose at a lower rate.

For TD youths, there is significant interaction between experiment type and amplitude, which affects the OSR. The interpretation is that the impact of the experiment type depends on the amplitude and vice versa. Across different amplitudes, the OSR for each experiment type is significantly different. For example, the results for an amplitude of 120 pixels are 17% for the Windows default, 41% for the CI-based algorithm with 10-sample window size, and 66% for the CI-based algorithm with 5-sample window size. There is no significant difference between OSRs across amplitude levels within each experiment type (P = 1.06). A similar scenario was found for the interaction effect on the overshoot between experiment type and width.

Table 6. NS and OSR.

Experiment Type	NS Average (SE)		OSR (%) Average (SE)	
	CP	TD	CP	TD
Default	8 (0.2)	3 (0.06)	40 (1.7)	15 (1)
CI_based(10)	12 (0.3)	4 (0.06)	62 (1.8)	50 (1.4)
CI_based(5)	11 (0.2)	4 (0.07)	65 (1.6)	65 (1.3)

6. Discussion

In the performance evaluation of the CI-based algorithm by conventional measures, both MT and ER show the inefficiency of the proposed algorithm overall compared with the Windows default system (higher MT and ER produced by the proposed algorithm). However, the CI-based algorithm exhibited benefits in terms of AS, MS, and the primary sub-movement phase.

For both TD youths and youths with CP, the AS per point-and-click trial and the average MS increased by using the CI-based algorithm compared with Windows default.

The algorithm did not improve the pointing task for youths with CP; the ER and maximum number of erroneous clicks within each trial are slightly higher than Windows default. However, the algorithm enhanced the performance of the primary sub-movement (less time and nearly equal distance compared with Windows default). It had the opposite effect on secondary sub-movements as it made the user lose control when the cursor speeds reached higher levels, resulting in overshoot movements, requiring more sub-movements to correct the movement toward the target.

The significant interaction between experiment type and amplitude affected OSR for TD youths (higher OSR by the CI-based algorithm) and the insignificant difference of overshoot across amplitude levels for TD youths and youths with CP suggest that the CI-based algorithm deteriorates the cursor control, especially for TD youths, who have more control and stability of movement than youths with CP. This can be viewed as the rapid increase of sub-movement corrections and traveled SSD by the CI-based algorithm compared with Windows default.

The PointAssist study on elderly whose motor control lacks sophistication showed no significant advantages on MT, NS, and number of slip-offs (target reentry). Our results are similar to PointAssist, although PointAssist aimed to enhance accuracy, and we aimed to enhance speed according to our findings in an earlier study [13].

The DMS algorithm improved the performance (MT) at greater amplitudes, but our algorithm did not indicating the need to improve the performance of secondary sub-movements.

The window size had a variable influence on path measures, although the 5-sample window size produced faster modification in C-D gain than the 10-sample window size. For youths with CP, the 5-sample window size allows the user to quickly reach MS and the target compared with the 10-sample window size. This can be seen by less PST and SST produced by the 5-sample windows size. This is because the 10-sample window size takes longer to modify, especially during the sub-movement corrections, whereas the user needs robust control of the cursor. The 5-sample window size provides better accuracy pointing than the 10-sample window size.

The 20-pixel width required more sub-movement correction and traveled SSD, suggesting that for designing small icons, it requires a pointing accuracy algorithm to be incorporated during sub-movement corrections similar to the PointAssist. PointAssist found that the algorithm had a significant effect on 16-pixel width for children [11].

7. Study Limitation

The study is limited by a lack of training. People are comfortable using Windows default settings. In addition, the number of participants with CP is not large due to the difficulty of finding youths who fit our experiment criteria.

8. Conclusion

The overall performance of the CI-based algorithm, for both TD and youths with CP, is less than the Windows default system, but there is a possibility to enhance this performance by decreasing the SST. The primary sub-movement is faster with the algorithm than with the Windows default mode.

9. Future Work

Future work will include performance enhancements in the secondary sub-movements phase, such as clamping extreme speeds to provide the youths with more control with the cursor during secondary sub-movements, or triggering a control mode once the NS goes over the threshold NS, to provide more robust control during sub-movement corrections. Clamping extreme speeds and control mode triggering can be implemented by dynamic adjustments of C-D gain. Another recommendation is a switch from the CI-based algorithm mode to Windows default mode once the cursor speed reaches an upper threshold speed and a switch from Windows default to the CI-based algorithm mode once the speed crosses a lower threshold speed.

References

[1] Yeargin-Allsopp M., Braun K. V. N., Doernberg N. S., Benedict R. E., Kirby R. S., and Durkin M. S., “Prevalence of Cerebral Palsy in 8-year-old Children in Three Areas of the United States in 2002: A Multisite Collaboration,” *Pediatrics*, vol. 121, no. 3, pp. 547-554, Mar. 2008.

[2] Access Economics Pty Limited, “The Economic Impact of Cerebral Palsy in Australia in 2007,” Apr. 2008.

[3] Benyon D., Crerar A., and Wilkinson S., “Individual Differences and Inclusive Design,” in *User Interfaces for All: Concepts, Methods, and Tools*, C. Stephanidis, Ed. London, UK: CRC Taylor and Francis, 2000, pp. 21-46.

[4] Davies T.C., Mudge S., Ameratunga S. and Stott N.S., “Enabling Self-Directed Computer Use for Individuals with Cerebral Palsy: A Systematic Review of Assistive Devices and Technologies,” *Develop. Medicine & Child Neurology*, vol. 52, no. 6, pp. 510-516, Jun. 2010.

[5] Copley J. and Ziviani J., “Barriers to the Use of Assistive Technology for Children with Multiple Disabilities,” *Occup Ther Int*, vol. 11, no. 4, pp. 229-243, Nov. 2004.

[6] Ellis J.B. “The Digital Divide in Special Education,” in *Int. Study Assoc. on Teachers and Teaching Conf.*, St. Catharines, Ontario, 2007.

[7] Davies T.C., Chau T., Fehlings D., Ameratunga S., and Stott N.S., “Youth with Cerebral Palsy with Differing Upper Limb Abilities: How Do They Access Computers?,” *Archives of Physical Medicine and Rehabilitation*, vol. 91, no. 12, pp. 1952-1956, Dec. 2010.

[8] Michaels C. A., Prezant F. P., Morabito S. M., and Jackson K., “Assistive and Instructional Technology for Students with Disabilities: A National Snapshot of Postsecondary Service Providers,” *J. of Special Edu. Technol.*, vol. 17, no. 1, pp. 5-14, 2002.

[9] Wobbrock J.O., Fogarty J., Liu S., Kimuro S., and Harada S., “The Angle Mouse: Target-Agnostic Dynamic Gain Adjustment Based on Angular Deviation,” in *27th Int. Conf. on Human Factors in Comput. Syst.*, 2009 ACM: Boston, MA, USA.

[10] Hourcade J.P., Nguyen C.M., Perry K.B., and Denburg N.L, “Point assist for Older Adults: Analyzing Sub-Movement Characteristics to Aid in Pointing Tasks,” in *28th Int. Conf. on Human Factors in Comput. Syst.*, 2010 ACM: Atlanta, Georgia, USA, pp. 1115-1124.

[11] Hourcade J.P., Perry K.B., and Sharma A., “Point Assist: Helping Four Year Olds Point with Ease,” in *7th Int. Conf. on Interaction Design and Children*, 2008, ACM: Chicago, Illinois, pp. 202-209.

[12] Tang K.-H. and Lee Y-H., “Dynamic Mouse Speed Scheme Design Based on Trajectory Analysis,” in *Int. Conf. on Ergonomics and Health Aspects of Work with Comput.*, 2007, Springer-Verlag: Beijing, China, pp. 329-338.

[13] Almanji A., Davies T.C., and Stott N.S, “Using Cursor Measures to Investigate the Effects of Impairment Severity on Cursor Control for Youths with Cerebral Palsy,” *Int. J. of Human-Comput. Stud.*, vol. 72, no. 3, pp. 349-357, Mar. 2014.

[14] Almanji A., Davies T.C., Payne A. and Amor R., “A Nonlinear Model for Mouse Pointing Task Movement Time Analysis Based on Both System and Human Effects, *IEEE Trans. on Neural Syst. and Rehabilitation Eng.*, pp. 1-10, Dec. 2014.

[15] Keates S., Hwang F., Langdon P., Clarkson P.J., and Robinson P., “Cursor Measures for Motion-Impaired Computer Users,” in *Fifth Int. ACM Conf. on Assistive Technol.*, 2002, ACM: Edinburgh, Scotland, pp. 135-142.

[16] MacKenzie I.S., Tatu K., and Miika S., “Accuracy Measures for Evaluating Computer Pointing Devices,” in *SIGCHI Conf. on Human Factors in Comput. Syst.*, 2001, ACM: Seattle, Washington, United States.

[17] Balakrishnan R., “"Beating" Fitts' Law: Virtual Enhancements for Pointing Facilitation,” *Int. J. of Human-Comput. Stud.*, vol 61, no. 6, pp. 857-874, Dec. 2004.

[18] Hwang F., Keates S., Langdon P., and Clarkson J., “Mouse Movements of Motion-Impaired Users: A Sub-Movement Analysis,” *ACM SIGACCESS Accessibility and Comput.,* no. 77-78, pp. 102-109, Sep. 2003.

[19] Hwang F., Keates S.,Langdon P., and Clarkson, J., “A Sub-Movement Analysis of Cursor Trajectories,” *Behaviour & Inform. Technol.*, vol. 24, no. 3, pp. 205-217, Feb. 2007.

[20] Chapuis O., Blanch R. and Beaudouin-Lafon M., “Fitts' Law in the Wild: A Field Study of Aimed Movements,” 2007.