Adaptive Click-and-Cross: Adapting to Both Abilities and Task Improves Performance of Users With Impaired Dexterity

Louis Li
Harvard School of Engineering and Applied Sciences
33 Oxford St., Cambridge, MA, USA
louis@seas.harvard.edu

Krzysztof Z. Gajos
Harvard School of Engineering and Applied Sciences
33 Oxford St., Cambridge, MA, USA
kgajos@eecs.harvard.edu

ABSTRACT
Computer users with impaired dexterity often have difficulty accessing small, densely packed user interface elements. Past research in software-based solutions has mainly employed two approaches: modifying the interface and modifying the interaction with the cursor. Each approach, however, has limitations. Modifying the user interface by enlarging interactive elements makes access efficient for simple interfaces but increases the cost of navigation for complex ones by displacing items to screens that require tabs or scrolling to reach. Modifying the interaction with the cursor makes access possible to unmodified interfaces but may perform poorly on densely packed targets or require the user to perform multiple steps. We developed a new approach that combines the strengths of the existing approaches while minimizing their shortcomings. We instantiated this approach as Adaptive Click-and-Cross, a novel interaction technique that introduces only minimal distortion to the original interface while making access to frequently used parts of the user interface efficient and access to all other parts possible. Our study demonstrates that, for sufficiently complex interfaces, Adaptive Click-and-Cross improves the performance of users with impaired dexterity compared to only modifying the interface or only modifying the cursor.
Keywords: Accessibility, area cursors, adaptive user interface

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

INTRODUCTION
Computer users with impaired dexterity often have difficulty with mainstream user interfaces, especially when these user interfaces contain small, densely-packed interactive elements.

In the past few decades, a variety of software-based techniques have emerged to assist such users. These approaches fall broadly into two categories: those that modify the user interface itself (e.g., ability-based user interfaces generated with SUPPLE [8]) and those that modify the user's interaction with the mouse pointer (e.g., area cursor [15], bubble cursor [11], enhanced area cursors such as Click-and-Cross [3]).

Approaches that adapt the user's abilities to the existing user interface by modifying the cursor make access possible without requiring substantial modifications to existing interfaces. However, these techniques may lack generality (e.g., area cursors and the bubble cursor enhance interaction only when clickable elements are sparsely laid out), or they may reduce the efficiency of the interaction (e.g., the Click-and-Cross technique from Findlater et al. [3] replaces a single click with two operations: a click in a general vicinity of the desired target followed by a crossing action to make a specific selection).

In contrast, approaches that adapt the user interface to the user's abilities by modifying the user interface enable efficient access to each item, optimizing the interaction to each user's strengths [8, 22]. However, adapting user interfaces to the abilities of users with impaired dexterity involves an important trade off: such adaptations typically involve making clickable elements larger at the cost of increased navigational complexity. This requires more scrolling and tab switching when navigating between user interface elements. Existing approaches often enlarge all clickable elements — even those that users rarely access — because not enlarging them might render them inaccessible. The increased navigational complexity from such a broad approach is a source of inefficiency.

We set out to combine the strengths of the two approaches: making access possible and efficient while minimizing modifications of the original design. To do this, we build on a third adaptive approach: user interfaces that adapt themselves to the user's *task* (e.g., [2, 6, 9, 19]). Such interfaces have been demonstrated to improve users' performance by leveraging predictive models for each user's actions to ease access to the features that the user is most likely to access next (e.g., by copying them to a more easily accessible location, by making them larger or more visually salient).

Building on these three ideas of adaptation, we have developed Adaptive Click-and-Cross. As illustrated in Figure 1, with Adaptive Click-and-Cross, user interface elements that are predicted to be most frequently accessed by the user are enlarged and can be accessed efficiently with a single click

Submitted for review to ACM IUI'14. Please do not redistribute.

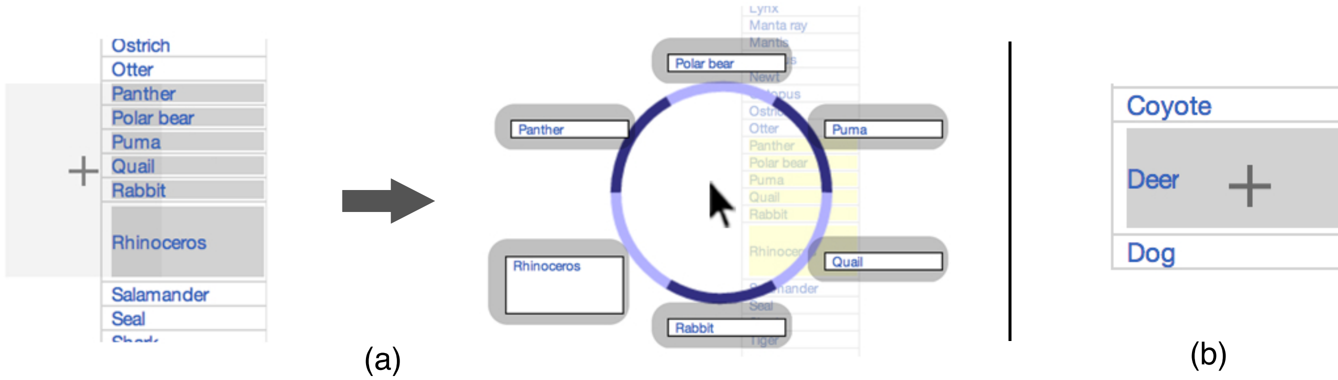

Figure 1: Adaptive Click-and-Cross. **(a)** When users click near or directly on small targets, Click-and-Cross is triggered. Users can then cross through the arc corresponding to an item to select the item. **(b)** Users can directly click on a large target to select it.

(adapting the interface to the user, adapting the interface to the task). The remaining elements are left unmodified and can be accessed through the Click-and-Cross technique. The user can click anywhere in the vicinity of the desired target and subsequently refine the selection with a crossing interaction (adapting the user's abilities to the interface). This approach achieves three things: it enables *efficient* access to frequently accessed user interface elements, makes access to all other elements *possible*, and minimizes the distortion of modifying the user interface.

The results of our study with 12 participants with impaired dexterity demonstrate that for a complex user interface (one where enlarging all interface elements substantially increases the cost of navigating the interface), Adaptive Click-and-Cross results in significantly shorter task completion times compared to either adapting the user interface by enlarging all elements or Click-and-Cross alone. We observed no significant differences in error rates or subjective preference across the three techniques. However, participants subjectively perceived the interface with all elements enlarged as more efficient than either Click-and-Cross or Adaptive Click-and-Cross.

RELATED WORK

Many existing software solutions improve accessibility by modifying users' methods of interaction. Such solutions may adapt the behavior of a pointing cursor to the user (e.g., Steady Clicks [20], Angle Mouse [23]), or they may introduce entirely new interaction techniques (e.g., area cursor [15], bubble cursor [11], enhanced area cursors [3]).

Approaches that directly modify the user interface have also been investigated. These approaches advocate adapting the user interface to users' needs (e.g., EyeDraw [13] and VoiceDraw [12]). Although creating accessible designs that are well suited to a particular set of abilities can be time consuming, previous work has begun to demonstrate how such modifications could be automated [8].

However, for complex user interfaces, adapting the user interface to the abilities of users with impaired dexterity requires either reducing the available functionality to fit all elements on the screen [13, 12] or increasing navigational complexity by requiring more scrolling, switching between tab panes, etc. [8]. Recent work has examined the efficacy of on-demand expansion of targets (i.e., dynamically expanding the target after the user begins to move the cursor in its direction) [14]. This approach prevents targets from being enlarged unnecessarily, minimizing the potential increase in navigational complexity from enlarging targets. While the experimental results show that the approach improves performance, its effectiveness is likely to diminish in densely packed user interfaces.

Adaptive Click-and-Cross aims to minimize the costs of modifying the user interface by leveraging an important finding of previous work: most users only access a small subset of the available functionality, though each user accesses a *different* subset [10, 17]. This finding has been used to design user interfaces that enable efficient access to a subset of items that are predicted to be of most use to the user. For example, in split interfaces [5, 6, 19], the elements predicted to be most useful are duplicated in a convenient location to support more immediate access. In contrast, morphing menus [1, 21], which have been tested with able-bodied users, do not duplicate predicted elements but instead enlarge predicted items to enable efficient access.

ADAPTIVE CLICK-AND-CROSS

We designed Adaptive Click-and-Cross to support users with impaired dexterity. We aimed for it to yield most of the efficiency benefits of approaches that adapt the user interface to the abilities of the user, while minimizing the amount of distortion to the original interface that such adaptations typically cause.

In Adaptive Click-and-Cross, a small number of the user interface elements — those predicted to be of immediate use to the user — are enlarged to enable efficient access (Figure 1). For the remaining elements, which may be too small for a user with impaired dexterity to access reliably, Adaptive Click-and-Cross employs the Click-and-Cross technique [3]. The Click-and-Cross technique is activated when the user clicks near or directly on one of the remaining items, activating a circular overlay. Pictures of up to six nearby items are arranged around the circle, and each one is associated with an

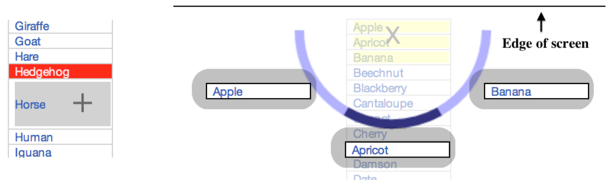

Figure 2: (a) In Adaptive Click-and-Cross, when target is bordered by an enlarged item, the target has a decreased amount of space for activating Click-and-Cross. **(b)** Near the edge of the screen, the Click-and-Cross cursor only displays a subset of the circle.

arc on the circle. Adjacent arcs alternate in color, making the lengths of the arcs easily distinguishable. The user can make a selection by moving the pointer across the corresponding arc. If the first click was made by mistake, performing another click inside the circle cancels the interaction.

The appearance of the cursor changes depending on the position of the pointer (Figure 1). By default, the cursor is an area cursor: a translucent, gray rectangle with a crosshair in the center. When placed directly over an enlarged item, the gray rectangle disappears, but the underlying item is highlighted in gray.

When the cursor is not over an enlarged element, the cursor will resize in order to surround those targets that are projected to appear if Click-and-Cross is activated. For useful visual feedback, these targets will also be highlighted in gray. Upon activation of the Click-and-Cross cursor, the gray rectangle disappears, revealing a traditional point cursor that the user can move to make the crossing selection.

In situations where only a part of the circle can be rendered on the screen, such as when the initial click occurs near the edge of the screen (Figure 2b), the area cursor is appropriately resized to provide accurate visual feedback as to which items can be accessed.

When activated, the Click-and-Cross cursor also includes any nearby enlarged items in the overlay, meaning that enlarged items can be acquired either through Click-and-Cross or through a direct click.

Arc Placement Algorithm

The cursors employing Click-and-Cross are aimed to place the targets around the crossing arcs in an intuitive manner. In the case of a linearly arranged menu interface, it is not possible to assign arcs to each nearby item based solely on angle, since a click in the middle of the menu would have items at approximately the same angle above and below the clicked point.

Our algorithm for matching items and crossing arcs differed slightly from that used in the original design [3], in part to handle the activation of the cursor on the edge of the screen.

Our algorithm first computes the positions of circle segments that will be visible. For example, if the cursor is near the top of the screen, only the three arcs in the bottom half of the circle, forming a semicircle, will have enough space to appear (Figure 2b).

Next, our algorithm assigns each visible arc to a nearby target. The arc corresponding to the angle from the clicked point to the single nearest target is calculated and proposed as the best arc index for the closest target. If this arc would not be on the screen, due to the click being near the edge of the screen, then the next arc with the closest angle from the activation point is taken.

Finally, the other targets are sorted by ascending distance, and each one is assigned the next available arc segment going counter-clockwise.

EXPERIMENT

Adaptive Click-and-Cross uses a model of a user's task to combine two adaptive approaches: Click-and-Cross and enlarging user interface elements. To evaluate this fusion of adaptive approaches, we conducted a user study with 12 participants with dexterity impairments, using the results to empirically compare how Adaptive Click-and-Cross affected efficiency and accuracy of pointing in comparison to just Click-and-Cross or just a design where all elements were enlarged. For completeness, we also included a non-adaptive baseline condition.

Participants

Twelve people (six male, six female) with dexterity impairments of varying severity participated in the study. Participants were between the ages of 19 and 65. Table 1 provides more detailed information about each of our participants.

Two people participated in person and 10 remotely. Remote participants volunteered by email in response to online advertisements. We communicated with them by email to obtain further understanding of their impairment and to explain how to participate in the study. Participants were offered compensation in the form of gift certificates. Remote participants performed the experiment independently at times of their convenience.

In addition to the twelve participants whose data were included in the analysis, three participants' data were discarded. The first did not report any functional limitations or motor impairments in the hands, although he had a spinal cord injury that paralyzed his legs. Another was due to a failure to properly adjust the mouse gain to a level comfortable for the participant before proceeding with the experiment, yielding unrepresentative results. The last was due to a technical error in the experiment that corrupted the data.

Recent work has provided compelling evidence showing that performance evaluations of user interfaces can be performed reliably with remote participants [16], provided that a few basic safeguards (such as testing for instruction comprehension, selecting appropriate outlier removal criteria) are maintained. We have built on those insights to ensure reliability of the results collected from our remote participants.

Apparatus

The experiment was implemented as a web site written in HTML, CSS, and JavaScript.

No.	Method	Age	Device	Gender	Condition	Fa	Co	St	Mo	Gr	Ho	Tr	Sp	Se	Dir	Dist
1	Remote	59	Mouse	M	Essential tremor	×			×		×	×				
2	Remote	23	Touchpad	M	C-6 quadraplegic				×				×	×		
3	Remote	49	Mouse	F	Spinal stenosis, ruptured cervical disks			×		×			×			×
4	Remote	62	Mouse	F	Multiple sclerosis	×	×	×	×					×		
5	Remote	38	Mouse	F	Ankylosing spondylitis and fibromyalgia			×		×	×				×	
6	Remote	42	Mouse	F	Duchene muscular dystrophy	×		×								×
7	Remote	65	Trackball	M	Spinal cord injury							×		×	×	×
8	Remote	38	Mouse	M	Spinal cord injury	×		×		×	×		×		×	×
9	Remote	43	Head mouse	M	Cerebral palsy		×			×	×		×		×	×
10	Remote	19	Mouse	F	Familiar essential tremor, Parkinson's							×	×			
11	In-person	49	Trackball	F	Multiple sclerosis				×							
12	In-person	59	Trackball	M	Multiple sclerosis	×	×			×	×		×		×	×

Table 1: Detailed information about the participants and their self-reported functional limitations. Legend: Fa=rapid fatigue, Co=poor coordination, St=low strength, Mo=slow movements, Gr=difficulty gripping, Ho=difficulty holding, Tr=tremor, Sp=spasm, Se=Lack of sensation, Dir=difficulty controlling direction, Dist=difficulty controlling distance movements, Gr=difficulty grip

Remote participants completed the study using their own computers and devices. Measures were taken to ensure consistency between participants: participants were asked to reset the zoom levels on their browsers and make their browser windows as large as possible, and the visible portion of the scrolling menu interface (i.e., the number of items displayed at a given scroll position) was held constant at 475 pixels. A summary of the input devices used by the participants can be found in Table 1.

Tasks

Building on prior empirical research on adaptive user interfaces [1, 4, 5, 21], we chose menu selection as our experimental task. This task naturally supports manipulation of navigational complexity (i.e., by changing what fraction of the menu is visible in the application window, we could control how much scrolling was required on average to reach a menu item [5]).

We tested four designs in the study:

1. *Enlarged*: traditional cursor pointing with all menu items enlarged (80×40 pixels);
2. *Click-and-Cross*: the Click-and-Cross cursor, menu items are the default size (80×10 pixels);
3. *Adaptive Click-and-Cross*: a menu where some items are large and can be acquired directly through normal clicking, and some items are the default size and can be acquired using Click-and-Cross;
4. *Baseline*: traditional mouse pointing, menu items are the default size.

The order of the conditions was counterbalanced using a partial Latin square design. The tasks in each condition were isomorphic, but each condition used a different vocabulary (i.e., fruits, vegetables, animals, colors) and differed in the order of the sets of trials within each condition.

The targets that participants had to acquire during the experiment were distributed uniformly throughout the menu. In Click-and-Cross and Baseline conditions (where all items were the default size), this resulted in approximately 60% of the trials with targets on the first screen — those targets could be acquired without scrolling. In the Enlarged condition, where all menu items were enlarged, fewer items were visible on the screen at once: in only 20% of the trials the desired targets could be reached without scrolling. In the Adaptive Click-and-Cross condition, where only a small fraction of the items were enlarged while the rest were the default size, in 50% of the trials the desired targets could be accessed without scrolling.

In the Adaptive Click-and-Cross condition, we simulated a system with a 70% accuracy in predicting what menu items the user would use. Similarly to others [2, 5, 6, 7], we did so by designing the experimental task such that 70% of the items that the participants were asked to select in that condition were enlarged, while the remaining 30% were not.

By default, menu items were 80 pixels wide and 10 pixels tall. Enlarged menu items were 80 pixels wide and 40 pixels tall. In each design, the menu interface consisted of 60 items. The height of the application window was held constant at 475 pixels for all participants regardless of their screen size or browser. Thus, only a 475 pixel section of the menu was visible on the screen at any one time, and participants had to scroll to see items further down.

Procedure

Each participant first filled out a demographic survey containing questions about his or her computer usage and motor and/or visual impairments.

Participants then proceeded to the main part of the experiment. For each participant, there were 4 conditions $\times$ 5 blocks $\times$ 10 trials $=$ 200 trials. At the beginning of each condition, each participant was presented with an instructional video describing the cursor behavior and interface for the condition. The first block of each condition was a practice block, allowing the participant to become accustomed to the design. Performance on the practice blocks was not included in the analysis. Thus, the analysis for each participant were performed using 4 conditions $\times$ 4 blocks $\times$ 10 trials $=$ 160 trials. At the end of each condition, participants rated the condition on a 7-point Likert scale on how easy, tiring, or efficient they found the particular design to be. At the end of the study, each participant ranked the conditions in order of overall preference and perceived efficiency.

The study took 40 to 80 minutes depending on individual abilities.

Design and Analysis

We used a within-subjects factorial design for our analysis with Design {Enlarged, Click-and-Cross, Adaptive Click-and-Cross} as the main factor.

For the non-adaptive Baseline condition, we did not necessarily expect to see benefits with respect to the non-adaptive interface: the results of a prior evaluation of Click-and-Cross [3] suggest that the technique provides substantial performance benefits for very small targets (8 pixels or smaller) or for participants with severely impaired dexterity. In our experiment, we used larger targets (10×80 pixels) to ensure that minor visual impairments, which are common among elderly participants, would not preclude participation in our study. Additionally, most of our participants had moderate rather than severe dexterity impairments. For these reasons, we excluded the non-adaptive baseline from our analysis.

The main measures in this experiment were *target acquisition time*, computed over error-free trials, and *error rate*, computed as the fraction of the trials that contained at least one error. Subjective measures for each condition also included *perceived efficiency*, *perceived fatigue*, *perceived ease of use*, *efficiency ranking*, and *overall preference ranking*.

The following (within-subjects) factors were also included in some of the follow-up analyses:

- On First Screen {Yes, No}: We recorded whether an item was *on the first screen* — the initial part of the menu presented to the participant — meaning that the participant did not have to scroll to acquire the target.

- On Edge {Yes, No}: For Click-and-Cross and Adaptive Click-and-Cross, we recorded whether an item was *on the edge* of the visible part of the application window at the time of acquisition. Because only part of the circular overlay is shown when there is not enough space for the circle, this could potentially impact acquisition time (Figure 2b).

- Bordered by Enlarged Item {Yes, No}: For Adaptive Click-and-Cross specifically, we recorded whether the target item was *bordered by an enlarged item*. Because enlarged items could immediately be acquired by the user through a normal click, small items bordering an enlarged item had less space in which the area cursor could be triggered. For example, if an enlarged item was directly above a small item, then the space above the small item could no longer be used to trigger the Click-and-Cross cursor (Figure 2a).

- Block Number {1 through 4}: In each condition, we presented participants with four experimental blocks. We modeled Block Number as an ordinal variable to test for the presence of any prominent learning effects.

In total, data were collected for 1440 acquisition trials. Trials with acquisition times outside of two interquartile ranges from the median were discarded as outliers ($42/1440 = 2.9\%$ of the trials). To account for the wide range of individual abilities, the outlier removal procedure was performed separately for each participant. This median-based approach was selected over the standard approach of discarding trials outside of ± 2 standard deviations, as it is more robust for remote experiments, where extreme outliers may heavily impact the mean and standard deviation [16].

After discarding outliers, timing data were log-transformed to account for the skewed distribution found in such data.

Analysis of acquisition time was performed using repeated measures analysis of variance.

Because a binary measure was used to capture whether an error occurred in each trial, we used binomial logistic regression to examine the effect of condition on error rate.

The subjective results were analyzed using non-parametric Friedman tests. The findings for subjective data that were statistically significant were followed up with pairwise Wilcoxon tests with Bonferroni correction.

For the analyses specific to the effect of *On Edge* and *Bordered by Enlarged Item* on performance with Click-and-Cross and Adaptive Click-and-Cross, paired t-tests were used. Trials with errors were included in those analyses. Because these analyses were designed to investigate whether the activation of a partial circle (On Edge) or the reduced space for activation (Bordered By Enlarged Item) affected acquisition time, we focused on the scenarios where the user was likely to make an error, either from failing to include a target near the edge or attempting to activate Click-and-Cross but instead clicking an enlarged item.

RESULTS

Preliminaries

Because users were likely to be more familiar with the traditional pointing in Enlarged than with Click-and-Cross, we first conducted an analysis to test for the presence of any prominent learning effects. We used Condition and Block Number as factors and acquisition time as the dependent variable. We observed no significant effect of block on acquisition time ($F_{3,9} = 1.322, p = 0.327$). There was also no significant interaction between condition and block number ($F_{6,6} = 0.164, p = 0.978$). These results indicate that, on average, participants' performance did not vary systematically from block to block after they had completed the practice trial for each condition. Thus, all blocks were used in the subsequent analyses.

Overall acquisition times

For acquisition time, analysis excluded trials with errors (62 / $1398 = 4.4\%$ of the trials).

We observed a significant main effect of Design on acquisition time ($F_{2,10} = 1.15, p < 0.05$). Adaptive Click-and-Cross had the lowest average acquisition time of the three adaptive designs: 5.4 s for Adaptive Click-and-Cross, 5.8 s for Click-and-Cross, and 6.0 s for Enlarged. These results are illustrated in Figure 3.

Ease of acquisition vs ease of navigation

There was a significant interaction effect between Design and On First Screen ($F_{2,21} = 0.67, p < 0.005$).Enlarged was the slowest overall, but for trials where targets were on the first

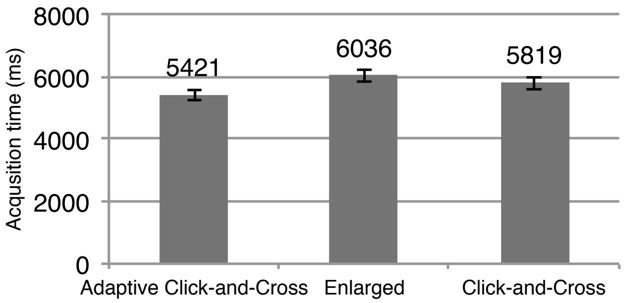

Figure 3: Mean acquisition times for each condition. Error bars represent ± 1 standard error of the mean (SEM).

screen and required no scrolling to acquire, Enlarged was the fastest with an average acquisition time of 2.6 s. Adaptive Click-and-Cross (3.8 s) was still faster than Click-and-Cross (4.6 s). In contrast, when users had to scroll in order to reach the target item, Adaptive Click-and-Cross and Enlarged had comparable acquisition times (6.6 s v 6.7 s). Click-and-Cross was again slower than the other two designs (7.0 s). These results are illustrated in Figure 4.

This supports the notion that, given large menu items that require no navigation to acquire, very large menu items are easy for users with dexterity impairments to acquire. However, despite the large speed advantage of Enlarged when items are on the first screen, this advantage is offset in the overall acquisition times by the increased amount of scrolling required in Enlarged.

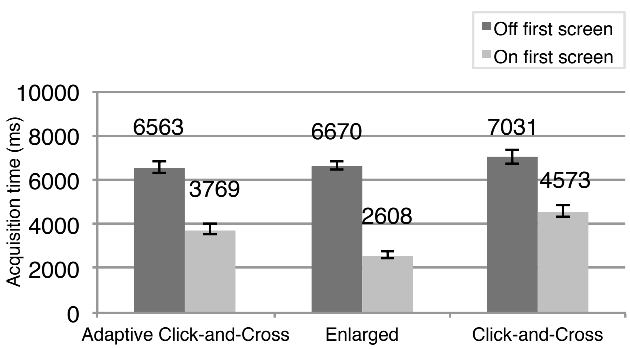

Figure 4: Mean acquisition times for each condition, grouped by whether or not the item was on the first screen presented to the user. Error bars represent ± 1 SEM.

Errors

Overall, participants were slightly more likely to make an error with Adaptive Click-and-Cross than with the other conditions, but the difference was not significant ($\chi^2_2 = 2.43, p = 0.2972$). Participants varied widely in individual performance, as reflected by the standard errors in Figure 5.

Subjective Results

After each condition, participants rated the design they had just interacted with on a 7-point Likert scale for how easy, efficient, or physically tiring they felt the particular condition to be. There was no significant effect of condition on any of these perceived traits, though the perception of efficiency was marginally significant (easy: $\chi^2_{(2,n=12)} = 3.74, p = 0.15$, efficient: $\chi^2_{(2,n=12)} = 5.87, p = 0.053$, tiring:$\chi^2_{(2,n=12)} = 2.72, p = 0.26$) with participants perceiving the Enlarged condition as being more efficient than the other two.

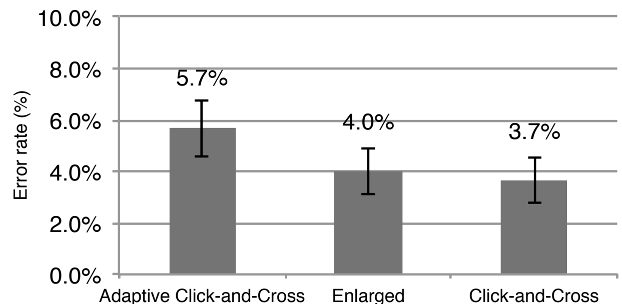

Figure 5: Error rates for each condition. Error bars represent ± 1 SEM.

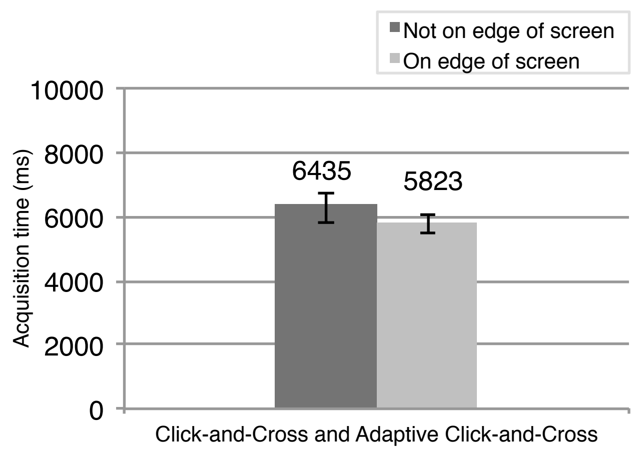

Figure 6: Mean acquisition times for Adaptive Click-and-Cross (trials where Click-and-Cross was used) and Click-and-Cross together, grouped by whether the target was near the edge of the screen at the time of acquisition. Error bars represent ± 1 SEM.

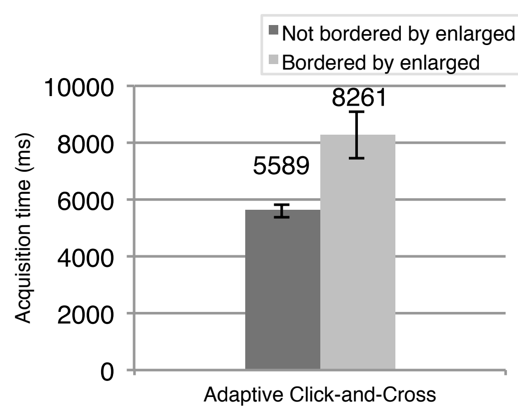

Figure 7: Mean acquisition times for Adaptive Click-and-Cross with trials grouped by whether the target item was bordered by an enlarged item. Error bars represent ± 1 SEM.

At the end of the study, participants ranked the designs in order of overall preference and perceived efficiency. There was a significant main effect of condition on subjective efficiency rankings ($\chi^2_{(2,N=12)} = 8.17, p < 0.05$). Pairwise comparisons showed that participants perceived Enlarged to be more efficient than either Click-and-Cross or Adaptive Click-and-Cross, but there was no significant difference in efficiency rankings between Click-and-Cross and Adaptive Click-and-Cross. There was no significant effect of condition on the overall preference rankings ($\chi^2_{(2,N=12)} = 4.67, p = 0.097$).

The perceived efficiency of Enlarged agreed with many of the comments from participants, who cited the enlarged elements in both Adaptive Click-and-Cross and Enlarged as favorable. One participant stated, "I liked the combined method the most, but with the large boxes ... I found the larger boxes easier to focus on and scroll over." Another stated, "Clicking bigger targets is easier and more efficient."

Some participants mentioned the cost of scrolling in the Enlarged interface. One participant said, "[The] target was larger, but [...] lots and lots of scrolling [was] needed."

Additional Analyses

We performed two additional analyses to investigate how design choices specific to Click-and-Cross and Adaptive Click-and-Cross impacted participants' performance.

Performance for targets located near the edge of the screen.
In both Click-and-Cross and Adaptive Click-and-Cross, if a user clicks on a user interface element located at the edge of the screen, only a fraction of the circular overlay can be shown on the screen. For example, as illustrated in Figure 2b, clicking near the top of the visible part of the screen will only bring up a semicircle with the three nearest targets included rather than a full circle with six targets. Anywhere from three to six targets may appear depending on the distance of the click from the edge of the screen.

We conducted an additional analysis over trials from the Click-and-Cross and Adaptive Click-and-Cross condition with On Edge as the within subjects factor. In Adaptive Click-and-Cross condition, only those trials where the Click-and-Cross technique was used to acquire the target were included in this analysis. Because we expected both Click-and-Cross and Adaptive Click-and-Cross to be affected in the same way by targets on the edge of the screen, these trials were analyzed together.

We observed a marginally significant main effect of On Edge on acquisition time ($t_{23} = 1.58, p = 0.06$) in the Click-and-Cross and Adaptive Click-and-Cross conditions. Acquisition times for items near the edge were slightly shorter (6.4 s vs 5.8 s). The findings are illustrated in Figure 6.

In the initial designs of Adaptive Click-and-Cross [18], it was found that users were sometimes surprised by this behavior. This was ultimately resolved by providing salient visual feedback in the form of highlighting to indicate the items that would be displayed (in addition to the visual feedback provided by the area cursor) before activation of Click-and-Cross. This allowed users to preview the items that would appear in the circular overlay.

Performance for targets that are bordered by enlarged items
In Adaptive Click-and-Cross, some small menu items were bordered by enlarged items. These items had a decreased amount of space in which the Click-and-Cross cursor could be activated. This scenario is illustrated in Figure 2(a).

We analyzed our data to investigate how this aspect of the design might impact performance. We observed a significant main effect of whether an item was bordered by an enlarged item on acquisition time ($t_{11} = 6.33, p < 0.0001$). In examining trials where the target item was small (i.e., not predicted to be useful) but bordered by an enlarged target on either the top or bottom, acquisition times were substantially longer than for targets that were bordered only by regularly sized neighbors (8.3 s vs 5.6 s). These findings are illustrated in Figure 7.

DISCUSSION

This study explored one point in the design space of interaction techniques that combine multiple adaptations: our Adaptive Click-and-Cross technique, which combines adaptation to user's motor abilities with adaptation to a user's task, was designed to explore this concept in the context of improving the performance of users with dexterity impairments.

In our study, Adaptive Click-and-Cross was shown to result in significantly faster performance than either Enlarged or Click-and-Cross. There were no significant differences in accuracy across the three conditions. There were also no significant differences in subjective preferences across the three designs, though participants perceived the Enlarged design to be subjectively more efficient than either Adaptive Click-and-Cross or Click-and-Cross. However, participants' comments during interviews indicated that they were aware of the trade-off of enlarging all interactive elements: they commented on the ease of clicking on enlarged targets, but they also noted the increased effort required to scroll to the desired target.

Our study also allowed us to explore several practical considerations relevant to any real deployments of either Click-and-Cross or Adaptive Click-and-Cross. First, we investigated the performance of Click-and-Cross and Adaptive Click-and-Cross when used to access items near the edge of the window, where there is not enough space to display the full overlay for the subsequent crossing interaction. Our results show that performance on such targets is actually marginally faster than for targets placed in the middle of the screen where the entire circular overlay can be displayed. This effect was not observed in an earlier variant of the Adaptive Click-and-Cross design [18], which lacked the prominent visual feedback showing what user interface elements would be available for selection in the crossing step. Such prominent visual feedback is present in the current design of Adaptive Click-and-Cross (Figure 2b).

Second, for Adaptive Click-and-Cross, our results show that acquisition time was negatively affected for non-enlarged targets that were bordered by an enlarged item. Because the enlarged item can be acquired through a direct click, the presence of the enlarged item reduces the available space for activating the Click-and-Cross interaction to acquire the neighboring non-enlarged item. This suggests a second design consideration for Adaptive Click-and-Cross: enlarging a larger number of items both increases the amount of scrolling required to navigate the interface *and* makes some of the non-enlarged items harder to access than they would have been with the Click-and-Cross technique alone. An important implication of this is that two enlarged items should be intelligently placed such that there is enough space to acquire the non-enlarged items in between.

One limitation of our study was that most of the participants we recruited had only moderate levels of impairment. For that reason, we were not able to demonstrate the benefit of Adaptive Click-and-Cross over non-adaptive interfaces. However, we were able to meaningfully demonstrate that Adaptive Click-and-Cross improves participants' performance in comparison to two existing approaches: adapting the size of elements to users' motor abilities and Click-and-Cross, both of which had been previously shown to benefit users with severely impaired dexterity [3, 8].

While this study evaluated a single technique, Adaptive Click-and-Cross, varying further parameters can provide insight into the different factors that affect such techniques, such as the choice of enhanced area cursor, different target sizes, predictive accuracy, and the severity of user impairments.

Overall, we believe that the approach of adapting the user interface and input technique to both abilities and task is promising. Adaptive Click-and-Cross illustrates the potential improvements in navigation from intelligently modifying the user interface while enabling access to smaller targets.

CONCLUSION

This work was spurred in part by the observation that the word "adaptive" is used to describe a multitude of different approaches in the context of interactive systems. In the accessibility community, adaptive technologies help a user with an impairment adapt him- or herself to user interfaces that were designed with able-bodied users in mind. The proponents of ability-based user interfaces, in contrast, advocate adapting *user interfaces* to the unique abilities of the individuals who use them. Yet others have recognized that most users use only a fraction of the capabilities available in any complex application, but each person uses a different subset of those capabilities. In those contexts, adaptation is performed by reallocating the most precious interaction resources to those tasks that the user is expected to do next. In previous work, these different adaptation approaches have been largely pursued in isolation. However, these approaches are not mutually exclusive: they have complementary strengths and weaknesses and — we hypothesized — they can be synergistically combined.

We explored this synergy through Adaptive Click-and-Cross, an interaction technique designed to improve the performance of users with severe dexterity impairments. Adaptive Click-and-Cross relies on knowledge of a user's task to combine two adaptive approaches: adapting frequently used interface elements to a user's motor abilities while using an adaptive accessibility technique (Click-and-Cross) to enable access to those elements that are unlikely to be frequently used.

Our results demonstrate that Adaptive Click-and-Cross improved efficiency without sacrificing accuracy compared to two previously studied approaches: enlarging all user interface elements to match a user's motor abilities, and Click-and-Cross, an interaction technique that enables access to small user elements possible (though inefficient) for users with severe motor impairments.

Our work explored one point in a large design space, but the results suggest that hybrid adaptive approaches are a promising area of inquiry.

REFERENCES

1. Cockburn, A., Gutwin, C., and Greenberg, S. A predictive model of menu performance. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM (New York, NY, USA, 2007), 627–636.
2. Findlater, L., and Gajos, K. Z. Design space and evaluation challenges of adaptive graphical user interfaces. *AI Magazine 30*, 4 (2009), 68–73.
3. Findlater, L., Jansen, A., Shinohara, K., Dixon, M., Kamb, P., Rakita, J., and Wobbrock, J. O. Enhanced area cursors: reducing fine pointing demands for people with motor impairments. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, UIST '10, ACM (New York, NY, USA, 2010), 153–162.
4. Findlater, L., and McGrenere, J. A comparison of static, adaptive, and adaptable menus. In *Proceedings of ACM CHI 2004* (2004), 89–96.
5. Findlater, L., and McGrenere, J. Impact of screen size on performance, awareness, and user satisfaction with adaptive graphical user interfaces. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, ACM (New York, NY, USA, 2008), 1247–1256.
6. Gajos, K. Z., Czerwinski, M., Tan, D. S., and Weld, D. S. Exploring the design space for adaptive graphical user interfaces. In *AVI '06: Proceedings of the working conference on Advanced visual interfaces*, ACM Press (New York, NY, USA, 2006), 201–208.
7. Gajos, K. Z., Everitt, K., Tan, D. S., Czerwinski, M., and Weld, D. S. Predictability and accuracy in adaptive user interfaces. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, ACM (New York, NY, USA, 2008), 1271–1274.
8. Gajos, K. Z., Weld, D. S., and Wobbrock, J. O. Automatically generating personalized user interfaces with Supple. *Artificial Intelligence 174* (2010), 910–950.

9. Greenberg, S., and Witten, I. Adaptive personalized interfaces: A question of viability. *Behaviour & Information Technology 4*, 1 (1985), 31–45.
10. Greenberg, S., and Witten, I. H. Supporting command reuse: mechanisms for reuse. *International journal of man-machine studies 39*, 3 (Sept. 1993), 391–425.
11. Grossman, T., and Balakrishnan, R. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '05, ACM (New York, NY, USA, 2005), 281–290.
12. Harada, S., Wobbrock, J. O., and Landay, J. A. Voicedraw: A voice-driven hands-free drawing application. In *ASSETS'07*, ACM Press (2007).
13. Hornof, A. J., and Cavender, A. Eyedraw: enabling children with severe motor impairments to draw with their eyes. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM Press (New York, NY, USA, 2005), 161–170.
14. Hwang, F., Hollinworth, N., and Williams, N. Effects of target expansion on selection performance in older computer users. *ACM Transactions on Accessible Computing (TACCESS) 5*, 1 (2013), 1.
15. Kabbash, P., and Buxton, W. A. The "prince" technique: Fitts' law and selection using area cursors. In *Proc CHI '95*, ACM (1995), 273–279.
16. Komarov, S., Reinecke, K., and Gajos, K. Z. Crowdsourcing performance evaluations of user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, ACM (New York, NY, USA, 2013), 207–216.
17. Lafreniere, B., Bunt, A., Whissell, J. S., Clarke, C. L. A., and Terry, M. Characterizing large-scale use of a direct manipulation application in the wild. In *Proceedings of Graphics Interface 2010*, GI '10, Canadian Information Processing Society (Toronto, Ont., Canada, Canada, 2010), 11–18.
18. Li, L. Adaptive click-and-cross: An interaction technique for users with impaired dexterity. In *Assets '13: Proceedings of the 15th international ACM SIGACCESS conference on Computers and accessibility* (2013), 2 pages. To appear.
19. Sears, A., and Shneiderman, B. Split menus: effectively using selection frequency to organize menus. *ACM Trans. Comput.-Hum. Interact. 1*, 1 (1994), 27–51.
20. Trewin, S., Keates, S., and Moffatt, K. Developing steady clicks:: a method of cursor assistance for people with motor impairments. In *Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility*, ACM (2006), 26–33.
21. Tsandilas, T., and Schraefel. An empirical assessment of adaptation techniques. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, ACM Press (New York, NY, USA, 2005), 2009–2012.
22. Wobbrock, J., Kane, S., Gajos, K., Harada, S., and Froehlich, J. Ability-Based Design: Concept, Principles and Examples. *ACM Transactions on Accessible Computing (TACCESS) 3*, 3 (Apr. 2011).
23. Wobbrock, J. O., Fogarty, J., Liu, S.-Y. S., Kimuro, S., and Harada, S. The angle mouse: target-agnostic dynamic gain adjustment based on angular deviation. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, ACM (New York, NY, USA, 2009), 1401–1410.