

Strategies for Accelerating On-line Learning of Hotkeys

Tovi Grossman, Pierre Dragicevic, Ravin Balakrishnan

Department of Computer Science

University of Toronto

www.dgp.toronto.edu

{tovi, dragice, ravin}@dgp.toronto.edu

ABSTRACT

Hotkeys are extremely useful in leveraging expert performance, but learning them is a slow process. This paper investigates alternative menu designs that can motivate and help users remember associations between menu commands and hotkeys. Building upon previous work on paired-associate learning, we suggest that the transition to expert use can be accelerated by manipulating feedback and cost associated with menu selection. We evaluate five designs in a pilot study and then two of the most promising ones in a formal experiment, showing that the speed of hotkey learning can indeed be significantly increased with little modifications to the standard menu/hotkey paradigm.

Author Keywords: Hotkeys, learning.

ACM Classification Keywords: H5.2 [Information interfaces and presentation]: User Interfaces - Graphical user interfaces.

INTRODUCTION

It has long been understood that for a graphical user interface to be effective it should provide shortcuts to allow for efficient interaction [9, 13]. One of the best examples of such a shortcut is a hotkey, also known as a keyboard accelerator. Hotkeys allow users to perform operations with a single key press or sequence of key presses, as an alternative to using the mouse to click on graphical objects such as icons and menus. Even the simplest keystroke level analysis indicates that using a hotkey will be faster than clicking on an icon, or navigating through a hierarchical menu [5]. While alternatives to hotkeys have been proposed [3, 7, 8, 12], hotkeys are still prevalent in the majority of today's graphical user interfaces. They are an important asset to many of today's casual and power users, and supporting them is strongly recommended by HCI practitioners [13].

However, a difficulty associated with the use of hotkeys is that a command's associated key presses may not be known to the user. Once known to the expert, a hotkey can be extremely useful. But it has little value to the novice user who may not even know that an associated hotkey exists for a given command. Surprisingly little research has been done looking at how systems can aid the user's transition from the novice behavior of using menus, to the expert behavior of using hotkeys.

Some solutions aimed at accelerating this novice to expert transition are implemented in commercial software, but have limitations. For example, one method is to display hotkey labels next to the associated commands in the menu (Figure 1). However, users commonly ignore the displayed hotkey label since once the menu is open they already have access to their desired command. Unless the user explicitly takes the time to learn the displayed mapping, the hotkey will remain unlearned and therefore go unused.

In this paper, we discuss our attempts at designing on-line techniques, which do not disturb the user's primary task, that better aid and motivate the user's transition to using hotkeys. We begin by discussing related work, and some relevant lessons from research on paired-associate learning. Informed by these lessons, we describe the design of a number of techniques which we have implemented. We then present a pilot study which allowed us to identify two of the most promising approaches: providing auditory feedback upon menu selection and disabling menu selection to motivate the use of hotkeys. In particular, these techniques accelerated the transition to hotkey use and received positive subjective feedback. In a formal experiment, we further evaluate these two techniques in comparison to a baseline condition, and find that they increase hotkey use by over 100%, and reduce task completion times.

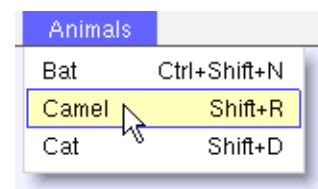


Figure 1. Hotkey labels are traditionally displayed to the right of their associated command in a menu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2007, April 23 – May 3, 2007, San Jose, California, USA.

Copyright 2007 ACM x-xxxxx-xxx-x/xx/xxxxx...\$x.xx.

RELATED WORK

In this section we describe relevant work. We first present a review of hotkeys, including some of their benefits and drawbacks, along with possible alternatives. We then discuss some important lessons which can be learned from the psychology literature in paired associate learning, which will help guide the design of our new techniques.

Hotkeys and Alternatives

Hotkeys have been used in computer interfaces for many years now, even before the prevalence of the GUI. For example, UNIX users will be familiar with applications such as *emacs* and *vi* for which hotkeys are used for almost every command and mode change. With the advent of the GUI, hotkeys are still popular. They nicely conform to Nielsen's heuristic of providing the user with shortcuts [13], and have been shown to be faster than selecting items from menus and toolbars [12, 14]. While hotkeys can be beneficial to end-users, they have the following limitations which we hope to address.

Hotkeys are hard to learn. Selecting drop-down menu items and using hotkeys are two radically different physical actions [9]. Learning the mappings between these two actions is especially difficult when they are arbitrary and when they are not consistent across applications. As the number of commands increases, the number of meaningful mappings quickly runs out, and the use of arbitrary mappings seems unavoidable. We are thus motivated to seek out strategies to aid the learning of these mappings.

Hotkeys lack visibility. In the most extreme case, commands are only available through hotkeys, and the only way to become aware of them is through documentation. More commonly, commands are accessible through menus where their associated hotkey labels are also displayed. However these displayed hotkey labels are commonly ignored, with the user remaining unaware of them [9].

While our work will focus on alleviating these difficulties, others have also explored alternative options to hotkeys. Speech and gesture recognition are two alternative ways of quickly invoking commands. However they each suffer from inherent error-proneness and unpredictability [16], and also suffer from the same visibility problem. A more practical alternative is localized tools, which avoid costly "cursor round-trips" [9]. Examples include ToolGlasses [3], Tracking Menus [7], and Marking Menus [8, 10]. These techniques are generally not seen as replacements for hotkeys, but more as techniques which can complement them in a user interface. Marking Menus are of particular interest to our work, as they elegantly allow for a seamless transition from novice to expert use. A novice brings up a localized radial menu, locates the desired option, and then makes a mark in that direction. An expert, knowing the direction of the desired option, can make a quick mark in that direction, without first displaying the menu. Our goal is to find techniques that similarly allow users to transition from novice use of menus to expert use of hotkeys.

Lessons From Paired-Associate Learning

To the best of our knowledge, the problem of improving on-line learning of potentially meaningless action mappings has never been studied in HCI. However, similar tasks have been studied in psychology under the term of "paired associate learning". Such a task requires the user to memorize the association between cues (e.g., Save) and responses (e.g., Ctrl+S) so that they are later able to give the responses in the presence of cues. While there are some differences in the natures of the paired-associate and hotkey learning tasks, such as the time allotted to learning the pairs, the findings from paired-associate learning studies can give us some useful insights into how we could design menus to improve hotkey learning. We now outline some of these findings.

Exposure. Unsurprisingly, the more often a pair is presented, the better it is retained [4, 15]. Because most frequent commands have a higher learning value, this supports the use of an on-line learning paradigm in which repetition is intrinsically linked to importance.

Attention. Memory research has shown that attention plays a crucial role in the learning process [4]. This suggests that calling the user's attention to the hotkey association when s/he selects a menu item might improve learning. This, however, requires some amount of attention and cognitive processing, which hardly seems possible with the extremely brief exposure times in current menu designs. Making these exposure times last longer and possibly calling the user's attention to the association before or after item selection could aid learning.

Incidental Learning. Although learning is often an effortful act [15, 18], studies have shown that learning can also occur during simple exposure to paired-associate stimuli, when the subject is given no instruction to learn [17]. Quite interestingly, this suggests that hotkey learning could also occur as a mere by-product of completing tasks. Therefore, a technique which provides users with an initial motivation to use the hotkeys, could aid the users learning of the mappings, through simple repetition.

Modality of Presentation. Research has shown that the use of a visual iconic presentation modality instead of text significantly improves paired-associate learning. The auditory channel also helps, although to a lesser extent [19]. The use of the auditory channel might however be particularly beneficial when most of the user's visual attention is already allocated to the ongoing computing task. Also, echoic memory lasts longer than iconic memory [2] and thus might provide longer study opportunities.

In summary, while hotkeys have a number of beneficial properties, they also have some associated shortcomings for which we are motivated to find possible solutions. With the lessons from paired associate learning in mind, we now present a number of techniques which we hope will both aid and motivate users to transition to using hotkeys.

TECHNIQUES

Most computer application users will never open a user's manual, sit down and study hotkeys. The intelligent tutoring paradigm, generally embodied by advice systems and agents, is promising [6], but it has been suggested that too much user guidance can also be counterproductive [18]. Users most often learn hotkeys by using applications. We, therefore, argue that the best approach for teaching hotkeys is probably an on-line approach, which emphasizes learning within the flow of the user's primary tasks. Our techniques will leave the existing menu and hotkey interaction styles mostly unchanged, so that they can be easily integrated into existing GUI toolkits and applications.

Feedback-Based and Cost-Based Approaches

Our previous discussions suggest that two main strategies can be employed to serve our purposes:

Manipulating menu feedback: Increasing the visibility and/or persistence of the command/hotkey association every time a menu item is browsed or selected to increase the users' attention to it.

Manipulating menu cost: Increasing the difficulty and/or the time cost of using the menu for selections to motivate users to instead use the hotkeys.

These strategies are relevant to both intentional and incidental learning: they can potentially affect intentional learning because they might provide the user incentive to learn, or act as a reminder to learn; they can also potentially affect incidental learning because they increase overall exposure to command/hotkey stimuli without totally disrupting the flow of the primary task.

Examples of Designs

We now describe traditional menus as well as eight possible design variations we implemented. We are interested in knowing more about the limits of acceptable distraction and disruption, thus we have intentionally devised some techniques that push these limits. The following techniques are meant to be elemental, as many can be combined together.

a. Traditional menus. Traditional menus display hotkeys next to menu items. Sometimes, menu items also rapidly fade out after they are selected (Figure 2a). These designs only weakly draw the hotkey to the user's attention.

b. Fading-out hotkey. In this technique, when the user clicks on a menu item, the menu closes and only the command's hotkey stays visible, but then it slowly fades out (Figure 2b). This is a slightly stronger feedback-based approach than the traditional menu.

c. Hotkey menu replacement. A problem with the traditional technique is that users may not pay any attention to the right column of the menu where the hotkeys are displayed. With this feedback-based approach, when the user hovers over a menu item, the hotkey is displayed in place of the command name (Figure 2c). This technique should increase the user's visual exposure to hotkeys, as the hotkey will be displayed at the location where the user is focusing.

d. Audio Feedback. This is another feedback-based approach, however the feedback is auditory. When the user clicks on a menu item, the menu closes and the item content is played using speech feedback (Figure 2d). Audio feedback builds upon the idea that the visual channel is often already saturated. Both the command name and hotkey are played to keep the hotkey in context in echoic memory.

e. System Delay. This is one of two cost-based techniques. When the user clicks on a menu item, the menu closes, the mouse cursor disappears, and a small progress bar appears with the hotkey next to it (Figure 2e). After two seconds, the progress bar diminishes, and the command is invoked. While this technique uses a cost-based approach, it also increases hotkey exposure as a byproduct. We also found that hiding the mouse cursor to be very effective at grabbing the user's attention.

f. Disabled menu items. This is our other cost-based approach. The menu grays out all command names. When the user clicks on a menu item, the command is not invoked and the shortcut briefly flashes (Figure 2f). This forces users to use the hotkey, and our hope is that it will aid in their learning of the hotkey mappings.

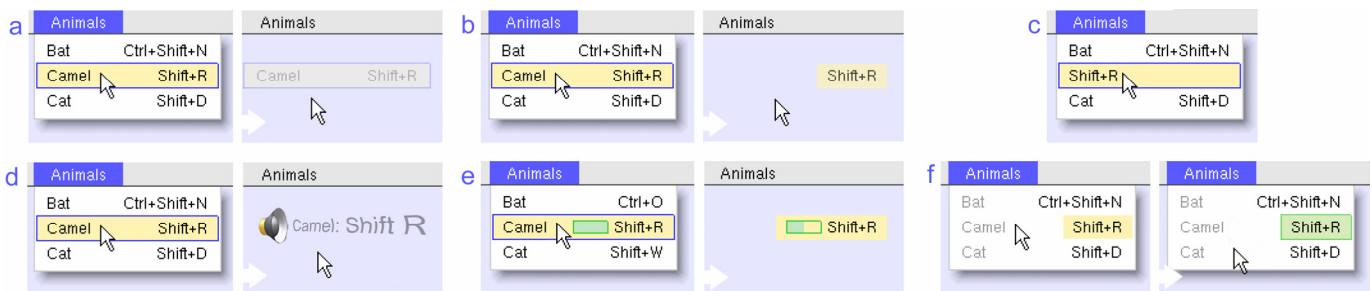


Figure 2. Examples of designs emphasizing hotkey mappings. (a) Traditional design. (b) Fading-out hotkey. (c) Hotkey menu replacement. (d) Audio feedback. (e) System delay. (f) Disabled menu items. Refer to text for detailed description.

g. *Blinking Hotkey*. When the menu closes in this technique, the hotkey stays visible and blinks for a short duration. This technique increases the level of feedback, with a stronger focus on capturing the user's attention.

h. *Following Hotkey*. This is another feedback-based approach. When the user clicks on a menu item, the menu closes but the hotkey stays visible and its position is bound to the mouse cursor. This technique is supposed to grab the user's attention even more strongly. It is based on the observation that after selecting a menu item, the user's attention and the mouse pointer will both switch to a different location on the screen.

i. *Visual feedback*. This is a combination of the *blinking hotkey*, *following hotkey*, and *hotkey menu replacement* techniques. Basically we wanted to provide the strongest possible visual feedback

PILOT STUDY

In order to get a sense of the relative effectiveness of our proposed techniques, we ran an initial within-participant pilot study. This allowed us to choose the techniques which showed the most potential for a more formal experiment. From the previously discussed techniques, we settled on four which we found to be promising in informal usage observations, for comparison with the *traditional* technique which acted as a control baseline. The four techniques we chose were *visual feedback* (i), *audio feedback* (d), *system delay* (e), and *disabled menu items* (f).

Apparatus

The experiment was conducted on a 3.2Ghz Pentium4 PC running Windows XP with OpenGL for graphics, and a 20" LCD display at 1600x1200 resolution. An optical mouse and standard Windows 104-key QWERTY keyboard were used for input. Participants were allowed to adjust the sensitivity and acceleration of the mouse to a level they were comfortable with. Participants wore headphones whose volume was adjusted to a comfortable level.

Participants

Five paid volunteers (all male) participated in the experiment. Participants ranged in ages from 25-31.

Procedure

A stimulus was presented in the form of a graphical image of a highly familiar object, such as a banana or a cat. The user was instructed to enter the object's name either by selecting it from a standard menu bar at the top of the screen, or by using the menu item's hotkey. Preliminary testing ensured that the icon/name associations were trivial and unique. The use of iconic stimuli was motivated by the fact that commands are not usually invoked in response to verbal or textual cues, but rather through some cognitive process. Using verbal or textual stimuli would have had too close a correspondence to the menu items.

A trial began with the user clicking in a 70x70 pixel box centered at the bottom of the display, upon which the stimulus image would appear inside. The user could then either select the corresponding item from the menu (Figure 3), or use its associated hotkey. Users were not instructed to use one method or the other; it was completely up to them. Once the command was successfully entered, the user would again click in the start box. This ended the current trial and immediately began the next. After 10 successive trials, a dialogue box displayed the user's progress, and allowed for short breaks. If the user made an error during a trial, by either selecting the wrong menu item or pressing the wrong hotkey, a 3 second time delay would be administered to penalize the participant. This helped to prevent users from rushing through the experiment without regard for accuracy, and also from rapidly guessing hotkeys until the correct assignment was found.

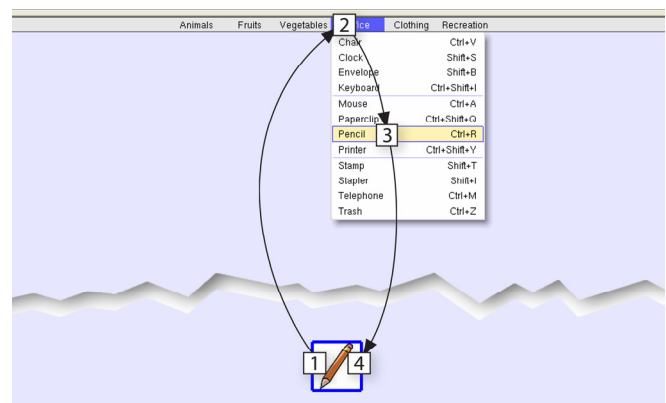


Figure 3. A menu selection in the experimental procedure. The labels 1-4 indicate the four steps in the process.

The menu bar was divided into 5 categories, each with their own menu containing 12 items, for a total of 60 items. The categories were *animals*, *fruits*, *vegetables*, *office*, and *clothing*. Only 4 items in each of these menus were actually used in the experiment, for a total of 20 items. The height of the menu bar and menu items was 20 pixels, which is roughly the size of menus commonly used in practice.

Keyboard hotkeys were assigned to each of the 20 items used in the experiment as follows. First, to control the biomechanical difficulty associated with pressing the hotkey, we only used the 10 leftmost alphabetic keys: Q, W, E, R, A, S, D, Z, X, and C. Next, each of these keys was combined with either SHIFT or CTRL for a hotkey assignment. Lastly, each item was randomly assigned one of these hotkeys such that no hotkey assignment involved any of the letters of its corresponding item, the letters Q and C being assumed equivalent. For example, cat could be assigned SHIFT-R, but not SHIFT-Q or SHIFT-A. This method of assignment devised from preliminary studies made all mnemonic correspondence between hotkeys and items equally difficult. The 40 unused items were randomly assigned other hotkeys. The item to hotkey mapping remained constant across all participants.

Participants were paid a base \$5 for completing the experiment. To motivate them to perform well, they were given \$1 for every two minutes under an hour that they completed the experiment, to a maximum of \$15. This time did not include warm-up trials and breaks. It did, however, include the time between command invocation and the subsequent click on the start box. They were thus motivated not to linger on the menus after a selection, which was important for the ecological validity of the experiment.

Because we wanted participants to understand that they could make choices (such as choosing to learn and use hotkeys), clear written instructions were given prior to doing the experiment. Participants were not told the purpose of the experiment, they were simply asked to be as fast as possible. With regards to technique usage, the only instruction given was that there were two ways to enter a command: using the menu or its associated hotkey. Differences between the menus were explained orally.

Design

A repeated measures within-participant design was used. The independent variable was menu technique (*control*, *visual*, *audio*, *delayed*, and *disabled*). Each of the 5 category menus was assigned to one technique throughout the experiment. The technique-to-menu mapping was counterbalanced across the five participants using a Latin square design.

Each participant performed the experiment in one session lasting about 40 minutes. The experiment was divided in two halves. In the first half, only two items from each menu would appear, and in the second half the other two items would appear. This meant that in each half of the experiment, only 10 items could appear across the 5 menus. The frequency of appearance of each item was uniformly distributed. This design was chosen because initial studies showed that presenting more than 10 items at a time would make it too difficult to learn any hotkeys, and presenting fewer than 10 items at a time would make learning the hotkeys too easy. Using this design meant that trials for each menu technique were being mixed together. It was very likely that this would result in interference between techniques. For example, if one menu technique provided motivation to learn hotkeys, then users may have been motivated to learn hotkeys for other menu techniques as well. However, the only alternative was to present the techniques one after the other, but we would still expect to see transfer effects, and there would be an added complexity of memory transfer. A benefit of this design is that it allows users to subjectively compare the techniques upon completion of the experiment, which was one of the main purposes of this pilot study.

Each half of the experiment consisted of 30 blocks. Each block consisted of 10 trials, with each of the ten possible items appearing once in random order. This design resulted in a total of 600 trials per participant. A short warm-up session was performed prior to beginning the experiment,

allowing the participant to learn the task, and the differences between the five techniques. The warm-up targets were different from those used in the experiment.

Results

Hotkey Use

Although users were paid based on their completion time, our main concern was how quickly the users learned and used the hotkeys. To analyze this, we defined *HOTKEY USE*, the proportion of trials in which participants used the hotkey, without first referring to the menu.

The *HOTKEY USE* measure is only measuring *expert* behavior with hotkeys. If the user first had to drop down the menu to find the hotkey assignment, it would be considered *novice* behavior and not included in the *HOTKEY USE* measure. If this distinction were not made, the *disabled* technique would have a misleading 100% *HOTKEY USE* value.

In our analysis of *HOTKEY USE*, we removed trials in which users made an error before using the correct hotkey (6.9% of the data). Repeated measures analysis of variance showed a significant main effect for menu technique on *HOTKEY USE* ($F_{4,16} = 17.39$, $p < .0001$). *HOTKEY USE* was 70% for *audio*, 69.2% for *disabled*, 59.1% for *delayed*, 57.1% for *control*, and 50% for *visual*. Post hoc multiple means comparisons showed that *audio* and *disabled* resulted in significantly greater *HOTKEY USE* than the other techniques ($p < .005$).

It is interesting to note that *visual* actually did worse than the *control* condition, although the difference was not significant. The most probable explanation for this is that users were concentrating on completing their task at hand, so as soon as they made a menu selection, they switched their visual attention to the start box. Even though the hotkey was blinking and following their cursor, they would not pay attention to it. This is the exact same problem the control condition suffers from, which uses even weaker visual feedback, and provides good justification for our current work, in search of more effective techniques.

In comparison, the *audio* condition did provide significant improvement ($p < .005$). This shows that while attempts at gaining the users' attention through their visual channel failed, attempts at gaining their attention through their audio channel succeeded.

It is also interesting to note that the *disabled* technique provided a significantly higher value of *HOTKEY USE* in comparison to the *delayed* technique ($p < .005$). This is an interesting difference since both techniques use the strategy of reducing the efficiency of the menu to motivate the use of the hotkey, by either delaying its responsiveness, or by completely disabling its functionality. The most important difference was probably that the *disabled* technique forced users to practice the expert behavior of using the hotkey.

Learning

Although there were 30 blocks, our analysis grouped the blocks in pairs to increase the number of samples at each level. As expected, block had a significant effect on *HOTKEY USE* ($F_{14,56} = 172.63$, $p < .0001$), with increased hotkey use as the experiment progressed. The interaction between block and menu technique was marginally significant ($F_{56,224} = 1.3$, $p < .07$), indicating that learning occurs differently for each technique (Figure 4). With the *audio* and *disabled* techniques, users transitioned to hotkey use early in the experiment, while the other techniques indicate similar hotkey use only later in the experiment.

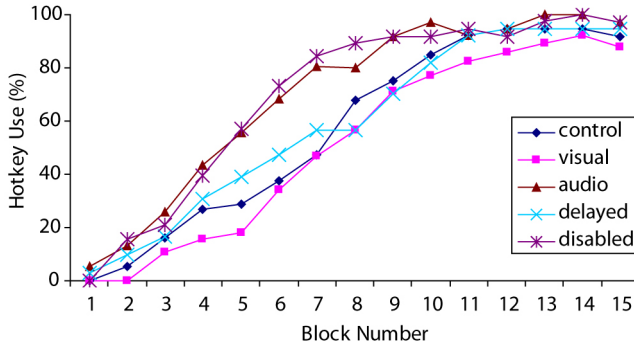


Figure 4. Hotkey use by block number and technique.

Trial Completion Time

Participants were motivated to be as fast as possible, so they should have only transitioned to hotkey use if such a transition were beneficial in terms of completion times. We defined trial completion time as the time between clicking the box to start and then end the trial. Repeated measures analysis of variance showed that menu technique had a significant effect on completion time ($F_{4,16} = 6.21$, $p < .0001$). Times were 3.16s for *disabled*, 3.21s for *audio*, 3.41s for *control*, 3.58s for *delay*, and 3.74s for *visual*. As anticipated, the techniques that had higher values of *HOTKEY USE* took less time. When we compare the completion times for when the menu was and was not used, we see a much stronger effect ($F_{1,4} = 366$, $p < .0001$), with average times of 5.19s for when the menu was used, and 2.31s for when it was not. This data does not include *disabled* and *delayed* menu use, as these techniques intentionally slow down the completion time.

Subjective Results

Upon completion of the study, participants were asked to rate the helpfulness of each of the new techniques on a 7 point scale, where 1 represents hindrance, and 7 represents helpfulness. The question was referring to the completion of the primary task, not the learning task. Analysis of variance showed that response was significantly affected by menu technique ($F_{3,12} = 7.289$, $p < .005$), with *audio* providing the highest score (6.6), followed by *visual* and *disabled* (4), and then *delayed* (2.2). This data indicates that the *delayed* technique is probably too disruptive to be accepted by users.

Summary

Our results indicate that our techniques can indeed aid the transition to hotkey use, and that this transition will result in faster command input. Even under our simplified within-participants design, the *audio* and *disabled* techniques increased the use of hotkeys, and decreased trial completion time, in comparison to other techniques.

Interestingly enough, these two techniques use different approaches to aid the transition to using hotkeys. The *audio* technique is a feedback-based approach, providing an explicit aid to learning hotkey assignments through the audio channel, as well as implicit motivation to make the transition to hotkey use. Conversely, the *disabled* technique is a cost-based approach, providing explicit motivation to use the hotkeys, while the memory aid only comes implicitly through incidental learning from repetitive use.

The fact that these two techniques have complementary properties, and both showed potential in the pilot study makes them good candidates for our follow-up experiment.

EXPERIMENT

In this experiment, we further evaluate the *audio* and *disabled* techniques which we identified as being particularly promising in our pilot study. Because our earlier study was only preliminary, we were forced to make a number of simplifications to both the experimental design and task. The goal of this study is to provide a more sound comparison of our two most promising techniques with the control condition. As such, we made the following modifications to the experimental procedure.

The first change was to use a between-participants design, with each participant being assigned to only one of the menu techniques. This was done to eliminate transfer effects from one technique to the next. Using a within-participants design and counterbalancing the presentation order of techniques would not suffice in this experiment for a number of reasons. First, if the participants were initially presented with a technique that increases the motivation of hotkey use, then they may apply that motivation to the subsequent techniques that they use. Secondly, it would be confusing if participants were forced to memorize new associations for the same accelerator keys. The only way to get around this would be to use different accelerators for each technique, which would again add an unwanted complexity to the experimental design. Also, this would prevent us from using a larger number of targets with a non-uniform distribution of repetitions, as further described.

The second change was that unlike in the pilot experiment, where the frequencies of all targets were distributed uniformly, the frequencies of the targets in this experiment were made to be more realistic. We use a Zipfian distribution which has been shown to represent command use frequency in real applications [20]. This will also allow us to compare how the frequency of a command affects the user's ability to learn its associated hotkey.

Apparatus

The apparatus was identical to that in the pilot study. All participants wore headphones, regardless of their assigned condition.

Participants

42 new paid volunteers (12 female, 30 male) participated in the experiment. Participants ranged in ages from 18-28.

Procedure

The procedure was similar to that used in the pilot study. One change was that instead of clicking in the box to start and end a trial, users would position the cursor inside of it, and then hit the space bar. This was done to control the position of the user's left hand, as this could affect the user's decision to use a hotkey. Another change was that an extra category, "recreation", was included, for a total of 72 menu items. A total of 14 target items distributed among the six categories was used for the experiment. Participants were again paid, between five and twenty dollars, based on their performance, on the same scale as in the pilot study. Written instructions were again given to the participants before the experiment began, describing the task, explaining hotkeys, and stating again that they could make selections with either the menu or hotkeys. They were additionally informed that they would be selecting items with a repetition pattern that would remain constant throughout the whole experiment.

Keyboard hotkeys were assigned to each of the 14 items in the same manner as the first experiment, using Q, W, E, R, A, S and D in combination with either SHIFT or CONTROL. Hotkeys involving the keys C, X and Z were discarded based on their common usage in current applications, making the difficulty even more uniform. The hotkey mapping remained constant across all participants.

Design

A mixed design was used. The between-participants independent variable was the menu technique (*control*, *audio*, *disabled*). Each participant was randomly assigned to one of the three menu techniques. The within-participant variable was item frequency, which was computed on a basis of 30 random drawings of 7 items and a standard Zipfian distribution of exponent 1 (relative frequency = $1 / \text{rank}$) [21]. Frequencies were rounded off and we subtracted 1 from the frequency of the last item to get a total of 30 drawings. This yielded the frequencies (12, 6, 4, 3, 2, 2, 1). There were 14 target items used for the experiment, two assigned to each frequency. The item to frequency mapping was counterbalanced across all participants, with each item mapped to each frequency an equal number of times.

Each participant performed the experiment in one session lasting between 30 and 60 minutes. The experiment was divided into 12 blocks of 60 trials each. Within each block, items were presented in random order, with the constraint that the frequencies of each target across the entire block matched its given frequency, as determined by the Zipfian

distribution. For example, the two items mapped to $F = 12$ would each appear exactly 12 times in every block. A short warm-up session consisting of 10 trials involving different items not used in the actual trials was performed prior to beginning the experiment,

Results

Hotkey Use

As with the pilot study, we define *HOTKEY USE* as the percentage of trials for which participants used the hotkeys, without reference to the menu. Repeated measures analysis of variance showed a significant main effect for menu technique on *HOTKEY USE* ($F_{2,82} = 47.0$, $p < .0001$) and item frequency ($F_{5, 205} = 29.9$, $p < .0001$). The *HOTKEY USE* values were 28.9% for *control*, 66.6% for *audio*, and 72.8% for *disabled* (recall that *HOTKEY USE* for *disabled* can be less than 100% because *HOTKEY USE* only considers trials where the menu is not opened before using the hotkey). Post hoc analysis shows that *control* produced significantly lower values than both *audio* and *disabled* ($p < .001$), and that *audio* and *disabled* were not significantly different. There was no interaction effect between menu technique and item frequency which may indicate that our techniques can improve the transition to both frequently and less commonly used commands.

These are important results, demonstrating that both of our tested techniques encouraged users to transition to the expert use of the hotkeys, resulting in more than twice as much hotkey use. Before looking further into these results, it is important to note that there were some large variations within each technique, from one subject to the next. In the *control* condition, 7 of the 14 participants did not make any effort to learn or use the hotkeys, so they had 0% *HOTKEY USE*. In the *audio* condition, there was only one such participant with 0% *HOTKEY USE*, and in the *disabled* condition all participants learned at least some of the hotkeys. Figure 5 illustrates. This demonstrates that the major benefit of our techniques was that they successfully motivated users to transition to expert use of hotkeys.

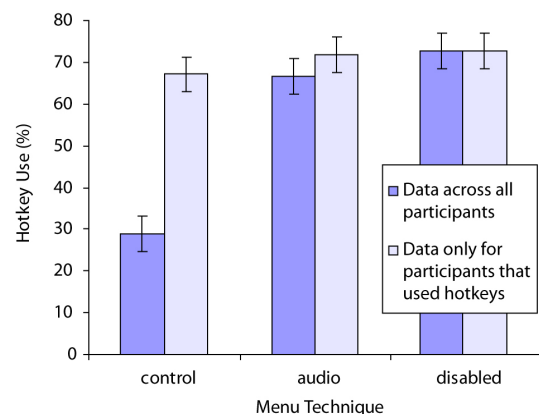


Figure 5. Hotkey use by technique, with 95% CI marked.

To account for these strategy differences, we performed further analysis after excluding data from the participants who did not use the hotkeys. Repeated measures of analysis of variance showed that menu technique still had a significant affect on *HOTKEY USE* ($p = 0.034$). The new *HOTKEY USE* values were 68.1% for *control*, 71.9% for *audio*, and *disabled* unchanged at 72.4%. Post hoc analysis showed that *control* was still significantly lower than *disabled* ($p = 0.028$), and although lower than *audio*, the significance was only weak ($p = 0.092$). As before, *audio* and *disabled* were not significantly different. This indicates that even when users assigned to the *control* condition were motivated to transition to hotkey use, they were still not able to do so as well as participants assigned to the other two conditions.

The effect of item frequency also remained significant after removing the data for participants who did not use hotkeys ($p < .0001$). The effect is illustrated in Figure 6. Post hoc analysis shows that all pairs are significantly different at the $p < .0001$ level except for item frequency = 4 and 6, for which $p = 0.03$. This result is somewhat expected, showing that users are more likely to learn and use hotkeys for the more frequently used commands.

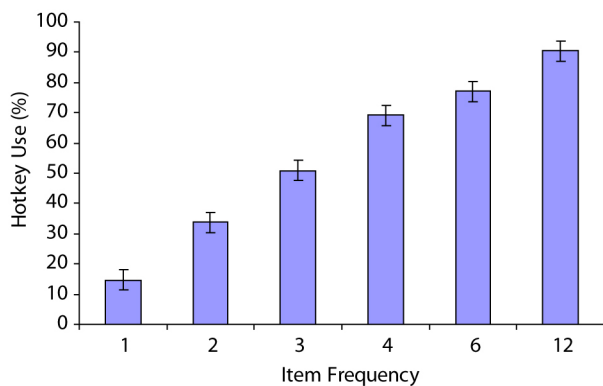


Figure 6. Hotkey use by item frequency, with 95% CI marked. Data is only for the participants who used hotkeys.

Learning

As in the pilot study, there was a strong effect for block number on *HOTKEY USE*. We, again, only include data for the participants who actually transitioned to using the hotkeys in our analysis. Repeated measures analysis of variance shows that *HOTKEY USE* was significantly affected by block ($F_{11,451} = 120$, $p < .0001$). Unsurprisingly, *HOTKEY USE* increased after each block. There was no interaction effect between block and menu technique.

Trial Completion Time

Trial completion time was again defined as the time between hitting the space bar to start and end the trial. Since participants were being paid based on their completion time, the transition to hotkey use would only be justified if it would indeed reduce completion time. Repeated measures analysis of variance showed that completion time was significantly affected by menu technique ($F_{2,42} = 161$,

$p < .0001$), with average trial completion times of 2.93 for *control*, 2.62 for *audio*, and 2.55 for *disabled*. Post hoc analysis shows that *control* was significantly slower than both *audio* and *disabled* ($p < .0001$), and *audio* was significantly slower than *disabled* ($p < .05$). This, in combination with the *HOTKEY USE* data, shows that not only did our techniques motivate participants to transition to hotkey use, but this transition also allowed them to finish the task faster.

There was also a significant block x menu technique interaction effect ($F_{22,902} = 20$, $p < .0001$) for completion time (Figure 7). It is interesting to note that in the first block *audio* and *disabled* actually provide slower completion times than *control*. Post hoc analysis shows that the difference is indeed significant ($p < .0001$). However, at the second block, all techniques provide similar completion times, and for all blocks after this, *audio* and *disabled* are faster, significantly so starting from the 4th block (all $p < .005$). This provides an explanation as to why many of the participants using the *control* technique did not transition to using the hotkeys, as they likely outweighed the initial cost of the transition over the potential overall benefit. This is why it is important for the designer to aid and motivate such a transition, further justifying our current work and new techniques.

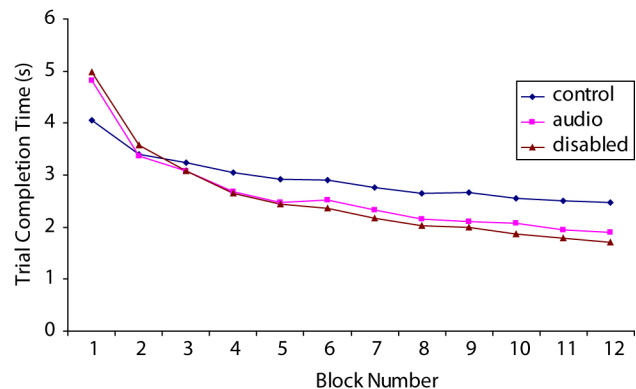


Figure 7. Completion times by block number and technique.

Errors

We defined errors as any trial for which a participant selected the wrong item from the menu, or used an incorrect hotkey. The overall error rate for the experiment was 4.65%. The majority of the errors were the result of incorrect hotkeys being used. Error rate was significantly affected by menu technique ($F_{2,82} = 15.25$, $p < .0001$), with error rates of 5.96% for *audio*, 4.19% for *disabled* and 3.81% for *control*. The higher error rate for *audio* compared to *disabled* may have been due to participants trying to make use of their echoic memory too long after they had completed a trial for the same item.

Subjective Results

Participants answered a short questionnaire upon completing the experiment. We were mostly interested in what the users thought about their assigned menu

technique. Specifically, we asked participants if the menu design helped them learn the hotkeys and if the menu design was distracting. Participants answered these questions on a 6 point scale. They were further asked to rank the menu design from very helpful to very hindering for the purposes of the task, on a 7 point scale. The results showed that menu technique only had a significant effect on the responses to the question about learning the hotkeys ($F_{2,82} = 12.83, p < .0001$), with *control* providing lower scores. This is not surprising, given our analysis of the experimental data. It is interesting to note that the menu technique did not have an effect on responses to the question about the menu being distracting, which may suggest that our techniques could be integrated into interfaces without significantly irritating users.

DISCUSSION

The purpose of our studies was to evaluate whether or not on-line hotkey learning can be accelerated by increasing exposure to hotkeys during menu selection (feedback approach) or by increasing the cost of menu selections (cost approach). We have found two such techniques to be particularly successful. The *audio* and *disabled* techniques both increased the use of hotkeys by more than 100%, and reduced task completion times. Our studies have also shown a number of important points that we now discuss.

Our initial pilot study demonstrated that using visual feedback to display a menu's associated hotkey may not aid the transition to hotkey use. Even when the hotkey flashed and followed the cursor, users still did not transition to using them faster than with the traditional menu design. This is likely a result of users concentrating on their primary task at hand, which is visual in nature, and they clearly do not want to divide their attention. Moreover, the stronger visual feedback was negatively perceived by users.

Contrary to this, the audio feedback was positively perceived by users, and significantly increased the use of hotkeys in both the pilot and full experiments, resulting in lowered task completion times. This was likely more effective than the visual feedback because the audio channel was otherwise unused by the participants.

As for the cost-based approaches, the *delayed* technique, which required users to wait 2 seconds after each menu selection, was negatively perceived by the users, and did not seem to provide any significant differences in comparison to the control condition in our pilot study. This indicates that the delay was frustrating, and users did not attend to the hotkey mapping during the delay period, as hoped. The *disabled* technique, on the other hand, was positively perceived, and significantly increased hotkey use. The benefit of this technique was that it forced users to learn the hotkey through repetition of use.

Even though these results provide strong support for our hypotheses, they should be interpreted with care. Our modified paired-associate task has been designed to

resemble computer use, but it might still be evaluating users in an artificial situation. For example, command invocations were elicited by the system with no connection to a realistic computing task. Also, the time to learn and make use of the hotkeys was bounded. In particular, the perceived helpfulness of a technique or the decision to make use of hotkeys might have been different if short term performance is not the main motivation.

This also raises the intricate question of the exact sources of participant motivation under each of the three conditions. In fact, it is extremely difficult to control all of them. At least for the *audio* condition, we might well be seeing a "good-subject effect", as participants may have felt that using the hotkeys was what they were supposed to do. Furthermore, the audio feedback might have provided a hint about the best strategy to adopt. However, participants were being paid based on their completion times, so we expect that they probably would not have made this transition unless they perceived a personal benefit. Furthermore, the same effect was not seen in the pilot study with the visual feedback, which participants also could have interpreted as an indication about what they were expected to do, and what was the better strategy.

Our studies leave other questions unanswered. In particular, it is not clear whether the observed improvements were the result of intentional or incidental learning. In other terms, did the participants simply decide to allocate more resources to learning hotkeys, or did they also learn them as a byproduct of being exposed to the hotkeys (for the audio condition) or using them (for the disabled condition)? The fact that the task gave little time for voluntary rehearsal speaks in favor of some incidental learning. Significant learning speed differences found in the pilot study (in which intentional learning was leveled by presenting all techniques together) also suggest an incidental learning component. Conversely, the fact that some users in the control condition decided not to use hotkeys at all, while others did, suggests that there was a strong intentional learning component. Overall, it seems reasonable to assume that both intentional and incidental learning are positively affected by our techniques.

CONCLUSIONS AND FUTURE WORK

We have discussed the pros and cons of hotkeys and explained why hotkeys are likely to be used in the future and thus deserve more research. In particular, we argued that several strategies can be employed to accelerate the often slow process of learning hotkey associations. We investigated two of them, namely feedback manipulation and cost manipulation, and demonstrated that they can accelerate the transition to expert use. There are still, however, a number of research directions to be explored.

In future work, we would like to evaluate the issues involved with integrating our techniques into actual applications. It would probably be best if our techniques could be turned on and off by the user when desired.

Another idea would be to allow users to set a maximum level of feedback. Systems could also identify commands that are being frequently used, for which the user is unaware of a hotkey. The system could then isolate these commands and only apply our techniques to them, until the users have learned the associated hotkeys. Although paired-associate learning research suggests that training on all the items at the same time would yield the best outcomes [4], applying our techniques to only certain commands may be less intrusive and put more reasonable demands on users.

Another line of future work is exploring spatial feedback of hotkeys. An example of this would be flashing hotkeys on an on-screen keyboard whenever a command is invoked. An alternative would be flashing the hotkeys directly on a physical keyboard with display capabilities [1]. Such visual feedback may yield different performance and user acceptance from the textual ones which we investigated.

Lastly it would be interesting to compare our techniques to the strategy of using the keyboard to navigate through menus. For example, to select *save*, a user could hit *alt* (to switch to the menu bar) then *f* (for file) then *s* (for save). While the drawback is that more keystrokes would be required, it does present a way for users to continuously transition to expert usage, as they learn the series of key presses for each desired command.

While our paper tackled the somewhat specific problem of hotkey use, there are some important generalizations which we would like to see applied to general user interface principles and design. First, we believe that Nielsen's heuristic, which states that shortcuts should be provided for the expert user, should be strengthened. Not only should these shortcuts be provided, but the user should be made aware of these shortcuts, and the system should aid the transition to their use. This does not only apply to the use of hotkeys, but to any type of accelerator used in an interface.

Related to this, we are also interested in further exploring the issue of user's perceived benefits and costs when considering new techniques. HCI researchers are frequently developing new interaction techniques which improve task completion times. However, our results show that even if there is a slight overhead cost to transitioning to a new technique, without a good motivation for the transition the user may discard the new technique even if it will be beneficial in the long run. We believe this to be an important issue that should be explored in greater detail.

REFERENCES

1. Art Lebedev Studios, *Optimus Keyboard*. <http://www.artlebedev.com/> Last visited. 09/26/2006.
2. Baddeley, A. *Human Memory: Theory and Practice*. 1997, Allyn & Bacon.
3. Bier, E. Stone, M., Pier, K., Buxton, W., DeRose, T. (1993). Toolglass and magic lenses: the see-through interface. *ACM SIGGRAPH*. 73-80.
4. Calfee, R. (1975). Associative learning. In *Human Experimental Psychology*. Holt, Rinehart and Winston. 379-399.
5. Card, S., Moran, T., Newell, A. (1980). The Keystroke-Level model for user performance time with interactive systems. *Communications of the ACM*, 23(7). 396-410.
6. Fischer, G. (2001). User Modeling in Human-Computer Interaction. *User Modeling and User-Adapted Interaction*. 11(1-2). 65-86. Kluwer Academic.
7. Fitzmaurice, G., Khan, A., Pieké, R., Buxton, W., and Kurtenbach, G. (2003). Tracking menus. *ACM UIST*. 71-79.
8. Kurtenbach, G. and Buxton, W. (1994). User learning and performance with marking menus. *ACM CHI*. 258-264.
9. Kurtenbach, G. (1993). *The Design and Evaluation of Marking Menus*. Ph.D. Thesis, University of Toronto.
10. Kurtenbach, G., Fitzmaurice, G., Owen, R., and Baudel, T. (1999). The Hotbox: efficient access to a large number of menu-items. *ACM CHI*. 231-237.
11. MacKenzie, I. and Guiard, Y. (2001). The two-handed desktop interface: are we there yet? *ACM CHI Extended Abstracts*. 351-352.
12. McLoone, H., Hinckley, K., and Cutrell, E. (2003). Bimanual interaction on the Microsoft Office Keyboard. *INTERACT*.
13. Nielsen, J. (1992). Finding usability problems through heuristic evaluation. *ACM CHI*. 73-80.
14. Odell, D., Davis, R., Smith, A., and Wright, P. (2004). Toolglasses, marking menus, and hotkeys: a comparison of one and two-handed command selection techniques. *Graphics Interface*. 17-24.
15. Pavlik, P. (in press). Timing is an order: Modeling order effects in the learning of information. *In order to learn: How the sequences of topics affect learning*. Oxford University Press.
16. Shneiderman, B. (2000). The limits of speech recognition, *Communications of the ACM*, 43(9). 63-65.
17. Shelton, D. and Newhouse, R. (1981). Incidental learning in a paired-associate task. *Journal of Experimental Education*, 50. 36-38.
18. van Nimwegen, C., Burgos, D., van Oostendorp, H., and Schijf, H. (2006). The paradox of the assisted user: guidance can be counterproductive. *ACM CHI*. 917-926.
19. Velayo, R. and Quirk, C. (2000). How do presentation modality and strategy use influence memory for paired concepts? *Journal of Instructional Psychology*, 27(2). 126-133.
20. Witten I., Cleary, J., and Greenberg, S. (1984). *On frequency-based menu-splitting algorithms*. Intl. Journal of Man-Machine Studies 21(2). 135-148..
21. Zipf, G. (1949). *Human behaviour and the principle of least-effort*. Addison-Wesley, Cambridge MA.