Modeling and Improving Selection in Cascading Pull-Down Menus Using Fitts' Law, the Steering Law and Force Fields

David Ahlström
Department of Informatics-Systems
University of Klagenfurt
Universitätsstrasse 65-67, A-9020 Klagenfurt, Austria
david@isys.uni-klu.ac.at
++43 (0)463 27 00 35 14

ABSTRACT
Selecting a menu item in a cascading pull-down menu is a frequent but time consuming and complex GUI task. This paper describes an approach aimed to support the user during selection in cascading pull-down menus when using an indirect pointing device. By enhancing such a cascading pull-down menu with "force fields", the cursor is attracted toward a certain direction, e.g. toward the right hand side within a menu item, which opens up a sub-menu, making the cursor steering task easier and faster. The experiment described here shows that the force fields can decrease selection times, on average by 18%, when a mouse, a track point, or touch pad is used as input device. The results also suggest that selection times in cascading pull-down menus can be modeled using a combination of Fitts' law and the steering law. The proposed model proved to hold for all three devices, in both standard and in enhanced cascading pull-down menus, with correlations better than $r^2 = 0.90$.

ACM Classification:
H.5.2. [Information Interfaces and Presentation]: User Interfaces – Graphical user interfaces, Input devices and strategies, Interaction styles, Theory and methods.

Keywords:
Cascading pull-down menus, Menu navigation, Selection, Fitts' Law, Steering law, Input devices, "Force fields".

INTRODUCTION
A cascading pull-down menu which is used to select operations is an integral part of many modern graphical computer applications. The process of selecting a menu item includes four cognitive activities: the user must 1) read the alternatives in the menu, 2) choose the desired one, 3) effect the choice and 4) ascertain the consequences [20]. This paper focuses on the activity of effecting the choice, i.e. how the user navigates through the menu hierarchy to select the desired menu item using a screen cursor operated by an indirect pointing device[1]. Activity 1, 2 and 4 are well-studied as separate activities (see Norman [20] for a detailed analysis). But studies focusing on the third activity are rare. Fitts' law [11] has successfully been used in studies to compare and to model selection times in menu systems where the choices can be selected using a one-directional motion, as in pie-menus [7] or the selection of a first-level item in a cascading pull-down menu [23]. There have been very few, if any, model based studies of effecting choices in pull-down menus over more than one menu-level. One reason for this seems to be the lack of a suitable model which helps to describe and to understand this activity. When Accot and Zhai [1] discovered the steering law (derived from Fitts' law), an important first step in this direction was made.

The process of selecting a menu item to invoke the corresponding operation is a frequent task which can be cumbersome and time consuming for many users. If the menu items are wide, a rather long horizontal motion is needed to navigate into a sub-menu. During the horizontal motion, it is important that the cursor movement does not diverge too much in the vertical direction and leave the parent item, which will close the sub-menu. Menu navigation becomes particularly difficult in the context of mobile computing where input devices with questionable ergonomic properties (such as the track point, the touch pad and the touch ball) are used. Furthermore, the environment in which mobile computing takes place (in planes, in trains etc.) most often puts additional constraints upon the user during GUI navigation, in particular when precise input device motions are required.

In the following two sections we 1) propose a model for selection times in cascading pull-down menus, based on Fitts' law and the steering law, and 2) introduce the usage of "force fields" to support the user during menu selection, making the process easier and faster. Then, an experiment conducted to verify the model and to assess the effects of the force fields is presented.

CHI 2005, April 2–7, 2005, Portland, Oregon, USA.

[1] In what follows, selection time refers to the time needed for the third activity exclusively, i.e. ignoring the time spent for the search and decision making activities.

MODELING SELECTION IN PULL-DOWN MENUS

Fitts' law is a robust model of human psychomotor performance widely used in human computer interaction to assess input devices and GUI designs. Selection time of interface widgets can be modeled using Fitts' Law (or variations thereof, such as the Shannon formulation shown in Equation 1), which states that the selection time T is proportional to the logarithm of the distance D to the target divided by target width W (see MacKenzie [17] for a detailed discussion).

$$T = a + b\,log_2\left(\frac{D}{W} + 1\right) \quad (1)$$

The logarithmic term in Equation 1 is commonly referred to as the *Index of Difficulty* (ID), carrying the unit of bits, and is a measurement of how difficult the selection task is. More difficult tasks having higher IDs. The parameters a and b are determined empirically through linear regression.

In a typical Fitts' law task, e.g. using a mouse to steer a cursor to a squared target, the shape of the motion trajectory is deemed irrelevant and is ignored. But if the trajectory is constrained, i.e. the cursor has to be moved along a predefined straight line, the task is better modeled by the steering law [1]. Such a linear steering task can be described as the task of steering the cursor through a tunnel (see Figure 1, A) without crossing the tunnel walls.

Accot and Zhai [1] found that the time needed to cover a distance d in a tunnel of width w, without crossing the tunnel walls, is given by the equation:

$$T = a + b\frac{d}{w} \quad (2)$$

where a and b are determined empirically through linear regression. Contrary to Fitts' law, the steering law is not logarithmic, and the IDs for linear steering tasks are only linked to the distance-width fraction. Accot and Zhai conducted their initial experiments using a graphical tablet and its stylus. Later studies [2, 9] have shown that the law is also applicable to other input devices over a broad range of IDs.

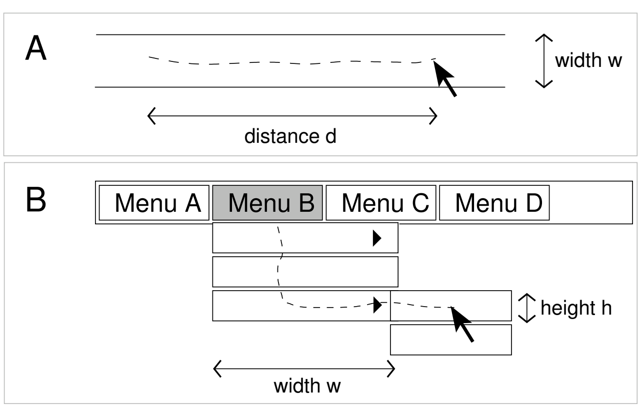

Figure 1. A: linear steering task, B: a vertical and a horizontal linear steering task is needed to select the first sub-menu item.

As they introduced the steering law, Accot and Zhai hypothesized that the law could be used to model selection times in cascading pull-down menus. If the menu selection task is viewed as a compound of two or more linear steering tasks (Figure 1, B), Accot and Zhai suggested that the time T_n needed to select the n^{th} sub-menu in a hierarchical menu might be approximated by solving:

$$T_n = \overbrace{a + b\frac{nh}{w}}^{Vertical} + \overbrace{a + b\frac{w}{h}}^{Horizontal} \quad (3)$$

$$= 2a + b(\frac{n}{x} + x) \quad with : x = \frac{w}{h} \quad (4)$$

where h is the height of the menu items and w is the width of the parent menu.

Using the steering law in this sense, they assumed that horizontal steering and vertical steering are driven by the same law. Accot and Zhai also pointed out, that, if driven by the same law, the coefficients a and b are likely to be different for horizontal steering and vertical steering. Dennerlein et al. [9] showed that the steering law is applicable to both horizontal and vertical steering tasks, and that the coefficients a and b are indeed different for the two tasks: vertical movements took longer than horizontal ones. [2]

While there are no limits regarding the direction of movement, there is a limit regarding the width of the path. Accot and Zhai [1] report that the law loses its predictability power as the width exceeds an upper bound limit of 70 pixels on a 19-inch monitor with 1280×1024 pixels resolution. This limitation certainly reduces the practicability of the law when modeling selection times in cascading pull-down menus, since the vertical motions, in most cases, are done through tunnels wider than 70 pixels.

If, instead, the vertical motions in a menu selection task are viewed as Fitts' law tasks, i.e., selection tasks with the accuracy constraint collinear to movement, the upper bound limit of the steering law is avoided. The menu selection task depicted in Figure 2 can be seen as a compound of three vertical Fitts' law tasks and two horizontal steering tasks. To model the total selection time, we need an ID_T which describes the difficulty of the compound task. Our hypothesis is, that this ID_T is obtained by adding the sum of all IDs for the required vertical movements, ID_V, to the sum of all IDs for the required horizontal movements, ID_H.

The IDs for each separate vertical task are calculated according to Fitts' law, using the menu item height h and item position p to form the distance D. The accuracy constraint is item height h.

[2]Dennerlein et al. [9] proposed that the differences in the joint kinematics required to perform vertical and horizontal movements may have been the source for the longer movement times in the vertical direction. The difference might also be a device dependent phenomenon. The pointing device used in the experiments was physically connected to a limited work space and did not allow the user to pick it up for repositioning. This might also have influenced the selection times differently according to movement direction.

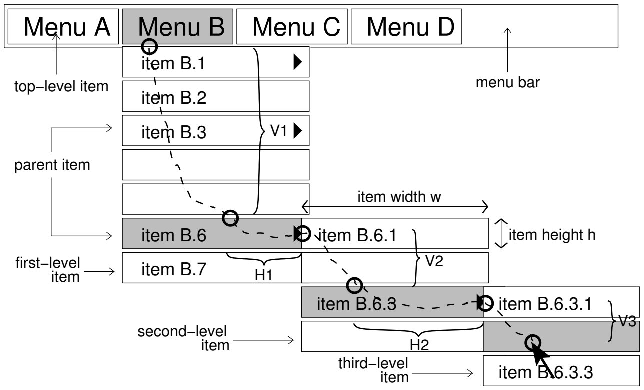

Figure 2. A cascading pull-down menu. Schematic representation of a selection of the second third-level menu item in the second top-level menu (Menu B). The selection task is a compound of three vertical Fitts' law tasks (V1, V2 and V3) and two horizontal steering tasks (H1 and H2).

The difficulty to select item B.p at position p in the first sub-menu depicted in Figure 2, is obtained through:

$$log_2\left(\frac{ph}{h}+1\right) = log_2\,(p+1) \tag{5}$$

while ID_{V_m}, which describes the difficulty of all vertical tasks needed to select a menu item in the third sub-menu ($m = 3$) of the menu hierarchy in Figure 2, is obtained through:

$$ID_{V_m} = \sum_{j=1}^{m} log_2\,(p_j+1) \tag{6}$$

where p_j is the position number of the target item for the Fitts' law task in the j^{th} sub-menu.

The IDs for each separate horizontal steering task are calculated according to the steering law. The length of the two steering tasks (H1 in parent item B.6 and H2 in parent item B.6.3, Figure 2) are added and the sum is divided by the menu item height h. The exact length of each steering task cannot be known a priori, but it seems reasonable to assume that on average, the length will not exceed half that of the menu width w.[3]

Equation 7 is used to calculate the total difficulty of all steering tasks needed to select an item in the m^{th} sub-menu:

$$ID_{H_m} = \sum_{j=1}^{m-1} \frac{0.5\ w_j}{h} \tag{7}$$

where w_j is the width of the j^{th} sub-menu.

Equation 6 and Equation 7 are used to obtain the total ID for the compound task:

$$ID_T = ID_{V_m} + ID_{H_m} \tag{8}$$

and finally, the time $T_{n,m}$ needed to select the n^{th} item in the m^{th} sub-menu can be approximated by calculating:

$$T_{n,m} = a + b\ ID_T \tag{9}$$

It is important to note the difference between how the steering law is used in this model and how it has been used in previous GUI studies [1–3, 9]. These studies have used the steering law to model cursor steering time of *accurate* and *error free* task trials. In the reported experiments, the participants were encouraged to balance speed with accuracy, and all trials where the cursor crossed a tunnel wall were excluded from the data analysis. In the case of menu selection, the accuracy of the horizontal steering tasks inside parent items is less important. Crossing the walls will not result in an error, and the importance of movement accuracy increases at the end of the tunnel, when the cursor is about to enter the open sub-menu.

IMPROVING SELECTION IN PULL-DOWN MENUS

Most techniques introduced to improve selection times in cascading pull-down menus have focused on the selection of first-level items. Shorter selection times have been reached by either decreasing the distance to the menu items, or by increasing the size of the menu item. A Split menu [10, 21] adapts to user behavior and relocates the menu items according to usage. Frequently selected items are moved into the top split of the menu and seldom selected items are pushed downward, i.e. the distance to an item depends on selection probability. Walker et al. [23] suggested a progressive vertical enlargement of menu items related to their distance from the top-menu, but without significant results. They also backed up menu items with impenetrable borders to prevent an overshot of the approaching cursor movement and concluded that *"systems that maximize the percentage of menu items with borders will have a decided advantage over other menu systems"*. However, in a pull-down menu, the number of candidate items for a border is limited.

Kobayashi and Igarashi [14] showed that a gesture based selection approach can reduce selection times of menu items in the second to fifth hierarchical level of a cascading pull-down menu. Their menu system analyzed the direction of the cursor movements and distinguished between horizontal and vertical movements. A horizontal cursor movement to the right in a parent item opened up a sub-menu which, like a pop-up menu, appeared directly at the cursor position.

[3] If the user knows that the wanted menu item is located in a sub-menu, the vertical movement toward the parent-item which opens the sub-menu, will be directed toward the right hand side, i.e. the vertical motion is more likely to be diagonal. The angle of the diagonal determines the actual length of the horizontal steering task which follows. The angle in turn, is dependent on the distance needed to be covered in order to reach the parent-item and the menu width. During a long task in a wide menu, more focus can be put on reaching the right hand side of the parent item than in a narrow menu or during a short task.

The long horizontal movement trajectories normally required to steer into a sub-menu were shortened in this way. A leftward motion closed an open sub-menu. The reported selection times however include both search time and decision time, and the new gesture technique was reported to influence the menu navigation negatively for some users, making it less fluent.

The approach assessed in this paper explores the possibility to optimize selection in cascading pull-down menus by partially overruling the user's control of the screen cursor. The main advantages of this approach is that no new interaction technique has to be learned, the visual structure or layout of the menus are unchanged and the approach is also applicable to pull-down menus with more than one menu-level.

Force Fields

In the case of an indirect pointing device, the control-display (C-D) gain maps the distance the device has been moved to a corresponding motion of the screen cursor. With a low C-D gain setting, a large device movement moves the cursor a moderate distance. With a higher C-D gain setting, a device movement of equal distance moves the cursor a greater distance, i.e. the difference is visually perceived as a change in cursor speed. The implications of C-D gain settings on GUI usage is a well studied domain [5, 13, 18, 22], and a dynamic adaptation of the gain setting according to different variables (e.g. the position of the cursor, the distance from the cursor position to target, cursor velocity, movement direction, see Blanch et al. [6] for a detailed overview) has been used to facilitate target acquisition in various ways.

A similar approach is to use a warping algorithm according to which small cursor displacements are made. Whereas a change of the C-D gain setting only results in changes collinear with the movement, the warping method also allows for sideway displacements. In this way, user cursor control can be overruled by software to both influence speed and direction of a cursor movement. By modifying the visual motion of the cursor in this way, a virtual force can be produced which pushes the cursor toward a certain coordinate of the screen. When the user sees how the cursor is attracted in one direction, the user also has the illusion as of to "feel" the attracting force when the input device is moved. This virtual force effect is often referred to as pseudo-haptics, or simulated force-feedback. Pseudo-haptic effects have been used to simulate various textures felt and differentiated by the users [15], and to simulate friction and stiffness [16]. The warping method has also been proposed to facilitate the user during different GUI tasks, e.g. point and click tasks [8, 19], accurate cursor positioning in drawing tasks [12] and general GUI navigation [4].

In our approach to improve navigation and selection in cascading pull-down menus, "force fields", within which the cursor is warped, are placed over menu items in order to help the user steering the cursor. We used two types of fields, one associated with a force point, for parent items, and one without a force point, a directional field, for non-parent items. The arrangement of the fields is depicted in Figure 3.

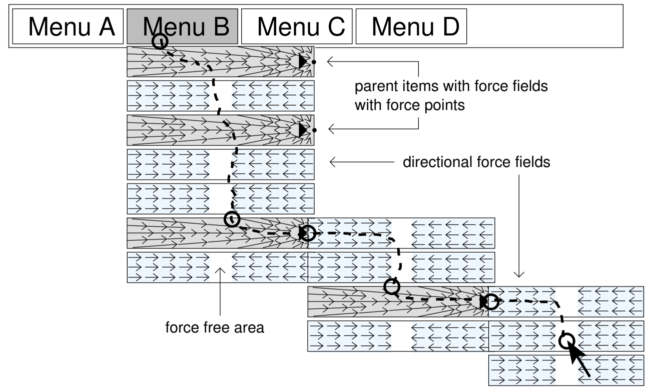

Figure 3. Cascading pull-down menu with force fields.

Inside a field with a force point, the cursor is attracted toward the force point by warping the cursor along both the horizontal and the vertical axis. In a parent item, the force field helps the user to steer the cursor within the menu item to the right by deflating vertical and leftward cursor movements and by reinforcing rightward movements. In non-parent items, overlayed with directional fields, the cursor is only warped along the horizontal axis, toward the middle of the menu item, in one direction, either to the right or to the left.

The software which implements the force fields tracks the current position of the cursor by intercepting mouse motion events generated by the pointing device. For each mouse motion event registered inside a force field, a new cursor position is calculated, and then the cursor is warped to the new position. The warping algorithm is based on real vector arithmetic. Screen coordinates for a new cursor position inside a field with a force point are calculated according to the following formula:

$$\boldsymbol{n} = \boldsymbol{a} + s \cdot \|\boldsymbol{a} - \boldsymbol{p}\| \cdot \frac{\boldsymbol{f} - \boldsymbol{a}}{\|\boldsymbol{f} - \boldsymbol{a}\|} \tag{10}$$

where:

$\boldsymbol{n} = (n_x, n_y)$ = (new) cursor position after applied force,
$\boldsymbol{a} = (a_x, a_y)$ = (active) cursor position after the last mouse motion,
$\boldsymbol{p} = (p_x, p_y)$ = (previous) cursor position, before last mouse motion,
$\boldsymbol{f} = (f_x, f_y)$ = position of the force point and,
s = strength of the force field.

The resulting reals (n_x and n_y coordinates) are rounded to the closest integers, allowing a displacement of the cursor position in the integer based screen coordinate system.

The cursor displacements inside a directional field are calculated in a similar way, but should not result in any changes in the vertical direction, therefore the following formula is used:

$$\boldsymbol{n} = \boldsymbol{a} + s \cdot \|\boldsymbol{a} - \boldsymbol{p}\| \cdot (\pm 1, 0) \tag{11}$$

Since the screen coordinates are counted from the top left corner, 1 is used for fields directed to the right and -1 for fields directed to the left (0 is used for the vertical direction).

Figure 4. Apparatus used in the experiment. A: track point, B: touch pad, C: optical mouse, D: selection button for track point and touch pad.

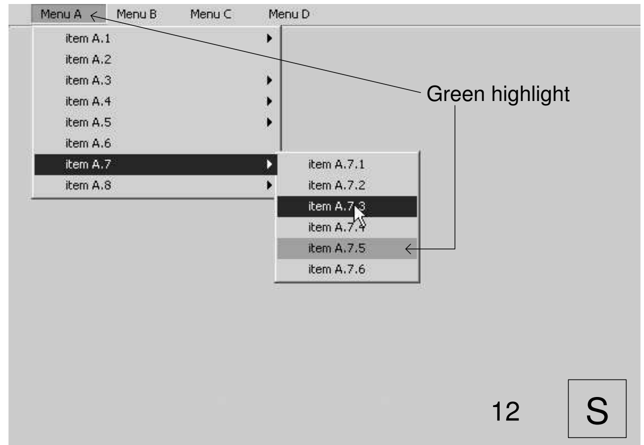

Figure 5. Upper left quarter of a screen dump showing the software used in the experiment (bottom right: trial counter and start box).

Equation 10 and Equation 11 imply that in settings with high enough strength or in situations with a large enough cursor movement toward the force point (or in the direction of the directional field), the force field can cause the cursor to overshoot beyond the force point (or outside the field). Too high a strength also makes it impossible to leave the field in another direction than going through the force point (or in the direction of a directional field). Pilot experiments showed that a strength of 0.8 pixels for fields with the size of a menu item is adequate to avoid these problems. The pilot experiments also showed that a form of escape functionality is needed, which helps the user if the cursor gets inside a field warping the cursor in an undesired direction. Therefore, after that the software has registered six consecutive mouse motions away from the force point (or in the direction opposite to the force inside a directional field), the force is turned off to allow for an easy escape from the field. The force is reactivated as soon as the cursor is moved toward the force point (or in the direction of a directional field).

EXPERIMENT

A controlled experiment was conducted to evaluate the benefits of force fields in cascading pull-down menus and to investigate if the suggested model can be used to predict selection times.

Apparatus and Participants

The experiment was conducted on a notebook running Windows XP with a 15-inch TFT monitor. A full-screen color mode with a 1024×768 resolution was used. Three input devices were used: a track point, a touch pad and a conventional optical mouse (Figure 4). All default system settings for the three devices were used.

Eighteen volunteers (9 male, 9 female, the age ranged from 17 to 54 years, with an average of 26 years, $SD = 10.5$ years) participated in the experiment. All participants had normal or corrected to normal sight. The participants performed the test using their preferred hand (one was left-handed). All participants were experienced computer users, using a mouse on a daily basis. The experience in using a track point and a touch pad varied. No participant used a track point on a daily basis. Seven were infrequent (less than once a week) users and eleven had never used a track point before the experiment. Eleven participants were infrequent touch pad users and seven had no experience with a touch pad prior to the experiment.

Task

In the experiment, six second-level and three third-level menu items had to be selected. Since we were only interested in the time a subject needed to steer the cursor to the right item and select it, we needed to cancel out the portion of the selection time the participant would need to localize the target item. To minimize this search time, we used the following trial procedure, which guided the participant to the target items by highlighting key items green.

A trial was started by a click in the start box located near the center of the screen (labeled S in the partial screen dump in Figure 5). One of the four top-level items in the menu bar was highlighted green. After a click in the green top-level item, a first-level menu opened up. One of the parent items in the first-level menu was marked by a green highlighting. As the cursor entered the marked first-level item a second-level menu opened up after a slight delay. The second-level menu contained one green menu item. Either this item was the target item or it was a parent item, and opened up a third-level menu which contained the target item. A click in the target item ended the timing, which started as the cursor exited the green top-level item. By using this highlighting procedure to help the participants to localize the target item, the sampled times consisted of only movement times, excluding any search times. If the wrong item was clicked, an error message was displayed, the trial was logged as an error trial, and the participant started a new trial from the start box.

Task	ID_V	ID_H	ID_T
1	5.32	10.84	16.16
2	2.58	5.97	8.55
3	5.58	5.97	11.55
4	7.07	5.76	12.83
5	7.78	10.26	18.04
6	4.58	3.5	8.08
7	5.32	3.5	8.82
8	5.61	2.87	8.48
9	7.22	8.31	15.53

Table 1. Task difficulty, ID_V: vertical direction, ID_H: horizontal direction, ID_T: Total task difficulty.

The behavior of the menus was the same as in most Windows applications, i.e.:

- while the cursor was inside a menu item, the item was highlighted blue,
- the color turned back to gray (or green for marked items) when the cursor exited an item,
- a parent item was identified as such by a black triangle near its right hand side border,
- the sub-menu of a parent item was displayed with a slight delay after the cursor entered the parent item,
- parent items with opened sub-menus stayed highlighted as long as its sub-menu was open,
- a click in a parent item instantly opened up its sub-menu,
- a click outside an open menu hierarchy closed it,
- the menus only reacted to clicks with the left mouse button, and
- after a click in one top-level item the menu bar was active and another top-level item could be activated by only moving the mouse inside it (no click was needed).

Contrary to pull-down menus in most Windows applications, when the mouse was dragged (i.e. the selection button was pressed inside one menu item and released inside another menu item) the selection was not valid and the trial was logged as an error trial. The nine different tasks used in the experiment are depicted in Figure 6. Four top-level items, $Menu\ A$, $Menu\ B$, $Menu\ C$ and $Menu\ D$ were used to open up different sub-menus. Three tasks were started from $Menu\ A$, two tasks were started from $Menu\ B$, $Menu\ C$ and $Menu\ D$ respectively. Each menu item was 19 pixels high. The width of the menu items were chosen based on the measurement of 240 first-level menus from 30 different Windows applications (average width 195 pixels, about 6 cm or 2.37 inches on a 15-inch monitor with 1024x768 resolution) and ranged from 83 pixels to 227 pixels. The IDs for each task, listed in Table 1, were calculated using Equation 6, 7 and 8.

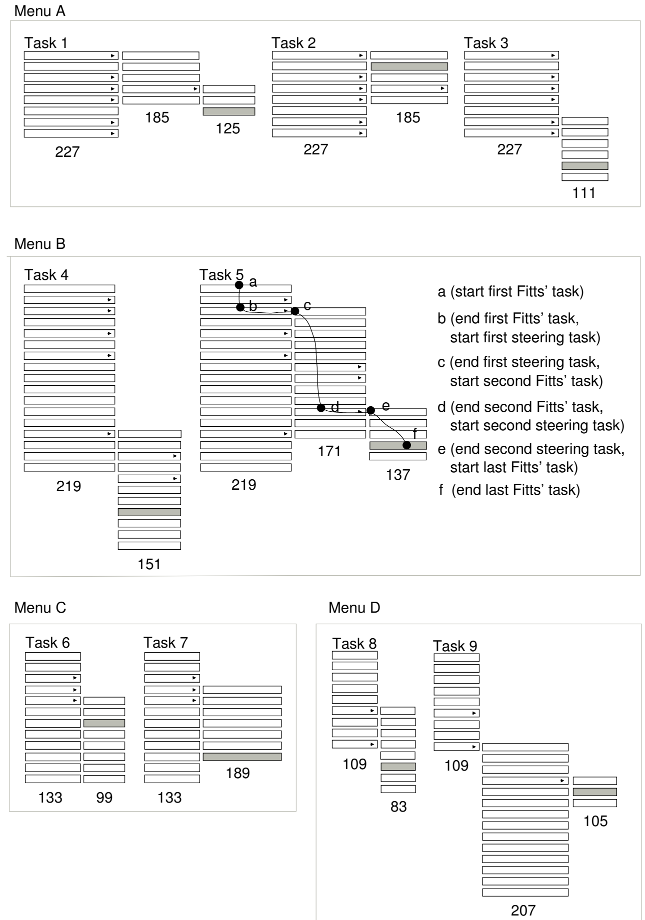

Figure 6. Schematic representation of the nine tasks used in the experiment. Four top-level menus were used, $Menu\ A$ to D. Target items are pictured with dark background. Item widths are given in pixels below each sub-menu. Item height was always 19 pixels. Start and end points of the three Fitts' law tasks and the two steering tasks needed to complete $Task\ 5$ are labeled a through f.

Experimental Design and Collected Data

Each participant performed three test sessions, one with each device. The order of testing of the three devices was counterbalanced between the 18 participants. There was at least a one hour long break between each session. If a device was new to a participant, the participant was instructed about its functionality and how to use it in the best way. Before the test began, all participants were allowed to have as many practice trials as they needed to get used to the device and to gain sufficient practical skill.

A test session consisted of 180 trials which were divided into 5 blocks. Within one block, all nine tasks were performed twice in the enhanced menu type and in the standard menu type. The order was randomized. After each block was completed, a recess screen was shown, and the subject could take a short break if desired. A session lasted on average for 30 minutes. The participants were not informed about the force fields and their functionality. The force fields were invisible.

The total number of trials in the experiment can be computed as follows:

18 subjects × 3 devices × 5 blocks × 9 tasks × 2 menu types × 2 trials per task/menu type combination = 9720 trials

The total selection time for each trial was measured (in milliseconds), timing started as the cursor exited the top-level item and ended with a click in the target item.

RESULTS, ANALYSES AND DISCUSSION

First, the results regarding force enhancement vs. standard menus are analysed, then, the modeling aspects are analysed.

A total of 9261 trials were used for the analysis. Excluded were 459 (4.7%) invalid trials (wrong menu item selected, and/or the mouse was dragged). An ANOVA with number of logged error trials as dependent variable and subject, device, menu type (standard or force enhanced), task (1 to 9) and block as independent variables, showed a significant main effect for subject ($F_{17,34} = 3.33, p < .01$), and a significant device×subject interaction ($F_{34,32} = 7.49, p < .001$). This indicated that there were differences between the devices, as well as between the subjects. Most errors were made with the touch pad, 180 logged errors. 135 errors were made with the track point, and 144 errors with the mouse. No further analysis concerning the error rate was made.

Force Enhancement vs. Standard Menus

The following analyses are given in five groups according to device and user experience: (AM) mouse all subjects, (ITR) infrequent track point users, (NTR) novice track point users, (ITO) infrequent touch pad users and (NTO) novice touch pad users.

Effect of practice

Five separate ANOVAs, one for each group, with selection time as dependent variable and block number, menu type (standard or force), task and subject as independent variables, showed significant main effects of block number, indicating learning effects. Since there were no significant block×menu type interactions, we conclude that the learning effect relates to the device and the menu selection exercise, not to the two menu types. Performance stabilized and did not vary significantly between the last four blocks in the AM, ITR and ITO groups ($F_{3,52} = .41, p > .05$, $F_{3,18} = 2.58, p > .05$ and $F_{3,30} = 2.77, p > .05$) and between the last three blocks in the NTR and NTO groups ($F_{2,20} = 2.23, p > .05$ and $F_{2,12} = 1.2, p > .05$). Therefore, the following analyses are based on data from these blocks only.

Selection time

The force enhanced menus were faster than standard menus in all device-experience groups (see Table 2). The novice touch pad users profited the most from the force fields, on average 1059 ms per menu selection (30.6%), followed by the novice track point users (838 ms, 18.3%). Infrequent track point users profited the least. On average, the infrequent track point users were 362 ms (11.2%) faster in the force enhanced menus than in the standard menus.

	Standard	Force	Δ	Δin%	F-statistic
AM	2480	2060	420	16.9	$F_{1,17} = 67.4$
ITR	3234	2872	362	11.2	$F_{1,6} = 13.4$
NTR	4584	3746	838	18.3	$F_{1,10} = 98.6$
ITO	3387	2909	478	14.1	$F_{1,10} = 25.2$
NTO	4515	3456	1059	30.6	$F_{1,6} = 22.6$

Table 2. Mean selection times (ms) and F-statistics ($p < .001$) for the standard menu and the force enhanced menu for the five device-experience groups.

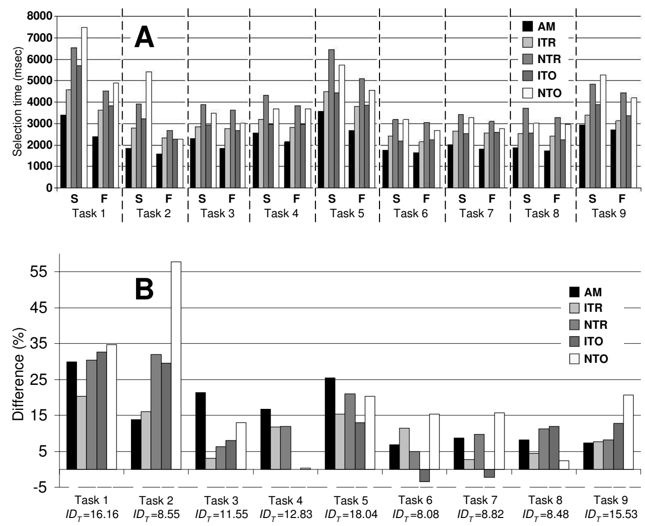

Figure 7. A: Comparison of the selection times for all device-experience groups for both menu types for each task. B: Difference between standard menus and force enhanced menus, for all device-experience groups in all tasks in percent.

Only two of the participants clicked, in a somewhat systematical way (in about 60% of the trials), on the parent items to instantly open up the sub-menus. Both participants were novice track point users and novice touch pad users and the strategy was only applied when using the mouse. A separate analysis of the selection times for these two participants did not reveal any significant differences between trials with clicks in parent items and trials without clicks.

All five ANOVAs showed significant menu type× task number interactions, indicating that the force fields influenced the selection times differently, depending on task (see Figure 7). Except for the touch pad groups (ITO and NTO), the force enhanced menus were faster than the standard menus in all tasks. In the ITO group, the standard menus were slightly faster in $Task$ 6 (75 ms, 3.4%) and in $Task$ 7 (57 ms, 2.2%). In $Task$ 4, there were no differences between the menu types in group ITO and in group NTO. In all but the NTO group, the difference between the standard menus and the force enhanced menus was greatest in $Task$ 1, in group

NTO the greatest difference was in $Task\ 2$. A closer inspection of the cursor trajectories needed to complete $Task\ 1$ and $Task\ 2$ explains why the force fields were more helpful in these tasks than in the other tasks. The two tasks require rather long horizontal movements inside the first menu item in the first sub-menu (item A.1 in Figure 5, cf. Figure 6). If, during the sequence of device manipulations required to steer the cursor to the right into the second sub-menu, just a minor incorrect device manipulation toward the top of the screen is made, the cursor leaves the sub-menu and enters the menu bar. As soon as the cursor enters a top-level item inside the menu bar, the top-level item becomes active, and a new first-level menu will instantly open up. In order to reach the target item and to complete the task, the cursor has to be moved back to the left, inside the menu bar to re-open $Menu\ A$ for a new attempt. Therefore, one inaccurate device manipulation can result in a substantial loss of time. However, with the force fields, inaccurate device manipulations toward the menu bar are weakened by the warping algorithm, which replaces the cursor inside the menu item, which in turn makes it easier to reach the sub-menu to the right.

From the fact that $Task\ 1$ and $Task\ 2$ caused considerable problems, the implications for menu design are clear: in a wide sub-menu, the placement of a parent item adjacent to the menu bar should be avoided in order to provide fast and easy menu navigation.

If $Task\ 1$ and $Task\ 2$ are excluded from the selection time comparison, the mean differences between the menu types are reduced to 13.5%, 9.4%, 10.6%, 5.9% and 12.6% for the AM, ITR, NTR, ITO and NTO groups, respectively. Further analyses of the benefit of the force fields, based on the previously suggested model of selection times in cascading pull-down menus, are made in next section.

Model Fit

To investigate if the suggested model can be used to predict selection times in cascading pull-down menus, ten separate linear regression analyses were made. One for each device-experience/menu type combination, using the previously calculated task difficulty, ID_T, as independent variable. $Task\ 1$ and $Task\ 2$ were not included in the regression analyses since the model does not take the above discussed problem with the menu bar into account when the task difficulty is calculated.

In each of the ten regressions, the data fit the model equation (Equation 9) with an r^2 value of 0.904 or above. This shows that the proposed model can be used to model selection times in cascading pull-down menus for the tested devices and user groups and that the menu selection task indeed can be described as a compound of vertical and horizontal sub-tasks.

The best fit was for the AM group, with an r^2 value of 0.974 followed by the ITO group ($r^2 = 0.955$). Also selection times in force enhanced menus correlated highly with ID_Ts calculated according to the model. The regression equations for all device-experience group/menu type combinations are listed in Table 3.

Group	Menu type	Model	r^2
AM	standard	$T = 420 + 169 \cdot ID_T$	0.974
	force	$T = 752 + 112 \cdot ID_T$	0.904
ITR	standard	$T = 947 + 180 \cdot ID_T$	0.917
	force	$T = 1199 + 136 \cdot ID_T$	0.929
NTR	standard	$T = 977 + 276 \cdot ID_T$	0.905
	force	$T = 1451 + 195 \cdot ID_T$	0.976
ITO	standard	$T = 622 + 206 \cdot ID_T$	0.955
	force	$T = 1044 + 153 \cdot ID_T$	0.966
NTO	standard	$T = 699 + 274 \cdot ID_T$	0.911
	force	$T = 1148 + 190 \cdot ID_T$	0.953

Table 3. Linear regression equations for the five devices-experience groups.

It is interesting that the force condition, in all except from the AM group, fit the model better than the standard condition. One possible explanation for this is the level of skill with which the pointing devices were operated. In all but the AM group, the practical skill was low to intermediate. The inaccuracy and irresoluteness with which infrequent and novice users operate pointing devices might be more sensitive to the influence of the force fields than the precise and distinct device manipulations made by highly skilled users. If this is true, then, the cursor paths made by the low skilled users in the force conditions would to a greater extent, follow the paths supported by the force fields (i.e. in non-parent items straight vertical paths, in parent items straight horizontal rightward paths, starting from the horizontal middle of the item, cf. Figure 3), than the cursor path made by the high skill users during force influence. Since the paths accounted for in the model are assumed to be straight, the paths supported by the force fields better match the model paths, which results in a better model fit.

Future deeper analyses of the exact cursor paths taken during selection, and of what the vertical and horizontal movements actually look like will help to further refine the model. It is questionable whether or not the horizontal movements on average starts from the half of the parent items, i.e. if they are best modeled by using a constant of 0.5 when calculating ID_H.

Task difficulty vs. benefit of force fields

The results show too great differences between the five device-experience groups as to draw any general conclusions regarding the relationship between task difficulty and the benefit of the force fields. Four vague indications can however be identified which point to a possible relationship between task difficulty and force benefit:

1. In all device-experience groups, the task with the highest ID_T ($Task\ 5$) profited the most from the force fields, which can be seen in Figure 7 and in the regression plots for each group, Figure 8.

2. If the relative difference in selection times between the two menu types in each task is computed over all groups, some regularities can be seen. The two tasks with the hig-

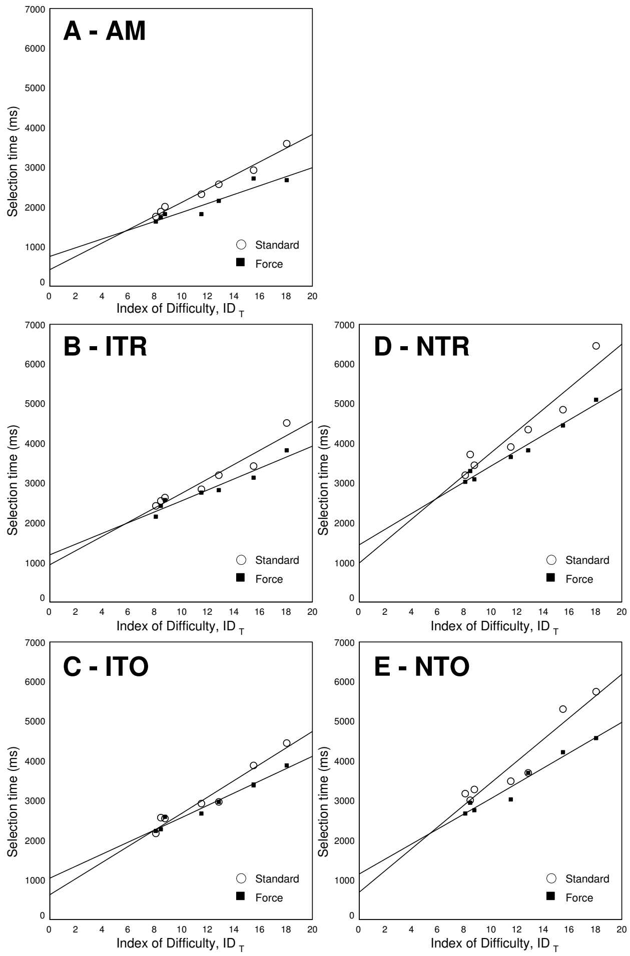

Figure 8. Regression plots for all device-experience groups, for both menu types.

hest ID_Ts benefited the most from the force fields, and least benefited the task with the lowest ID_T. But, as can be seen in Table 4, there are two mismatches in the benefit ranking. At $Rank$ 3 comes $Task$ 3 with the fourth highest ID_T, instead of $Task$ 4. Also $Task$ 8 with lower ID_T benefited more from the force fields than $Task$ 7 which has a higher ID_T.

3. In the linear regression equations for all device-experience groups, the intercept coefficient, a, is much higher for the force enhanced menu type than for the standard type (see Table 3.

4. In all device-experience groups, but the ITO group, the two regression lines intersect at a point between 5.34 and 5.85 ID_T (at 7.96 ID_T for the ITO group)(see Figure 8).

The last two regularities concerning a relationship between task difficulty and force benefit are particularly interesting since they imply that there is a low lower bound of task difficulty, beyond which, the force fields impede during menu navigation. Further experiments with tasks over a broader range of difficulty, and where the same ID_Ts are based on several different ID_V-ID_H combinations, are required for more precise conclusions concerning a task difficulty-force benefit relationship and a possible lower bound.

Rank	1	2	3	4	5	6	7
Δin%	19.00	11.30	10.31	8.16	7.60	7.33	6.99
ID_T	18.05	15.53	11.55	12.83	8.48	8.82	8.06
Task	5	9	3	4	8	7	6

Table 4. Difference between standard menus and force enhanced menus for each task, calculated over all devices-experience groups.

Participants' Subjective Impressions

In the experiment the participants were not informed about the force fields, and the fields were not visually presented, nevertheless the users were assisted by them. When debriefed after the last test session, only three of the 18 participants stated that they had noticed a change in the behavior of the pointing device. All three participants had noticed the changed behavior during the mouse session. Two could not specify what it was and had not seen any behavioral patterns. One described the change as a variation of cursor speed.

Considering the presented experimental results and the subjective impressions given, it seems reasonable to assume that the force fields could be even more helpful if the user is aware of them and knows about their functionality, and thus being able to actively take advantage of them.

CONCLUSION

This paper has focused on how selection times in cascading pull-down menus can be modeled and how the selection times can be shortened by using a simple cursor warping algorithm to implement "force fields", which helps the user steering the cursor during the selection task. The force fields and the proposed model of selection times, based on a combination of Fitts' law and the steering law, has been evaluated through a controlled user experiment, in which the users selected menu items in standard menus and in menus enhanced with force fields. Even though the users in the experiment did not know about the force fields, they benefited to a great extent from the fields, selections in force enhanced menus were on average 18% faster than selections in standard menus.

The results from the experiment also showed that a menu selection task can indeed be seen as a compound of several separate vertical and horizontal tasks, as accounted for in the proposed model. By calculating a separate index of difficulty for each one of the vertical and horizontal tasks, a total index of difficulty for the compound task can be determined. This index of difficulty showed to be a robust predictor of the time it takes for a user to complete the modeled menu selection task, i.e., a linear relationship was found. Moreover, the model was shown to hold for all three devices (mouse, track point and touch pad) used in the experiment.

Beside these unambiguous results, the analyses also showed other less clear results, e.g., the relationship between the index of difficulty of a menu selection task and the benefit of the force fields. During selections in tasks with high index of difficulty, the users tended to profit more from the fields than during selections in tasks with a low index of difficulty. However, no general and for all tested devices applicable regularity was found. Further experiments are required to sort out this ambiguity. Similar experiments will also be helpful in order to optimize the current model and to gain a deeper understanding of user behavior during selection in cascading pull-down menus.

REFERENCES

1. Accot, J. and Zhai, S. Beyond Fitts' Law: Models for Trajectory-Based HCI Tasks. In *Proc. CHI 1997*, ACM Press (1997), 295–302.
2. Accot, J. and Zhai, S. Performance Evaluation of Input Devices in Trajectory-based Tasks: An Application of The Steering Law. In *Proc. CHI 1998*, ACM Press (1998), 466–472.
3. Accot, J. and Zhai, S. Scale Effects in Steering Law Tasks. In *Proc. CHI 2001*, ACM Press (2001), 1–8.
4. Ahlström, D., Hitz, M. and Leitner, G. Improving Mouse Navigation - A Walk Through the "Hilly Screen Landscape". In *Proc. DSV-IS 2002*, Springer-Verlag (2002), 185–195.
5. Arnaut, L.Y. and Greenstein, J.S. Is Display/Control Gain a Useful Metric for Optimizing an Interface? *Human Factors 32(6)*, (1990), 651–663.
6. Blanch, R., Guiard, Y. and Beaudouin-Lafon, M. Semantic Pointing: Improving Target Acquisition with Control-Display Ratio Adaption. In *Proc. CHI 2004*, ACM Press (2004), 519–526.
7. Callahan, J., Hopkins, D., Weiser, M. and Shneiderman, B. An Empirical Comparison of Pie vs. Linear Menus. In *Proc. CHI 1988*, ACM Press (1988), 95–100.
8. Cockburn, A. and Firth, A. Improving the acquisition of small targets. In *Proc. HCI 2003*, (2003), 181–196.
9. Dennerlein, J.T., Martin, D.B. and Hasser, C. Force-Feedback Improves Performance For Steering and Combined Steering-Targeting Tasks. In *Proc. CHI 2000*, ACM Press (2000), 423–429.
10. Findlater, L. and McGrenere, J. A comparison of Static, Adaptive, and Adaptable Menus. In *Proc. CHI 2004*, ACM Press (2004), 89–96.
11. Fitts, P.M. The Information Capacity of Human Motor Systems in Controlling the Amplitude of a Movement. *Journal of Experimental Psychology, 47*, (1954), 381–391.
12. Foley, J.D., van Dam, A., Feiner, S.K. and Hughes, J.F. Computer Graphics: Principles and Practice. 2nd Edition. Addison-Wesley, Reading, MA, USA, 1994.
13. Jellinek, Herbert D. and Card, Stuart K. Powermice and User Performance. In *Proc. CHI 1990* ACM Press (1990), 213–220.
14. Kobayashi, M. and Igarashi, T. Considering the Direction of Cursor Movements for Efficient Traversal of Cascading Menus. In *Proc. UIST 2003*, ACM Press (2003), 91–94.
15. Lécuyer, A., Burkhardt, J-M. and Etienne, L. Feeling Bumps and Holes without a Haptic Interface: the Perception of Pseudo-Haptic Textures. In *Proc. CHI 2004*, ACM Press (2004), 239–246.
16. Lécuyer, A., Coquillart, S., Kheddar, A., Richard, P., and Coiffet, P. Pseudo-Haptic Feedback: Can Isometric Input Devices Simulate Force Feedback? *IEEE VR*(2000).
17. MacKenzie, S. I. Fitts' Law as a Research and Design Tool in Human-Computer Interaction. *Human-Computer Interaction, 7*, (1992), 91–139.
18. MacKenzie, S. I. and Riddersma, S. Effects of Output Display and Control-Display Gain on Human Performance in Interactive Systems. *Behaviour & Information Technology, 13(5)*, (1994), 328–337.
19. van Mensvoort, K. What You See Is What You Feel: exploiting the dominance of the visual over the haptic domain to simulate force-feedback with cursor displacements. In *Proc. DIS 2002*, ACM Press (2002), 345–348.
20. Norman, K. The Psychology of Menu Selection: Designing Cognitive Control at the Human/Computer Interface. Ablex Publishing Corp.
21. Sears, A. and Shneiderman, B. Split menus: effectively using selection frequency to organize menus. *ACM Trans. Comput.-Hum. Interact., 1(1)*, ACM Press (1994), 27–51.
22. Tränkle, U. and Deutschmann, D. Factors Influencing Speed and Precision of Cursor Positioning Using a Mouse. *Ergonomics, 34(2)*, Taylor & Francis Ltd. (1991), 161–174.
23. Walker, N., Smelcer, J.B. and Nilsen, E. Optimizing Speed and Accuracy of Menu Selection: a Comparison of Selection Times from Walking and Pull-Down Menus. *Int. J. Man-Machine Studies, 35*, (1991), 871–890.