An Evaluation of Sticky and Force Enhanced Targets in Multi Target Situations

David Ahlström, Martin Hitz, and Gerhard Leitner
Department of Informatics-Systems
Klagenfurt University
Universitätstrasse 65, A-9020 Klagenfurt, Austria
{david | martin | gerd}@isys.uni-klu.ac.at

ABSTRACT
In this paper we explore the usage of “force fields” in order to facilitate the computer user during pointing tasks. The first study shows that pointing time can be reduced by enhancing a pointing target with an invisible force field that warps the screen cursor toward the target center. The application of force fields is further supported in that we show how performance of force enhanced pointing can be predicted by using Fitts’ law and a force adjusted index of difficulty. In the second study, the force field technique is compared with the “sticky target” technique [20] in two realistic pointing situations which involve several closely placed targets. The results show that the force fields improve pointing performance and that the sticky target technique does not.

Author Keywords

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

INTRODUCTION
In standard personal computing environments, the user interacts with the computer over a graphical user interface (GUI) using a keyboard, a screen and a pointing device. Most often, an indirect pointing device, such as a mouse, is used to steer a cursor on the screen in order to directly manipulate the screen objects. In such interfaces, pointing at and selecting targets are fundamental user tasks. Much research has been conducted to improve the interfaces and to understand the various aspects influencing the interaction process. In 1991, MacKenzie [14] ends one of his early works on human performance models in HCI by concluding:

“As human-machine dialogues evolve and become more “direct”, the processes and limitations underlying man’s ability to execute rapid, precise movements emerge as performance determinants in interactive systems.”

Several researchers have recognized the need for interfaces in which the user is able to perform fast and accurate pointing and have experimented with a pointing facilitating technique which has become known as the “sticky” target technique [6–8,11,20]. The technique is based on a modification of the mapping from input device motions to screen cursor motions. When an indirect pointing device is used, the device movements performed in control space (e.g., on a table when a mouse is used) are translated into corresponding cursor movements in display space on screen. The mapping from displacement distances in control space to distances in display space is defined by the control-display ratio coefficient (for short CD ratio). The CD ratio can be thought of as the “sensitivity of the input device”, as with a high CD ratio, a large device movement moves the cursor a moderate distance and with a lower CD ratio, a device movement of equal distance moves the cursor a greater distance, i.e., the difference is perceived as a change in cursor speed.

When using the CD ratio to support the user during pointing [6,11,20], the basic idea has been to dynamically change the CD ratio according to the cursor position and the location of pointing targets: when the cursor enters a target, the CD ratio is increased, this makes the device less sensitive and the user can now, presumably, faster and easier stop the cursor to accurately position it over the target. As soon as the cursor exits the target, the CD ratio is set back to the normal value. In this way, the user perceives the target as being “sticky”. Similar sticky effects can also be produced by using cursor warping techniques [7, 8].

Whereas previous research [6–8,11,20] shows that the sticky target technique can support the user in various simplified targeting situations, there is no empirical evidence for its usefulness in typical GUI situations, which include several closely placed icons. If all icons in such situations are made sticky, not only the stickiness of the target icon will influence the user, but also the stickiness of the icons the cursor passes by on the way toward the target. This might result
in slower pointing times in general. Without a reliable algorithm which accurately predicts the target icon\(^1\) and turns off the stickiness of all other icons, the problem caused by intervening icons casts doubt on the effectiveness and applicability of the sticky targets method. Therefore, the first main objective of the research presented in this paper was to evaluate the method in realistic usage situations, which involve several tightly packed sticky targets. The second main objective was to evaluate and compare an alternative pointing facilitating technique, which, instead of making targets sticky, uses a cursor warping algorithm to create “force fields” that steer the cursor toward the target center by inserting additional cursor displacements, a technique successfully used for menu navigation tasks [1, 2] and exact cursor positioning in alignment tasks [5].

In the following sections, we first present a review of the relevant literature, then we describe the force field technique and evaluate it in a pilot study. After that, we present the main study, evaluating the sticky target and force field techniques in two common pointing situations that involve several closely placed targets. Finally, the results are discussed and conclusions are made.

**RELATED WORK**

In general, the advantage with a low CD ratio is that great distances on the screen can be covered faster as with a high setting. The disadvantage with a low setting is that high-precision movements are harder to perform and are more time consuming [16] since the slightest device motion is amplified in display space. To pin down the optimal CD ratio is a more or less infeasible task since, among other factors, individual differences in motor control, habituation and personal preferences are all crucial determinants. Furthermore, the trade-off between fast positioning for large movements and speed and ease of fine-tuning movements are task dependent and, therefore, according to MacKenzie [16], the optimal CD ratio would be the one which minimizes the total positioning time across all performed tasks, as illustrated in Figure 1. However, the CD ratio which allows for a minimal total positioning time does not necessarily have to be the same as the one which produces fewest errors [15], a fact which further complicates an identification of an optimal CD ratio setting (see Balakrishnan [4] for an overview of literature comparing different CD ratios).

Most computer systems allow for a user defined CD ratio and an option for dynamic adaptation of the CD ratio, often referred to as “mouse acceleration”. The mouse acceleration functionality changes the CD ratio according to the speed at which the input device is moved. When the device is moved at low speed, a high CD ratio is activated and when fast device motions are detected, a lower CD ratio is used. The mouse acceleration functionality is motivated by the assumption that, when the user is moving the device fast, she/he intends to cover a greater distance, and when the device is moved slowly, the user is engaged in fine-tuning positioning. Whereas there is no empirical evidence that veloc-

\(^{1}\)To the authors’ knowledge, today such an algorithm does not exist (see the discussion in McGuffin and Balakrishnan [17]).

![Figure 1. The intersection of the two curves indicates the presumably optimal CD ratio which minimizes total positioning time (after MacKenzie [16]).](image)

![Figure 2. The test environments as described by (A) Worden et al. [20], (B) Keyson [11], (C) Blanch et al. [6], and (D) Cockburn & Firth [7] and Cockburn & Brewster [8].](image)
and B) report less dramatic improvements in both targeting time and error rate.

As can be seen in Table 1 (which provides an overview of the main results of these previous experiments) and in Figure 2, the experiments differ widely regarding task design, implementation of the stickiness, used target sizes, target distances and hardware. These differences make it hard to draw other than very general conclusions about the effectiveness of the sticky target technique. However, in summary, the reported results indicate that sticky targets might be a promising approach to support the user during pointing tasks, especially in pointing situations where the targets are small [6, 7, 20] and for inexperienced users [11] or users with reduced motor abilities [20].

Table 1. Survey of previous pointing task experiments with sticky targets.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Task/ Distractor</th>
<th>Sticky implementation</th>
<th>Target size (in pixels)</th>
<th>Distance (in pixels)</th>
<th>Pointing device/ Monitor size, Res.</th>
<th>Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Woorden et al. [20]</td>
<td>2D/Yes</td>
<td>target CD ratio 3 times higher than the outside ratio</td>
<td>4, 8, 16, 32 100, 200, 400</td>
<td>mouse 14&quot;, 640×480 pix.</td>
<td>6.2% 48%</td>
<td>20.5% 4 26% 4</td>
</tr>
<tr>
<td>Keyson [11]</td>
<td>2D/Yes</td>
<td>toward: reduced by half(^1) away: doubled(^2)</td>
<td>9.6, 14.4, 28.8(^2) 60, 120</td>
<td>track ball 21&quot;, -</td>
<td>3.6% 0%</td>
<td></td>
</tr>
<tr>
<td>Blanch et al. [6]</td>
<td>1D/No</td>
<td>target CD ratio 2 alt. 4 times higher than the outside ratio</td>
<td>2, 4, 8, 16, 32, 64</td>
<td>tablet w. puck 22&quot;, 1600×1200 pix.</td>
<td>10.9% 31% 5</td>
<td></td>
</tr>
<tr>
<td>Cockburn &amp; Firth [7]</td>
<td>1D/No</td>
<td>cursor warping (see the footnote on previous page)</td>
<td>6 10-100, 101-200, 201-300, 301-390</td>
<td>mouse 19&quot;, 1600×1200 pix.</td>
<td>27% 41%</td>
<td></td>
</tr>
<tr>
<td>Cockburn &amp; Brewster [8]</td>
<td>1D/No</td>
<td>cursor warping (see the footnote on previous page)</td>
<td>8 128, 256, 490</td>
<td>mouse 15&quot;, 1024×768 pix.</td>
<td>25% 50%</td>
<td></td>
</tr>
</tbody>
</table>

1) The target CD ratio was reduced by half for movements detected toward the target center and doubled for movements away from the center.
2) measured in mm 3) young users 4) old users 5) for doubled CD ratio 6) for quadrupled CD ratio

There exists only one study in which stickiness has been applied in a real GUI situation which included several sticky targets. However, the accessed task was not a pointing task but a menu navigation task. In one of their experiments, Cockburn and Brewster [8] made all menu items in a cascading pull-down menu sticky, leaving a narrow sticky-free area between each item. As can be anticipated, due to the cursor trajectories performed during menu navigation, which require the cursor to cross over several menu items, and due to the close spacing between menu items, the stickiness hampered navigation, slowed down menu selection times, and made the mouse feel “annoyingly slow” [8].

Since previous pointing task studies have used pointing tasks highly incompatible with pointing situations found in realistic GUIs, i.e., where pointing targets are located next to each other as e.g. in a toolbar, it is still questionable whether the sticky target approach scale to pointing in real GUIs.

FORCE FIELDS

Whereas the supportive effect produced by a sticky target rely on a change in cursor speed, i.e., the cursor behavior is only changed along the movement direction, a cursor warping algorithm can be designed to affect cursor behavior both collinear with movement and in sideway directions. By inserting small additional cursor displacements, a cursor warping software can overrule user cursor control to influence both speed and direction of cursor movements. 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 of “feeling” the attracting force when the input device is moved, especially when an isometric pointing device such as a track point is used. This virtual force effect is often referred to as pseudo-haptics, or simulated force-feedback. Such pseudo-haptics effects have previously been used to: simulate various textures felt and differentiated by the users [12], simulate friction and stiffness in virtual reality environments [13], facilitate the user during cursor positioning in alignment tasks [5] and during menu navigation [1, 2].

Recently, Ahlström [1] presented and successfully used a cursor warping technique in pull-down menus. The technique is based on the notion of invisible “force fields” associated with “force points” placed in the GUI. When the cursor enters a force field, the force implementing software intercepts the mouse motion events generated by the pointing device and inserts additional cursor displacements. Cursor movements toward the force point are reinforced and movements away from the force point are weakened. 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 and screen coordinates for a new cursor position inside a force field are calculated according to the following formula:

\[
\mathbf{n} = \mathbf{a} + s \cdot \left(\mathbf{a} - \mathbf{p}\right) \cdot \frac{\mathbf{f} - \mathbf{a}}{\|\mathbf{f} - \mathbf{a}\|} \quad (1)
\]

where:

- \(\mathbf{n} = (n_x, n_y)\) = (new) cursor position after applied force,
- \(\mathbf{a} = (a_x, a_y)\) = (active) cursor position after the last mouse motion,
- \(\mathbf{p} = (p_x, p_y)\) = (previous) cursor position, before last mouse motion,
- \(\mathbf{f} = (f_x, f_y)\) = position of the force point and,
- \(s = \text{strength of the force field.}\)
The resulting reals \( n_x \) and \( n_y \) coordinates are used as values for \( p \) in the next displacement computation and their rounded integer values are used to position the cursor in the integer based screen coordinate system.

Ahlström found that a medium strength setting is suitable for force fields placed over menu items and points out that too high a strength might, when large enough cursor movements toward the force point are made, cause the cursor to overshoot beyond the force point. Too high a strength also makes it hard for the user to leave the force field. Therefore, Ahlström used an escape functionality to help the user to exit force fields if the cursor entered an undesired field. The escape functionality deactivates the force after that the software has registered six consecutive mouse motions away from the force point and reactivates the force as soon as the cursor is moved back toward the force point.

**Force Enhanced Pointing Targets**

Compared to the sticky target technique, the main advantage of the force field technique, if applied in pointing situations, seems to be the possibility to extend the cursor influencing area beyond the target border. Whereas a CD ratio facilitation technique is limited to the screen area occupied by the target, the usage of a force field which steers the cursor toward the target center can also take advantage of possible interaction free void areas around the target, and so, the cursor influencing area can be made larger, which in turn might make it even more easy for the user to select the target.

We furthermore hypothesize that selection times of force enhanced targets can be predicted in a similar way as selection times of sticky targets are predicted. Selection time prediction of sticky targets is based on Fitts’ law [6], which is a temporal model of rapid hand movements [9], widely and successfully applied in human computer interaction to design and evaluate interaction techniques and pointing devices. According to Fitts’ law, the time \( T \) needed to move the cursor to a screen object is proportional to the logarithm of the distance \( D \) to the target divided by the target width \( W \):

\[
T = a + b \log_2 \left( \frac{D}{W} + 1 \right)
\]

The logarithmic term in the equation is commonly referred to as the *Index of Difficulty* (ID), measured in bits, and is a measurement of the difficulty of the movement task, more difficult tasks having higher IDs. The parameters \( a \) and \( b \) are determined empirically through linear regression.

In their study on sticky targets, Blanch et al. [6] focused on the theoretical aspects of sticky targets and analyzed the consequences in terms of Fitts’ law. They viewed the CD ratio as “the motor space scale relative to the visual space...” and showed that, since a local increase of the CD ratio over a target only changes the motor space representation of the target (i.e., makes the target bigger in motor space but leaves the display space size), selection times of sticky targets are more accurately predicted when the motor space size is used in the Fitts’ law equation instead of the display space size.

As in the case of CD ratio adaptation, the usage of force enhancement does not change the visual size of a target. Instead, the resulting change occurs in motor space. From Equation 1 we know that when the cursor is inside a force field and the input device is manipulated to move the cursor toward the force point, the warping algorithm elongates each cursor displacement. A motion in the opposite direction results in shortened cursor displacements. For the sake of simplicity, assume that, when a mouse is used as input device and no force field is present, moving the mouse one millimeter on the table in a rightward horizontal line results in a cursor displacement of one pixel to the right in visual screen space. If the same motion is made when the cursor is inside a force field with strength 0.8, and the motion is made toward the force point, the cursor is displaced by 1.8 pixels, as can be computed by solving Equation 1. That is, inside a force field a gain of 0.8 pixels is made for each millimeter the mouse is moved. Similarly, 0.8 pixels are lost for each millimeter in motor space if the mouse is moved to bring the cursor away from the force point. To compensate for these additional cursor displacements the index of difficulty of the force enhanced selection task has to be adjusted. The adjusted index of difficulty \( ID_{F} \) for a squared target with a squared force field, as depicted in Figure 3, is made by recalculating the distance to the target and the target size.

To obtain the force adjusted distance, \( D_{F} \), the width \( F \) of the force field is divided by two to obtain the distance along the approach axis between the border of the force field and the force point. For the selection task depicted in Figure 3 this distance is 12 pixels. In a non-force situation, the mouse has to be moved 12 millimeters for the cursor to travel across these 12 pixels on the screen. But when the force is applied, the warping algorithm displaces the cursor an additional 0.8 pixels per millimeter in motor space and it is only necessary to move the mouse 12/1.8 millimeters in order to cover the same screen distance. The difference between the necessary movement lengths in a standard condition and in a force enhanced condition is then subtracted from the distance \( D \) (the total distance from the start position to target center in a non-force situation) to obtain the force adjusted distance \( D_{F} \):

\[
D_{F} = D - \left( \frac{F}{2} - \frac{F}{1 + s} \right) = D - \frac{F}{2} \cdot \frac{s}{1 + s}
\]

where \( D \) is the distance in motor space between the start position of the selection task and the target center when no
force is applied, $F$ is the width of the force field, and $s$ is the strength of the force field.

To compute the force adjusted target width $W_F$ for the selection task in Figure 3 we begin with considering a non-force situation where the cursor is positioned in the center of the target. In this situation, to move the cursor outside the target taking a straight horizontal path to the right, the mouse has to be moved 4 millimeters since 4 millimeters in motor space displace the cursor 4 pixels on the screen. In a force enhanced situation, the same mouse motion would result in a reduced displacement of the cursor since it is moved away from the force point. From the discussion above (and according to Equation 1) we know that if the strength of the force field is set to 0.8, 0.8 pixels are lost per one millimeter motion in motor space. Therefore, $4 \cdot 0.8 = 3.2$ pixels are lost during the movement sequence. To cover the whole distance of 4 pixels, the mouse has to be moved $4 + 3.2 = 7.2$ millimeters. The same applies to the situation when the cursor is moved from the target center in the opposite direction, i.e., back toward the start position of the selection task. Therefore, the force adjusted target width $W_F$ is $2 \cdot 7.2 = 14.4$ millimeters in motor space, which can be computed using the following formula:

$$W_F = W + \left(\frac{W}{2} \cdot s \cdot 2\right) = W (1 + s) \quad (4)$$

where $W$ is the width of the target in motor space when no force is applied, and $s$ is the strength of the force field.

When the force adjusted target width, $W_F$, and distance, $D_F$, are obtained, the force adjusted index of difficulty, $D_F$, can be computed and the selection time $T$ can be predicted according to Fitts’ law:

$$T = a + b \text{ ID}_F, \text{ with } \text{ ID}_F = \frac{D_F}{W_F} + 1 \quad (5)$$

USER STUDY 1
A pilot user study was conducted to verify the application of force field in simple pointing tasks and to investigate if the suggested force adjusted Fitts’ law model can be used to predict selection times of force enhanced targets.

Apparatus and Participants
The experiment was performed on a notebook running Windows 2000 with a 14.1-inch monitor (resolution 1024×768 pixels). A conventional mechanical mouse was used as pointing device. No mouse acceleration and a CD ratio of 1:4 were used. Ten (7 male) right-handed experienced computer users aged between 22 and 40 years (mean=28.6, SD=5.5) participated.

Task and Experimental Design
A rectangle on the left side of the screen was used as the start point for each task. After a click in the start box a blue target square appeared to the right on the horizontal line from the center of the start box. The participants were asked to select the blue target with a click as fast as possible. Timing started on exit of the start box and was stopped with a click inside the target square. After a successful selection the target square disappeared and a new trial could be started from the start box. All targets were placed within an invisible square shaped force field of either strength 0 (i.e., no force - standard cursor behavior), 0.4, or 0.8. The size of the force field was in all trials nine times larger than the target and was centered over the target.

The study design was within subjects $3 \times 3 \times 3$ (force strength × target width × target distance) with 16 repetitions for each cell. Target widths were $8 \times 8, 16 \times 16$, and $32 \times 32$ pixels, and distances 128, 256, and 512 pixels. Each participant performed 432 trials which were divided in 8 blocks with 2 repetitions of each force strength-distance combination presented in random order. Each block was divided in three sets of 18 trials. After each set was completed, a recess screen was shown, and the participant could take a short break if desired. The study took about 40 minutes per participant, including one block of practice. The participants were not informed about the presence of the force fields.

Analyses and Results

Error rates
A total of 98 trials (2.3%) were logged as invalid trials. In 42 trials participants clicked outside the target, and 56 trials were slip-off errors where the mouse button was pressed inside the target but released when the cursor was positioned outside. More than half of the errors (55 trials) were made at the smallest target size. Most errors were made in the standard condition with no strength, 53 errors compared to 34 and 11 in the weak and strong strengths conditions respectively ($\chi^2 = 27.71; df = 2; p < .001$). A dichotomization of the standard condition vs. the two strength conditions was applied ex post in order to investigate the distribution of error. The Odds Ratio was 2.4, i.e., the risk of making an error in the standard condition was more than two times higher than in any of the force enhanced conditions. We can conclude that the force fields with weak strength reduced the error rate with about 36% and the stronger fields with about 79%, a results comparable with most of the previous studies made with sticky targets (cf. Table 1). No further analysis concerning error rate was made, and the following analyses are based on the error free trials only.

Practice effects
An ANOVA with selection time as dependent variable, participant as random factor and block and strength of force field as fixed factors showed a significant main effect for block ($F_{7,63} = 2.37, p < .05$) indicating a practice effect. After the first block, performance stabilized and no differences were present between the last seven blocks ($F_{6,54} = 1.73, p = .130$). Therefore, data from the last seven blocks are used for the following analyses.

Selection times
An overall 8.7% performance improvement was measured in favor of strength 0.4 over the standard condition. Strength 0.8 was on average 17.3% faster than the standard condition. The standard condition and strength 0.4 condition had overall mean selection times of 909 ms and 830 ms, respec-
significant main effects for all factors were found (strength: in line with previously reviewed experiments [6, 7, 20]). Significant participant × width interaction ($F_{18, 26} = 4.27, p < .001$) and a significant strength × width ($F_{3, 36} = 2.93, p < .05$) interaction were found. After close inspection of Figure 4 A it is clear that the force was more effective on small targets that on large targets, a result in line with previously reviewed experiments [6, 7, 20]. Significant main effects for all factors were found (strength: $F_{2, 18} = 30.05, p < .001$, distance: $F_{2, 18} = 133.08, p < .001$, width: $F_{2, 18} = 66.57, p < .001$, and participant: $F_{9, 20} = 6.09, p < .001$). The ANOVA confirms the descriptive data in Figure 4 A and clearly shows that the selection times were improved by the force fields, whereby the strongest fields reduced selection time more than the weaker ones.

**Prediction of selection time**

Figure 4 B shows selection time as a function of index of difficulty, $ID$, without any force adjustments applied. That is, it is assumed that the pointing tasks in the tested conditions (strength 0, i.e. standard, and strength 0.4 and 0.8) were all equally hard to perform. Regressing selection time against $ID$ results in an $r^2$ value of 0.82, i.e., 82% of the variation in the data can be explained by the Fitts’ law model. In Figure 4 C selection time is shown as a function of the force adjusted index of difficulty, $ID_F$, obtained by applying Equation 3, 4 and 5 in turn and order. The force adjustments result in a leftward shift of the data points from tasks with a force strength greater than zero (applying the force adjusting computations using a strength $s = 0$ does not change the index of difficulty). A regression after the force adjustment results in an $r^2$ value of 0.906, i.e., more of the variation in the data can be explained now. This in turn indicates that the force adjustments were justified and it confirms the usage of a force adjusted index of difficulty for force enhanced targets.

**USER STUDY 2**

After the encouraging results from the pilot study a second user study was conducted to evaluate the implications of sticky targets and force enhanced targets in realistic GUI situations which involve several closely placed targets.

**Apparatus and Participants**

The experiment was performed on a notebook running Windows XP with a 15-inch monitor (resolution 1024 × 768 pixels). A conventional optical mouse and the standard touch pad built into the notebook with its associated selection button were used as pointing devices. No mouse acceleration and a CD ratio of 1:4 were used.

Twelve volunteers (7 male and 5 female, aged between 18 and 34 years, mean=28.0, SD=5.23) used the mouse. All twelve volunteers were expert computer users and used a mouse five days a week. Twelve other volunteers (8 male and 4 female, aged between 21 and 36 years, mean=27.4, SD=5.05) used the touch pad. All twelve participants were expert computer users and used a touch pad about one to two times a week. All 24 participants had normal or corrected to normal sight, were right-handed and performed the test using their preferred hand.

**Tasks and Experimental Design**

The experimental task setup was chosen to be as natural as possible. In order to mimic realistic situations, two frequently performed GUI selection tasks were deployed. The first task, the Toolbar Task, included selecting one of several icons displayed in the toolbar at the bottom of the screen, as displayed in Figure 5. In the second task, the Window Task, one of the window manipulation icons, minimize, maximize and close, placed in the upper right corner of a window had to be selected. Each selection was started from the start box located in the center of the screen.

The tasks were performed in three different interfaces, one standard interface, one with CD ratio adaptation over targets, and one with force enhanced targets. In the CD ratio interface, the CD ratio was increased to ten times the outside target ratio as the cursor entered a target (from 1:4 to 1:0.4). The increased CD ratio was kept as long as the cursor was inside an icon and was moved toward the icon center. When
the cursor was moved away from the center or when it exited the icon, the CD ratio was reduced back to the outside target ratio. But as soon as a motion toward the target center was registered, the CD ratio was increased again.

In the force enhanced interface the icons were equipped with force fields which pushed the cursor toward the icon center. The force was turned off after that six consecutive movements inside the field away from the center were registered. When moving the cursor toward the center, the force was turned on again. In the Toolbar Task, all eight icons were enhanced and in the Window Task all window icons were enhanced. No other icons were enhanced.

The participants were encouraged to select the targets as fast as possible but to balance speed and accuracy. The timing started when the cursor left the start box and ended when the target was selected. If the wrong icon was selected or if the device selection button was released outside an icon, an error trial was registered and a message was displayed. After an error, a new trial was started from the start box. After a successful selection the toolbar and the windows disappeared, the trial counter next to the start box was incremented and the next trial could be started by clicking in the start box. After a click in the start box, the toolbar and the windows were displayed and one icon, the new target icon, was painted green. The participants were instructed to first locate the green icon, and then, only after the target had been located, start moving the cursor.

All participants performed three test sessions, one for each interface. The order in which the different interfaces were tested was counterbalanced. Each one of the six interface order combinations was assigned to two participants in each device group. A test session included 40 Toolbar Task trials followed by 72 Window Task trials. The toolbar trials were divided in four blocks, each block containing two trials for each of the five target icons in the toolbar, which are numbered in italics in Figure 5 B. The window trials were also divided in four blocks, each block containing two trials for each one of the nine window icons (numbered 6 to 14 in Figure 5 B). All error trials within a block were repeated in random order until the participant had made 10 respectively 18 successful trials. The experiment included a total of 8064 trials, 4032 trials per device. Participants were informed about the different interfaces and their functionality and had one block of practice trials before each test session. A session lasted about 15 minutes and participants were allowed breaks between blocks of trials.

### Analyses and Results

The analyses and results from the mouse and the touchpad tests are presented in parallel, beginning with error rates followed by practice effects and selection times.

#### Error rates

Table 2 lists the number of errors according to error type (Miss: click outside target, Slip: selection button pressed inside target but released outside, and Icon: wrong target)

<table>
<thead>
<tr>
<th>Task</th>
<th>Error</th>
<th>Mouse Stand. CD</th>
<th>Mouse Force</th>
<th>Touch pad Stand. CD</th>
<th>Touch pad Force</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toolbar</td>
<td>Miss</td>
<td>9</td>
<td>0</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>Slip</td>
<td>10</td>
<td>6</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Icon</td>
<td>9</td>
<td>6</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Window</td>
<td>Miss</td>
<td>22</td>
<td>13</td>
<td>18</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>Slip</td>
<td>4</td>
<td>4</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Icon</td>
<td>3</td>
<td>10</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>57</td>
<td>39</td>
<td>50</td>
<td>61</td>
</tr>
</tbody>
</table>

Table 2. Errors.

---

3 A demo application with force enhanced toolbar icons is available at http://ias.uni-klu.ac.at/en/interaktion.php
icon selected). Error rates were generally low, with 146 error trials (3.5%) in the mouse test, and 162 error trials (3.8%) in the touch pad test. As seen in Table 2 there were only slight differences across the three interfaces. We can therefore conclude that neither the force enhancement nor the CD ratio adaptation influenced the error rate. Following analyses are based on the error free trials only.

**Practice effects**

Two separate ANOVAs, one for each device test, with selection time as dependent variable, participant as random factor, block (1-4), task (Toolbar, Window), and interface (standard, force, CD ratio) as fixed factors were performed to examine any possible learning effects. Neither ANOVA showed any main effect for block (mouse: $F_{3,33} = .263, p > .05$, touch pad: $F_{3,33} = 2.13, p > .05$) or any interaction effects involving the block factor.

**Selection times**

Two separate ANOVAs were performed for each device test, one including the Toolbar Task, one including the Window Task. Each ANOVA had selection time as dependent variable, participant as random factor, icon (1-5 for the Toolbar Task, 6-14 for the Window Task) and interface (standard, force, CD ratio) as fixed factors. Neither ANOVA showed any significant threefold interaction effects but main effects for all three factors and several pairwise interaction effects. The significant effects are listed in Table 3. The mean selection times for all interfaces are shown in Table 4 which lists all time comparisons between the interfaces according to task and pointing device. Simple contrasts computed for

<table>
<thead>
<tr>
<th>Device/Task</th>
<th>Main/interaction effect</th>
<th>F-statistic</th>
<th>$\alpha$-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mouse/Toolbar</td>
<td>interface</td>
<td>$F_{2,22} = 4.91$</td>
<td>$p &lt; .01$</td>
</tr>
<tr>
<td></td>
<td>icon</td>
<td>$F_{4,44} = 39.69$</td>
<td>$p &lt; .001$</td>
</tr>
<tr>
<td></td>
<td>subject</td>
<td>$F_{11,26} = 7.80$</td>
<td>$p &lt; .001$</td>
</tr>
<tr>
<td></td>
<td>interface×icon</td>
<td>$F_{6,88} = 4.76$</td>
<td>$p &lt; .001$</td>
</tr>
<tr>
<td></td>
<td>interface×subject</td>
<td>$F_{2,22} = 2.72$</td>
<td>$p &lt; .01$</td>
</tr>
<tr>
<td>Mouse/Window</td>
<td>interface</td>
<td>$F_{2,22} = 10.20$</td>
<td>$p &lt; .001$</td>
</tr>
<tr>
<td></td>
<td>icon</td>
<td>$F_{6,88} = 106.22$</td>
<td>$p &lt; .001$</td>
</tr>
<tr>
<td></td>
<td>subject</td>
<td>$F_{11,32} = 9.76$</td>
<td>$p &lt; .001$</td>
</tr>
<tr>
<td></td>
<td>interface×icon</td>
<td>$F_{2,16,176} = 3.77$</td>
<td>$p &lt; .001$</td>
</tr>
<tr>
<td></td>
<td>interface×subject</td>
<td>$F_{2,21,176} = 2.97$</td>
<td>$p &lt; .001$</td>
</tr>
<tr>
<td>Touch pad/Toolbar</td>
<td>interface</td>
<td>$F_{2,22} = 15.51$</td>
<td>$p &lt; .001$</td>
</tr>
<tr>
<td></td>
<td>icon</td>
<td>$F_{4,44} = 13.83$</td>
<td>$p &lt; .001$</td>
</tr>
<tr>
<td></td>
<td>subject</td>
<td>$F_{11,32} = 4.81$</td>
<td>$p &lt; .001$</td>
</tr>
<tr>
<td></td>
<td>interface×icon</td>
<td>$F_{8,88} = 4.53$</td>
<td>$p &lt; .001$</td>
</tr>
<tr>
<td></td>
<td>interface×subject</td>
<td>$F_{2,22} = 5.81$</td>
<td>$p &lt; .001$</td>
</tr>
<tr>
<td></td>
<td>icon×subject</td>
<td>$F_{4,44} = 2.66$</td>
<td>$p &lt; .001$</td>
</tr>
<tr>
<td>Touch pad/Window</td>
<td>interface</td>
<td>$F_{2,22} = 24.20$</td>
<td>$p &lt; .001$</td>
</tr>
<tr>
<td></td>
<td>icon</td>
<td>$F_{8,88} = 151.35$</td>
<td>$p &lt; .001$</td>
</tr>
<tr>
<td></td>
<td>subject</td>
<td>$F_{11,29} = 10.32$</td>
<td>$p &lt; .001$</td>
</tr>
<tr>
<td></td>
<td>interface×icon</td>
<td>$F_{2,22} = 2.29$</td>
<td>$p &lt; .001$</td>
</tr>
<tr>
<td></td>
<td>interface×subject</td>
<td>$F_{2,22} = 3.91$</td>
<td>$p &lt; .001$</td>
</tr>
<tr>
<td></td>
<td>icon×subject</td>
<td>$F_{8,88,176} = 1.58$</td>
<td>$p &lt; .001$</td>
</tr>
</tbody>
</table>

Table 3. Significant main and interaction effects.

post hoc comparisons showed that the force enhanced interface was significantly faster than both the standard and the CD ratio interface in both device tests in both tasks. The CD ratio interface was only significantly faster than the standard interface in the Toolbar Task when the touch pad was used, in all other comparisons between the standard and the CD ratio interface, the differences were not significant.

Figure 6 A and B show the mean selection times for each icon for both pointing devices. Overall, in the mouse test, the force enhanced interface was 7.8% faster than both the standard interface and the CD ratio interface (mean selection time 742 vs. 804 and 801 ms for force, standard and CD ratio respectively). In the touch pad test the force enhanced

Table 4. Comparisons of selection times.

<table>
<thead>
<tr>
<th>Device/Task</th>
<th>Comparison</th>
<th>Selection time (in ms.)</th>
<th>Difference (in %, $\alpha$-value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mouse/Toolbar</td>
<td>Stand. vs. Force</td>
<td>798 vs. 737</td>
<td>7.6, ($p &lt; .01$)</td>
</tr>
<tr>
<td></td>
<td>Stand. vs. CD</td>
<td>798 vs. 784</td>
<td>1.7, ($p &gt; .05$)</td>
</tr>
<tr>
<td></td>
<td>CD vs. Force</td>
<td>784 vs. 737</td>
<td>5.9, ($p &lt; .05$)</td>
</tr>
<tr>
<td>Mouse/Window</td>
<td>Stand. vs. Force</td>
<td>810 vs. 746</td>
<td>7.9, ($p &lt; .05$)</td>
</tr>
<tr>
<td></td>
<td>Stand. vs. CD</td>
<td>810 vs. 819</td>
<td>1.0, ($p &gt; .05$)</td>
</tr>
<tr>
<td></td>
<td>CD vs. Force</td>
<td>819 vs. 746</td>
<td>8.9, ($p &lt; .01$)</td>
</tr>
<tr>
<td>Touch pad/Toolbar</td>
<td>Stand. vs. Force</td>
<td>1167 vs. 984</td>
<td>15.6, ($p &lt; .001$)</td>
</tr>
<tr>
<td></td>
<td>Stand. vs. CD</td>
<td>1167 vs. 1093</td>
<td>6.3, ($p &lt; .05$)</td>
</tr>
<tr>
<td></td>
<td>CD vs. Force</td>
<td>1093 vs. 984</td>
<td>9.9, ($p &lt; .01$)</td>
</tr>
<tr>
<td>Touch pad/Window</td>
<td>Stand. vs. Force</td>
<td>1142 vs. 1017</td>
<td>10.9, ($p &lt; .001$)</td>
</tr>
<tr>
<td></td>
<td>Stand. vs. CD</td>
<td>1142 vs. 1160</td>
<td>1.5, ($p &gt; .05$)</td>
</tr>
<tr>
<td></td>
<td>CD vs. Force</td>
<td>1160 vs. 1017</td>
<td>12.3, ($p &lt; .001$)</td>
</tr>
</tbody>
</table>

Figure 6. Mean selection times for the different target icons in the mouse (A) and touch pad (B) test.
interface was 13.3% faster than the standard interface (1000 vs. 1154 ms.) and 11.1% faster than the CD ratio interface (1000 vs. 1126 ms.).

However, these differences have to be seen in light of the significant interaction effects listed in Table 3. The interface×subject interactions indicate that at least one participant reacted differently from the other participants to at least one interface. Which in turn indicates that the tested interfaces might be differently suitable for different users. Similarly, the icon×subject interactions in the touch pad test indicate that some targets were particularly hard or easy to select for one or more participants. However, the most interesting interactions are the interface×icon interactions which show that the effect of the two enhancements depends on which icon they are applied to. A closer inspection of Figure 6 A and B clarifies the situation. For example, in the mouse test, the greatest relative time reductions caused by the force fields were for icon 3, 8, and 12 (by over 15%). But for other icons the reduction was lower, or, as in the case of icon 9 and 10, the pointing was even slightly hampered. The results from the touch pad test vary in similar ways with improvements ranging from 5% (icon 9) to over 20% for icon 1, 4, and 14.

DISCUSSION AND FUTURE WORK

The presented results clearly show the advantage of the force field technique compared to the sticky target technique and a standard non-enhanced interface. Whereas, the stickiness of the targets with adaptive CD ratio only marginally helped the user during the selection of some few icons, the force fields were supportive in all targeting situations but two (in the mouse test for icon 9 and 10). If we analyse Figure 7, which displays the relative difference in selection times between the force enhanced targets and the standard targets for each icon in both the mouse and touch pad test, in parallel with Figure 5 it is clear that the users profited least from the force fields when selecting icons which were positioned behind other icons (i.e., icon 9, 10, and 11). On the other hand, the highest gain tends to be for icons where the force fields are larger, as for icon 1, 3, 4, 8, 12, and 14. But this trend is not fully consistent across the two devices. Particularly interesting is the situation in which icon 7 is involved. This was the only situation in which the gain was higher for the mouse users than for the touch pad users. At the moment this fact is not explainable. Further experiments with pointing tasks which are more comparable and which allow for more systematic analysis are needed to interpret such situations. Future experiments should also focus more on how neighboring force fields distract the user, e.g., by analyzing cursor trajectories taken toward targets and by logging the number of visited icons during the pointing task. From the experiment at hand we can only conclude that, although a mechanism which deactivated the force was used, the potential effectiveness of a force field is reduced by its neighboring force fields. In the case of sticky targets, this unfortunate effect was even larger and made the sticky interface no better than the standard interface.

Since the touch pad users, which were less experienced in operating the device than the mouse users were, profited most from the force fields, the presented results indicate that users with little experience in operating a pointing device can profit more from pointing facilitating techniques than highly skilled users. This fact might guide researchers to concentrate on less skilled users which have a notably problem with controlling a screen cursor. For such users there are more room for improvement.

Finally, it is important to stress that the size, shape and strength of the force fields used in a particular GUI situation should be highly dependent on that specific situation and its screen design. Not all pointing tasks in all contexts seem to be suitable for force enhancement. We therefore plan to systematically explore the GUI design space in order to experiment with and optimize the force fields in other suitable pointing situations.

CONCLUSIONS

In this paper we studied two related pointing facilitating methods: the force field technique and the “Sticky target” technique. The force field technique aims to support the user by using a simple cursor warping algorithm to implement “force fields”, which help the user steering the cursor toward a pointing target. The Sticky target technique relies on local changes in the CD ratio to restrict movement as the cursor passes over a screen object. The two techniques were assessed and compared in a controlled user experiment which deployed two multi-target pointing tasks. The results showed that pointing in a force enhanced interface is on average 7.8% to 13.3% (depending the users’ skills in operating the pointing device used) faster than in a standard interface or an interface with sticky targets. The results also showed that both techniques fail to reduce error rates.

We have made four main contributions. First, we demonstrated that the force field technique, which was originally designed to enhance cascading pull-down menus, can also be used to support the user during pointing tasks. Second, we showed how a Fitts’ law model can be adjusted and used to predict selection time of force enhanced targets. Third, we showed that the force technique does not only facilitate pointing in simplistic pointing situations, but also in more complex and realistic situations where several enhanced targets

![Figure 7. Relative difference between the force enhanced interface and the standard interface.](image-url)
are involved. And fourth, we extended previous research on sticky targets by exploring the technique in realistic settings and showing that the technique fails to support the user in such situations.

REFERENCES


In the pointing facilitating literature, a rather seldom tested situation, cf. Balakrishnan [4].