Trace Select: Acquiring Remote Targets with Gesture Guides Generated from Glyphs

Atsuhiko Maeda
NTT Corporation.
1-1 Hikari-no-oka, Yokosuka-shi
Kanagawa, 239-0847 Japan
atsuhiko.maeda@gmail.com

Minoru Kobayashi
NTT Corporation.
1-1 Hikari-no-oka, Yokosuka-shi
Kanagawa, 239-0847 Japan
minoru@acm.org

CHI'13, April 27 – May 2, 2013, Paris, France.

Abstract

When selecting targets in a Smart TV service, the user is frequently required to select remote targets such as menu items, but is unable to touch the targets directly or use a mouse. Solutions include gesture-based rapid selection techniques. However, such techniques have problems in that they hide the original information. We propose Trace Select, a novel technique for selecting remote targets; it generates gesture guides from the targets' characters. The guides don't conceal the original targets. An experiment shows that Trace Select outperforms cursor-based pointing and activation-required gestural selection.

Author Keywords

Selection; Gestures; Remote target

ACM Classification Keywords

H.5.2. User interfaces: Graphical user interfaces (GUI), Input devices and strategies, Interaction styles.

General Terms

Human Factors; Design

Introduction

When using a Smart TV service to access rich content via Internet, the user is required to input control operations for menu selection or screen transition by using a remote control much more frequently than is common with the conventional TV service. Recently, as one of the solutions to reduce this user's burden, TV set / set-top box developers are adopting remote controls equipped with a touchpad [7, 8, 9]. The advantage of a touchpad-based remote is that it can be used not only for target selection but also for speed-controllable scrolling by wipes along an edge of the touchpad or for gesture input for executing particular functions [1]. However, further advances in interface efficiency are desired. In this paper, we focus on making it easier to select targets containing text by using touchpad-based remotes.

Effective solutions are to use gestural shortcut signs that the user inputs via the device [2, 4, 6]. These signs are related to the targets and are laid over them. Such techniques enable the user to select the desired target while gazing at the screen unlike number-based selection [4]. However, always displaying such visual signs prevents the user from viewing the original information. If the targets are wide images, it is possible to place the signs on their edges [2]. However, the same approach is not possible with dense text targets. Even if the signs are shown offset in balloons, these might hide peripheral information of the targets. Therefore, these techniques generally require turning the gesture representations on/off, which increases the number of wasteful operations, cognitive load for gesture identification (see below experiment), and selection time.

To solve the above problem, we propose Trace Select, a novel technique that uses gesture representations that can be always displayed because they do not hide the original information (Figure 1). The technique dynamically finds possible gesture candidates based on glyphs of texts, which are the targets.

Figure 1. Gesture representations (fill-in with blue along glyphs) can be always displayed because they do not hide the original information, and users can identify and draw the gesture of the desired target immediately. All the user need do is to trace the highlighted mark, rather than any character in the target string.

Trace Select

The proposed technique is inspired by the handwriting recognition system Palm OS' Graffiti [3], in which each alphabetical character is mapped to one simple single-stroke gesture, and the Glyph-based text entry method [5]. To cover as many targets as possible, we extracted as many gesture candidates from one character as possible even though this creates duplicated candidates. In total, we created 114 gestures for uppercase and lowercase characters and numbers; 48 are unique (Figure 2).

To assign glyphs to all targets, the algorithm preferentially determines the gestures of targets with fewer gesture candidates. Details are as follows.

1. In advance, all glyphs which are available for each character are registered in a database.

2. In actually assigning the glyphs, all targets (strings) in the screen are extracted. In each target, all gesture candidates related to its characters are extracted from the database.

3. The targets are sorted in ascending order of the total number of gesture candidates and the top target (has the fewest candidates) is popped from the list.

4. One of the candidates of the popped target is randomly selected and prevented from being considered as a candidate of subsequent targets.

5. Until all targets have been assigned glyphs, step 3 is reentered.

Visual design of the glyphs

We designed the glyphs for high visibility regardless of the target's font size. During pilot testing with 3 users, we explored several design forms (Figure 3). When the font size is small, users felt that the fill-in approach (left side of the figure) offers the easiest identification.

Limitation and the solution

The number of gestures that can be assigned to each character is limited. The test we performed indicated that the rate of successfully assigning unique glyphs to random English movie titles fell under 95% when 36 titles were presented simultaneously (the mean of 10000 samples). However, the mean number of targets in one TV screen is 27.7 [4]. Even if the number of targets exceeds 36, it is possible to combine contiguous two characters in each target string to increase gestural shortcuts (This technique is not used in the experiment below).

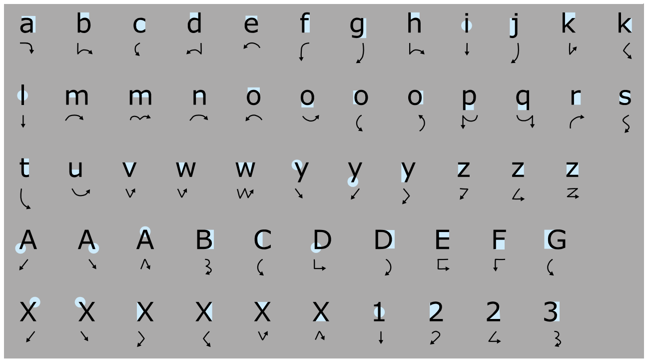

Figure 2. Sample gesture representations in Trace Select. One character has one or more gesture candidates. The dominant stroke directions are straight down and cursive gestures.

Figure 3. Visual designing the glyphs in Trace Select. When the font size is small, users felt that fill-in (left) is more visible than partial color change (center) or thickness change (right).

Experiment

In this experiment, we wanted to determine the following:

- Influence of target's distance (amplitude) and size on Trace Select performance: Since Trace Select does not require users to pick a target using a conventional pointing-based approach, we wanted to determine whether this new technique is dependent on target distance and size, especially whether users can recognize the glyphs rapidly even if the font size is small.

- Effect of eliminating the need to switch gesture-guide on/off: The key advantage of Trace Select is that it does not hide the targets and is always available. To confirm these benefits we created two versions of Trace Select: TS-Show-Act and TS-Entry-Act. The former does not initially show the glyphs and demands that the user tap the panel to activate glyph display (and entry). The latter shows the glyphs but the user must tap the panel to activate glyph entry. Thus, whether glyphs appear from the start or not depends on the variation, but both need the same number of operations. Here it is verified that eliminating the need to switch the gesture-guide on/off decreases not only the number of operations needed but also the time taken to read/identify the gesture representations.

- Comparison with conventional techniques: The first baseline is cursor-based pointing, which uses the default Microsoft Windows 7 pointer acceleration technique. Although many selection techniques have been academically proposed, cursor-based selection is still the commercial standard, even on touchpad remote controls, and many users prefer it to the others. Moreover, recent cursor-based selection is more sophisticated with use of acceleration. Therefore, it is necessary to primarily show that Trace Select outperforms cursor-based selection using acceleration and is subjectively preferred to it on a touchpad. The second is activation-required gestural selection. Although Gesture Select [1] is suitable, we adopt TS-Show-Act due to its lower operation costs. TS-Show-Act is similar to Escape and Gesture Select; these techniques require turning the gesture representations on/off.

Tasks

The first task was a 2-D reciprocal selection task. On the screen, one target (a movie title) was displayed at one position. After selecting the target, the next target (another movie title) was displayed at another position using another font size. Participants were required to successfully select the target to continue.

The second task was a video streaming service-like selection task. To determine the performance in a realistic situation, we created a targeting task by imitating current video streaming services such as Apple TV and ROKU, etc. In detail, we designed 6 screen layouts (see Figure 4). In this task, the movie title indicating the target appeared first at the top of the screen. After 3 seconds passed, several other movie titles together with the target were presented using one of 6 layouts. After successfully selecting the target, targeted movie title and the screen layout were changed.

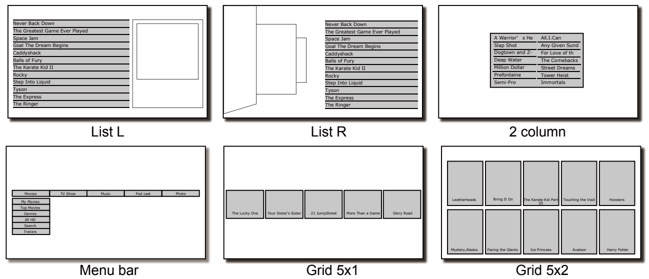

Figure 4. Screen layouts based on commercial video streaming services.

Participants and Apparatus

Eighteen participants (aged 19-30, 8 males, one left-handed) were recruited from a temporary staffing agency. None had ever used a device with a gesture-based interface.

The experiment used a 2GHz Core 2 Duo PC running Windows 7. The display (TV) connected to the PC was a 42-inch LCD with 1360x768 pixel resolution. The distance between the display and a user sitting on a chair is 2 meters, the distance at which all participants could read the smallest font (12pt) used in the experiment. As the input device, we used EVERGREEN's 5 inch USB touchpad DN-HCP112, which enabled not only normal relative coordinate input but also absolute coordinate input for gesture input. Our software was HTML5 and Javascript-based and ran on Google Chrome Browser in full screen mode. All touch events were recorded on the millisecond time scale. The font used was Verdana. The font color was black, background was light gray, and glyphs were sky-blue. As the targets, 600 actual English movie titles were used in the experiments.

Experimental Design

A repeated-measures full factorial within-participant design was used. In the 2D reciprocal selection task, the independent variables were technique (Cursor Selection, Trace Select, TS-Show-Act, and TS-Entry-Act), distance (Near: 90-419 px, Mid: 420-749 px, Far: 750-1079px, font-size (12, 18, 24, and 36 pt): 18 participants x 4 techniques x 3 distances x 4 font sizes x (3+3) blocks (training + measured) = 5184. In the video streaming service-like pointing task, the independent variables were technique and screen layout: 18 participants x 4 techniques x 6 layouts x (3+3) blocks (training + measured)= 2592. The order of technique used was randomized; the reciprocal selection task preceded the video streaming task.

Results

2-D RECIPROCAL SELECTION TASK

The measured selection times are presented in Figure 5. Bonferroni's multiple comparison test showed that Trace Select was significantly faster than Cursor (t_{17}=11.53, p<0.00001) and TS-Show-Act (t_{17}=4.93, p<0.0003), and TS-Entry-Act was significantly faster than TS-Show-Act (t_{17}=5.46, p<0.0001); there was no significant difference between Trace and TS-Entry-Act (t_{17}=0.2, p=0.85). Regarding initial reaction time (i.e., the time taken to start a gesture), Trace and TS-Entry-Act were significantly faster than TS-Show-Act (t_{17}=6.22, p<0.00001 / t_{17}=6.59, p<0.00001); there was no significant difference between Trace and TS-Entry-Act (t_{17}=0.84, p=0.42). Finally, the effects of Distance and Font Size for Technique on initial reaction / finger movement time (i.e., the time taken to draw a glyph) are presented in Table 1. The initial reaction times of Trace Select and the variants were significantly influenced by Font Size; finger movement times were significantly influenced by Distance (see Table 1 for details).

VIDEO STREAMING SELECTION TASK

Figure 6 shows the selection time. Bonferroni's multiple comparison test showed that Trace Select was significantly faster than Cursor (t_{17}=14.0, p<0.00001), TS-Show-Act (t_{17}=5.44, p<0.0001), and TS-Entry-Act (t_{17}=3.86, p<0.0025), and TS-Show-Act and TS-Entry-Act were significantly faster than Cursor (t_{17}=5.46 / p<0.0001, t_{17}=12.51 / p<0.00001) in all screen layouts. Regarding initial reaction time, Trace was significantly faster than TS-Show-Act (t_{17}=5.90, p<0.0001) and TS-Entry-Act (t_{17}=4.52, p<0.0003). TS-Entry-Act was significantly faster than TS-Show-Act (t_{17}=3.85, p<0.0013).

Overall

For both tasks, mean error rates were 0.41% for Cursor, 1.85% for Trace Select, 1.85% for TS-Show-Act and 1.96% for TS-Entry-Act; there was no significant difference in error rate ($F_{3, 51}$=2.0253, p=0.1212). The mean time taken to draw a glyph was 0.31 sec. The mean mode change latency of TS-Show-Act and TS-Entry-Act was 0.147 sec.

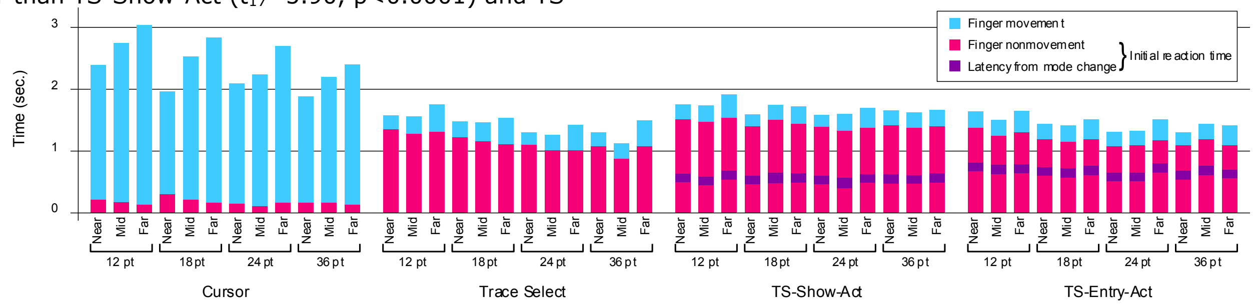

Figure 5. Selection time in 2-D reciprocal selection task. Trace Select was significantly faster than Cursor and TS-Show-Act; there was no significant difference between Trace Select and TS-Entry-Act.

	Initial reaction time				Finger movement time			
	Cursor	Trace Select	TS-Show-Act	TS-Entry-Act	Cursor	Trace Select	TS-Show-Act	TS-Entry-Act
Distance	> Near - Mid - Far	n.s.	n.s.	n.s.	< Near < Mid < Far	< Near < Mid < Far	< Near - Mid < Far	< Near - Mid < Far
Font Size (pt)	12 - 18 > 24 - 36	> > 12 - 18 - 24 - 36 >	> > 12 - 18 > 24 - 36 >	> > 12 -18 > 24 - 36	> > 12 > 18 - 24 > 36 >	n.s.	n.s.	n.s.

Table 1. The result of multiple comparison in 2D reciprocal selection task. The initial reaction times of Trace Select and the variants were significantly influenced by Font Size; finger movement times were significantly influenced by Distance. ("<" and ">" mean being significantly faster than and slower than at p<0.01 level, respectively. "-" and "n.s." mean no significant difference.)

Discussion

Trace Select significantly outperformed the conventional Cursor-based technique in both tasks (37% faster on average), and was preferred overall by 16 of 18 users although Trace Select's error rate was comparatively high. A single-blind human analysis of errors revealed that as many as 71% of Trace Select and the variants' errors were false negatives caused by the software recognizer. That is, the recognizer used in our prototype is imperfect. However, fifty-five percent of all false negatives were gestures that trace the entire shapes of 3, S, and W. These gestures are relatively complicated but have characteristic forms. Therefore we believe that the recognizer can be improved immediately.

Regarding Trace Select and the variations, as the font size decreases, initial reaction time significantly increases; the mean difference between 12 pt and 36 pt was, for all Trace Select versions, 0.13 sec. However, 12/18 users commented that finally they could select targets without looking at the glyphs carefully since they had memorized the glyphs on each characters through the experiment. Interestingly, as the distance increases, finger movement time increases whereas initial reaction time does not change. Clarifying the reason for this is a future task.

In the 2-D reciprocal selection task, there is no significant difference between Trace Select and TS-Entry-Act. In the video streaming task, the mean difference is only 0.083 sec. (without system latency), although Trace Select significantly outperformed TS-Entry-Act. On the other hand, the mean difference between TS-Show-Act and TS-Entry-Act is 0.467 sec., even though they use the same operations. It is suggested that the increase in gesture identification time needed to turn the glyphs on strongly affects the total time.

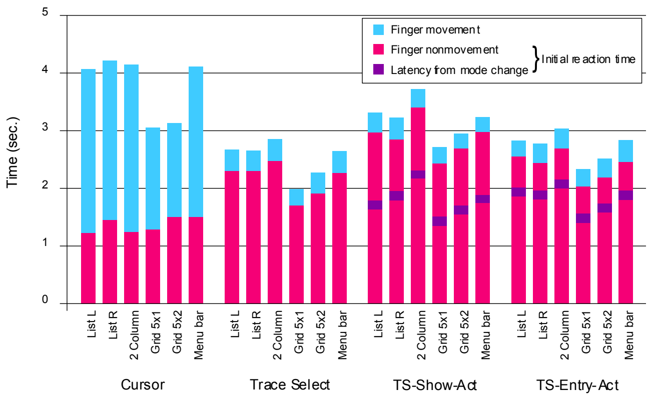

Figure 6. Selection time in Video streaming service-like selection task. Trace Select was significantly faster than Cursor and the other Trace Select variations.

Conclusion

We have presented Trace Select, which applies visually-oriented gestures with high visibility without hiding the original information of the text targets. Experiments found that Trace Select outperforms cursor-based selection enhanced by acceleration and is subjectively preferred to it, as well as the object-based target selection technique; since Trace Select eliminates the need to turn the gesture representations on/off, it outperforms the techniques that demand these steps if the other conditions are the same (TS-Show-Act). Moreover, they verified the concrete efficiency of eliminating the need to turn the gesture representations on/off: it decreased not only the number of operations needed but also the time taken to read/identify the gesture representations. In the future, we plan to internationalize the proposed technique to support non-English character sets.

References

[1] Aoki, R., Ihara, M., Maeda, A., Kobayashi, M. and Kagami, S. Unicursal gesture interface for TV remote with touch screens, In Proc. ICCE '11, (2011), 99-100.

[2] Bragdon, A. and Ko, H. Gesture Select: acquiring remote targets on large display without pointing. In Proc. CHI 2011, ACM Press (2011), 187-196.

[3] Blickenstorfer, C. H. Graffiti: wow!!!! Pen computing magazine, (1995), 30-31.

[4] Maeda, A., Inagaki, H., and Abe, M. Arrow Tag: a direction-key-based technique for rapidly selecting hyperlinks while gazing at a screen. In Proc. CHI 2009, ACM Press (2009), 1025-1028.

[5] Poirier, F. Glyph: A new stroke-alphabet for stylus-based or key-based text entry. In Proc. HCII 2005, (2005).

[6] Yatani, K., Partridge, K., et al. Escape: a target selection technique using visually-cued gestures. In Proc. CHI 2008, ACM Press (2008), 285-29.

[7] http://reviews.cnet.com/digital-media-receivers/sony-nsz-gs7/4505-6739_7-35118320.html

[8] http://www.pocket-lint.com/review/5848/panasonic-tx-l55wt50b-55-inch-smart-viera-led-backlit-lcd-tv-review

[9] http://gizmodo.com/5380388/glidetv-navigator-a-trackpad-for-your-hand