EdgeSplit: Facilitating the Selection of Off-Screen Objects

Zahid Hossain[1], **Khalad Hasan**[1], **Hai-Ning Liang**[2], **Pourang Irani**[1]
[1]Department of Computer Science,
University of Manitoba,
Winnipeg, Manitoba, Canada
{zahid, khalad, irani}@cs.umanitoba.ca
[2]Department of Computer Science & Software Engineering
Xian Jiaotong-Liverpool University
Suzhou, China
haining.liang@xjtlu.edu.cn

ABSTRACT

Devices with small viewports (e.g., smartphones or GPS) result in interfaces where objects of interest can easily reside outside the view, into off-screen space. Researchers have addressed this challenge and have proposed visual cues to assist users in perceptually locating off-screen objects. However, little attention has been placed on methods for directly selecting these objects. Current designs of off-screen cues can result in overlaps that can make it difficult to use the cues as handles through which users can select the off-screen objects they represent. In this paper, we present *EdgeSplit*, a technique that facilitates both the *visualization* and *selection* of off-screen objects on small devices. EdgeSplit exploits the space around the device's borders to display proxies of off-screen objects and then partitions the border regions to allow for non-overlapping areas that make selection of objects easier. We present an effective algorithm that provides such partitioning and demonstrate the effectiveness of EdgeSplit for selecting off-screen objects.

Author Keywords

Off-screen object visualization; off-screen target selection.

ACM Classification Keywords

H.5.2. [Information Interfaces and Presentation]: User Interfaces. Graphical User Interfaces Information.

INTRODUCTION

Maps and many other workspaces are often considerably larger than the available display on mobile devices. This often requires users to navigate and inspect objects located outside of the viewport. Off-screen visualization techniques can assist users in determining where objects of interest may be located (Figure 1a) [1,5,4]. The effectiveness of these techniques has primarily been based on how accurately users can locate or judge the relative positions of objects. While significant effort has been expended on visualizing [1, 5], less effort has been invested [6] in facilitating the selection of targets that are located in the off-screen space. Allowing users to inspect off-screen objects via their on-screen proxies can minimize unnecessary navigation. For example, instead of issuing multiple navigation commands, the user can enable selection by directly selecting the on-screen proxy.

We introduce *EdgeSplit* (Figure 1b), a novel technique that facilitates both, the visualization *and* selection of off-screen targets. EdgeSplit displays off-screen objects via proxies placed on the border of the device, in its 'radar' space. These proxies are pixel size representations of the relative locations of the off-screen objects. EdgeSplit then tessellates the 'radar' region into rectilinear polygons, thus ensuring the space designated for one target does not overlap with that of another (Figure 1b). This can alleviate cases of ambiguous selection that could occur with existing methods for off-screen visualization (Figure 1a).

(a) (b)

Figure 1. (a) Off-screen objects represented via Wedge [4] can make it difficult to select an item of interest; (b) EdgeSplit enables the visualization+selection of objects located outside of the main viewport. Off-screen objects are represented as proxy 'dots' around the border of the device. The borders are divided into rectilinear areas, with each area hosting one object and through which users can select objects.

We compared EdgeSplit to a state-of-the-art off-screen visualization technique, Wedge [4] (Figure 1a), in a selection task and in a task to identify the relative positions of objects. Results of our experiment suggest that EdgeSplit facilitates rapid and accurate selection of off-screen objects.

RELATED WORK

EdgeSplit is inspired by off-screen visualization cues and selection techniques.

Off-screen location cues

Several techniques provide location information about off-screen objects. Halo [1] draws circles around off-screen objects, part of which protrude into the viewport to give a sense for where an object is located. To mitigate the con-

MobileHCI'12, September 21–24, 2012, San Francisco, CA, USA.

cern of overlapping arcs with Halo, Wedge [4], represents an off-screen object using an acute isosceles triangle, thus providing information about the direction and distance of off-screen objects. These techniques can lead to clutter and overlap, making them difficult to use for selecting objects. Overview+detail interfaces show miniature 'dot' representations of objects in the workspace. These interfaces can occlude the main workspace. To avoid such occlusion, EdgeRadar [5] shifts the overview proxy 'dots' to the four edges of the screen (Figure 2a). The proxies convey relative distance and direction of their off-screen counterpart.

Target selection

Target selection is a well-studied area but less so for off-screen objects. *Hopping* [6] is a proxy technique that bring targets closer to the user's cursor to allow for rapid selection. Other techniques, such as *Area Cursor* [7] and *Bubble Cursor* [3], work effectively on desktop interfaces. On touch devices, touch augmentations with high-precision call-outs as used in Shift [8] allow users to select small and clustered targets. Shift's call-out opens when refined selection is necessary. As an alternative to touch enhancements, researchers have suggested that the workspace on which the targets are located be modified. Starburst [2] tessellates the workspace so that each target in a cluster has its own region. Selection can be effectuated quickly and more accurately in dense clusters with such partitioning methods. We exploit this approach in the design of EdgeSplit.

EDGESPLIT

EdgeSplit has two components. The first component is inspired by EdgeRadar [5] and provides a mechanism to visually represent off-screen objects. EdgeSplit uses four docked rectangles (or 'radars') to display proxies of objects located outside the viewport. An off-screen object is represented by a 'dot' on a radar resulting in less clutter and overlap of such cues in comparison to other techniques (e.g., Wedge [4] or Halo [1]). The second component is designed for rapid selection of off-screen objects. Given the small size of these dot-type cues, EdgeSplit tessellates the radar space along the device's edges into polygons such that each subdivided area hosts only one proxy and through which the object can be selected. This technique uses the entire space of EdgeRadar and provides users with non-overlapping areas for selection, thus making it superior to other techniques.

We implemented two space partitioning algorithms in the design of EdgeSplit. The first was inspired by the Voronoi space partitioning method, used in systems such as StarBurst [2] and Bubble Cursor [3]. Partitioning with this method however provided zones with small areas in the radar zones, making them difficult to select.

One of our contributions is a novel partitioning method, as per the following description. (1) objects are sorted according to their positions along the x and y axes; (2) vertical and horizontal bisecting lines are drawn between neighboring proxy objects, thus generating a set of partitioned cells (Figure 2b-c); (3) we assign adjacent off-screen proxies a unique color (Figure 2d); (4) each partitioned cell is assigned the color of its closest proxy (Figure 2d); (5) this is repeated for all targets (Figure 2e), thus creating a pattern seen in Figure 2f (also in Figure 1b). Each colored region hosts at most one object and its selection occurs upon touching the region.

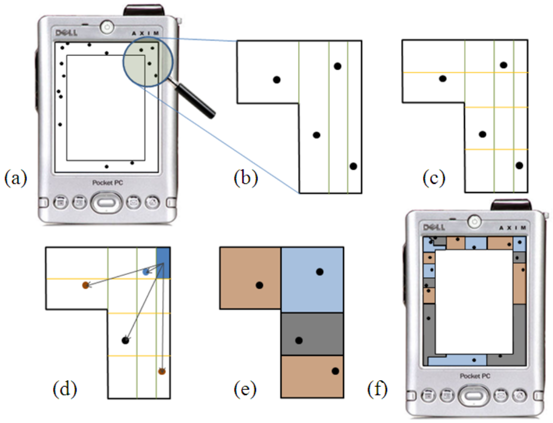

Figure 2. The partitioning algorithm used in EdgeSplit: (a) Targets are presented as 'proxy dots' along the edge of the device; (b) vertical and (c) horizontal bisecting lines are drawn between each proxy; (d) neighboring proxies are assigned a unique color; (e) all cells are filled with the color of its closest proxy; (f) the algorithm is applied to all partitioned cells.

EXPERIMENT

In this experiment we compare the performance with *EdgeSplit* to *VoronoiSplit* and *Wedge* [4] (see Figure 3). We chose Wedge because its visual cues have minimal overlap in comparison to other techniques (e.g., Halo), and provides a good size handle for selection. We did not include zooming, panning, and scrolling as these techniques often require considerable navigational effort from users. In our experiment, the intrusion depth for Wedge was equivalent to the space provided by the radar regions. We modified Wedge so that it would allow for the selection of targets, by touching directly on any region of its visible segment. We implemented all other features of Wedge (including overlap avoidance) as described in [4].

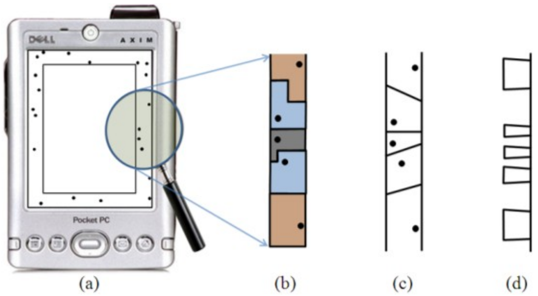

Figure 3. (a) Proxies of off-screen objects; (b-d) represented using EdgeSplit, VoronoiSplit and Wedge. Rectilinear regions in EdgeSplit facilitate selection of objects.

Methods

Apparatus

The experiment ran on Dell Axim X30 Pocket PC with a 3.5” TFT (resistive) touch screen and dimensions of 116×75×12mm. Participants could interact with regions and objects of interest through direct touch. Our techniques should provide similar performance regardless of the touch screen type. The system was developed using C# .NET.

Participants

Twelve participants (all males) between the ages of 21 and 35 were recruited from a local university to participate in this study. All participants were right-handed and had experience using mobile devices.

Design

We used a 3×2×2×2 within-subject design with four independent variables: *Technique* (*EdgeSplit; VoronoiSplit; Wedge*), *Task* (*Select*; *Closest*: finding the closest target), *Target Distance* (*Near*; *Far*), and *Target Density* (*Low*; *High*). The two distances were intended to help determine whether there would be any effect of target distance in these tasks. We also included *Target Density* to assess if the number of distractors along an edge would affect performance. Two densities were used along an edge: *Low* (1-3 distractors), and *High* (4-6 distractors). Therefore, with 3 *Techniques,* 2 *Tasks,* 2 *Target Distance,* 2 *Target Density,* 8 repetitions and 12 participants, we collected 2304 trials.

Tasks and procedure

The goal target (or object) was always placed at a random distance from the center of the screen (for *Near*, it was at most 60 pixels from the edge and for *Far*, from 60 pixels to a maximum of 240 pixels). In the first task participants were asked to *Select* the goal target (drawn in red). In the second task, participants were asked to select the off-screen object that was closest to a randomly place on-screen target (drawn in red). To provide assistance with ambiguous target selection, we implemented Shift [8]. In the *Select* task the goal target was displayed in red and the distractors in black. Participants had 25 seconds per trial to select the target. All participants were shown how each task and technique would work and received 16 practice trials per technique, with more practice trials if requested. Participants then performed the trials which were counterbalanced by techniques using a Latin Square design. All other variables were presented randomly. At the end of the study, participants filled an exit questionnaire. The experiment lasted 50 minutes.

Measures

We collected task completion time and number of attempts. *Task completion time* was the time from when participants click the Start button to when they successfully selected the target. The *number of attempts* was logged by the number of times participants attempted to select the target. In the Closest task, if participants failed to properly identify the closest target, they were requested to repeat the trial. Upon completion, participants ranked their preferences for the techniques using a 1-5 Likert scale.

Hypotheses

The properties of the three techniques led to the following hypotheses:

H1: The availability of larger selection regions in *EdgeSplit* and *VoronoiSplit* will lead users to achieve better time completion with these than with *Wedge*.

H2: *EdgeSplit* and *VoronoiSplit* make most effective use of the available space, thus requiring users to make fewer attempts to select a target with these than with *Wedge*.

Results

The data were analyzed using Repeated-Measures ANOVA and Bonferroni corrections for post-hoc comparisons.

Task Completion Time

Completion time measured the time from the target’s appearance to the time participants successfully selected it.

The RM-ANOVA yielded a significant main effect of *Technique* ($F_{2,22}$=55.91, p<0.001) on completion time. Post-hoc comparisons showed that *EdgeSplit* (M=2808ms, s.e.=126) was significantly faster than *VoronoiSplit* (M=3042ms, s.e.=144; p<0.01). In addition, both techniques were significantly faster than *Wedge* (M=4253ms, s.e.=157; p<0.001) (Figure 4a).

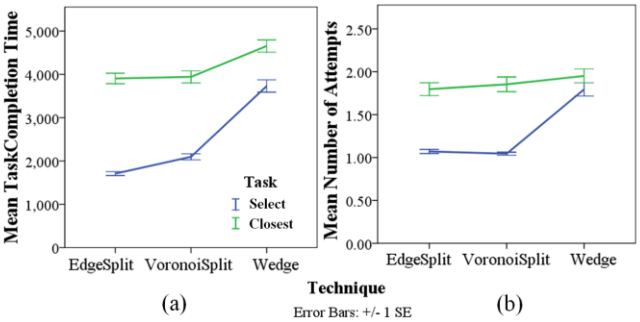

Figure 4. (a) Task completion time by technique and task; (b) Number of attempts by technique (Error bar: ± 1 SE).

There was a significant effect of *Task* on completion time ($F_{1,11}$=103.99, p<0.001), with participants performing significantly faster in the *Select* task (M=2558ms, s.e.=131) than the *Closest* task (M=4177ms, s.e.=138; p<0.001). We found a significant effect of *Target Density* on task completion time. Participants were significantly faster with *Low* density targets (M=3173ms, s.e.=121) than *High* density targets (M=3562ms, s.e.=117; p<0.001). However, we did not find any main effect of *Target Distance* ($F_{1,11}$= 0.01, p=0.93). There were significant interaction effects between *Technique* × *Task* ($F_{2,22}$=17.21, p<0.001). Participants were faster in the *Select* task for all techniques (*EdgeSplit*: M=1709ms, s.e.=101; *VoronoiSplit*: M=2099ms, s.e.=121; *Wedge*: M=3867ms, s.e.=261) than in the *Closest* Task (*EdgeSplit*: M=3907ms, s.e.=187, *VoronoiSplit*: M=3986ms, s.e.=193; *Wedge*: M=4639ms, s.e.=142).

There were also significant interaction effects between *Technique* × *Distance* ($F_{2,22}$=15.83, $p<0.001$) and *Task* × *Distance* ($F_{1,11}$=7.23, $p<0.05$). Participants performed better when targets were located at *Near* (M=2666ms) than *Far* (M=2950ms) distance with *EdgeSplit.* For *VoronoiSplit,* we found similar results with *Near* (M=2895ms) followed by *Far* (M=3189ms). However, for *Wedge*, participants were slower in selecting a target at *Near* distance (M=4523ms) than *Far* (M=3983ms). Wedge uses the size of the triangles to represent distance: the nearer the targets to the edge, the smaller the visible portion of the triangle and hence the more difficult to select (see Figure 3d above).

Number of Attempts

We counted the number of times a participant attempted to select a target and recorded these as *number of attempts* for that trial. The trial did not stop until the proper target was selected or it was timed out (after 25s).

The RM-ANOVA yielded significant main effects of *Technique* ($F_{2,22}$=15.97, $p<0.001$) on the number of attempts. Post-hoc analysis showed that *Wedge* (M=1.89, s.e.=0.1) had significantly higher number of attempts than *EdgeSplit* (M=1.44, s.e=0.07) and *VoronoiSplit* (M=1.45, s.e.=0.08) ($p<0.01$). However, there was no significant difference between the last two (Figure 3b).

There were main effects of *Task* ($F_{1,11}$=28.28, $p<0.001$) on number of attempts. Participants made more attempts in the *Closest* task (M=1.86, s.e.=0.11) than in the *Select* task (M=1.32, s.e.=0.05). However, we did not find any main effects of *Target Distance* ($F_{1,11}$=2.23, p=0.18) and *Target Density* ($F_{1,11}$=4.92, p=0.05). Finally, there was an interaction effect for *Technique* × *Task* ($F_{2,22}$=12.95, $p<0.001$). The *Select* task required fewer attempts for all techniques (*EdgeSplit:* M=1.07, s.e.=0.03; *VoronoiSplit*: M=1.04, s.e.=0.02; *Wedge:* M=1.83, s.e.=0.12) compare to the *Closest* task (*EdgeSplit*: M=1.8, s.e.=0.13; *VoronoiSplit*: M=1.85, s.e.=0.16; *Wedge*: M=1.94, s.e.= 0.1).

Subjective feedback

On a 5-point Likert scale participants preferred *EdgeSplit* (M=4.17) the most followed by *VoronoiSplit* (M=4.08) and *Wedge* (M=2). They also expressed that both *EdgeSplit* and *VoronoiSplit* (both M=3.83) were relatively easier to use than *Wedge* (M=1.83).

Discussion

Task Completion Time

Our results reveal that *EdgeSplit* and *VoronoiSplit* facilitate better off-screen target *visualization+selection* than *Wedge* (affirming **H1**). EdgeSplit afforded the best overall result (in completion times and number of attempts). This was primarily due to *EdgeSplit*'s larger selection area than *Wedge*, while its rectilinear regions provided easier access than the polygonal regions of *VoronoiSplit*.

We decomposed task completion time into: (1) *Decision time* (beginning of the trial to the time when a participant made a selection); and (2) *Selection time* (the time from the first touch to the end of the trial). We found that *Decision time* was similar across the techniques (EdgeSplit: 1766ms; VoronoiSplit: 1761ms; and Wedge: 1716ms). In contrast, there were differences in *Selection* time. Our results showed that *EdgeSplit* was the fastest in *Selection Time* for both tasks (*Locate*: 571ms; *Closest*: 1485ms), followed by *VoronoiSplit* (*Locate*: 704ms; *Closest*: 1789ms). *Wedge*, took the longest (*Locate*: 2384ms; Closest: 2566ms).

Number of attempts

Our results suggest that the total number of attempts for the *Locate* task was lower than for the *Closest* task for all techniques. For the *Locate* trials, *VoronoiSplit* exhibited the lowest number of attempts (M=1.04), closely followed by *EdgeSplit* (M=1.07). This supports **H2**. Due to the smaller selection area of the triangles, *Wedge* performed poorly (M=1.79). For the *Closest* task, the number of attempts was relatively high (from 1.80 to 1.95 attempts/trial). *EdgeSplit* had the lowest number of attempts (M=1.8), followed by *VoronoiSplit* (M=1.85) and *Wedge* (M=1.95).

CONCLUSIONS AND FUTURE WORK

In this paper, we open discussion on the selectability of off-screen cues. We propose EdgeSplit, an effective partitioning algorithm that facilitates visualization+ selection of off-screen content. EdgeSplit achieves completion times comparable to an existing state-of-the-art technique (Wedge) for identifying the relative position of objects. We also show that for off-screen object selection, EdgeSplit can facilitate improved selection over a voronoi partitioning method or against Wedge. Our results are encouraging and will allow further exploration on the effectiveness of EdgeSplit for other tasks, target layouts in corners, and real-world applications such as in GPS applications.

REFERENCES

1. Baudisch, P. and Rosenholtz, R. 2003. Halo: a technique for visualizing off-screen objects. *CHI '03*, 481-488.
2. Baudisch, P., Zotov, A., Cutrell, E., and Hinckley, K. 2008. Starburst: a target expansion algorithm for nonuniform target distributions. *AVI '08*, 129-137.
3. Grossman, T. and Balakrishnan, R. 2005. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. *CHI '05*, 281-290.
4. Gustafson, S., Baudisch, P., Gutwin, C., and Irani, P. 2008 Wedge: clutter-free visualization of off-screen location. CHI '08, 787-796.
5. Gustafson, S. and Irani, P.P. 2007. Comparing visualizations for tracking off-screen moving targets. *CHI '07.*
6. Irani, P., Gutwin, C., and Yang, X. 2006. Improving selection of off-screen targets with hopping. CHI'06, 299-308.
7. Kabbash, P. and Buxton, W. 1995. The "Prince" technique: Fitts' law and selection using area cursors. *CHI'95*, 273-279.
8. Vogel, D. and Baudisch, P. 2007. Shift: A Technique for Operating Pen-Based Interfaces Using Touch. CHI'07, 657-666.