Comparing Visualizations for Tracking Off-Screen Moving Targets

Sean Gustafson
Department of Computer Science
University of Manitoba
Winnipeg, MB, Canada
umgusta1@cs.umanitoba.ca

Pourang P. Irani
Department of Computer Science
University of Manitoba
Winnipeg, MB, Canada
irani@cs.umanitoba.ca

CHI 2007, April 28–May 3, 2007, San Jose, California, USA.
ACM 978-1-59593-642-4/07/0004.

Abstract

In games, aircraft navigation systems and in control systems, users have to track moving targets around a large workspace that may extend beyond the users' viewport. This paper presents on-going work that investigates the effectiveness of two different off-screen visualization techniques for accurately tracking off-screen moving targets. We compare the most common off-screen representation, Halo, with a new fisheye-based visualization technique called EdgeRadar. Our initial results show that users can track off-screen moving objects more accurately with EdgeRadar over Halos. This work presents a preliminary but promising step toward the design of visualization techniques for tracking off-screen moving targets.

Keywords

Off-screen visualization, moving targets, Halo, EdgeRadar, visualization

ACM Classification Keywords

H.5.2. [Information interfaces and presentation]: User Interfaces

Introduction

Tracking moving targets that may or may not be on-screen is commonly carried out in computer games, aircraft navigation systems, car navigations systems, air traffic control and military command and control systems. Studies have shown that the human perceptual system is limited to tracking fewer than six moving targets [3, 4] when these targets are visible. However, little is known about the limits of tracking targets that move outside the viewport, or off-screen.

Several techniques have been developed to help users determine the location of off-screen objects. One well known representation is the Halo [1] technique. A halo represents an off-screen object by drawing a circle centered at the object and which slightly protrudes into the edge of the viewport (Figure 1). With halos, the user can determine the location of the object in the off-screen area by mentally completing a circle from the arc visible on the edge of the screen.

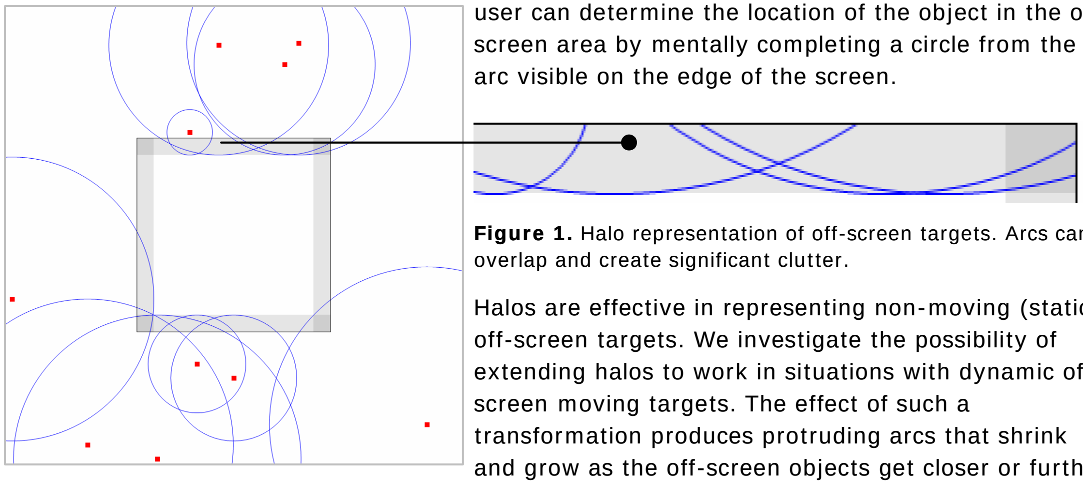

Figure 1. Halo representation of off-screen targets. Arcs can overlap and create significant clutter.

Halos are effective in representing non-moving (static) off-screen targets. We investigate the possibility of extending halos to work in situations with dynamic off-screen moving targets. The effect of such a transformation produces protruding arcs that shrink and grow as the off-screen objects get closer or further away from the edge of the viewport. However, with a large number of off-screen objects, halos can create a significant amount of clutter and make the tracking task more difficult (Figure 1).

To alleviate some of the apparent problems with halos, we designed a visualization technique called EdgeRadar (Figure 2). EdgeRadar was inspired by Radar windows used in overview+detail interfaces.

EdgeRadar creates a small overlay region on all four edges of the screen (Figure 2). In the overlay region EdgeRadar shows the off-screen objects in the corresponding off-screen region as scaled-down icons. This is analogous to a segmented radar window where each segment represents a portion of the complete environment. This also reduces on-screen clutter.

EdgeRadar was also inspired by the well known 2D bifocal lens [2]. Although similar in many ways to the bifocal display, EdgeRadar has several unique design elements. First, in EdgeRadar, the representation for each off-screen region is a transparent layer at each corresponding edge of the viewport; whereas in the bifocal lens the off-screen region is distorted to fit into the viewport. Second, EdgeRadar does not distort the visual appearance of the objects. In fisheye views, an off-screen object to the left or right of the display will scale it's on-screen representation in the x-dimension but not in its y-dimension. The result is that the objects appear distorted (i.e. an item that is square becomes rectangular under the distortion). EdgeRadar scales both dimensions of the image equally, thereby creating an object that resembles the original element. However, to do this, EdgeRadar must have semantic information of the objects, which the bifocal display does not require; i.e. the bifocal display will work well even with single bitmap images.

EdgeRadar also borrows properties from City Lights [5]. City Lights is a fisheye technique that displays a small colored bar at the edge of a window to represent off-screen objects in that direction. EdgeRadar differs from this technique by displaying a 2D representation of the off-screen space (instead of 1D) with miniature icons resembling the off-screen objects.

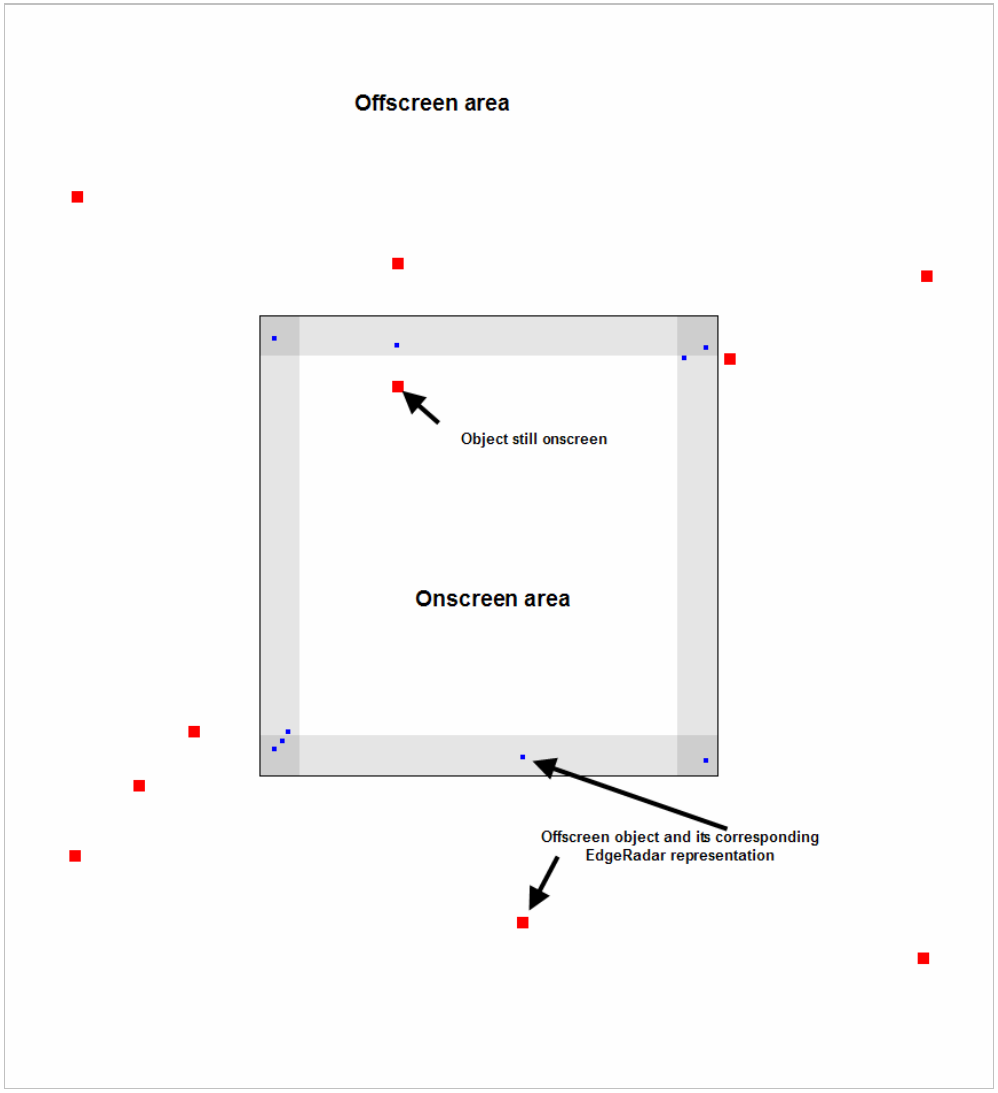

Figure 2. EdgeRadar representation showing various off-screen objects.

Although EdgeRadar resolves certain problems that exist with prior techniques it also has several limitations. First, it lacks information with respect to the absolute location of the object. It primarily provides directional information and relational distance between objects. Although many tasks do not require knowledge of exact positioning of off-screen objects, EdgeRadar may not be suitable for those tasks that do require this information. Second, there is no graceful degradation of the visual representation when the object approaches the edge of the represented workspace—it simply disappears. Third, the corner visualizations represent a larger off-screen area than the sides. This mapping is disproportionate and can be problematic if the off-screen area is significantly larger than the viewport. Despite these limitations, we believe that EdgeRadar can facilitate certain tasks that are required for tracking multiple moving off-screen objects.

Experiment

We conducted a preliminary experiment to compare and evaluate the effectiveness of EdgeRadar and Halo for tracking off-screen moving targets. This task does not require explicit knowledge of the targets' location. We hypothesized that it would be easier to track off-screen targets when they are represented using EdgeRadar, compared to Halo. We were also interested in any interactions between visualization type, object speed, number of targets and number of distractors.

SUBJECTS

We recruited six volunteers from a local university to participate in the experiment. All subjects were between 21 and 30 years, were frequent computer users and were exposed to computer gaming.

APPARATUS

An experimental application was developed in C#.Net which implemented the techniques, controlled the experiment and gathered the results. The experiment was conducted on a Windows XP computer with a 1280x1024 pixel 17” monitor.

INDEPENDENT VARIABLES

Four independent variables were used in the experiment:

- Visualization technique: Halo and EdgeRadar;
- Speed of objects: 60 and 180 pixels/sec;
- Number of target objects: 2, 3, and 4;
- Number of distractor objects: 3 and 6.

Each subject repeated all conditions twice. Half the participants started with the Halo condition and the other half with EdgeRadar.

TASK

To test our hypothesis we replicated the Multiple Object Tracking (MOT) task designed by Plyshyn [3] and used in most studies investigating multiple target tracking [4]. In the MOT task, participants are asked to track several moving targets, which are initially shown by flashing them at the start of the trial for a brief period of time. Participants are required to keep track of the flashing objects. Once they stop flashing all targets and distractors start moving. Objects can bounce off each other. In the typical MOT task objects can also bounce off the edges of the screen. However, in our task objects can move outside the viewport. At the end of a trial a randomly picked object is highlighted and the participant has to determine whether that object appeared in the initial set of flashing targets.

PROCEDURE

The experiment was conducted in a lab setting. At the start of each session the subject was given a short tutorial on the experimental task. The visualization techniques were explained and the subject was able to perform several practice trials. The experiment began when the subject indicated that they were comfortable with the environment and the task.

The subject was presented with a 950x950 pixel white panel, which represented the entire workspace region. In the center of this panel was a separate 400x400 pixel panel, which represented the on-screen area. The region outside the internal panel was considered off-screen. All visualizations were displayed along a 35 pixel border of the on-screen panel. When an object travels out of the viewport panel it was no longer visible but instead its on-screen representation (either Halo or EdgeRadar) was shown. Objects were depicted as 10x10 pixel red squares and both visualizations were drawn in blue. Figure 2 shows the screen used in the experiment (the locations of off-screen objects were only shown at the end of the trial—described below).

At the start of each trial all objects were randomly placed in the on-screen area. The targets were highlighted with a thick black square for several seconds and then flashed 5 times. After the flashing completed all of the objects began moving in their predefined but random directions. All objects traveled at the same speed. Objects bounced off each other and off of the edge of the entire workspace. The objects returned onto the viewport if their particular trajectory led them there.

After 10 seconds the objects stopped moving and their current position was shown in the workspace, including the off-screen area, as in Figure 2. At this time, one object was randomly highlighted with a bounding box and the participant was asked if this object was one of the originally flashing targets. The user entered their response (Y/N on the keyboard) and the time to make a response was recorded.

The object chosen to be highlighted at the end of the trial had a 50% chance of being a target. Therefore the subject was expected to enter an incorrect response no more than 50% of the time.

Each participant performed: 2 visualization techniques x 2 speeds x 3 levels of targets x 2 distractor levels x 2 trials/condition = 48 trials. In total with 6 participants, 288 trials were recorded.

At the end of the session, the participant was given a short questionnaire on which they were asked to rate their subjective performance with each technique on a 5-point Likert scale.

Results and Discussion

The results of the experiment showed an overall increase in accuracy when the EdgeRadar technique was used compared with the Halo technique. Overall users had an error rate of 16.67% for EdgeRadar in comparison to 21.53% for Halo. This difference was not statistically significant (F(1,10)=.803, p=.391). This is primarily due to the low number of participants and trials. With more participants and more trials we believe the difference in means will be significant.

Comparing the accuracy in response between the techniques for different number of targets, we found that although EdgeRadar outperformed Halo in all cases there was no increase in performance as the number of targets increased, see Figure 3.

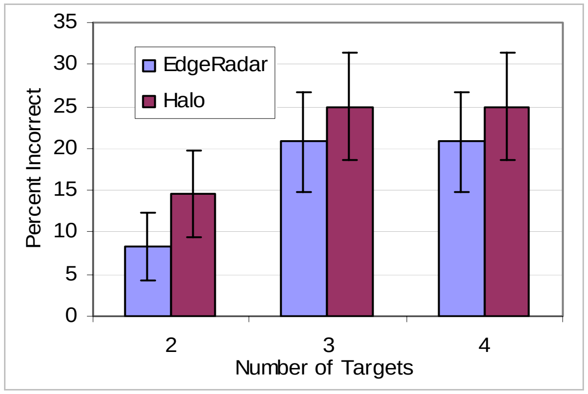

Figure 3. Error rate by number of targets.

With respect to increasing the number of distractors, we found no clear pattern when comparing the error rate with 3 distractors to those with 6 distractors. This suggests that participants were able to easily ignore 6 distractors. Future work will use a larger number of distractors to determine at which point this becomes critical.

Comparing error rates for different target speeds, we observed a similar performance increase with EdgeRadar for both speed levels as shown in Figure 4, i.e. users responded correctly more often with EdgeRadar than with Halo. The differences were not significant, however the trends do suggest that EdgeRadar may reduce the overall complexity in tracking targets off-screen.

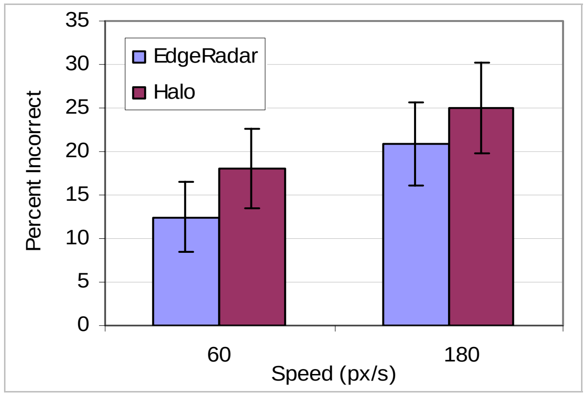

Figure 4. Error rate by target speed.

We were unable to gain much insight from analyzing the time required to make a response as the results were not showing any clear patterns. This was perhaps due to the high variability in responses resulting from the limited number of participants in this study.

The questionnaire results were analyzed and the participants had a slightly higher preference for EdgeRadar (3.17/5.0) over Halo (2.5/5.0). Both values were quite low suggesting that participants found the task difficult in general and thought the visualization techniques did not sufficiently assist them.

Comments from participants gave some insight into the low scores. One stated that although he believed that he performed better with EdgeRadar, he felt more comfortable with Halo. Another participant stated that Halo was easier to track with fewer objects but lost this benefit when there were many objects being tracked.

Conclusion and Future Work

This study is an initial step toward the design of new techniques for tracking off-screen moving targets. The results of the experiment showed an increase in accuracy using the EdgeRadar technique vs. Halo. While the trends in the averages show clear advantages for EdgeRadar over Halo, a more extensive study is needed to determine if the results are generally applicable. However, we believe (but need to follow up more rigorously on this claim) that radar-like visualizations can offer clear benefits when tracking moving targets in off-screen space. In future work, we would like to investigate the effect on task performance in more realistic scenarios. This would include a map background, standard PDA sized viewport, and other tasks such as those that use knowledge of an object's exact location. We would also like to investigate other novel off-screen visualization designs including incremental improvements to existing techniques.

References

[1] Baudisch, P. and Rosenholtz, R. 2003. Halo: a technique for visualizing off-screen objects. *CHI '03*, 481-488.

[2] Leung, Y. K. and Apperley, M. D. 1994. A review and taxonomy of distortion-oriented presentation techniques. *ACM TOCHI,* v1(2), 126-160.

[3] Pylyshyn, Z.W. and Storm, R.W. 1998. Tracking multiple independent targets: Evidence for a parallel tracking mechanism. *Spatial Vision*, v3(3), 179-197.

[4] Yantis, S. 1992. Multielement Visual Tracking: Attention and Perceptual Organization. *Cognitive Psychology*, v24(3), 295-340.

[5] Zellweger, P. T., Mackinlay, J. D., Good, L., Stefik, M., and Baudisch, P. 2003. City lights: contextual views in minimal space. *CHI '03 Ext. Abstracts,* 838-839.