
An Evaluation of Techniques for Selecting Moving Targets

Tyler J. Gunn

Department of Computer Science
University of Manitoba
Winnipeg, MB, Canada
tygunn@cs.umanitoba.ca

Pourang Irani

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

John Anderson

Department of Computer Science
University of Manitoba
Winnipeg, MB, Canada
andersj@cs.umanitoba.ca

Abstract

Moving targets are found in numerous applications such as computer games, air traffic control systems, and video surveillance. The selection of moving targets is considerably more difficult and error prone than traditional stationary target selection. In this paper, we introduce *Comet Tails* and *Target Lock*, two techniques that support the selection of moving targets. Our goal is to facilitate accurate and fast selection of moving targets. We compare our two techniques to unassisted selection in a controlled experiment. The results show that for moving target selection, *Comet Tails* and *Target Lock* can outperform unassisted selection, and result in fewer errors. According to post-experiment questionnaires, participants indicate a stronger preference for assisted target selection with *Comet Tails* and *Target Lock* than unassisted selection.

Keywords

Moving target selection, Fitts' law, target selection.

ACM Classification Keywords

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

Introduction

Animated visuals are ubiquitous and are found in a number of applications, such as simulations, air traffic control systems, video surveillance systems, and computer games. A fundamental task in many of these systems requires that the user select moving targets. In a computer game, for example, a user might select a

moving battle tank in order to give it new deployment orders, while an air traffic controller might need to select an airplane to be able to view its flight plan, and a security guard might wish to select a suspicious person on a video surveillance screen to review recordings of their past locations.

Fitts' law [2] provides a performance model for a human performing a physical pointing task. MacKenzie [6] studied the applicability of Fitts' Law in modeling the performance of a human performing the Fitts' pointing task in a virtual world. One of the advantages of considering the Fitts' pointing task on a computer is that the task takes place in the virtual world of the computer system [1]. Unlike pointing in the physical world, on a computer we can introduce enhancements to the pointing process that would not otherwise be possible in the real world. While the selection of moving targets has garnered considerable interest within other research groups in cognitive science and in psychology [4,5], very little work in HCI has concentrated on the selection of moving targets.

In this paper we introduce the task of selecting a moving target and describe some of the challenges it presents over and above those involving the selection of stationary targets. We introduce two virtual enhancements to improve the selection of moving targets. We carry out a study with eleven volunteers and the results illustrate that these virtual enhancements can definitely assist a user in selecting a moving target.

Moving Target Selection

To facilitate the study of moving target selection, we first describe how this task would be accomplished using standard unassisted mouse pointing. In this task,

the workspace viewport contains several targets distributed within the window. The targets move on a path within the window area. When a target collides with the side of the window the target bounces off it. To select a target, the user must position the pointer on top of the target and click the left mouse button. Although the user can select any number of targets, we can assume that the user will only be interested in one particular target at any one time.

Selecting a moving target is difficult because the user must continually track the target as he/she moves the cursor towards it. If the user stops moving the cursor to click the mouse button, the target may move away from the pointer before the mouse click is registered by the system. This task is made even more difficult due to the point-and-click model used in Microsoft Windows. Clicking a target involves pressing down the left mouse button on top of the target, and then releasing the left button. However, the pointer must still be within the target when the user releases the left mouse button for the selection to take place. When targets are moving, it is possible for the target to move away from the pointer while the left mouse button is pressed down, meaning when the button is released the selection does not take place.

We designed two techniques, *Comet Tails* and *Target Lock* to alleviate some of the difficulties that occur in selecting moving targets. We also implemented *Bubble Cursor* [3], as an alternative to these techniques. However, the moving targets caused continual size changes in the cursor, which was found to be distracting to users in preliminary experiments. We ultimately did not evaluate moving target selection with *bubble cursor*.

Comet Tails

The *Comet Tails* pointer enhancement makes it easier for a user to select a moving target by displaying a comet tail behind the target. The comet tail provides a larger area in which the user may aim the pointer, effectively enhancing the Fitts' Law *target width*. This mechanism is inspired by techniques that increase the effective width of a target, such as with the bubble cursor [3].

Figure 1 illustrates how the comet tails enhancement works. Each target has a trigger area larger than the comet tail – this is shown as a blue outline in the Figure. When the pointer enters the trigger area, the grey comet tail becomes visible. Finally, when the user moves the pointer inside the comet tail, it gets highlighted. Clicking the mouse button inside the comet tail selects the target. A comet tail consists of a head and tail. The size of the comet head and comet tail depend on the size of the target. Thus, a larger target will have a larger comet tail and head. Initial pilot tests revealed that a comet tail 4.3 times the width of the target resulted in comet tails that were visually appropriate compared to the target size.

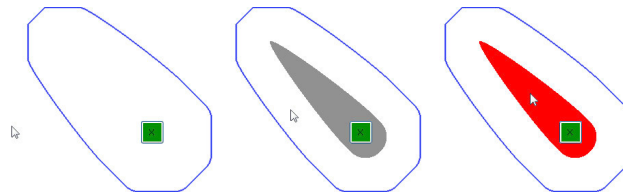


Figure 1. *Comet Tails* walkthrough. (1) The comet tails are normally hidden. The blue line represents the "trigger area". (2) Moving the pointer inside the trigger area shows a grey comet tail. (3) The user highlights the target by moving the pointer inside the comet tail. Clicking inside the tail selects the target.

Target Lock

The *Target Lock* enhancement makes it easier to select a moving target by eliminating the need to click within a moving area. Once a user highlights a target the Fitts' Law *target width* in effect becomes infinite, as the user can click anywhere to complete the selection. Figure 2 illustrates how the *Target Lock* enhancement works. Initially no target is highlighted. As the pointer moves over top of a target a highlight appears around the target. The target remains highlighted even after the pointer moves outside of the target. To select the highlighted target, the user clicks the left mouse button. The pointer does not need to be within the highlighted target for selection to take place. If the user wishes to highlight a different target, they move the pointer over top of the target they wish to highlight instead. The highlight effect for the target lock technique depends on the size of the target. The highlight is 26 pixels in diameter larger than the width of the target. Thus, smaller targets have a smaller highlight.



Figure 2. *Target Lock* technique walkthrough: (1) User moves the pointer toward a target. (2) Moving the pointer over the target highlights the target. (3) Clicking the left mouse button selects the highlighted target.

Experiment

We conducted a controlled experiment to compare *Comet Tails* and *Target Lock* to unassisted selection of moving targets. Based on the properties of our techniques introduced previously, we hypothesized the following:

1. Both the *Comet Tails* and *Target Lock* enhancements will result in greater reduction in selection time compared to Unassisted selection.
2. Both the *Comet Tails* and *Target Lock* enhancements will result in fewer click errors than unassisted selection.
3. The *Target Lock* technique will result in fewer click errors than the *Comet Tails* technique.

Subjects

Twelve volunteers between 20 and 40 years of age participated. All participants were professionals at a local company, are frequent mouse users and occasionally play computer games. One of the twelve participants gave up on the first trial, out of apparent frustration with unassisted selection – his information has been excluded from the experimental results. None of the participants were color blind.

Apparatus

The experimental application is a standard Windows Forms application, developed in C#.Net. The experiment ran on a Windows XP PC equipped with a 19" LCD monitor and a Microsoft Mouse.

Independent variables

To facilitate comparison of the three techniques, a within-subject experiment design was used. The experiment made use of four independent variables, combined in a factorial manner to generate the conditions of the experiment:

- Selection technique: *Unassisted*, *Comet Tails* and *Target Lock*
- Target size: 35 pixels, 50 pixels
- Target speed: 180 pixels/sec, 360 pixels/sec
- Number of moving targets: 1 moving target, 13 targets moving

The slower target speed approximates the speed of vehicles moving on residential roads as viewed from a video surveillance feed, while the faster speed provides a target speed more typical of action video games.

All subjects performed 5 trials with each condition, yielding 120 trials per subject. Participants were given practice trials with each technique. With 11 participants, we collected a total of 11x120 or 1320 trials.

Task

We developed an application that requires the user to perform moving target selection. The user must select a moving green target from among twelve grey distractor targets.

Experimental setup

There are a total of thirteen targets in the experimental implementation. The targets always begin in the same locations and have pre-determined movement directions (see Figure 1). In trials where there is a single moving target, only the target that the user must select is moving.

There are nine pre-defined targets that the system can randomly choose to be tracked and selected by the user during a trial. This target is colored green. The position of the trial *Start* button changes to ensure that the pointer is 550 pixels away from the center of the target, and that the target is moving away from the pointer at the start of the trial.

The trials are grouped together by technique, and the order of presentation is counterbalanced between participants in a Latin Square order. At the start of each group of techniques, the program displays an instruction screen with a stationary example of the technique. The user is provided with the opportunity to practice with three training trials, and is given a

reference sheet explaining the techniques. Before participating, each subject was given a short introduction and explanation of the three different techniques.

Measures

We collected the trial completion times, pointer movement distance and error rates. Trial completion time was determined as the time from when the user clicked the *Start* button to when they selected the required target. Errors were recorded if the user clicked the mouse but failed to select a target, or if the user accidentally selected a distractor target. Upon completing the experiment, we asked each participant to fill out a questionnaire where they ranked the three techniques according to *preference*. They were also asked to provide comments and feedback.

Results and Discussion

We first analyzed the results for one and multiple moving targets separately. We found no significant difference in performance time or error rate, thus we combined the results for both conditions and present them below. We used the univariate ANOVA test and Tamhane post-hoc pair-wise tests (unequal variances) for our analyses with subjects as random factor.

Completion Time

There was a significant effect of *technique* ($F_{2,20} = 7.714, p=0.003$), of *target size* ($F_{1,10} = 7.122, p<0.05$) and *target speed* ($F_{1,10} = 16.180, p=0.002$) on completion time. We also found significant interaction effects for *technique*×*target size* ($F_{2,20} = 4.955, p=0.018$) and for *technique*×*target speed* ($F_{2,20} = 8.197, p=0.003$). Post-hoc pair wise comparisons reveal significant differences between *Unassisted* and *Target Lock*, and *Unassisted* and *Comet Tails*. Users completed the task on average 1633.02 msecs,

1682.944 msecs, 3314.25 msecs with *Target Lock*, *Comet Tails* and *Unassisted*, respectively. Figure 3 (left) shows average completion time for each technique by target speed.

Error Rate

We define an error when the user clicks but does not acquire the target, or when the user clicks on a distractor target. There was a significant effect of *technique* ($F_{2,20} = 12.828, p<0.001$), of *target size* ($F_{1,10} = 9.323, p=0.012$) and *target speed* ($F_{1,10} = 12.249, p=0.006$) on completion time. We also found significant interaction effects for *technique*×*target size* ($F_{2,20} = 9.361, p=0.001$) and for *technique*×*target speed* ($F_{2,20} = 12.566, p<0.001$). Post-hoc pair wise comparisons reveal significant differences between *Unassisted* and *Target Lock*, and *Unassisted* and *Comet Tails*. Users on average obtained an error rate of 1.38 clicks and for *Unassisted* vs. .032 for *Comet Tails* and .011 for *Target Lock*. Figure 3 (right) shows errors for each technique by target speed.

Discussion

Both the *Comet Tails* and *Target Lock* techniques resulted in a significant decrease in the average time it took for users to select a target. Overall unassisted selection results in a higher percentage of trials having an error. With no assistance, participants usually made multiple errors. The comet technique resulted in a much lower percentage of trials having errors, and when errors occurred it was usually only 1-2 per trial. The target lock technique resulted in the lowest number of trials with errors overall.

The poor performance of the *Unassisted* technique is likely due to the fact that when targets are small or fast moving, participants had a difficult time clicking inside

the target. Many participants reported resorting to rapid clicking in order to select targets unassisted. Although the comet technique resulted in a lower number of errors than unassisted, the target lock technique proved to result in the fewest errors overall. Some participants reported that the comet tails (especially with the larger target sizes) got in the way, which could explain the number of errors seen.

The responses to participant surveys indicate that both techniques provide a valuable aid when selecting moving targets. Although there is a definite preference towards the target lock technique, both techniques ranked favorably. Both techniques offer many future directions for enhancement and study, as well as the opportunity to combine favorable elements of both into a further new technique. We also plan to model the selection of moving targets with and without virtual enhancements.

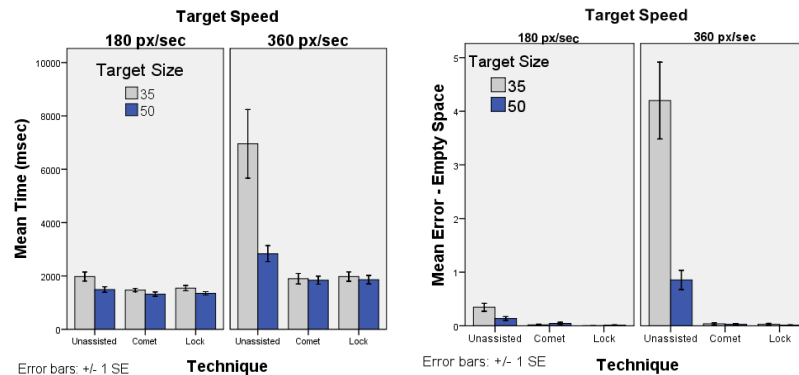


Figure 3. (left) Average completion time; (right) Average Number of clicks.

Conclusion and Future Work

Although both the *Comet Tails* and *Target Lock* techniques rated favorably in the participant responses, the results of the experiment indicate that there is still

potential room for improvement. Some participants felt that the *Comet Tails* were too large. A future version of *Comet Tails* could include user-adjustable comet tail width and length values (inspired by the *Bubble Cursor*).

With the *Target Lock* technique, the most common errors were due to the user selecting a distractor target. This is likely due to a distractor moving beneath the pointer after the participant already highlighted the correct target. A possible solution would be to prevent the highlight from changing if the pointer is not moving. Selecting moving targets is a task that becomes more difficult as the size of targets decreases, and the speed of the targets increase. The *Comet Tails* and *Target Lock* enhancements both result in a significant improvement in the time it takes a user to select a moving target. Although both techniques resulted in a similar reduction in the time to complete a trial, the target lock technique appears to facilitate the target selection process in a more error-free manner.

References

- [1] Balakrishnan, R. (2004) "Beating" Fitts' law: virtual enhancements for pointing facilitation. *International Journal of Human-Computer Studies*, 61(6):857-874.
- [2] Fitts, P. M. (1954) The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47:381-391.
- [3] Grossman, T. and Balakrishnan, R. (2005) The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. *ACM CHI '05*, 281-290.
- [4] Hoffmann, E. R. (1991) Capture of moving targets: A modification of Fitts' law. *Ergonomics*, 34, 211-220.
- [5] Jagacinski, R. J., Repperger, D. W., Ward, S. L., & Moran, M. S. (1980) A test of Fitts' law with moving targets. *Human Factors*, 22, 225-233.
- [6] MacKenzie, I. S. (1991) Fitts' law as a performance model in human-computer interaction. PhD thesis, University of Toronto, Toronto, Ontario, Canada, 1991.