Comet and Target Ghost: Techniques for Selecting Moving Targets

Khalad Hasan[1], Tovi Grossman[2], Pourang Irani[3]

[1]Department of Computer Science
University of Manitoba,
Winnipeg, Manitoba, Canada
khalad@cs.umanitoba.ca

[2]Autodesk Research
210 King St. East,
Toronto, Ontario, Canada
tovi.grossman@autodesk.com

[3]Department of Computer Science
University of Manitoba,
Winnipeg, Manitoba, Canada
irani@cs.umanitoba.ca

ABSTRACT

Numerous applications such as simulations, air traffic control systems, and video surveillance systems are inherently composed of spatial objects that move in a scene. In many instances, users can benefit from tools that allow them to select these targets in real-time, without having to pause the dynamic display. However, selecting moving objects is considerably more difficult and error prone than selecting stationary targets. In this paper, we evaluate the effectiveness of several techniques that assist in selecting moving targets. We present Comet, a technique that enhances targets based on their speed and direction. We also introduce Target Ghost, which allows users to select a static proxy of the target, while leaving the motion uninterrupted. We found a speed benefit for the Comet in a 1D selection task in comparison to other cursor and target enhancements. For 2D selection, Comet outperformed Bubble cursor but only when Target Ghost was not available. We conclude with guidelines for design.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

General terms: Design, Human Factors.

Keywords: Moving targets selection, Comet, Target Ghost.

INTRODUCTION

Animations consisting of moving targets are ubiquitous and are found in applications such as video surveillance systems, molecular simulations and air traffic control displays. Pointing is a fundamental task in direct manipulation interfaces and users can benefit from being able to directly select one or more moving targets. For example, an air traffic controller may select an airplane to view its flight plan. Similarly, video tracking software can allow users to select objects in real-time, to retrieve statistics on a basketball player, in mid-play.

Selecting a moving target is challenging. The user must continually track the target and simultaneously plan to move the cursor over it. If the user stops moving the cursor, clicking on the mouse button for selection could cause the target to slip away from the pointer. Currently, we possess limited knowledge on which interaction techniques can best support this task.

One option is to allow users to pause the motion for selection. However, this may cause viewers to miss important information [5] in certain real-time applications such as air traffic control systems or video games.

Figure 1 – Moving objects can be enhanced with a Comet to enlarge their activation area. We show its value in the case of a video feed (vehicle paths were annotated manually). When a car is selected more information (top left corner) is available.

In this paper we evaluate the performance of various existing and new techniques that can assist in selecting moving targets. We present *Comet* and *Target Ghost*. Comet enhances a target based on its movement trajectory and speed to facilitate selection. With Comet the target's activation size is increased with an appended tail, analogous to comets that are seen in the sky. Faster targets have longer tails than slower targets, thus leveling the field for selecting targets with various movement speeds (Figure 1).

We implement a variation of pausing by designing Target Ghost, a proxy-based technique [2, 4]. Target Ghost creates static proxies of all the objects in the scene based on their position at time of invocation, but does not disrupt the overall movement of objects. Users can then select the motionless targets. Ghosting can be used in conjunction with new or existing techniques, such as the Area cursor [16] and Bubble cursor [11].

We conducted a 1D moving target acquisition experiment, and found that user performance was significantly improved with Comet over the Bubble cursor and variants of Target Ghost, including the Basic and Bubble Ghosts. In a 2D experiment, we also manipulated the predictability of the object's movement path, to create conditions where objects' movements were highly predictable (i.e. straight line) or less predictable (i.e. random movement). Our results reveal that Comet still outperforms the Bubble cursor, but only without using it with Target Ghost.

Our contributions in this work are: 1) an evaluation of competing interfaces for selecting moving objects; 2) designs of Comet and Target Ghost that work in 2D; 3) techniques for selecting targets with high and low predictability movement paths; 4) a set of guidelines for assisting interface designers in supporting moving target selection.

LITERATURE REVIEW

Our work is inspired by work done on the selection of motion-less and motion-based targets.

Selection of motionless targets

Pointing tasks with motionless targets can be generally captured by Fitts' law [10, 17]. The movement time T needed to point to a target of width W and at distance (or amplitude) A can be expressed as:

$$T = a + b \log_2 (A/W + 1) \qquad (1)$$

where a and b are empirically determined constants [17]. The logarithmic term log (A/W + 1) is referred to as the Index of Difficulty (ID) (measured in "bits") of the task. Researchers have also extended Fitts' law to model two- and three dimensional targets [17] and also to targets of varying shapes [12]. Based on the above equation, performance benefits can be obtained by increasing the width of the target, decreasing its amplitude, or by doing both. Since virtual pointing tasks are less constrained than pointing in the physical world, researchers have shown that altering the properties of the virtual object or of the cursor can lead to significant performance benefits [1].

The Area cursor [16] converts a point cursor by giving it a larger width and thereby increases its effective activation area. The Area cursor has been shown to follow the properties of Fitts' law where the width of the target is instead replaced by the width of the cursor [16] in equation (1).

Grossman and Balakrishnan [11] built upon the concept of the area cursor with the introduction of the Bubble cursor. Their technique dynamically resizes the cursor's activation area based on the number and proximity of targets, and ensures that only one is selected from any given cursor position. The Bubble cursor is known to be effective under a variety of contexts [11]. However, there has not been any systematic evaluation of the bubble or area cursors for selecting moving targets.

One concern in using cursor enhancements for selecting moving targets is their lack of precision in a dense and dynamically changing field of targets. This would require the cursor to constantly adapt to disambiguate selection. An alternative to enhancing the cursor is to change the target properties. Such enhancements are generally based on alterations to the target itself [18, 19], to the target's position [2, 4], or even on the workspace [3].

An example of such target enhancements is to expand its size (as with buttons on some operating systems). McGuffin and Balakrishnan [18] found a clear advantage to expanding targets, even when the expansion started after 90% of the total movement to the target. However, when targets are densely populated, target expansion creates problems with targets overlapping one another [18]. To mitigate this problem, alterations can be made to the workspace. Baudisch et al. [3] presented Starburst, a space-partitioning algorithm that identifies the available areas of screen space and places each target into a different tessellated region on the screen. The user can simply click on the associated region of the workspace to select the target. For this method to work with dynamically refreshed targets, the partitioning would have to constantly occur, thus changing the visual and motor properties of the workspace as targets move around. Our design of Comet is inspired by target expansion techniques as it enlarges the target's activation area.

Instead of modifying properties of the target one can reduce the distance to the target. In drag-and-pop [2] and the vacuum filter [4], the user can invoke a replica of the target that is placed in proximity to the cursor, thus shrinking A in equation (1). The user then needs to only move the cursor minimally to select the target. These techniques have shown benefits for selecting targets at large distances. In the case of moving targets, shrinking the target distance as it moves away is an appealing solution, and one we used as a basis for our Target Ghost techniques.

Selection of moving targets

Researchers in the fields of cognitive psychology have explored users' performance in selecting moving targets. Jagacinski et al [15] demonstrated that selection of moving target is highly correlated to the velocity of the object and offered an analytical model for movement time estimation:

$$T = a + bA + c(V+1)(1/W-1)$$

where A is the initial amplitude, V is the target velocity, W its width, and a, b, c, are empirically determined constants.

Despite the growing abundance of data displayed in a dynamic manner, only a few studies in HCI have explored the selection of moving targets. Faure et al [9] investigated the acquisition of pop-up and animated targets, as found on some operating systems such as the Mac OS X. They examined the effects of different transition and animation delays on pop-up targets and found no significant difference with selecting static targets.

Ilich's [14] recent investigation into the selection of moving targets was done on interactive video browsing. Ilich showed that when pausing the entire scene with a Click-to-Pause technique, the selection of moving targets comes close to that of static targets. Click-to-Pause involves first depressing the mouse button to pause all onscreen moving objects; once the cursor moves over the item of interest, releasing the button selects it. In a user evaluation, Ilich found that Click-to-Pause results in lower selection times than the unassisted pointer for small and/or fast targets [14]. Target Ghost was partly inspired by this approach, but instead of pausing the scene we created static in-place proxies to aid selection. This has the added benefit of working in environments where pausing is not possible or when the users do not wish to not miss any information [5].

DESIGN CONSIDERATIONS

A number of factors can influence the design and performance of techniques for selecting moving targets. We outline those that have guided the design of the Comet and Target Ghost. These include: target speed, movement direction, movement type and feedback type.

Target speed

Studies on moving targets [13, 15, 21] show that the objects' speed is a strong determinant of targeting performance. In Jagacinski's model [15], the index of difficulty is directly proportional to target velocity (faster targets are more difficult to select than slower ones). It is possible that new enhancements would only help in acquiring faster targets instead of slower ones. Our techniques dampen the impact of fast movements by either increasing the target's activation area or by creating static proxies of the targets.

Movement direction

In a study by Tresilian and Lonegram [21] participants were required to select a moving target by intercepting it, by hitting a ball with a baseball bat. This task was found to be considerably different than prior work on moving targets [13, 15] where participants were instead pursuing the targets. Tresilian's model also differed considerably than Jagacinski's and was attributed to the different motor control processes between both task types. These results suggest that the direction of motion, i.e. away, toward or orthogonal to the cursor can result in different targeting strategies and motor control movements. This factor played an important role in the design of 2D Comet's design.

Movement type

Movement type is a multidimensional attribute consisting of the level of predictability of a target's trajectory vector, the shape of the path (straight, curved, random), and the rate of changes in these. For example in the video feedback of a hockey game, players can move in highly unpredictable paths, changing directions at random moments. There are numerous types of movement paths, either in nature or simulated (i.e. in video games) and the design of techniques for acquiring moving targets needs to consider this factor.

Feedback type

Target feedback type can also affect the selec-tion of moving targets. Mould and Gutwin [20] investigated target feedback on multiple moving objects in a gaming environment. They compared selection performance for conditions of no target feedback, feedback on the target only and feedback on all items. Feedback on the target or on all items significantly improved task completion. Since feedback on all targets can result in too many distracters, target-only feedback was proposed as being practical.

TECHNIQUES

We describe our designs of Comet and Target Ghost.

The Comet

The *Comet* is based on the concept of target expansion used in static environments [18]. The technique has a physical resemblance to the astronomical comet that displays a trailing blaze of dust and ice as it moves along the sky, and is similar to the past position trails displayed in some air traffic control systems. The design of comet tail was motivated by motor control theory which proposes that target acquisition consists of an initial ballistic movement, followed by corrective sub-movements [10]. With a tail length proportional to the target speed, we could improve the chance that after the user's initial ballistic movement towards the original target position, the cursor would be in the general region of the comet, even if the target had moved away.

Figure 2 illustrates how the Comet behaves. As shown, each target has a tail whose size is based on the speed and width of a target. A slower target will have a shorter tail than a target that is moving faster. This means that if the user aims at the original target position, the cursor will land on the tail if it misses it. Similarly the thickness of the tail is dictated by the width of the target. For example, a target moving at a speed of 500 pixels/sec and of size 50 pixels had a Comet tail of 337 pixels (where Tail length = Speed/c + Width/2, where c is a constant scale factor in our case, 1.6 which was tuned after initial pilot testing). When the cursor enters the trigger area, the tail becomes a solid object, denoting that it is selectable (Figure 2.b). Furthermore, if the tail overlaps with another target, it is rendered below the target so it does not occlude the target (Figure 2.c).

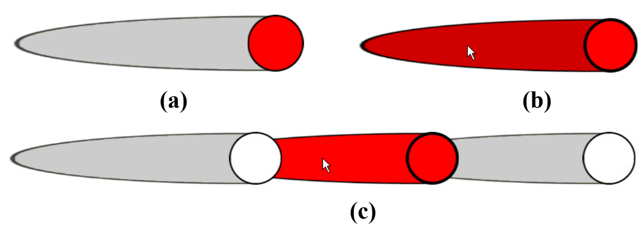


Figure 2. (a) Target and its comet tail; (b) the tail gets highlighted as the cursor moves over it; (c) tails can be overlapped by adjacent targets.

Target Ghost

Target Ghost is inspired by proxy-based techniques [2, 4] that bring the target closer to the user's cursor. The exception here is that the proxies do not come closer to the cursor, since this would cause clutter and thus require additional layout algorithms. Instead, as the targets are ghosted, they leave behind a proxy in the position at the moment when the user invoked the *Ghost* (Figure 3). The ghosted targets are rendered in a dimmer shade making them less visible, but moving along their paths. Showing continued motion is critical in environments where users cannot stop the scene or wish to not lose continuity between frames [5].

Ghosting is invoked by pressing the Shift key with the non-dominant hand. To select a target users simply move their cursor over the target's proxy and select it with a mouse click. If the user does not select the target and instead releases her finger off the Shift key, the targets get un-ghosted and continue moving along their trajectory. Since Target Ghost is not a cursor based enhancement, any cursor technique can be applied in conjunction with Target Ghost, to create a Bubble Ghost or Comet Ghost, for example. Note that the user can only select the proxy of the ghosted target and not the target itself.

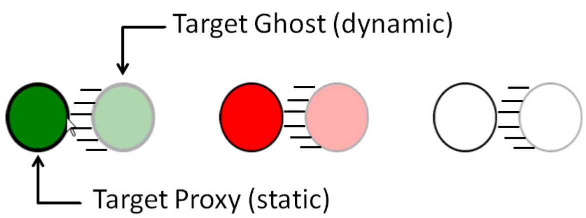


Figure 3. (Arrows and annotations are only for illustration) Target Ghost technique with the basic cursor. When Ghosted, the original target is faded (as a ghost) but keeps moving along its trajectory. A much sharper proxy of the object remains at the target's position when the Shift key was depressed and becomes enabled for selection. Note that the user can only click on the proxy to select the target.

EXPERIMENT 1 – 1D TARGET SELECTION

Goal

In our first experiment we compared the performance of different cursor techniques in a 1D selection task. Each cursor technique was also accompanied with its Ghost equivalent. In this experiment the targets were always moving along the horizontal axis and cursor movement was restricted along the same axis. Based on the properties of our techniques, we hypothesized the following:

H1: the dynamically enlarged activation area or Comet will result in greater reduction in selection time compared to the Basic cursor and Area cursor;

H2: the constantly changing size of the Bubble cursor may be visually distracting to the user and would thus negatively affect targeting performance;

H3: both the Comet and the Bubble cursor techniques will result in fewer click errors than the Basic and Area cursors as the former two techniques result in selecting targets with a larger activation area;

H4: since Ghosting results in selecting static targets, these will show faster selection times and fewer errors than their un-ghosted counterparts.

Methods

Apparatus

The experiment ran on a Windows XP PC equipped with a 22 inch LCD monitor with a resolution of 1680 * 1050 pixels (1 pixel = 0.28 mm in real world units) and a Microsoft mouse that used the default Windows XP settings. The experimental system is a standard Windows gaming application, developed using the Microsoft XNA framework.

Subjects

Twelve participants (9 male and 3 female) who ranged in age from 21-35, participated in this experiment. All of them were right-handed. They were all frequent mouse users and occasional computer gamers. Participants were paid a small sum of money for volunteering.

Task

The experimental system required that the user select a moving goal target in a 1D environment by moving their cursor from the Start position to the target. All targets and distracters were solid circles. The target was drawn in red and placed between two white distracters. The distracter targets were placed on opposite sides and were equidistant from the target center, controlling effective width for the bubble cursor and comet techniques. The task was successful when the user selected the goal target, which was highlighted in green when selected. An error was recorded if the user missed the target. Users could keep attempting to select the object until it disappeared off the screen.

Design

A within-subject design was used to compare the performance of each technique. The independent variables we selected were *Technique*, *Target Speed*, *Target Width* or *Width*, and *Distracter Distance*.

The eight *Techniques* were:

- *Basic cursor:* this is the basic Windows pointer and served as a baseline;
- *Area cursor:* this was implemented as a circular cursor with a width of 100 pixels, which was the same as the maximum target width;
- *Bubble cursor:* this was implemented as originally designed [11];
- *Comet*: this technique was implemented as described above. Target speed and target width were the determining factors for assigning a size to the Comet's tail;
- *Ghosts*: each of the above techniques also included a version of its ghost, resulting in Basic Ghost, Bubble Ghost, Area Ghost and Comet Ghost. The user would trigger the Ghost by pressing the Shift key using their non-dominant hand.

We chose *Speeds* of: 500 pxs/sec, 650 pxs/sec and 800 pxs/sec, which are values that could occur in video streams, video games, and simulations of natural phenomenon. Targets were set to *Widths*: 50 pixels, 75 pixels and 100 pixels. *Distracter distances* were set at 250 pixels, 300 pixels and 350 pixels. We settled on levels of these factors through iterative pilot testing, to provide a reasonable spectrum of difficulty levels. Participants were shown the various techniques and the experimenter demonstrated the task. They were given at least 2-3 practice trials and more if needed. When ready, subjects performed four test trials with each condition, yielding 8×3×3×3×4=864 trials per subject, or 10,368 trials in total. All of them completed the experiment in one session, lasting approximately 45 minutes. The trials were grouped by technique, and the techniques order of presentation was randomized among participants.

Experimental Setup

The goal target always began at a constant distance (300 pixels) from the start button and had a pre-determined movement direction. The target to be selected was colored red, and the distracters were rendered in white. When participants correctly selected the target, it turned green. The target was always moving away from the pointer at the start of the trial. If the target crossed the application window, it was marked as a failed trial. At the start of each technique, the program displayed an instruction on the screen.

Measures

In this experiment, we collected the trial completion time, pointer movement distance and error rates for our data analysis. *Trial completion time* was the time from when the user clicked the start button to when they successfully selected the target. *Errors* were logged if users clicked the mouse button but failed to select the target. Upon completion, participants ranked (Likert scale: 1-5) the techniques according to their preference.

Results

We used the univariate ANOVA test and Tamhane post-hoc pair-wise tests (unequal variances) for all our analyses.

Task Completion Time

Outliers defined by 3 s.d. away from the mean for selection technique were removed, resulting in less than 2% of all trials being excluded from the analysis. Results showed a main effect of *Technique*, *Target Speed*, *Target Width*, and *Distracter Distance* (all $p<0.001$) on trial completion time with $F_{1,7}=656.2$, $F_{1,2}=15.3$, $F_{1,2}=76.4$, and $F_{1,2}=7.1$, respectively. There were significant interaction effects (all $p<0.005$) for *Technique×Target Speed* ($F_{1,14}=3.7$), *Technique×Target Width* ($F_{1,14}=20.9$), and *Technique×Distracter Distance* ($F_{1,14}=2.4$). Other interaction effects were not significant.

Post-hoc pair-wise comparisons of *Techniques* yielded significant differences across all pairs of techniques ($p<0.01$), with the exception of the Area cursor vs. Bubble Ghost ($p=1.0$). Participants were fastest with the Comet (381ms), then the Bubble cursor (415ms), followed by the Comet Ghost (435ms), the Area cursor and Bubble Ghost (470ms), then the Basic Ghost (689ms) and finally the Basic cursor (785ms) (Figure 4).

Post-hoc pair-wise comparisons of *Target Speed* yielded significant differences in trial completion times between 500 and 800 pxs/sec ($p=0.003$) and between 650 and 800 pxs/sec ($p=0.001$). Post-hoc pair-wise comparison of *Target Width* yielded significant differences (all $p<0.01$) in trial completion times for all pairs of widths. Post-hoc pair-wise comparison of *Distracter Distance* yielded significant differences only between the largest two distances, 250 and 350 pixels ($p=0.004$) in trial completion times.

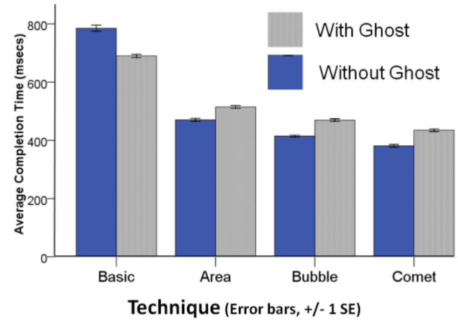


Figure 4. Task completion times across techniques with and without the Ghost.

Number of Errors

Results showed a main effect of *Technique*, *Target Speed*, *Target Width*, and *Distracter Distance* (all $p<0.002$) on trial completion time with $F_{1,7}=349.1$, $F_{1,2}=6.3$, $F_{1,2}=28.2$, and $F_{1,2}=9.6$ respectively. There were significant interaction effects (all $p<0.05$) for *Technique×Target Speed* ($F_{1,14}=2.4$), *Technique×Target Width* ($F_{1,14}=19.1$), and *Technique×Distracter Distance* ($F_{1,14}=1.9$). Other interaction effects were not significant.

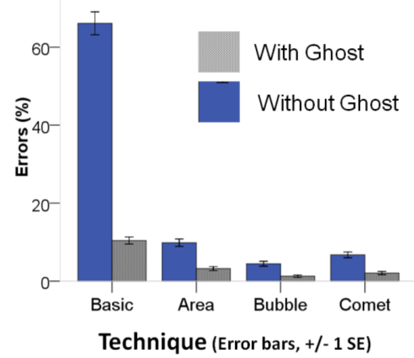


Figure 5. Error rates across techniques with and without the Ghost.

Post-hoc pair-wise comparisons of *Techniques* yielded significant differences between the Basic cursor and all other techniques ($p<0.01$). The Ghosted techniques were also all significantly less error prone than the non-ghosted techniques ($p<0.05$), except for the Bubble cursor which was on par with the Area Ghost ($p=0.983$). With the exception of the Basic Ghost (10% error rate), the other ghosted techniques exhibited error rates less than 3% (Area Ghost=3%, Comet Ghost=2%, Bubble Ghost=1%). In the non-ghosted techniques, the Basic cursor had the highest error rate (66%), followed by the Area cursor (10%), the Comet (7%) and the Bubble (4%) (Figure 5). It is not surprising that users have to click less in the ghosted version of the techniques, as the proxies of the moving targets are stationary. However, interestingly the Basic Ghost (i.e. unassisted by other enhancements) performs either at the same level or worse than the basic cursor enhancement techniques.

Post-hoc pair-wise comparisons of *Target Speed* yielded no significant differences in the number of errors between pairs of speeds. Post-hoc pair-wise comparison of *Target Width* yielded significant differences in the number of errors between 50 and 100 pixel target sizes ($p<0.001$). Similarly, a post-hoc pair-wise comparison of *Distracter Distance* only yielded significant differences between distances of 250 and 350 pixels ($p<0.001$) in number of errors.

Subjective feedback

In an exit survey, participants ranked (the Likert scale consisted of equally spaced scalar values from 1 – least preferred - to 5 – most preferred), the Bubble and Comet Ghost, 3.66 and 4.16 respectively. This order is reflected in the error rates. These were then followed by the Bubble, Comet and Area cursor techniques. A similar pattern prevailed concerning the level of control for each technique.

Discussion

Technique

The results of our study show that participants were fastest with an enhancement to the target, such as with the Comet over cursor based enhancements (supporting H1). The Comet tail increases the effective width of the target thus facilitating the selection. Based on our exit survey, participants did not seem affected by the rapidly changing cursor size with the Bubble, thus rejecting H2.

The Basic cursor performed the slowest and was also the most error prone. Interestingly, the interaction effect of technique and speed is most apparent with the basic cursor as users were faster at higher speeds than at lower ones. This effect has been previously reported in the literature [14] but primarily when users have intercepted the target (i.e. waiting ahead on the target's path so that the target falls under the cursor). Indeed, in observing users select targets with this cursor type, we found that once an error occurred (i.e. they missed the target), they would convert their motion into one involving interception instead of pursuit. This occurred less with the other cursor types.

The only Ghost technique that showed any improvement in time performance in comparison to its counterpart was the Basic Ghost. All other Ghost techniques were slower, thus rejecting H4. This was in part due to pressing the Shift key to create the proxies of the moving objects, causing an overhead of around 200 ms. However, we found that the Ghost techniques exhibited the lowest error rates. It is also interesting to observe the wide range in error rates from the Basic technique compared to all the other cursor types, supporting our hypothesis that selecting moving targets (at the levels tested) requires significant assistance. Our results also support H3, suggesting that the increase in activation area resulting from the Comet and Bubble would minimize errors in comparison to the Basic and Area cursors.

Distance to target

We looked at the cursor-to-target distance when selection occurred. If this distance was large, it means that users did not move their cursor too far to select. This gives us insight as to whether users were taking advantage of the enhanced activation areas of the Comet and Bubble cursors. Interestingly, we see that the distance-to-target is not only a function of the distracter distance, but also of the speed of the targets. As we see in Figure 6.a, this distance is highly affected by the target speed. In contrast, we see a lesser effect of distracter distance on distance-to-target (Figure 6.b). Interestingly, the largest variance with respect to distracter distance was with the Comet technique and not the Bubble cursor as we would have expected. This happens because when the Bubble cursor is highlighting the correct target, users needed to readjust their motor movement before they complete the selection as fast moving distracters could cancel out the Bubble cursor's selection. For this reason, enhancements to the target may have a slighter advantage over techniques with cursor enhancements for this task.

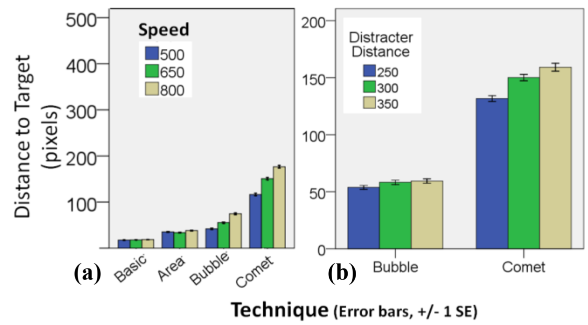


Figure 6. Distance-to-target (a) across techniques without the Ghost; (b) by technique and distracter distance.

Target Speed

As expected, faster targets are harder and more error prone to select. Static proxies of the targets, as with Ghosting reduce the error rates slightly. However this is not significantly different than leaving the targets in full movement, suggesting that bringing targets to a full halt may not necessarily resolve erroneous selection. Other alternatives, discussed later are needed to reduce such errors.

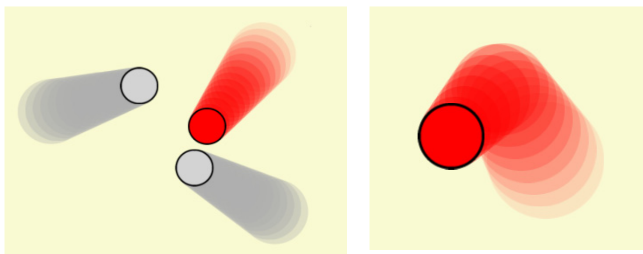
Overshooting the target

We found that with the Area and Bubble cursors participants would select a target when the pointer was in front of it. We found that this occurred more often with the Area cursor (11% of all trials with this cursor) than the Bubble cursor (7%). These trends are also very similar for the Area and Bubble Ghost techniques. Surprisingly, this occurred more frequently at the lower target speeds.

COMET AND TARGET GHOST FOR 2D SELECTION

In the 1D experiment subjects were able to quickly access targets using the activation area introduced by the comet tail. However, pilot tests for the 2D setup revealed that the 1D Comet design was weak on two aspects. The first consisted of the shape of the tail. In the 1D Comet, tail thickness degrades gradually providing an elliptical activation area for the cursor to land on. However, results from the first study show that participants largely selected targets when the cursor was at the tip of the tail. This part of the tail needed a significantly larger activation area and we therefore increased the thickness of the tail at its tip by a factor of target width×1.5 (Figure 7).

In this design the shape of the tail follows the changes in the object's movement direction (Figure 7.b). As a result, users do not need to make drastic changes to the cursor's path when the target is moving along less predictable trajectories. Clicking the mouse button when the cursor is over the comet tail or on the target selects the object. If two comet tails overlap the closest target gets selected.

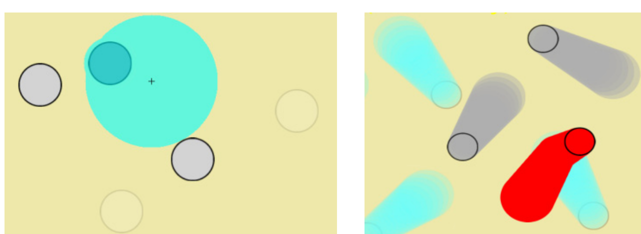


(a) (b)

Figure 7. (a) The tail's thickness grows towards its end. (b) Tails bend and turn based on the target's movement path. This helps users in not having to alter their initial ballistic movement that was aimed at the target before it moved.

The tail gradually increases in transparency until it completely blends with the background. We rendered the tail using a series of slightly wider circles positioned along the target's movement path. The part of the tail closest to the target is solid and then gradually becomes more transparent toward its end. The purpose of this gradual transformation was to reduce the amount of clutter that would appear on the screen particularly in more densely populated scenes.

Target Ghost did not vary significantly from the 1D setup (Figure 8.a). In Comet Ghost, a proxy of both the target and the tail were created. To reduce clutter, the Ghosted object was drawn in a transparent blue shade. The target proxies are red and those of the distracters are grey.



(a) (b)

Figure 8. (a) Bubble Ghost, and (b) Comet Ghost. With Comet Ghost targets were transparent to reduce clutter.

EXPERIMENT 2 – 2D TARGET SELECTION

Goal

Results of experiment 1 showed the potential of the Comet as a selection technique for 1D moving targets, due to its increased activation area. However, it could be possible that the Comet may not be suitable for more complex target acquisition tasks that one commonly encounters in real 2D GUI interfaces. Using the Bubble cursor, a user can select the target with the closest distance to the pointer. However, in Comet, the selection area is only enlarged along the movement vector. Therefore if the cursor movement is orthogonal to the tail or the Comet, then selection could be impaired as the activation region may be less beneficial [18]. To investigate the performance of the various cursor types and their ghosted versions, we conducted a second experiment with targets moving in 2D. Since the movement vector has shown to effect acquisition times [21], we also examined the effect of a less predictable target paths on selection performance. We excluded the Area cursor from this experiment as it did not show significant benefits in the 1D task. We hypothesized that:

H5: the ghosted techniques will outperform the others in selection times and errors as users will select a static target;

H6: the enlarged activation area brought by the Comet will mainly show benefits in the absence of ghosting.

Methods

Apparatus

The apparatus used was the same as in experiment 1.

Subjects

12 right-handed participants (8 male) whose age ranged from 21 to 30, volunteered for this experiment. All of them were right-handed. They are frequent mouse users and occasionally play computer games.

Task

We used 20 distracters along with the target. Users were asked to select the goal target, which was moving based on its movement predictability and speed. Targets were also allowed to bounce off the edges of the application window.

Design

We used a within-subject design to compare the performance of the techniques in the 2D environment. Here, the independent variables were *Technique*, *Path Predictability*, *Target Speed*, and *Density*.

The *Techniques* selected were, the Basic cursor, Bubble cursor, Comet, and their ghosted equivalent.

Path predictabilities were defined as:

- High - the target moved in a straight line.
- Medium - the target changed direction between 15º and 45 º, at intervals of 400-600 msecs.
- Low - the target changed direction between 45º and 315 º, at intervals of 200-400 msecs.

Target speed was set to 400 pixels/sec, 550 pixels/sec and 700 pixels/sec.

The width of the targets and distracters was fixed at 75 pixels. The design was balanced based on cursor type (6 types), with 3 blocks of trials for each technique. Within each block we used 3 path predictabilities × 3 target speeds × 2 distracter distances × 3 repetitions for each condition, yielding 972 trials per subject or 11,664 trials in total. All the participants completed the experiment in one session, lasting approximately 50 minutes. Before starting the trials, subjects were given practice trials with each technique. The techniques order of presentation was counterbalanced among participants using a Latin Square design.

Experimental Setup

At the start of each trial, the target and the distracters were initialized with a random movement vector. When targets hit the edge of the application window, we redirected the path at an angle of reflection equal to that of the incidence. After each successful selection, a new target and a set of distracter targets were displayed, with the goal target appearing at a fixed distance (400 pixels) from the start button but at different *x-y* positions. To test the effect of movement type, we used three different path predictabilities (see above). In the case of high path predictability, the targets started with a random vector, and only the direction vector changed when it bounced off the edges. Targets with medium path predictability also started with a random initial vector, but the vector changed after a longer random time interval and with a smaller random angle. Low path predictability had the same initial conditions, but the vector changed at a high variance and at a higher angular degree. The target to be selected was colored red, and the distracters were rendered as light gray with the same width as the goal target. When selected correctly the target turned green.

Measures

In this experiment, we also collected the trial completion time and error rates. We asked participants to fill out a post-experiment survey to rank the techniques according to preference and perceived level of control.

Results

We used the univariate ANOVA test and Tamhane post-hoc pair-wise tests (unequal variances) for all our analyses.

Task Completion Time

Outliers defined by 3 s.d. away from the mean for selection technique were removed, resulting in less than 5% of all trials being excluded from the analysis. Results showed a main effect of *Technique*, *Target Speed*, and *Path Predictability* (all $p<0.001$) on trial completion time with $F_{1,5}=1660.09$, $F_{1,2}=146.03$, $F_{1,2}=108.35$, and $F_{1,1}=6.04$ respectively. There were significant interaction effects (all $p<0.005$) for *Technique×Target Speed* ($F_{1,10}=30.74$), *Technique×Path Predictability* ($F_{1,10}=59.74$). Other interaction effects were not significant.

Post-hoc pair-wise comparisons of *Techniques* yielded significant differences across all pairs of techniques ($p<0.01$), with the exception of the Bubble cursor vs. Basic Ghost ($p=1.0$). Participants completed the target selection task in less time using all the ghosted techniques than their non-ghosted version. Participants were fastest with the Bubble Ghost (609ms), then the Comet Ghost (709ms), followed by the Comet (753ms), the Bubble cursor (794ms), then the Basic Ghost (810ms) and finally the Basic cursor (1,841ms) (Figure 9.a).

Post-hoc pair-wise comparisons of Target Speed yielded significant differences in trial completion times between all pairs of speeds. Post-hoc pair-wise comparisons of *Path predictability* yielded significant differences in trial completion times between *High* and *Low* and no significant difference between other pairs. Figure 9.b shows the effect of path predictability on performance.

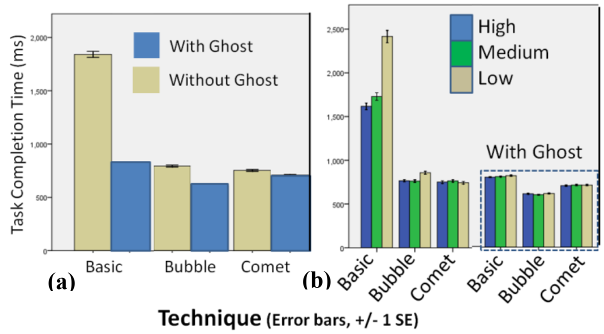


Figure 9. Task completion times (a) across techniques with and without the Ghost; (b) by path predictability.

Number of Errors

Results showed a main effect of *Technique*, *Path Predictability,* and *Target Speed* (all $p<0.001$) on errors with $F_{1,5}=1030.2$, $F_{1,2}=10.36$, $F_{1,2}=109.05$, and $F_{1,1}=14.5$ respectively. There were significant interaction effects (all $p<0.05$) for *Technique×Target Speed* ($F_{1,10}=15.08$), and *Technique×Path Predictability* ($F_{1,10}=6.49$). Other interaction effects were not significant.

Post-hoc pair-wise comparisons of *Techniques* yielded significant differences between the Basic cursor and all other techniques ($p<0.01$). The Ghosted techniques were also all significantly less error prone than the non-ghosted techniques ($p<0.01$). The ghosted techniques exhibited error rates less than 10% (Comet Ghost=5%, Bubble Ghost=2% and Basic Ghost = 9%). In the non-ghosted techniques, the Basic cursor had the highest error rate (74%), followed by the Bubble cursor (38%) and then Comet (33%) (Figure 10). This sharp difference between the ghosted techniques and their counterpart is not surprising particularly since low path predictabilities result in many mis-clicks.

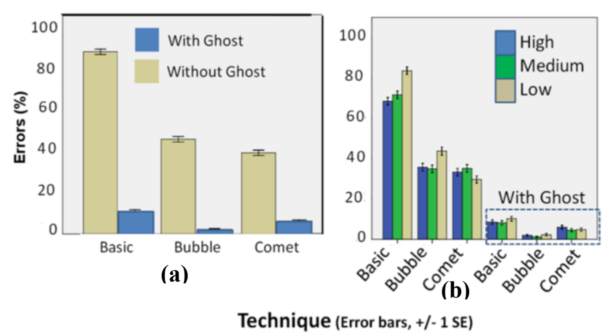


Figure 10. Error rates (a) across techniques with and without Ghost; (b) by technique and path predictability.

Post-hoc pair-wise comparisons of *Target Speed* yielded significant differences ($p<0.001$) in the number of errors between pairs of speeds. Post-hoc pair-wise comparison of

Path Predictability yielded no significant differences in number of errors between pairs of path predictabilities.

Subjective feedback

On a Likert scale from 1 (lowest preference) to 5 (highly preferred), the Bubble and Comet Ghosts were the most preferred techniques (4.16 for both) followed by the Comet and then the Bubble. Understandably, Basic cursor was the least preferred among all techniques.

DISCUSSION

We first discuss our results from experiment 2.

Discussion of results from Experiment 2

Our results show that additional cursor enhancements such as target expansion or static proxies are necessary for selecting moving targets. With different path movement types, ghosted techniques had faster selections and fewer errors than without ghosting (supporting H5). However, without the ghosting, we see that the Comet's activation area is again beneficial to users, as they are faster and less error prone than the Bubble cursor (partial support for H6). Our results strongly support the use of cursor or target enhancements even with static proxies. This is not surprising in terms of performance times, but in terms of errors we see that significantly fewer errors occur with the Bubble Ghost and Comet Ghost than the Basic Ghost. This result can be partly explained by users over/undershooting targets whose proxies stop abruptly. One solution might be to bring the static proxies into a gradual halt.

We were surprised to see that in the second experiment, even with ghosting, i.e. creating static proxies, we were not able to completely remove all errors. Users still needed some form of enhancement, such as the Comet or Bubble cursor to complement proxy selection. In the case of the enhanced cursor types, a large part of the errors resulted from clicking the wrong target, suggesting that finer improvements are needed to make selection practical. One solution might be to list all objects 'under' the cursor such that the user can then select one from the list. Our technique would then work as a pre-filter to provide the user with a list of potential items. Another possibility might be to disambiguate selection by correlating the movement direction with the cursor's movement path. Movements that are deemed parallel may indicate a strong likelihood of a user wanting to select these targets.

Applications

Many applications can benefit from the results obtained here. We implemented our techniques in a video browsing environment, to emulate what a user of a video tracking system may have. Direct manipulation interfaces in such an environment have shown to significantly improve user interactions [7,8]. To do this, we captured video footage of cars on a highway. We then manually annotated the motion paths by extracting them from all the frames. This allowed us to separate moving objects from the background, to which we then applied our techniques (Figure 1, and video). Other applications that can benefit from these techniques include molecular or weather simulations and educational tools that are based on animated displays. Typically, animations such as these are only viewable and accept very little input from the user. However, direct manipulation can aid in better understanding the information being displayed. For example, as video tracking becomes more commonplace one can foresee target selection available for multiple media types, including televised games. In these scenarios, it might be particularly difficult to pause the entire scene. With the use of our techniques, users can select players or other items of interest to briefly inspect relevant statistics. Finally, video games include a large number of moving objects. Typically, designers select the right level of difficulty to provide a sufficient balance. This avoids losing game players who either get bored because it is easy, or who give up because of the difficulty level. Our results can assist designers in selecting the appropriate balance by introducing techniques at the right level of user expertise.

Pausing vs. Ghosting

Ghosting is inspired by proxy-based techniques [2, 4] but also borrows properties of pausing an animated scene. While pausing is also possible, there are cases where Target Ghosts would be more useful. For example, in air-traffic control, pausing the display would not be a viable option since releasing the pause would create a disjointed view between frames. In contrast, the continuity that is available with Ghosting would still allow the user to inspect the ghosted targets whose movements are not interrupted. Future work is required to find applications where either type of interaction is the most appropriate.

Guidelines for Designers

From our results we propose the following guidelines:

- designers should take the cost of an error into account when selecting a design;
- for target movements in 1D, Comet should be the technique of choice;
- in 2D, selection of moving targets is best achieved with static proxies, such as Target Ghost;
- cursor enhancements such as the Comet or Bubble cursor should be employed, even when using static proxies of moving objects.

Limitations

Comet and Target Ghost also present some limitations. For example, in scenes with a large number of objects, Comet can add clutter. This makes it more difficult to select those targets that may lie beneath the clutter. One solution might be to create more intelligent techniques that only add comet tails to those objects that are needed or that fall in line with the cursor's movement. Similarly, the duplicates of objects with Target Ghost also create clutter that can be difficult to manage in a scene. One approach to resolve the clutter might be to create proxies of only a few targets that are in the vicinity of the cursor. Furthermore, performance of the comet may degrade when two tails overlap. It would be interesting to investigate this further using advanced heuristic methods for overlapping tails. Finally, if targets all have similar appearance, it should be investigated how well users can keep track of their goal target after activating the Target Ghost technique.

Another concern worth mentioning is the small number of participants used for higher number of conditions in the experiments. While the parameters used in our studies are consistent with similar HCI studies, we feel that a larger number of participants could further ascertain our claims.

CONCLUSION AND FUTUREWORK

In this work we present techniques that aid in selecting moving targets. We demonstrate through two experiments the need for assisting the basic cursor in selecting such types of targets. Comet enhances objects by adding a selectable tail. This increases the activation area for selection. Target Ghost creates proxies of the moving object. These proxies are static and created at the moment of invocation. In a 1D selection task, Comet outperformed existing techniques such as the Area and Bubble cursors. Comet also outperformed Ghosted versions of all techniques. In a 2D selection task, Comet outperformed the Bubble, but only in conditions without the ghost. Our results show that even if the user were to create static versions of the moving objects, an enhancement such as the Comet or Bubble is critical to assist in selecting these.

In future work we will test the performance of our techniques in real applications. This could lead to new improvements to the Comet and Target Ghost, such as reducing the amount of clutter and minimizing the overhead of creating static proxies. We will also test these techniques with different input devices, for example, by simulating a TV remote control. Finally, we are also interested in studying whether the selection of targets moving in 3D has different features than the ones we describe here.

REFERENCES

1. Balakrishnan, R. (2004). Beating Fitts' law: virtual enhancements for pointing facilitation. Int. J. Hum.-Comput. Stud. 61, 6, 857-874.
2. Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P. Bederson, B., and Zierlinger, A. Drag-and-Pop and Drag-and-Pick (2003). Techniques for Accessing Remote Screen Content on Touch- and Pen-operated Systems. *Proc. of Interact'03*, pp. 57-64.
3. Baudisch, P., Zotov, A., Cutrell, E., and Hinckley, K. (2008). Starburst: a target expansion algorithm for non-uniform target distributions. *Proc. of AVI '08*, 129-137.
4. Bezerianos, A., Balakrishnan, R. (2005). The Vacuum: Facilitating the manipulation of distant objects. *Proc. of CHI'05*, 361-370.
5. Bezerianos, A., Dragicevic, P., and Balakrishnan, R. 2006. Mnemonic rendering: an image-based approach for exposing hidden changes in dynamic displays. *Proc. of UIST '06*, 159-168.
6. Chapuis, O., Labrune, J., and Pietriga, E. (2009). DynaSpot: speed-dependent area cursor. *Proc. of CHI '09*, 1391-1400.
7. Cheng, K., Luo, S., Chen, B., and Chu, H. (2009). SmartPlayer: user-centric video fast-forwarding. *Proc. of CHI '09*, 789-798.
8. Dragicevic, P., Ramos, G., Bibliowitcz, J., Nowrouzezahrai, D., Balakrishnan, R., and Singh, K. (2008). Video browsing by direct manipulation. *Proc. of CHI '08*, 237-246.
9. Faure, G., Chapuis, O., and Beaudouin-Lafon, M. (2009). Acquisition of Animated and Pop-Up Targets. *Proc. of INTERACT '09*, 372-385.
10. Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47, 381-39.
11. Grossman, T. and Balakrishnan, R. (2005). The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. *Proc. of CHI '05*, 281-290.
12. Grossman, T., Kong, N., Balakrishnan, R. (2007): Modeling pointing at targets of arbitrary shapes. *Proc. of CHI'07*, 463-472.
13. Hoffmann, E. R., (1991) Capture of Moving Targets: a Modification of Fitts' Law. *Ergonomics*, 34, 211–220.
14. Ilich, M.V. (2010) Moving Target Selection in Interactive Video, *M.Sc. Thesis, University of British Columbia*, pp. 110.
15. Jagacinski, R. J., Repperger, D. W., Ward, S. L., and Moran, M. S. (1980). A test of Fitts' law with moving targets. *Human Factors*, 22(2):225-233.
16. Kabbash, P. and Buxton, W. (1995). The "Prince" technique: Fitts' law and selection using area cursors. *Proc. of CHI'95*, p. 273-279.
17. MacKenzie, I. S. and Buxton, W. (1992). Extending Fitts' law to two-dimensional tasks. *Proc. of CHI '92*, 219-226.
18. McGuffin, M. and Balakrishnan, R. (2002). Acquisition of expanding targets. *Proc. of CHI '02*, 57-64.
19. McGuffin, M. J. and Balakrishnan, R. (2005). Fitts' law and expanding targets: Experimental studies and designs for user interfaces. *TOCHI*, 12, 4, 388-422.
20. Mould, D. and Gutwin, C. (2004). The effects of feedback on targeting with multiple moving targets. *Proc. of Graphics interface 2004*, 25-32.
21. Tresilian, J. R. & Lonergan, A. (2002) Intercepting moving objects: effect of temporal precision and movement amplitude. *Experimental Brain Research*, 142, 193-207.