Acquisition of Animated and Pop-up Targets

Guillaume Faure, Olivier Chapuis, Michel Beaudouin-Lafon

In INTERACT '09: Proceedings of the 12th IFIP TC13 Conference in Human-Computer Interaction - LNCS 5727. Springer & IFIP, 372–385, 2009.

Acquisition of Animated and Pop-up Targets

Guillaume Faure[1,2], Olivier Chapuis[1,2], and Michel Beaudouin-Lafon[1,2]

[1] LRI – Univ. Paris-Sud & CNRS, Orsay, France
[2] INRIA, Orsay, France

Abstract. Pop-up targets, such as the items of popup menus, and animated targets, such as the moving windows in Mac OS X Exposé, are common in current desktop environments. This paper describes an initial study of pointing on pop-up and animated targets. Since we are interested in expert performance, we study the situation where the user has previous knowledge of the (final) position of the target. We investigate the effect of the DELAY factor, i.e. the delay before the target pops up (for pop-up targets) or the duration of the animation (for animated targets). We find little difference between the two techniques in terms of pointing performance (time and error), however a kinematic analysis reveals differences in the nature of the pointing movement. We also find that movement time increases with DELAY, but the degradation is smaller when the target is farther away than when it is closer. Indeed, larger distances require a longer movement time therefore the target reaches its destination while the participant is still moving the pointer, providing more opportunity to correct the movement than with short distances. Finally we take into account these results to propose an extension to Fitts' Law that better predicts movement time for these tasks.

Keywords: Pop-up targets, Animated targets, Movement analysis, Fitts' Law.

1 Introduction

Pointing is a fundamental action in graphical user interfaces (GUIs) that has been the subject of much research to both understand pointing actions and improve pointing techniques. Most of this research has focused on static targets that are displayed before pointing starts and stay still during the pointing action. GUIs however have always featured *pop-up targets* that appear after the pointing movement starts, e.g., pop-up menus and dialog boxes. More recently, GUIs have also started to feature *animated targets* that move while the pointing gesture is being performed, e.g., the windows in Mac OS X Exposé. While we know that animation enhances user interaction with the system [13, 24] by providing a continuous feedback that increases the user's sense of direct and indirect interaction [26], its effect on pointing has not, to the best of our knowledge, been studied. In particular, beyond the empirical evidence that users take advantage of their knowledge of the behavior of targets to anticipate their apparition or final position, we are not aware of any systematic study of this phenomenon.

This paper presents what we believe is the first controlled study designed to better understand the acquisition of animated and pop-up targets. Our main factor is DELAY, the delay before the target pops up (for pop-up targets) or the duration of the animation (for animated targets). We have focused on short values of DELAY — between 0 and

In INTERACT '09: Proceedings of the 12th IFIP TC13 Conference in Human-Computer Interaction - LNCS 5727. Springer & IFIP, 372–385, 2009.
Authors Version

500 *ms* — which are typical of GUIs, and we have operationalized the common situation where users can anticipate the final position of the target by ensuring that users have prior knowledge of its location.

The experiment compares pointing performance (in time and error) for static, animated and pop-up targets under various values of DELAY. We found a strong effect of DELAY: movement times increase with the delay, but the degradation is smaller for longer distances to target than for shorter ones. We also found little performance difference between pop-up and animated targets, but an interesting qualitative difference when looking at the kinematic profiles. In particular, this analysis shows evidence of how users anticipate the final position of the target.

Finally we show that Fitts' Law does not accurately predict movement times for pop-up and animated targets. Using the insights gained by the analysis of the experimental results, we propose an extension to Fitts' Law that takes into account the unique characteristics of these tasks to better predict movement time.

The rest of the paper is organized as follows. After some background and related work, the next section provides a set of examples and motivates the present study. The subsequent sections describe the experiment, present the results regarding performance (movement time and errors), analyse the kinematic aspects of typical movements (velocity vs. distance and number of sub-movements) and propose an extension to Fitts' Law. The paper then concludes with a discussion and directions for future work.

2 Background and Related Work

The acquisition of an on-screen target by means of a pointing device (*pointing* for short) is one of the basic tasks in Graphical User Interfaces and one that has been extensively studied in Human-Computer Interaction (HCI). Previous and ongoing research in this area includes novel techniques to facilitate target acquisition, comparison of input devices performance, and models to predict movement time and better understand target acquisition tasks of various types.

The major theoretical tool for studying pointing is Fitts' Law [11], which predicts the movement time to acquire a target of width W at a distance D. The most widely used form of Fitts' Law in HCI [19] is:

$$MT = a + b \log_2 \left(\frac{D}{W} + 1 \right)$$

where a and b are empirically determined constants and where $\log_2 \left(\frac{D}{W} + 1 \right)$ is called the *index of difficulty* (ID) of the task. While a number of variations of this law have been proposed (see [23] for a review), the Psychology literature has not reached a consensus yet on an explanation of this Law.

The most popular explanation of Fitts' Law in the HCI community is Meyer et al.'s sub-movements model [20]: first, a fast sub-movement towards the target (often called the ballistic phase of the movement) is performed, then if this movement does not hit the target, another (probably smaller) sub-movement towards the target is performed, and this process continues until the target is reached. An important characteristic of the sub-movements model that may explain the logarithmic form of Fitts' Law is that the distance vs. velocity graph shows a bell shape with a clear velocity peak.

The originality of the study presented in this paper is that the target is not always visible on the screen during the acquisition movement. Most previous work in HCI has investigated the acquisition of an initially invisible target using navigation tasks, e.g., scrolling and pan-and-zoom, rather than a pure pointing task. A notable exception is Cao et al. [7] where the targets are initially invisible and are revealed by moving a display window attached to the cursor (coupled cursor). The pointing task in that case involves two phases: first, reveal the target, then acquire it. In our study the targets are revealed automatically (no user action) and users know in advance where the targets are so they can anticipate their final position. We expect a more integrated motion than in the above study, with dynamic adaptation to the target behavior.

Some studies have shown that the shortest delay needed for sensory information to affect hand movement is approximately 100 *ms* [17]. For example Flash and Henis [12] have observed movement adaption between 100 and 200 *ms* after an experimentally controlled modification of the cursor trajectory. Closer to HCI, Zhai et al. [28] have shown that users can take advantage of *unpredictable* target expansion, suggesting a dynamic adaptation of the user's pointing movement.

Menus are the main example of pop-up targets and have been widely studied in HCI. With linear menus, it is assumed that the menu pops up immediately and studies have focused on factors such as the number of items and the depth of the hierarchy of menus rather that the pop-up delay and involve in addition of Fitts' Law the Steering Law (see, e.g., [9]). With marking menus [18] and their variants, there is a distinction between novice mode, where the user *waits* for the menu to pop up, and expert mode, where the menu does not appear at all. Here, the absence of motion during the delay is used to activate novice mode (see also [14]).

Regarding animated targets, previous studies in Psychology, e.g., [15], and in HCI [21] have addressed the acquisition of moving targets. These studies however have considered *capturing tasks*, i.e. acquiring the target while it is moving whereas in our case the target can be acquired only when it has reached its destination. Other examples that involve moving targets include facilitation techniques that bring the targets (with a possible animation) close to the cursor after the movement has started, e.g., Drag-and-pop [4] and Vacuum [5]. These techniques however make it difficult for the user to anticipate the final position of the target.

Memorization of target positions is an important point of our experimental design and has been the subject of previous work, e.g., Hornof and Kieras [16]. However, we only consider the memorization of a single target in short term memory. This phenomenon has been studied in the domain of Human Vision (see, e.g., [2]) where it has been shown that humans can recover spatial positions via saccades (rapid eye movements) from a few memorized gaze positions.

Surprisingly, we have not found in the literature any previous study of pointing on memorized "invisible" targets. The reverse paradigm however, pointing with an invisible cursor, has been widely studied (see [6] for instance) to better understand whether arm movements follow a position model or an amplitude model. It is interesting to note that our study seems to support the amplitude control model.

3 Motivation and Examples

In the rest of this article we call $Anim$ the condition where the subject is pointing on an animated target and $Popup$ the condition where the subject is pointing on a pop-up target. When the duration of the animation is null (DELAY = 0), the $Anim$ condition is equivalent to the $Popup$ condition: the target appears immediately when movement starts. This is different however from the acquisition of a persistent target, when the target is visible at its destination before the movement starts. This latter condition is the one usually considered in pointing experiments and will act as a control. We call it the $Static$ condition.

One motivation for the design of the experiment was the following scenario. The user wants to drag-and-drop a hidden icon from the desk[3] to an area of the working window, e.g., to attach a file to an email. The user typically must trigger a command to make the desk visible, grab the icon and then make the original working window visible again to drop the icon at the desired destination. With Mac OS X, the user may use Exposé to make the desk visible: all windows are moved outside the screen with an animation and then animated back to their original position when the user starts the drag operation. This corresponds to the $Anim$ condition because the target of the drop operation is an animated window. With Microsoft Windows and in most modern X Window environments the user may use the "show desktop" feature to immediately remove the windows and make the desk visible, start dragging and then show the windows again. This corresponds to the $Popup$ condition with, ideally, a very short delay. A variant of the above technique technique, *desk pop*, has been recently proposed [10]: the desk is moved to the foreground and rendered with a semi-transparent black background while preserving the opacity of the icons. This technique provides access to the icons while keeping the working windows visible. Thus, the drop area is kept visible during the whole operation. This case corresponds to the $Static$ condition.

Dialog boxes are another examples of animated and pop-up targets with a potential *a priori* knowledge of the position of the targets by the user. Such windows often pop up at the same position and contain only a few buttons (typically "Ok" and "Cancel"). Users often know the action to perform before the dialog box pops up, and we have observed [8] that they tend to anticipate the display of the dialog box by moving their mouse pointer toward the target button before it appears. An example of a delayed pop-up window is the "how to download" dialog box of the Firefox browser, which shows up when clicking a link to non-HTML content. Due to network delays, the box pops up with an unpredictable lag and yet users anticipate its appearance. Finally, an example of an animated target is under Mac OS X, where some dialog boxes "slide" out of their parent window's titlebar. Here too, users anticipate the location of the button they want to click.

Finally, an interesting category of pop-up targets is given by web navigation. Users often revisit the same web pages [22] and end up knowing their layout. They also often follow the same navigation paths [25]. For example, a user loads a page, clicks on the login button of pre-filled form that loads a new page, and finally clicks on a link to navigate to the desired page. The delays involved in displaying the various pages and

[3] the background area of the screen containing icons

the targets they contain depend not only on the speed of the network connection but also on the browser's rendering algorithm and the structure of the page. Understanding the effect of delays on pointing may thus have some implications on browser and web page design and, more generally, GUIs.

4 Experiment

We conducted an experiment to study pop-up and animated targets by reducing the problem to a one-dimensional Fitts-like pointing task. Fitts' Law is inherently a 1D model and considering 2D pointing would have involved additional factors [1] that we did not want to include in this first study.

4.1 Apparatus

The experiment was conducted on a 2.33 Ghz Core2-Duo Macbook Pro with an ATI Radeon X1600 graphics card connected to a 24 inch LCD monitor at a resolution of 1920x1200 pixels. We used a 1000 dpi Logitech RX 250 optical mouse with Apple's Mac OS X default acceleration. The Graphics where rendered using the Java *SwingState* library [3]. The animations were linear translations computed at a rate of 60 steps by second.

4.2 Participants

12 unpaid adult volunteers (10 male and 2 female) participated in the experiment. They were all right-handed and aged from 23 to 35 years old (average 27.6, median 26). All the participants were experienced computer users and familiar with mouse pointing.

4.3 Task and Procedure

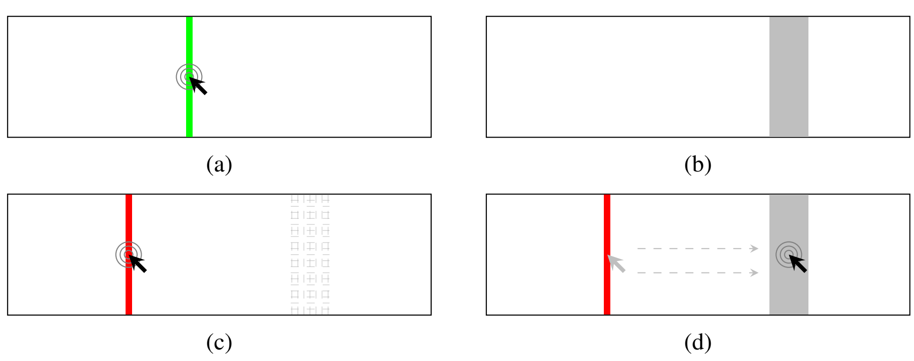


Fig. 1. (a) Beginning of the trial: click green target; (b) memorization phase: show main target for one second; (c) target on screen ($Static$) or hidden (other conditions); click red target to start pointing; (d) end of trial: click main target (after animation or pop up depending on the condition).

A trial is a 1-D target acquisition task decomposed as follows. First, a green target is displayed on the screen (Figure 1(a)) and the participant clicks on it when ready. The green target then disappeared and the main target (in grey) appears for one second (Figure 1(b)). This is the *memorization* phase where the participant is informed of the final position of the main target. After the main target disappears, a red target is displayed (Figure 1(c)). In the $Static$ condition the main target stays on screen while in the other two conditions the main target disappears. In all three conditions, the participant then has to click on the red target, which starts the recording of the movement time. In the $Static$ condition, the main target remains static for the rest of the trial. In the $Anim$ condition, the main target is smoothly animated to its final position in DELAY *ms*. In the $Popup$ condition, the main target appears after DELAY *ms*. The trial ends when the participant successfully acquires the main target (Figure 1(d)) after is has appeared at its final position.

The participants' movement time is recorded from the time the mouse button is released on the red target until the successful button press on the main target. If the participant clicks outside the target or before it reaches its final position, the trial is considered an error but continues until a successful click. Participants are instructed to perform the task as fast and as accurately as possible. To prevent participants from using the cursor to memorize the position of the target the mouse pointer is hidden throughout the memorization phase. Finally, in the $Anim$ condition, the main target moves from the closest border of the screen to its final position in the opposite direction to the acquisition movement.

4.4 Design

The experiment is a partial $2 \times 5 \times 3 \times 3$ within-participant design with the following factors:

- 2 "techniques"conditions (TECH): $Anim$ and $Popup$;
- 5 "delays" conditions (DELAY): $Static$ and 0, 200, 350 and 500 milliseconds (*ms*) representing the duration of the animation or the delay for the pop-up;
- 3 widths (W) for the main target: 16, 32 and 64 pixels;
- 3 acquisition distances (D) to the main target: 256, 512 and 768 pixels.

The design is only partial (not fully factorial) because crossing TECH and DELAY leads to only $2 + 2 \times 3 = 8$ conditions since the DELAY conditions $Static$ and 0 are the same for both TECH conditions: an animation of 0 *ms* makes the target pop up immediately. $Static$ is not really a "delay" condition but it is a convenient way to consider it as both a control and an extreme condition (note that this transforms DELAY into a nominal factor, but we will later transform it into a continuous one).

We use 350 *ms* as the median animation time because this duration is commonly used in graphical user interface animations, e.g., Mac OS X Exposé. We ran pilot studies in order to determine two other times that cover the range of reasonable animation and pop-up times, resulting in 200 *ms* for the lower bound and 500 *ms* for the upper bound.

We divide a run of the experiment into two parts. The first is made of two blocks, one with DELAY = $Static$ and the other with DELAY = 0. The second part of the experiment corresponds to the crossing of TECH (*Anim* and *Popup*) with the DELAY conditions > 0, i.e., $2 \times 3 = 6$ blocks. We block by the TECH condition and use a 3×3 latin square to cross the 2 possible orders of TECH with the 3 non-zero DELAY conditions. This gives 6 counter-balanced orders that we cross with the first two blocks. We divide the 12 participants into two groups of 6. Both groups use these 6 orders, but the first group starts with DELAY = $Static$ while the second group starts with DELAY = 0.

Each of the 8 blocks described above has seven replications of the $3 \times 3 = 9$ combination of D by W, presented in a random order. The first replication is considered a warm up. A pause is offered to participants at the beginning of each replication.

To summarize, the total number of logged trials in the experiment is 8 blocks $\times$ 9 width-distance combinations $\times$ 6 replications $\times$ 12 participants = 5184 trials. We logged 72 trials for each full condition, 6 for each participant. The experiment lasted from 42 to 54 minutes (average 48, median 47).

5 Results

In this analysis, movement time MT is measured until a *successful* button press in the target (as opposed to the first button press). This has the advantage of accounting for penalties caused by errors.

We duplicate data for $Static$ and DELAY $= 0$ in order to simulate these conditions for both $Popup$ and $Anim$. This allows us to perform a standard full-factorial repeated measures analysis of variance MT $\sim$ TECH $\times$ DELAY $\times$ D $\times$ W $\times$ Random(PARTICIPANT). Outliers[4], which represent 0.35% of the trials, are removed from all analyses[5].

Factors	DF	DFDen	F	p
TECH	1	22	4.66	0.0539
DELAY	4	44	97.74	< 0.0001
D	2	22	68.63	< 0.0001
W	2	22	481.94	< 0.0001
TECH × DELAY	4	44	1.43	0.2396
TECH × D	2	22	0.19	0.8303
TECH × W	2	22	1.05	0.3653
DELAY × D	8	88	8.33	< 0.0001
DELAY × W	8	88	1.90	0.0686
D × W	4	44	2.67	0.0444
TECH × DELAY × D	8	88	0.19	0.9920
TECH × DELAY × W	8	88	0.35	0.9436
TECH × D × W	4	44	0.51	0.7274
DELAY × D × W	16	176	1.11	0.3469
TECH × DELAY × D × W	16	176	1.05	0.4071

Table 1. ANOVA for MT $\sim$ TECH $\times$ DELAY $\times$ D $\times$ W $\times$ Random(PARTICIPANT).

Table 1 shows the results of the repeated analysis of variance for movement time. As expected D and W have a (strong) significant effect on movement time: the harder

[4] defined as a movement time 2 standard deviations a away from the mean movement time (for each PARTICIPANT, TECH, DELAY, D and W).

[5] including the outliers yields results that are very similar to those described here.

the task, the longer it takes to complete (Figure 2). Tech fails to reach significance for movement time and we find no significant interaction with the other factors. The difference in mean between $Anim$ and $Popup$ is only 17 *ms* in favor of $Popup$. This difference (see Figure 3) is less than 2% of the mean movement time[6].

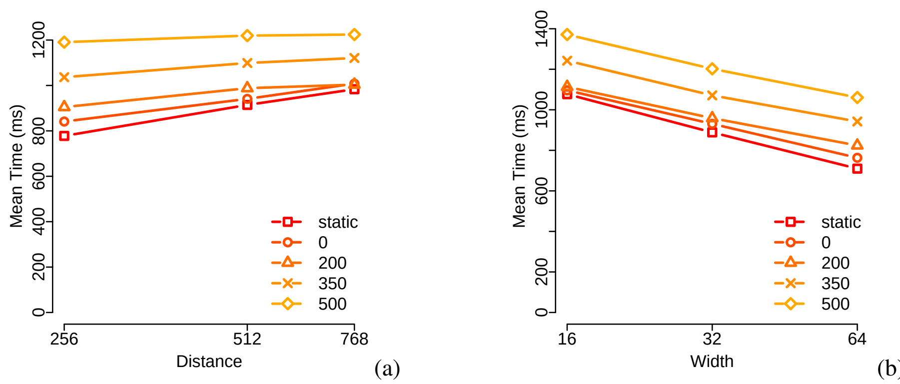


Fig. 2. (a) Movement time as a function of distance for each Delay condition. (b) Movement time as a function of width for each Delay condition.

We observe a (strong) significant effect of Delay on movement time (Figure 3). A Tukey post-hoc test ($\alpha = 0.05$) shows no significant difference in mean between $Static$ and Delay = 0 nor between Delay = 0 and Delay = 200. However, it shows a significant difference in mean between $Static$ and Delay = 200 (74 *ms* in favor of $Static$, a speed-up of 7.7%). It also shows that Delay = 200 is significantly faster than Delay = 350 (120 *ms*, 11.1% speed-up) and that Delay = 350 is significantly faster than Delay = 500 (126 *ms*, 10.4% speed-up).

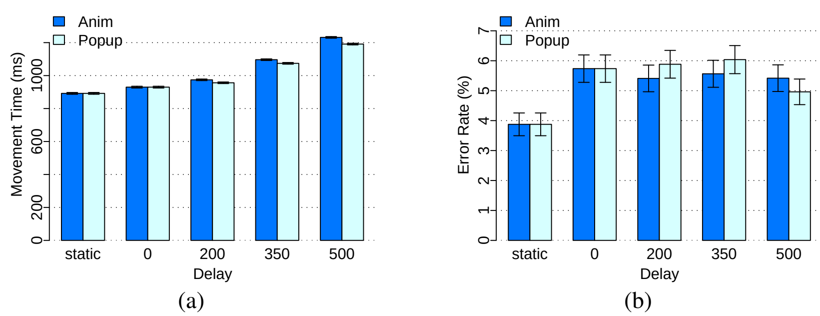


Fig. 3. Movement time (a) and Error rate (b) for each Delay and Tech.

We also observe a significant interaction effect between Delay and D (Figure 2.(a)), suggesting that the effect of distance on movement time is less strong as Delay in-

[6] note that the data duplication for $Static$ and Delay = 0 does not influence these statistical results since they cancel each other out.

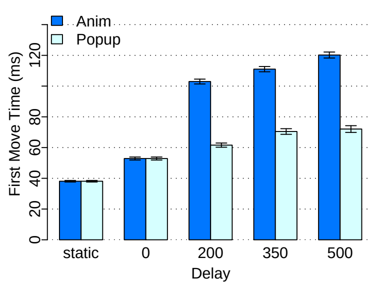
creases from $Static$ to DELAY = 0 to DELAY = 500. Indeed, a post-hoc Tukey test ($\alpha = 0.05$) shows that there is a significant difference (in mean) between each distance for $Static$ and DELAY = 0, a significant difference between distances 256 and 512 but not between distances 512 and 768 for DELAY = 200, a significant difference between distances 256 and 768 but not between distances 256 and 512 and distances 512 and 768 for DELAY = 350, and finally no significant difference at all for DELAY = 500.

Figure 2.(a) also suggests that the difference in movement time between $Static$, DELAY = 0 and DELAY = 200 decreases as distance increases. Indeed, a post-hoc Tukey test shows a significant difference in mean between $Static$ and DELAY = 200 for D = 256 but no such difference for D = 768.

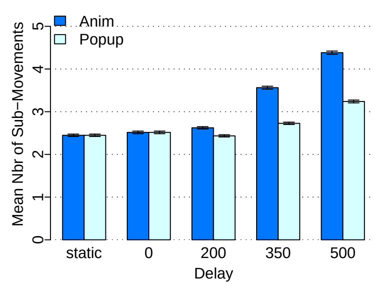
The above phenomenon can be explained by the fact that participants start their acquisition movement as soon as possible, i.e., before the end of the animation or before the target pops up, and that this ballistic movement is more precise when the target pops up or stops its animation earlier. For example, with DELAY = 500 and D = 256, participants can typically move the pointer close to the target before it pops up (or finishes its animation), but then have to wait before performing the final adjustment. On the other hand, with DELAY = 200 and D = 768, the target pops up (or finishes its animation) before the end of the ballistic movement and participants can adjust their movement as they go.

Regarding errors, we measured an error rate of 5.25%, a typical value for a pointing experiment. Figure 3.(b) shows the error rate as a function of DELAY for each TECH. Logistic two by two Pearson tests[7] show a significant difference ($\chi^2 = 4.880$, $p = 0.0272$ for DELAY = 0) between $Static$ (an error rate of 3.87%) and the other DELAY conditions (error rates between 5% and 6%.) There is no significant difference for errors among the remaining DELAY conditions. Other Pearson tests did not reveal any significant effect on errors for the other factors TECH, D and W.

6 Kinematic Analysis



(a)



(b)

Fig. 4. (a) Time before first mouse move as a function of DELAY and TECH. (b) Mean number of sub-movements as a function of DELAY and TECH.

[7] tests adapted to "discrete" measures.

A more detailed analysis of the pointing movements shows that participants effectively start their movement before the target pops up (when TECH = $Popup$ and DELAY > 0) or before the target ends its animation (when TECH = $Anim$ and DELAY > 0). Figure 4.(a) shows the time taken by the participants between the mouse button release (when they click on the red starting target) and the first mouse movement. These are clearly shorter than DELAY when DELAY > 0. Moreover, an ANOVA shows no significant difference between the DELAY conditions for TECH = $Popup$, nor between the non-zero DELAY conditions for TECH = $Anim$. However, as shown in Figure 4.(a), the first mouse move times for TECH = $Anim$ and DELAY > 0 are significantly longer than those for the other conditions (~ 40 *ms*).

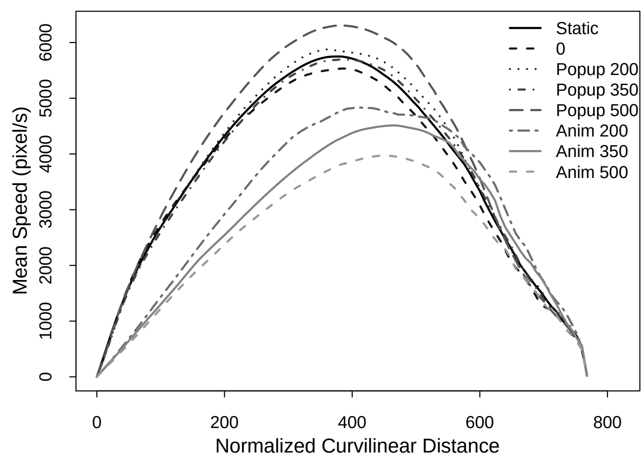


Fig. 5. Mean speed as a function of the (normalized curvilinear) distance for D = 768 for each DELAY × TECH condition (grey curves with a lower velocity peak are the $Anim$ curves)

An analysis of the kinematic record of the movements shows that $Popup$ and $Anim$ have slightly different kinematic profiles (see Figure 5 for an example with D = 768, similar shapes arise with the other distances). The distance/speed curve for $Popup$ is similar to those for the $Static$ and DELAY = 0 conditions: they all have the usual bell-shaped curve with a velocity peak at about 50% of the distance. The curves for $Anim$ however are qualitatively different: they are right-skewed with a lower acceleration and a velocity peak at about 60% of the distance to target (these differences can be shown to be statistically significant). Moreover, in the $Anim$ condition we observe slower movements and in particular a lower velocity peak as DELAY increases. In the $Popup$ condition we do not observe any such pattern.

A possible explanation of these phenomena is that in the $Anim$ condition the participants try to follow the moving target or are distracted by it, leading to a slower movement. Conversely, in the $Popup$ condition, participants move directly to the memorized position of the target, leading to a movement close to the classical target acquisition profile (note that in Figure 5.(a) the movement is even faster for DELAY = 500, but this is not the case for the other distances).

To confirm the above interpretation and to better understand the end of the movement we computed the number of sub-movements for all non-error trials. To do so, we count the number of times when the mouse does not move during at least 50 *ms* (to account for a null velocity) after the velocity peak and before the final mouse press. We add one to this number to account for the final movement, and interpret this number as a good estimate of the number of sub-movements. Figure 4.(b) shows the mean value of this number as a function of TECH and DELAY. An analysis of variance shows a significant effect of TECH and DELAY and an interaction between these two factors[8]. There are more sub-movements for $Anim$ than for $Popup$ and the difference (about one sub-movement) is significant for DELAY > 200.

In summary, we found that while $Popup$ and $Anim$ have similar performance, the pointing motion for $Popup$ is closer to static pointing than to $Anim$. While $Popup$ seems to follow a simple amplitude control model (jump to the anticipated position of the target and then adjust), $Anim$ features a more complex movement.

7 Extending Fitts' Law

In order to better understand the effect of DELAY we used Fitts' Law to analyze movement time. Table 2 shows the Fitts' Law parameters and the adjusted r^2 for all data by DELAY. We averaged movement times across participants, repetitions and TECH since we have shown that this factor did not have a significant effect on MT. In other words, we take the mean of all trials for each DELAY, D and W conditions. As we expected, the fit for the complete dataset is not good (strong effect of DELAY, see Figure 6.(a)) but the fit for each DELAY condition is good for $Static$ (usual Fitts' pointing) and degrades as DELAY increases because of the unusual effect of distance on movement time. The fit for each DELAY condition can be improved by using the classical Welford model [27] $MT = a + b.log_2(D) + b.log_2(W)$ (r^2 around 0.95), but this model fails to fit the complete dataset ($r^2 = 0.5448$).

DELAY	a	b	adj. r^2	# pts.
ALL	454	135	0.4998	45
$Static$	180	174	0.9656	9
0	299	153	0.9074	9
200	462	120	0.8973	9
350	599	116	0.7974	9
500	731	115	0.6654	9

Table 2. Fitts' Law parameters $MT = a + b.ID$

[8] D and W also have significant effects – more sub-movements for smaller W and more sub-movements for D = 512 and 768 than for 256 – but there is no significant interaction effect with TECH and DELAY.

In order to compute regression models that include DELAY, we create a continuous factor TV as follows: We map $Static$ to 0 *ms*; We map DELAY = 0 to 100 *ms* since this is the delay needed for sensory information to affect the physical movement [17]; Finally we map the other values of DELAY to themselves (considering the 100 *ms* we added for DELAY = 0 as a threshold).

We can now consider the simple model $MT = a + b.ID + c.TV$, which dramatically improves the fit: $MT = 307 + 135.ID + 0.64.TV$ with an adjusted r^2 of 0.8921. This fit can be improved by using Welford's model instead: $MT = a + b.log_2(D) + c.log_2(W) + d.TV$, which yields $MT = 977 + 74.log_2(D) - 158.log_2(W) + 0.64.TV$ with an adjusted r^2 of 0.9478. This can be further improved, only slightly but significantly, to 0.9665 by adding yet another term, $log_2(D) * TV$, to account for the D by DELAY interaction observed in the previous section).

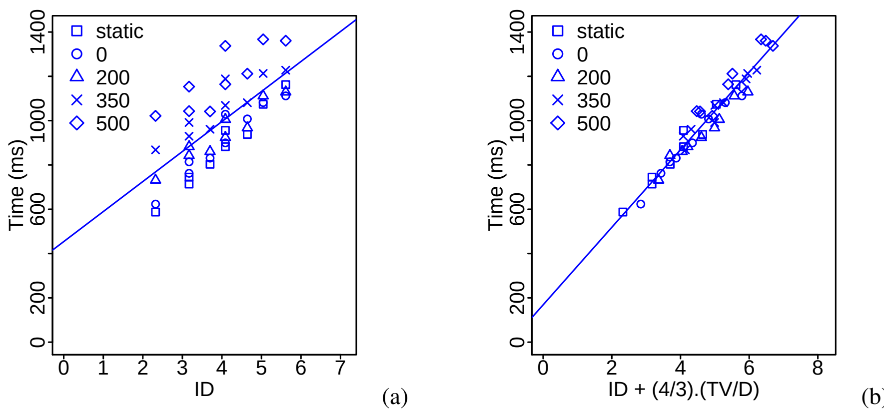


Fig. 6. Regression with Fitts' Law (a) and with the modified ID including a $\frac{TV}{D}$ term (b)

However good the fit, none of these models are particularly intuitive. They simply use the fact that DELAY has an effect and that distance has an unusual interaction with DELAY. But having more than three free variables ($a, b, c, ...$) is problematic if we are to be consistent with Fitts' Law since we have only one additional main factor, DELAY.

We have seen in the previous section that movement time increases when TV increases but that this depends on distance: as distance increases, the degradation on movement time for a given value of TV decreases. Thus, the combined effect of TV and D may be captured by the ratio $\frac{TV}{D}$. We therefore consider the following model:

$$MT = a + b.ID + c.\frac{TV}{D}$$

The regression with this model yields $MT = 171 + 174.ID + 237.\frac{TV}{D}$ and a good adjusted $r^2 = 0.9414$. Taking advantage of the fact that $\frac{237}{174} \sim \frac{4}{3}$, we use the new index of difficulty $ID + \frac{4}{3}.\frac{TV}{D}$ to plot the resulting data in Figure 6.(b).

Of course, the validity of this model should be further tested and may be altered by considering larger sets of distances, target widths and DELAY conditions. Note that an important property of our experimental design is that the values for DELAY are shorter than the expected movement times in the $Static$ condition. It is fairly obvious that if TV is larger than the pointing time in the static condition, a different model is likely to apply since the user would have to move the pointer towards the memorized position of the target, wait until the target pops up (after TV *ms*), and finally acquire the target of width W at a distance $err(D)$ where err is a function modeling the distance error when pointing to a memorized invisible target at a distance D. A candidate model for this situation could be $MT = TV + a + b.\log_2(\frac{err(D)}{W} + 1)$, but remains to be tested.

8 Discussion and Future Directions

The work presented in this paper is the first study of acquisition of pop-up and animated targets. We did not find significant differences in performance between the acquisition of a static target and the acquisition of a "memorized" target that pops up immediately. However, we found a significant difference regarding errors in favor of static pointing and we also found that static pointing is significantly faster than pointing on animated and pop-up targets with a delay longer than 200 *ms*. This suggests that delays and animations can indeed impair performance, and that techniques that keep the context (and the target) visible should be preferred.

We also did not find significant performance differences between the acquisition of targets that pop up immediately and animated or pop-up targets with a delay of 200 *ms*. However, we observed large differences when delays increase to 350 and 500 *ms*, the magnitude of the difference being comparable to the increase in delay. This suggests that animation should be kept close to a duration of 200 *ms* whenever the acquisition of a target inside the animated object is desirable. Also, GUIs should do their best to keep the delay for popping up a potential target under 200 *ms*. Note however that other design factors may come into play in the real world that are more important than pointing performance, such as the ability to perceive causality with animations.

Our study suggests that users are able to adapt their pointing movement dynamically. It is interesting to note that this adaptation depends on the feedback given to the user: animations affect the nature of the movement, slowing it down and leading to more sub-movements than popping up. Since we did not observe a significant difference in performance between the acquisition of an animated target and a pop-up target (under the same delay) it is possible that users are able to take advantage of the target animation during the acquisition movement to better predict the final position of the target. We intend to pursue this hypothesis in future work.

Another area for future work is to extend the scope of the study. In this initial study, we conducted a "pure" experiment idealizing the real world. Now that we have a better understanding of the basic factors involved, we can start to examine more realistic situations, such as 2D pointing with real windows and icons. We also want to consider more complex tasks that involve pop-up/animated targets such as those described in section 3. Finally we need to study the effects of other factors such as the animation type, e.g., slow-in/slow-out, and the degree of position memorization. For pop-up targets, we also plan to investigate larger delays and the extreme case of memorized invisible targets in order to refine our movement time model.

9 Acknowledgements

We wish to thank the member of the InSitu team and in particular Stephane Huot, our experiment participants and the anonymous reviewers for their feedback.

References

1. Accot, J., Zhai, S.: Refining Fitts' law models for bivariate pointing. In: Proc. CHI '03, ACM (2003) 193–200
2. Aivar, M., Hayhoe, M., Chizk, C., Mruczek, R.: Spatial memory and saccadic targeting in a natural task. Journal of Vision **5**(3:3) (2005) 177–193
3. Appert, C., Beaudouin-Lafon, M.: SwingStates: Adding state machines to Java and the Swing toolkit. Software: Practice and Experience **38**(11) (2008) 1149 – 1182
4. Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P., Bederson, B., Zierlinger, A.: Drag-and-pop and drag-and-pick: Techniques for accessing remote screen content on touch- and pen-operated systems. In: Proc. INTERACT '03, IOS Press, IFIP (2003) 57–64
5. Bezerianos, A., Balakrishnan, R.: The vacuum: facilitating the manipulation of distant objects. In: Proc. CHI '05, ACM (2005) 361–370
6. Bock, O., Eckmiller, R.: Goal-directed arm movements in absence of visual guidance: evidence for amplitude rather than position control. Exp. Brain Res. **62**(3) (1986) 451–458
7. Cao, X., Li, J.J., Balakrishnan, R.: Peephole pointing: Modeling acquisition of dynamically revealed targets. In: Proc. CHI '08, ACM (2008) 1699–1709
8. Chapuis, O.: Gestion des fenêtres: enregistrement et visualisation de l'interaction. In: Proc. IHM '05, ACM (2005) 255–258
9. Cockburn, A., Gutwin, C., Greenberg, S.: A predictive model of menu performance. In: Proc. CHI '07, ACM (2007) 627–636
10. Faure, G., Chapuis, O., Roussel, N.: Power tools for copying and moving: Useful stuff for your desktop. In: Proc. CHI '09, ACM (2009) To appear
11. Fitts, P.M.: The information capacity of the human motor system in controlling the amplitude of movement. Journal of Experimental Psychology **47** (1954) 381–391
12. Flash, T., Henis, E.: Arm trajectory modifications during reaching towards visual targets. J. Cognitive Neuroscience **3**(3) (1991) 220–230
13. Gonzalez, C.: Does animation in user interfaces improve decision making? In: Proc. CHI '96, ACM (1996) 27–34
14. Hinckley, K., Guimbretiere, F., Baudisch, P., Sarin, R., Agrawala, M., Cutrell, E.: The springboard: multiple modes in one spring-loaded control. In: Proc. CHI '06, ACM (2006) 181–190
15. Hoffmann, E.R.: Capture of moving targets: a modification of Fitts' law. Ergonomics **34** (1991) 211–220
16. Hornof, A.J., Kieras, D.E.: Cognitive modeling demonstrates how people use anticipated location knowledge of menu items. In: Proc. CHI '99, ACM (1999) 410–417
17. Jeannerod, M.: The neural and behavioural organization of goal directed movements. Clarendon Press, Oxford (1988)
18. Kurtenbach, G., Buxton, W.: Issues in combining marking and direct manipulation techniques. In: Proc. UIST '91, ACM (1991) 137–144
19. MacKenzie, I.S.: Fitts' law as a research and design tool in human-computer interaction. Hum.-Comput. Interact. **7** (1992) 91–139
20. Meyer, D., Smith, J., Kornblum, S., Abrams, R., Wright, C.: Optimality in human motor performance: Ideal control of rapid aimed movements. Psych. Review **95** (1988) 340–370

21. Mould, D., Gutwin, C.: The effects of feedback on targeting with multiple moving targets. In: Proc. GI '04, Canadian Hum.-Comp. Comm. Soc. (2004) 25–32
22. Obendorf, H., Weinreich, H., Herder, E., Mayer, M.: Web page revisitation revisited: Implications of a long-term click-stream study of browser usage. In: Proc. CHI '07, ACM (2007) 597 – 606
23. Plamondon, R., Alimi, A.: Speed/accuracy trade-offs in target-directed movements. Behavioral and Brain Sciences **20**(2) (1997) 279–349
24. Schlienger, C., Conversy, S., Chatty, S., Anquetil, M., Mertz, C.: Improving users' comprehension of changes with animation and sound: An empirical assessment. In: Proc. INTERACT '07, LNCS 4662, Springer (2007) 207–220
25. Tabard, A., Mackay, W., Roussel, N., Letondal, C.: Pagelinker: integrating contextual bookmarks within a browser. In: Proc. CHI '07, ACM (2007) 337–346
26. Thomas, B.H., Calder, P.: Animating direct manipulation interfaces. In: Proc. UIST '95, ACM (1995) 3–12
27. Welford, A.T., Norris, A.H., Shock, N.W.: Speed and accuracy of movement and their changes with age. Acta. Psychol. **30** (1969) 3–15
28. Zhai, S., Conversy, S., Beaudouin-Lafon, M., Guiard, Y.: Human on-line response to target expansion. In: Proc. CHI '03, ACM (2003) 177–184

Guillaume Faure, Olivier Chapuis and Michel Beaudouin-Lafon. **Acquisition of Animated and Pop-up Targets**. In **INTERACT '09**: *Proceedings of the 12th IFIP TC13 Conference in Human-Computer Interaction* - LNCS 5727. Springer & IFIP, 372–385, 2009.
