Modeling User Performance for Moving Target Selection with a Delayed Mouse

Mark Claypool[1], Ragnhild Eg[2], and Kjetil Raaen[2]

[1] Computer Science, Worcester Polytechnic Institute
Worcester MA 01720, USA
claypool@cs.wpi.edu
[2] School of Art, Communication and Technology, Westerdals – Oslo
Christian Krohgs gate 32, 0186 Oslo
{egrag|raakje}@westerdals.no

Abstract. The growth in networking and cloud services provides opportunities to host multimedia on remote servers, but also brings challenges to developers who must deal with added delays that degrade interactivity. A fundamental action for many computer-based multimedia applications is selecting a moving target with the mouse. While previous research has modeled both moving target selection and target selection with delay, there have not been models of moving target selection with delay. Our work presents a user study that measures the effects of delay and target speed on the time to select a moving target with a mouse, with analysis of trends and derivation of a model. The analysis shows delay and speed impact target selection time exponentially and that selection time is well-represented by a model with three terms - two with exponential relationships for delay and speed and one an important interaction term.

1 Introduction

Interactive multimedia applications vary in purpose and content, but some user actions are nearly ubiquitous. One such action is using a mouse or similar device to select a target in an interface, a task common from Web browsers and spreadsheets to video editors and computer games.

Although target selection is commonly in use already, target selection with delay has not been thoroughly explored or modeled. When performing target selection with a mouse, there is always delay between moving and clicking the physical mouse and the corresponding rendered results on the screen. While the delay due to the local device is often negligible, when used in ever more popular cloud computing, all actions are rendered on a cloud server, accumulating delays from the local client, network and remote server [1]. These delays may affect the interaction in at least two ways: 1) reducing user performance and, thereby, productivity, and 2) degrading the quality of experience and causing frustration.

Fitts' early seminal work in target selection [2] provided a robust model (Fitts' Law) for the time it takes a user to select a target of a specified size at

a specified distance. Extensions to this work expanded the model to two dimensions (such as for a computer mouse) [7] and moving targets [6]. While effective for modeling target selection, the models fail to incorporate delays present in Internet multimedia applications. A model that does incorporate delay [5] concentrates on much higher delays than are typically encountered on the Internet and does not combine delay with moving target selection. Ideally, real-time, multimedia developers would have a model as far reaching and robust as Fitts' Law, but incorporating moving targets, common to many multimedia applications (e.g., computer games) and delay, present in all networked multimedia applications. Our work takes a first step towards providing such a model.

We design and implement a game that isolates the fundamental action of selecting a moving target with a mouse and allows for control of the delay between the user input and the rendered action. With our game, task performance is the time it takes the user to hit the target and quality of experience is the user's subjective opinion of the game's responsiveness. The game is central to a user study of over 80 total participants, studying added delays ranging from 0 to 400 milliseconds and target speeds ranging from 150 to 1550 pixels/second.

Analysis of the results shows the time to select a moving target with a mouse increases exponentially with both delay and target speed – this is in contrast to earlier works that showed a linear relationship with delay [5] and a linear relationship with speed [6]. The combination of delay and speed results in higher times than either alone would suggest. As such, we propose a model for the time to select a moving target with delay that uses exponential terms for both delay and target speed and has a combined interaction between them. Our model fits our experimental data well and provides a basis for future models and multimedia development. Brief analysis of the quality of experience shows a marked decay in perceived responsiveness with delay.

The rest of this paper is organized as follows: Section 2 describes our methodology, including game development and user study; Section 3 analyzes the user study results for trends and presents our derived model; and Section 4 summarizes our conclusions and presents possible future work.

2 Methodology

To investigate how delay affects users when selecting a moving target with a mouse, we deployed the following methodology: 1) Design and develop a game that isolates the target selection action with controlled delay (Section 2.1); 2) Conduct user studies with the game to evaluate the impact of delay on moving target selection, measuring game performance and perceived game responsiveness (Section 2.2); and 3) Analyze the user study results and develop an analytic model for moving target selection with delay (Section 3).

2.1 Application for Experiments

In order to avoid the monotony that often occurs during behavioral experiments, we designed the experimental task as a game that requires both temporal and

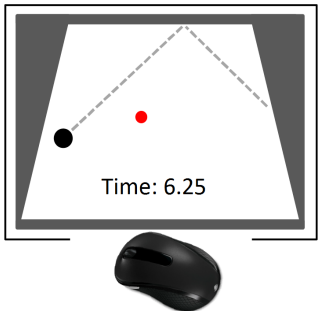

Fig. 1: Puck Hunt. Players select moving target (black puck) with mouse cursor (red ball), with controlled delay and varied target speeds.

spatial precision isolated in mouse interactions. In the game *Puck Hunt*[3], the objective is simple – move the cursor and click on the target as quickly as possible. To minimize the inherent local delay, Puck Hunt is written in C++ using OpenGL with the Angel 2D[4] game engine.

Puck Hunt consists of a series of short rounds; each round starts with a randomly placed large, black puck (the target) and a smaller red ball (the mouse cursor). The puck bounces around in a space confined by the screen according to simple 2d physics. The user moves the mouse to control the ball, and attempts to "hit" the puck by placing the ball over the puck and clicking the mouse button. Once the user has successfully hit the puck, the puck disappears and a notification pops up telling the user to prepare for the next round. Upon pressing space, a new round starts, with the puck at a new starting position and velocity. User performance is defined by the time it takes to complete a round, shown by a timer that counts up from zero. Puck Hunt adds a configurable amount of delay to the mouse input, both moving and clicking, and uses a configurable set of puck speeds to control game difficulty. Since rounds with fast pucks and high delays are quite difficult, Puck Hunt has an upper time limit of 30 seconds to avoid situations where a user may just give up. So a recorded score of 30 or higher indicates the game task was not completed the task for that round.

Our experiments had two different versions of the game at separate university locations – Worcester Polytechnic Institute (WPI) and Westerdals-Oslo (WO) – the sole difference being the speeds of the puck. While the differences in the puck speeds at each university was an inadvertent side effect of the game engine on a Mac (at WO) versus Windows (at WPI), the results do provide for analysis of a wider speed spread than would the same speeds at each university. Both game versions use 3 possible speeds varied randomly between rounds, for a total of 6 speed levels total, shown in Table 1a. Effectively, these speeds create different levels of difficulty.

[3] A pun on the classic game *Duck Hunt* (Nintendo, 1984).
[4] `http://angel2d.com/`

Table 1: Independent variables for user study.

Speed (pixels/second)	
WPI: Slow	150
WPI: Medium	300
WPI: Fast	450
WO: Slow	550
WO: Medium	1100
WO: Fast	1550

(a) Target speeds.

Delay (milliseconds)
0, 25, 50, 75
100, 125, 150, 175
200
300
400

(b) Added mouse input delays.

Both game versions add a controlled amount of delay selected from a set of 11 possible values, shown in Table 1b. The delay is added to all mouse movements and button clicks for the duration of the round. Each delay & speed combination is presented to each user 5 times to control for variation, and all combinations and repetitions are shuffled so as to appear in a random order and prevent learning effects.

Every 30 rounds, the game stops for a minimum of 20 seconds to allow the user to rest his/her visual focus and regain concentration, displayed as a countdown timer via a popup window.

Puck Hunt runs in fullscreen mode at 1080p resolution (1920x1080 pixels). The puck is 100 pixels in diameter and the red ball (the mouse cursor) is 25 pixels in diameter.

2.2 User studies

To test a wide range of delay and speed combinations, experiments were run at our two locations in parallel, Worcester Polytechnic Institute (WPI) and Westerdals-Oslo (WO). Conditions and procedures for the participants were matched as closely as possible, but practicalities necessitate some differences.

At WPI, 32 participants were solicited through advertising via university email lists. Incentives included a $25 gift card raffle for participating and a $25 gift card for the user with the highest score. Game development students that participated also received 1 extra point on their final exams. At WO, 52 participants were recruited by approaching students on campus and asking them if they wanted to take part in a research experiment. A bag of candy was offered as incentive.

The experiments were conducted in dedicated computer labs, running up to four participants at a time at WPI, and up to five participants at a time at WO. Once participants were seated at their respective computers, they listened to a scripted brief outlining the study and signed a consent form. Next, participants were asked to make themselves comfortable by adjusting chair height and monitor angle/tilt so as to be looking at the center of the screen. At WO, participants could also adjust desk height. Importantly, participants were en-

couraged to shift the mouse to whichever hand they preferred. Participants then completed an online questionnaire that assessed potential visual impairments, handedness, computer competence and gaming experience; Table 2 summarizes the most relevant information.

Table 2: Summary of key demographics. Standard deviations in parentheses for mean values. Mean Ability is from 1 (low) to 5 (high).

				Self-reported Mean Ability	
Uni.	**Mean Age**	**Gender**	**Mouse Hand**	**PC**	**PC Game**
WPI	20.9 (1.9)	23 ♂, 8 ♀, 1 ?	32 right	3.5 (1.3)	3.5 (1.3)
WO	23.9 (3.5)	45 ♂, 8 ♀	52 right, 1 left	4.1 (1.9)	3.8 (1.0)

Upon completion of the questionnaire, play commenced immediately. The first two rounds were practice trials to familiarize participants with the game and controls – these results were not recorded. Play then proceeded through 5 iterations of shuffled combinations of all the delays (Table 1b) and puck (target) speeds (Table 1a), where WPI used only the slower 3 speeds, and WO only the faster 3 speeds. The forced break was introduced every 30 rounds.

To assess user perception of the experience, participants were asked to "rate the quality of responsiveness of the last round" (1 to 5) once for every combination of delay and puck speed (in total, 33 times).

In total, each participant played 165 recorded rounds, taking a total of 15-30 minutes.

Note, the delays in Table 1b added by Puck Hunt are in addition to any base delays inherent in the computer system. Since such base delays have been shown to be significant [8], we measured the base delay for mouse actions in each lab. At WPI, we used a Blur-busters type technique[5] whereby an instrumented mouse and a high-speed camera were used to precisely measure the delay between a mouse click and visual output. The measurement method was repeated 5 times, resulting in a mean base delay of 100 milliseconds, which is added to all subsequent delay values in the WPI data. At WO, we used an alternate technique whereby a mouse button and a photosensor connected to an oscilloscope provided precise readings of mouse input delay. Again, the measurement method was repeated 5 times, resulting in a mean base delay of 20 milliseconds which is added to all subsequent delay values in the WO data. While the techniques differed at the two universities because of available hardware (e.g., an oscilloscope, a high-speed camera), given that both techniques have milliseconds of precision, the difference in the reported base delays is likely not due to technique.

[5] `http://www.blurbusters.com/gsync/preview2/`

3 Analysis

This section first analyzes user performance in selecting a moving target using a mouse with delay (Section 3.1), derives our model (Section 3.2) and provides a brief look at quality of experience (Section 3.3).

3.1 User Performance

Participant performance is defined by the time taken to hit the puck (i.e., select the moving target with a mouse). Figure 2 depicts two graphs of selection time versus delay. The x-axes correspond the input delay and the y-axes the median time to select the puck, recorded for all participants. The two graphs use the same data, where the right-hand graph applies logscale to both axes. There are six trend-lines in each graph, one for the three puck speeds tested in the two game versions. From the graphs, there are two clear trends: 1) the time to hit the puck increases with delay, a trend that appears exponential; and 2) the time to hit the puck increases with puck speed, but only for the three fastest pucks – speed has little effect for the three slowest pucks.

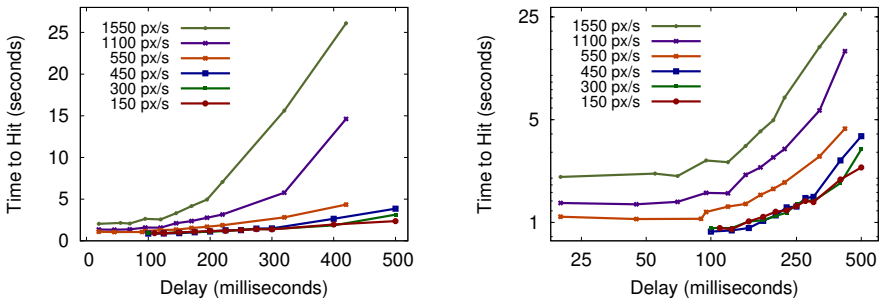

Fig. 2: Median hit time across input delays. Trend-lines are in order of decreasing puck speed, top to bottom. Same data in both graphs, right is logscale.

While the median times illustrate how moving target selection time is affected by delay, modeling from median values does not provide expected (mean) values and is not comparable to other research that models expected values.

Unfortunately, the mean values recorded for fast pucks at high delays are skewed by the 30 second time limit per round. This skew is illustrated by the cumulative distribution functions (CDFs) for hit times across delay in Figure 3 (left). The x-axis represents the time to hit the puck in seconds and the y-axis the cumulative distribution. Again, the six trend-lines correspond to the puck speeds tested, with the slowest speeds on top and to the left. In this graph,

the plotted distributions are only for total delays of about 400 milliseconds.[6] The distributions show clear separation for all trend-lines, with the slower pucks closer together than the faster pucks. The three fastest puck speeds all have maximum scores at 30 seconds and the two fastest puck speeds have a significant portion of the distribution at 30 seconds.

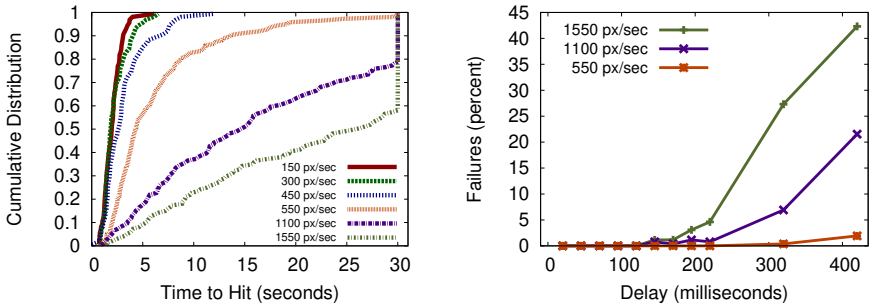

Fig. 3: Distribution of hit time (left). Percentage of failure to hit (right).

While all participants hit the puck within the 30 second time limit for the three slowest puck speeds, this was not the case for the three fastest speeds. Figure 3 (right) illustrates the rate of failure for the three fastest puck speeds. The x-axis corresponds to input delay and the y-axis the percentage of failures (based on the fraction of rounds that participants failed to hit the puck within 30 seconds). The three trend-lines represent the three fastest speeds, 550 pixels/second (px/s), 1100 px/s and 1550 px/s. For delays under 100 milliseconds, users always hit the puck within 30 seconds. However, for delays of 125 milliseconds and above, increasingly more users failed to hit the pucks moving at 1100 and 1550 px/s. At the highest added delays, 300 and 400 milliseconds, high puck speeds led to fairly high rates of failure, with a maximum failure percent of more than 40% for the 1550 px/s puck at 400 millisecond delay. This illustrates that the combination of speed and delay can severely impact selecting a moving target with delay for a range of difficulty (speed) levels.

In order to obtain representative means for puck hit times, specifically for fast speeds and high delays, we model the CDFs of the three fastest puck speeds using only scores below 30 seconds. This way, the model can be extended past the 30 second limit to provide for a full distribution range and allowing for derivation of a mean. Because CDFs naturally trend to 1, we fit[7] a logarithmic regression model to data points below 30 seconds for the rounds that had failures and then

[6] Because of differences in local latencies, total delays are 410 ms for the slowest puck speeds and 425 ms for the fastest puck speeds.

[7] Using R, `https://www.r-project.org/`

integrated to yield the means. Table 3 summarizes the results, showing good fits for the analytic models (R^2 0.93 or above).

Table 3: Analytic models for CDFs for speed & delay combinations with failures.

Speed (px/s)	Delay (ms)	Equation	R^2	Mean (ms)
550	420	$0.39 \cdot ln(x) - 0.11$	0.96	6.2
1100	320	$0.34 \cdot ln(x) - 0.09$	0.98	8.2
1100	420	$0.27 \cdot ln(x) - 0.20$	0.94	23.1
1550	195	$0.29 \cdot ln(x) + 0.05$	1.00	7.4
1550	220	$0.30 \cdot ln(x) - 0.07$	0.97	10.3
1550	320	$0.25 \cdot ln(x) - 0.17$	0.94	30.0
1550	420	$0.21 \cdot ln(x) - 0.22$	0.93	66.8

For illustration, Figure 4 depicts the CDFs of the 1550 px/s puck speed with the models. The x-axis corresponds to the hit time and the y-axis the cumulative distribution. The six trend-lines correspond to the input delay, beginning with the lowest delays on top and to the left. The distribution builds on users' mean data for delays of 20 and 120 milliseconds, while data for 220, 320 and 420 milliseconds delays are derived from the models in Table 3.

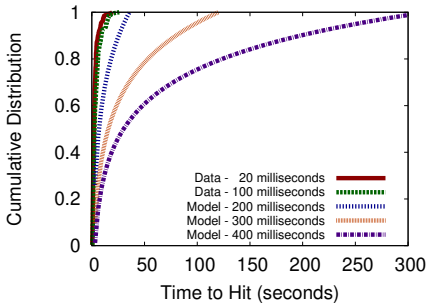

Fig. 4: Distribution of hit time. Trend-lines are in order of increasing delay, left to right. Delays of 220 ms and above are derived from models in Table 3.

Using the mean values both computed and modeled, Figure 5 presents graphs for mean hit time versus delay. In both graphs, the x-axes correspond to input delay and the y-axes to the time to hit the puck. The two graphs are based on the same data, with logscale applied to the right-hand graph. The six trend-lines belong to each of the six puck speeds. From the graphs, similar trends hold as for the median graphs (Figure 2) – an exponential growth in the mean time

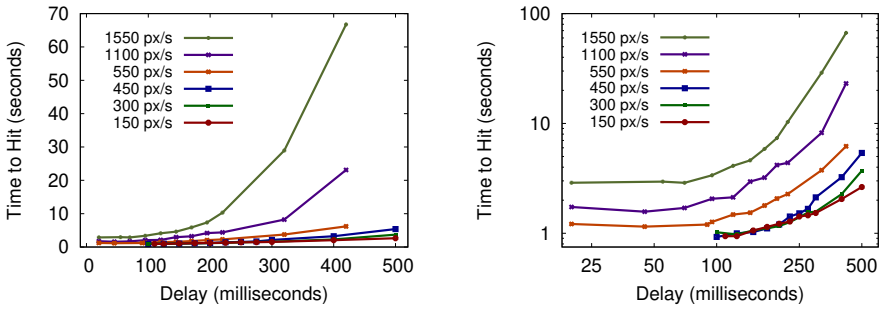

Fig. 5: Mean time to hit puck versus delay. Trend-lines are in order of decreasing puck speed, top to bottom. Same data in both graphs, right is logscale.

to hit a puck with delay. Also similar, the mean times show no difference for the three slowest pucks and for delays less than 175 milliseconds. An additional observation relates to the flatness of the mean distributions for short delays – specifically, there is a near-absence of an increase in targeting time for the three fastest pucks with delays below 100 ms. This observation suggests that moving target selection is only impacted by delays once delays exceed 100 ms.

In Figure 6, the same data are clustered by speed. Here, the x-axis defines the speed, in pixels per second, and the y-axis the time to hit the puck. Both axes are logscale. The six trend-lines in each graph represent six different delay values, ranging from 100 to 420 ms.[8] Delays of 100 & 120 ms correspond to approximately the same total delays, same as 200 & 220 ms and 400 & 420 ms (the discrepancies are attributed to the base delays); the pairs are plotted with the same colors. The graph shows a visual trend with gradually increasing target selection time as puck speeds increase (the left side of the graph), followed by sharper increase for the longer delays (on the right). The overall trend appears exponential with speed.

3.2 Model

While trends in user performance with delay provide valuable insights, an analytic model illustrating the relationships can be used by developers and researchers for designing and implementing more effective interactive systems and applications in the presence of delay. Hence, we model the mean time to select a moving target with a delayed mouse.

Based on the previous analysis (e.g., Figure 5), there is a clear upward trend in mean time to select a moving target with increased delay, a trend that appears exponential. The trends for mean time to select a moving target with increased speed are less clear, being flat for low delays and speeds (e.g., Figure 6), but

[8] Not all delay values studied are shown in the graph to keep it readable.

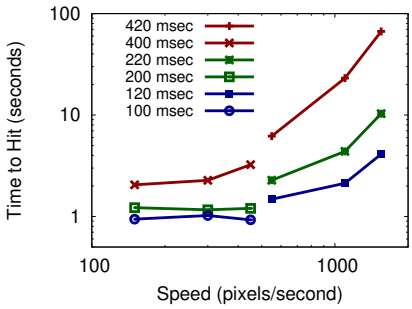

Fig. 6: Mean time to hit puck versus speed. Trend-lines are in order of decreasing delay, top to bottom.

overall also appears exponential. Moreover, previous work has suggested an exponential relationship with speed in the time to select a moving target without delay [4, 3]. The greater increase in time to select a moving target at fast speeds and high delays suggests an interaction between speed and delay.

Thus, we propose modeling the time to select a moving target with a mouse (T) with exponential functions for delay (D) and speed (S), as well as an interaction term:

$$T = k_1 + k_2 \cdot e^D + k_3 \cdot e^S + k_4 \cdot e^D \cdot e^S \tag{1}$$

where k_1, k_2, k_3 and k_4 are constants determined empirically through user study. We standardized our user study data[9] using the values in Table 4.

Table 4: User study data values used for standardization.

Factor	Mean	StDev
Delay	206 milliseconds	122
Speed	683 pixels/second	488

Fitting a regression model to the standardized data[10] yields an adjusted R^2 0.99, F-stat 3151 and $p < 2.511 \times 10^{-16}$, and the simplified final model for the time to select a moving target with a delayed mouse (T):

$$T = 1 - 0.7e^d - 0.5e^s + 2e^d e^s \tag{2}$$

where d and s are the standardized delay (D) and speed (S), respectively – $d = \frac{D-206}{122}$ and $s = \frac{S-683}{488}$.

[9] Subtracting the mean and dividing by the standard deviation.
[10] Using R, `https://www.r-project.org/`

3.3 Quality of Experience (QoE)

Our presented model for moving target selection with a delayed mouse provides insights into how user performance deteriorates with delayed visual feedback, but does not capture the user's quality of experience (QoE) with a delayed system. The user study experiments gathered QoE in the form of users' opinions on the responsiveness from 1 (worst) to 5 (best), which we analyzed and then graphed in Figure 7. The x-axis corresponds to input delay and the y-axis the QoE. As before, the six trend-lines correspond to the puck speeds. The QoE decreases with increasing delay values without a clear trend with puck speed. These data could indicate a user's QoE for responsiveness is predominantly affected by the mouse delay and not the moving target's speed. However, given the variance in the trend-lines, further study is warranted.

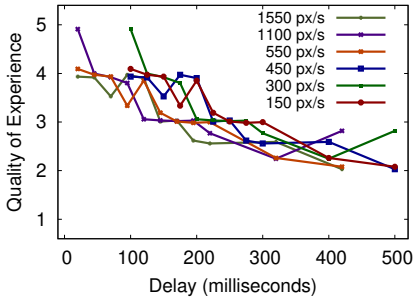

Fig. 7: QoE versus delay.

4 Conclusion

Many multimedia applications involve target selection with a mouse, and in many cases (e.g., computer games) the targets are moving. With the growth in networking and cloud services, applications are increasingly hosted on servers, adding significant delay between user input and rendered results. In order to build systems that accommodate delay, multimedia designers benefit from improved understanding of human performance with delay. This work presents a user study and analysis that provides the foundation for a model of target selection in the presence of delay.

We design and develop a custom game that focuses on moving target selection while allowing explicit control of input delay. The game is the basis for experiments with over 80 participants, providing over 32k game sessions with target speeds from 150–1550 pixels/second and total delays from 20–500 milliseconds.

Our findings reveal a rise in target selection time that is exponential with both delay and target speed. Slow speeds (below 550 pixels/second) and low

delays (below 100 milliseconds total) have little impact on target selection time. However, delay and speed added to the difficulty of the selection task. Fast speeds (550+ pixels/second) yielded longer selection times at all delays, while high delays (100+ milliseconds total) extended selection times at all speeds. Moreover, an interaction between speed and delay demonstrates the difficulty in selecting a target that is both severely delayed and moving fast and takes longer than the combination of speed and delay would suggest. Preliminary analysis of QoE indicates that ratings for responsiveness may only be affected by delay, independent of target speed, with further confirmation required.

Based on the trends, we model the time to select a moving target with a delayed mouse with additive exponential terms for delay and speed and a multiplicative interaction term. This last term is critical for accuracy as models tried without delay & speed interaction fit poorly. As proposed, the model has an excellent fit (adjusted R^2 of 0.99) and should be relatively simple to use in multimedia development and assessment.

Applying this work to practical scenarios can validate our model, while delving deeper into user backgrounds, such as age or gaming experience, may provide potential predictors to target selection performance. Future investigations could examine other forms of interactions, input devices, or measures of performance. In some cases, QoE is as important as performance, possibly warranting a model that includes both performance and QoE, as dependent on delay and task difficulty.

References

1. Chen, K.T., Chang, Y.C., Hsu, H.J., Chen, D.Y., Huang, C.Y., Hsu, C.H.: On the Quality of Service of Cloud Gaming Systems. IEEE Transactions on Multimedia 26(2) (Feb 2014)
2. Fitts, P.M.: The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement. Journal of Experimental Psychology 47(6) (1954)
3. Hajri, A.A., Fels, S., Miller, G., Ilich, M.: Moving Target Selection in 2D Graphical User Interfaces. In: Proceeding of IFIP TC Human-Computer Interaction (INTERACT). Lisbon, Portugal (Sep 2011)
4. Hoffmann, E.: Capture of Moving Targets: A Modification of Fitts' Law. Taylor & Francis Ergonomics 34(2) (1991)
5. Hoffmann, E.: Fitts' Law with Transmission Delay. Taylor & Francis Ergonomics 35(1) (1992)
6. Jagacinski, R., Repperger, D., Ward, S., Moran, M.: A Test of Fitts' Law with Moving Targets. The J. of Human Factors and Ergonomics Society 22(2) (Apr 1980)
7. MacKenzie, I., Buxton, W.: Extending Fitts' Law to Two-dimensional Tasks. In: Proceedings of ACM CHI Human Factors in Computing Systems (1992)
8. Raaen, K., Petlund, A.: How Much Delay Is There Really in Current Games? Proceedings of the 6th ACM Multimedia Systems Conference (2015)