Datasets – Moving Target Selection with Delay

Shengmei Liu
sliu7@wpi.edu
Worcester Polytechnic Institute
Worcester, Massachusetts, USA

Mark Claypool
claypool@wpi.edu
Worcester Polytechnic Institute
Worcester, Massachusetts, USA

Andy Cockburn
andy@cosc.canterbury.ac.nz
University of Canterbury
Christchurch, New Zealand

Ragnhild Eg
ragnhild.eg@kristiania.no
Kristiania University College
Oslo, Norway

Carl Gutwin
gutwin@cs.usask.ca
University of Saskatchewan
Saskatoon, Saskatchewan, Canada

Kjetil Raaen
kjetil.raaen@kristiania.no
Kristiania University College
Oslo, Norway

ABSTRACT

When software and hardware processes cause delay between user input and resultant output, consequences can be both harmful and annoying. In games, this type of delay can cause actions to lag behind a player's inputs. Studies on delayed game actions show negative impact on both player performance and quality of experience. Further research could expand upon this knowledge with more robust data modeling and analysis, as well as comparison to other actions. This paper describes publicly available datasets from four user studies on the effects of delay on the fundamental game action of selecting a moving target with a pointing device. They include performance data, such as time to selection, and demographic data, such as age and gaming experience. Example analysis illustrating the use of the datasets is shown, including comparison of pointing devices, demographic analysis with performance, and analytic modeling.

CCS CONCEPTS

• **Applied computing** → **Computer games**; • **Human-centered computing** → *User studies.*

KEYWORDS

latency, gamer, lag

ACM Reference Format:
Shengmei Liu, Mark Claypool, Andy Cockburn, Ragnhild Eg, Carl Gutwin, and Kjetil Raaen. 2021. Datasets – Moving Target Selection with Delay. In *12th ACM Multimedia Systems Conference (MMSys '21) (MMSys 21), September 28-October 1, 2021, Istanbul, Turkey*. ACM, New York, NY, USA,7 pages. https://doi.org/10.1145/3458305.3478455

1 INTRODUCTION

Interactive multimedia applications are pervasive, with rich animations and sounds available on devices ranging from smart-phones and tablets to laptops and personal computers to consoles and large screen TVs. Computer games are among the most popular interactive applications, with a global market that is projected to grow by about $120 Billion USD (9%) [1]. The increase in the number of gamers and the global penetration of the Internet and smart phones has fueled the growth of cloud-based games, supported by established companies like Sony, NVidia, Google and Microsoft.

With a cloud-based streaming game service, all player inputs and resulting outputs are delayed by the round-trip time from the client to server [23]. On top of that, game actions are also affected by local delay. For instance, the signal transfer from a mouse and the corresponding visual rendering of the movement can cause as much as 240 milliseconds delay [16]. Unsurprisingly, delay can have strong effects on both player performance and quality of experience (QoE) [2–4, 10, 11, 15]. To continue to design and maintain compelling games on modern systems, it is important to understand how delay effects vary across fundamental game actions.

Studies of moving target selection, an action common to many games, found a super-linear relationship with delay's impact on target selection, both time to select and accuracy, with faster targets being especially sensitive to delay [18]. On the one hand, players are able to adapt when the delay is constant and actions are predictable [21]. On the other hand, repeated motor-visual interactions are typical of games, and repetitions can make inherent delays more noticeable [19]. Moreover, the effect of delay varies across individuals [19], game types [13, 22], input devices [17], and task difficulties [12].

The effect of delay in games is clearly a multi-faceted challenge and researchers tend to approach it from different angles. Despite recent progress, future endeavours need to generalize findings beyond the relatively narrow ranges of games and applications studied. Additional analysis could also examine factors that could influence the effects of temporal delays, such as prior gaming experience or even musical experience. Local delays are fairly constant, but network delays can vary considerably, thus additional study of the effects of delay jitter on game action is warranted. And most studies have sample pools drawn from university students, where additional studies could cover a broader range of the gamer demographic for comparison to existing work.

All these research areas, and more, would benefit from access to datasets already gathered in prior studies in order to do direct analysis (e.g., modeling), reproduce prior work, or run comparative experiments with different parameters. The availability of open datasets that support comparative and repeatable experimentation

MMSys 21, September 28-October 1, 2021, Istanbul, Turkey

ACM ISBN 978-1-4503-8434-6/21/09…$15.00
https://doi.org/10.1145/3458305.3478455

is valuable to replicate research results, much needed in the interactive multimedia field. This, in turn, can help improve the performance of multimedia systems and the reproducibility of published papers.

This paper contributes four datasets from user studies with over 175 users playing two different games that measure the fundamental game action of selecting a moving target. The datasets are obtained from users playing a simple game where the player clicks on a moving target with a pointing device, with controlled delay for the input actions. User performance data includes the elapsed time to select the target as well as number of clicks with the pointing device. Demographic information gathered on users is available for correlation analysis.

The rest of this paper is organized as follows: Section 2 provides an overview of the datasets and user studies that generated them, Section 3 gives examples of analysis and modeling that can be undertaken with the data, and Section 4 summarizes the datasets and mentions potential uses and future work.

2 DATASETS

This section describes four sets of data obtained from prior user studies [5, 7, 9]: *Mouse-A*, *Mouse-B*, *Thumbstick* and *Motion*. The datasets are available through a single, public `git` repository:

git@bitbucket.org:claypool/target-selection-datasets.git

Each dataset was obtained from users playing a game with controlled amounts of delay, and all games focused on the same action of selecting a moving target with a pointing device. Target selection is a player action common to many game genres, including: 1) top-down shooters where players aim projectiles at a target (e.g., *Nuclear Throne*, Vlambeer, 2015); 2) first-person shooters (FPS) where players use a pointing device to pan the field of view and align a reticle over a moving opponent (e.g., *Call of Duty*, Activision, 2003); and 3) multiplayer online battle arenas (MOBAs) where players move a skill shot indicator with a pointing device to target a moving opponent (e.g., *League of Legends*, Riot Games, 2009).

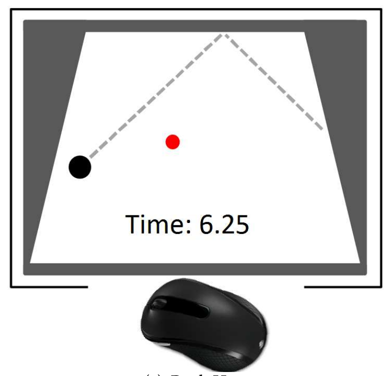
For each dataset, the researchers measured baseline local delay in the experiment setup. This is reported in the README file for each experiment in the repository. Baseline delay is the delay between input and output when no delay is added by the experiment. To get the delay experienced by the user, baseline delay has to be added to the added delay values reported in the data.

2.1 Games

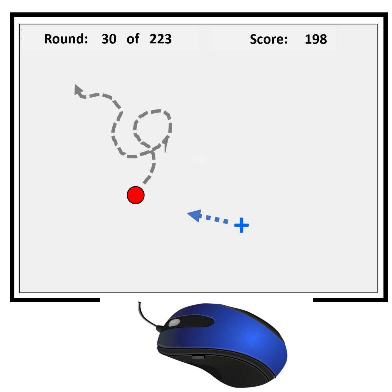
The datasets are obtained from two custom games: *Puck Hunt* for Mouse-A, Mouse-B and Thumbstick, and *Juke!* for Motion.

2.1.1 Puck Hunt. The Mouse-A, Mouse-B and Thumbstick datasets were gathered using a custom game called Puck Hunt that allows for the study of moving target selection with controlled amounts of delay. In Puck Hunt, depicted in Figure 1(a), the objective is to select the moving target, a bouncing black ball (the puck) that moves with kinematics (velocity). The user proceeds through a series of short rounds, each time using the mouse or game controller thumbstick to control the red ball on screen and move it over the target and select it with a click. When the user successfully selects the target, it disappears and a notification appears to tell the user to press any key to start the next round. With the new round, the target has a new starting location, a new orientation and a new speed. The game is scored via a timer that counts up from zero at the beginning of each round, stopping when the target is selected. To avoid user frustration, if the user fails to select the target within 30 seconds, the time 30 is recorded and the game proceeds to the next round.

Players use a mouse to control the red ball in datasets Mouse-A and Mouse-B, and a game controller in Thumbstick. The 28 mm target (the puck) moves at three different speeds (42, 84, 126 mm/s for the Mouse-A and Thumbstick datasets and 154, 308 and 434 mm/s for the Mouse-B dataset). Added delay varies between 11 values (0, 25, 50, 75, 100, 125, 150, 175, 200, 300, and 400 ms). Each combination of speed and delay is repeated five times, completely randomized. To assess quality of experience, a single 5-point Likert-scale question ("Rate the quality of experience of the last round") is presented once for each combination. Objective measures of performance recorded are the elapsed time to select the target and the number of clicks required to do so.



(a) Puck Hunt



(b) Juke!

Figure 1: Users click on the moving target with a cursor. The game adds delay to the input and varies target movement parameters between each round.

2.1.2 Juke! The fourth dataset is from a custom game called Juke!, depicted in Figure 1(b), that also allows study of target selection

Table 1: Summary of dataset game variables

Dataset	Total Rounds	Performance Measures	Test Conditions	Input Delay	Device
Mouse-A [5, 8, 9, 14]	167	time, clicks	3 speeds, 11 delays	20 ms	mouse
Mouse-B [9, 14, 20]	167	time, clicks	3 speeds, delays as in Mouse-A	100 ms	mouse
Thumbstick [6]	167	time, clicks	speeds and delays as in Mouse-A	50 ms	thumbstick
Motion [7, 18]	223	time, distance	3 turns, 3 angles, 4 delays	50 ms	mouse

with controlled amounts of delay like Puck Hunt. Juke! has the same objective, to select the moving target, a moving red ball. However, Juke!'s target movement is governed by force-based physics (acceleration), turn angle and turn frequency. Through a series of short rounds, the user controls the mouse to move the cursor (a blue '+') to the target and click it. Each round begins by clicking a green circle at the center of the screen, to ensure a consistent starting position. Once the green circle disappears, the red target appears at a random location at a short distance from the center and starts moving. The game progress is displayed in the top left corner, and the user's score is displayed in the top right corner. The score corresponds to the running total of the cursor's distance to the target when clicked, lower is better.

The 8 mm target moves with force-based physics, accelerating in the target's intended direction and with a set limit on maximum speed. The target turn interval varies between 3 values (30, 90, and 150 ms) and the target turn angle between 4 values (0, 90 and 180 degrees). The game adds a fixed amount of delay selected from 4 different values (0, 62.5, 125, and 250 ms). Each combination of turn interval, angle & delay appears 5 times, completely randomized. A single 5-point Likert-scale question ("How much lag did you experience") is presented once for each combination. Objective measures of recorded performance include elapsed time before clicking the mouse and distance between the mouse and the target when clicked.

2.1.3 Summary. Table 1 provides a summary of the main variables for the game in each dataset, with the columns as follows: "Rounds" refers to the number of game rounds the users played in Puck Hunt or Juke!; "Performance Measures" has the user performance measures gathered by the game; "Test Conditions" summarizes the game configuration conditions (i.e., target motion and delay) tested; "Local Delay" is the base delay of the system used in the study before any added delay; and "Input Device" indicates the user input device used by the game.

2.2 Users

All user studies were conducted in dedicated computer labs where users played the game in isolation on computer hardware more than adequate to support the games using LCD monitors. The Mouse-A study was conducted at a Norway university (Westerdals) while the Mouse-B, Thumbstick and Motion studies were conducted at a U.S. university (WPI). The data collected in Norway adhered to national ethical regulations. The studies conducted in the U.S. received IRB approval prior to user recruiting. For all studies, participants consented after receiving information on the purpose of the study and their rights as participants, and their data were stored and handled securely and anonymously.

The demographic questionnaires included a question on participant age. Figure 2(a) shows boxplots depicting the age distribution for each dataset. Each box depicts quartiles and median, with the mean shown with a '+'. Points higher or lower than 1.4 × the inter-quartile range are deemed outliers, depicted by the dots. The whiskers span from the minimum non-outlier to the maximum non-outlier. Most users were relatively young, reflecting the typical ages of undergraduate and graduate students from the universities from which the participants were solicited.

The demographic questionnaire also included a gender question with options for "male", "female", "other" and "prefer not to say". All participants provided an answer of male or female except for 3 users in the Thumbstick dataset and 1 user in the Mouse-B dataset.

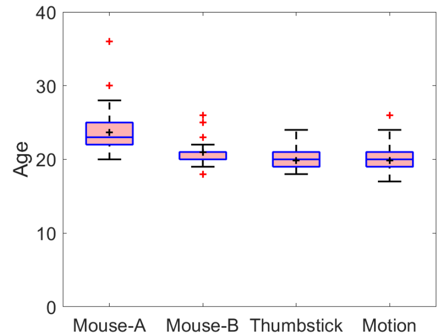
Figure 2(b) shows a stacked bar chart depicting the total number of participants in each dataset, with the blue and pink region having numbers (and corresponding percents) for males and females, respectively.

The demographic questionnaire included the question "rate yourself as a computer gamer" with responses given on a 5 point scale (1 - low to 5 - high). Table 2 shows the breakdown of self-rated skills for each dataset, with the mean and standard deviation (SD) reported by in the last two columns. All datasets have a slight skew towards higher self-rated skill (the mean self-rated skill is slightly above 3 and the mode is 4 for each dataset), but there are players of all self-rated skill levels in each set.

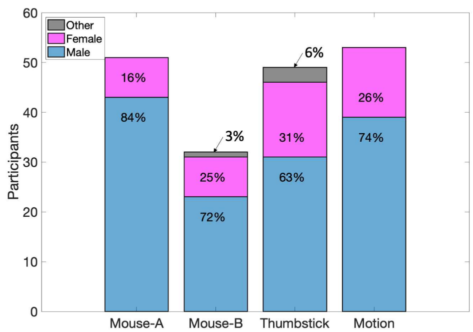
Table 2: Dataset breakdown of self-rated skill

	Self-rated skill						
Dataset	1	2	3	4	5	Mean	SD
Mouse-A	1	3	5	24	18	4.1	0.9
Mouse-B	4	2	9	8	9	3.5	1.3
Thumbstick	4	7	8	18	12	3.6	1.2
Motion	1	7	17	19	9	3.5	1.0

Table 3 provides a summary of the main user variables in each dataset, with the columns as follows: "Dataset" denotes the source and references publications that have used the dataset; "Age" is the mean participant age in years, with the standard deviation in parentheses; "Gender" gives the breakdown of number of males and females, with "?" indicating "other" or "prefer not to say"; and "Skill" provides the self-rated skill, as in Table 2.



(a) Age



(b) Gender

Figure 2: Dataset user demographics

Table 3: Summary of dataset participant variables

Dataset	Participants	Age (years)	Gender	Skill (1-5)
Mouse-A	51	23.7 (3.1)	43 ♂ 8 ♀	4.1 (0.9)
Mouse-B	32	20.9 (1.9)	23 ♂ 8 ♀ 1 ?	3.5 (1.3)
Thumbstick	49	19.8 (1.5)	31 ♂ 15 ♀ 3 ?	3.6 (1.2)
Motion	53	19.8 (1.5)	39 ♂ 14 ♀	3.5 (1.0)

2.3 Limitations

When comparing or combining the four datasets, the samples are drawn from college-aged students but a limitation is that they are from different pools: Mouse-A (a Norway University in 2016), Mouse-B (a U.S. University in 2016), Thumbstick (a U.S. University in 2017) and Motion (a U.S. University in 2018). While the U.S. studies were conducted in the same dedicated lab, the Norway study was not (obviously). In addition, the Norway study used Mac OS and Apple computers and hardware while the U.S. studies used Microsoft Windows with PC hardware.

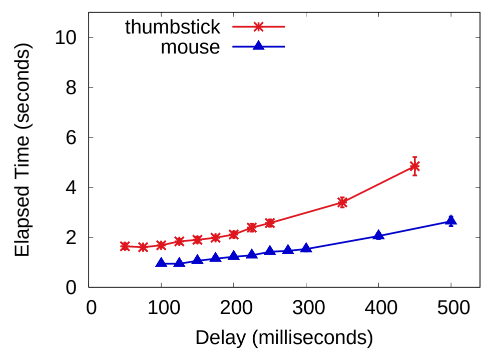
The QoE assessment was limited to a single question on responsiveness for each study. Partly, this was because the Puck Hunt and Juke! games were limited to one action and so did not provide for a full game experience, but also each participant played hundreds of rounds so time and interruptions needed to be minimized.

3 EXAMPLE ANALYSES

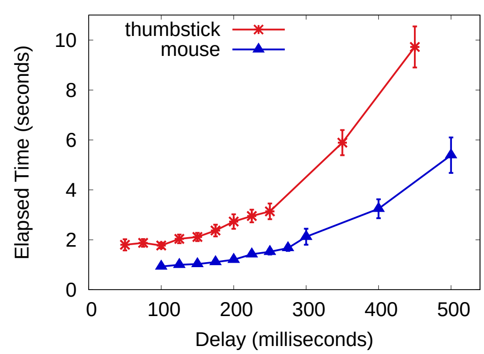
This section illustrates by example the use of the datasets in 4 different ways: devices (Section 3.1), demographics (Section 3.2), comparisons (Section 3.3) and modeling (Section 3.4). While the focus here is on the performance data, the datasets also include user quality of experience data that could be similarly analyzed.

3.1 Devices

The Mouse-B and Thumbstick datasets have identical user study parameters except for the selection device – mouse or game controller thumbstick, respectively. This allows for a comparison of the effects the two different pointing devices have on selection times. Figure 3(a) and Figure 3(b) depict graphs of selection time versus delay, with data grouped by target speed. The x-axes are the total input delays (added delay + base delay) and the y-axes are the times to select the moving target. The trendlines are for each pointing device tested. Each point is the mean time for all users for that device, delay & speed combination, shown with a 95% confidence interval.



(a) Slow Targets



(b) Fast Targets

Figure 3: Mean selection time versus delay

Overall, there is an increase in mean selection time as delay increases. This increase appears exponential over the range of delays tested. Comparing the thumbstick data to the mouse data shows the thumbstick as a selection device takes about twice as long as

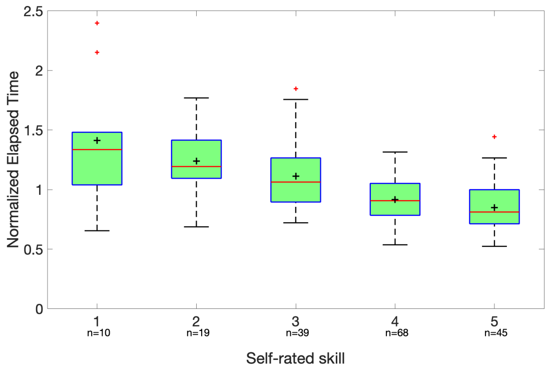


Figure 4: Elapsed time versus self-rated skill

the mouse for the same delays. Comparing Figure 3(a) to Figure 3(b) shows this 2x elapsed time holds irrespective of target speed.

3.2 Demographics

Responses to the datasets' demographics questions can be used to answer queries related to user background experience and attributes in relation to game-specific data.

Since each dataset included the question "Rate your ability as a computer gamer (1 - low to 5 - high)", correlations between answers to this question and in-game performance may indicate if self-rated game skills are reasonable indicators of actual game performance. We grouped participants' normalized target selection times (lower is better) by their self-ratings of ability. Figure 4 shows boxplots (as in Figure 2(a)) of normalized elapsed time on the y-axis for users clustered by self-rating on the x-axis. The "n=" labels on the x-axis indicate the number of participants in each self-rating group.

From the figure, the mean and median normalized elapsed times decrease (improve) approximately linearly with self-rated skill. However, the spread indicated by the boxes shows that some individuals with lower self-ratings performed better than others with higher self-ratings. A one-way between subjects ANOVA test shows there was a significant effect of self-rated skill on elapsed time at the 0.05 significance level for the five conditions, F(4, 176) = 17.86, $p < .001$.

3.3 Comparisons

Results from the target selection datasets can be compared to prior studies.

Long and Gutwin [17] measured selection time for two different sized targets (40 mm and 80 mm) that moved with simple kinematics for delays of 49, 99, 149, 249, and 299 ms. They found significant main effects for delay on selection time and that the effects of delay are exacerbated by fast target speeds.

The Mouse-A, Mouse-B datasets and the Juke! dataset (when the target does not jink) also have targets moving in a straight line (kinematics) and performance measured by target selection time. This allows for a direct, visual comparison of results, shown in Figure 5. The x-axis is the delay (in ms), the y-axis is the elapsed time to select the target (in ms). Each point is the average time to select the target at the indicated delay. There are 5 groups shown, clustered by target speed (in mm/s).

Overall, the results across the three studies present a consistent picture of a higher selection time with a higher target speed. The one anomalous result is the "Motion 13 mm/s" line which, despite having slow targets, has a higher selection time than conditions with much higher speeds (e.g., "Long 158 mm/s"). The likely reason for this difference is that the target width for the *Juke!* study was 8 mm, which is substantially smaller than the targets in Long's study (40 mm and 80 mm), and a slow but small moving target could have a similar difficulty to select as a larger, faster target.

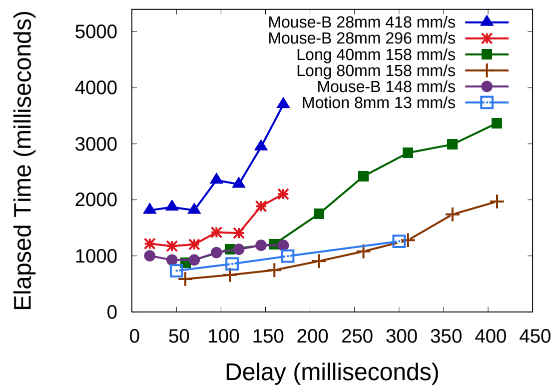


Figure 5: Selection time, targets moving in a line, data grouped by target speed

Although the slower targets appear to have flatter slopes with delay, most of the data shows a super-linear relationship with an increase in delay. The super-linear increase occurs at different delay amounts, with faster targets seeing a rise at lower delays than slower targets. This means that high-velocity targets (e.g., "Mouse-B 418 mm/s" in Figure 5) are extremely sensitive to delay.

3.4 Modeling

Analytic models of player performance lay the groundwork for a broad exploration of the impact of delay in games. The described datasets can be used to model the distribution of the times to select moving targets, providing for models to be used in simulations. We illustrate this by modeling the distribution times for the time to select a target with a mouse using the Mouse-A and Mouse-B datasets [14].

Before modeling, we standardized the delay and speed by subtracting the means (Delay 206 ms, Speed 683 px/s) and dividing by the standard deviations (Delay 122 ms, Speed 488 px/s).

Since the distribution of elapsed times appears log-normal (not shown), we use multivariate, multiple regression to model the mean and standard deviations of the logarithm of the elapsed times. This allows generation of normal distributions of the log of response times which can then be exponentially expanded to generate a distribution of elapsed times, useful for simulations.

Based on prior work ascertaining the most parsimonious models of mean elapsed time [9], we propose the following models for the

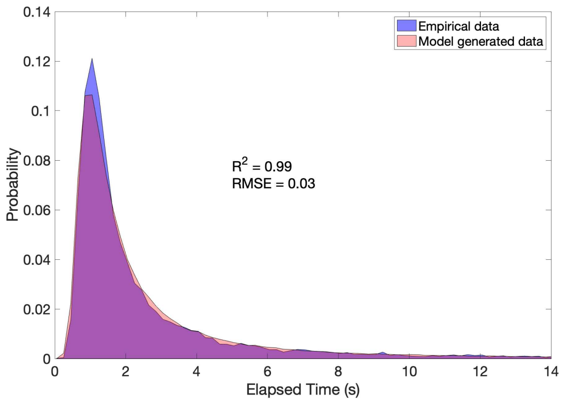


Figure 6: PDF of modeled data and empirical data

mean and standard deviation of the natural log of elapsed time (T) is:

$$
\begin{aligned}
mean(\ ln(T)\ ) &= 0.685 + 0.506d + 0.605s + 0.196ds \\
stddev(\ ln(T)\ ) &= 0.567 + 0.126d + 0.225s + 0.029ds
\end{aligned} \quad (1)
$$

where d is the standardized delay ($d = \frac{D-206}{122}$), and s is the standardized speed ($s = \frac{S-683}{488}$). The models have an adjusted R^2 of 0.96 for both.

With the models predicting mean and standard deviation for $ln(T)$, given speed and delay, the normal distribution with the predicted mean and standard deviation can generate elapsed times by taking the exponent.

Figure 6 shows the model generated data compared to the actual data. The x-axis is the elapsed time in seconds, and the y-axis is the cumulative distribution. The blue shape is the probability distribution of the elapsed time for the empirical data, and the red shape is generated from the normal distribution using the modeled mean and standard deviation from Equation 1. The model has an excellent fit for the data with R^2 of 0.99 and RMSE of 0.03.

4 SUMMARY

Interactive applications, and games in particular, are growing in popularity, fueled by the penetration of networks and the power of even small computing devices to support rich, multimedia content. Such applications come with delay local to the computing device and remote when connected via a network and a server such as with cloud-based game streaming. Delay has been demonstrated to degrade the quality of experience for players, as well as negatively affect player performance. Research on the effects of delay on player actions can benefit from additional datasets that allow for reproduction of prior experiments, modeling of the effects of delay, and comparative analysis for new user studies.

This paper presents four datasets from prior user studies that measure the effects of delay on the fundamental game action of selecting a moving target with a mouse. Nearly 200 users played about one hundred hours of games based on the mechanic of selecting a target moving across the screen, with the system controlling the amount of input delay. This paper describes the dataset variables and presents example analyses made possible with the datasest, including demographic analysis, device comparisons, comparative analysis with prior work and modeling. Accompanying this paper, we make the datasets available for use through a public `git` repository:

git@bitbucket.org:claypool/target-selection-datasets.git

The potential for future work in this area is great. Few studies look at isolated game actions, implying that there are several other interactions modes to explore with delay, such as navigation. Additional studies would also extend the grounds for comparative analyses and combined modeling and simulation.

ACKNOWLEDGMENTS

Thanks to Marco Duran and Matthew Thompson for conducting the Mouse B user study, Robyn Domanico and Linh Hoang for conducting the Thumbstick user study, and Chaiwat Ekkaewnumchai and Bhon Bunnag for development of Juke! and conducting the Motion user study.

REFERENCES

[1] 2020. *Market Report - Global Video Games Industry*. Report Linker. 167 pages.

[2] Rahul Amin, France Jackson, Juan E. Gilbert, Jim Martin, and Terry Shaw. 2013. Assessing the Impact of Latency and Jitter on the Perceived Quality of Call of Duty Modern Warfare 2. In *Proceedings of HCI – Users and Contexts of Use*. Las Vegas, USA.

[3] Anastasiia Beznosyk, Peter Quax, Karin Coninx, and Wim Lamotte. 2011. Influence of Network Delay and Jitter on Cooperation in Multiplayer Games. In *Proceedings of the International Conf. on Virtual Reality Continuum and Its Applications in Industry* (Hong Kong, China). 4 pages.

[4] K. Chen, P. Haung, G. Wang, C. Huang, and C. Lee. 2006. On the Sensitivity of Online Game Playing Time to Network QoS. In *Proceedings of IEEE Infocom*. Barcelona, Spain.

[5] Mark Claypool. 2016. On Models for Game Input with Delay - Moving Target Selection with a Mouse. In *Proceedings of the IEEE International Symposium on Multimedia (ISM)*. San Jose, CA, USA.

[6] Mark Claypool. 2018. Game Input with Delay - Moving Target Selection with a Game Controller Thumbstick. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 14, 3s (Aug. 2018).

[7] Mark Claypool, Andy Cockburn, and Carl Gutwin. 2019. Game Input with Delay - Moving Target Selection Parameters. In *Proceedings of the 10th Annual ACM Multimedia Systems Conference (MMSys)*. Amherst, MA, USA.

[8] Mark Claypool, Ragnhild Eg, and Kjetil Raaen. 2016. The Effects of Delay on Game Actions: Moving Target Selection with a Mouse. In *Proceedings of the ACM Annual Symposium on Computer-Human Interaction in Play (CHI PLAY)*. Austin, TX, USA.

[9] Mark Claypool, Ragnhild Eg, and Kjetil Raaen. 2017. Modeling User Performance for Moving Target Selection with a Delayed Mouse. In *Proceedings of the 23rd International Conference on MultiMedia Modeling (MMM)*. Reykjavik, Iceland.

[10] Tobias Fritsch, Hartmut Ritter, and Jochen H. Schiller. 2005. The Effect of Latency and Network Limitations on MMORPGs: a Field Study of Everquest 2. In *Proceedings of the 4th ACM Network and System Support for Games*. Hawthorne, NY, USA.

[11] Michael Jarschel, Daniel Schlosser, Sven Scheuring, and Tobias Hossfeld. 2011. An Evaluation of QoE in Cloud Gaming Based on Subjective Tests. In *Proccedings of the Conf. on Innovative Mobile and Internet Services in Ubiquitous Computing*. Seoul, Korea.

[12] Ricardo Jota, Albert Ng, Paul Dietz, and Daniel Wigdor. 2013. How Fast is Fast Enough? A Study of the Effects of Latency in Direct-touch Pointing Tasks. In *Proceedings of the SIGCHI conference on human factors in computing systems*.

[13] Yeng-Ting Lee, Kuan-Ta Chen, Han-I Su, and Chin-Laung Lei. 2012. Are all games equally cloud-gaming-friendly? An electromyographic approach. In *11th Annual Workshop on Network and Systems Support for Games (NetGames)*. IEEE.

[14] Shengmei Liu and Mark Claypool. 2021. Game Input with Delay - A Model of the Time Distribution for Selecting a Moving Target with a Mouse. In *Proceedings of the 27th International Conference on MultiMedia Modeling (MMM)*. Virtual Conference.

[15] Shengmei Liu, Atsuo Kuwahara, Jamie Sherman, James Scovell, and Mark Claypool. 2021. Lower is Better? The Effects of Local Latencies on Competitive First Person Shooter Game Players. In *Proceedings of theACM Conference on Human Factors in Computing Systems (CHI)*. Virtual (was Yokohama, Japan).

[16] Michael Long and Carl Gutwin. 2018. Characterizing and Modeling the Effects of Local Latency on Game Performance and Experience. In *Proceedings of the ACM Computer-Human Interaction in Play (CHI PLAY)*. Melbourne, VC, Australia.

[17] Michael Long and Carl Gutwin. 2019. Effects of Local Latency on Game Pointing Devices and Game Pointing Tasks. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)*. Glasgow, Scotland, UK.

[18] Andy Cockburn Mark Claypool and Carl Gutwin. 2020. The Impact of Motion and Delay on Selecting Game Targets with a Mouse. *ACM Transactions on Multimedia, Computing, Communication and Applications (TOMM)* 16, 2 (May 2020).

[19] Kjetil Raaen and Ragnhild Eg. 2015. Instantaneous Human-Computer Interactions: Button Causes and Screen Effects. In *Proceedings of the 17th HCI International*.

[20] Kjetil Raaen Ragnhild Eg and Mark Claypool. 2018. Playing with Delay: With Poor Timing Comes Poor Performance, and Experience Follows Suit. In *Proceedings of the Conference on Quality of Multimedia Experience (QoMEX)*. Sardinia, Italy.

[21] S. S. Sabet, S. Schmidt, C. Griwodz, and S. Möller. 2019. Towards the Impact of Gamers' Adaptation to Delay Variation on Gaming Quality of Experience. In *Proceedings of the International Conf. on Quality of Multimedia Experience (QoMEX)*.

[22] S. S. Sabet, S. Schmidt, S. Zadtootaghaj, C. Griwodz, and S. Möller. 2020. Delay Sensitivity Classification of Cloud Gaming Content. In *Proceedings of the International Workshop on Immersive Mixed and Virtual Environment Systems (MMVE)*.

[23] R. Shea, L. Jiangchuan, E. Ngai, and Y. Cui. 2013. Cloud Gaming: Architecture and Performance. *IEEE Network* 27, 4 (Jul-Aug 2013), 16–21.