User Study-based Models of Game Player Quality of Experience with Frame Display Time Variation

Xiaokun Xu and Mark Claypool
Worcester Polytechnic Institute
Worcester, MA, USA

ABSTRACT

Computer games are often rendered with inconsistent frame timing (frame jitter), particularly in cloud-based game streaming where frames traverse network bottlenecks before being rendered. While previous studies have helped understand the Quality of Experience (QoE) with frame jitter, derived models have tended to be limited in their prediction ability for conditions not yet tested. This paper combines results from four different user studies that assess QoE based on frame jitter, the studies differing in games, game systems, and methods of induced frame time variation. Analysis of the results shows the degree to which frame jitter degrades QoE, and that playout interruption sizes matter while interrupt frequencies do not. The rich user study-based data set provides the basis for models for predicting game player QoE with frame jitter – models which should be predictive for both cloud-based game streaming and traditional games, and for a wide range of player actions and game genres.

ACM Reference Format:

Xiaokun Xu and Mark Claypool. 2024. User Study-based Models of Game Player Quality of Experience with Frame Display Time Variation. In *ACM Multimedia Systems Conference 2024 (MMSys '24), April 15–18, 2024, Bari, Italy*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3625468.3647625

1 INTRODUCTION

Computer games are one of the world's most popular forms of entertainment, with global sales increasing at an annual rate of 10% or more [35]. Computer games are typically dominated by visuals, where a key factor impacting player experience is the frame rate. Generally, a higher frame rate yields smoother, more immersive visuals and is preferred over a lower frame rate. However, a high average frame rate alone does not always ensure good quality since variation in frame rates can degrade the player's experience even while the average stays high [12]. Unfortunately, smooth frame delivery is a challenge faced by many game systems as new games continually push the graphics and processing capabilities of today's computers. This is particularly true for cloud-based game streaming systems, where the server renders the game frames at 60 Hz with up to 4k resolution and streams video to the client over the Internet, adding network variation to any frame playout. While traditional displays may use client-side buffers to smooth video playout, for games, playout buffers need to be small in order to minimize delay and preserve system interactivity, making residual frame jitter likely.

Previous studies have looked at relationships between network conditions and QoE for games and streaming video. Studies of traditional streaming video have equated the impact of delay from buffering with that of an interrupt in playback [2–4, 33], while studies in interactive media have found the types of interaction matter when determining the impact of delay and loss on QoE [1, 10, 19]. Building upon this research with nsights into the effects of frame jitter on QoE can be helpful for optimizing playout buffer sizing where frame jitter can be smoothed out by delay.

This paper presents results from four separate user studies that assess QoE based on frame time variation, the studies differing in games, game systems, and method of inducing frame jitter. We setup test beds that control the variation in frame delivery times and either selected or developed games, giving us control over the game types and frame jitter conditions. Study participants played short rounds of one or more games under different frame jitter conditions, providing a QoE rating each round. While the results from the individual studies is of interest with each providing a detailed analysis of the study results' assessment of frame jitter on QoE, combined the data sets provide a base for more powerful, generalizable models of game player QoE with frame jitter than can be done by any one study. Our models of QoE based on frame time standard deviation, average frame rate, and interrupt magnitude have high predictive power over the range of 11 games tested and are robust in the face of 10x cross fold validation and leave one out cross validation.

A quantitative understanding and model of the effects of frame jitter can be helpful for: (1) *players* to make informed decisions on computer system upgrades and for adjustments to game display settings; (2) *game developers* to implement display-related optimizations where appropriate to provide better experiences for game players; and (3) *computer system developers* to provide frame rate variation targets while improving computer processors and graphics cards and their software.

The rest of this paper is organized as follows: Section 2 introduces research work related to our paper; Section 3 describes our methodology to measure and assess game player QoE with frame jitter; Section 4 analyzes the results and derives models of QoE with frame display variation; Section 6 mentions limitations of our work and suggests possible future work; and Section 7 summarizes our conclusions.

MMSys '24, April 15–18, 2024, Bari, Italy

ACM ISBN 979-8-4007-0412-3/24/04…$15.00
https://doi.org/10.1145/3625468.3647625

2 RELATED WORK

This section describes related work in three main areas: jitter and video streaming (Section 2.1), jitter and cloud-based game streaming (Section 2.2), and frame rates and computer games (Section 2.3). The specific methodologies used vary, but generally all measure user QoE via user questionnaires similar to our study.

2.1 Jitter and Video Streaming

Previous studies have examined the effects of jitter on users passively watching video streams. Packet-level delay jitter can perturb the video delivery and, hence, frame playout times causing frame jitter. Orosz et al. [21] investigate the correlation between subjective QoE assessment via a Mean Opinion Score (MOS), measuring QoS parameters (packet loss and delay jitter) and objective video performance metrics. From a case study they find linear combinations of some QoS metrics such as jitter, packet loss and reordering can correlate well with QoE. Guan-Ming et al. [14] find wireless networks, like 3G and 4G LTE, can be unstable for video streaming. They point out delay jitter due to the network can harm playback smoothness which significantly degrades viewing comfort, thus impacting QoE. Rao et al. [22] introduce a framework for correlating network QoS metrics with streaming video QoE metrics. They conducted a measurement study and find delay jitter can be the main source of QoE degradation.

While these previous works shows relationships between video QoE and network degradation (e.g., delay jitter), they do not apply to games which typically have more interactivity than does typical streaming video and even video conferencing.

2.2 Jitter and Cloud-based Game Streaming

There are some, albeit fewer, studies analyzing the effects of network jitter on QoE for users actively playing cloud-based games. Rossi et al. [23] ran a user study to investigate the subjective QoE of cloud-based game streaming over mobile networks played on smartphones. Their results indicate that game streams are affected differently by network QoS attributes such as packet loss, round-trip time and delay jitter compared to traditional network games and online mobile games. Suznjevic et al. [32] conducted a user study to evaluate the impact of the GeForce Now cloud-based game service adaptation algorithm on player QoE under various network conditions. They found added delay jitter can cause the system to abruptly reduce the amount of data sent and drop the frame rate and frame resolution to their minimal supported values.

While these previous works examine game QoE and network degradation, they do not directly analyze QoE and frame jitter nor produce models of the same.

2.3 Frame Rate and Computer Games

Previous studies have analyzed the effects of system and game configurations on player performance and QoE, generally focusing on frame rate as an independent variable in their analysis. Spjut et al. [29] demonstrate that a 30-millisecond reduction in latency benefits first-person targeting tasks more than frame rates above 60 f/s. Claypool and Claypool [6] show that player actions requiring precise, rapid response, such as shooting, are significantly impacted by frame rates below 30 f/s in first-person shooter games. Claypool et al. [9] find that frame rate affects player performance and game enjoyment, while frame resolution has little impact on performance and some on enjoyment in first-person shooter games. Slivar et al. [27] find that lowering frame rates to 25 f/s does not significantly degrade the gaming experience across different game types, but first-person shooter games are more sensitive to frame rate degradations. Zadtootaghaj et al. [42] investigate the impact of frame rates and bitrates on QoE and find no significant difference in quality and performance ratings between 25 f/s and 60 f/s. Most similar to our work, Liu et al. [12] induce frame jitter directly and conduct a user study with three commercial games. They propose models of QoE-based on different metrics of variation, with frame rate floor the one they recommend. While somewhat broad (a study of 3 games with 7 variation conditions), game genres are considerably broader than those studied as are the type and variety of frame jitter distributions.

Our work extends the above approaches by studying the effects of frame jitter on QoE, with analysis that develops predictive models of player QoE with frame jitter. Compared to Liu et al. [12], our analysis uses an additional 3 data sets and validates the derived models.

3 METHODOLOGY

To approach finding a robust model for predicting game quality with frame jitter that works across many game types and works for both traditional games and cloud-based streamed games, we gathered data from four different user studies and then derived models of player QoE from their data. The user studies are similar in that they are within subjects, using a balanced, playable game in a laboratory environment. However, they differ in their games used, participant sample, and methods of adding frame jitter, which should help a derived model that generalizes beyond the game and system tested.

The overall approach used for each study has the main steps in common:

(1) **Select games.** While games come from a variety of genres, it has been shown that traditional classification of game genres are not sufficient for assessing game QoE [26]. In addition to genre, considerations include visual effects (temporal complexity and spatial complexity [7]), perspectives (camera angle [8]) and degree and type interactivity [24].

Screenshots of games selected for all four studies are shown in Figure 2.

(2) **Setup testbed.** A laboratory setting is used to insulate participants and their game systems from uncontrolled experimental variation. The dedicated lab houses computers powerful enough so as not to induce frame jitter save for the amount controlled.

(3) **Induce frame jitter.** Frame time variation (frame jitter) occurs when the playout time between successive frames is inconsistent. By focusing on the variation in the frame timings that the user experiences (versus, say, network packet delay jitter), analysis and derived models should pertain to both traditional games and cloud-based game streaming.

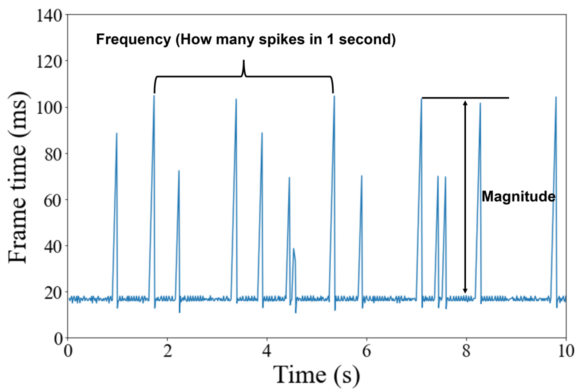


Figure 1: Frame jitter – interrupt frequency and interrupt magnitude.

The individual interrupts in an otherwise smooth playout can be quantified by both magnitude and frequency. Figure 1 depicts interrupts in a frame playout for a game playing at 60 Hz. The x-axis is time and the y-axis is the frame time (the time between the display of consecutive frames). Without any interrupts, the frame display times would all be displayed 16.7 ms apart. However, the "spikes" in the playout are evident, resulting in interrupts that have a magnitude of 70 - 100 ms in this figure and that occur at a frequency of about 1.5x per second.

Causes of frame interrupts include varied processing times and rendering of the images themselves, but also include variance from processing other game-related tasks (e.g., game world updates). Cloud-based game streams have an additional source of frame jitter from the network, particularly significant when there is congestion due to the high bitrate requirements of the stream [41]. For traditional games, local clients handle the processing and rendering, so client hardware, especially for an underpowered client, can also be a contributor to frame jitter.

For all studies, Presentmon [13] – a lightweight tool that captures performance metrics for graphics applications – is used to record the time between frames displayed to the user while the games are running.

(4) **Select parameters.** Pilot studies are used to set fixed values for the games under test (e.g., game length) and to choose the range of values for the variables of interest (e.g., frequency of interrupts in frame playout). All participants for a single study play through each of the parameter values for a within-subjects comparison for that study.

Each user study explores three levels of frame jitter: low, medium, and high. To determine the parameters for each condition, pilot studies are used with various combinations of frame playout interrupt frequency and magnitude to determine perceptibility of each combination – that determines the low point – and maximum tolerance of each combination – that determines the high point. As such parameters differ by game and to a lesser extent the method of inducing frame jitter, we test each game in every user study. Table 1 outlines the specific parameters used.

(5) **Recruit users.** All studies are approved by our University's Institute Review Board (IRB). Volunteers are solicited through university mailing lists and interested game-centric groups, with various incentives to participate (e.g., remuneration or course credit).

(6) **Conduct study.** Participants sign consent forms, provide demographic data, and familiarize themselves with the setup before practicing each game. They then play short rounds for each game condition while the system records frame display timings. The order of games and frame jitter conditions is randomly shuffled for each participant, who completes all rounds for one game before moving on to the next.

After each game round, participants provide a QoE rating via a Mean Opinion Score (MOS) type question – "Please rate your experience" – with a text box for a 1.0 to 5.0 point numeric entry, shown along with a scale: Bad, Poor, Fair, Good, Excellent. MOS testing has been used for decades for traditional interactive voice calls and adapted to Voice over IP (VoIP) in the ITU standard [34].

Longer questionnaires, while offering comprehensive game experience evaluations [5, 11], are impractical for user studies examining multiple parameters within a short timeframe. These questionnaires, with dozens of questions, are better suited for evaluating game design and participation. In contrast, commercial cloud-based game streaming providers like Amazon Luna use a single MOS-type question, which we also employ in our studies.

Participants can pause between rounds for as long as needed before starting the subsequent round and are free to quit the study at any time.

After completing all the game rounds, participants are given an additional questionnaire with demographics questions about overall gamer experience – average time spent playing games and self-rated expertise with computer games.

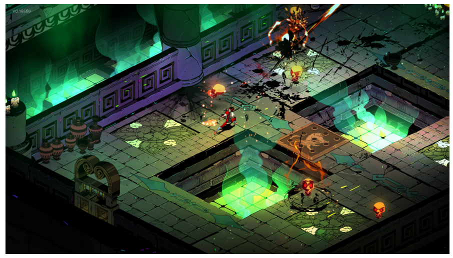
(7) **Analyze data.** The studies each gather a variety of data, but the primary focus in this paper is the relationship between QoE and frame jitter.

Subsections 3.1-3.4 below provide a brief description of the methodology for each of the four studies that provide our combined dataset, providing specifics for items #1-#5 above, with item #6 (analysis) in Section 4.

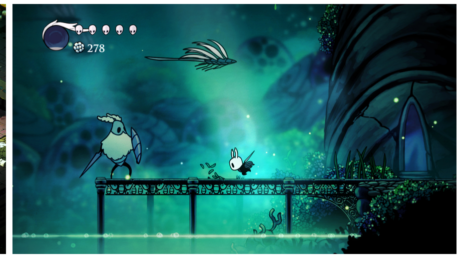
3.1 Study 1 (S1)

In Study 1, we examined four games that had different camera types and spatial information (SI) / temporal information (TI) values. The games included: 1) Bloons Tower Defense 6 (BTD6) [38] (Figure 2d) – This is a tower defense game where players place monkeys to pop balloons before they can travel across a track. The game has a top-down, fixed camera view. 2) Hollow Knight [37] (Figure 2c) – This is a platforming, combat, and exploration game with a 2D camera in a side-scrolling view. 3) Hades [39] (Figure 2b) – This is a rogue-like fighting game where players navigate through randomly ordered rooms. The game has a 2D isometric camera pointed down at the player avatar. 4) Counter-Strike: Global Offensive (CS:GO) [36](Figure 2a) – This is a first-person shooter game with a first-person camera perspective.

(a) CS:GO (S1)

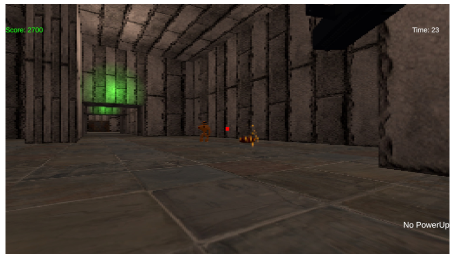


(b) Hades (S1)



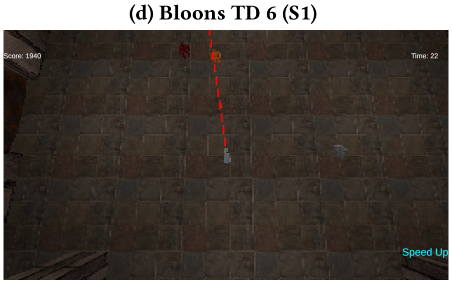
(c) Hollow Knight (S1)

(d) Bloons TD 6 (S1)



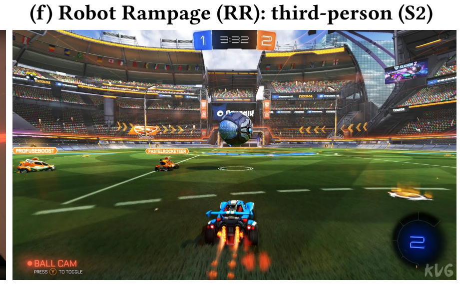
(e) Robot Rampage (RR): first-person (S2)

(f) Robot Rampage (RR): third-person (S2)

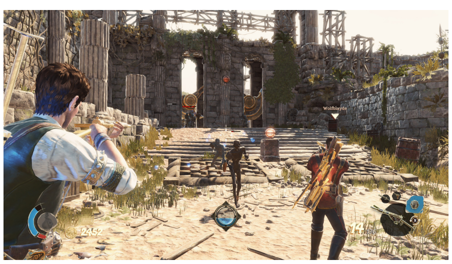


(g) Robot Rampage (RR): overhead (S2)

(h) Valorant (S3)



(i) Rocket League (S3)



(j) Strange Brigade (S3)

(k) SuperTuxKart (S4)

Figure 2: Screenshots of games selected across the four user studies (S1 to S4).

Each game required players to complete a specific task or mission. In BTD6, players placed monkey towers to defend against balloons. In Hollow Knight, players navigated through platforming challenges and encountered enemies in the tutorial. In Hades, players aimed to defeat as many enemies as possible with the default sword, starting a new run upon death. In CS:GO, players completed timed shooting challenges in a training course without human or NPC enemies.

The open-source system Moonlight [15] and Sunshine [18] provided for a cloud-based gaming system setup that: 1) hosted the game server and controlled frame jitter over the network, and 2) had a quality akin to commercial systems – streaming 60 Hz @ 1080p.

The testbed setup is depicted in Figure 3. The cloud-game client was a PC running Windows 10 Pro, connected to the cloud-game server via Moonlight. The PC was an Intel i7 eight-core CPU @ 2.0 GHz with 64 GB RAM with a Gb/s Ethernet NIC and a 1920x1080 LED monitor running at 60 Hz. The cloud-game server PC had the same hardware as the client and streamed the game via Sunshine. The game client connected to the server via a Gb/s switch to a Raspberry Pi 4 configured to act as a network router. The Pi had a 1.5 GHz 64-bit quad-core CPU with 8 GB of RAM and ran Ubuntu 20.04 LTS, Linux kernel version 5.4, using `netem` to control the

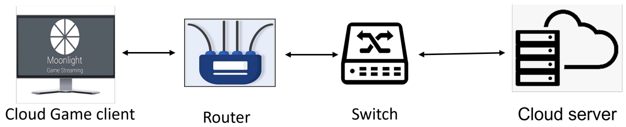


Figure 3: Testbed for Study 1 and Study 2.

Table 1: Target frame jitter conditions for S1, S2 and S4. Study S3 did not have explicit frame jitter targets.

User study	Frame Jitter	Perfect	Low	Medium	High
S1	frequency (/s)	0	1.6	2.5	5
	magnitude (ms)	0	60	120	180
S2	frequency (/s)	0	-	1.9	-
	magnitude (ms)	0	60	-	120
S4	frequency (/s)	0	0.5	0.7	1
	magnitude (ms)	0	30	115	200

network conditions. The router connected to the cloud-game server via a Gb/s switch. The server and client were on the same local area network, so the baseline `ping` round-trip time measurements from our client to server were consistently around 1 ms.

To control the frequency and magnitude of frame playout interrupts for our user study, a Raspberry Pi configured as a router with netem [40] manipulated network traffic. We configured netem with customized distribution files that controlled the frequency and magnitude of packet delays, translating to controllable frequency and magnitude of frame playout interrupts. Table 1 shows the target frequency (interrupts per second) and magnitude (milliseconds) for study S1 for 3 conditions: low, medium and high.

Each round lasted 50 seconds and the entire study took about an hour to complete. Participants received remuneration of $10 for their time and were eligible for class-credit, as appropriate.

3.2 Study 2 (S2)

Study 2 did not use commercial games, but instead we developed a bespoke shooter game called *Robot Rampage* (RR) that allowed the same game to be played with three different camera perspectives: first-person (Figure 2e), third-person (Figure 2f) and overhead (Figure 2g). As in S1, the game also ran at a resolution of 1920x1080 at 60 Hz.

In Robot Rampage, players aim to score points by defeating robots, collecting power-ups, and minimizing damage taken. The game featured endless room and robot generation for continuous play across all rounds.

The study deployed the game in the same cloud-based gaming system setup as Study 1 and on the same testbed. Frame jitter was induced the same way, too – via a network middlebox – although with different frequency and magnitude targets (Section S2 in Table1).

Each round lasted 30 seconds and the entire study took about 40 minutes to complete. Participants received remuneration of $10 for their time and were eligible for class-credit, as appropriate.

3.3 Study 3 (S3)

Study 3 was conducted previously by a different research team [12], having gathered data on QoE with frame jitter for three games.

Rocket League and Strange Brigade offered players a third-person viewpoint, allowing them to see the avatar they controlled, whereas Valorant provided a first-person perspective. Rocket League took place in a spacious virtual arena, while Valorant was set in a smaller indoor room, and Strange Brigade took place in a large outdoor area. In all three games, players controlled their avatars using continuous input, such as holding down keys to move. However, Strange Brigade and Valorant also incorporated discrete input for shooting.

Each player did the same task/mission in each game. In Rocket League, players controlled cars to score goals against computer-controlled opponents in a 1v1 user study setting. In Valorant, players engaged in combat using various projectile weapons in a tutorial setting where they planted a "spike" and defended it against computer-controlled opponents. In Strange Brigade, players navigated a fictional world, combating enemies with various weapons. In the user study, gameplay focused on a single area where players continuously fought waves of zombies.

Participants played on a gaming PC with an NVIDIA GeForce RTX 2080 graphics card, 11th Gen Intel Core I9-11900k @ 3.50 GHZ CPU, Samsung SSD 70 EVO Plus 2 TB disk driver, 32 GB RAM that ran Microsoft Windows 10 Pro. The PC had a gaming mouse: a Logitech G502, 12k DPI with a 1000 Hz polling rate; and a high refresh rate monitor: a 25" Lenovo Legion, 1920x1080 16:9 pixels @ 240 Hz with AMD FreeSync (Gsync compatible) and a 1 ms response time.

Each game was tested with two target frame rates, 60 Hz and 120 Hz, capped using Rivatuner Statistics Server (RTSS) [20] – a multi-function tool that supports frame rate limiting.

Frame rate variation was added using extra load for the CPU via an infinite Fibonacci number counter written in Python. The counter ran as a separate process infinitely computing sequences of numbers (i.e., it is CPU bound) and so competed for use of the CPU with the game. By controlling the number of counters running simultaneously with the game, different amounts of frame rate variation were observed in the display, although unlike in studies S1, S2 and S4, there was not an explicit target for frequency and magnitude of interrupts. The frame rate variation had 4 conditions for each target frame rate (60 Hz and 120 Hz) based on the number of counters: a "perfect" condition without any counters, a "low" condition where the counters caused frame jitter that was just noticeable to players, a "high" condition where the counters cause severe frame variation but the game was still barely playable and one "mid" point between "low" and "high".

Users were invited to participate in the study based on their familiarity with the three games studied and overall gaming experience, favoring participants that played games regularly and had played the three games over those that did not.

Each game round lasted for 1 minute and it took each user about 60 minutes to complete all the tasks in the study. Participants received remuneration of $15 for their time and were eligible for class-credit, as appropriate.

3.4 Study 4 (S4)

Study 4 used an open source game SuperTuxKart [30] (Figure 2k) that mimicked a popular commercial 3D racing game (MarioKart). In SuperTuxKart, players raced against AI-controlled opponents on fictional race tracks filled with obstacles and player collisions. In the user study, the gameplay was limited to a single map, Zen Garden, which could be completed in approximately one minute.

Participants played on a gaming PC with an 11th Gen Intel I9 processor with 32 GB of RAM and an NVIDIA GeForce RTX 2080 graphics card, running Microsoft Windows 10 Pro. The peripherals were a Logitech G502 gaming mouse (12k DPI and 1000 Hz polling), a 240 Hz Lenovo Legion, monitor with 1920x1080 16:9 pixels and AMD FreeSync.

For frame jitter, the frequency and magnitude of frame display interrupts were controlled by modifying the game engine (hence, our choice of an open-source game). We added a sleep time in the game loop that caused frame playout interrupts. The magnitude was controlled by the duration of the sleep time and the frequency was controlled by how often the sleep time is added (e.g., sleeping once every 250 ms provides a frequency of about 4 interrupts per second, each about 250 ms long). We combined the magnitude and frequency to get different settings of "low", "medium" and "high" frame jitter (Section S4 in Table 1), tuned based on pilot studies. The SuperTuxKart game engine ran at 60 Hz.

3.5 Summary of Studies

Table 2 summarizes the 4 user studies, highlighting key differences. The common methods used to gather data (e.g., the same QoE questions) coupled with the differences (e.g., different games and frame display variation, i.e., frame jitter, methods) provide for a broad dataset from which to derive a robust model for predicting player QoE. Game session durations were determined from pilot studies, considering the genre of the game. Fast-paced games favored shorter sessions for focused feedback, while narrative or strategy games required longer sessions. Levels or rooms were chosen to ensure consistentcy in gameplay across participants. Rounds lasting 30-60 seconds allowed for enough time for players to evaluate the interactivity and visual characteristics for the games.

Table 3 summarizes the information for each game in the 4 user studies. We measured spatial information (SI) and temporal information (TI) [17] for a representative 30-second video recording of each game as played on the PC used in their respective studies. SI is calculated by computing the brightness of each pixel in the filtered image of a given frame. The variation in brightness across all pixels determines the SI value for one frame, with the maximum SI computed for all frames being the game's SI value. Games with high contrast visuals tend to have high SI values. TI is computed by comparing the change in brightness for each pixel in a given frame from its brightness in the previous frame. As with SI, the variation in change in brightness for all pixels in a given frame is the frame's TI, with the maximum TI computed for all frames being the game's TI value. Games with sudden visual changes tend to have high TI values. Figure 4 depicts the SI-TI data from Table 3 in a scatter plot. From the graph, the selected games cover the range shown by typical commercial games [3].

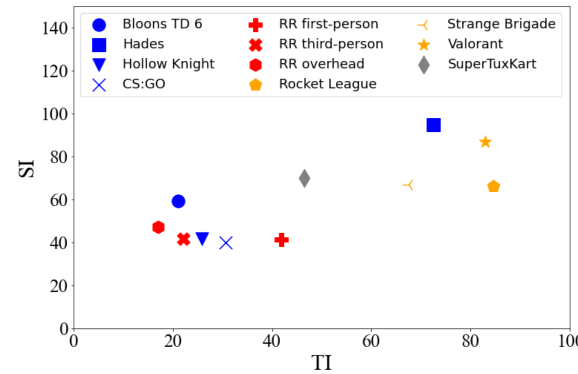


Figure 4: Spatial information (SI) and temporal information (TI) for games across all 4 user studies (S1 to S4). Colors differentiate camera type and shapes differentiate games.

4 ANALYSIS

Section 4.1 provides summary demographics for the user study participants; Section 4.3 analyzes the QoE for several predictive measures of QoE – 95% frame rate floor, frame rate average, playout interrupts, and frame time standard deviation; Section 4.4 cross-validates and compares the models to each other; and Section 4.5 provides a model summary.

4.1 Demographics

Table 4 summarizes the demographic information of participants for the 4 user studies. Gamer self-rating is in response to the question "rate your experience as a gamer" on a five-point scale, 1-low to 5-high. For age and gamer self-rating, the mean values are given with standard deviations in parentheses. Since participants are recruited from on-campus students for all user studies, the age, and gamer demographics are similar. The participants ranged from 18-26 years old, encompassing a typical college age range (18-22). The user self-rating of experience playing computer games trended above the mid-point for S1, S2 and S3, and lower for S4. Most of the participants majored in Computer Science or Game Development.

Participants indicated their familiarity with the games under test in pre-screening questionnaires. All were experienced with gaming in general, and most with related games in the same genres as the games being tested.

Across most games, QoE scores given by the participants for base conditions (games played without induced frame jitter) clustered between 4 and 5, with mean values between 4.2 and 4.3. The only exception was the Robot Rampage game played with the overhead view (RR overhead) which had QoE scores lower by about a point, likely due to the unclear visuals and the challenging gameplay from the top perspective. Overall, this suggests that QoE models derived from our datasets do not need to be normalized to an individual game baseline, with the possible exception of uniquely bad games, like RR overhead.

4.2 QoE versus Frame Display Variation

We analyze QoE versus 5 different measures of frame jitter:

Table 2: Summary of 4 user studies (S1 to S4).

User Study	System	Games	Frame Jitter Method	Jitter Settings	Round Time	Rounds
S1	cloud-streaming	4	network middlebox	10	50 seconds	10 for each game
S2	cloud-streaming	1 w/3 modes	network middlebox	3	30 seconds	15 for each mode
S3	traditional	3	CPU load	7	1 minute	7 for each game
S4	traditional	1	game engine	10	≈ 1 minute	2 for each setting

Table 3: Summary of game information.

User study	Game	Genre	Perspective	SI	TI
S1	Bloons TD 6	Tower defense	Overhead	59.3	21.2
	Hades	Action	2D top	94.9	72.5
	CS:GO	Shooter	First-person	40.1	30.7
	Hollow Knight	Action	2D side	41.6	25.8
S2	RR first-person	Shooter	First-person	41.3	41.8
	RR third-person	Shooter	Third-person	41.7	22.2
	RR overhead	Shooter	Overhead	47.1	17.1
S3	Rocket League	Sports	Third-person	66.3	84.5
	Strange Brigade	Shooter	Third-person	66.7	67.6
	Valorant	Shooter	First-person	86.7	83.0
S4	SuperTuxKart	Racing	Third-person	69.8	46.5

Table 4: Participant demographics.

Study	Users	Age (yrs)	Gamer Self-rating
S1	28	20 (2.0)	4.2 (0.8)
S2	35	20.8 (1.8)	None
S3	31	20.1 (2.0)	3.6 (1.2)
S4	38	20.2 (2.3)	2.4 (1.1)

AFR Average frame rate. Frame rate is a commonly used indicator of video and game-display performance. The average frame rate is calculated from the total number of frames displayed divided by the round duration (units are frames / second).

FTSD Frame time standard deviation. Since standard deviation is a commonly used measure of variation for any kind of numerical data, the frame time standard deviation as a measure of frame jitter is computed by taking the standard deviation of the frame times across all frames displayed in a round.

FRF Frame rate floor. Earlier work [12] (using study S4) recommended the 95% frame rate floor for predicting QoE with frame jitter since that metric fit the data from their single user study. The 95% frame rate floor is computed by taking the distribution of frame times, selecting the top 5th-percentile value and converting that to a frame rate.

IM, IF Frame playout interrupts. Another previously used indicator of visual quality, particularly used for streaming video, are interrupts to the playout stream [2] – gaps larger than the frame time (e.g., 60 Hz has a frame time of 16.7 ms, so a frame time larger than 33.3 ms is a gap) – which can be measured by both their frequency and magnitude. Interrupt frequency (IF) is the total number of frame gaps divided by the round time (units are interrupts per second). Interrupt magnitude is the total of all interrupt times (in milliseconds) greater than the frame time divided by the round time (units are milliseconds per second).

Figure 5 depicts the results, one graph for each measure of frame jitter. For each graph, the y-axis is the QoE with a x-axis specific to each measure. The data in each graph are grouped by frame jitter condition for each study (based on the "Jitter Setting" column in Table 2) – S1 has 10 groupings, S2 has 3, S3 has 7 and S4 has 10. The QoE values for all users in each group are averaged and shown with a 95% confidence interval. Colors and shapes differentiate the different studies: blue triangles are from S1, red circles are from S2, orange crosses are from S3, and grey stars are from S4. The green line in each graph is the linear regression trendline through the mean values of all four user studies, with the adjusted R^2 shown in the legend. The adjusted R^2 for each user study's data individually is also shown in the legend, although those trendlines are not shown in the graphs to keep them readable.

From Figure 5a, the average frame rate correlates pretty well with QoE across all games, R^2 0.92. This confirms that average frame rate has merit as an indicator of QoE even when there is frame jitter i.e., the average is preditive overall even though it "hides" variation in frame delivery times. However, given that underlying frame jitter is not accounted for in an average and such variation (e.g., a temporal gap in playout, illustrated in Figure 1) can degrade quality, we look to metrics that are still accurate in predicting QoE but also incorporate an explicit measure of the underlying variation.

From Figure 5b, the 95% frame rate floor model does not fit all the data well, at least with a linear relationship. While a 95% frame rate floor model had been espoused as a good model for predicting QoE since it fit the data from S3 by itself quite well (R^2 of about 0.96), when modeling data from all four user studies, the R^2 is only about 0.61. The weakness of the frame rate floor model for all data may be due to the wider range of frame variation and game types when used for all studies. Either way, it motivated our exploration to find a more accurate model that is robust across all games and conditions studied.

From Figure 5c, interrupt frequency has little visual correlation with average QoE over the range of interrupts tested, R^2 0.18. This may be because the range of frequencies tested by themselves do not cover a range that is differentiable by the users. In contrast, from Figure 5d, interrupt magnitude correlates with the mean values well, R^2 0.93. This indicates that the size of the interrupt matters to the QoE more than how often an interrupt happens.

From Figure 5e, frame time standard deviation – perhaps the most straightforward measure of frame display variation – also correlates with the mean values well, R^2 0.92.

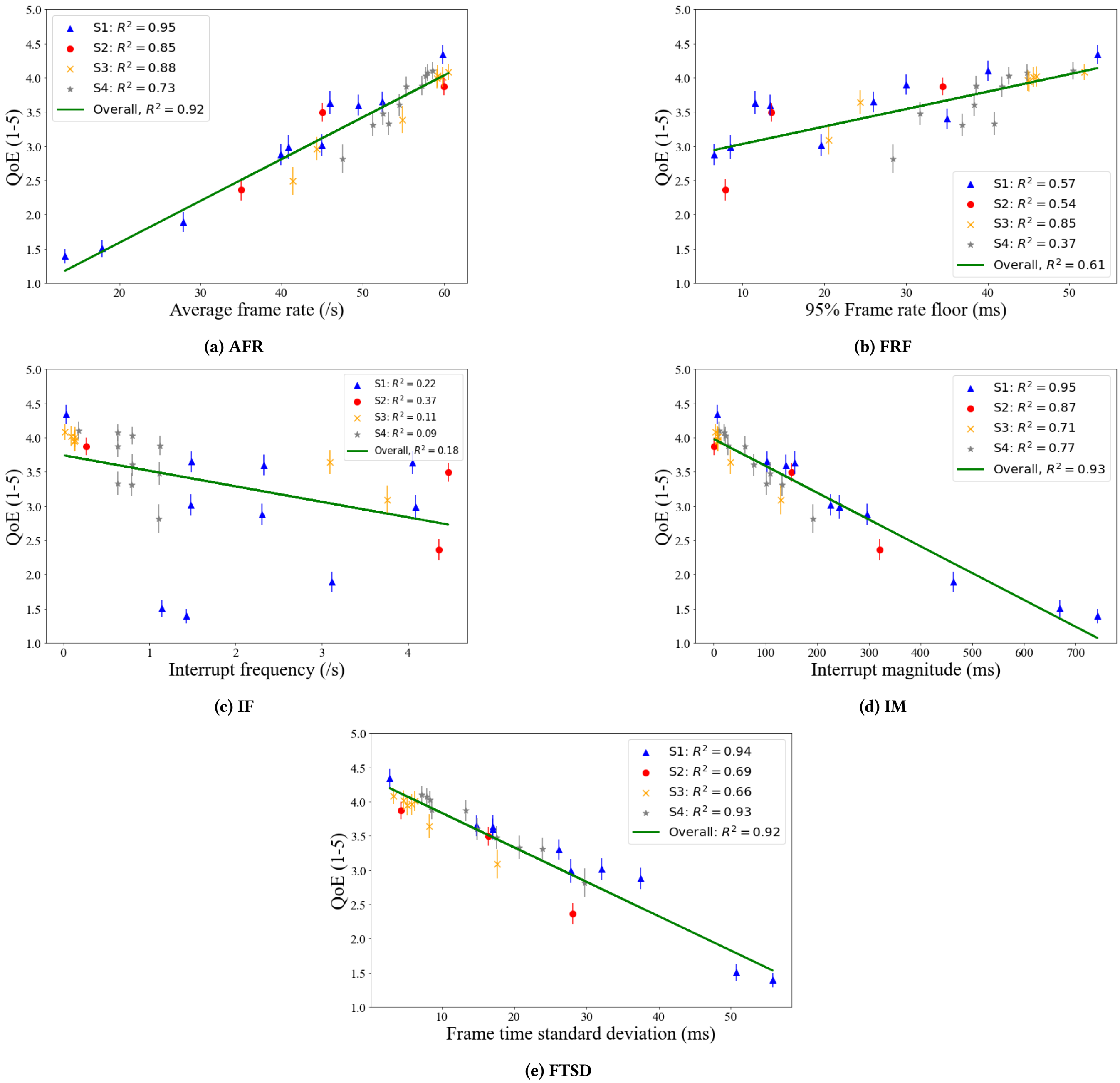


Figure 5: QoE models: Average frame rate (AFR), 95% frame rate floor (FRF), interrupt frequency (IF), interrupt magnitude (IM), and frame time standard deviation (FTSD).

4.3 Models

We also analyzed how each measure of frame jitter model fits the individual games in each study. Table 5 has the results. Most games approximately match the R^2 of the user study they belong to (see the legend of Figure 5e for the overall R^2 for each study), with the notable exception of the overhead mode for Robot Rampage (RR overhead) in S2 (R^2 0.19). However, RR overhead had low average QoE ratings for all rounds, even the "perfect" rounds with no frame display variation, perhaps because the game looked awkward from a top-down view and was more difficult to play, as well.

When viewed across all games, AFR, IM and FTSD all have similar overall fits (adjusted R^2 0.90, 0.91 and 0.92, respectively). However, AFR has a weak fit for several games – CS:GO 0.37, Hollow Knight 0.19 and Strange Bridgade 0.26 – whereas IM and FTSD have fewer

Table 5: Model fits per individual game (adjusted R^2): average frame rate (AFR), 95% frame rate floor (FRF), interrupt frequency (IF), interrupt magnitude (IM) and frame time standard deviation (FTSD).

User study	Game	AFR	FRF	IF	IM	FTSD
S1	Bloons TD 6	0.82	0.33	0.13	0.86	0.83
	Hades	0.96	0.72	0.25	0.96	0.92
	CS:GO	0.37	0.60	0.37	0.95	0.92
	Hollow Knight	0.19	0.58	0.19	0.90	0.95
S2	RR first-person	0.90	0.59	0.54	0.89	0.79
	RR third-person	0.85	0.54	0.37	0.87	0.74
	RR overhead	0.34	0.18	0.96	0.23	0.19
S3	Rocket League	0.74	0.72	0.60	0.52	0.63
	Strange Brigade	0.26	0.80	0.26	0.72	0.75
	Valorant	0.68	0.41	0.87	0.59	0.75
S4	SuperTuxKart	0.73	0.37	0.09	0.77	0.93
	Overall	0.90	0.63	0.16	0.91	0.92

weak fits for individual games. However, IM is slightly weak for Rocket League 0.52 and Valorant 0.59, while FTSD is only weak for Rocket League 0.63.

4.4 Cross-Validation and Comparison of Models

Model over-fitting is when a model performs well on the observed data but fails to generalize to new, unseen data (e.g., for our frame jitter models this can be to users, games and jitter conditions not yet tested). Overfitting can occur when the model is overly complex relative to the size of the dataset.

Table 6 summarizes the adjusted R^2 for the models explored, ordered in increasing order of model complexity. For the equations, the k parameters (e.g., k_1) are constants, e is the exponential function and s is the frame time standard deviation. The first model, k_1, is just the average QoE value which provides the simplest of predictions. The third model uses an exponential (Equation #3) but does not fit as well as a linear model (Equation #2) and using a 2nd-order term (Equation #4) does not improve the adjusted R^2 over linear. Using both the first- and second-order terms (Equation #5) does bring up the adjusted R^2 a little bit higher than that of a linear model (Equation #2). Comparing across models, IF and FRF do not fit well for any equations whereas AFR, IM and FTSD do well in both Equation #2 and Equation #5. Since the differences in adjusted R^2 between Equation #2 and Equation #5 is minimal in all cases, the linear model (Equation #2) is recommended and used for the rest of this analysis.

Cross-validation is used to check if the proposed models overfit the data. With 10-fold cross validation, the data is first split into 10 subsets (folds). Then, the first fold is used as a test, and a regression model is built from data in the other 9 folds and evaluated against the first test fold. This process is repeated for each fold, resulting in 10 evaluations, with the adjusted R^2 averaged across all folds. Since our sample pool is only 30 points (one for each jitter condition), to avoid imbalanced sampling, we used stratified 10-fold cross-validation where the folds are striated across the QoE scores. Leave-one-out cross validation (1-out) is a special case of cross fold validation where only one data point is held out, the regression model built from the remaining data, and the mean squared error (MSE) computed for the held-out point. The validation results are summarized in the next section (Section 4.5).

Table 6: Comparison of models predicting QoE (average adjusted R^2 over 10-fold cross validation) – e is the exponential function, and k's are constants (different for each model) fit to data gathered through the user studies.

	model	AFR	FRF	IF	IM	FTSD
1	k_1	0	0	0	0	0
2	$\boldsymbol{k_1 + k_2 s}$	**0.921**	0.612	0.185	**0.930**	**0.915**
3	$k_1 + k_2 e^s$	0.219	0.126	0.059	0.324	0.300
4	$k_1 + k_2 s^2$	0.912	0.617	0.097	0.747	0.875
5	$k_1 + k_2 s + k_3 s^2$	0.925	0.621	0.353	0.949	0.922

4.5 Summary

The linear regression models (Equation #2) explored are summarized in Table 7, ordered low to high by adjusted R^2 and also shown with the mean square error (MSE). From the table, the AFR, FTSD, and IM models all fit the mean QoE data well, with an R^2 of 0.92, 0.92 and 0.93, respectively. Conversely, the IF model does not, with an R^2 of 0.18, and the FRF model is in-between with an R^2 of 0.61. The cross-validation results (columns "10-fold" and "leave-one-out") suggest the IM and AFR models are over-fitting the data slightly more so than the FTSD model.

Table 7: Linear prediction models: $QoE = slope \cdot x + intercept$.

				10-fold		1-out
Predictor	R^2	Slope	Intercept	R^2	MSE	MSE
Interrupt freq. (/s)	0.18	-0.023	3.7	0.18	0.57	0.52
95% Frame rate floor (f/s)	0.61	+0.026	2.8	0.53	0.13	0.11
Average frame rate (f/s)	0.92	+0.061	0.4	0.78	0.06	0.05
Frame time std. dev. (ms)	0.92	-0.056	4.6	0.84	0.05	0.05
Interrupt mag. (ms/s)	0.93	-0.004	4.0	0.81	0.06	0.05

5 DISCUSSION

The experiments focused on how frame jitter affects players' QoE in games. It was expected that the visual aspects of individual games would influence this effect, but all 11 tested games showed similar trends with frame jitter, mostly independent of visual content such as graphics fidelity or camera perspective. Additionally, while it was assumed that visual QoE would worsen with more frequent playout gaps, little degradation was observed within the tested range. Instead, the size of the gaps emerged as a key factor affecting QoE, suggesting that humans are more sensitive to large gaps in frem playout than smaller, if more frequent, gaps in playout.

The expectation was that average frame rate would only loosely reflect QoE based on previous work [12], presumably since the average hides the underlying variance. Yet somewhat surprisingly, the average frame rate correlated well with player QoE overall. Conversely, we expected that the 95% frame rate floor (FRF) would be predictive of QoE based on those earlier results – yet when analyzed across all games the FRF was not effective. These two results emphasize the importance of studies that reproduce other's work. Our results do confirm previous results that frame time standard deviation is an effective predictor of QoE.

Previous cloud-based QoE studies often focus on the network conditions for individual game systems, hindering comparison with predictive models for traditional games. However, our findings cross cloud-based systems and traditional game systems and suggest frame variation's impact on QoE is mostly independent of the underlying system. This simplifies predictive modeling, allowing focus on frame jitter irrespective of game system, eliminating the need for separate models for cloud-based and traditional games.

Frame jitter in cloud-based game streaming is typically smoothed out by a client-side playout buffer, a highly effective technique used in traditional video streaming [31]. However, sizing playout buffers for game streaming is tricky since the buffers must be small to avoid degrading interactivity and the player experience [24]. This sets up an interesting trade-off between delay and frame jitter worthy of exploring, where buffer delay smooths out the frame playout but decreases the interactivity.

Our models can provide guidance for players, game developers, and computer system developers: (1) players could use the models for their own game systems to pick graphics settings that reduce frame jitter, or even decide whether or not to do a computer upgrade; (2) game developers could use the models for their target systems to decide if additional optimizations are needed to smooth out the frame playout; (3) computer system developers could use the models with reference games to decide if and how much the hardware and software improvements they are developing improve visual QoE. To be used in practice, a sample of the gameplay would be recorded and the frame times measured (we use Presentmon [13], but other tools should be usable including hooks directly in a game engine), metrics computed (average frame rate, frame time standard deviation, or interrupt magnitude) and plugged into the models.

6 LIMITATIONS AND FUTURE WORK

From the graphs, the range of frame jitter for some user studies is small with data concentrated in a narrow area (i.e., little spread of the data points along the x-axis). This is caused by the parameters used to produce the frame jitter where a broader range of settings may increase the spread. More specifically, the range of perturbations in frame jitter in our study were notably controlled for studies S1, S2 and S4 (fixed magnitude, fixed frequency), whereas real-world frame jitter is likely more varied. Future work could gather frame jitter values "in the wild" as traces and replay the frame jitter values observed in a testbed experiment.

The lack of familiarity of some of our users with the games tested could affect the QoE ratings. Previous work [28] shows that players with more skill in a game are more sensitive to network perturbations, and this could hold for game familiarity and frame jitter. Future studies could control for user familiarity with the games under test.

The studies all used a single QoE question for the assessment. Future research could investigate the effects of frame jitter with a more comprehensive assessment of QoE, perhaps using tools such as the Gaming Input Quality Scale (GIPS) questionnaire [25]. While the study format used here with questions after a short game round would need to change, the multiple-question approach could offer additional information useful for improving QoE in game systems with frame jitter.

The QoE models presented here focus on frame jitter, but consider interactivity necessary for users to play the games. However, missing is any additional system latency, say, that may arise from networking delays. Other factors like visual effects [8] and control-display gain [16] could also influence QoE. Future work could study frame jitter combined with other factors where there may be confounding effects on QoE.

7 CONCLUSIONS

Computer games continue to push the boundaries of display technologies with better graphics and higher frame rates. This often results in frames displayed with inconsistent timing (frame jitter). Cloud-based game streaming, in particular, can have frame jitter as rendered frames must also traverse the network before being rendered. While it is well-known that higher, smoother frame rates for games provide a better QoE than lower frame rates, how frame jitter display is perceived by users is not well-known. Predicting QoE with frame jitter can help gamers, game developers and game system developers with understanding and improving the experience for players and also practitioners of interactive media.

Our paper presents results from four different user studies that assess game player QoE with frame jitter, testing 11 games with a variety of visual and interactive characteristics, using 3 different techniques to control frame jitter across 30 different settings with both cloud-based and traditional game systems. Over 130 participants played short rounds with different frame jitter characteristics, providing QoE opinions each round via a survey.

Analysis of the results shows the frame time standard deviation and interrupt magnitude are good predictors game player QoE (R^2 of 0.92 and 0.91, respectively), with linear models being the most parsimonious. The models fit most individual games well, too, despite the differences in the visuals across the games tested. While average frame rate is also a good overall predictor of game player QoE, it fails to accurately predict QoE for some games. Average frame rate and interrupt magnitude may also slightly overfit the data. Interrupt frequency is not a good predictor of QoE and, contrary to previous recommendations [12], a model based on frame rate floor is not effective at predicting QoE for all games and game conditions.

ACKNOWLEDGEMENTS

Thanks to Thomas Flanagan, Carter Nakagawa, Michael Oliveira for conducting study S1, Ryan Darcy, Sean O'Connor, and Wenjie Zhang for conducting study S2, Shengmei Liu for conducting studies S3 and S4, and Atsuo Kuwahara and James Scovell for collaboration on the approach.

REFERENCES

[1] Hamed Ahmadi, Sepideh Khoshnood, Mahmoud Reza Hashemi, and Shervin Shirmohammadi. 2013. Efficient Bitrate Reduction using a Game Attention Model in Cloud Gaming. In *Proceedings of the IEEE International Symposium on Haptic Audio Visual Environments and Games (HAVE)*. Ottowa, ON, Canada, 103–108.

[2] Josh Allard, Andrew Roskuski, and Mark Claypool. 2020. Measuring and Modeling the Impact of Buffering and Interrupts on Streaming Video Quality of Experience. In *Proceedings of the 18th International Conference on Advances in Mobile Computing & Multimedia (MoMM)*. Chiang Mai, Thailand.

[3] Nabajeet Barman, Saman Zadtootaghaj, Steven Schmidt, Maria G. Martini, and Sebastian Möller. 2018. GamingVideoSET: A Dataset for Gaming Video Streaming Applications. In *Workshop on Network and Systems Support for Games (NetGames)*. Amsterdam, The Netherlands, 1–6. https://doi.org/10.1109/NetGames.2018.8463362

[4] Kashif Bilal and Aiman Erbad. 2017. Impact of Multiple Video Representations in Live Streaming: A Cost, Bandwidth, and QoE Analysis. In *IEEE International Conference on Cloud Engineering (IC2E)*. Vancouver, Canada, 88–94. https://doi.org/10.1109/IC2E.2017.20

[5] J.H. Brockmyer, C.M. Fox, K.A. Curtiss, E. McBroom, K.M. Burkhart, and J.N. Pidruzny. 2009. The Development of the Game Engagement Questionaire: A Measure of Engagement in Video Game Playing. *Journal of Experimental Social Psychology* (2009), 624–634.

[6] Kajal Claypool and Mark Claypool. 2007. On Frame Rate and Player Performance in First Person Shooter Games. *Springer Multimedia Systems Journal (MMSJ)* 13 (2007), 3–17.

[7] Mark Claypool. 2009. Motion and Scene Complexity for Streaming Video Games. In *Proceedings of the 4th ACM International Conference on the Foundations of Digital Games (FDG)*. Florida, USA.

[8] Mark Claypool and Kajal Claypool. 2009. Perspectives, Frame Rates and Resolutions: It's all in the Game. In *Proceedings of the 4th ACM International Conference on the Foundations of Digital Games (FDG)*. Florida, USA.

[9] Mark Claypool, Kajal Claypool, and Feissal Damaa. 2006. The Effects of Frame Rate and Resolution on Users Playing First Person Shooter Games. In *Proceedings ACM/SPIE Multimedia Computing and Networking (MMCN) Conference*. San Jose, CA, USA.

[10] Mark Claypool, David Finkel, Alexander Grant, and Michael Solano. 2012. Thin to Win? Network Performance Analysis of the OnLive Thin Client Game System . In *Proceedings of the 11th ACM Network and System Support for Games (NetGames)*. Venice, Italy.

[11] A.L. Cox, C. Jennett, C. Paul, S. Dhoparee, A. Epps, T. Tijs, and A. Walton. 2008. Measuring and Defining the Experience of Immersion in Games. *International Journal of Human-Computer Studies* 66, 9 (2008).

[12] (Anonymous for double-blind review). [n. d.].

[13] GameTechDev. 2022. PresentMon V1.8.0. Online: https://github.com/GameTechDev/PresentMon.

[14] Su Guan-Ming, Su Xiao, Bai Yan, Wang Mea, Athanasios V. Vasilakos, and Haohong Wang. 2015. QoE in Video Streaming over Wireless Networks: Perspectives and Research Challenges. *Wireless Networking* (2015). https://doi.org/10.1007/s11276-015-1028-7

[15] Cameron Gutman. 2022. Moonlight PC v4.3.1. Online: https://github.com/moonlight-stream.

[16] Ravin Balakrishnan Géry Casiez, Daniel Vogel and Andy Cockburn. 2008. The Impact of Control-Display Gain on User Performance in Pointing Tasks. *Human–Computer Interaction* 23, 3 (2008), 215–250. https://doi.org/10.1080/07370020802278163

[17] International Telecommunication Union. 2008. ""ITU-T Recommendation P.910: Subjective Video Quality Assessment Methods for Multimedia Applications. https://www.itu.int/rec/T-REC-P.910-200804-S/en.

[18] LizardByte. 2022. Sunshine. Online: https://github.com/LizardByte/Sunshine.

[19] M. Manzano, J. A. Hernández, M. Urueña, and E. Calle. 2012. An Empirical Study of Cloud Gaming. In *Proceeding of the 11th Annual Workshop on Network and Systems Support for Games (NetGames)*. 1–2.

[20] Alexey Nicolaychuk. 2013. RivaTuner Statistics Server . Online: https://www.guru3d.com/content-page/rivatuner.html.

[21] Péter Orosz, Tamás Skopkó, Zoltán Nagy, Pál Varga, and László Gyimóthi. 2014. A Case Study on Correlating Video QoS and QoE. In *IEEE Network Operations and Management Symposium (NOMS)*. Krakow, Poland, 1–5. https://doi.org/10.1109/NOMS.2014.6838399

[22] N. Rao, A. Maleki, F. Chen, W. Chen, C. Zhang, N. Kaur, and A. Haque. 2019. Analysis of the Effect of QoS on Video Conferencing QoE. In *15th International Wireless Communications & Mobile Computing Conference (IWCMC)*. 1267–1272. https://doi.org/10.1109/IWCMC.2019.8766591

[23] Henrique Souza Rossi, Niclas Ögren, Karan Mitra, Irina Cotanis, Christer Åhlund, and Per Johansson. 2022. Subjective Quality of Experience Assessment in Mobile Cloud Games. In *GLOBECOM - IEEE Global Communications Conference*. Rio de Janeiro, Brazil, 1918–1923. https://doi.org/10.1109/GLOBECOM48099.2022.10001407

[24] Saeed Shafiee Sabet, Steven Schmidt, Saman Zadtootaghaj, Carsten Griwodz, and Sebastian Möller. 2020. Delay Sensitivity Classification of Cloud Gaming Content. In *Proceedings of the International Workshop on Immersive Mixed and Virtual Environment Systems (MMVE)*. Istanbul, Turkey.

[25] S. Schmidt. 2022. *Assessing the Quality of Experience of Cloud Gaming Services*. Springer International Publishing, Switzerland. https://books.google.com/books?id=LRaIEAAAQBAJ

[26] S. Schmidt, S. Zadtootaghaj, and S. Moller. 2017. Towards the Delay Sensitivity of Games: There is More than Genres. In *Proceedings of the International Conference on Quality of Multimedia Experience (QoMEX)*. Erfurt, Germany.

[27] Ivan Slivar, Mirko Suznjevic, and Lea Skorin-Kapov. 2015. The Impact of Video Encoding Parameters and Game Type on QoE for Cloud Gaming: A Case Study Using the Steam Platform. In *Seventh International Workshop on Quality of Multimedia Experience (QoMEX)*. Costa Navarino, Messinia, Greece.

[28] Ivan Slivar, Mirko Suznjevic, Lea Skorin-Kapov, and Maja Matijasevic. 2014. Empirical QoE Study of In-home Streaming of Online Games. In *13th Annual Workshop on Network and Systems Support for Games*. Nagoya, Japan, 1–6. https://doi.org/10.1109/NetGames.2014.7010133

[29] Josef Spjut, Ben Boudaoud, Kamran Binaee, Jonghyun Kim, Alexander Majercik, Morgan McGuire, David Luebke, and Joohwan Kim. 2019. Latency of 30 ms Benefits First Person Targeting Tasks More Than Refresh Rate Above 60 Hz. In *SIGGRAPH Asia Technical Briefs*. ACM, New York, NY, USA.

[30] Steve and Oliver Baker. 2022. SuperTuxKart. https://supertuxkart.net/ (Accessed Sept 10, 2023).

[31] Ya-Fan Su, Yi-Hsuan Yang, Meng-Ting Lu, and Homer H. Chen. 2009. Smooth Control of Adaptive Media Playout for Video Streaming. *IEEE Transactions on Multimedia* 11, 7 (2009), 1331–1339. https://doi.org/10.1109/TMM.2009.2030543

[32] Mirko Suznjevic, Ivan Slivar, and Lea Skorin-Kapov. 2016. Analysis and QoE Evaluation of Cloud Gaming Service Adaptation under Different Network Conditions: The Case of NVIDIA GeForce NOW. In *Eighth International Conference on Quality of Multimedia Experience (QoMEX)*. Lisbon, Portugal, 1–6. https://doi.org/10.1109/QoMEX.2016.7498968

[33] Ayano Tatematsu, Yutaka Ishibashi, Norishige Fukushima, and Shinji Sugawara. 2010. QoE Assessment in Haptic Media, Sound and Video Transmission: Influences of Network Latency. In *IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR 2010)*. Vancouver, British Columbia, Canada, 1–6. https://doi.org/10.1109/CQR.2010.5619913

[34] International Telecommunication Union. 2001. Perceptual Evaluation of Speech Quality (PESQ) - An Objective Method for End-to-end Speech Quality Assessment of Narrow-band Telephone Networks and Speech Codecs. ITU Recommendation ITU-T P.862.

[35] wepc.com. 2022. Video Game Industry Statistics, Trends and Data In 2022. Online: https://www.wepc.com/news/video-game-statistics/. (Accessed October 26, 2022).

[36] Wikipedia contributors. [n. d.]. Counter-strike: Global Offensive (CS:GO). https://en.wikipedia.org/wiki/Counter-Strike:_Global_Offensive [Accessed 28-Nov-2023].

[37] Wikipedia contributors. 2017. Hollow Knight. https://en.wikipedia.org/wiki/Hollow_Knight [Accessed 12-Sep-2023].

[38] Wikipedia contributors. 2018. Bloons TD 6. https://en.wikipedia.org/wiki/Bloons_TD_6 [Accessed 12-Sep-2023].

[39] Wikipedia contributors. 2020. Hades (video game). https://en.wikipedia.org/wiki/Hades_(video_game) [Accessed 12-Sep-2023].

[40] Wikipedia contributors. 2022. Network emulation. https://en.wikipedia.org/wiki/Network_emulation [Accessed 20-Oct-2022].

[41] Xiaokun Xu and Mark Claypool. 2022. Measurement of Cloud-Based Game Streaming System Response to Competing TCP Cubic or TCP BBR Flows. In *Proceedings of the 22nd ACM Internet Measurement Conference*. Nice, France, 305–316.

[42] Saman Zadtootaghaj, Steven Schmidt, and Sebastian Möller. 2018. Modeling Gaming QoE: Towards the Impact of Frame Rate and Bit Rate on Cloud Gaming. In *Tenth International Conference on Quality of Multimedia Experience (QoMEX)*. Sardinia, Italy, 1–6.