Is 60 FPS Better than 30? The Impact of Frame Rate and Latency on Moving Target Selection

Benjamin F. Janzen
McMaster University
1280 Main Street
Hamilton, ON L8S 4L8 Canada
janzenbf@mcmaster.ca

Robert J. Teather
McMaster University
1280 Main Street
Hamilton, ON L8S 4L8 Canada
teather@mcmaster.ca

CHI 2014, Apr 26 - May 01 2014, Toronto, ON, Canada
ACM 978-1-4503-2474-8/14/04.
http://dx.doi.org/10.1145/2559206.2581214

Abstract

We present a pilot study investigating the relationship between frame rate and latency and their effects on moving target selection. In several latency/frame rate conditions, participants were given a 20 second time frame to click as many moving targets as possible. Performance with 60 FPS frame rate was 14% higher than 30 FPS, but the difference between 45 and 60 FPS was not significant. Latency alone had lower impact than the corresponding frame rate difference. While both factors impact performance, frame rate had a larger effect than the latency it introduces.

Author Keywords

Pointing; Latency; Frame Rate

ACM Classification Keywords

H.5.2. Information interfaces and presentation (e.g., HCI): User Interfaces.

Introduction

Video game developers strive to improve visual detail, but highly detailed scenes take longer to render. This limits the frame rate, typically reported as the number of frames rendered per second. High frame rates yield smooth animation, while low frame rates cause moving objects to "stutter" or jump as they move further between frames [1]. The performance impact of frame rate is hotly debated amongst gamers, especially since some platforms allow manual control of frame rate by adjusting graphical detail. Reducing graphics settings can improve frame rate, and vice versa.

We investigate the effect of frame rate on pointing tasks, which are well-understood [8] and fundamental to interacting with games. Low frame rates reduce performance in games [1] and 3D selection [10]. Most previous work used stationary targets [5, 6, 9], but we investigate these effects on the *moving* target selection. This appears to be a novel contribution.

A possible explanation is latency (input delay) differences between frame rate conditions. Decreasing frame rate yields extra latency due to the increased time between frames. Latency is known to negatively affect performance [6, 7, 9]. Lower frame rates also make it more difficult to discern what is happening in animation due to the aforementioned jumping effect [4]. Not only is there a longer delay between frames, there are fewer frames per time unit. Thus, we investigate how much of the performance cost of frame rate can be attributed to each of these effects.

Related Work

The impact of latency on pointing performance has been extensively studied in 2D user interfaces [5, 6], 3D user interfaces and virtual reality [9, 10], in pursuit tracking tasks [7], and on touchscreen systems [3]. Pavlovych and Stuerzlinger report no performance difference between latency levels lower than 58 MS, but a 10-15% difference at around 83 to 100 ms of latency [6, 7], consistent with earlier reports by Teather et al. [9]. We include latency in our experiment, as frame rate differences yield latency differences. For example, changing frame rate from 60 FPS to 30 FPS introduces an additional 16.6 ms of latency. Latency and frame rate can also be independent. Consider, for example, a fast but delayed frame rate (e.g., 120 FPS at a 100 ms latency delay) due to processing or network latency. We thus consider both factors together.

Small frame rate drops are more easily perceived than small increases in latency, and thus may have a larger performance cost. Keval and Sasse [4] found that low frame rate impedes perception of object movement. Participants were unable to consistently choose which of three events took place in video recorded at lower than eight frames per second. This does not apply to static objects, however. Garaj et al. [2] found that low frame rate did not impact participants' ability to detect static hazards.

Claypool and Claypool focused on gaming tasks with keyboard controls, testing participants' ability to shoot targets or pick up items under several different frame rates, display resolutions, and view perspectives [1]. Performance dropped significantly at 15 FPS but they did not use frame rates higher than 30 FPS. We focus on more common mouse pointing tasks and include frame rates of 45 and 60 FPS in our experiment.

Methodology

Participants

Twelve university students (8 male) were recruited for the experiment. Nine were between 20 and 30 years old, and the rest were over 30 years. All but one used computer mice right-handed; the last (left handed) participant was later dropped as an outlier.

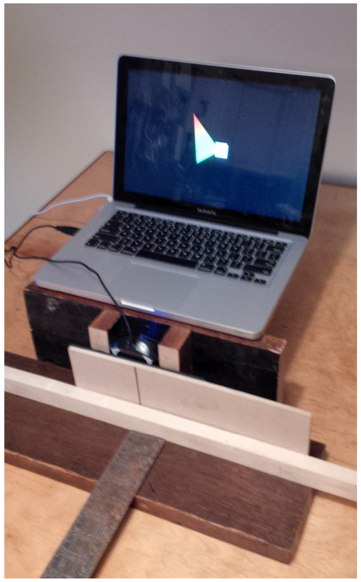
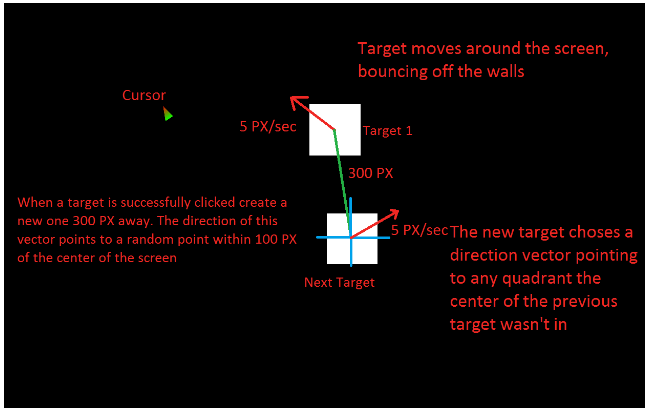


Figure 1. Latency measurement apparatus. The board had a mouse pad attached to it, which was moved at a fixed rate across the mouse sensor. Peak movement differences (in number of frames) in mouse and pointer motion were counted to derive the baseline system latency.



Target moves around the screen, bouncing off the walls

When a target is successfully clicked create a new one 300 PX away. The direction of this vector points to a random point within 100 PX of the center of the screen

The new target choses a direction vector pointing to any quadrant the center of the previous target wasn't in

Figure 2. Software showing a newly created target (labeled "Next target") and the target just clicked by the user ("Target 1"). Note that arrows and text shown only for reference and did not appear in the actual software

Apparatus

The experiment was performed on a MacBook Pro running OS X 10.8.4 with a 2.9 GHz Intel Core i7 Processor, 8 GB of RAM, and Intel HD Graphics. The display measured 13'' diagonally at a 1280 × 720 pixel resolution. We used a Razer *DeathAdder* mouse at 3500 DPI precision, a 1000Hz sampling rate, and 1 ms response times. Mouse acceleration was disabled and CD gain was set to the default level.

We used custom software written in C++ and OpenGL that displayed a single white 100 x 100 pixel moving target. See Figure 2. Upon successfully clicking the target, it flashed green for 100 ms and a new target was created at a random position 300 pixels away from the previous target. The new target would then start moving along a random vector at a speed of 5 pixels per second. If a target reached the screen edge, it would rebound on the same angle. Frame rate was controlled with an internal clock to swap screen buffers at specific times. Input latency was simulated by queuing both pointer motion and mouse button events for the specified duration.

Base system latency was determined using a variation of the "pendulum" approach [6, 7, 9]. The latency measurement apparatus is shown in Figure 1. The mouse was held stationary while a mouse pad affixed to a marked board was moved across its sensor. Video recorded at 60 FPS was then analyzed for frame differences between the mouse and pointer motion. Over 10 trials mean latency was around 38±5 ms.

Design

The experiment used a 4x3 within-subjects design. The independent variables were frame rate (15, 30, 45, 60 FPS) and latency (0, 50 and 100 ms) added to the base system latency of 38 ms. The twelve combinations of latency and frame rate were counterbalanced with a balanced Latin square. To simulate variable frame rates or network latency while playing a game, a single trial encompassed several such conditions. Each trial lasted for 80 seconds. Every 20 seconds a new combination of the independent variables was chosen while the task continued. Three such trials were needed to test all conditions. The dependent variables were targets clicked (number clicked in 20 seconds), and error rate (number of clicks missing the target).

Results and Discussion

As mentioned earlier, one participant was excluded as an outlier due to have scores lower than 3 standard deviations from the mean score.

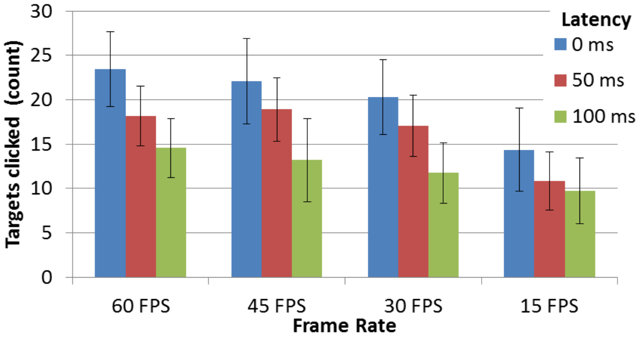


Figure 3. Mean number of targets clicked within 20 seconds for all conditions. Latency indicates the amount of artificially added extra latency (in addition to the baseline 38 ms). Error bars show ±1 SD.

There were significant main effects for both frame rate ($F_{3,10} = 40.4$, $p < .001$) and latency ($F_{2,10} = 187.1$, $p < .001$). The interaction effect between frame rate and latency was also significant ($F_{6,60} = 3.6$, $p < .005$). The mean number of targets clicked for each condition is shown in Figure 3.

The combination of 60 FPS and 0 ms of extra latency was considered the baseline condition. We first discuss only the effect of frame rate (i.e., comparing across the 0-latency conditions, shown with blue bars in Figure 3). Baseline performance was approximately 6% higher than 45 FPS. According to posthoc tests, this was not significant ($p = 0.26$). Baseline performance was 14% higher than the 30 FPS condition. This difference *was* significant ($p < .005$). The most pronounced performance difference was between the baseline and 15 FPS. Baseline performance was approximately 39% higher than this low frame rate condition, and this too was significant ($p < .0001$).

This confirms previous results which reported a large performance difference between 30 and 15 FPS frame rates [1]. However, we included higher frame rates than Claypool and Claypool. Our results show a moderate performance improvement past 30 FPS which tapers off somewhere between 45 and 60 FPS.

Latency too significantly impacted performance, also confirming previous results [5, 6, 9]. Considering only latency, we first compare the baseline to the other two latency conditions with 60 FPS frame rates. Per posthoc tests, baseline performance was approximately 22% higher than the 50 ms latency condition. This difference was statistically significant ($p < 0.001$). Baseline performance was around 38% higher than the 100 ms latency condition and this too was significant ($p < 0.0001$). We note that the relative performance cost of latency was much higher than reported in previous work. Based on previous work, we expected that adding 100 ms of latency would degrade performance by around 15% [6, 9].

A possible explanation for this difference is that the performance cost of latency was higher in our experiment due to moving targets. Most previous work on latency used static targets [5, 6, 9, 10]. Moving targets are harder to hit than static targets and latency exacerbates this. Targets continue to move *during* the latency-delayed click, hence the likelihood of missing increases. The participant clicks the mouse, but the button event is delayed until *after* the target moves out from under the pointer. Participants can partially compensate for this by leading the target, but we believe this explains the stronger effect of latency in our experiment relative to previous work.

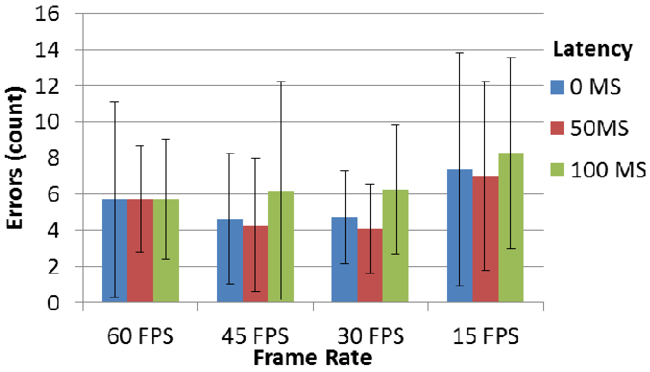


Figure 4. Mean number of errors made within 20 seconds for each frame rate/latency condition. Latency represents the amount of artificially added extra latency (in addition to the baseline 38 ms). Error bars show ±1 SD.

Lower frame rates also increase latency, since the time until the next frame is presented increases [10]. For most frame rate differences in our experiment, the latency difference is small. For example, the latency difference between 60 FPS and 45 FPS is only around 5.5 ms (16.67 ms frame time vs. 22.22 ms frame time). Such small latency differences are unlikely to matter. Hence the performance differences in these conditions are likely due to frame rate differences alone. However, there is a 50 ms latency difference between 60 FPS and 15 FPS frame rates (16.67 ms frame time vs. 66.67 ms frame time). As noted above, 50 ms of latency at 60 FPS yielded a 22% performance cost. In contrast, the corresponding frame rate (15 FPS) yielded an approximately 39% performance cost - as much as the 100 ms latency condition. We thus argue that frame rate has a larger impact on moving target selection than an equivalent amount of latency. This is further supported by the interaction effect between latency and frame rate noted above: latency alone did not significantly affect performance in the 15 FPS frame rate conditions.

Error rates are shown in Figure 4. Since error rate data were not normally distributed ($W = 0.84$, $p < .0001$), these were analyzed these using the non-parametric Friedman test. While there is a visual trend of increased error rates at lower frame rates and higher latencies, there was no significant difference in any conditions ($\chi^2_{11} = 2.4$, $p > .05$). The variance in the error rate was simply too high to derive any useful results. This is likely due to the fact that targets were not reset if the participant missed the target. Some participants would rapidly click in the general area of the target while refining their aim. This allowed participants with less experience in clicking moving targets to have comparable scores in terms of total targets clicked, but clearly added too much variability into the error rate to gain any additional insight.

Conclusion

We investigated participants' ability to select moving targets under several frame rate and latency conditions. This experiment confirms that low frame rates have a significant performance cost. This improves somewhat by increasing the frame rate (e.g., from 30 to 60 FPS). The negative impact of latency was also confirmed. Notably, in the lowest frame rate conditions, latency did *not* significantly affect performance. These results suggest that frame rate more strongly affects moving target selection than latency. These results provide further evidence of the importance of high frame rates. We thus suggest that game developers and enthusiasts alike adjust graphics settings to target a frame rate of at least 45 FPS.

Future Work

We plan to modify the task and use more participants to try to reduce variance in the dependent variables. In particular, and as noted earlier, error rates were simply too variable to analyze. This could be improved by resetting the target after a miss, and preventing users from clicking repeatedly near targets, and may provide more meaningful results.

Future work could also investigate the abnormally high impact (relative to previous work) of input latency we observed. We speculate that this is due to the use of moving targets in the experiment. The participants had to lead the targets to ensure it didn't move before the click registered. We plan to investigate whether users can train to adjust for this latency with practice, and if such training happens faster or slower with stationary targets. One question raised is if having a higher but consistent level of latency is better for performance than a smaller amount of variable latency.

References

[1] Claypool, M. and Claypool, K., Perspectives, frame rates and resolutions: it's all in the game, Proceedings of the International Conference on Foundations of Digital Games, (New York: ACM, 2009), 42-49.

[2] Garaj, V., Hunaiti, Z., and Balachandran, W., The effects of video image frame rate on the environmental hazards recognition performance in using remote vision to navigate visually impaired pedestrians, Proceedings of the International Conference on Mobile Technology, Applications, and Systems (New York: ACM, 2007), 207-213.

[3] Jota, R., Ng, A., Dietz, P., and Wigdor, D., How fast is fast enough? A study of the effects of latency in direct-touch pointing tasks, Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems - CHI 2013, (New York: ACM, 2013), 2291-2300.

[4] Keval, H. and Sasse, M. A., To catch a thief - you need at least 8 frames per second: the impact of frame rates on user performance in a CCTV detection task, Proceedings of the 16th ACM International Conference on Multimedia, (New York: ACM, 2008), 941-944.

[5] MacKenzie, I. S. and Colin, W., Lag as a determinant of human performance in interactive systems, Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems - CHI '93, (New York: ACM, 1993), 488-493.

[6] Pavlovych, A. and Stuerzlinger, W., The tradeoff between spatial jitter and latency in pointing tasks, Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems, (New York: ACM, 2009), 187-196.

[7] Pavlovych, A. and Stuerzlinger, W., Target following performance in the presence of latency, jitter, and signal dropouts, Proceedings of Graphics Interface 2011, (Toronto: Canadian Human-Computer Communications Society, 2011), 33-40.

[8] Soukoreff, R. W. and MacKenzie, I. S., Towards a standard for pointing device evaluation: Perspectives on 27 years of Fitts' law research in HCI, International Journal of Human-Computer Studies, 61, 2004, 751-789.

[9] Teather, R. J., Pavlovych, A., Stuerzlinger, W., and MacKenzie, I. S., Effects of tracking technology, latency, and spatial jitter on object movement, IEEE Symposium on 3D User Interfaces, (New York: IEEE, 2009), 43-50.

[10] Ware, C. and Balakrishnan, R., Reaching for objects in VR displays: Lag and frame rate, ACM Transactions on Computer-Human Interaction 1 (4), 1994, 331-356.