Performance of Modern Gaming Input Devices in First-Person Shooter Target Acquisition

Alexander Zaranek
McMaster University
1280 Main St. West
Hamilton, ON L8S 4K1 CA
zaranea@mcmaster.ca

Bryan Ramoul
McMaster University
1280 Main St. West
Hamilton, ON L8S 4K1 CA
ramoulb@mcmaster.ca

Hua Fei Yu
McMaster University
1280 Main St. West
Hamilton, ON L8S 4K1 CA
yuhf3@mcmaster.ca

Yiyu Yao
McMaster University
1280 Main St. West
Hamilton, ON L8S 4K1 CA
yaoy4@mcmaster.ca

Robert J. Teather
McMaster University
1280 Main St. West
Hamilton, ON L8S 4K1 CA
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/10.1145/2559206.2581291

Abstract

We present a pilot study quantifying the targeting performance of several modern game input devices. These included a mouse, a game controller, the PS Move and the Kinect. Our study used a 3D first-person shooting game task, based on the ISO 9241-9 experimental paradigm for evaluating pointing devices. Comparison of performance measures indicated that the mouse was best, with the game controller coming in a close second. Performance of the 3D input devices (Move and Kinect) was much worse.

Author Keywords

Fitts' law, 3D target selection, first-person shooter.

ACM Classification Keywords

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

Introduction

There is great variety in modern game input devices. While these are often tied to a specific platform (e.g., the "Wiimote") gamers can sometimes choose which input device to use. This had led to arguments about what input device offers the best performance in first-person shooter (FPS) games. To help settle this debate,

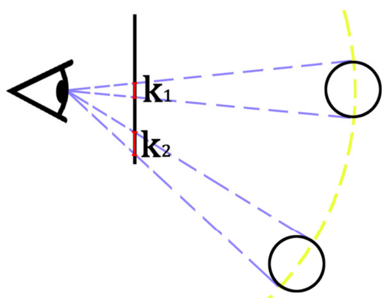

Figure 1. Perspective distortion of peripheral objects. The black line represents the display surface and k_1 and k_2 are the projected size of the same target at different view angles. Although the object size and distance from the eye is the same, k_2 is clearly larger as it is closer to the display edge. Perspective scaling further complicates this when targets are presented at different depths.

we compared several game devices in a shooting task. Previous work [4-6, 9] suggests that the mouse will outperform a game controller, the Sony *PlayStation Move*, and the Microsoft *Kinect*. However, we evaluate this using a more ecologically valid approach than previous work.

Past experiments used 2D pointing tasks based on Fitts' law [2, 7] and the ISO 9241-9 standard [3] for evaluating pointing devices. While there are many advantages to using this standardized methodology, it does not fully represent targeting tasks in an FPS game environment. Evaluation of FPS shooting tasks is complicated by the fact that aiming also rotates the viewpoint [6]. In addition, there are multiple targets presented at varying depths in the 3D scene, so perspective also affects targeting [12]. To address these issues we developed a 3D shooting-range game that presents stationary targets at varying depths. The software uses rotation-based targeting and perspective projection to simulate FPS games more accurately.

Related Work

Since shooting in an FPS is ultimately a point selection task, Fitts' law should apply. Fitts' law states that the time to acquire a target is logarithmically related to the distance and target size [2, 7]. The law predicts movement time as $MT = a + b \times ID$. The a and b coefficients are empirically derived via linear regression. ID is the index of difficulty (in bits) and given as $ID = \log_2(D/W + 1)$. D and W are the distance to and width (size) of the target, respectively. However, both distance and size scale due to perspective, which influences the pointing task [12]. This is further complicated by the fact that perspective will distort the size of targets closer to the screen edge, see Figure 1. Hence we also consider target depth in our experiment and use screen-space projections of targets for calculating D and W.

Fitts' Law has been widely used in comparing mice [6, 7, 12], trackballs [10], game controllers [1, 8, 9], and other input devices. Past work has shown that the mouse tends to outperform game controllers in target acquisition tasks [1, 4, 5, 9]. However, most of these studies used 2D pointing tasks. Isokoski and Martin [4] compared the mouse to game controllers in a first-person shooter task. While they report superior mouse performance, their experiment does not conform well to standardized Fitts' law methodology. Looser et al. [6] used a variant of a Fitts' law pointing task with a first-person perspective including mouse-based viewpoint rotation. They compared this to a standard fixed viewpoint. While the first-person view task conformed to Fitts' law, pointing speed was slower compared to traditional fixed viewpoint selection tasks [6].

Methodology

Participants

Twelve university students (mean age 20.9 years, *SD* 2.2) were recruited. Nine had little experience with FPS games. The remaining three played FPS games for between 1 and 10 hours per week.

Apparatus

The experiment was performed on a laptop (2.4 GHz Intel Core i7 CPU, 8 GB of RAM, Intel HD Graphics 4600 GPU) running Windows 7. The display measured 17 in. diagonally with a 1920x1080 pixel resolution and a 60Hz refresh rate. Four input devices were used: a mouse, a Microsoft *Xbox 360* game controller, a Sony *Playstation Move*, and a Microsoft *Kinect*. The Move also

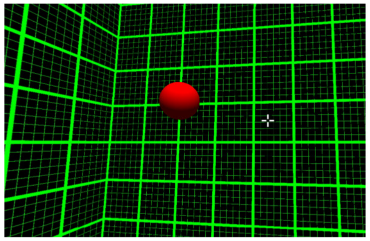

Figure 2. Software used in the experiment. The red sphere is a target, and the white crosshair is the pointer. Orientation information is provided to the viewer by way of the grid lines depicted in the background.

required a *Playstation Eye* camera for tracking. The Kinect and Move devices used third party software to map 3D input to 2D mouse cursor movement by ignoring the depth component of the 3D input. These programs were Kinect Magic Cursor[1] and MoveForPC[2] respectively. Control sensitivity and mappings were left at default values for all devices. All devices took translational input which was mapped to the viewport rotation.

The software was developed in Unity and presented a 3D shooting range, see Figure 2. Participants were required to aim (point) at and shoot (select) several targets presented at varying depths. The software used a 70º field of view and perspective projection. The background in the software was a green grid; this helped the user maintain a sense of orientation. One stationary red target sphere was displayed at a time. Upon clicking, the next target appeared if the previous target was hit or nearly hit. Misses that were farther away than 200 pixels of the target in screen-space counted as misses and did not advance the trial.

Targets were positioned in predetermined but unpredictable locations at one of three planar depths away from the user. These were classified as “near”, “medium” and “far” with depths of 5 m, 10 m, and 15 m respectively. Four targets were presented consecutively at each depth. Targets were placed relatively close together (i.e., not behind the user) to ensure that they were always visible to the user, limiting visual search time when the target appeared. The camera position was fixed and the user could only control the camera orientation. Target hits and misses were indicated through audio feedback. The software automatically logged time to shoot each target, misses, and the number of target reentries. We omit our analysis of target reentries due to space constraints. The software also calculates the screen-space projected size and distance of the targets. These screen-space projected values are used to compute *ID* for the trial, and ultimately throughput as described below.

Procedure

Each participant completed the target shooting trials for all input devices. After giving informed consent, the participant was seated half a meter from the display and given the first input device. Participants using the Kinect were seated approximately 2 meters away in order for their entire body to be within frame. Participants were instructed to select the red target sphere, and to focus on speed over accuracy. Upon completion of all trials with a device, participants were given the next device. The entire experiment took around twenty minutes for each participant.

Design

The study used a 4 x 3 within-subjects design. The independent variables and their levels were:

Device: mouse, controller, Move, Kinect
Target Depth: near, medium, far

Input device order was counterbalanced with a 4x4 balanced Latin square. Target depth was ordered randomly (without replacement) in each trial. The

[1] Renton, David, Kinect magic cursor V1.7, 2013. http://drenton72.wordpress.com/2013/05/09/kinect-magic-cursor-version-1-7-with-gesture-support

[2] Rosado, Osvaldo, MoveForPC, 2012. http://osvaldojr.com/index.php/2011/04/18/playstation-move-as-pc-pointing-device

Effect	d.f.	Movement *F*	Movement *p*	Error Rate *F*	Error Rate *p*	Throughput *F*	Throughput *p*
(D)evice	3, 11	18.7	< .0001*	11.3	< .0001*	175.5	< .0001*
(T)arget Depth	2, 11	2.6	> .05	38.8	< .0001*	20.1	< .0001*
D × T	6, 66	2.9	< .05*	12.5	< .0001*	8.7	< .0001*

Table 1. Statistical effects for movement time, error rate, throughput, and target re-entry results. Significant results are marked with an asterisk by the p value.

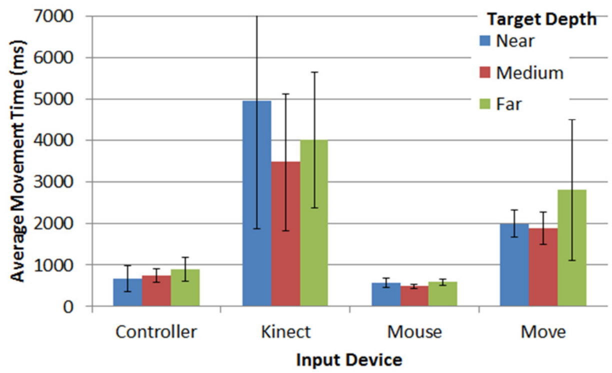

Figure 3. Mean movement time by input device and target depth. Error bars show ±1 *SD*.

dependent variables were pointing throughput (in bits per second), error rate (count of misses per trial) and movement time (ms). Each participant completed 48 trials, for a total of 576 recorded trials overall.

Results

Data was analyzed using repeated measures ANOVA. Statistical reports are shown in Table 1.

Movement Time

Mean movement time scores are shown in Figure 3. The mouse was the fastest overall with an average movement time of 539 ms. The Kinect was slowest, with an average movement time of 4145 ms. The interaction effect between device and target depth indicates that the Move was significantly worse for far targets ($p < .05$). Although the Kinect appears to be worse for *near* targets, the scores were too variable to determine if this was significant. The controller (average movement time 760 ms) was not significantly slower than the mouse.

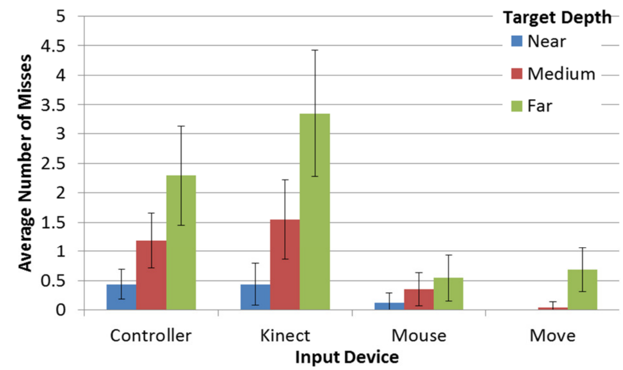

Figure 4. Mean errors per selection. Error bars show ±1 *SD.*

According to Fitts' law, smaller targets should take longer to aim at and shoot. In the case of our 3D shooting range, when the target is further away from the view point, the participant has a harder time aiming since the target appeared smaller due to perspective. This is reflected in the much worse error rates for the far conditions, but is also visible in the significantly worse movement time for far targets with the Move. The difference in movement time for near and medium targets are otherwise not significant. This is likely because in screen-space, the near and medium targets are not substantially different in size from each other compared to the far targets. This is due to the nature of perspective, in particular, the aforementioned effect of skewing near the screen edges.

Accuracy

Error rates are shown in Figure 4. Although the controller was not significantly slower than the mouse, its error rates are significantly higher according to the Tukey-Kramer HSD test ($p < .05$). This may be because unlike commercial games, we did not calibrate the controller to reduce noise for simplicity. The Move and mouse had similar error rates, despite large differences in movement time – these were not significantly different ($p > .05$). Participants were more cautious and accurate with the Move, and thus slower.

Overall, error rates were significantly higher for far targets, especially with the Kinect. This is likely due to the combined effects of perspective scaling and input device noise.

Throughput

Throughput was calculated as $TP = \log_2(D/W_e + 1)/MT$ in accordance with the ISO 9241-9 standard [3]. W_e is the effective width of the target and was calculated as $4.133 \times SD_x$. SD_x is the standard deviation of the over/undershoot distances relative to the target center along the movement axis. Prior to this accuracy adjustment, all targets and motions were first projected to the screen plane. Based on previous work [12], we believe this "screen-projected" throughput makes more sense than using 3D size/distance measures, since all of our input devices operate in the screen plane.

Throughput scores are shown in Figure 5. Mouse throughput was highest, followed by the controller. The Kinect and Move were lowest. Based on previous work [12], we did not expect the effect of target depth to be significant. However, for the mouse, close targets had significantly lower throughput. We believe this is because of differences in the experimental methodology between our experiment and previous work. In particular, this may be due to measuring width and distance in a 3D environment with pan-based viewpoint movement. Previous work [12] used a fixed viewpoint.

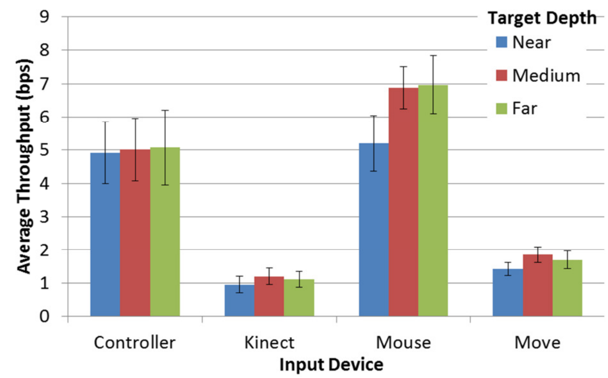

Figure 5. Average throughput for each condition. Error bars show ±1 *SD.*

As mentioned earlier, in 3D space, perspective skews the distance and width of targets that are closer to the display edge. We believe this artificially inflated mouse throughput, especially for medium and far depths. These scores are also abnormally high – approximately 7 bps vs. the expected 4.5 bps [12]. Throughput for the near condition was comparable to previous work though. We thus argue that additional extensions to throughput to better compensate for 3D pan-based target selection would yield more accurate results. This is a topic for future consideration though.

Discussion and Conclusions

Performance with the Kinect and Move was much worse than the mouse and controller, although the Move was slightly better than the Kinect. This is likely because the Kinect was subject to large amounts of input noise, which made accurate aiming extremely difficult. Limited participant familiarity with these devices may also have contributed to these results. Both devices require better spatial awareness than the mouse and controller. While this yields more natural input mappings, these devices tend to have noisier input and are subject to greater latency; both can degrade performance [11]. The Kinect is also intended to be operated in a standing position. Since our participants were seated device noise may have been higher than normal.

Our results confirmed that while the mouse offered the best performance, the controller was competitive; both devices offered much better performance than the Move and Kinect. This may be due in part to tactile feedback offered by the mouse (which slides along a stable surface) and the controller thumbstick (which is locked in a socket). The much lower performances of the Move and Kinect are good indications as to why these devices aren't commonly employed for FPS games and are often seen as a novelty. These devices allow 3D input which is not utilized in most FPS games, as target aiming in such systems is ultimately a 2D screen-plane task.

Future work would focus on tactile feedback and dimensional constraints of the input devices. Further work could also look at FPS games incorporating full 3D input. We also plan to look further at the computation of throughput for FPS shooting tasks.

References

[1] Castellucci, S. J., Teather, R. J., and Pavlovych, A., Novel metrics for 3D remote pointing, ACM Spatial User Interaction, (New York: ACM, 2013), 17-20.

[2] Fitts, P. M., The information capacity of the human motor system in controlling the amplitude of movement, Journal of Experimental Psychology, 47, 1954, 381-391.

[3] ISO, ISO 9241-9 Ergonomic requirements for office work with visual display terminals (VDTs) - Part 9: Requirements for non-keyboard input devices: International Standard, International Organization for Standardization, 2000.

[4] Isokoski, P. and Martin, B., Performance of input devices in FPS target acquisition, International Conference on Advances in Computer Entertainment Technology, (New York: ACM, 2007), 240-241.

[5] Klochek, C. and MacKenzie, I. S., Performance measures of game controllers in a three-dimensional environment, Graphics Interface, (Toronto: Canadian Information Processing Society, 2006), 73-79.

[6] Looser, J., Cockburn, A., and Savage, J., On the validity of using first-person shooters for Fitts' law studies, British HCI Conference, (London: British Computer Society, 2005), 33-36.

[7] MacKenzie, I. S., Fitts' law as a research and design tool in human-computer interaction, Human-Computer Interaction, 7, 1992, 91-139.

[8] McArthur, V., Castellucci, S. J., and MacKenzie, I. S., An empirical comparison of Wiimote gun attachments for pointing tasks, Engineering Interactive Computing Systems, (New York: ACM, 2009), 203-208.

[9] Natapov, D., Castellucci, S. J., and MacKenzie, I. S., ISO 9241-9 evaluation of video game controllers, Graphics Interface, (Toronto: Canadian Information Processing Society, 2009), 223-230.

[10] Natapov, D. and MacKenzie, I. S., The trackball controller: improving the analog stick, ACM FuturePlay, (New York: ACM, 2010), 175-182.

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

[12] Teather, R. J. and Stuerzlinger, W., Pointing at 3D target projections using one-eyed and stereo cursors, ACM Conference on Human Factors in Computing Systems - CHI 2013, (New York: ACM, 2013), 159 - 168.