The Trackball Controller: Improving the Analog Stick

Daniel Natapov I. Scott MacKenzie

Department of Computer Science and Engineering
York University, Toronto, Canada

{dnatapov, mack}@cse.yorku.ca

ABSTRACT

Two groups of participants (novice and advanced) completed a study comparing a prototype game controller to a standard game controller for point-select tasks. The prototype game controller replaces the right analog stick of a standard game controller (used for pointing and camera control) with a trackball. We used Fitts' law as per ISO 9241-9 to evaluate the pointing performance of both controllers. In the novice group, the trackball controller's throughput was 2.69 bps – 60.1% higher than the 1.68 bps observed for the standard controller. In the advanced group the trackball controller's throughput was 3.19 bps – 58.7% higher than the 2.01 bps observed for the standard controller. Although the trackball controller performed better in terms of throughput, pointer path was more direct with the standard controller.

Categories and Subject Descriptors

H.5.2 User Interfaces. Input devices and strategies.

General Terms

Design, Experimentation, Human Factors.

Keywords

Game controllers, trackball, analog joystick, pointing devices, performance evaluation, Fitts' law, video games

1 INTRODUCTION

The video game industry is a profitable one. Computer and video game software sales in the United States alone were estimated at $11.7 billion in 2008, up from $9.5 billion in 2007 [3]. Despite the profitability of the industry, little empirical research exists on the topic of video game usability. One such un-researched area is video game input, or game controllers. The focus of this paper is the improvement of the camera controlling analog stick of standard handheld game controllers. To illustrate the reason for the problem, before discussing the improvement, we provide a short history of game controllers, both on consoles and the PC.

1.1 Console Controls

The current design of console controllers originated in 1983 with the Nintendo *Entertainment System* (*NES*) gamepad. The *NES* gamepad had a directional pad (D-Pad) – a 'plus' shaped button that is pressed in four directions (up, down, left, right) with four diagonal combinations. The D-Pad specified the movement direction of a game character. The gamepad also had two buttons for game-dependant actions, such as jump or shoot. Since most games were simple and two-dimensional, this relatively low number of inputs was sufficient. Despite many future additions and improvements, the D-Pad persists on all standard controllers for all consoles introduced after the *NES*.

Shortcomings of the D-Pad became apparent with the introduction of 3D games. The Sony *PlayStation* and the Sega *Saturn*, introduced in 1995, supported 3D environments and third-person perspectives. The controllers for those consoles, which used D-Pads, were not well suited for 3D, since navigation was difficult. The main issue was that game characters could only move in eight directions using the D-Pad. To overcome this, some games, such as *Resident Evil*, used the forward and back directions of the D-Pad to move the character, and the left and right directions for turning. But this was cumbersome [1]. There was also no easy way to pan the camera up or down.

The Nintendo *64* (*N64*), introduced in 1996, included an analog stick. The analog stick was essentially a thumb joystick. Unlike its name, the *N64* analog stick was not actually analog, but digital. It had a set number of sensitivity levels, dependant on how far the joystick was displaced, which made it feel analog [8]. Most modern analog sticks use potentiometers and are actually analog. The analog stick was a better fit for 3D games as it enabled motion in all directions, as opposed to just eight. Additionally, it had different sensitivity settings which enabled it, for instance, to differentiate walking from running depending on the analog stick's displacement. The *N64* controller also had four buttons to pan the camera. The analog stick is now used more prominently than the D-Pad [13].

In 1997 Sony introduced a new controller for the *PlayStation* called the *Dual Analog* controller. Like the *N64*, it used an analog stick for movement. It also replaced the *N64's* four camera buttons with a second analog stick on the right side of the controller. The second analog stick positioned the camera in 3D games, by panning in a certain direction [13]. The analog stick was better suited for camera control since, unlike buttons, it allowed panning in more than four directions and at different speeds. The configuration of two analog sticks became the standard for all console controllers since, with the exception of the Nintendo *Wiimote*, which uses an infrared camera for pointing.

1.2 PC Controls

In parallel to advances in console game controllers, game input evolved differently on the PC. During the period of 2D games, the keyboard's arrow keys were generally used for character movement, with other keys used for game-dependant actions. With the introduction of 3D games, new controls were needed to move the camera, or in the case of First Person Shooter (FPS) games, to aim. In 1993, id Software released *Doom*, which was one of the earliest FPS games. It used the forward and back arrow keys to move, and the left and right keys to turn, similar to how the D-Pad was used in *Resident Evil* on the *PlayStation*.

Quake was released by id Software in 1996. The default configuration used the arrow keys to move and turn, and

FuturePlay @ Vancouver Digital Week 2010, May 6-7, 2010, Vancouver, Canada.

PAGE DOWN and DEL to look up and down respectively. *Quake*, unlike previous FPS games, was truly 3D, meaning it included height levels. This made looking up and down necessary to target enemies at different heights. A second control option for *Quake* was free look (or mouselook) [1]. Mouselook allowed players to control the camera using the mouse instead of buttons. The mouselook option was so successful that it is now the default method of camera control for PC games.

Overtime, the arrow keys in FPS games were replaced with the W, S, A, and D keys, with W and S for forward and backward movement, and A and D for strafing (lateral movement), as opposed to turning. This switch occurred because of the redundancy of the turning and camera control keys with mouselook, and because the configuration allowed the left hand to reach more neighboring keys (such as the number keys for weapon selection).

This led to a large discrepancy between PC controls and console controls. The PC uses the mouse for pointing, targeting and camera control, while gamepads use the right analog stick. The mouse is better suited for these tasks since it is a position-control device, unlike the analog stick, which is a rate-control device. While the mouse can position the camera directly, the analog stick only specifies the panning direction and speed of the camera, thus limiting the speed and accuracy of direct positioning.

Though the analog stick has different sensitivity settings, it does not allow complete control over camera speed – there is a maximum and a minimum panning speed and this makes certain tasks more difficult. For instance, to turn around in a FPS game, the camera must pan by 180 degrees, but with a limit on maximum panning speed, there is a cap on the turnaround speed. It is possible to turn faster by quickly moving the mouse than by panning the camera with the analog stick. Additionally, since there is a minimum speed of panning with the analog stick, small accurate movements are more difficult, making it easy to overshoot a nearby target, or pan farther than desired.

Pointing is similar to camera panning in 3D games. With pointing, the cursor is brought toward the target; with camera panning, the target (and scene) is brought toward the center of the screen where the cursor is positioned. It has been shown that Fitts' law, which is used to evaluate pointing performance (discussed in the next section), can also be used to accurately model 3D pan-based target acquisition [9]. We have previously evaluated the analog stick using Fitts' law; its throughput was 60.8% lower than the mouse's [12]. The research presented here is a follow on to these findings.

The lack of a precise pointing method on game controllers affects not only camera control in existing games, but also hinders certain genres from being developed for consoles. For instance, Real Time Strategy (RTS) games are more often developed for the PC than for consoles [14]. This is due, at least in part, to the difficulty of selecting individual units or structures with the analog stick. RTS games display many strategic units at once, so the size of each unit is small. Comparatively, selecting these small units is simple with the mouse. A higher precision pointing method on game controllers could alleviate this and facilitate RTS games as well as other genres for consoles.

We aim to improve console controllers by modifying the method of pointing and camera control. One such improvement is using a position-control device such as the mouse, as it offers better precision and acceleration control than the analog stick. The drawback of the mouse is that it requires a surface to operate on, making it unusable in a handheld configuration. Instead, we have created a prototype controller that replaces the second analog stick with a trackball (Figure 1; described in detail below).

Figure 1: Trackball controller

While the trackball has been previously evaluated using Fitts' law (e.g., [11]), it has not been evaluated under handheld conditions, such as on a gamepad. Handheld use of a trackball may prove unstable and thus not beneficial with game controllers. As such, a reevaluation of the trackball on a game controller is required. We conducted a comparative evaluation of the prototype controller (the "trackball controller") using Fitts' law. This paper presents our findings.

1.3 Related Work

1.3.1 Game Controllers

Previous work compared the pointing accuracy of an analog stick controller, the Nintendo *Wii Remote*, and a mouse [12]. Throughput was 3.78 bps for the mouse, 2.59 bps for the *Wii Remote*, and 1.48 bps for the analog stick controller. Our research is based on these findings. We aim to improve the poor pointing performance of the analog stick controller.

Klochek and MacKenzie presented five new performance metrics, such as the Time on Target (the time the cursor is kept within a moving target) and Mean Time-To-Reacquire (the time it takes for a lost target to be reacquired), to quantify differences between video game controllers [7]. The experiment compared an *Xbox* gamepad controller and a standard PC mouse in a target-tracking task. Performance was compared based on five cursor path metrics. The significant finding was that the mouse allowed participants more control over acceleration than the gamepad, which in turn helped correct errors in position. We believe that the use of a trackball will also allow more control over acceleration.

Isokoski and Martin compared a number of input devices, including an *Xbox 360* controller and a mouse and keyboard combination for use in FPS games [5]. Participants played a simple game with results showing that the keyboard and mouse combination performed better in terms of the number of target hits than the *Xbox 360* controller.

Looser et al. examined using FPS game environments to make Fitts' law experiments more entertaining and to motivate participants [9]. The FPS method of target acquisition involves panning the target to the cursor at the screen-center. Results showed that Fitts' law is well suited for the FPS target acquisition metaphor.

1.3.2 Evaluation of Pointing Devices

The ISO 9241-9 [4] standard is directed at the evaluation of non-keyboard input devices. It proposes a standardized methodology for evaluating performance and comfort. We used the multi-directional tapping task for evaluating performance. The main dependent variable is throughput, which is based on Fitts' Index of Performance [10]. Throughput (*TP*, in bps) is computed by dividing the index of difficulty (*ID,* in bits, averaged over a block of trials) by the average movement time (*MT*, in seconds):

$$TP = \frac{ID_{average}}{MT_{average}} \tag{1}$$

The calculation of *ID* is a logarithmic term known as the Shannon formulation. It includes *D* for movement distance and *W* for target width:

$$ID = \log_2\left(\frac{D}{W} + 1\right) \tag{2}$$

Instead of using the presented *IDs* (Eq. 2), the standard prescribes use of effective *IDs* to accommodate the spatial variability observed in responses [10]:

$$ID_e = \log_2\left(\frac{D_e}{4.1333 \times SD_x} + 1\right) \tag{3}$$

The term D_e represents effective distance – the distance a participant actually traversed along the task axis. The task axis is a straight line from the center of the source to the center of the target. The term *x* is the distance from the participant's click to the center of the selected target, projected on the task axis. SD_x is the standard deviation in *x* over a block of trials using the same *D* and *W*. Note that *x* can be positive or negative, depending on whether selection was an overshoot or undershoot, respectively.

Calculating throughput (Eq. 1) using ID_e (Eq. 3) yields an overall performance measure that includes both the speed and accuracy in user responses.

Considerable previous work exists on the comparison and evaluation of pointing devices (both following ISO 9241-9 and otherwise). We consulted the methodologies proposed in those papers for our own comparison and evaluation.

Research by Douglas et al. evaluated the scientific validity and practicality of ISO 9241-9 [2]. An experiment was conducted comparing a finger-controlled isometric joystick and a touchpad. Participants were asked to perform a point-select task, after which throughput for each device was calculated.

Kantowitz and Elvers evaluated an isometric joystick using Fitts' law [6]. Two control systems were compared with the isometric joystick: one of position-control and one of rate-control. Additionally the control-display gain, or sensitivity, with which the joystick moves the cursor, was modified. Using the joystick as a rate-control device resulted in steeper movement time slopes (lower throughput) than using it as a position-control device.

MacKenzie et al. proposed seven new accuracy measures for differentiating devices in precision pointing tasks [11]. The new measures were designed to capture patterns of movement during a trial. Four pointing devices, including a trackball, mouse, and joystick, were used in an experiment to validate the measures.

In the following section, we describe our methodology. The goal was to evaluate and compare the traditional controller with our prototype trackball controller using ISO 9241-9 and some of the measures proposed by MacKenzie et al. [11]. ISO 9241-9 employs a series of pointing tasks and, as was confirmed in research by Looser et al. [9], such tasks exemplify targeting and camera control in video games.

2 METHOD

2.1 Participants

The experiment included two groups of participants: novice game controller users and advanced game controller users. This design allows us to investigate both how easy the trackball controller is to use for new game players (as compared to using a standard controller), and how much retrogression in performance will occur for players already proficient with standard game controllers. Prior to beginning, all participants completed a pre-experiment demographic questionnaire. Three additional questionnaire items asked about experience with standard game controllers, trackballs, and video game playing. The responses were chosen from the following categories:

- Non-existent (never used)
- Rare (once a month or less)
- Occasional (several times a month)
- Frequent (at least once a week)

Each group had 10 paid participants recruited from the local campus and the local community. On average, participants took one hour to complete the study.

2.1.1 Novice Users

In the novice user group (5 male, 5 female) all participants were right-handed, though this was not by design. The mean age was 25.3 (*SD* = 3.19). Only participants with game controller experience rated as non-existent or rare were in this group. Seven stated that their experience with trackballs was non-existent, and the remaining three stated they used trackballs rarely. Six participants stated they played video games rarely, two stated they played occasionally, and two stated they played frequently. It was possible for game playing experience to exceed standard controller experience since a number of participants played games on the PC with a keyboard and mouse, or on hand-held platforms.

2.1.2 Advanced Users

In the advanced user group (10 males, not by design) eight participants were right-handed. The mean age was 25.2 (*SD* = 3.96). Only participants with game controller experience rated as occasional or frequent were in this group. Eight stated they used a standard controller frequently and two occasionally. Two participants rated their experience with trackballs as nonexistent; six as rare, one occasional and one frequent. Eight participants stated they played video games frequently, and two that they played occasionally.

2.2 Apparatus

2.2.1 Hardware

The experiment was conducted on a PC running *Windows XP*. We used a NEC *NP60* DLP projector (resolution 1024 × 768) for output, to simulate a large screen, since console games are generally played on large screen televisions. This was meant to increase external validity. Participants sat 3 meters from the projected image. The size of the projected image (diagonal) was 115 cm. The experimental setup is seen in Figure 2.

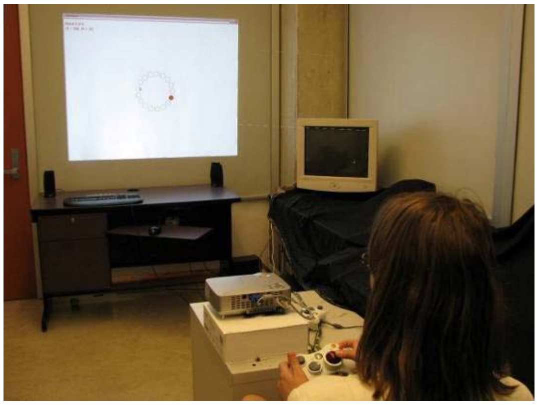

Figure 2: Experimental setup

A Microsoft *Xbox 360* wired controller was used as the standard controller (connected through USB). For the trackball controller a prototype was built from a Microsoft *Xbox 360* wired controller, and a wired Logitech *Trackman Wheel* trackball. We used an *Xbox 360* controller as the housing for the prototype so that both form factors were the same, except for the trackball replacement.

For the construction of the prototype controller, we replaced the right analog joystick with a trackball. The optical camera and original housing of the trackball were glued to the underside of the top half of the *Xbox 360* controller. The trackball was placed in the housing, and a plastic lid was attached so that the trackball would be moveable but secure. The rest of the trackball, excluding all unnecessary parts (such as button and scroll wheel controls), were attached to the bottom of the *Xbox 360* controller, where the battery pack is on a wireless controller.

The resulting prototype is seen in Figure 1. The prototype connects to the PC with two USB cables: one from the trackball, which controls mouse motion, and the other from the game controller, which controls all other inputs.*

For a real product, the trackball must be lower in height than in our prototype. We were limited in the placement of the trackball due to space constraints in the *Xbox 360* controller. The relatively high placement likely affected finger fatigue, since the thumb operated at an uncomfortable height. It also may have adversely affected performance, due to the height of the trackball exacerbating clutching (releasing the trackball and repositioning the thumb).

Pointing acceleration (or gain) was turned off to avoid bias, since the analog joystick cannot benefit from pointing acceleration, but the trackball could. In real use, pointing gain should be enabled when using the trackball controller. This would increase throughput and reduce clutching.

2.2.2 Software

We used software called *XPAdder* [15], which allows emulation of game controller input as mouse and keyboard input. In this research we used it to emulate a left trigger press as a left mouse button click, and to the right analog joystick (in the condition of the standard controller) as mouse cursor control.

XPAdder allows setting the analog stick's sensitivity when emulating a mouse. To select the best sensitivity, we conducted a short, preliminary user study with the *Xbox 360* controller, comparing the default setting to four other sensitivities (10% less, 5% less, 5% more and 10% more sensitive).

We used 10 participants in the preliminary study (different participants from the actual study) with each performing a short Fitts' task with one index of difficulty (*ID* = 3.16 bits) and 17 targets. The order of the five sensitivities (slow, medium-slow, medium, medium-fast, and fast) was counterbalanced between participants. Each participant took approximately ten minutes to complete the study. The results for throughput were 1.18 bps (slow sensitivity), 1.59 bps (medium-slow), 1.60 bps (medium), 1.45 bps (medium-fast), and 1.34 bps (fast sensitivity). Overall the difference was significant ($F_{4,36} = 5.97, p < .001$), but a Tukey-Kramer multiple comparison test revealed that the difference was not significant between the three medium settings (medium-slow, medium, medium-fast). Since the difference between these three settings was not statistically significant, we used the one with the highest throughput, which was also the medium and default setting of *XPAdder*.

The experiment used a *Java* program implementing Fitts' task according to ISO 9241-9. Seventeen circular targets were arranged in a circle centered in the middle of the projected screen. Targets were positioned at regular intervals along the perimeter of the circle, as prescribed by ISO 9241-9 (Figure 3). The next target to click was highlighted in red. Clicking the first active target would begin a trial and make the opposite target the active one. Clicking the next active target would end one trial and begin the next one, and so on. Each time a target was missed a "beep" sounded to indicate an error.

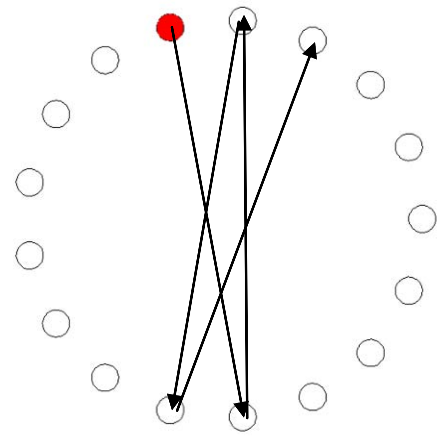

Figure 3: Screen capture of the experiment task, showing the sequence of the first five selections

Upon completion of all trials within a circle, a summary of the participant's performance for that *ID* appeared. When the participant clicked "OK", the next *ID* circle appeared. The duration between clicks and whether the selection was an error were logged. Supplementary data were also logged, including trace data for the cursor path.

2.2.3 Procedure

The purpose of the experiment was explained to participants. They were then asked to give informed consent for their participation and to complete the pre-experiment questionnaire. Before using each of the two controllers, participants were given a brief practice session. The results of the practice session (one *ID*) were discarded. During regular trials participants were instructed to click each red target as quickly and accurately as possible.

*A similar prototype was demonstrated at E3 2006, but not yet commercially produced or empirically evaluated (www.reflexcontrol.com)

2.2.4 Design

The experiment employed a 2 × 2 × 7 mixed design. The independent variables were:

- Experience (novice, advanced)
- Controller type (standard, trackball)
- Block (1 to 7)

Experience was a between-subjects factor. Controller type and Block were within-subjects factors. Controller type was counterbalanced with half the participants performing the task with the standard controller first. The other half performed the task with the trackball controller first.

Six combinations of target size and distance comprised the 6 indices of difficulty. See Table 1. The ordering of *ID*s within a block was randomized without replacement.

Size (pixels)	Distance (pixels)	*ID* (bits)
20	128	2.88
20	256	3.78
20	512	4.73
35	128	2.21
35	256	3.05
35	512	3.96

Table 1: Presented Indices of Difficulty (*IDs*)

There were 16 trials per *ID* circle. In total, participants in each group performed a total of 16 (trials per *ID*) × 6 (*ID*s) × 7 (blocks) × 2 (controller types) × 10 (participants) = 13,440 trials. For the two levels of Experience, 26,880 trials were recorded.

The dependent variables were throughput (bps), error rate (%), movement direction changes per trial (count), and movement error (pixels). The latter two measures were used in previous research [11] (described in the next section).

3 RESULTS AND DISCUSSION

3.1 Throughput

The main measure of comparison of different input devices is throughput. Throughput is measured in bits per second (bps) and is a quantitative measure of input device performance based on the speed and accuracy of selections.

3.1.1 Novice Group

The results for the novice group are shown in Figure 4.

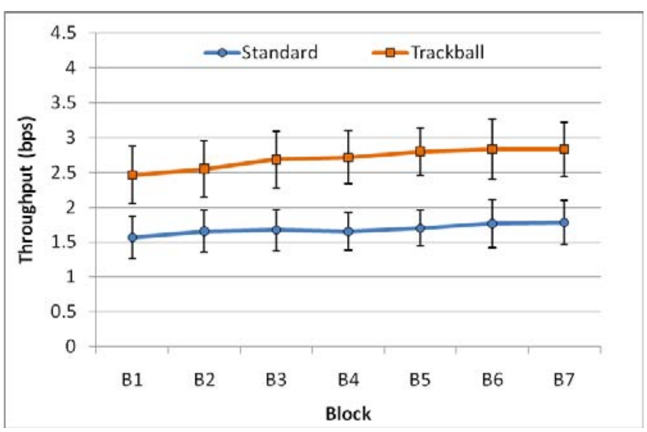

Figure 4: Novice group throughput per block

The average throughput of the standard controller across the seven blocks was 1.68 bps (*SD* = 0.07). The trackball controller's throughput was 2.69 bps (*SD* = 0.14). Based on one-way ANOVA, the difference was statistically significant ($F_{1,9} = 92.43$, $p < .0001$) and represents a 60.1% increase in throughput for users who are not familiar with either control scheme. The effect of block on throughput was also significant ($F_{6,54} = 8.39$, $p < .0001$). The effect of order group on throughput was not statistically significant ($F_{1,9} = 0.01$, ns) indicating that counterbalancing had the desired effect.

3.1.2 Advanced Group

The results of throughput by block for the advanced group are shown in Figure 5. The average throughput of the standard controller across the seven blocks was 2.01 bps (*SD* 0.08). The trackball controller's throughput was 3.19 bps (*SD* 0.19), which represents a 58.7% increase in throughput over the standard controller. The difference was statistically significant ($F_{1,9} = 134.40$, $p < .0001$). The effect of block was also significant ($F_{6,54} = 21.24$, $p < .0001$). The effect of group was not significant ($F_{1,9} = 0.84$, ns), indicating that counterbalancing was effective. When the data for the two experience levels are combined, the Experience × Controller Type interaction effect was not significant ($F_{1,18} = 1.36$, $p > .05$).

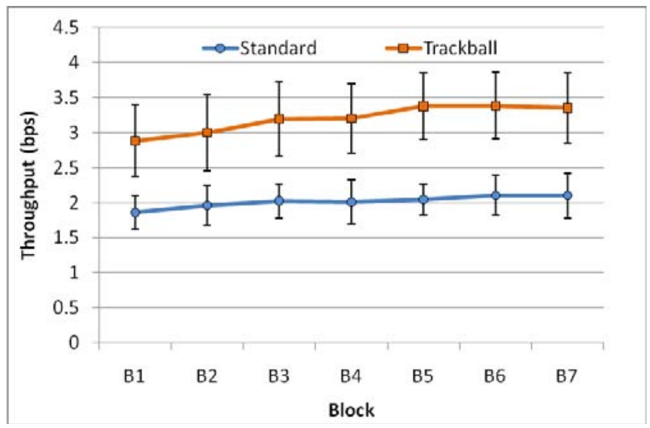

Figure 5: Advanced group throughput per block

3.1.3 Discussion

The trackball controller outperformed the standard controller in each block, for both groups. Surprisingly, even for participants who were proficient with standard game controllers, but largely novice trackball users, performance with the trackball controller was higher. These are our most important results: the trackball controller is not only significantly better (in terms of throughput), but it also requires little or no re-learning for users who are experienced with standard game controllers. Overall, novices using the trackball controller outperformed experienced players using the standard controller.

It is worth noting that the throughput for the standard controller in this study is slightly higher than the throughput measured in our earlier evaluation of game controllers [12]. The previously reported throughput of the analog stick was 1.48 bps for a group of participants with mixed expertise. In this study, throughput was 1.68 bps for novice users and 2.01 bps for advanced users, but the results are not necessarily contradictory. The difference could be explained by a number of factors, ranging from a different participant pool, to different analog stick sensitivity settings, and most importantly to a different controller. The previous study used the Nintendo *Classic Controller*, while this study used the

Microsoft *Xbox 360* controller. It is also likely that there exists a range of throughputs for analog sticks, just as there is for other pointing devices.

The throughput of the trackball controller corresponds to previously measured throughputs of trackballs, despite being operated in the hand as opposed to on a surface. MacKenzie et al., for instance, measured trackball throughput at 3.0 bps [11]. In our study, throughput for the trackball controller was 2.68 bps for the novice group, and 3.19 bps for the advanced group. With mouse gain turned on, and for users who are experienced with the trackball controller, our prototype's throughput would be higher.

3.2 Error Rates

Though throughput is a measure combining the speed and accuracy of selection, the accuracy aspect is based on the distance from the point of selection to the center of the target. It does not provide an estimate of error rates, which reflect how often targets were missed. For that reason we provide a separate report on error rates, as a percentage of the targets missed per *ID*.

For novice users the average error rate was 5.81% (*SD* = 0.57%) for the standard controller and 2.96% (*SD* = 0.44%) for the trackball controller. The difference was not statistically significant ($F_{1,9} = 3.02, p > .05$). For the advanced group the average error rate was 5.87% (*SD* = 0.80%) for the standard controller and 5.63% (*SD* = 0.91%) for the trackball controller. The results are again not statistically significant ($F_{1,9} = 0.16$, ns).

3.2.1 Discussion

Unfortunately, the results for error rates did not reveal interesting information, since the differences were not statistically significant for either group of users. However, it is interesting that with the trackball controller the novice group had an average error rate of 2.96%, while the advanced group had an average error rate of 5.63%. Though this difference was not statistically significant ($F_{1,19} = 0.11$, ns), it is still surprising since in terms of throughput, advanced users outperformed novice users. We hypothesize that novice users made less errors because they are less confident using the controllers, and consequently more careful with selection. But since the differences were not statistically significant, this cannot be confirmed.

3.3 Movement Direction Change

Throughput and error rate are excellent measures of pointing selection, in terms of how quickly and accurately the target was selected, but they do not provide information about the cursor's path. To provide a more detailed evaluation of the performance of both controllers we used additional accuracy measures of evaluation [11]. We only report the results for measures considered appropriate here.

Movement direction change (*MDC*) is a measure of discrete events that characterize the cursor's path and reports how often the cursor's movement changed direction. It is reported as a mean count per trial. In an ideal trial, the cursor's motion is in a straight line, with no direction changes, so *MDC* would be zero. In realistic usage, any direction change in the path adds to the count. In general, a lower *MDC* is better.

3.3.1 Novice Group

MDC results for the novice group are shown in Figure 6. The average *MDC* was 3.05 (*SD* = 0.11) for the standard controller, and 4.49 (*SD* = 0.31) for the trackball controller. The difference was statistically significant ($F_{1,9} = 73.01, p < .0001$).

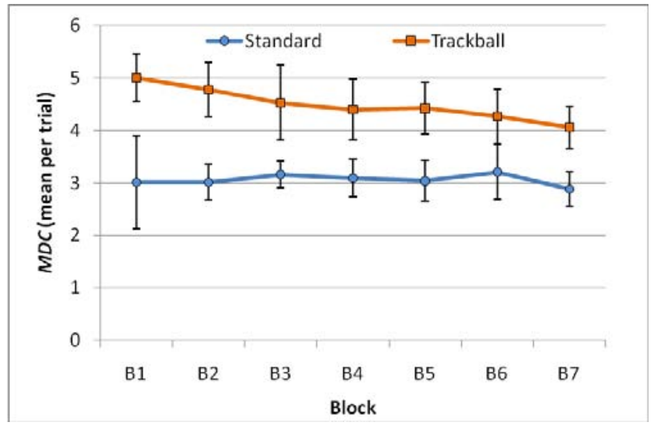

Figure 6: Novice group *MDC* per block

3.3.2 Advanced Group

MDC results for the advanced group are shown in Figure 7. The average *MDC* for the advanced group was 2.83 (*SD* = 0.04) for the standard controller and 4.14 (*SD* = 0.20) for the trackball controller. The difference was statistically significant ($F_{1,9} = 83.33, p < .0001$). The results for both groups indicate fewer direction changes with the standard controller than with the trackball controller. This is not surprising, since the analog stick specifies a direction in which the cursor moves in a straight line.

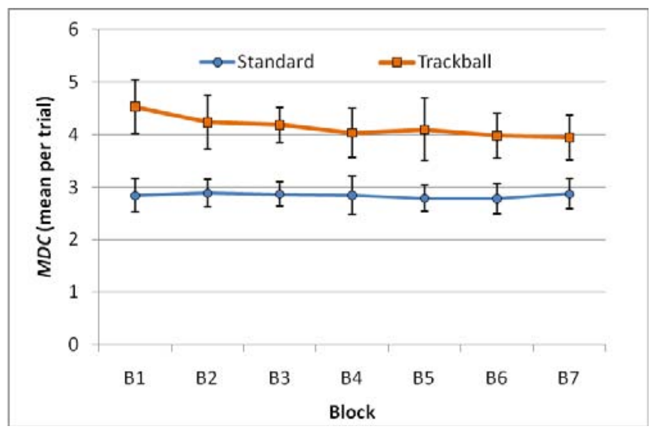

Figure 7: Advanced group *MDC* per block

3.4 Movement Error

Unlike the previous discrete measure of cursor movement, movement error (*ME*) is a continuous measure. It is computed from the coordinates of the cursor's path, and represents the average deviation of sample points from the task axis. It is an absolute value in pixels, regardless of whether sample points are above or below the task axis. Assuming the task axis has $y = 0$, then

$$ME = \frac{\sum |y_i|}{n} \tag{4}$$

A lower *ME* indicates that the sample points lie closer to the task axis, and that the movement of the cursor was more direct. For a perfect trial, *ME* is 0 px.

3.4.1 Novice Group

Figure 8 shows the results for *ME* for the novice group. The average *ME* was 13.68 px (*SD* = 0.69) with the standard controller, and 15.86 px (*SD* = 0.41) with the trackball controller. The difference was statistically significant ($F_{1,9}$ = 11.02, p < .01).

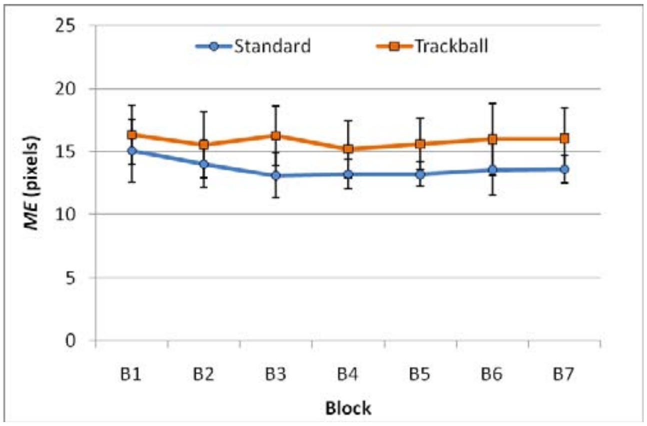

Figure 8: Novice group *ME* per block

3.4.2 Advanced Group

ME for the advanced group is shown in Figure 9. The average *ME* was 12.99 px (*SD* = 0.56) with the standard controller and 15.70 px (*SD* = 0.36) with the trackball controller. The difference was statistically significant ($F_{1,9}$ = 30.26, p < .001).

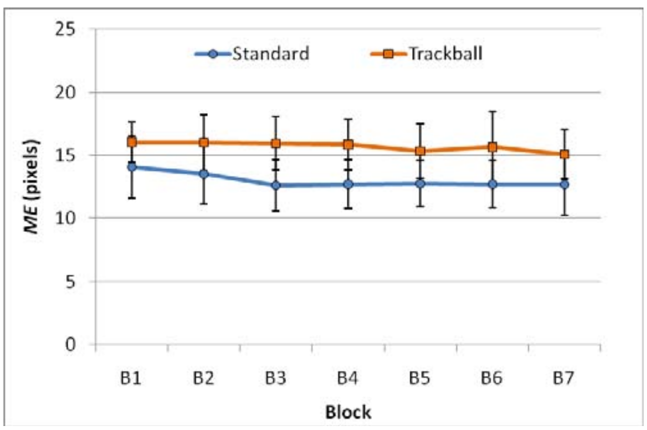

Figure 9: Advanced group *ME* per block

3.4.3 Discussion

For both experience groups, *ME* was lower with the standard controller than with the trackball controller, indicating that straight-line motion is easier with the standard controller. Again, this result is expected, since the analog stick specifies a direction. The cursor moves in a straight line in that direction, as hand jitter has less effect on the analog stick than on the trackball.

MDC and *ME* both indicate that the cursor's path is smoother with the standard controller. Though the trackball controller does not perform comparatively as well by these measures, it still offers an overall performance improvement over the standard controller. While *MDC* and *ME* are good supplementary measures for evaluating the path of motion, ultimately throughput is the more significant performance measure. Since it is possible to reach a target faster and more accurately, the path taken to the target is secondary.

3.5 Trail Plots

In addition to the analyses of the path of motion, we present example plots of the cursor's path. The *ID* and participant were selected to be illustrative. The aim of these examples is to clarify the reasons for some of the aforementioned numerical differences.

Figure 10 shows a trail plot of a novice participant using the standard controller. The trails are smooth and direct. A number of participants mentioned that they had difficulty releasing the analog stick, or "stopping" the cursor's movement at the right time. This is a result of the minimum motion speed present with the analog stick – even when the analog stick is displaced only slightly, the cursor's motion can be too fast to not overshoot a nearby target. Consequently, a number of overshoots are visible in Figure 10. The problem of minimum motion speed can be alleviated by decreasing the sensitivity of the analog stick, but this creates a new problem in the form of too low a cap on how fast the cursor moves at maximum analog stick displacement.

Figure 11 shows a trail plot of the same participant using the trackball controller. The trails are less direct. This illustrates the numerical differences with *MDC* and *ME*, between the standard and trackball controllers. In terms of *MDC*, each "spike" along the trail is considered a change of direction. Each spike also moves the cursor's path away from the task axis, which increases *ME*. It is therefore not surprising that in terms of both of these measures the standard controller outperformed the trackball controller.

A spike along the trail in Figure 11 is a "clutch", or the act of releasing the trackball and repositioning the thumb. Both figures show the *ID* with the largest target distance (512 px). Based on the plots, the number of clutches is between two and three. It is this high number of required clutches that adversely affected the performance of the trackball controller in terms of *MDC* and *ME*. It likely had an adverse effect on throughput as well.

There are a number of factors that can improve clutching. As mentioned, turning pointing acceleration (gain) on would enable a faster spin of the trackball to move the cursor farther. This would directly reduce the needed number of clutches. Additionally, placing the trackball lower in height on should improve thumb positioning when clutching. Finally, it is worth noting that participants had relatively little experience with trackballs. With experience, smaller movement deviations from a straight motion path should occur when clutching.

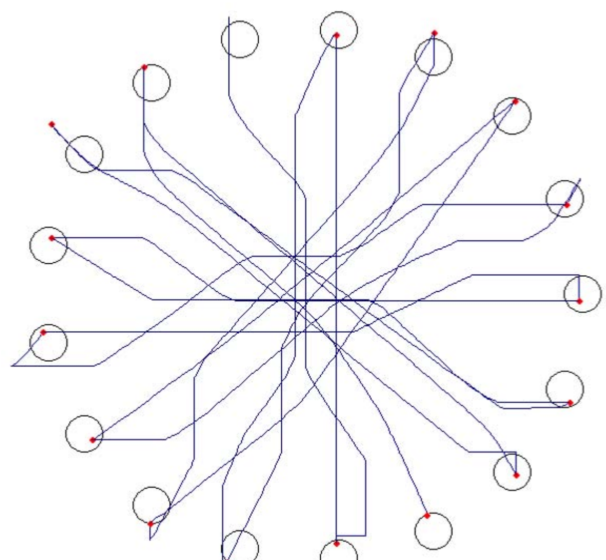

Figure 10: Sample trail plot with the standard controller

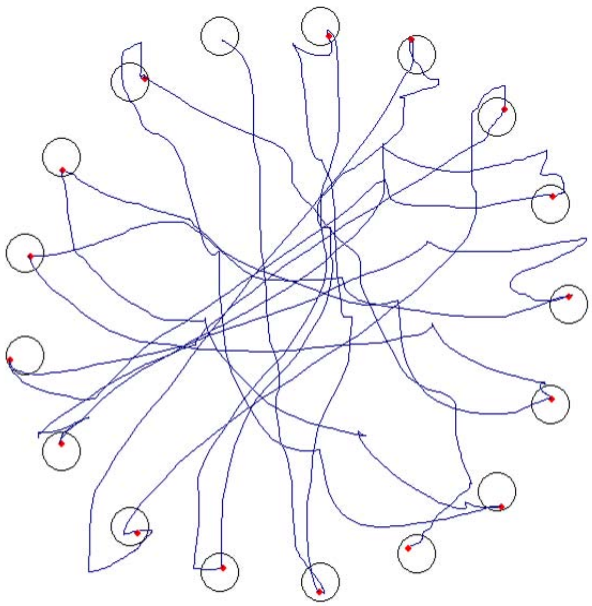

Figure 11: Sample trail plot with the trackball controller

4 CONCLUSION

In this research we presented an improvement to the control capabilities of the right analog stick used by most console controllers. We built a prototype of a new game controller replacing the right analog stick with a trackball. We validated the trackball controller in a user study, by comparing it to a standard controller in a pointing task as prescribed by ISO 9241-9.

Results were favorable. The trackball provided a 60.1% increase in throughput over the standard controller for novice participants. More impressively, it was shown to provide a 58.7% increase in throughput for participants already adept with the use of a standard controller. It was also shown that the pointer path during a trial was more direct with the standard controller than with the trackball controller. The indirect nature of the path with the trackball controller resulted, in part, from clutching the trackball. A number of factors that can alleviate the clutching problem and further improve the trackball controller's performance were discussed.

Overall, the results suggest that the trackball controller offers a significant performance improvement, for both novice and adept players, over the current standard of game controllers. Game playing is generally goal oriented and competitive, so any increase in performance is beneficial. Consequently, a performance increase as large as that observed with the trackball controller could be game changing.

ACKNOWLEDGMENTS

We would like to thank Andreea Izdraila for her editorial help. This research is sponsored by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

REFERENCES

[1] Cummings, A. H., The evolution of game controllers and control schemes and their effect on their games, 2007. http://users.ecs.soton.ac.uk/ahc08r/mms.pdf. Accessed September 2009.

[2] Douglas, S. A., Kirkpatrick, A. E., and MacKenzie, I. S., Testing pointing device performance and user assessment with the ISO 9241, Part 9 standard, *Proceedings of the ACM Conference on Human Factors in Computing Systems - CHI '99*. New York: ACM, 1999, 215-222.

[3] ESA, The Entertainment Software Association: Industry Facts, http://www.theesa.com/facts/.

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

[5] Isokoski, P. and Martin, B., Performance of input devices in FPS target acquisition, *Proceedings of the International Conference on Advances in Computer Entertainment Technology - ACE '07*. New York: ACM, 2007, 240-241.

[6] Kantowitz, H. B. and Elvers, C. G., Fitts' law with an isometric controller: Effects of order of control and control-display gain, *Journal of Motor Behavior 20*, 1988, 53-66.

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

[8] Kohler, C., Controller's History Dynamite: 1UP.com, 2005. http://www.1up.com/do/feature?cId=3143627. Accessed September 2009.

[9] Looser, J., Cockburn, A., and Savage, J., On the validity of using First-Person Shooters for Fitts' law studies, *Proceedings of the British HCI Conference*. 2005, 33-36.

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

[11] MacKenzie, I. S., Kauppinen, T., and Silfverberg, M., Accuracy measures for evaluating computer pointing devices, *Proceedings of the ACM Conference on Human Factors in Computing Systems*. New York: ACM, 2001, 9-16.

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

[13] Pfau, J., The game controller: From the beginning, 2008. http://proteus.brown.edu/13things/7643. Accessed September 2009.

[14] Purchese, R., Console RTS "still not close to PC": EuroGamer, 2008.http://www.eurogamer.net/articles/console-rts-still-not-close-to-pc. Accessed September 2009.

[15] XPAdder, http://xpadder.com. Accessed September 2009.