An Empirical Comparison of *"Wiimote"* Gun Attachments for Pointing Tasks

Victoria McArthur, Steven J. Castellucci, and I. Scott MacKenzie
Department of Computer Science and Engineering
York University
Toronto, Ontario M3J 1P3 Canada
vickymc@yorku.ca, stevenc@cse.yorku.ca, mack@cse.yorku.ca

ABSTRACT

We evaluated and compared four input methods using the Nintendo *Wii Remote* for pointing tasks. The methods used (i) the "A" button on top of the device, (ii) the "B" button on the bottom of the device, (iii) the Intec *Wii Combat Shooter* attachment and (iv) the Nintendo *Wii Zapper* attachment. Fitts' throughput for all four input methods was calculated for both button-up and button-down events. Results indicate that the throughput of the *Wii Remote* using the A button is 2.85 bps for button-down events. Performance with the Intec *Wii Combat Shooter* attachment was significantly worse than with the other input methods, likely due to the trigger mechanism. Throughput for button-down target selection using the B button was highest at 2.93 bps.

Categories and Subject Descriptors

H.5.2 [**Information interfaces and presentation**]: User Interfaces – *Input devices and strategies*

General Terms

Performance, Human Factors.

Keywords

Remote pointing, gaming input devices, ISO 9241-9, Fitts' law, performance evaluation, Wiimote.

1. INTRODUCTION

The Nintendo *Wii* was released in November 2006. Since then it has experienced unprecedented success in the global market [12]. The popularity of the *Wii* has been largely attributed to its innovative controller, the *Wii Remote*, henceforth "*Wiimote*" [10]. Rather than detecting player input through a series of button presses, the *Wiimote* encourages a more intuitive way to interact with video games, including gestures and remote pointing.

The *Wiimote* communicates with the console via Bluetooth and an infrared sensor bar placed above or below the display. The sensor bar provides two points of infrared light, which are detected by the high-speed infrared camera on the *Wiimote* and are used to determine where the device is pointing. It also sports a three-axis linear accelerometer that provides the *Wiimote* with motion-sensing capabilities [10].

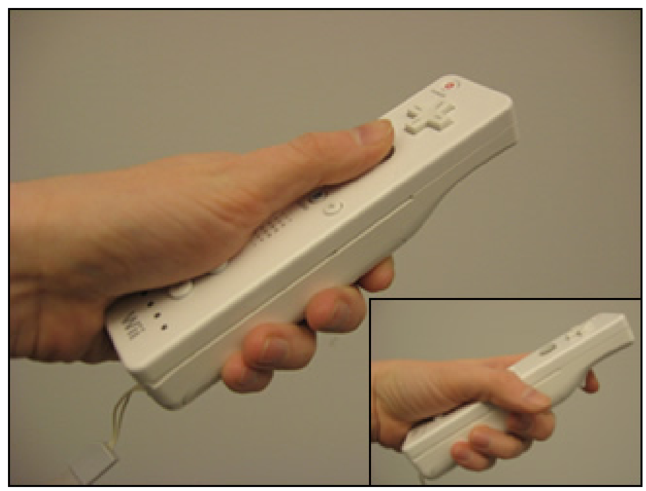

Figure 1. The Nintendo *Wiimote* using the A button and using the B button (inset).

Since its release, a number of official hardware attachments have been developed for the *Wiimote*, including the *Wii Zapper*, a plastic gun-shaped shell designed for shooter games, the *Wii Wheel*, designed for driving games, and several peripherals for use with music video games. Several third-party hardware expansions have emerged, including the *Wii Blaster* by Core Gamer, and the *Perfect Shot* by Nyko. The popularity of games using these and related peripherals is the focus of research in "neo-immersion" [14]. Neo-immersion is a new trend in game design that encourages the use of advanced immersive technologies.

The motivation for this research is to investigate whether gun-shaped peripherals help or hinder pointing performance with the *Wiimote*. This is elaborated in the next section.

2. MOTIVATION

As an input device, the *Wiimote* has eight buttons: seven on top (including a directional pad) and one trigger-like button on the bottom. Typically, the thumb rests on the "A" button, and the fingers wrap around the bottom of the device (see Figure 1). While normally held like a television remote, the *Wiimote* supports other orientations, allowing different fingers and muscle groups to interact with the device.

Zhai et al. investigated human performance differences for devices which utilize the muscle groups of the fingers for six degree-of-freedom (DOF) manipulation [16]. The authors were interested in investigating muscle groups for 6DOF devices since there was no standard for such devices. They conducted an experiment comparing the completion times of 6DOF manipulation tasks using two devices: one that included use of the fingers and one that excluded the fingers. The authors found that completion times were significantly faster with the device that employed the fingers, suggesting that future designs of 6DOF devices should include the use of these muscle groups.

Balakrishnan and MacKenzie investigated the performance differences in the fingers, wrist, and forearm [1]. The authors conducted an experiment to calculate the throughput of these muscle groups and concluded that throughput varies greatly depending on the specific motion involved. For example, the throughput of the unsupported index finger is approximately 3.0 bps (bits per second) while the throughput of the thumb and index finger working together in a pinch grip is 4.5 bps. Compared to the index finger, the throughput for the wrist and forearm are reported at 4.1 bps each. The authors concluded that whether or not the finger outperforms other muscle groups in computer input control is context-dependant.

How these different muscle groups affect performance is especially relevant when considering interaction styles and is the motivation for our study. Our study quantifies and compares the selection performance of top-mounted selection buttons and trigger-based selection techniques.

3. RELATED WORK

3.1 Remote Pointing Devices

The *Wiimote's* Bluetooth capabilities, IR camera, and accelerometer promote an intuitive interaction with games. With a remote pointing device, gamers can interact with games in new ways. Conventional games tend to focus on an avatar, through which the player approximates actions using abstract interactions mapped to buttons or joysticks [14]. The interactions with the *Wiimote* are more intuitive since a player's hand and arm motions can map to similar in-game actions. For shooter games, the player can acquire her target by aiming the device directly at the screen. In a tennis game, the player can return a serve by swinging the *Wiimote* in mid-air.

Traditionally, laser pointers were the remote pointing devices of choice in computing. Oh and Stuerzlinger presented a performance evaluation of laser pointers as input devices using the mouse as a baseline [13]. Their results showed an average throughput of 3.04 bps for the laser pointer. Myers et al. conducted a comparative study which evaluated different ways of holding the device and selecting targets [11]. A study by Jiang et al. evaluated a camera-based remote interaction technique called Direct Pointer, in which a camera is mounted on a Logitech *Cordless Presenter* [9]. The use of a camera is especially interesting in the context of this research considering that the *Wiimote* uses an infrared camera to track its position. The results showed a mean throughput of 3.21 bps.

While the high cost of these systems has been noted in the literature [13], the comparatively low price of the *Wiimote* makes it appealing as a remote pointing device.

3.2 Wiisearch

Many projects investigating alternative uses for the *Wiimote* have surfaced in the literature. This brand of research, dubbed "Wiisearch", includes a number of interesting applications. Lee reverse engineered the *Wiimote* for a variety of alternative uses including finger and object tracking, head tracking for desktop VR displays, and gesture recognition [10].

Gallo et al. created a system that used the *Wiimote* to interact with three-dimensional reconstructed organs in a virtual environment [5]. The system was implemented, but not evaluated.

An implementation by Wong et al. used gesture recognition with the *Wiimote* in musical performance [15]. Specifically, the *Wiimote* was used for conducting, percussion, and using orientation data for sound manipulation. A qualitative assessment revealed that the system was intuitive for musicians.

Castellucci and MacKenzie investigated the *Wiimote* as an input device for gesture-based character input [3]. The authors proposed a gesture-based alphabet called *UniGest* and administered a web-based empirical study to gather movement times for the primitive motions used in their proposed alphabet. Using these movement times the authors predicted an upper-bound text entry rate of 27.9 wpm using the *UniGest* alphabet with the *Wiimote*.

Castellucci and MacKenzie also evaluated the *Wiimote* as a remote pointing device following the methodology described in Part 9 of the ISO 9241 standard [2]. Using a 2D target selection task, they calculated the throughput of the *Wiimote* using infrared tracking, the *Wiimote* using accelerometer input, a gyroscopic mouse, and an optical mouse. The authors report a throughput of 2.80 bps for the *Wiimote* infrared technique. Their research studied the throughput of target selection tasks using the A button located on top of the *Wiimote.*

This paper builds on that research by evaluating the device using the B button on the bottom of the device (a "trigger"), and the A button as a baseline condition. Two additional input methods using gun-shaped attachments are also evaluated.

4. PERFORMANCE EVALUATION

To evaluate pointing device performance, we followed the methodology in Part 9 of the ISO 9241 standard for non-keyboard input devices [6]. Specifically, we undertook a user study employing two-dimensional serial target selection. Performance was quantified by the throughput of each device.

The calculation of throughput is based on Fitts' law [4] and requires the measurement of index of difficulty (*ID*) and movement time (*MT*). Movement time is the mean trial duration over a series of target selection tasks. The following calculation of index of difficulty, called the Shannon formulation, incorporates the width (*W*) and distance (*D*) of the targets selected:

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

However, the ISO standard recommends the use of effective target width (W_e). Effective target width captures the selection variability over a series of trials; it reflects how participants performed, rather than what was presented to them. To calculate W_e, the selection points are first projected onto the task axis – a line from the center of the source target to the center of the intended target. Then, the distance, x, from the projection to the center of the intended target is determined. A positive x is an overshoot, while a negative x is an undershoot. W_e for a series of trials is 4.133 times the standard deviation of x. Using W_e in the calculation of index of difficulty yields the effective index of difficulty (ID_e).

As an adjunct to effective width, effective distance (D_e) reflects the distance actually traversed along the task axis. Research by Isokoski and Raisamo revealed that also using effective distance increases throughput validity, especially for large target widths [8]. While the use of effective distance in research is sparse, we incorporated it into our calculation of effective index of difficulty, as follows:

$$ID_e = \log_2\left(\frac{D_e}{W_e}+1\right), \text{ where } W_e = 4.133 \times SD_x$$

Movement time is usually reported in seconds, while index of difficulty is in “bits”. Thus, the unit for throughput is bits per second (bps):

$$TP = \frac{ID_e}{MT}$$

Unfortunately, the ISO standard does not specify whether to evaluate throughput using mouse-down events (presses of a selection button) or mouse-up events (releases of a selection button). Research by Isokoski revealed a 4% increase in throughput when using mouse-down events versus mouse-up events [7]. To determine if the same disparity exists with the remote pointing techniques under investigation, we captured, calculated, and compared performance data associated with both selection events. To facilitate further comparison, we also investigated pointer control characteristics associated with selection events.

5. METHOD

In this section, we describe an experiment to compare the throughput of four input methods using the *Wiimote*. In addition to testing two different buttons on the device, we also tested two gun-shaped attachments. One was Nintendo’s official gun peripheral, the *Wii Zapper*. The second was Intec's *Wii Combat Shooter* – a third-party peripheral, which was chosen for the spring-loaded mechanics of its trigger.

5.1 Participants

Twelve participants, ten male and two female, volunteered for the study. The age range was 25 to 34 years (*mean* 28.1, *SD* 2.64). Of these participants, six had experience with the *Wiimote* and reported a mean average of three hours per week using the device. One participant was excluded from this calculation as an outlier, reporting 24 hours per week of use.

5.2 Apparatus

Participants were presented with 15 circular targets, arranged in a circle in the centre of the screen. Figure 2 demonstrates the 2D target selection task. Targets are highlighted one at a time and participants select the highlighted target as quickly and accurately as possible using the input device. Making a selection (whether a hit or a miss) ends the current trial and begins the next one. Successful selections are accompanied by a subtle click, while misses are indicated by an abrupt chord. (Both sounds were derived from the `C:\WINDOWS\Media` folder of a typical *Windows XP* installation.)

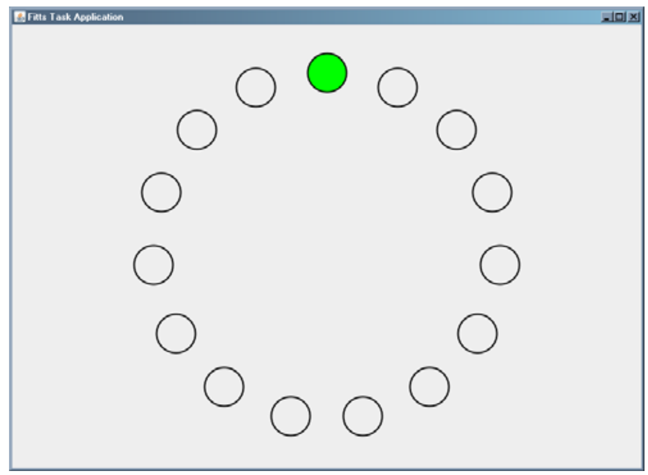

Figure 2. The ISO 9241-9 task interface.

The computer was a 3 GHz Pentium-class desktop computer with 512 MB of RAM and a separate display (see Figure 4). The system included *Windows XP Professional* with *SP3,* and the *Java Runtime Environment* (v. 1.6.0).

An InFocus projector displayed the task on a flat surface approximately 183 cm away from seated participants. The projection area was approximately 127 cm wide and 95 cm tall.

Communications between the *Wiimote* and the computer were enabled using the MSI *Star Key* Bluetooth adapter. The *GlovePie* framework running the included “IRMouse.PIE” script was used to emulate mouse input. The Nyko *Wireless Sensor Bar* was centered below the projected display and provided the infrared reference for the *Wiimote*.

Four input methods were tested: two using different buttons on the Nintendo *Wiimote* (Figure 1) and two using the *Wiimote* with gun-shaped attachments (Figure 3). These peripherals are plastic devices designed to work with the *Wiimote.* The *Wiimote* is placed directly inside the gun with the trigger positioned directly below the B button. Through mechanical linkage, pressing the trigger also presses the B button. The Intec *Wii Combat Shooter* employs a spring-trigger mechanism, whereas the Nintendo *Wii Zapper* adds no additional force effect to the trigger.

The *Wii Zapper* has two grips. When used with both hands, one hand aims and selects, while the other uses the joystick to navigate the game environment. One-handed operation of the *Wii Zapper* is typical with "on rails" games, where the player's navigation is pre-determined. Because our target selection study did not require navigation, participants were instructed to hold the *Wii Zapper* with one hand.

Figure 3. The Nintendo *Wii Remote* affixed in the Intec *Wii Combat Shooter* (top) and in the Nintendo *Wii Zapper* (bottom).

5.3 Procedure

Participants were required to complete the 2D target selection task using each of the four input methods tested. Two methods used the *Wiimote* alone: Wiimote A, using the A button on top of the device and Wiimote B, using the B button on the bottom of the device. Two additional input methods used the Intec *Wii Combat Shooter* and the Nintendo *Wii Zapper*. In all, there were four input methods.

Figure 4. A participant taking part in the experiment, using the *Combat Shooter* method.

Once the participants filled out the demographic questionnaire, the principal investigator described the target selection procedure and hardware used in the experiment. Participants were offered a trial block prior to beginning. They were also asked to hold the device in their dominant hand and to use the same hand for all four input methods. This was slightly difficult for some with the Nintendo *Wii Zapper*, since its two handles (see Figure 3, bottom) encouraged two-handed use.

5.4 Design

The experiment used a within-subjects design with the following independent variables and levels:

Input Method:	Wiimote A (*Wiimote* using the A button)
	Wiimote B (*Wiimote* using the B button)
	Zapper (*Wiimote* in the *Wii Zapper* gun case using the trigger)
	Combat Shooter (*Wiimote* in the Intec *Wii Combat Shooter* case using the trigger)
Target Width:	30, 50, 80 pixels
Target Distance:	450, 600 pixels
Trial:	1 – 15
Block:	1 – 3

We were primarily interested in the effect of input method on performance. The other independent variables were included to ensure that a reasonable amount of data was collected and that the tasks covered a representative range of difficulties.

The dependent variables were throughput and error rate. Throughput was measured in bps (bits per second) and error rate was reported as the percentage of selections outside the target. The four input methods were administered according to a balanced 4 x 4 Latin square.

6. RESULTS AND DISCUSSION

6.1 Throughput

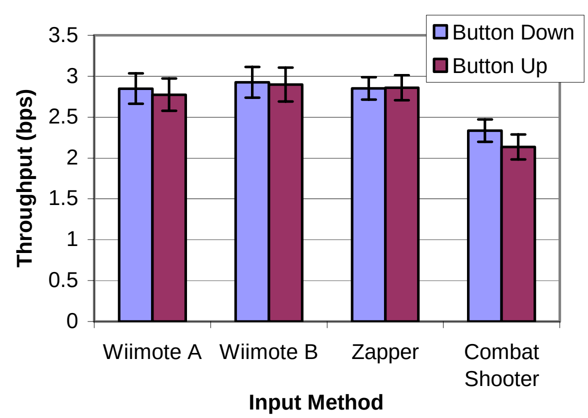

Figure 5. Throughput (bps) by input method and button down/up. Error bars show ±1 *SE*.

The grand mean for throughput was 2.70 bps. Figure 5 illustrates the throughput for each of the four input methods, calculated separately for the button-down and button-up events. For the Wiimote A, Wiimote B, Zapper, respectively, throughputs were 2.85 bps, 2.93 bps, and 2.85 bps for button-down events and 2.77 bps, 2.90 bps, and 2.86 bps for button-up events. Throughput was lower for the Combat Shooter method at 2.33 bps and 2.13 bps for the button-down and button-up events, respectively. The main effect of input method on throughput was statistically significant ($F_{3,30} = 12.5$, $p < .0001$) as was the button main effect ($F_{1,10} = 6.35$, $p < .05$) and the input method by button interaction ($F_{3,30} = 14.8$, $p < .0001$). A Scheffé multiple comparisons test revealed significant differences ($p < .05$) among comparisons involving the Combat Shooter method.

All group effects on throughput were not significant ($p > .05$) suggesting that counterbalancing was effective in offsetting learning effects.

Our throughput values for Wiimote A are very close to the 2.80 bps figure reported by Castellucci and MacKenzie in a separate experiment [2], thus validating the methodology.

6.2 Accuracy

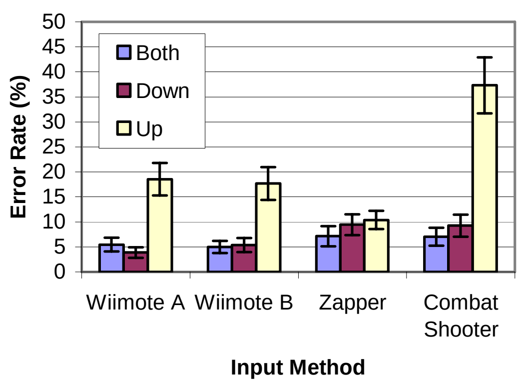

Figure 6. Error rate by input method and button down/up. Error bars show ±1 *SE*.

The error rates for button-up events were significantly higher for Wiimote A, Wiimote B, and Combat Shooter, but not for Zapper (see Figure 6). We believe the extremely high percentage of misses on the button-up event for Combat Shooter was due to the spring-loaded trigger action, which dramatically increases the force required to initiate a button-down event.

Conversely, the relatively low number of misses on button-up events for Zapper is interesting. The trigger in Nintendo's gun attachment adds no additional spring-loaded action to button events. While the feedback of a more spring-loaded action might have seemed desirable for shooter games, our results show that the additional force required and time for actuation has a negative impact on performance. This also appears to have affected the throughput when using Combat Shooter.

6.3 Participant Feedback

After the experiment, participants were asked to complete a questionnaire with 52 items (13 per input method). The items were based on questions in ISO 9241-9 [6]. Responses were coded using a 5-point Likert scale. In all cases, a high number indicated a negative response.

Overall, participant rankings of Wiimote A, Wiimote B, and Zapper were fairly moderate. Those questions in which Combat Shooter was ranked especially high (poor) are illustrated in Figure 7. Most interesting are the results for finger fatigue, physical effort required, and force required. The mean scores were 3.8, 4.2, and 4.3, respectively. We feel this is reflected in the throughput of the device and explains the extremely high error rate for button-up events in the Combat Shooter condition.

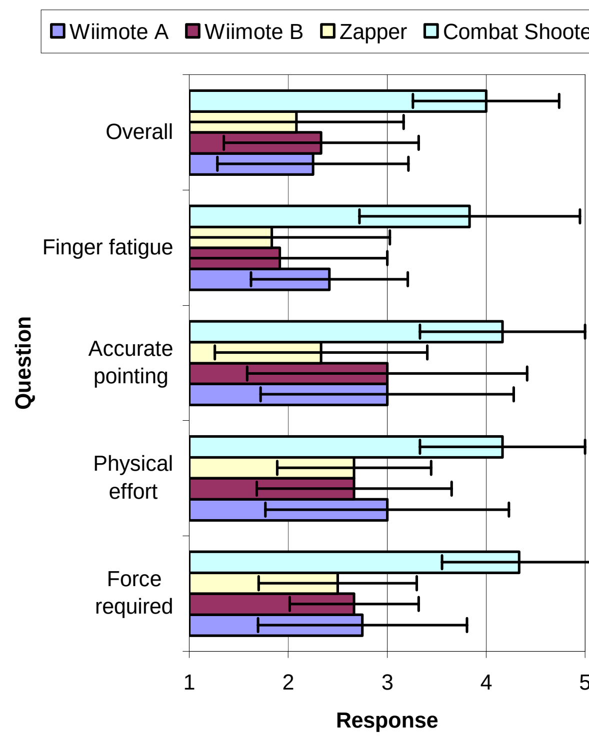

Figure 7. Feedback from participants. High numbers indicate a negative response.

During the experiment, the principal investigator noticed different strategies in the way participants aimed the devices. For example, aiming with Wiimote A and Wiimote B seemed to occur mostly at the wrist. Conversely, this motion occurred mostly at the elbow for Zapper and Combat Shooter. The shape and size of the gun used for Zapper may not lend itself to wrist-based aiming. We hypothesize that the shape of the device explains why the same behaviour was observed for Combat Shooter.

7. CONCLUSIONS AND FUTURE WORK

In this paper we evaluated and compared four input methods using the *Wiimote* for remote pointing tasks. Mean throughputs for methods Wiimote A, Wiimote B, and Zapper were 2.85 bps, 2.93 bps, and 2.85 bps, respectively, for button-down events and

2.77 bps, 2.90 bps, and 2.86 bps for button-up events. Combat Shooter was significantly worse with a mean throughput of 2.33 bps for button-down events and 2.13 bps for button-up events. We believe the disparity in performance is due in part to the spring-loaded trigger action associated with Combat Shooter. Moreover, the Combat Shooter received poor scores in the participant feedback survey.

We are planning performance studies using other *Wiimote* peripherals. In addition, we plan to conduct qualitative evaluations on Neo-immersion [14].

8. ACKNOWLEDGEMENT

We wish to thank the EICS reviewers for their constructive feedback. This research was funded by the Natural Sciences and Engineering Research Council of Canada.

9. REFERENCES

[1] Balakrishnan, R. and MacKenzie, I. S., Performance differences in the fingers, wrist, and forearm in computer input control, *Proc CHI '97*, New York: ACM, 1997, 303-310.

[2] Castellucci, S. J. and MacKenzie, I. S., Evaluation of remote pointing techniques using ISO 9241 - part 9, in *Submitted for publication*.

[3] Castellucci, S. J. and MacKenzie, I. S., Unigest: Text entry using three degrees of motion, *Proc CHI 2008,* New York: ACM, 2008, 3549-3554.

[4] Fitts, P. M., The information capacity of the human motor system in controlling the amplitude of movement, *J Exp Psychology*, *47*, 1954, 381-391.

[5] Gallo, L., Pietro, G. D., and Marra, I., 3D interaction with volumetric medical data: Experiencing the Wiimote, *Proc Int Conf Ambient Media and Systems*. Quebec, Canada: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

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

[7] Isokoski, P., Variability of throughput in pointing device tests: Button-up or button-down? *Proc NordiCHI 2006*, New York: ACM, 2006, 68-77.

[8] Isokoski, P. and Raisamo, R., Speed and accuracy of six mice, *Asian Information-Science-Life*, *2*, 2004, 131-140.

[9] Jiang, H., Ofek, E., Moraveji, N., and Shi, Y., Direct pointer: direct manipulation for large-display interaction using handheld cameras, *Proc CHI 2006,* New York: ACM, 2006, 1107-1110.

[10] Lee, J. C., Hacking the Nintendo Wii Remote, *IEEE Pervasive Computing*, *7*, 2008, 39-45.

[11] Myers, B. A., Bhatnagar, R., Nichols, J., Peck, C. H., Kong, D., Miller, R., and Long, A. C., Interacting at a distance: Measuring the performance of laser pointers and other devices, *Proc CHI 2002,* New York: ACM, 2002, 33-40.

[12] Nintendo, Consolidated Financial Highlights, 2009. http://www.nintendo.co.jp/ir/pdf/2008/080424e.pdf

[13] Oh, J.-Y., and Stuerzlinger, W., Laser pointers as collaborative pointing devices, *Proc Graphics Interface 2002*, Toronto: CIPS, 2002, 141-149.

[14] Whitson, J., Eaket, C., Greenspan, B., Tran, M. Q., and King, N., Neo-immersion: Awareness and engagement in gameplay. *Proc Future Play 2008,* New York: ACM, 2008, 220-223.

[15] Wong, E. L., Yuen, W. Y. F., and Choy, C. S. T., Designing Wii controller: A powerful musical instrument in an interactive music performance system. *Proc Int Conf Advances in Mobile Computing and Multimedia,* New York: ACM, 2008, 82-87.

[16] Zhai, S., Milgram, P., and Buxton, W., The influence of muscle groups on performance of multiple degree-of-freedom input. *Proc CHI '96,* New York: ACM, 1996, 308-315.