Be Careful How You Point That Thing: Wiimote Aiming For Large Displays

Chris Pelling

chrisp@inventivedingo.com

Torben Sko

torben.sko@gmail.com

Henry J. Gardner

henry.gardner@anu.edu.au

School of Computer Science, CECS,
The Australian National University, Canberra, ACT 0200, Australia

ABSTRACT

Previous work demonstrated that the Wii Remote (Wiimote) can be used as a control device for large displays by the use of multiple sensor bars. While this work showed the system to perform quite successfully, the limited vertical aiming range of the Wiimote was noted to be a shortcoming and, to address this issue, an accelerated aiming technique was introduced. The present work extends the study of Wiimote aiming for large displays by implementing two further techniques that consider relative movements and clutching. User testing is conducted and the results from all three techniques show that clutching performed worst while the absolute and relative techniques could not be statistically differentiated.

Author Keywords

aiming, pointing, interaction techniques, Wii remote, Wiimote, virtual reality, large displays, first-person-shooter, games, absolute, relative, clutch

ACM Classification Keywords

H.5.2 User Interfaces: Evaluation/Methodology; I.3.7 Three-Dimensional Graphics and Realism: Virtual Reality

INTRODUCTION AND MOTIVATION

Since its introduction, the Nintendo Wii games console (Nintendo, 2007) has revolutionised the video gaming industry by way of its unique input device, the Wii Remote, which boasts a suite of features previously unseen in a consumer product. This “Wiimote” includes a 3-axis accelerometer, an infrared (IR) camera, Bluetooth wireless connectivity, and rumble feedback in addition to standard buttons. The IR camera, located on the front of the controller, enables the pointing direction of the Wiimote to be tracked relative to two IR sources (each normally consisting of a cluster of IR emitters) located on either end of a “sensor bar” (this commercial term is misleading as the bar contains emitters rather than sensors) which is traditionally mounted above or below a display.

The narrow viewing range of the IR camera restricts the

aiming range of the Wiimote to half-angles from 14° to 22° which roughly provides a 40° aiming range in both the horizontal and vertical directions (Schou and Gardner, 2007). As such, the Wiimote traditionally only lends itself for use with small to medium-sized displays. When considering its use with larger or multi-screen displays, such as those in virtual reality theatres, this limitation means that only a proportion of screen space can be reached. However, due to the Wiimote's low cost, wide distribution, diverse functionality and established user base, the ability to use this device in combination with large screens is highly desirable.

Previous work (Schou and Gardner, 2007; Schou and Gardner, 2007; Sko and Gardner, 2009) presented a multiple sensor bar (MSB) system which allowed the Wiimote to be used in large display environments. Using this system with five vertically-mounted sensor bars, a user could accomplish a 180° horizontal sweep, which was sufficient for use in an immersive virtual reality theatre. In contrast, the vertical range was felt to be inadequate for that setup. As shown in Fig. 1, the vertical aiming range varied from 76% of the screen height in the centre to 55% at the left and right edges. In order to address this issue, the previous work introduced an accelerated aiming technique based on a non-linear scaling of the absolute position of the cursor relative to the Wiimote. Alternative aiming schemes were not considered at that time.

In this paper, we explore the space of Wiimote aiming schemes, and, following a process of in-house iterative development, present two new techniques together with a modified version of the original technique. The target application is a first-person-shooter game environment. In the last part of this paper, we describe the procedure and results of formal user testing which has been undertaken to evaluate and compare these three techniques.

TECHNIQUES

For this study, we implemented three distinctive techniques that are representative of the space of realistic aiming schemes (Card et al., 1990; Hinckley et al., 1994). We considered absolute (position-based) scaling, relative (velocity-based) scaling, and clutching. We motivate and explain these techniques in the following subsections.

Permission to make digital or hard copies of part or all of this work or personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee.

Q\ EJ K'09, P qx 45-49, 2009, O gndqwtpg."Cwmtcrkc

© ACM 2009 ISBN: 978-1-60558-: 76-4/09/13...\$10.00

Figure 1. A user playing a first-person-shooter game using a Wiimote in a two-walled virtual reality theatre. The white boxes overlaid on the image roughly illustrate the aiming coverage provided by each sensor bar. The centre sensor bar provides the largest amount of vertical coverage because it is furthest away from the user.

Technique A: Scaled Absolute Position

Like touchscreens and joy-sticks, the Wiimote functions as an absolute input device i.e. it reports a position within a fixed coordinate system. In the case of the Wiimote, the origin of this coordinate system is defined by the position of the sensor bar.

Because the Wiimote is an absolute device, the easiest way to interpret the IR data is to map the absolute aim of the Wiimote to an absolute pixel location on the screen. This technique was adopted in previous work where an algorithm described in (Sko and Gardner, 2009) was used to compute an unscaled cursor position from the IR data. The vertical component of the cursor's position was then scaled as follows:

$$\mathbf{cur}_i = \text{irToPixel}(\text{IR_data})$$

$$\mathbf{cur}_{i,y} = f(\mathbf{cur}_{i,y} - h/2) + h/2$$

where $\text{irToPixel}()$ maps the IR data to cursor position at the i^{th} time slice, and $\mathbf{cur}_{i,y}$ is the vertical component of that cursor position in pixels and h is the height of the screen in pixels. From the formula, the position of the cursor is scaled relative to its vertical displacement from the mid-plane of the screen. In the already-cited previous work, a non-linear mapping for f was used which resulted in a non-uniform scaling across the screen. In the present study, in-house testing was used to derive a simpler, linear scaling function: $f(\mathbf{p}_{i,y} - h/2) = 2(\mathbf{p}_{i,y} - h/2)$. Using this function, the vertical range is doubled allowing the user to reach the entire screen space but with the drawback that the scaling results in lower precision and amplifies the effects of hand-jitter as displacements increase away from the mid-plane.

Technique B: Scaled Relative Movements

For the next technique, we scaled relative movements of the Wiimote, similar to a standard mouse, rather than its absolute position. A mouse reports to the computer the distance and direction of its movement during each time

slice (e.g. 10 units to the right and 6 forward). The operating system then typically scales this value using a predefined function, known as the *velocity transfer function*, and adds the result to the current position of the cursor.

For this technique we computed the cursor position using the following formulae:

$$\mathbf{cur}_i = \text{irToPixel}(\text{IR_data})$$

$$\mathbf{cur}_{i,y} = g(\mathbf{cur}_{i,y} - \mathbf{cur}_{i-1,y}) + \mathbf{cur}_{i-1,y}$$

where $g(\mathbf{cur}_{i,y} - \mathbf{cur}_{i-1,y})$ represents the velocity transfer function which was applied to the vertical component of the cursor displacement. We undertook extensive in-house testing to choose an appropriate g . Various quadratic and other steeply curved functions were all found to be difficult to control over long distances and/or insufficiently responsive at slow speeds. Instead, we chose to adopt the velocity transfer function used by Microsoft for their operating systems: a tuned, piecewise-linear function which curbs slow movements but linearly increases faster movements (Microsoft, 2002).

Technique B does not increase the Wiimote's range directly but, instead, allows the user to effectively shift the range along the vertical axis by using gestures of differing speeds. It has the advantage over Technique A that precision can be increased, and the effects of hand-jitter reduced, at any position on the screens by using slower hand movements. A potential drawback of this technique is that the sequence of actions required to deliberately effect large shifts in cursor position is not necessarily intuitive or reproducible. For example, a movement of the cursor from the bottom to the top of the screens can be effected by a quick gesture upwards. However, if a user then slowly gestures downwards they may not be able to easily return to their original cursor position. While one author of this paper was able to use this technique very effectively, we hypothesized that this might not be true of users in general.

Technique C: Clutching

The final technique implemented *clutching* as an alternative to scaling for absolute movement. Clutching allows the user to shift the Wiimote's aiming range along the vertical axis by engaging and disengaging the mapping from the controller to the cursor position. In the "engaged state" the cursor position was computed as for Technique A but omitting the use of f . In the "disengaged state" the Wiimote could be repositioned arbitrarily without moving the cursor. This allowed the user to shift the aiming range arbitrarily to control which parts of the screen were accessible.

One question that arises with this technique is how to implement the clutch. Initially we tried a button, however in-house testing showed this approach to be non-intuitive. As an alternative, we tried using the Wiimote's accelerometer. We settled on the idea of thrusting the Wiimote forward to engage and pulling it back to disengage, drawing a natural analogy to lifting a mouse

Figure 2. Overhead view of the purpose-built testing environment (left) and a close-up view of the custom virtual environment from the player's perspective (right). The player's avatar is trapped in the raised area at the centre, surrounded by four pillars. Enemies emerge from balconies on the three surrounding buildings. The topmost balcony on each building is not used.

on and off a mouse pad. In-house testing was used to refine the amount of movement required and this testing also revealed a need for enhanced visual feedback. Consequently, a second cursor was added to the display to show the physical position at which the user was *pointing* - in contrast to the main cursor which showed where the user was *aiming*. In the engaged state, the second (pointing) cursor was made more transparent so that attention was focussed on the aiming cursor. In the disengaged state, the primary (aiming) cursor was made more transparent so that attention was focussed on the pointing cursor. This feedback served to indicate the current state to the user, to focus attention on the important cursor for each state and to display the displacement between the aiming and pointing positions.

IMPLEMENTATION AND USER TESTING

The three techniques were integrated into a system consisting of a two-walled virtual reality theatre, a Wiimote driver (Peek, 2009) using multiple-sensor-bars, and a modified version of the Source game engine which implemented the virtual world (Valve, 2009). The Source engine was further modified to implement the new techniques but other parts of the system remained the same as in the previous cited work. For the sake of consistency in the user tests, the two cursors (for aiming and pointing) were displayed for all three techniques.

Six participants were recruited from across campus to evaluate the usability of each of the three techniques. The techniques were order balanced so that each of the six possible orderings was used exactly once. Five of the participants were male, and one was female. All were between the ages of 15 and 24. They reported wide ranges of previous Wiimote experience (from "never" to "monthly") and first-person-shooter experience (from "once or twice" to "weekly"). The experiment was covered by a human ethics protocol and participants were reimbursed with a movie ticket.

For each technique, the users were instructed to shoot several waves of enemies, where each wave consisted of two enemies. As shown in Fig. 2, enemies could appear

almost anywhere on the screen, requiring the user to aim at the extremities of the screens. For any given wave, enemies attacked from a single direction (e.g. top tier, west tower), to avoid overloading the user. Effort was made to decorate the level to provide the user with clear reference points to avoid disorientation. Shadows were disabled to avoid difficulties with seeing enemies in dark areas.

Participants were given an opportunity to practice with each technique before undertaking the test with that particular technique. Each practice session lasted until the participant claimed to be comfortable with the technique. Typical practice sessions were 3-5 minutes long, although one went for 20 minutes. Each test session lasted for a maximum of five minutes or 12 waves, depending which occurred first. In practice, all participants experienced at least 8 waves per technique.

Errors and elapsed time for each test were logged, and each participant was given a post-experiment questionnaire after completing all three tests. The questionnaire was an abbreviated version of Witmer and Singer's presence questionnaire including only the questions relevant to *control factors* (Questions 1,2,3,13,26,27 and 31 of (Witmer and Singer, 1998)).

RESULTS AND DISCUSSION

An analysis of variance (ANOVA) test on the *time taken* by participants showed the choice of technique to have a significant effect ($p < 0.05$) on the completion time. No significant difference between users was seen, suggesting that the observed differences between techniques were not due to individual variation. Inspection of the data showed technique A to be the best, with a mean completion time of 166 seconds compared to 211 and 259 for techniques B and C respectively. Although the difference in means for techniques A and B appears large, this disparity is not reflected in the medians, which were 174 and 186 seconds respectively. The median for technique C was 254 seconds. Further analysis using paired Wilcoxon tests revealed a significant difference

($p < 0.05$) between the timings for techniques A and C, but not for any other pairs of techniques.

An ANOVA test on the *error counts* showed a significant effect due to choice of technique ($p < 0.05$). As with timings, participant variation was not shown to be significant. Inspection of the data showed techniques A and B to be difficult to separate with means of 6.3 and 4.2 errors and medians of 3.5 and 5 errors respectively. Technique C was much worse with a mean of 13.5 errors and a median of 14 errors. Box plots for the completion times and error rates exhibited asymmetries which is to be expected with this relatively small sample population.

Questionnaire data were coded on a Likert scale between 1 and 7 and were analysed using non-parametric Wilcoxon tests. With one exception, the scores for technique C were noticeably worse than those for techniques A and B and this difference was statistically significant for many questions. The one exception was Question 31 of (Witmer and Singer, 1998) (“Did you learn new techniques that enabled you to improve your performance?”) which received affirmative and nearly equal scores for each technique and which was probably interpreted to be more about learning than control. The scores for A and B were unable to be differentiated at reasonable confidence levels for any questions.

Although conducted with a small number of participants, the formal user tests revealed two clear conclusions. Firstly, technique C performed universally worse, and was more disliked by users, than techniques A and B. Secondly, techniques A and B were unable to be statistically differentiated on the basis of performance or user satisfaction.

The most noticeable and widespread problem with technique C was with operating the clutch. This motion-sensitive control had been adjusted to balance the competing concerns of making it easy to trigger on purpose, but difficult to set off accidentally. In practice this resulted in the worst of both worlds, as participants had obvious difficulty with triggering the control on purpose, and yet still reported occasional trouble with triggering it accidentally. Participants found it difficult to produce motions which were only just firm enough to register and frequently overcompensated, flailing the Wiimote backwards and forwards in overly violent motions. Furthermore, these motions rotated the Wiimote with respect to the sensor bars and thus caused additional unwanted cursor movements, worsening the lack of control. The implementation of clutch controls has been noted as being a problematic issue in the literature (Hinckley, 1994).

CONCLUSION

We have implemented three techniques for alleviating the Wiimote's inadequate vertical range, and have subjected

them to formal user testing. Technique A (absolute scaling), most akin to a previously used technique, was found to generally perform best but was not statistically differentiated from technique B. Technique C (clutching) was shown to be much worse than the other two, suggesting a need to refine, or avoid, the clutch control.

Our view is that the low cost and high excitement value attached to using the Wiimote motivates a further refinement of the work presented here. With some care, there is a good prospect for synthesising the various approaches in this paper into one or more techniques for using the Wiimote with large displays. Future work will investigate such a synthesis for first-person-shooter applications as well as for the use of the Wiimote as a smart pointer for “slower” applications such as presentations, walk-throughs and interactive art.

REFERENCES

- Card, S.K., Mackinlay, J.D., and Robertson, G.G. The design space of input devices. In Proc. CHI 1990, ACM Press (1990), 117-124.
- Hinkley, K., Pausch, R., Goble, J.C. and Kassell, N.F. A survey of design issues in spatial input. In Proc. UIST 1994, ACM Press (1994), 213-222.
- Microsoft Corporation, Pointer ballistics for Windows XP. [http://www.microsoft.com/whdc/archive/pointer-bal.msp, \(2002\).](http://www.microsoft.com/whdc/archive/pointer-bal.msp, (2002).)
- Nintendo of America Inc., Nintendo Wii. [http://www.nintendo.com/wii, \(2007\).](http://www.nintendo.com/wii, (2007).)
- Peek, B Managed library for Nintendo's Wiimote. [http://www.codeplex.com/WiimoteLib, \(last accessed 2009\).](http://www.codeplex.com/WiimoteLib, (last accessed 2009).)
- Schou, T and Gardner, H.J A Wii remote, a game engine, five sensor bars and a virtual reality theatre. In Proc. OzCHI 2007, ACM Press (2007), 231-234.
- Schou, T and Gardner, H.J A surround interface using the Wii controller with multiple sensor bars. In Proc. VRST 2008, ACM Press (2008), 287-288.
- Sko, T and Gardner, H. The Wiimote with multiple sensor bars: creating an affordable, virtual reality controller. In Proc. CHINZ 2009, ACM Press (2009), 41-44.
- Valve Corporation, Valve Developer Community website. [http://developer.valvesoftware.com/ \(last accessed 2009\).](http://developer.valvesoftware.com/ (last accessed 2009).)
- Witmer, B.G and Singer, M.J. Measuring presence in virtual environments: A presence questionnaire. Presence: Teleoperation and Virtual Environments, 7, 3 (1998) 225-240.