

Evaluating Visual/Motor Co-location in Fish-Tank Virtual Reality

Robert J. Teather, Robert S. Allison, Wolfgang Stuerzlinger

Department of Computer Science & Engineering
York University
Toronto, Canada
{rteather | allison | wolfgang}@cse.yorku.ca

Abstract— Virtual reality systems often co-locate the display and input (motor) spaces. Many input devices, such as the mouse, use indirect input mappings, and are disjoint from the display space. A study of visual/motor co-location was conducted to determine if there is any benefit to working directly “in” a virtual environment. Using a fish-tank VR setup, participants performed a 3D object movement task. This required moving an object from the centre of the environment to target regions, using a tracked pen, in both co-located and disjoint display/input conditions. Results were analyzed in the context of Fitts’ Law, which models rapid aimed movements. Ultimately, no significant differences were found between co-located and disjoint conditions. However, when analyzing object movement in specific directions, the co-located condition was somewhat better than the disjoint one. In particular, movement *into* the scene was faster when the display and input device were co-located rather than disjoint.

Keywords—component; Co-located input and display, virtual hand, human-computer interaction, 3D user interfaces.

I. INTRODUCTION

Virtual reality (VR) interfaces afford users a tightly-coupled loop between input to the system, and its displayed results. The VR interaction metaphor is motivated by the assumption that the more immersive and realistic the interface, the more efficiently users will interact with the system. Ideally, users will be able to leverage existing real-world motor and cognitive skills developed through a lifetime of experience and millennia of evolution, resulting in unparalleled ease-of-use.

In practice, a variety of technical issues have limited the success of this metaphor. A primary issue is that available 3D input devices tend to be expensive, and lack the speed and precision of commonly available devices such as the mouse and keyboard. Consequently, researchers have sought to develop systems that compromise between fully immersive VR systems, and desktop computers.

Fish-tank VR is one such compromise, which in its typical form, adds immersive techniques such as stereo graphics, head-tracking, and 3D tracked input devices to a desktop computer [21]. This is different from “traditional” VR, in that the visual and motor (input) spaces are completely disjoint. A common variant uses a mirror positioned above the working space,

which reflects the display to the user’s eyes. This allows their hands to work “within” the virtual environment, i.e., underneath the mirror. This additional equipment is intended to increase immersion, and allow users to directly manipulate virtual objects, as they would with real objects [1].

In contrast to such direct manipulation, the mouse necessitates decoupling of the visual space from the motor space. Mouse input is indirect and disjoint, as the hand is not co-located with the cursor, the logical position of the pointing device. Additionally, the scaling between hand and cursor motion is arbitrary and hand motion in a horizontal plane drives cursor motion in a vertical plane. To control the pointing device on the display, users must rely on proprioception (the sense of the relative positions of one’s extremities) and indirect visual feedback to infer how their hand movement maps to cursor movement. In practice, this works quite well. Considering the prevalence of the mouse-driven desktop interface, one may question the value of direct visual/motor coupling. However, pen-based computers and touch screens used with standard 2D desktop interfaces are becoming more popular. Although these devices use the same desktop interface as mouse-driven user interfaces, they clearly favour direct coupling and co-location of the display and input space.

We present a study evaluating the benefits of co-locating the input and display space in a fish-tank VR system when input-display movement planes are congruent. The study uses a 3D object movement task, modeled after the 2D pointing task commonly employed in Fitts’ Law studies. The goal of this study was to quantify the benefits of visual/motor coupling in fish-tank VR object movement.

II. RELATED WORK

We review relevant literature from 3D manipulation and VR, especially studies that compared virtual hand and disjoint manipulation interfaces. We also briefly describe Fitts’ Law.

A. 3D Manipulation

Our research relates to the mapping of input to action, i.e., the co-location of the input and display spaces. Immersive VR systems often use direct manipulation metaphors to simulate the grasping and moving of virtual objects. In other words, the user directly works in the display space. Bowman et al. provide an excellent overview of a variety of such techniques [4].

General 3D manipulation is a 6 degree-of-freedom (6DOF) task requiring three degrees of control in translation and three in rotation. Most VR systems use a 3D input device to allow simultaneous control of all 6DOF [1, 3, 11, 15, 16, 22, 23].

A common goal of VR is to create a compelling illusion of reality, wherein the user manipulates objects as in the real world. However, if immersion is not required, conventional input devices such as a mouse can suffice for 3D input [2, 5, 14]. Although the mouse controls only cursor position in the x and y directions, and is thus a 2DOF device, input mapping techniques can overcome this limitation. These techniques are largely modal, and/or based on ray-casting and constraints [4].

We have previously examined factors suspected to contribute to observed performance differences between the mouse and 3D devices [18, 19]. This included factors such as orientation of the motor space relative to the display, presence of a physical support surface, and the trade-off between input latency and tracking noise. Ultimately, no effect was found for display and input space orientation, or even physical support. However, latency had a stronger impact than spatial jitter when using the same constrained movement technique. There was about a 15% performance loss in the presence of 40 ms of additional latency, which corresponds to the measured latency difference between mouse and tracker.

B. Displacement and Frames of Reference

Previous work has examined the effects of using proprioception and haptic displays in VR object manipulation. Mine et al. [11] suggest that if objects are manipulated within arm's reach, proprioception may compensate for the absence of haptic feedback provided by virtual objects. They used a scaled-world grab which, like the Go-Go technique [15], essentially allows users to extend their virtual arm to bring remote objects close for manipulation. The rationale is that humans rarely manipulate objects at a distance, and stereopsis and head-motion parallax cues are strongest within arm's reach. They conducted a docking study comparing manipulating objects in-hand, versus at an offset distance. They found that participants were able to complete docking tasks more quickly when the manipulated object was co-located with their hand, than when it was at either a constant or variable offset distance.

Arsenault and Ware [1] conducted an experiment to determine if correctly registering the virtual object position relative to the real eye position improved performance in a tapping task. Their results indicate that this did improve performance slightly, as did haptic feedback. They thus argue for correct registration of the hand in the virtual environment. Sprague et al. [17] performed a similar study, came to different conclusions. They compared three VR conditions with varying degrees of accuracy of head-coupled registration to a real pointing task with a tracked pen. They found that while all VR conditions performed worse than reality, head registration accuracy had no effect on pointing performance.

This work suggests that people can quickly adapt to small mismatches between visual feedback and proprioception. Such plasticity has been extensively studied using the prism adaptation paradigm. In these experiments, prisms placed in

front of the eyes optically displace targets from their true position. When one reaches for these objects (or even looks at their hand) there is an initial mismatch between the visual direction of the target and its felt position [8]. However, observers quickly adapt to this distorted visual input over repeated trials effectively recalibrating the relationship between visual and proprioceptive space. Note, however, that temporal delay (latency) between the movement and the visual feedback degrades one's ability to adapt [8].

Groen and Wekhoven [7] examined this phenomenon in a virtual object docking task, performed using a head-mounted display and a tracked glove interface. They were also interested if virtual hand displacement would result in the "after-effects" reported in the prism adaptation literature. As they adapt to a visual prism displacement, participants gradually adjust (displace) their hand position to match its perceived position. If the visual displacement is eliminated the participant will continue to displace their reach resulting in an after-effect opposite to the initial error before adaptation. Such effects are temporary and participants re-adapt to the non-distorted visual-motor relationship. The authors found no significant differences in object movement/orientation time, or error rates between displaced (adapted) and aligned hand conditions. Furthermore, a small after-effect of displaced-hand was reported. This suggests users can rapidly adapt to displaced visual and motor frames of reference in VR.

Ware and Arsenault [20] also examined the effect of *rotating* the hand-centric frame of reference when performing virtual object rotations. Rotation of the frame of reference beyond 50° significantly degraded performance in the object rotation task. A second study also examined the effect of displacing the frame of reference, while simultaneously rotating it. They found that the preferred frame of reference also rotated in the direction of the translation. In other words, if the frame of reference was displaced to the left, it was also better to rotate it counter-clockwise to compensate.

C. Fitts' Law

Fitts' law [6] is an empirical model for rapid aimed movements and is given by the equation:

$$MT = a + b \cdot \log_2 \left(\frac{D}{W} + 1 \right) \quad (1)$$

where MT is movement time, D is the distance to the desired target, and W is the target width. The log term is the Index of Difficulty (ID), which is assigned a unit of bits. The coefficients a and b are determined empirically via linear regression for a given interaction technique.

Although developed for essentially 1-dimensional rapid aimed motions, Fitts' Law works well for 2D motions and is commonly used in evaluating pointing device performance [10]. However, straight-forward 3-dimensional extensions tend not work as well [12]. Alternatives models include a two-component model, which considers aimed motions as a ballistic motion, followed by a correction phase to home in on the target. Liu et al. [9] report that the control phase was about 6

times longer in VR than in reality. The reasons are not entirely clear, but reduced depth perception or tracking fidelity may be factors. Assuming that 3D extensions of Fitts' law model the ballistic but not control phase, the finding that the control phase is exaggerated in virtual reality may help explain why previous 3D extensions of Fitts' Law tend to underperform relative to similar 2D extensions.

Ultimately, the interpretation of equation 1 is that smaller, farther objects are more difficult to hit with rapid aimed motions than nearby, larger targets. Consequently, it is convenient to use *ID* rather than individual target size and distance, as it captures the overall difficulty of a movement task independent of the individual parameters. For this reason, we use *ID* in our analyses, rather than target size or movement distance. Equation 1 is commonly referred to as the Shannon formulation of Fitts' law [10].

III. METHOD

We conducted an experiment to quantify the benefits of co-locating the input and display space in fish tank VR.

A. Participants

Twelve volunteers took part in the study. Their ages ranged from 22 to 34 years, with a mean age of 27.3 years. Nine were male. All were students of York University, and were recruited by in-person request. They reported using a computer for an average of around 44 hours per week.

B. Apparatus

The experiment was conducted on a PC running at 3 GHz, with 1 GB of RAM and an NVidia *QuadroFX 3400* graphics card. We used a 17" CRT display at a resolution of 800 x 600 pixels and a 120 Hz screen refresh rate (60 Hz per eye). Stereoscopic graphics were enabled via *Stereographics CrystalEyes* shutter glasses and emitter.

1) Tracking System

We used NaturalPoint's *Optitrack*, a camera-based, optical 3D tracking system [13]. The cameras perform an on-board threshold operation on captured images, reducing transmission demands and processing requirements on the host system. This results in high update rates (120 Hz). Our setup used three *OptiTrack Flex:C120* cameras mounted on a rigid metal frame, shown in Figure 1.

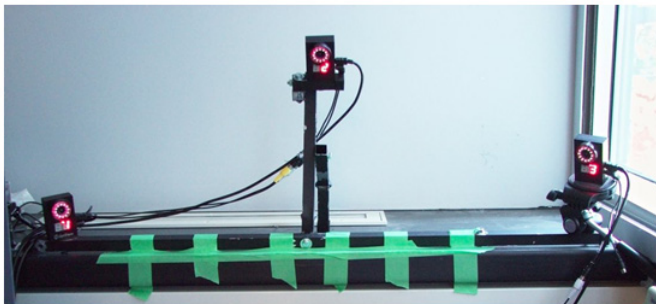


Figure 1. NaturalPoint *OptiTrack* cameras mounted on the metal frame.



Figure 2. Tracked pen used. The button is under the thumb in this figure.

The cameras also contain infrared illuminators that emit infrared light, which is reflected off retro-reflective markers. After calibrating the tracker, the NaturalPoint *Rigid Body Toolkit* allows real-time 6DOF motion capture of rigid bodies within the overlapping fields-of-view of the cameras. In our experiment, the rigid body consisted of six markers on a tracked frame. The cameras were positioned to cover the working area from multiple angles.

2) The Input Device

The tracked frame was rigidly attached onto a 10 cm pen chassis, which had a single button on it. This was connected to the computer via PS/2, providing a left-mouse button click event whenever the button was pressed. The tracked pen device can be seen in Figure 2.

3) Screen Setup

Since the tracking system requires line of sight to the tracked object, tracking on the surface of an upright monitor was difficult due to space constraints. We instead tracked the area above a monitor positioned lower than the table, with the screen facing the ceiling. This monitor was mounted on a wheeled cart, and secured at an angle of about 15° to allow for easier viewing when seated in front of it. The cart was used to move the monitor from underneath to beside the tracked space, to switch between the co-located and disjoint display conditions, respectively.

A transparent plastic panel was securely mounted to the bottom of the table, such that it covered the monitor in the co-located display/input condition. This served as the working space; the tracking system was calibrated to track the pen over this panel. Figure 3 depicts these two conditions.

4) Software Setup

Custom C++ code using the OpenGL graphics API presented a simple immersive virtual environment, depicting the inside of a wireframe box, see Figure 4. The software used quad-buffering and off-axis frustums to provide a stereoscopic display. It used the NaturalPoint API to track the pen.

On each trial a red sphere was rendered in the centre of the environment and a target wireframe sphere was rendered at another location, which depended on the condition. A white sphere depicted the virtual tip of the tracked pen. This was

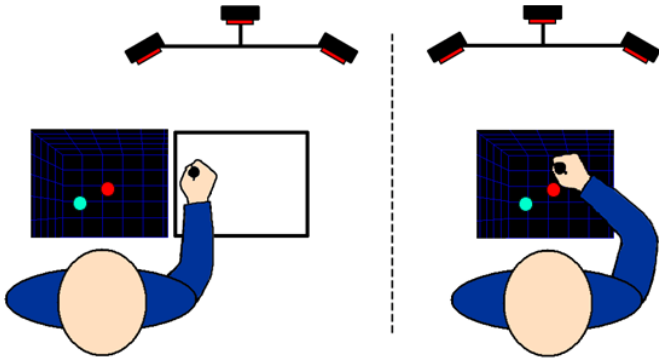


Figure 3. The screen setup, with the tracker. The left image depicts the disjoint condition. The pen was tracked over the plastic panel. In the co-located condition (right), the screen was positioned under the plastic panel.

positioned on a ray extending 10 cm into the scene from the tip of the physical pen, i.e., 20 cm from the tracker. We choose this displacement to avoid potential occlusion issues between the pen tip and the 3D position. The red sphere was semi-transparent to aid in seeing objects behind it, and also to have clear feedback when the pen tip intersected it [23].

We logged movement time and the number of pen button clicks outside of the red sphere (misses).

C. Procedure

Participants were seated in front of the display. They were first instructed in the purpose of the experiment, and the use of the system. They were then given several practice trials to familiarize themselves with the object movement.

Following the training period, participants were asked to move the red sphere to the blue target sphere, as quickly as possible. This was done in both the co-located and disjoint display conditions. To select the red sphere, participants would intersect the white virtual pen tip with it and click the pen button. The red sphere would then move and rotate with the pen tip. Intersecting it with the blue wireframe sphere completed a given movement trial.

Upon completing all object movements in one condition, the monitor was wheeled to the alternate position, and the next block began. Upon completing the experiment, participants filled out a short questionnaire.

D. Design

The experiment used the following independent variables and levels:

<i>Display Mode:</i>	co-located with input space, or disjoint from input space
<i>Target Direction:</i>	26 vectors from centre
<i>Target Distance:</i>	3, 5 and 7 cm
<i>Target Size:</i>	1.00, 1.50 and 2.00 cm

The target directions were determined as all combinations of positive, negative, or no movement along each of the x , y and z axis, excluding the centre position itself. Each target position

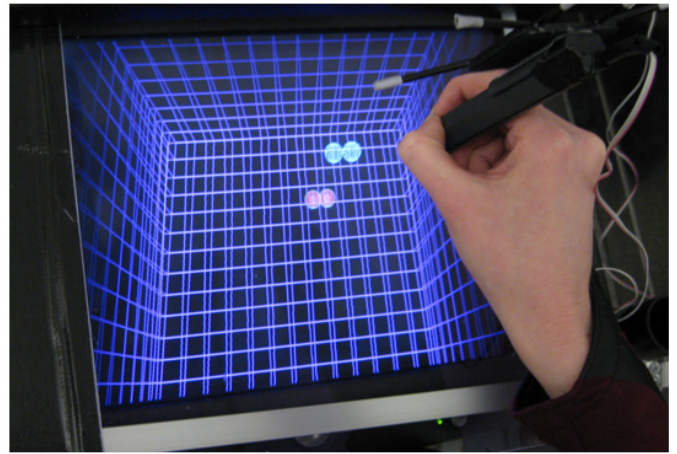


Figure 4. Software used in the experiment. The white “pen tip” sphere is currently inside the object, which the user is moving toward the target sphere.

was given by the combination of the direction vector and target distance for the given trial. Consequently, the design of the experiment was $2 \times 26 \times 3 \times 3$, for a total of 468 object movements per participant. In total, it took approximately 30 minutes to complete all trials.

The nine combinations of target radius and movement distance gave nine indices of difficulty, ranging from 1.32 to 3.0. These were computed using the Shannon formulation [10], and are summarized in Table 1.

Table 1. Summary of IDs by target radius and distance.

	Target Diameter (cm)		
Target Distance (cm)	2.00	1.50	1.00
3	1.32	1.58	2.00
5	1.81	2.11	2.58
7	2.16	2.50	3.00

Target direction, distance and radius were ordered randomly within a block, without replacement. Half of the participants performed the task in the co-located condition first; the rest started with the disjoint condition, to complete the counterbalancing.

The dependent variables for the experiment were movement time (ms) and error rate. Error rate was measured as the click events that missed the red sphere.

IV. RESULTS AND DISCUSSION

A. Movement Time

We were primarily interested in how fast participants were able to complete the object movement tasks. The mean movement time was 973.89 ms. Results were analyzed with repeated measures ANOVA. The mean time for the co-located display condition was 948.58 ms, and 1004.14 ms for the disjoint condition. These were not significantly different ($F_{1,11} = 1.27$, $p = 0.28$). The statistical power, however, was quite low (0.18), likely due to the great amount of variability due to directional effects, discussed below.

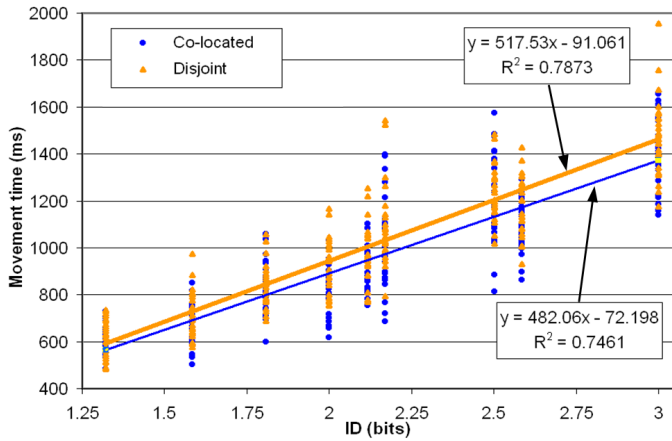


Figure 5. Movement time as a function of ID .

We also analyzed movement times as a function of ID . The relatively low correlation coefficient ($R^2 \sim 0.75$) indicates that the Shannon formulation of ID is a poor fit for 3D pointing, as suggested in previous research [9, 12]. Consequently, the predictive capabilities of this model are limited. Figure 5 depicts the regression analysis of movement time on ID .

1) Movement Direction

Movement direction was found to have a significant effect on movement time ($F_{25,275} = 2.70$, $p = .00004$, power ≈ 1). On average it took longer (1024 ms vs. 947 ms) to move downward along the y -axis – or in other words, *into* the scene – than in other directions. However, no such effect was found for either the x axis ($F_{2,22} = 0.01$, ns) or z axis ($F_{2,22} = 0.88$, ns).

A significant interaction effect was found between display mode and direction ($F_{25,275} = 2.12$, $p = .0166$, power = 0.99). Movement down/into the scene was generally slower in the disjoint condition than in the co-located condition. Three other conditions involving left/right movements took significantly longer in the disjoint condition, especially when combined with z motions. These can be seen in Figure 6.

Movement down into the scene was generally found to take longer than any other direction, see Figure 6. Even movement up out of the scene was not significantly worse than lateral movement. This is likely due to lack of good depth cues when the target was located below the starting position of the object. Participants could rely only on occlusion, stereo and

perspective to judge the distance of 3D objects. Although the red sphere was partially transparent, it still mostly occluded the targets that appeared below it. Thus participants may not have immediately noticed the position of the target. Previous research has also confirmed that it is more difficult to translate objects along a ray's direction [11].

B. Error Rate

We looked at the difference in error rate between the two display modes. Given that each trial ended upon intersecting the red sphere with the target sphere, it was impossible to “miss” during movement. Hence, only initial clicks to select the red sphere are considered here.

In total, participants missed the red sphere 763 times in the co-located display mode, and 705 times in the disjoint display mode. These were not significantly different ($F_{1,22} = 0.07$, ns). These correspond to about 25% of all trials. The high error rate was likely due to the poor depth cues, and the fact that multiple misses were possible per trial.

Once the object was selected, a trial could only end with successful intersection. This is likely partially responsible for the poor fit of the regression lines in Figure 5. If a participant missed the target sphere, they were required to simply home back to it until they successfully hit it. Consequently, the actual movement most likely did suffer from the same lengthy correction phases observed in previous work [9].

C. Overall Discussion

Object motion appeared to benefit very little from display/input space co-location. As can be seen in Figure 6, most movements took slightly longer in the disjoint condition. However, most differences are less than 10%, and, on the whole, were not significant. The exceptions are three conditions with no y movement: movement left and forward, right and forward, and right and back, which took longer in the disjoint condition than the co-located condition. This was most likely caused partially by the distance the participants had to reach during the disjoint condition. A second factor was that the limited depth cues made it difficult to tell if the blue target sphere was in the same plane as the red sphere. Differing target sizes could be mistaken for perspective effects as smaller targets in the plane can be confused for distant targets. Co-location appeared to somewhat help with this.

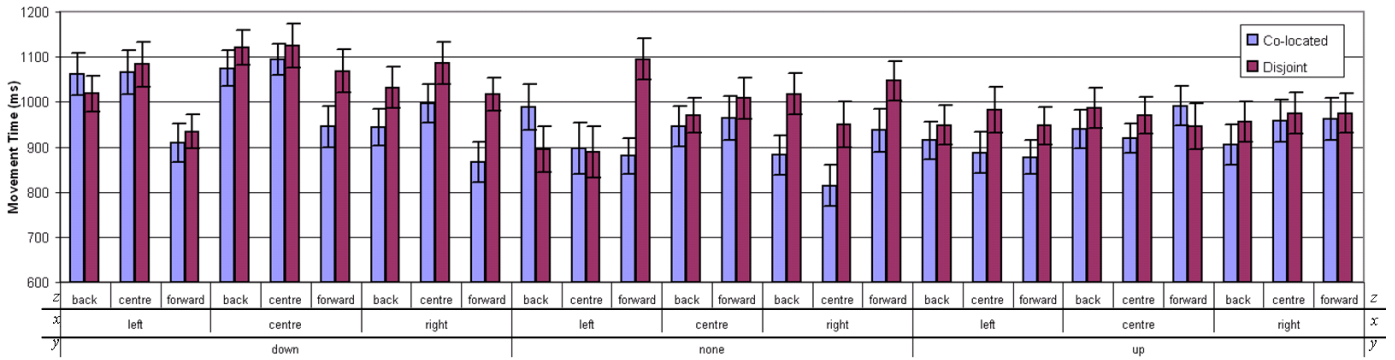


Figure 6. Movement times by each movement direction. Error bars represent ± 1 standard error.

A possible explanation of these results is that sensorimotor adaptation reduced the impact of the disjoint condition. Participants were able to perform roughly the same whether working on or off the display space. However, it appears that certain movements may have been more difficult in the disjoint condition.

V. CONCLUSIONS

We conducted a study of display/input space co-location in a fish-tank VR setup. Participants were asked to move objects using a tracked pen from a centre position to targets that appeared around it in 3D. Although we analyzed the data via Fitts' Law, the variability limits the model as a precise predictor for 3D motions.

Results indicate that co-locating the display and input spaces had little effect on user performance, except in specific cases. Movement into the scene was significantly worse overall than other directions of movement. However, co-location helped somewhat. Movement in depth was slightly easier with the display and input co-located, as were cases in which the depth of the targets was ambiguous. Consequently, there may be some value to the co-location typically used in VR systems.

A. Future Work

We would like to further study pointing and reaching in virtual reality systems. In particular, some conditions appeared to suffer from confusion between perspective and target size. This could be examined by holding target size constant, while varying depth of targets (and hence their perspective distorted perceived size). Similarly, movement in depth appears to warrant further study as well.

Finally, unlike a typical Fitts' Law tapping task, our task did not allow for participants to *miss* a target. Instead, they were required to home in on the target until they hit it to end the trial. This makes it impossible to study error rates, which are also of interest when evaluating the efficiency of pointing devices. We would like to add this capability to the system.

ACKNOWLEDGMENTS

Thanks to Andriy Pavlovych for assistance calibrating the display, Vicky McArthur for help with the figures, and the participants for helping out with the study. Thanks also to NSERC for supporting this research.

REFERENCES

- [1] R. Arsenault and C. Ware, "Eye-hand co-ordination with force feedback," in Proceedings of the ACM Conference on Human Factors in Computing Systems - *CHI 2000*. New York: ACM, 2000, pp. 408-414.
- [2] E. Bier, "Skitters and jacks: interactive 3D positioning tools," in Proceedings of the 1986 Workshop on Interactive 3D Graphics. New York: ACM, 1987, pp. 183-196.
- [3] D. Bowman, A. D. Johnson, B., and L. Hodges, F. , "Testbed evaluation of virtual environment interaction techniques," in Proceedings of the ACM Symposium on Virtual Reality Software and Technology - *VRST '99*, New York: ACM, 1999, pp. 26-33.
- [4] D. Bowman, A. , E. Kruijff, J. LaViola, J. , and I. Poupyrev, 3D User Interfaces: Theory and Practice: Addison Wesley Longman Publishing Co., Inc., 2004.
- [5] B. D. Conner, S. Snibbe, S. , K. Herndon, P, D. Robbins, C. , R. Zeleznik, C. , and D. van, Andries "Three-dimensional widgets," in Proceedings of the 1992 Symposium on Interactive 3D Graphics. New York: ACM, 1992, pp. 183-188.
- [6] P. M. Fitts, "The information capacity of the human motor system in controlling the amplitude of movement," *Journal of Experimental Psychology*, vol. 47, 1954, pp. 381-391.
- [7] J. Groen and P. J. Werkhoven, "Visuomotor adaptation to virtual hand position in interactive virtual environments," *Presence: Teleoperators & Virtual Environments*, vol. 7, 1998, pp. 429-446.
- [8] R. Held, A. Efstathiou, and M. Greene, "Adaptation to displaced and delayed visual feedback from the hand," *Journal of Experimental Psychology*, vol. 72, 1966, pp. 887 - 891.
- [9] L. Liu, R. v. Liere, C. Nieuwenhuizen, and J.-B. Martens, "Comparing aimed movements in the real world and in virtual reality," in IEEE Virtual Reality Conference - *VR '09*. New York: IEEE, 2009, pp. 219-222.
- [10] I. S. MacKenzie, "Fitts' law as a research and design tool in human-computer interaction," *Human-Computer Interaction*, vol. 7, 1992, pp. 91-139.
- [11] M. R. Mine, J. Frederick P. Brooks, and C. H. Sequin, "Moving objects in space: exploiting proprioception in virtual-environment interaction," in Proceedings of Computer Graphics and Interactive Techniques. New York: ACM, 1997, pp. 19-26.
- [12] A. Murata and H. Iwase, "Extending Fitts' law to a three-dimensional pointing task," *Human Movement Science*, vol. 20, 2001, pp. 791-805.
- [13] NaturalPoint, Inc., "NaturalPoint OptiTrack.": Available at www.naturalpoint.com/optitrack, 2008.
- [14] J.-Y. Oh and W. Stuerzlinger, "Moving objects with 2D input devices in CAD systems and desktop virtual environments," in Proceedings of Graphics Interface 2005. Toronto: CIPS, 2005, pp. 195-202.
- [15] I. Poupyrev, M. Billinghurst, S. Weghorst, and T. Ichikawa, "The go-go interaction technique: non-linear mapping for direct manipulation in VR," in Proceedings of the ACM Symposium on User Interface Software and Technology - *UIST '96*. New York: ACM, 1996, pp. 79-80.
- [16] I. Poupyrev, Ichikawa T., S. Weghorst, and M. Billinghurst, "Egocentric object manipulation in virtual environments: Empirical evaluation of interaction techniques," in Proceedings of Eurographics '98. vol. 17, 1998, pp. 41-52.
- [17] D. W. Sprague, B. A. Po, and K. S. Booth, "The importance of accurate VR head registration on skilled motor performance," in Proceedings of Graphics Interface 2006. Toronto: CIPS, 2006, pp. 131-137.
- [18] R. J. Teather, A. Pavlovych, W. Stuerzlinger, and I. S. MacKenzie, "Effects of tracking technology, latency, and spatial jitter on object movement," in IEEE Symposium on 3D User Interfaces - *3DUI '09*. New York: IEEE, 2009, pp. 43-50.
- [19] R. J. Teather and W. Stuerzlinger, "Assessing the effects of orientation and device on (constrained) 3D movement techniques," in IEEE Symposium on 3D User Interfaces - *3DUI '08*. New York: IEEE, 2008, pp. 43-50.
- [20] C. Ware and R. Arsenault, "Frames of reference in virtual object rotation," in Proceedings of the 1st Symposium on Applied Perception in Graphics and Visualization. New York: ACM, 2004, pp. 135-141.
- [21] C. Ware, K. Arthur, and K. S. Booth, "Fish tank virtual reality," in Proceedings of the ACM Conference on Human Factors in Computing Systems - *CHI '93*. New York: ACM, 1993, pp. 37-42.
- [22] C. Ware and D. Jessome, R, "Using the bat: A six-dimensional mouse for object placement," *IEEE Computer Graphics and Applications*, vol. 8, 1988, pp. 65-70.
- [23] S. Zhai, W. Buxton, and P. Milgram, "The "silk cursor": Investigating transparency for 3D target acquisition," in ACM Conference on Human Factors in Computing Systems - *CHI '94*. New York: ACM, 1994, pp. 459 - 464.