

**Modeling Three-Dimensional Interaction Tasks
for Desktop Virtual Reality**

Copyright ©2011 by Lei Liu

All rights reserved. No part of this book may be reproduced, stored in a database or retrieval system, or published, in any form or in any way, electronically, mechanically, by print, photoprint, microfilm or any other means without prior written permission of the author. A catalogue record is available from the Eindhoven University of Technology (TU/e) Library.

ISBN: 978-90-386-2839-4

Cover design by Paul Verspaget & Lei Liu, inspired by Matrix Digital Rain.

Printed by TU/e Printservice.



Centrum Wiskunde & Informatica



Netherlands Organisation for Scientific Research

This research was supported by the Netherlands Organization for Scientific Research (NWO) under project number 600.643.100.05N08. Title: Quantitative Design of Spatial Interaction Techniques for Desktop Mixed-Reality Environments (QUASID). The work reported in this thesis was carried out at the Centrum Wiskunde & Informatica (CWI), the Dutch national research institute for Mathematics and Computer Science, within the theme Visualization & 3D User Interfaces (INS3) and Software Analysis & Transformation (SEN1).

Modeling Three-Dimensional Interaction Tasks for Desktop Virtual Reality

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
rector magnificus, prof.dr.ir. C.J. van Duijn, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen
op maandag 28 november 2011 om 16.00 uur

door

Lei Liu

geboren te Jilin, China

Dit proefschrift is goedgekeurd door de promotor:

prof.dr.ir. R. van Liere

献给我亲爱的爸爸妈妈

Contents

Preface	xi
1 Introduction	1
1.1 Motivation	1
1.1.1 3D Interaction	2
1.1.2 Interaction Models	3
1.2 Objective	4
1.3 Approach	4
1.4 Scope	6
1.5 Contributions	6
1.6 Thesis Outline	6
1.7 Publications from This Thesis	7
2 Modeling Interaction Tasks: An Overview	9
2.1 Modeling Pointing	9
2.1.1 Fitts' Law	10
2.1.2 Application of Fitts' Law	11
2.1.3 The Two-Component Model	13
2.2 Modeling Steering	14
2.2.1 The Steering Law	14
2.2.2 Application of the Steering Law	16
2.3 Object Pursuit	17
2.4 Differences between Pointing, Steering and Object Pursuit	18
2.5 Stevens' Power Law	19
3 Pointing	21
3.1 Introduction	21
3.2 Comparing 3D Pointing Tasks in the Real World and in Virtual Reality	21
3.2.1 Experiment	22
3.2.2 Results	25
3.3 Designing 3D Selection Techniques	32
3.3.1 Interaction Technique Implementation	33
3.3.2 Experiment	34
3.3.3 Results	35

3.4	Discussion	38
3.5	Conclusions	39
4	Steering	41
4.1	Introduction	41
4.2	Ball-and-Tunnel Task	42
4.3	Path Curvature and Orientation	44
4.3.1	Experiments	45
4.3.2	Results	49
4.4	Varying Path Properties	59
4.4.1	Task Design	60
4.4.2	Experiment	62
4.4.3	Results	62
4.5	Long Path Steering	66
4.5.1	Experiment	66
4.5.2	Results	67
4.6	Haptic Path Steering	69
4.6.1	Types of Force Feedback	70
4.6.2	Experiments	71
4.6.3	Results	73
4.7	Discussion	82
4.7.1	Steering Models	82
4.7.2	Steering Movements	87
4.7.3	Learning Effect Analysis	90
4.8	Conclusions	90
5	Object Pursuit	93
5.1	Introduction	93
5.2	Object Pursuit Task	93
5.3	Experiment	94
5.3.1	Subjects	94
5.3.2	Procedure	95
5.4	Results	95
5.5	Discussion	100
5.6	Conclusions	102
6	Conclusions and Future Work	103
6.1	Conclusions	103
6.2	Future Work	105
	Appendices	107
	Appendix A Data Processing and Model Analysis	109
A.1	Data Transformation	109
A.1.1	Approach and Rationale	109
A.2	Confidence Intervals for Repeated-Measures Designs	112
	Appendix B Path Steering as Goal-Crossing Movements	113

Appendix C 3D Movement Parsing	115
C.1 Non-Real-Time 3D Movement Parsing	115
C.2 Real-Time 3D Movement Parsing	116
C.2.1 Parsing Criteria Evaluation	118
Appendix D Velocity Profiles in Long Path Steering	119
Bibliography	123
Summary	135
Curriculum Vitae	137

Preface

I am a bundle of contradictions. Ever since I was a child, I have demonstrated a singing talent and wished to become a pop singer. Ironically, a few days before my 30th birthday, I am about to finish writing my Ph.D. thesis on virtual reality. The moment you bring my motivation into question, you should probably start to read my thesis. Deep in my heart, I always believe that it is the scientists who will eventually “rock” the world.

This thesis is the result of my four-year Ph.D. research that was carried out at CWI, the Dutch national research institute for the mathematics and computer science, under the supervision of Prof. dr. ir. Robert van Liere. It is a succession of my two-year master program at VU University Amsterdam which began my Amsterdam life. Looking backwards, I am quite grateful and lucky for having so many people around, without whom the thesis would not have been possible. It is my pleasure to take this opportunity to express my appreciation to them.

First and foremost, I owe my deepest gratitude to my supervisor and promotor Robert van Liere, who introduced me to the academic world with his invaluable guidance, profound insights and rigorous scholarship. He portrayed what a real scientist is like. Throughout my Ph.D. research and thesis writing, he provided considerable encouragement, inspiration and enthusiasm. In particular, I was taught to develop the ability to stand back and look at things from a higher level, and not to overemphasize the details at the price of missing a bigger picture. I would have been lost without his supervision.

I am indebted to Jean-Bernard Martens for his knowledge and assistance with statistics and modeling. As a co-author, he always brought a large number of contributions, which substantially enhanced the quality of our papers. I also humbly acknowledge the constructive advice and challenging questions from Bernd Fröhlich, following my presentation during every international conference. Special thanks go to Jack van Wijk and Pieter Jan Stappers, who as reading committee members made many perceptive comments on this thesis.

I would also like to thank my colleagues of INS3, Chris Kruszyński and Ferdi Smit for offering their abundant technical support and sharing interesting stories and life experiences. I greatly appreciate and wish to thank all participants, who were willing to voluntarily act as the “lab rats” of the 9 experiments involved in this thesis, including Alexander, Arjen, Bei, Bo, Chao, Chris, David, Eefje, Eleftherios, Fabian, Fangbin, Fangyong, Fengkui, Ferdi, Fujin, Hairong, Holger, Irish, Jian (Fu), Jian (Shi), Jun, Liying, Longyuan, Marco, Mengxiao, Nan, Qiang, Sara, Shan, Si, Stephan, Thijs, Theo, Xiang, Xirong, Xu, Xue, Yanjing, Ying, Yinqin.

In addition to the direct contributors to this thesis, I would like to show my heartfelt gratitude to my circle of friends, for their company, understanding and support. Si Yin and Fujin Zhou have walked me through my 6-year Amsterdam life from the very beginning. We witnessed how each of us progressed and shared tears and laughter. I guess it is just

impossible to “kick” me out of their life. Chao Li and Bei Li, who respectively represent each side of me, made me believe “scientist” and “singer” could “negotiate”. Wei Li, though on the other side of the ocean, has never stopped comforting me while I am down and “bothering” me with her complain of overwork. In return, I never missed the chance to “bother” back. Nan Tang always made “unexpected” phone calls from Scotland, giving an illusion that he was still in Amsterdam. Ying Zhang, whose Dutch is as good as a native speaker, was usually forced to be my exclusive Dutch-Chinese translator. Mariya Mouline, a smart Amsterdamer with an “overabundance” of energy and an entrepreneurial mindset, has played a role of my Amsterdam life mentor since the very moment I landed at Schiphol Airport. Rene de Vries and Marja Zeegers, probably the best landlords I can expect, always treated me to the most native Dutch food and culture. I would also like to thank other party- and karaoke-mates from the Chinese community: Guowen, Gurong, Huiye, Jianan, Jing (Xu), Jing (Zhao), Leimeng, Ling (Shan), Ling (Zhang), Meng, Ming, Peng, Ronald, Weiqiang, Xi, Xu, Yang, You, Yuan, Zhen, etc., as you brought me so many fond memories and laughter. Thank you for your appearance in my Amsterdam life and hopefully you feel the same.

Most importantly, before I conclude my acknowledgements, I wish to express my greatest love and gratitude to my beloved parents, Hongxuan Liu and Yafan Li, for their unconditional and endless dedication over the years. My family is always the most invaluable treasure in my life.

Lei Liu 刘磊

Amsterdam, 5th October 2011

Chapter 1

Introduction

1.1 Motivation

A virtual environment is an interactive, head-referenced computer display that gives a user the illusion of presence in real or imaginary worlds. Research in virtual environments dates back to the 1960s, when Sutherland [Sut65] envisioned that an ultimate computer display could serve as a means for a user to actively participate within a three-dimensional (3D) virtual world. The computer-generated virtual world would be displayed and respond realistically to user inputs in real time. Two most significant differences between a virtual environment and a more traditional interactive 3D computer graphics system are the extent of user's sense of presence and the level of user participation that can be obtained in the virtual environment.

Since the early 1980's, advances in 3D computer graphics hardware, and computer graphics modeling and rendering software have substantially enhanced the realism of computer-generated images. For example, in the stereoscopic film "Avatar", the realism obtained by rendering images of the virtual planet "Pandora" is very impressive. These advances have contributed to the progress of research in presence. Simultaneously, many aspects that can affect user's sense of presence have been studied (*e.g.*, [HD92, She92, WS98]).

Unfortunately, such progress on user interaction with a virtual environment has not been observed. In fact, it is safe to state that 3D interaction with a virtual environment is still very cumbersome and can rapidly introduce user fatigue and stress [Sha98, KLJ⁺11, MCB⁺11]. According to Brooks [Bro99], interaction has been considered as one of the most crucial issues to be addressed in virtual environment research.

What are the reasons that 3D interaction is so cumbersome? One may be attributed to the intrinsic nature of 3D interaction in virtual environments. It has been argued that users have difficulty in controlling multiple degrees of freedom (DOFs) simultaneously [MM00], moving and responding accurately based on depth perception [TWG⁺04], interacting in a volume rather than on a surface [BJH01] and understanding 3D spatial relationships in virtual environments [HvDG94]. Another reason could be that multimodal cues that exist in the real world are poorly supported or even missing in virtual environments. For example, continuous haptic cues, such as gravity and friction, which are essential for real-life interaction, are often not available or are of low fidelity [MBS97].

Ample research has been performed, in an attempt to challenge the difficulties that exist in 3D interaction. The research has resulted in a large number of solutions (*e.g.*, in-

novative paradigms, techniques and applications), most of which were developed in different VR frameworks and platforms. However, evidence is accumulating that it is difficult to compare these solutions across various implemented environments, to design new technologies on the basis of previous work and to make progress in developing theories [WTN00, New94, HvDG94].

Two notable approaches devoted to the evaluation of such solutions are the development of interaction taxonomies and interaction models. An interaction taxonomy is an approach to categorize the interaction techniques or devices according to the tasks supported. For example, Bowman et al. [BJH01, BH99], proposed a taxonomy of interaction techniques for several common interaction tasks, including travel (viewpoint motion control), selection and manipulation. Arns et al. [ACN02] extended Bowman's taxonomy to locomotion tasks. Card et al. [CMR90, BB87, Bux83, FWC84] developed the taxonomies of interaction devices, which systematically integrated the methods for both generating and testing the design space of input devices. An interaction model describes a relationship between users' temporal performance of carrying out an interaction task and the spatial characteristics of the task. Examples of interaction models include Fitts' law [Fit54], which predicts the time to point to a target as a function of the distance to and size of the target, and the steering law [AZ97], which predicts the time of navigating through a path as a function of the path length and width.

Both approaches allow for a high-level understanding of 3D interaction tasks, a scientific design, evaluation and application of interaction techniques, and a systematic comparison between interaction devices. Moreover, interaction models have several advances over interaction taxonomies. One of the most outstanding merits is the quantitative consideration that is introduced for measuring 3D interaction. Interaction models can transform the spatial characteristics of a task into the quantitative prediction of users' performance, which provides user interface (UI) designers supportive arguments for good design solutions. In addition, the development of interaction models is independent of ad hoc VR systems and environments and does not require much knowledge of the interaction techniques and devices.

This thesis focuses on the development, evaluation and application of interaction models for 3D pointing, steering and object pursuit tasks.

1.1.1 3D Interaction

In HCI, interaction refers to the act of exchanging information between users and computers. 3D interaction is a form of interaction that occurs in 3D space. As depicted in Figure 1.1, the interaction process can be described by three crucial elements:

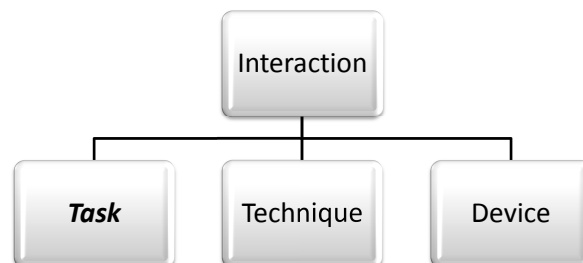


Figure 1.1: The interaction process (adopted from [vR06]).

- An interaction task is the unit of an exchange of information, which is performed to achieve a goal. Interaction tasks can be carried out in virtual environments through interaction techniques and by utilizing interaction devices. Common interaction tasks in virtual environments have been classified as selection, manipulation, navigation, and system control [Bow98]. Examples of common interaction tasks include pointing to a target, navigating through a tunnel, pursuing a moving object, etc., which can be compounded into more sophisticated interaction tasks.
- An interaction technique is the fusion of input and output, consisting of all software and hardware elements, which provides a way for users to accomplish an interaction task [Tuc04]. For example, one can select a virtual object by casting a ray from an input device to the object, intersecting the object with a volume cursor and so on. Interaction techniques are usually classified based on the common interaction tasks they support. Techniques that support navigation tasks are classified as navigation techniques, whereas those that support object selection and manipulation are labeled as selection and manipulation techniques. Interaction techniques can be thought of as the glue between interaction devices and interaction tasks [BL00].
- An interaction device is an I/O peripheral that transfers the information between the user and the computer. An input device is the instrument used to manipulate objects and send control instructions to the computer system. Input devices can be classified according to the modality of the input (*e.g.*, mechanical motion, audio, visual, etc.), the continuity of the input (*e.g.*, discrete key presses, continuous position updates, etc.) or the number of DOFs involved (*e.g.*, 2DOF conventional mice, 6DOF styli, etc.). An output device is used to provide information or feedback to the user. The output devices include visual displays, auditory displays, haptic displays, etc.

1.1.2 Interaction Models

The dependency of users' temporal performance for an interaction task can be attributed to many sources, among which the strategy adopted by the users to balance the speed-accuracy tradeoff and the spatial characteristics of the task play important roles. An interaction model is developed to quantitatively describe users' temporal performance for the task, in which users are instructed to perform as fast as possible without sacrificing accuracy. A key issue is to make sure that they do not trade accuracy for speed or vice versa. Under the circumstances, an interaction model can be defined as the relationship between users' temporal performance for an interaction task and the spatial characteristics of the task. As shown in Figure 1.2, the spatial characteristics of the interaction task represent the variables that can be independently controlled and manipulated during the interaction process and thus function as the independent variables; the users' temporal performance of the task is the variable that responds to the change of the independent variables and is defined as the dependent variable. Therefore, the interaction models describe users' temporal performance as a function of the spatial characteristics of the tasks¹. The development of an interaction model is the process of identifying the independent and dependent variables for the interaction task, and formulating their relationship mathematically. Modeling refers to the act of devising or use of interaction

¹There are also other ways to model interaction tasks, *e.g.* describing users' accuracy in completing the tasks as a function of the spatial/temporal characteristics of the tasks. Following a modeling tradition in HCI, however, interaction models in this paper is defined as the time to complete an interaction task as a function of the spatial characteristics of the task such that models can be used as metrics to compare interaction efficiency.

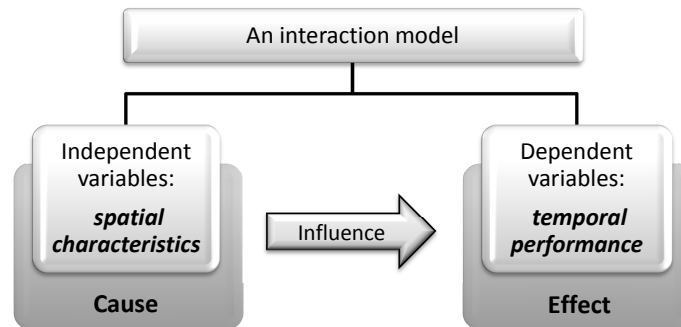


Figure 1.2: The composition of an interaction model.

models for the associated tasks. Therefore, modeling is an approach that directly deals with interaction tasks, independent of interaction techniques and devices.

Interaction models can be used to quantitatively predict the time to complete an interaction task as a function of the spatial characteristics of the task [Fit54, AZ97]. They offer a way to compare and evaluate interaction techniques that were developed for different environments and platforms [MSB91, KB95, MO98]. Interaction models also serve as metrics of comparing the performance of various input devices for the same type of interaction tasks² [MSB91, Mac92, MB93]. In addition, design implications and guidelines for UIs can be deduced and derived as a result of an in-depth analysis of the interaction models [KB95, AZ97, GB04].

1.2 Objective

The objective of the research is twofold. First, we aim to develop interaction models for 3D pointing, steering and object pursuit tasks that occur in a desktop virtual environment, in an attempt to facilitate VR UI designers in developing new interaction techniques or input devices. The models need to be validated and, when required, extend existing 1D/2D interaction models. The second objective is to gain a better understanding of users' movements in the virtual environment through the analysis of interaction models and experimental results.

1.3 Approach

The research approach incorporates four procedures as shown in Figure 1.3. The procedures can be applied to the study of each interaction task involved in this thesis.

- *Variable Identification*

The first procedure aims to identify the independent and dependent variables for the interaction task to be modeled. It can be achieved either by borrowing from existing 1D/2D interaction models of the same type of task, or through exploring new features of the 3D tasks. Two important independent variables, *i.e.*, the length to be traveled

²Interaction models provide a quantitative and objective way to compare input devices. The assessment of comfort, which needs to be addressed through subjective assessment, such as asking users to fill in questionnaires, is beyond the scope of interaction models.

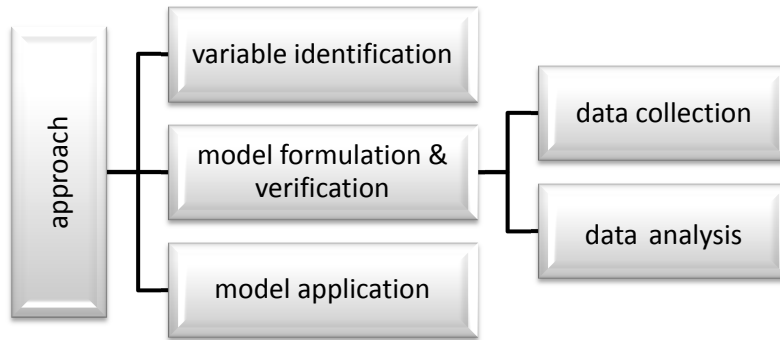


Figure 1.3: The research procedures.

L and the width of the constraint W , should be initially considered for all interaction tasks. Depending on the tasks, other variables might be involved.

- *Data Collection*

User studies offer a scientifically sound approach to get the data which can be used to formulate and verify an interaction model. User studies are implemented by carrying out reliable, replicable and generalizable VR experiments that are of a within-subject (repeated-measures) or a between-subject (independent-measures) design, and recording users' temporal and spatial information while they are performing the interaction tasks.

- *Data Analysis*

The data analysis involved in this thesis mainly follows a modeling methodology, which consists of two steps³. The first step focuses on how users' temporal performance (dependent variable: T) is affected by the length to be traveled during the interaction task over the width of the constraint (independent variable: L/W) of the task. The goal is to derive/verify a linear relationship between $\log(T)$ and $\log(L/W)$, *i.e.*,

$$\log T = a + b \log\left(\frac{L}{W}\right) \quad (1.1)$$

through repeated-measures regression analysis⁴ so that the relationship can be transformed to Stevens' power law [Ste57]:

$$T = 10^a \left(\frac{L}{W}\right)^b \quad (1.2)$$

where the exponent b depends on the type of interaction task and the stimulus condition for the same task, and thus can be used to compare between/within interaction tasks.

The second step aims to examine how a and b in Equation 1.1 depend on other independent variables and analytically describe a and b as a function of the independent variables, respectively. Therefore, a complete interaction model can be derived by replacing a and b with the associate functions that are composed of other independent variables.

³The modeling methodology was originally proposed by J.-B. Martens in paper [LMvL10].

⁴The reason for such a choice, rather than directly examining the relationship between T and L/W is specified in Section 2.5 and Appendix A.1.

- *Model Application*

The use of interaction models is approached by applying the derived models for a quantitative evaluation of the interaction techniques and input devices, and gaining an insight into users' 3D movements in the virtual environment.

1.4 Scope

The interaction tasks studied in this thesis, including 3D pointing, steering and object pursuit, are performed in a desktop virtual environment [Fur06], where the 3D virtual world is displayed by using a stereoscopic 3D view on a regular 2D monitor. The interaction is achieved by means of a six degree-of-freedom input device with translation in three perpendicular axes (x , y , and z) and rotation about three axes (pitch, yaw and roll) in a 3D space, and a three degree-of-freedom haptic device (translation only) that creates a realistic sense of touch. The scene-in-hand metaphor [WO90] is utilized to design the interaction techniques that enable a user to have an external view of an object and manipulate the object directly via hand motion.

1.5 Contributions

The contributions are twofold in accordance to the objective. First, we have developed, extended and validated five interaction models to describe three interaction tasks in the virtual environment:

- Fitts' law, which was proposed for 1D/2D pointing tasks, is verified for 3D pointing tasks;
- the two-component model is used to compare 3D pointing tasks in the real world and in virtual reality;
- the steering law is extended to 3D manipulation tasks;
- a new pursuit model is formulated for 3D object pursuit tasks;
- Stevens' power law is used as a general law to model/compare 3D pointing, steering and object pursuit tasks.

Second, it is demonstrated that 3D pointing movements in the virtual environment can be broken into a ballistic phase and a correction phase, and the correction phase in the virtual environment contains more sub-movements and takes longer than in the real world; 3D steering movement for the ball-and-tunnel task (see Section 4.2) is composed of several small and jerky sub-movements when performed with a 6DOF stylus device in the virtual environment, but the movements become smoother with haptic feedback presented.

1.6 Thesis Outline

The rest of the thesis is organized as follows:

- In chapter 2, a survey of the relevant research on pointing, steering and object pursuit is provided. The emphasis is to review some of the commonly accepted models for each interaction task, and illustrate the use of the interaction models in evaluating and comparing the available interaction techniques and input devices.

- Chapter 3 focuses on the study of pointing tasks. It commences with a comparison between pointing tasks in the real world and in virtual reality. The results are further used to develop a methodology that enables the development and evaluation of pointing-task-oriented interaction techniques.
- Chapter 4 aims to model path steering for 3D manipulation tasks. In particular, the influence of path curvature and orientation is experimentally modeled/examined on paths of constant/variable properties. In addition, we also investigate path steering in the presence of force feedback, which is achieved by comparing haptic steering with non-haptic steering and modeling the effect of force magnitude.
- Chapter 5 introduces an object pursuit task to HCI and studies the interaction task with moving objects. A spatio-temporal relationship that resembles Fitts' law and the steering law is initially proposed and empirically verified for the object pursuit task.
- In chapter 6, we summarize the work, draw the conclusions, discuss the remaining issues and exploit the potential for future work.

1.7 Publications from This Thesis

The thesis is based on the peer-reviewed conference and journal publications as listed below:

1. L. Liu, R. van Liere, C. Nieuwenhuizen, and J. -B. Martens. Comparing aimed movements in the real world and in virtual reality. In *VR'09: Proceedings of IEEE Virtual Reality 2009*, pages 219-222, 2009. (**Chapter 3** and **Appendix C**)
2. C. Nieuwenhuizen, L. Liu, R. van Liere, and J. -B. Martens. Insights from dividing 3D goal-directed movements into meaningful phases. *IEEE Computer Graphics and Applications (CG&A)*, volume 29, issue 6, pages 44-53, November/December 2009. (**Chapter 3**)
3. L. Liu and R. van Liere. Designing 3D selection techniques using ballistic and corrective movements. In *EGVE'09: Proceedings of Eurographics Symposium on Virtual Environments 2009*, pages 1-8, 2009. (**Chapter 3** and **Appendix C**)
4. L. Liu, J. -B. Martens, and R. van Liere. Revisiting path steering for 3D manipulation tasks. In *3DUI'10: Proceedings of the 2010 IEEE Symposium on 3D User Interfaces*, pages 39-46, March 2010. [best paper award] (**Chapter 4** and **Appendix A**)
5. L. Liu and R. van Liere. The effect of varying path properties in path steering tasks. In *JVRC'10: Proceedings of Joint Virtual Reality Conference of EuroVR - EGVE - VEC 2010*, pages 9-16, 2010. (**Chapter 4**)
6. L. Liu, J. -B. Martens and R. van Liere. Revisiting path steering for 3D manipulation tasks. *International Journal of Human-Computer Studies (IJHCS)*, volume 69, issue 3, pages 170-181, March 2011. [extension of Publication 4] (**Chapter 4** and **Appendix A**)
7. L. Liu, R. van Liere, and K. J. Kruszyński. Modeling the effect of force feedback for 3D steering tasks. In *JVRC'11: Proceedings of Joint Virtual Reality Conference of EuroVR - EGVE 2011*, pages 31-38, September 2011. (**Chapter 4**)

8. L. Liu, R. van Liere, and K. J. Kruszyński. Comparing path steering between non-haptic and haptic 3D manipulation tasks: Users' performance and models. In *GRVR11: Proceedings of the IASTED International Conference on Graphics and Virtual Reality 2011*, pages 1-8, July 2011. (**Chapter 4**)
9. L. Liu and R. van Liere. Modeling object pursuit for 3D interactive tasks in virtual reality. In *VR'11: Proceedings of IEEE Virtual Reality 2011*, pages 1-8, March 2011. (**Chapter 5**)
10. L. Liu and R. van Liere. Modeling object pursuit for desktop virtual reality. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*. [submitted as an invited paper; extension of Publication 9] (**Chapter 5**)

Chapter 2

Modeling Interaction Tasks: An Overview

This chapter presents an overview on the research of interaction models and their applications in the context of pointing, steering and object pursuit tasks. Comparisons between the three types of interaction tasks are also provided.

2.1 Modeling Pointing

Pointing is an aimed movement that requires one to depart from a source and rapidly move toward and select a target. Figure 2.1 shows an example of the pointing task in HCI. It is one of the most common interaction tasks that are frequently performed in a variety of user interfaces and thus has been studied extensively in HCI.

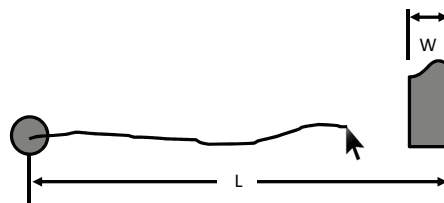


Figure 2.1: An example of the pointing task in HCI.

In the literatures, two approaches have been proposed to model pointing tasks. The first approach considers pointing as a whole. The dependent variables under observation focus on the characteristics of the total movement, such as the total time or displacement. Fitts' law, which predicts the time of the total movement with the spatial characteristics of the tasks, falls into this category. The second approach models the total movement with several sub-movements, each of which provides some information about the overall movement. Woodworth's two-component model is a classic instance. In this section, a survey into both approaches is provided.

2.1.1 Fitts' Law

Fitts' law is a model of human movement that is used to quantitatively describe the act of pointing. It predicts the time to rapidly move and point to a target as a function of distance to and size of the target. Fitts' law was proposed by Paul Fitts [Fit54] for 1D rapid aimed movements in the discipline of information theory in 1954, extending Shannon's Theorem 17 [Sha48]. Card et al. [CEB78] introduced it to HCI in 1978.

Over the years, variations of Fitts' law have been formulated (*e.g.*, [Wel60, BGM⁺72, Kva80, KK78, JRWM80]), among which the following form is commonly accepted [Mac89, Mac92]:

$$T = a + b \log_2\left(\frac{L}{W} + 1\right) \quad (2.1)$$

where

- T is the time to complete the pointing task;
- a and b are experimentally determined constants that can be derived from fitting a straight line to the observed data (a linear regression);
- L is the distance to the target (between the starting point and the center of the target);
- W is the width of the target (along the axis of motion).
- $\log_2(L/W + 1)$ is referred to as the index of difficulty (ID) of the task.

Intuitively, Fitts' law states that acquiring a big target within a short distance requires less time than a small target at a long range.

Mathematically interpreted, Fitts' law is a linear regression between the movement time and ID . The regression coefficient b is the slope of a straight line, whose reciprocal, *i.e.*, $1/b$, characterizes how quickly pointing can be done, independent of the specific targets involved, and is defined as the index of performance (IP). IP , in bits/second, is adopted by ISO 9241 part 9 standard [ISO98] to define a throughput (TP) which can be used to measure the performance of a non-keyboard input device. Figure 2.2 shows an example of how Fitts' law and IP can be used to compare input devices. Each straight line represents a regression

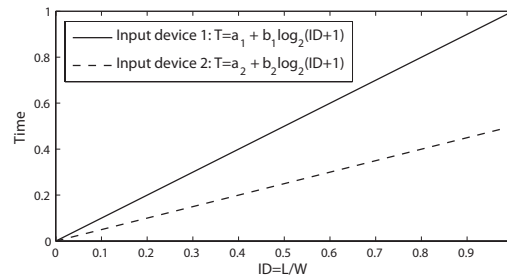


Figure 2.2: An example of input device comparison using Fitts' law ($a_1 = a_2 = 0; b_1 > b_2$).

result derived from an input device. Given $IP_1 = 1/b_1$, $IP_2 = 1/b_2$ and $b_1 > b_2$, it can be deduced that $IP_1 < IP_2$, indicating input device 2 is faster in doing pointing tasks than input device 1.

Fitts' law was originally derived from a 1D experiment, which inevitably restricts its application in higher dimensional pointing tasks. Inspired by the implication of Fitts' law, HCI researchers started to develop new models on top of Fitts' law, which could be used in 2D and 3D pointing tasks. The research mainly focuses on how to appropriately interpret the

target width “ W ” in Fitts’ 1D model for 2D and 3D pointing tasks. For 2D, the first extension was made by Crossman [Cro56], who introduced the vertical height of a 2D target “ H ” to Fitts’ law, in addition to W . The conclusion was that “the restriction in the extra dimension appeared to affect performance in much the same way as the restriction in width, but to a slightly lesser degree” [HS94]. MacKenzie and Buxton [MB92] proposed five possible extensions of Fitts’ law and experimentally chose two that gave the best description of the empirical data. One extension replaced W with the minimum of H and W , while the other replaced W with “the apparent width” in the direction of motion. Accot and Zhai [AZ03] revisited the pointing and proposed a model that was similar to Crossman’s idea, but the effect of L/W and L/H was combined in a Euclidean way. Another way to extend Fitts’ law is to introduce other independent variables, rather than an appropriate extension of the target width. For example, Murata [MI01] studied how acquiring targets placed at different directions influenced the movement time, which is an important factor for 3D interaction.

For 3D, Ware et al. [WB94, WL97] involved the depth of a target “ D ” in their model as a consequence of introducing an extra dimension to the pointing tasks, together with W and H . The term L/W in Fitts’ law was replaced by $L/\min(W, H, D)$. Grossman and Balakrishnan [GB04] extended Ware et al.’s work and further proposed a 3D version of Accot and Zhai’s weighted Euclidean model [AZ03], where the term L/W , L/H and L/D were assigned different weights. Their work indicates that the effect of target depth in 3D space is as similar as that of target width and height in 2D space.

2.1.2 Application of Fitts’ Law

Input Device Evaluation

One application of Fitts’ law is the widespread use of the index of performance IP (see Section 2.1.1) in comparing input device performance either within a study or across studies. Card et al. [CEB78] used Fitts’ law to quantitatively compare the performance of completing a pointing task between the mouse, isometric joystick, step keys and text keys. The experimental results indicated that the mouse outperforms all the other three types of input devices. MacKenzie et al. [MSB91] made a similar comparison between the mouse, trackball and tablet with stylus. MacKenzie [Mac92] also compared the performance of the mouse, trackball, joystick, touchpad, helmet-mounted sight, and eye tracker from six independent studies, extending Fitts’ law to a cross-study analysis. There are also a number of similar studies concerning the use of Fitts’ law for input device comparison (e.g., [DKM99, Epp86, Zha04, KE88, Mac91, RVL90]). In 1998, IP was employed as an official standard [ISO98] for assessing the performance of a non-keyboard input device.

Interaction Technique Evaluation & Development

Similarly, the index of performance IP was also used to compare the performance of interaction techniques. For instance, MacKenzie et al. [MSB91] compared point-and-click and drag-and-drop techniques for the same pointing tasks. It was shown that point-and-click has a higher IP than drag-and-drop, *i.e.*, pointing with point-and-click is faster. This is intuitive as the increased muscle tension can make drag-and-drop more difficult. Kabbash and Buxton [KB95] proposed a “Prince” technique, which was compared with the traditional pointing techniques using the IP . Their conclusion was that the Prince technique can be an alternative approach to pointing, since its IP is as high as traditional pointing techniques. For touchpad pointing tasks, MacKenzie and Oniszczak [MO98] adapted IP to the comparison of three

selection techniques, including a physical button, lift-and-tap and finger pressure with tactile feedback. The empirical results showed that the tactile condition was 20% faster than lift-and-tap and 46% faster than using a button for selection.

Fitts' law offers a way to develop interaction techniques, as it quantitatively describes how pointing efficiency can be improved by adjusting the distance to the target and the size of the target. For instance, decreasing the distance to the target can result in a group of interaction techniques, including *drag-and-pop* [BCR⁺03] which remotely drags the target towards the cursor; *pop-up linear/pie menus* [CHWS88] which pop up linear/pie menus at cursor's current position, avoiding travel before selection occurs; *object pointing* [GBBL04] that skips across the empty space; *Go-Go* [PBWI96] which makes the virtual hand travel faster in reaching distant objects; and *snap-dragging* [Bie88] that makes the cursor snaps the target as the cursor approaches the target. In addition, increasing the size of the target can also lead to a series of interaction techniques, such as the *area cursor* [WWBH97] which has larger activation area; *volume cursor* [ZBM94] which represents the cursor with a volume, rather than a point; and *Mac OS dock* that expands the target as the cursor approaches. There are also interaction techniques that are designed by simultaneously decreasing the distance to the target and increasing the size of the target. Examples include *semantic pointing* [GBBL04] and *PRISM* [FKK07] which dynamically adjust C-D ratio between the hand and the controlled object to provide increased control when moving slowly (equivalent to increasing the size of the target), and unconstrained interaction when moving rapidly (equivalent to decreasing the distance to the target).

Design Guideline Formulation

Fitts' law can be simply interpreted as that the time to acquire a target can be reduced if the target becomes bigger and is located closer. Following Fitts' law, theoretical principles and guidelines for designing efficient user interfaces can be formulated. For example, a frequently-triggered incident should be assigned to a relatively larger button and should be placed at a closer distance to the cursor position [Fit, Zha02]; edges and corners of a computer screen (*e.g.*, the location of the start button in Microsoft Windows and the menus and Dock of Mac OS X) are easier to point at [Hal07, Atw06], since the cursor is bounded to the area regardless of how much further the mouse is moved and the area can be thought of as having infinite width (see Figure 2.3 for demonstration); pop-up menu (right-click menu) is usually faster to acquire than pull-down menu, since users avoid traveling; items in a pie pop-up

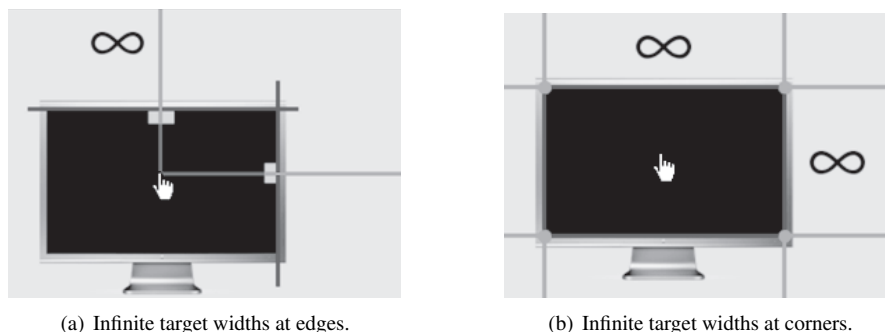


Figure 2.3: Design implications of Fitts' law [Hal07].

menu are usually faster to acquire than those in a linear pop-up menu [Hop91]. However, it is worth noting that the guidelines above are just a few examples. In practice, user interface designers usually have to balance the tradeoff between applying Fitts' law and other design decisions, such as the organization of the available screen space.

2.1.3 The Two-Component Model

Despite Fitts' law and its extensions have been evidenced to be valid in describing the complete movement time for a pointing task, they cannot provide other information during the movement. A different approach is to decompose a pointing movement into meaningful sub-movements and study users' performance in each sub-movement. One of the well-formulated models that utilized this idea is the *two-component model* [Woo99], which was proposed by Woodworth early in 1899. It assumes that an aimed movement is composed of a ballistic phase and a correction phase. The ballistic phase is programmed under the central control to bring the limb into the region of the target, while the correction phase comes immediately after the ballistic phase when the limb enters into the range of the target. It is at this moment that visual feedback is used to generate more small adjustments and corrective behaviors. Figure 2.4 depicts a typical velocity profile¹ of an aimed movement as a function of distance traveled to the target. As shown, the entire movement can be broken into the two phases on the basis of the velocity.

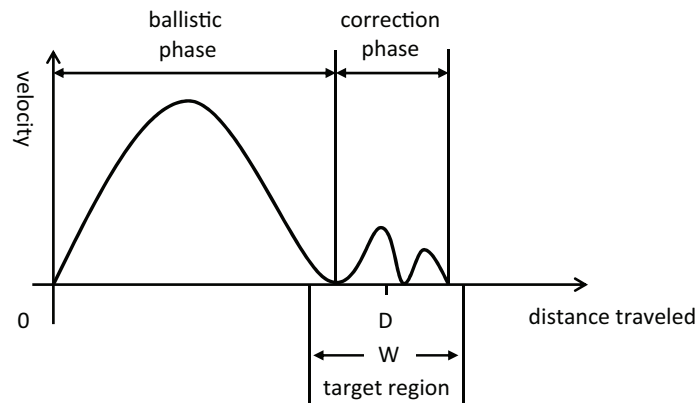


Figure 2.4: The two-component model based on the movement velocity profile.

The two-component model is important in modeling pointing tasks, as it allows for the analysis of users' movement during the tasks, which can assist us in gaining an insight into the pointing movement. Following the two-component model, ample research has been conducted, which resulted in several variations of the model (*e.g.*, [BH70, Car81, MAK⁺88, EHMT04]). Experimental evidence also showed that practice can lead to a correction phase that begins earlier [EHG⁺10] and can also significantly reduce the pointing errors [KFG98].

In order to use the two-component model in the analysis of movement, one needs to divide a movement into sub-movements. Meyer et al. [MAK⁺88] proposed a series of sub-movement parsing criteria, which could decompose the movement when any of the following types of sub-movements is detected:

¹The instantaneous velocity is measured along the task axis.

- Type 1 sub-movement (Figure 2.5, left): returning to the target after overshooting. It occurs when a zero velocity in displacement is reached from positive to negative.
- Type 2 sub-movement (Figure 2.5, middle): undershooting and re-accelerating to the target. It occurs when a zero acceleration² is reached from negative to positive and corresponds to a local minimum in the velocity profile.
- Type 3 sub-movement (Figure 2.5, right): a slight decrease in the rate of deceleration. It occurs when a zero jerk³ is reached from positive to negative and corresponds to an inflection point in the velocity profile.

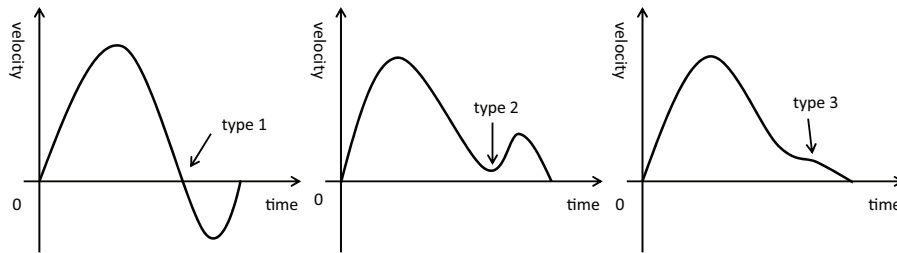


Figure 2.5: Three types of sub-movements.

2.2 Modeling Steering

Path steering is the act of rapidly navigating through a path (or a tunnel) within a given boundary. Driving a car down a road, for instance, is a typical path steering task in the real world. Path steering is also one of the most common interaction tasks that are frequently performed in various user interfaces. Navigating through nested-menus, drawing curves within boundaries, locomotion along a predefined track and navigating through a vessel wall as shown in Figure 2.6 are just a few examples of interaction tasks that can be thought of as path steering.

For years, a variety of models for path steering have been put forward (*e.g.*, [Ras59, Dru71, AZ97]), among which *the steering law* proposed by Accot and Zhai [AZ97] in 1997 has a widespread application.

2.2.1 The Steering Law

Accot and Zhai's steering law is an interaction model that describes users' performance of steering through a path. The governing idea of the steering law assumes that a path steering task can be broken into an infinite number of subtasks, each of which can be treated as a goal-crossing task with the same index of difficulty (see Appendix B for details). The total movement time can then be modeled by Fitts' law, whose *ID* can be derived from calculating the integral of all the subtask *ID*s. If the path width varies along the path, the generic steering law can be expressed by the following formula:

$$T_C = a + b \int_C \frac{ds}{W(s)} \quad (2.2)$$

²Acceleration is the derivative of velocity with respect to time.

³Jerk is the derivative of acceleration with respect to time.

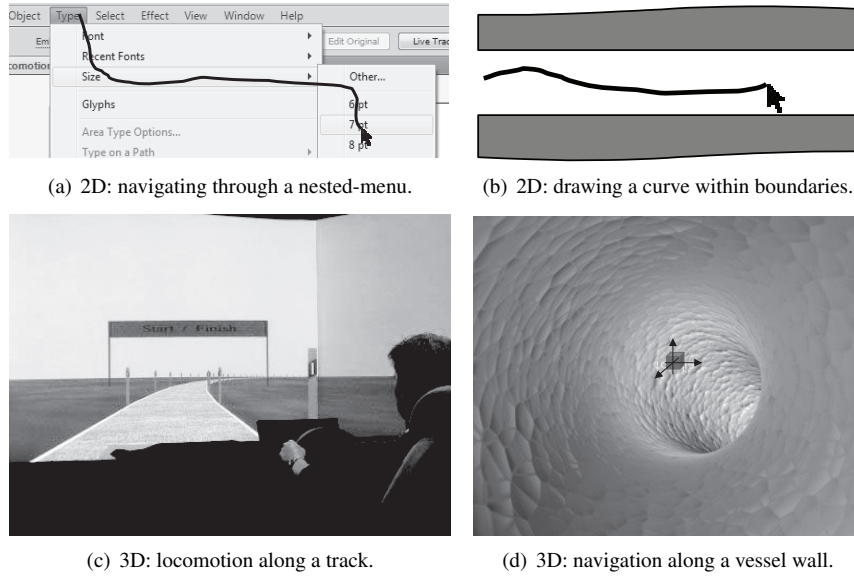


Figure 2.6: Four examples of path steering tasks.

where

- T is the time to navigate through the path;
- a and b are empirically determined constants;
- C is a curved path;
- s is an elementary path length along C ;
- $W(s)$ is the path width at s ;
- The term $\int_C \frac{ds}{W(s)}$ is referred to as the index of difficulty (ID) of the steering task;
- $1/b$ is the index of performance (IP) that is widely used to evaluate interaction techniques and input devices.

In those cases where path width is constant, the steering law can be rewritten as:

$$T_C = a + b \frac{L}{W} \quad (2.3)$$

where L and W represents the total length along C and the width of the path, respectively. Differentiating Equation 2.2 in terms of s on both sides, Accot and Zhai [AZ97] derive a local law as shown in Equation 2.4, which describes the instantaneous velocity of the movement.

$$v = \frac{ds}{dT} = \frac{W(s)}{\tau} \quad (2.4)$$

For a path of constant width W , Equation 2.4 can be simplified as:

$$v = \frac{W}{\tau} \quad (2.5)$$

Equation 2.5 implies that the instantaneous velocity does not vary if path width is kept constant.

The steering law has been adapted to various conditions. For instance, Kattinakere et al. [KGS07] proposed to take the “thickness” of a path into account when the area above the display is used for interaction, which leads to a 3D steering law. Yang et al. [YIBB09] studied a 2D haptic steering task, in which a force guidance was applied in such a way that any deviation from the center of the path is pulled back with a force that is proportional to the distance deviated. It resembles the effect of installing a spring at the center of the path, which is equivalent to increasing the width of the path. They considered the amount of force feedback for steering through a 2D tunnel as an independent variable for the steering time and derived a model based on Accot and Zhai’s goal-crossing idea. There are also studies aiming to extend the steering law to 3D haptic steering tasks. For example, in Keefe’s work [Kee07], the effect of local curvature and orientation on movement time and velocity was examined in the presence of force feedback, respectively.

2.2.2 Application of the Steering Law

Input Device Evaluation

Accot and Zhai have used the steering law to evaluate the performance of five input devices, including mouse, tablet with stylus, trackball, touchpad and trackpoint in trajectory-based tasks [AZ99]. Their experimental results showed that the mouse and the tablet had significantly greater indices of performance *IP*s than the other three devices, indicating that the mouse and tablet are more efficient in performing steering tasks. Dennerlein et al. [DMH00] used the steering law to compare between a force-feedback mouse and a conventional mouse. The conclusion was that steering with the force-feedback mouse was faster, which was evidenced by a higher *IP* value. In particular, the vertical movement time was significantly improved according to the *IP* values obtained.

Interaction Technique Evaluation

One example of applying the steering law to the evaluation of interaction techniques is the use of the index of performance *IP* in determining an appropriate control-display (C-D) ratio for input devices. C-D ratio [CVBC08] is the ratio of the movement of the input device to the change of the visual feedback. It is a common interaction technique to adjust the C-D ratio to achieve either a faster movement or a better control of the constrained interaction. Accot and Zhai carried out an experiment [AZ01], in which users were required to navigate through paths in different C-D ratio scenarios. For each scenario, the steering law was used to calculate the corresponding *IP*. The empirical results showed that as the C-D ratio increases, *IP* tends to have a concave downward parabolic shape. This indicates that the steering law can be used to determine an appropriate C-D ratio, with which *IP* reaches the peak, making steering the most efficient.

Design Guideline Formulation

The steering law can be intuitively interpreted as navigating through a wide and short path takes less time than through a narrow and long path. The underlying idea implies that when designing user interfaces, pull-down menus should be kept wide and short. This rule has been widely adopted for designing current user interfaces. For example, in Microsoft Windows operating systems, the pull-down menus are designed in such a way that once a main menu is selected and unfolded, the menu does not disappear even if the cursor goes beyond the

boundary. This is equivalent to a steering task with infinite path width and thus it transforms the steering task into a pointing task, decreasing the difficulty of the task.

However, this is not always true when the pull-down menus can have nested submenus. In [AZ97], Accot and Zhai mathematically derived that the time to steer through nested menus can be minimized when the menu width and length are kept at a fixed proportion. As shown in Figure 2.7, navigating through a nested menu can be considered as two separate steering tasks (one in the vertical direction and one in the horizontal direction), each of which can be modeled by the steering law. The total time for selecting the n^{th} menu can be calculated by

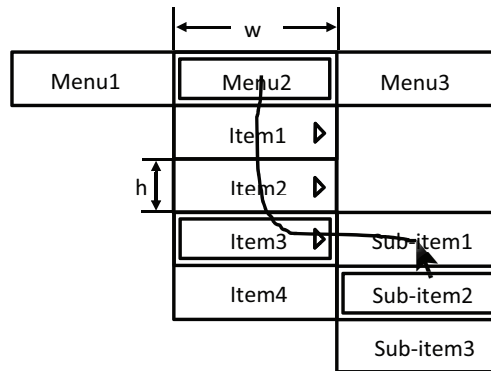


Figure 2.7: Navigating through a nested menu.

Equation 2.6, from which it is deduced that T_n reaches the valley when $x = \sqrt{n}$, *i.e.*, $w = h\sqrt{n}$. This provides us a guideline for designing user interfaces with nested menus.

$$T_n = \underbrace{a + b \frac{nh}{w}}_{\text{vertical steering}} + \underbrace{a + b \frac{w}{h}}_{\text{horizontal steering}} = 2a + b \left(\frac{n}{x} + x \right) \quad \text{with } x = \frac{w}{h} \quad (2.6)$$

2.3 Object Pursuit

Pursuit is the action of following or pursuing someone or something. In this thesis, object pursuit is defined as an interaction task which requires users to track a moving target in user interfaces. A shooting game with moving targets as shown in Figure 2.8 is a typical pursuit task. Object pursuit can be found in gaming, video surveillance systems, air traffic control systems, etc. Object pursuit is a fundamentally distinct interaction task in that users interact with a non-stationary target, which is kept stationary in a pointing or a path steering task.

Given the parts of the human body that are used, pursuit can be categorized into eye movement [LMT87, BM83], locomotion [CF07], manipulation tasks [SBJ⁺97], etc. As one of the important human skills, pursuit has been extensively studied in the discipline of psychology. For example, it was used to differentiate normal subjects from psychiatric patients [IMB⁺92, Flo78, GMK00] or to qualify a pilot [Hes81, MR76]. To our knowledge, however, it has never been researched as an interaction task in HCI. Accordingly, there is no available model that allows for quantitative understanding of the task and no metrics can be used to evaluate the interaction techniques and input devices that are designed for such a task.



Figure 2.8: The example of an object pursuit task in a shooting game.

2.4 Differences between Pointing, Steering and Object Pursuit

Pointing, steering and object pursuit are three types of interaction tasks that are distinct in nature. As shown in Figure 2.9, the constraints imposed on the interaction task determine the intrinsic characteristics of the task. In a pointing task, users are not restricted to any boundary

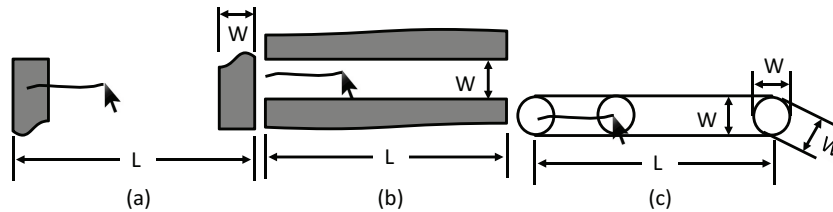


Figure 2.9: The constraints imposed on pointing, steering and object pursuit tasks. (a) pointing: L is the distance between the source and the target and W is the width of the target along the movement; (b) steering: L is the length of the path and W is the width of the path; (c) object pursuit: L is the length of the path to be crossed by the moving object and W is the width of the object in all directions.

before approaching the vicinity of the target; in a path steering task, the movement has to be performed within the boundary of the path⁴; in an object pursuit task, users are not only constrained by the spatial boundary, but also the temporal boundary, *i.e.*, the movement needs to be done within a certain area at a certain time. It is obvious that the level of constraints in such a sequence is increasing.

In addition, there are also differences between the visual feedbacks that are used by users in each task. Although continuous visual feedbacks are always presented during the tasks, users do not necessarily take full advantage of them instantaneously. In pointing tasks, users have a priori knowledge of the destination before a trial starts. The task does not strongly rely on a continuous visual feedback in the first movement phase (a ballistic movement) and thus is

⁴As the width of the boundary becomes larger and larger, the constraint from the boundary becomes smaller and smaller. If the width of boundary is beyond a threshold, the steering task may turn into a goal-crossing task that can be captured by Fitts' pointing law.

an open loop. During the second phase, users usually need to adjust the movement (if an overshoot or undershoot occurs) according to a continuous visual feedback, which makes the phase a closed loop [EHC01]. Ample fundamental research assumes steering to be a continuous error-correcting mode with permanent visual feedback [RSB81, MW69, MH93], *i.e.*, a closed loop. However, there are also an equal amount of researchers who argue that under many circumstances steering does not require permanent error control [GOD85, Sal01, WCTT07], *i.e.*, users do not have to adjust their movement in a continuous mode, but rather in a discrete mode when only an error correction becomes necessary. Their experimental results suggest that “steering control can be characterized as a series of unidirectional, open-loop steering movements, each punctuated by a brief visual update”. This indicates that small ballistic phases during the steering tasks might be observed. In object pursuit tasks, however, the destination is not known in advance and moreover users have to dynamically adjust their position according to the object’s current position (visual feedback), which inevitably increases the difficulty of the task and generates a closed-loop movement.

2.5 Stevens' Power Law

Stevens' power law [Ste57] is a model which was originally used to describe a relationship between the magnitude of a physical stimulus and its perceived intensity or strength. The general form of the law is

$$\psi = kI^a \quad (2.7)$$

where I is the magnitude of the physical stimulus intensity, ψ is the perceived intensity, and k and a are empirically determined constants that depend on the type of the stimulation. Table 2.1 lists several examples of the exponents reported by Stevens. As shown, one stimulus

Continuum	Exponent (a)	Stimulus condition
Loudness	0.67	Sound pressure of 3000 Hz tone
Vibration	0.95	Amplitude of 60 Hz on finger
Vibration	0.6	Amplitude of 250 Hz on finger
Brightness	0.5	Point source
Lightness	1.2	Reflectance of gray papers
Taste	1.3	Sucrose
Taste	1.4	Salt
Taste	0.8	Saccharin
Warmth	1.6	Metal contact on arm
Heaviness	1.45	Lifted weights
Electric shock	3.5	Current through fingers

Table 2.1: Examples of exponents collected by Stevens.

condition can be modeled using the power law with a specific exponent a , which provides a way to identify the stimulus condition(s)⁵.

The modeling methodology adopted in this thesis uses the power law as a starting point. For each interaction task, we examine whether there is a power relationship between the

⁵Note: one exponent may be associated with several stimulus conditions.

movement time and the term L/W , *i.e.*,

$$T = a\left(\frac{L}{W}\right)^b. \quad (2.8)$$

This is due to the fact that the term L/W which represents the length to be traveled during the interaction task over the width of the constraint is evidenced to play a significant role in affecting the movement time T in pointing and steering tasks, according to Fitts' law and the steering law. The power law comprises a more general class of models that can be approximately transformed into different interaction models. It is of particular interest to find out how T exponentially varies with L/W for different interaction tasks. For example, when the power b is equal to 1, Equation 2.8 represents the steering law with zero intercept. If $b = 1/3$, the curve representing the power law very much resembles the curve representing Fitts' law [LMvL10]. By adjusting the value of power b , we aim to investigate if different interaction tasks can be modeled using the same law in the first step. Then, other variables on which the constants a and b in Equation 2.8 depend might be involved at a later phase of modeling.

Taking the logarithm of both sides of Equation 2.8, we can derive that

$$\log T = \log a + b \log(L/W), \quad (2.9)$$

which implies that instead of examining a power relationship between T and L/W for each interaction task, we can address the question by verifying if there is a linear relationship between $\log T$ and $\log(L/W)$ (see Figure 2.10). This is a better modeling approach from a statistics pointing of view, as T and L/W collected from user studies do not necessarily have normal distribution and equal variance that need to be satisfied before performing statistical analysis, such as regression and ANOVA; taking logarithm of both T and L/W can usually help to meet these assumptions such that the validity of using statistical analysis can be promised.

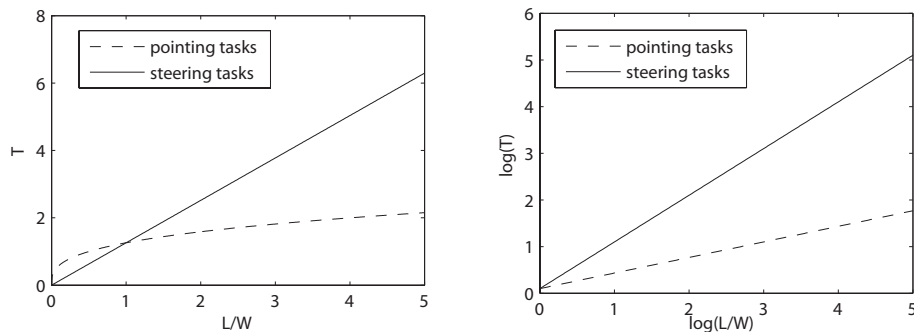


Figure 2.10: Left: T as a power function of L/W ; right: $\log T$ as a linear function of $\log(L/W)$. The linearity of the functions (right) on double logarithmic coordinates indicates that T is a power function of L/W . The slope of the line corresponds to the exponent of the power function.

Chapter 3

Pointing

3.1 Introduction

Fitts' law and the two-component model are two commonly accepted interaction models that are proposed for pointing tasks. Fitts' law quantitatively describes users' movement time with the distance to and the size of the target, which serves as an important guideline for the development of interaction techniques. The two-component model considers a pointing movement as a combination of two movement phases, each of which provides different information during the movement. It is a means to better understand the pointing movements.

In this chapter, the two-component model is employed as the starting point for the study of 3D pointing tasks in a virtual environment. Our goal is to address the following questions:

- Can the two-component model be used to model 3D pointing in the real world and in virtual reality? If so, how can 3D pointing be compared by the two-component model? What is the difference between 3D pointing in the real world and in virtual reality?
- How can the two-component model be used to design interaction techniques that improve users' pointing performance?

These questions are approached through two steps. First, movement parsing criteria are proposed to break 3D pointing movements into the ballistic and correction phases. In each phase, pointing movements collected in the virtual environment are compared to their counterparts in the real world, in an attempt to identify the differences (Section 3.2). Furthermore, a methodology for designing interaction techniques is developed by combining the two-component model with Fitts' law. New interaction techniques are implemented based on the methodology (Section 3.3).

3.2 Comparing 3D Pointing Tasks in the Real World and in Virtual Reality

Despite 2D pointing with 2DOF input devices in traditional desktop UIs is of high accuracy, efficiency and usability, it is often argued that 2D pointing does not make use of the

natural and intrinsic knowledge of how information exchange takes place with physical objects in the real world [FIB95, UI00]. Enabling 3D interaction in the virtual environments should have allowed for more intuitive and efficient interaction, whereas the fact is that 3D pointing utilizing multiple-DOF input devices in virtual reality is usually difficult and time-consuming [BJH01]. As direct 3D pointing in virtual environments uses the metaphor of how pointing occurs in the real world but is different from real-world pointing, it is worth to compare the 3D pointing tasks performed in the real world and in virtual reality, which may help us understand why 3D pointing in VR is difficult and takes longer.

The comparison can be achieved by decomposing 3D pointing movements into the ballistic and correction phases using the two-component model and comparing the real-world and virtual-world movements in each phase. In order to distinguish the correction phase from the ballistic phase, 3D movement parsing criteria [NbMLvL09, LvLNM09] that resemble Meyer et al.'s 1D criteria [MAK⁺88] have been developed and described in Appendix C.1.

3.2.1 Experiment

In this section, we describe a controlled experiment, where users were asked to repeatedly perform pointing movements in the real world and the same movements in virtual reality (orders may vary between users). The goal is to collect users' movement trajectories and the related temporal information under the two conditions and further compare them in the following data analysis.

Apparatus

The experiment was performed in a desktop virtual environment, equipped with

- a desktop PC with an Intel (R) Core (TM) 2 Quad CPU Q6600 @ 2.40GHz and a Nvidia Quadro FX 5600 GPU,
- a 20-inch stereo-capable Iiyama HA202D DT monitor,
- a pair of NuVision 60GX stereoscopic LCD glasses,
- an ultrasound Logitech 3D head tracker,
- and a Polhemus FASTRAK connected with one 6-DOF input stylus.

The FASTRAK sampled the stylus at 120Hz. The monitor resolution was set to 1400×1050 at 120Hz and the head tracker was refreshed at 60Hz. The overall end-to-end latency of the system during the experiment was measured to be approximately 45ms, using the method proposed by Steed [Ste08].

Subjects

The experiment involved 12 skilled computer users, among whom 6 had experience of working in virtual environments, and 6 were naive users. There were 9 males and 11 right-handed users. The participants' age ranged from 25 to 36 years, with an average of 30.7 years old. All subjects had normal or corrected to normal vision and none of them was stereo blind.

Task

In the real-world condition, a physical model as shown on the left of Figure 3.1 was provided as the platform where pointing movements were performed. The model was made up of a chessboard-sized floor and 13 vertical cylinders with a radius of 0.0085m. One cylinder,

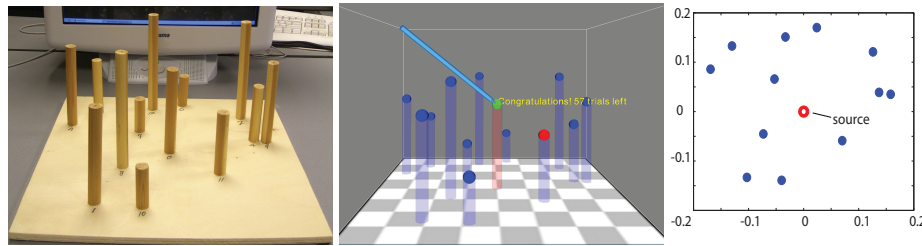


Figure 3.1: Left: real-world platform; middle: virtual-world platform; right: 2D layout from the top view (unit:m).

representing the starting point, was positioned at the center of the floor. It was defined as the source cylinder with a height of 0.14m. The rest of the cylinders, representing the possible destinations, were scattered around the target cylinder¹. They were defined as the target cylinders and might have a height of 0.06, 0.10, 0.14 or 0.18m. The 2D layout of the cylinders from the top view is shown on the right of Figure 3.1, in which each target cylinder has a different distance to the source cylinder. In the virtual-world condition, a virtual model with the same size and layout was designed (see Figure 3.1, middle). The virtual modeled was encapsulated by a fish tank virtual environment.

The experiment was a multi-directional pointing task that resembled the ISO 9241-9 tapping task [ISO98]. Users were required to initialize the task by tapping on the source cylinder, and rapidly perform a pointing movement toward one of the target cylinders. To terminate the task, they needed to tap on the intended target cylinder. Users were asked to hold the tracked stylus as they performed the task. They also had to press and quickly release the stylus button when tapping on the cylinders. The button press event must take place in the vicinity of the cylinders, otherwise it would not be considered as a valid tapping.

In the virtual environment, the vicinity was defined by the volume of a sphere on top of each cylinder. At the beginning of each task, the sphere on the source cylinder (source sphere) and that on the *intended* target cylinder (intended target sphere) were colored red. Other target spheres were colored blue (see Figure 3.1, middle). Once the stylus was brought into any of the spheres, a change in color from red/blue to green would be shown. If users pressed the button within the source sphere, both the source sphere and the intended target sphere would turn to yellow and simultaneously the color of the background would change from grey to black, indicating the start of the pointing and every single motion from then on would be recorded. At the end of the pointing task, the intended target sphere would change back to green if the user successfully pressed the button inside the intended target sphere. Otherwise, they had to proceed until they succeeded.

The color clue presented in the virtual environment was replaced by a numerical clue in the real world due to the differences between the two environments. Each cylinder was assigned with a number between 0 and 12, where 0 represented the source cylinder. At the start of each task, the monitor was used to indicate which of the 12 cylinders the intended target was. The monitor also provided a visual feedback when the task ended. In the real world, the vicinity (on top of each cylinder) for a valid button-press event was calibrated to be the same size and position as in the virtual environment.

¹The layout makes sure that none of the cylinders is an obstacle while pointing to other cylinders.

Experimental Setup

The experimental setup is shown in Figure 3.2, where the differences between the real world and virtual environment were well controlled. In both conditions, users were seated 0.6m in

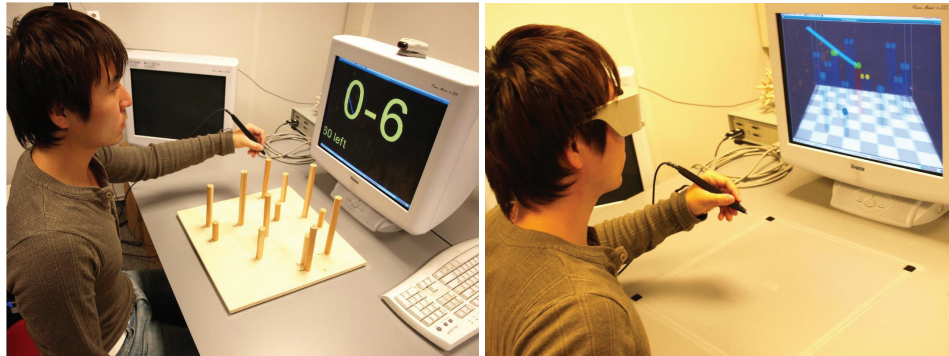


Figure 3.2: The experimental setup. Left: real-world environment. Right: virtual environment. Note: the visual and motor space in the virtual environment were non-co-located.

front of the CRT monitor. The space between the user and the monitor was the motor space where the interaction occurred. The motor space was located such that the virtual cylinders were placed at precisely the same location as the real world cylinders.

There were also differences between the real-world and virtual-world experimental setup. In the real world, the motor space co-located with the visual space, while in the virtual environment, there was a distance 0.3m between the motor space and the visual space, and the motor space was closer to the user; the quality of the visual system, such as the brightness, contrast and resolution of the objects, was inevitably poorer in the fish tank virtual environment; the virtual-world condition introduced the system latency; the haptic feedback presented in the real-world condition was missing in the virtual environment.

Procedure

The experiment was a repeated-measures design with $2 \times 12 \times 5 \times 12$ (number of blocks \times number of targets \times number of repeats \times number of subjects) trials. The experiment was grouped into 2 blocks: one block for the real-world condition and the other for the virtual environment. A block was composed of 60 trials, 5 repetitions for each of the 12 targets.

Trials in a block were presented in a random order which, however, was fixed to the same in the real world and virtual environment for the same subject. Subjects could take a break between the trials or blocks, but this was strictly prohibited within a trial.

A practice session in both the real world and virtual environment was carried out before the data were collected. To counterbalance the learning effect, one half of the subjects were required to complete the real-world block before the virtual-world block, while the other half were in the opposite order.

3.2.2 Results

Pointing Movement Comparison

An instantaneous velocity was calculated as the displacement in a time frame of (approximately) 1/120s over the time frame in the direction of the task axis. For each pointing movement, a velocity profile is a plot of instantaneous velocity as a function of time. Figure 3.3 illustrates the velocity profiles of two pointing movements for one typical trial in the real world and one in the virtual environment. Both movements can be broken into a ballistic phase and a correction phase by applying the non-real-time movement parsing criteria [NbMLvL09, LvLNM09] described in Appendix C.1. In each phase, time and velocity differences can be observed between the real world and virtual world tasks.

To provide statistical evidence that pointing movements in the real world and in virtual reality are different, comparisons are made with seven metrics, including the total movement time, the ballistic phase time, the correction phase time, the proportion of the correction phase time in the total movement time, the number of sub-movements involved in a complete movement, in the ballistic phase and in the correction phase, respectively. For each metric, We have performed a two-way repeated-measures ANOVA, taking the environment condition (real world vs. virtual reality) and the length traveled over the width of the constraint (L/W) as two within-subject variables. The repeated conditions for the same *environment* \times L/W \times *subject* were averaged. Table 3.1 shows the standard ANOVA result for the comparison of the total movement time. Both L/W ($F(11, 121) = 5.0388, p < 0.0001$) and *environment* ($F(1, 11) = 81.7536, p < 0.0001$) are evidenced to have significant effects on the total pointing time. There is also an effect of the interaction term *environment* \times L/W ($F(11, 121) = 2.6546, p = 0.0044$).

source	SS	df	MS	F	p
L/W	4.1186	11	0.3744	5.0388	1.9332e-006
<i>environment</i>	83.1346	1	83.1346	81.7536	2.0033e-006
$L/W \times \textit{environment}$	1.9439	11	0.1767	2.6546	0.0044
$L/W \times \textit{subject}$	8.9910	121	0.0743		
<i>environment</i> \times <i>subject</i>	11.1858	11	1.0169		
$L/W \times \textit{environment} \times \textit{subject}$	8.0550	121	0.0666		

Table 3.1: Comparison between the total pointing time: two-factor ANOVA with repeated measures on L/W and *environment*.

Analogously, another six two-way repeated-measures ANOVAs have been performed to identify the effect of environment and L/W on other metrics. Statistical evidence shows that there are significant differences for each metric that stems from the environment, *i.e.* 3D pointing movements in the real world are significantly different from those in the virtual reality. (see Table 3.2).

In the following analysis, we illustrate the differences for each metric. Figure 3.4 depicts the average time for the total movement, the ballistic phase and the correction phase in both environments (cross L/W analysis). As shown on the left, there is a significant difference between the total movement time in the real world and the virtual world. 3D pointing movements in the virtual environment are less efficient (on average more than two times as long) than those in the real world. Similarly, there is a significant difference between the ballistic phase time. The ballistic phase is about 70% longer in the virtual world than in the real world.

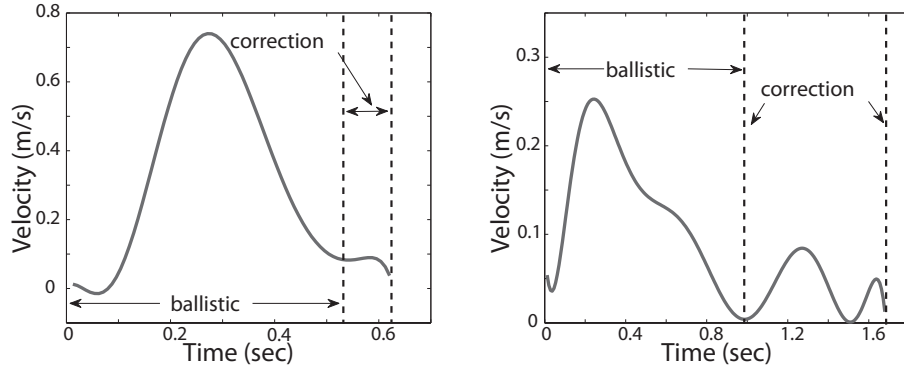


Figure 3.3: Typical velocity profiles of pointing movements (left: real world; right: virtual environment).

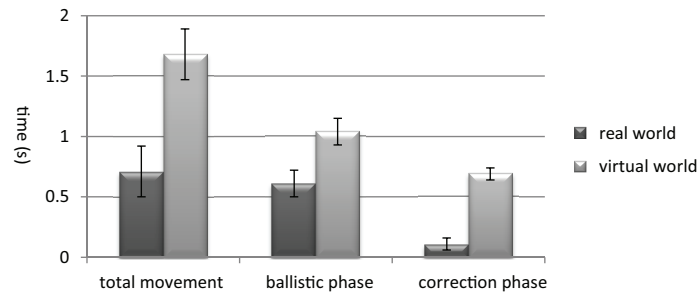


Figure 3.4: Comparisons between the pointing time in the real world and in virtual reality in terms of the total movement, the ballistic phase and the correction phase (across L/W analysis). The top ends of the bars indicate the observation means and the error bars represent the 95% confidence intervals calculated according to repeated-measures designs [ML03] (see Appendix A.2 for more details).

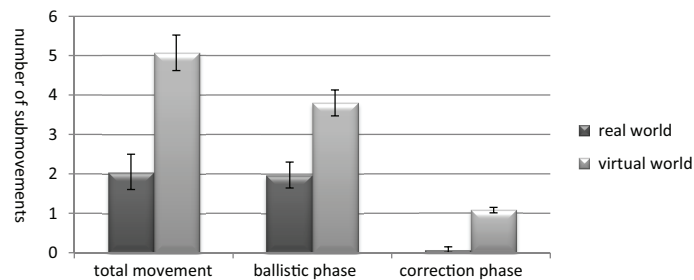


Figure 3.5: Comparisons between the number of sub-movements in the real world and in virtual reality in terms of the total movement, the ballistic phase and the correction phase (across L/W analysis).

real world vs. virtual reality	F(1,11)	p
Ballistic phase time	86.30	< 0.0001
Correction phase time	87.23	< 0.0001
proportion of correction phase	75.87	< 0.0001
#submovement in total movement	113.76	< 0.0001
#submovement in ballistic phase	88.67	< 0.0001
#submovement in correction phase	84.00	< 0.0001

Table 3.2: Effect of environment: the real world vs. the virtual reality conditions.

It is also shown that significant differences also take place between the correction phase time. Figure 3.4 demonstrates that the greatest difference lies in the correction phase where the time under the virtual-environment condition is more than six times as long as that in the real-world condition.

Of particular interest is the analysis of the ratio correction / total movement time. In the real world, only 20.00% (95% confidence interval [14.33%, 25.67%]) of the total pointing time is spent in the correction phase, while in the virtual world, the proportion is 34.64% ([26.14%, 43.14%]), which is 42.26% higher than in the real world. This indicates that the composition of the pointing movements is significantly influenced by the environment.

A similar analysis has been made for the number of sub-movements involved in a total movement, the ballistic phase and the correction phase. As illustrated in Figure 3.5, the number of sub-movements in the total movement is 59.57% greater in the virtual world. Significant differences can also be found in the ballistic phase and the correction phase: 48.16% and 92.59% higher in the virtual world, respectively. The analysis on the number of sub-movements also reveals that over 90% of the trials in the real world have a few (up to three) sub-movements. However, in virtual reality, about 75% of the trials are of three to six sub-movements. This specifies that 3D pointing movements in virtual reality are less smooth but jerkier.

Pointing Model Comparison

In this section, we examine the validity of Fitts' law and Stevens' power law in modeling the pointing tasks in the real world and virtual reality, and compare the performance of the laws in terms of goodness of fit and information loss. The environmental condition, L/W and subject are treated as within-subject variable.

- *Stevens' Power Law*

The verification of Stevens' power law on pointing tasks corresponds to the first step described in the modeling methodology (see Section 1.3), *i.e.*, to look for statistical evidence for Equation 1.1 in the empirical data. First, the total movement time and the associated L/W were transformed logarithmically. Figure 3.6 and 3.7 depict the linear relationship between $\log T$ and $\log(L/W)$ in the real world and in virtual reality. As shown, the linear model (solid line) crosses all of the 95% confidence intervals, while the constant model (dotted line) does not, which indicates that a linear relationship is a better description of the data. Table 3.3 and 3.4 show the corresponding parameter estimates of the linear regression. Statistical evidence indicates that $\log T$ can be linearly related to $\log(L/W)$. For each environmental condition, there is no evidence for lack of fit (real world: $F(7, 711) = -10.065, p = 1.0000$;

virtual world: $F(7, 711) = -45.376, p = 1.0000$), *i.e.*, $\log(L/W)$ is an adequate predictor of $\log T$ and considering the separate effects of L and W are not necessary.

- *Fitts' Law*

Fitts' law requires a linear relationship between the task completion time T and the index of difficulty of the task $\log_2(L/W + 1)$, which can be evidenced in Figure 3.8 and 3.9 for the real world and virtual world pointing tasks. Similar to Stevens' power law, a constant model is not sufficient in modeling the pointing time, while a linear regression appropriately describes the trend. No evidence for lack of fit can be found in both environmental conditions ($F(7, 711) = -76.021, p = 1.0000$; $F(7, 711) = -61.311, p = 1.0000$). The corresponding parameter estimates are shown in Table 3.5 and 3.6. All parameters have significant effects, except the intercept in Table 3.6. Statistical evidence shows that Fitts' law can be used to model 3D pointing tasks in both the real world and the virtual world conditions.

- *Model Comparison*

As shown from Figure 3.6 to 3.9, both models have lower goodness of fit (power law: $R^2 = 0.3579$; Fitts' law: $R^2 = 0.3375$) for the real world pointing tasks, but higher goodness of fit (power law: $R^2 = 0.8395$; Fitts' law: $R^2 = 0.7927$) for the virtual world tasks. In both environmental conditions, Stevens' power law shows a better goodness of fit than Fitts' law. In terms of information loss, a comparison between the two models using Akaike Information Criteria (AIC) also indicates that the power law loses less information than Fitts' law in modeling the 3D pointing tasks (real world: $AIC_{powerlaw} = -77.0815 < AIC_{Fitts'law} = -64.8478$; virtual world: $AIC_{powerlaw} = -86.4909 < AIC_{Fitts'law} = -48.8526$).

Figure 3.10 illustrates that despite Stevens' power law and Fitts' law are formally different, the shape of the curves representing both models shows a high degree of similarity for both real world and virtual world conditions. This implies that Fitts' law can be (approximately) converted into the power law with an exponent that falls into the interval $[0.20, 0.42]$.

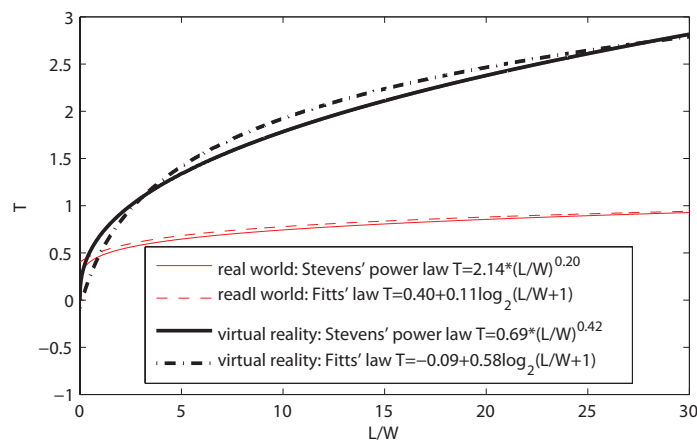


Figure 3.10: The relationship between Stevens' power law and Fitts' law.

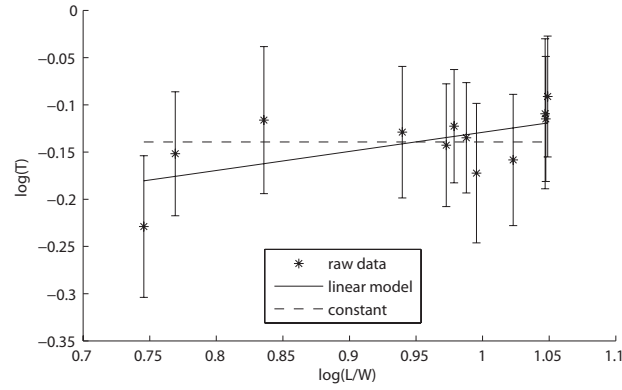


Figure 3.6: The linear regression between $\log T$ and $\log(L/W)$ in the real world condition ($R^2 = 0.3579$). Error bars represent the 95% confidence interval.

	Estimate	t value	P > t	[95% Conf. Interval]
<i>a</i>	-0.331	-7.032	< 0.001	[-0.513, -0.149]
<i>b</i>	0.202	4.094	< 0.001	[0.011, 0.392]

Table 3.3: The parameter estimates of $\log T = a + b \log(L/W)$ for the real world condition.

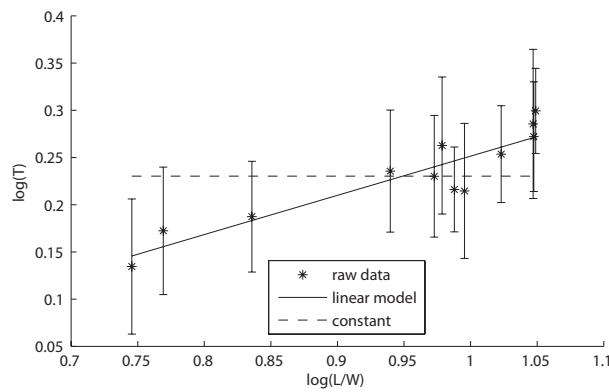


Figure 3.7: The linear regression between $\log T$ and $\log(L/W)$ in the virtual world condition ($R^2 = 0.8391$).

	Estimate	t value	P > t	[95% Conf. Interval]
<i>a</i>	-0.164	-2.996	0.003	[-0.296, -0.042]
<i>b</i>	0.415	7.246	< 0.001	[0.287, 0.544]

Table 3.4: The parameter estimates of $\log T = a + b \log(L/W)$ for the virtual world condition.

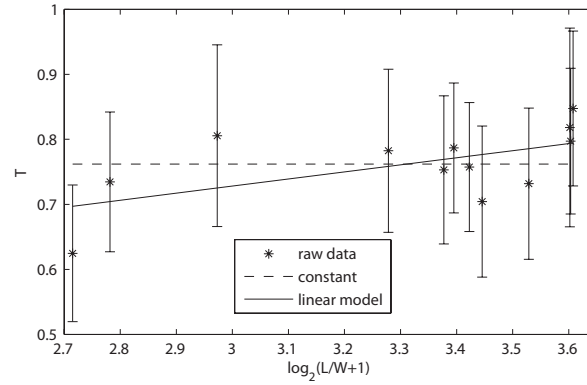


Figure 3.8: The linear regression between T and $\log_2(L/W + 1)$ in the real world condition ($R^2 = 0.3375$).

	Estimate	t value	P > t	[95% Conf. Interval]
a	0.402	4.140	< 0.001	[0.201, 0.592]
b	0.109	3.730	< 0.001	[0.052, 0.166]

Table 3.5: The parameter estimates of $T = a + b \log_2(L/W + 1)$ for the real world condition .

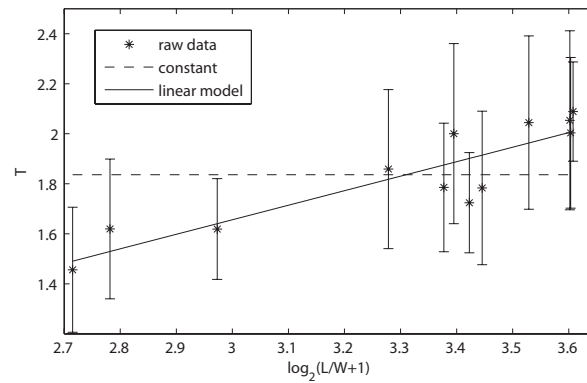


Figure 3.9: The linear regression between T and $\log_2(L/W + 1)$ in virtual world condition ($R^2 = 0.7927$).

	Estimate	t value	P > t	[95% Conf. Interval]
a	-0.085	-0.251	0.802	[-0.751, 0.581]
b	0.580	5.689	< 0.001	[0.380, 0.781]

Table 3.6: The parameter estimates of $T = a + b \log_2(L/W + 1)$ for the virtual world condition.

Learning Effect Analysis

Before starting the experiment, the six naive users were asked to practice 60 trials in both the real world and virtual reality². The data were also recorded and smoothed by a polynomial curve fitting method with the 9th degree of freedom.

Figure 3.11 plots the variation of the movement time as the number of trials performed increases. The plot demonstrates that the time for the total movement, the ballistic phase,

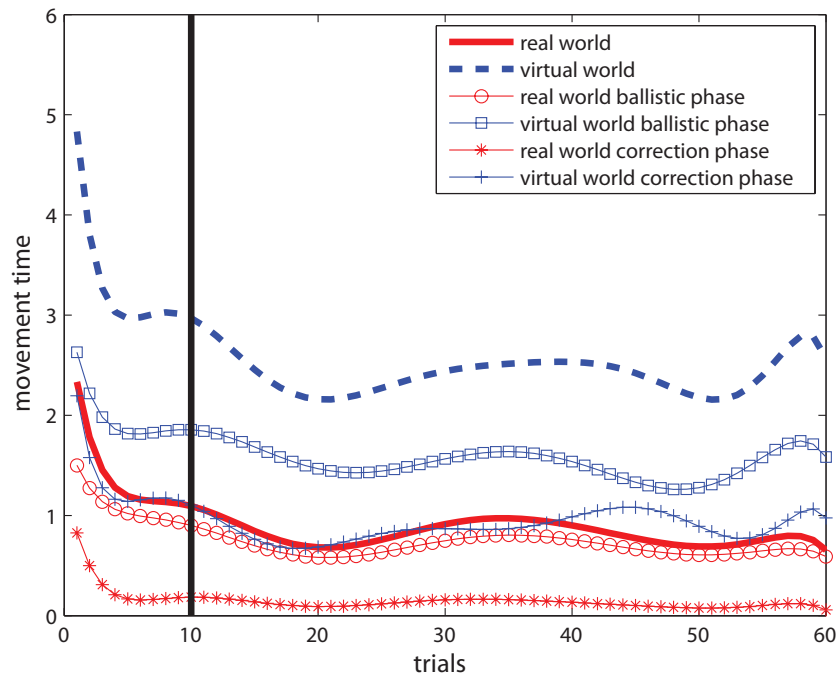


Figure 3.11: The effect of learning (across-subject analysis). Within the first 10 trials, users improve substantially in efficiency. The improvement in the virtual world is greater.

the correction phase decreases substantially within the first 10 trials, but oscillates about an average after the zooming learning period³. The time reduction in the virtual-world scenarios (three blue curves) is larger than its counterpart in the real-world scenarios (three red curves). The total movement time, the ballistic phase time and the correction phase time in the real world drops by 1.68s, 0.91s and 0.77s, respectively. In the virtual world, the decrease is 2.26s, 1.05s and 1.22s. As shown, users progress more in the virtual environment than in the real world. This could be attributed to the fact that they have been performing the pointing movements in the real world and the room for the improvement is relatively small. Furthermore, the time reduction in the real world is more due to the ballistic phase, while the reduction in the virtual reality is more due to the correction phase.

²The number of trials for practice is set under the condition that naive users get acquainted with the pointing tasks, but avoid practicing too much to form a pattern as in musical instrument playing.

³The effect of *ID* has been reduced using across-subject analysis, since *ID* is randomly assigned to trials for each subject.

3.3 Designing 3D Selection Techniques

Although the two-component model was proposed for pointing tasks in the real-world settings, in Section 3.2, we have shown that the two-component model can also be applied to the 3D pointing tasks performed in virtual reality, *i.e.*, the pointing movements in the virtual environments can be broken into a ballistic phase and a correction phase. As shown in Figure 3.3 (right), the ballistic phase usually covers the bulk of the distance to the target⁴ with high velocities. The correction phase, although traversing only in the vicinity of the target, takes almost as much time as in the ballistic phase due to the low velocities and several small adjustments.

Fitts' law predicts the total time for performing a pointing task as a function of the index of difficulty (ID) of the task, which is further determined by the distance to the target (D) and the width of the target (W). This indicates that in order to reduce the movement time, an interaction technique can be devised in such a way that it decreases the distance to the target, increases the target width, or even both simultaneously.

Our aim is to provide a methodology which can be used to design 3D selection techniques by combining the two-component model with Fitts' law. The general idea is to parse a pointing movement in real time into the ballistic and the correction phases, and reduce the index of difficulty of the task in each phase. Similar interaction techniques have been proposed in 2D desktop environments. For example, Balakrishnan[Bal04] proposed to design 2D interaction techniques by decreasing the distance to the target in the ballistic phase and increasing the target width in the correction phase. Worden, et al.[WWBH97] developed the "sticky icon" technique which dynamically adjusted the C-D ratio in different phases. To our knowledge, however, this idea is new for 3D spatial interaction.

If the two movement phases could be distinguished in real time, we can apply different strategies to reducing the ID in each movement phase. Table 3.7 lists several possibilities for designing interaction techniques. One strategy could focus on only one option in one

	Ballistic	Correction	Total movement
D↓	①	②	③
W↑	④	⑤	⑥
D↓ & W↑	⑦	⑧	⑨

Table 3.7: The methodology of designing interaction techniques for 3D pointing tasks. D↓: decreasing the distance to the target; W↑: increasing the target width.

phase, *e.g.*, decreasing the distance to the target in the correction phase as specified by option ②. Another strategy is to combine some of the options and design more complex interaction techniques. For instance, combining option ① and ⑤ would result in a technique that decreases the distance to target in the ballistic phase and increases the target width in the correction phase.

In parallel, to decrease D in the visual space, either the cursor can be automatically brought to the target as what 'snap-dragging'[Bie88] does, or the target can be dragged towards the current cursor position, like 'drag-and-pop'[BCR⁺03]. To increase W , we can visually expand either the target width, as in Apple Mac OS X "dock", or the cursor width, as in 2D 'Area cursor'[WWBH97]. In addition to the visual space, the interaction techniques

⁴The area between the velocity profile and the x-axis in the ballistic phase is much bigger than that in the correction phase.

can also be designed to deal with motor space. For example, the motor space can be scaled by adjusting C-D ratio during the ballistic or the correction phases [BGBL04]. Therefore, Table 3.7 provides considerable possibilities for designing selection techniques.

As shown in Section 3.2, the time spent on correction phase in virtual reality is rather long, which takes up 34.64% of the total movement time and is about 6 times as long as its counterpart in the real world. There are also significantly more sub-movements involved in the virtual-world correction phase. But the differences in the ballistic phase are relatively small. According to the two-component model, the correction phase comes only when the input device enters the vicinity of the target, indicating that the correction phase only covers a very small percentage of the distance towards the target. It can be concluded that during the correction phase of a 3D pointing movement in the virtual environment, users usually spend a large amount of time to traverse a very small part of the distance. Designing a different interaction technique in such a phase may significantly improve users' pointing efficiency. Furthermore, during the correction phase, the spatio-temporal information is also sufficient to more accurately predict a target among multiple adjacent 3D objects and thus applying interaction techniques in this phase can also lead to lower error rates. Therefore, we propose to apply the interaction techniques to reducing the index of difficulty of a 3D pointing task during the correction phase, *i.e.*, the option ②, ⑤ and ⑧ in table 3.7.

In order to decompose a pointing movement into the ballistic phase and the correction phase on the fly, new movement parsing criteria need to be developed, as the criteria proposed in Appendix C.1 can only function as a post-processing step. In Appendix C.2, we provide 3D real-time movement parsing criteria that include data preprocessing, global peak detection, sub-movement detection, end of ballistic phase determination and target prediction in real time.

3.3.1 Interaction Technique Implementation

Following the methodology, two interaction techniques, named “AutoWidth” and “AutoDistance”, have been implemented. *AutoWidth* is the technique of expanding the target to a fixed size⁵ during the correction phase of the movement. *AutoWidth* takes effect immediately after the moment that the parsing algorithm detects the end of ballistic phase and a target is predicted.

AutoDistance is defined as the technique which drags the cursor toward the predicted target and snaps the cursor to the center of the target immediately after the end of ballistic phase. Its idea is similar to the 2D “snap-dragging”[Bie88]. Once the cursor snaps at the target, it only has 3 DOFs, *i.e.*, the rotation around three axes, and the translation is locked. Under these circumstances, users have to move the stylus in the motor space faster than a predefined threshold d to release it (Appendix C.2). The snap dragging event only takes place in the visual space. In motor space, however, no haptic feedback is provided to automatically bring the tracker to the destination, which may lead to visuo-proprioceptive conflicts. After several training trials, subjects were able to quickly adapt to it. Since the snap dragging event locks the cursor's position during the correction phase, there tends to be a cumulative effect on the difference between the original tracked position and the translated tracked position, which can result in a strong deviation of the hand position from the center of the motor space. This is solved by translating the cursor position back to the original place at the end of each trial.

Freehand is the interaction technique that does not provide any assistance during the point-

⁵The size of the expanded target is set twice as big as the original one.

ing tasks. But the real-time movement parsing criteria have also been applied, with the aim of being compared with AutoWidth and AutoDistance and serving as a benchmark.

3.3.2 Experiment

We have carried out a controlled experiment, in which users were asked to perform the same multi-directional pointing tasks as in Section 3.2.1, using AutoWidth, AutoDistance and Freehand. The spatial and temporal information during the movements were collected. Some aspects of the current experiment, such as the apparatuses and the task, are similar to those of the previous experiment (see Section 3.2.1), and the overlapped part was not repeated.

Apparatus

The experiment was performed in a desktop fish tank virtual environment, which was created with the same apparatuses used in Section 3.2.1, except that the 20-inch CRT was replaced by a Samsung HL67A750 67-inch 3D-capable LED DLP HDTV. During the experiment, the resolution of the display was set to 1920×1080 at 120Hz. The overall end-to-end latency of the VR system was measured to be around 80ms using the same method as in Section 3.2.1.

Subjects

11 males and 7 females, aged from 28 to 45 years (average 32.1), voluntarily participated in the experiment. Half of them were naive users in VR and 9 had previous experience of working with VR. They were all right-handed. 6 of them, including 3 naive users, were invited to do a pilot study with the purpose of acquiring proper thresholds as mentioned in Appendix C.2. The remaining 12 subjects were instructed to take part in the main experiment with thresholds obtained from the pilot study.

Experimental setup

The experiment was performed in a non-co-located 1:1 sized virtual environment (see figure 3.12). Subjects needed to wear the stereo glasses and a helmet onto which the head tracker was attached. They were required to hold the stylus with their dominant hands. The focal point of the camera was set in such a way that the scene was in front of the screen. The center of the visual space was 0.75m in front of the subjects when they were seated, while the motor space was 0.3m from the subject, resulting in a distance of 0.45m between the visual and motor space. The scene resembled the ISO 9241 part 9 pointing task [Smi96] and included a $0.4 \times 0.4 \times 0.28$ sized box encapsulating 12 spherical targets and 1 spherical source, each of which was connected to a semi-transparent vertical column. To enhance the depth perception, the floor of the box was covered by a virtual chessboard.

Procedure

The experiment adopted a repeated-measures design with $12 \times 5 \times 3 \times 18$ (number of targets \times number of repeats \times number of blocks \times number of subjects) trials, among which $12 \times 5 \times 3 \times 6$ were used for the threshold acquisition in the pilot study and $12 \times 5 \times 3 \times 12$ for ANOVA analysis. For each target, subjects performed a pointing movement with 5 times, constituting 60 trials. Trials were then extended to 3 blocks, which used Freehand, AutoWidth and AutoDistance as selection techniques, respectively. We gave trials in a block a random

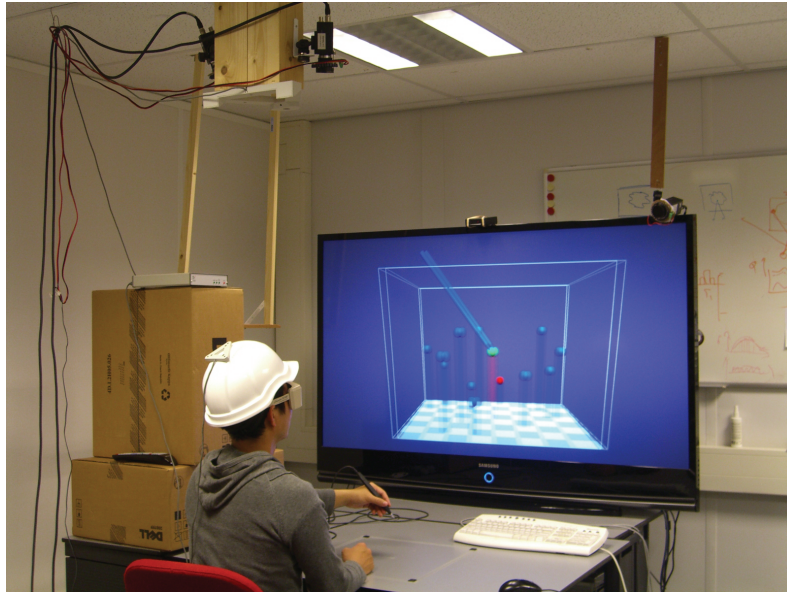


Figure 3.12: The experimental setup: the visual and motor space were non-co-located; C-D ratio = 1.

order which, however, was fixed for the three blocks of the same subject. The pilot study was performed prior to the main experiment, but had the same procedure. To compensate the practice effect, either interference or learning effect, we adopted the incomplete repeated-measures design [SZZ06] where 12 subjects were equally put into 6 groups. Subjects in different groups had to undergo all blocks, but were given in different orders. Before we collected the data, subjects were asked to practice an equal number of trials using each of the three techniques.

3.3.3 Results

All subjects confirmed that both AutoWidth and AutoDistance were much more helpful and easier to control in acquiring the targets than Freehand. 11 out of 12 subjects reported that AutoDistance is more helpful than AutoWidth and 1 reported the other way round.

- **Total Movement Time**

A two-way repeated-measures ANOVA (see Table 3.8) has been performed on the total movement time, treating interaction technique and index of difficulty of the pointing tasks as two within-subject factors. A significant difference ($F(2, 22) = 45.3295$, $p < 0.0001$) can be found between the total movement time of the three scenarios, to which Freehand, AutoWidth and AutoDistance were respectively applied. As shown on the left of Figure 3.13 (cross *ID* and subject analysis), the total movement time has decreased significantly in the scenarios with AutoWidth and AutoDistance, compared to with Freehand. AutoDistance also results in significantly shorter time than AutoWidth⁶. Figure 3.14 (left) demonstrates the ratios by which the total movement time of AutoWidth and AutoDistance has decreased with respect

⁶All three scenarios are significantly different from each other.

to Freehand. After applying the two interaction techniques, the total movement time has been on average improved by 20.67% and 29.43%.

source	SS	df	MS	F	p
<i>technique</i>	29.8022	2	14.9011	45.3295	1.5748e-008
<i>ID</i>	5.3743	11	0.4886	5.1345	1.4248e-006
<i>technique</i> × <i>ID</i>	1.7153	22	0.0780	0.0030	1
<i>technique</i> × <i>subject</i>	7.2320	22	0.3287		
<i>ID</i> × <i>subject</i>	11.5138	121	0.0952		
<i>technique</i> × <i>ID</i> × <i>subject</i>	5.2904e+003	242	25.9332		

Table 3.8: Comparison between the total pointing time: two-factor ANOVA with repeated measures on L/W and *technique*.

• Ballistic Phase Time

A similar two-way repeated ANOVA has been done for the ballistic phase time⁷. The null hypothesis that the means of the three groups are all equal cannot be rejected at the 95% level of confidence ($F(2, 22) = 2.352$, $p = 0.1187$), *i.e.*, no statistical evidence supports that there is a significant effect of the interaction techniques in the ballistic phase. The cross *ID* and *subject* analysis is shown in the middle of Figure 3.13. The decreased ratios of the ballistic phase time by applying AutoWidth (5.58%) and AutoDistance (2.16%) to the movement as shown in the middle of Figure 3.14 can be considered as random effects.

• Correction Phase Time

The ANOVA result on the correction phase time is similar ($F(2, 22) = 107.815$, $p < 0.0001$) to that of the total movement time, except that the differences between Freehand, AutoWidth and AutoDistance are even bigger. As illustrated on the right of Figure 3.13, both interaction techniques lead to significantly shorter correction phase. The percentage of time saved during the correction phase is on average 37.66% and 50.55% (see Figure 3.14, right), respectively.

• Button-Pressed Times

Button-pressed times represent the frequency with which the button of the input device is pressed during a trial. As users were required to press the button one time to initiate a trial and another time to terminate the trial, button-pressed times must be equal to or greater than two. The extra press times may be introduced once users press the button outside the source or the target. It usually occurs during the correction phase when users believe that the button was pressed inside the target, which was found to be outside. By applying AutoWidth and AutoDistance, this chance has been reduced by 2.30% and 3.80% as shown in Figure 3.15.

• Index of Performance Comparison

In the virtual environment, 12 targets were positioned around the source and each target had a different distance to the source. As the size of the targets was kept identical, the pointing tasks were composed of 12 different indices of difficulty ($ID = L/W$). Figure 3.16 depicts the relationships between the movement time and *ID*, and how Fitts' law fits the empirical

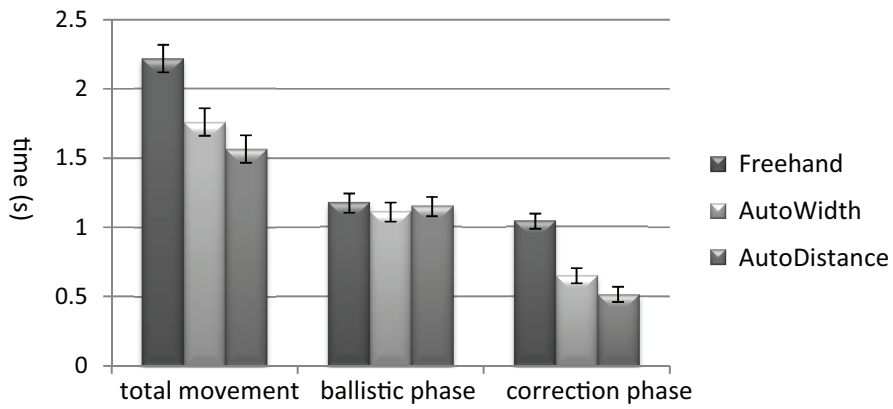


Figure 3.13: Comparisons between the pointing time using Freehand, AutoWidth and AutoDistance in terms of the total movement, the ballistic phase and the correction phase (across *ID* and *subject* analysis).

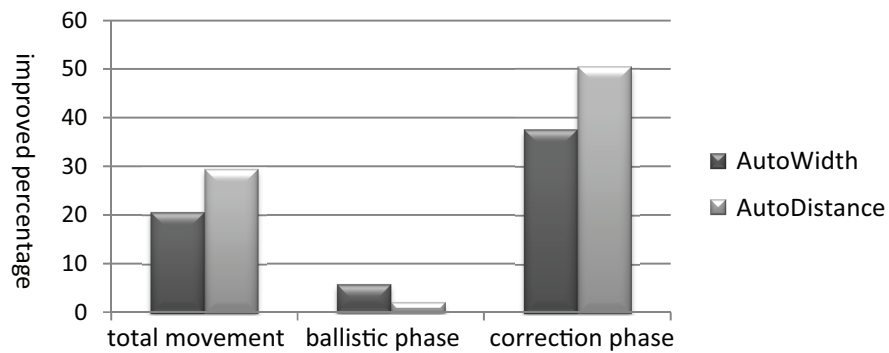


Figure 3.14: Improved percentage of time by AutoWidth and AutoDistance in terms of Freehand.

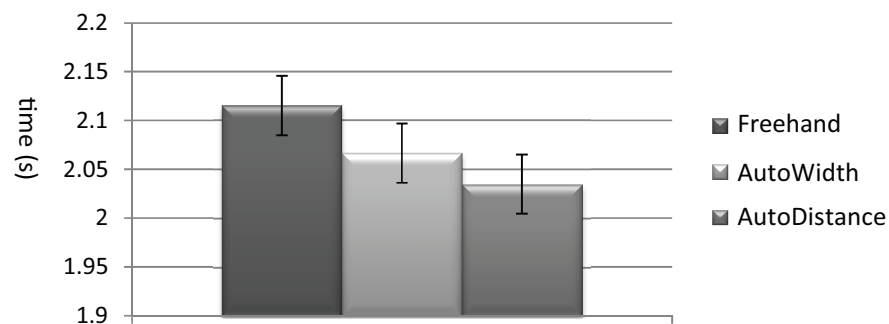


Figure 3.15: Comparisons between the pointing time using Freehand, AutoWidth and AutoDistance in terms of the number of button presses in a trial (across *ID* and *subject* analysis).

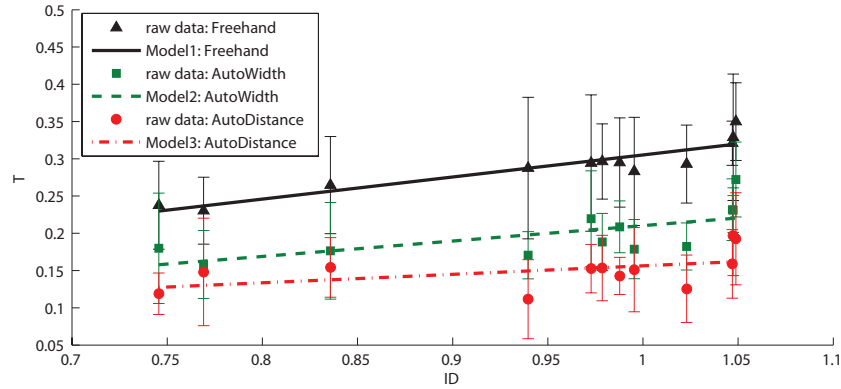


Figure 3.16: Fitting Fitts' law onto the empirical data for each interaction technique.

data with linear regressions for each interaction technique. As demonstrated, the regression models vary in the slope. For the same ID , the pointing task with Freehand takes longer than AutoWidth, and AutoDistance is the most efficient interaction technique. The associate fitting results are shown below:

$$\text{Freehand model : } T = 0.2964ID + 0.0088 \quad (t = 5.3470; p < 0.0001) \quad (3.1)$$

$$\text{AutoWidth model : } T = 0.2063ID + 0.0039 \quad (t = 4.4526; p < 0.0001) \quad (3.2)$$

$$\text{AutoDistance model : } T = 0.1140ID + 0.0424 \quad (t = 2.8449; p = 0.0046) \quad (3.3)$$

The indices of performance ($IP = 1/b$) of the three interaction techniques are 3.3737, 4.8467 and 8.7743 *bits/s*, characterizing how quickly the pointing tasks can be done with the corresponding interaction techniques.

The IP comparison between the three interaction techniques indicates that AutoWidth and AutoDistance can improve the efficiency of pointing to a target in the virtual environment, compared to Freehand. AutoDistance makes users move faster towards the target in a time unit, almost twice as fast as AutoWidth. The IP s for the three interaction techniques described in this chapter can be quantitatively used to compare with other interaction techniques. They are independent of the specific environments and the apparatuses.

3.4 Discussion

In this section, we have used the two-component model to break 3D pointing movements into a ballistic phase and a correction phase in both the real world and the virtual environment. For each phase, the time in the virtual environment is significantly longer than its counterpart in the real world (see Table 3.9 for a summary). In particular, the greatest difference lies in the correction phase, where the virtual-world time is 6.3 times as long as the real-world time.

The differences might be attributed to several reasons. First, the intrinsic difficulty of 3D interaction that occurs in the virtual environment could play a role. It is widely known

⁷The phase division is determined by the real-time movement parsing criteria described in Appendix C.2.

	ballistic	correction
virtual reality	1.04s	0.61s
real world	0.69s	0.11s
virtual / real	1.7	6.3

Table 3.9: A comparison between the time in the real world and in virtual reality.

that depth perception is one of the key issues to be addressed in virtual reality [WRMW95, NCDT10]. Many questions within this area are still under investigation. Particularly, users tend to misestimate the distance in the virtual environment [PKCR05, CrWGT03, RBGA], which may influence the correction phase more than the ballistic phase, since users need to concentrate on the estimation of the distance to the target during the correction phase based on the visual feedback. Moreover, the visual quality of the virtual environment was not as good as in the real world. The scene rendered in our experiment was inevitably of low resolution, contrast, brightness, hue and color saturation, which was far from “photo realistic”. In addition, multimodal interaction was not supported in our experiment. Particularly, the haptic feedback, which may function as one of the important cues for real-life interaction, was missing during the experiment.

Second, the technical issues related to the virtual environment could also be important causes. The VR system utilized for the experiments was measured to be of 45-80ms end-to-end latency, which might lead to users’ poor pointing performance, especially in the correction phase, where the visual feedback plays a more important role than in the ballistic phase. Also, the non-co-located experimental setup (the visual and the motor space were separated) could result in different pointing behaviors than a co-located setup. The pointing efficiency may to some extent be affected.

Last but not least, the learning effect might also contribute to the difference. Users cannot obtain the same amount of practice in the virtual environment as in the real world, as they have been practicing pointing in the real world for years. It might require some time before users get acquainted with the virtual environment and can freely perform pointing tasks.

3.5 Conclusions

Experimental evidence in this chapter has shown that the two-component model can be used to model 3D pointing movements performed in both the real world and the virtual environment, *i.e.*, the movements can be broken into a ballistic phase and a correction phase. The pointing movements collected from the real world were compared to their counterparts from the virtual environment in terms of each movement phase. As the results indicated, significant temporal differences occur in each phase, but the difference is much bigger in the correction phase. There is also a significant difference in the number of sub-movements involved in the correction phase. All in all, the correction phase in the virtual environment is much more time-consuming and includes significantly more small adjustments and corrections.

Fitts’ law and Stevens’ power law can both be used to model 3D pointing movements in the real world and in virtual reality, but the power law outperforms Fitts’ law as it losses less information. Fitts’ law can be converted to the power law with exponent close to 0.42 in case of pointing in the virtual reality, but the exponent is close to 0.20 in the real world pointing tasks.

In addition, it has also been demonstrated that the two-component model, in conjunction with Fitts' law, can be used to develop a methodology, which decomposes a pointing movement into the ballistic phase and the correction phase in real time, and reduces the index of difficulty of the task in each movement phase. With our own insight, the implemented interaction techniques "AutoWidth" and "AutoDistance" were particularly designed to facilitate the correction phase. The experimental results have shown that "AutoWidth" and "AutoDistance" can significantly improve users' efficiency in fulfilling the pointing tasks. This indicates that the methodology offers a new way of designing pointing techniques.

In conclusion, the two-component model has several advantages in modeling pointing tasks. It provides more information than Fitts' law, *i.e.*, two movement phases vs. an overall movement; it provides a better insight into the pointing movements, *e.g.*, movement phases play different roles in pointing, which helped us understand the bottleneck of the pointing movements in VR; it presents a new way of designing interaction techniques by taking movement phases into account.

Chapter 4

Steering

4.1 Introduction

Path steering is a common interaction task in GUIs. Similar to pointing, path steering also requires users to quickly traverse from one location to another. The difference is that path steering is a more constrained movement that needs to be performed within the boundary of a path. A variety of interaction models have been developed for the path steering tasks. One of the commonly accepted formulations is the steering law, which was proposed by Accot and Zhai in 1997. Accot and Zhai's steering law predicts the time of navigating through a path as a function of the path length and width. This relationship has been verified for 2D path steering tasks conducted with 2-DOF input devices (*e.g.*, [AZ01, Pas06, NKK04, GHB⁺06]). For 3D path steering, Zhai et al. [ZAW04] have shown that the steering law can also be used to model locomotion tasks, where users were exposed to a driving simulator and were asked to drive along 3D roads.

In this chapter, the steering law serves as a starting point in the study of path steering for 3D manipulation tasks. The goal is to develop interaction models that can be used to predict users' performance on such tasks. In particular, this chapter aims to address the following questions:

- Can Accot and Zhai's steering law proposed for 2D steering tasks be used to model 3D ball-and-tunnel (see Section 4.2) manipulation tasks? If so, are there any additional independent variables that should be taken into account?
- What is the role of haptic feedback in path steering?
- Should the steering movement of the ball-and-tunnel tasks be modeled as an infinite number of goal-crossing tasks?

These questions are examined with the following approaches. First of all, a ball-and-tunnel task, representing path steering for 3D manipulation tasks, is defined and used throughout this chapter. Then, formal user studies are carried out to verify the steering law on the ball-and-tunnel task. Several important variables that can influence users' temporal performance, including path curvature and orientation (both constant and variable), are experimentally explored and statistically modeled [LMvL10, LbML11, LvL10]. In addition, we

investigate the role of adding force feedback to the steering tasks [LvLK11b, LvLK11a]. An interaction model that introduces the force magnitude as an independent variable is developed for the haptic steering tasks. Finally, the new models proposed in this chapter are discussed and compared to the original steering law, with the purpose of understanding if the ball-and-tunnel task should be modeled as continuous goal-crossing movements.

4.2 Ball-and-Tunnel Task

In [ZAW04], Accot and Zhai posed the question if the steering law could also be used to model users' performance on 3D manipulation tasks. The *ring-and-wire* task as shown in Figure 4.1 was given as an example, *i.e.*, a user navigates a torus with a 6-DOF input device across a trajectory. To perform such a task, one needs to manipulate the ring's position and

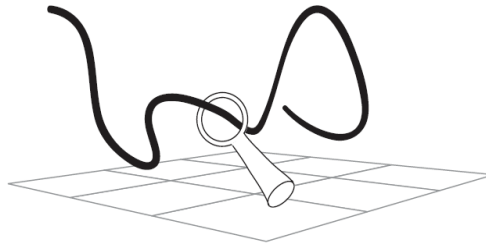


Figure 4.1: The ring-and-wire task defined by Zhai et al. [ZAW04]. It requires the manipulation of both the position and the orientation of the ring, which is substantially more complex than only translating the input device in the 3D space.

orientation simultaneously. In this chapter, we propose to decouple the position from the orientation of the input device, which results in a simplified steering task, the *ball-and-tunnel* task. It only requires a user to translate a ball through a tunnel [LMvL10].

The ball-and-tunnel task is illustrated in Figure 4.2. Using an input stylus, a user is

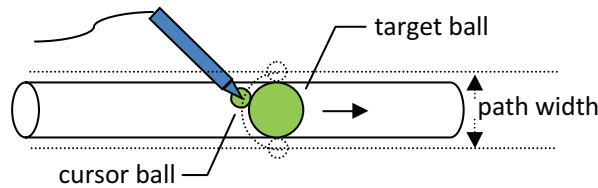


Figure 4.2: The ball-and-tunnel task: a cursor ball pushes a target ball through a tunnel. The orientation of the input stylus does not play a role in the task. Tunnel width = diameter of target ball; Steering path width = tunnel width + $2 \times$ radius of cursor ball = $2 \times$ (radius of target ball + radius of cursor ball)

required to push a virtual target ball from one end of a semitransparent tunnel to the other as fast as possible. The visual feedback of the stylus consists of a pen and a small cursor ball (on the tip of the pen). The cursor ball is used to visually highlight the interaction area between the stylus and the target ball, since it is easier to push the target ball with a volume than with a point. The use of a spherical shape for both the target and the cursor makes sure that they intersect at a point and the orientation of the input device does not play a role during the

steering tasks. The target ball is constrained to the boundary of the tunnel such that the width of the tunnel is defined by the diameter of the target ball. The steering path width is defined as the motion amplitude of the cursor ball with which the cursor ball is in contact with the target ball, *i.e.*, the tunnel width plus two times the cursor ball radius (see Figure 4.2).

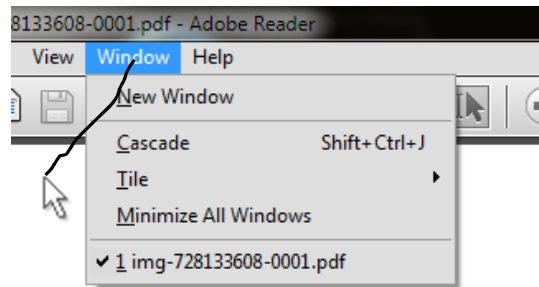
To push the target ball, three requirements must be met:

- The cursor ball must intersect the tunnel;
- The cursor ball must be in contact with the target ball;
- The cursor ball must be positioned on the left of the target ball, *i.e.*, pushing the target ball from the left to the right.

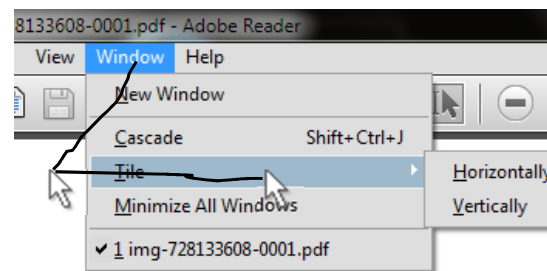
If one of these requirements is not fulfilled, *e.g.*, the cursor is not within the boundary of the path, a correction phase is needed to steer the cursor ball back to the tunnel and continue the task where the user left off. The target ball during this phase remains where it was, which serves as a progress indicator for the task. The total time users spend on correction phases in a complete steering task is defined as *the correction time*.

It is important to note that the ball-and-tunnel task allows for correction once the cursor goes out of the boundary, which is not possible in Accot and Zhai's task [AZ97]. There are many reasons for this design choice. From an application point of view, the steering task substantiated by the ball-and-tunnel task is more widely applied in practice. For example, the original nested menu selection tasks which were considered as an important application of steering tasks by Accot and Zhai [AZ97] have already been adapted to be able to handle the correction phase. As illustrated by Figure 4.3(a), the main menu still remains unfold even if the cursor goes out of the boundary. Users do not have to start over from the very beginning of the task, but are given the opportunity to correct the movement by steering the cursor back to the main menu and continuing the task from where they left off (see Figure 4.3(b)). It seems equally important to measure users' movement time both inside and outside of the boundary. As UI improves, the classical nested menu navigation tasks have substantially been changed from Accot and Zhai's original steering tasks to the steering tasks where the movements inside and outside of the boundary are all considered. Another example stems from the famous racing game "Need for Speed", which is made up of typical ball-and-tunnel-type steering tasks. The scenarios are designed such that once the racing car goes off the track or crashes, the game is not over. Instead, the player is provided the chance to correct the behavior immediately. The same rule applies to the real life car racing. From a VR design perspective, freehand steering without going out of boundary is unrealistically difficult to perform in a virtual environment. Users usually have to repeat one trial many times before they accomplish the task, which easily introduces fatigue and is of no practical significance. Compared to Accot and Zhai's steering task, both UI designers and users would prefer the ball-and-tunnel steering as it is more efficient and easier to control.

There are many other alternative solutions for designing path steering for 3D manipulation tasks. Two of them are illustrated in Figure 4.4, where (a) replaces the tunnel with a line which fastens the target ball and (b) removes the target and cursor ball, making it similar to a 3D curve-drawing task. Figure 4.4(b) is a 3D version of Accot and Zhai's 2D task, in which off-track behavior cannot be corrected. The task as shown in Figure 4.2 is chosen due to its richness in visual feedback and the provision of progress indicator for the steering task. The ball-and-tunnel task which implements the act of pushing a target ball through the tunnel is also different from the act of dragging or being pulled by the target ball.

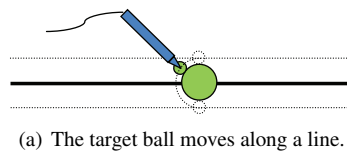


(a) The cursor goes out of the boundary.

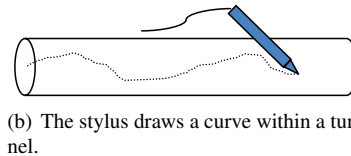


(b) The cursor goes back to the boundary.

Figure 4.3: A nested menu navigation task in Windows 7.



(a) The target ball moves along a line.



(b) The stylus draws a curve within a tunnel.

Figure 4.4: Two alternative solutions of the ball-and-tunnel task.

4.3 Path Curvature and Orientation

For a 3D manipulation steering task, such as the “ball-and-tunnel” task shown in Figure 4.2, it is important to consider the effect of the shape of a path on users’ steering performance. In real-world driving, for example, it is intuitive to slow down when steering the vehicle from a straight road onto a roundabout, and moreover the smaller the roundabout, the more likely we are about to reduce the speed. As a result, navigating through a small roundabout with a given length and width would take longer than through a big roundabout or a straight road with the same length and width (see Figure 4.5). It seems that the movement time is closely related to the degree to which the path is curved, *i.e.*, the path curvature¹.

One of the fundamental distinctions between 2D and 3D steering is how interaction is

¹The curvature of a path, represented by ρ is defined as the reciprocal of the radius of the path’s osculating circle at each point. A circular path, hence, has a constant curvature equal to the reciprocal of its radius at any point. For a path of irregular shape, path curvature needs to be described as a function of the radii of the osculating circles.

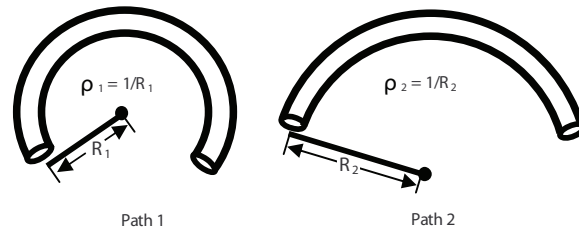


Figure 4.5: Two paths of the same length and width, but of different curvatures. Intuitively, it is easier and faster to steer through path 2 than through path 1.

influenced by the additional dimension (depth) introduced. Conclusive evidence demonstrates that the perception for depth in a virtual environment can be difficult and problematic [WRMW95], which may lead to the fact that users' steering performance on the path positioned more in the depth orientation (along z-axis, see Figure 4.6, right) is different from that in other orientations. Researchers have shown that significant temporal differences arose

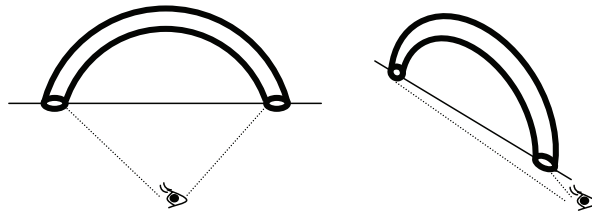


Figure 4.6: Two paths of the same length, width and curvature, but of different orientations.

even in 2D and 3D pointing tasks where users were asked to point at targets that were placed in different orientations with respect to users' viewing direction ([SECK00, MI01]). There is ample reason to believe that 3D steering tasks are more likely affected by the orientation with which the steering paths are placed.

It is still not clear how path curvature and orientation quantitatively influence users' performance on 3D steering tasks. Our goal of this section is to verify if the influence of path curvature and orientation can be introduced into the steering law; if so, how the influence should be formulated quantitatively; and how we can benefit from introducing the additional factors to the original steering law. This is achieved by conducting two experiments, in which users operated a pen input device to navigate a virtual ball through the paths of varying length, width, curvature and orientation. The first experiment aims to address how path curvature affects the steering movement, while the second experiment focuses on the effects of path orientation with respect to the viewing direction.

4.3.1 Experiments

Apparatus and Environment

Both experiments were performed in a head tracked stereoscopic fish tank virtual environment with a 67-inch display (Figure 4.7). The experiments used the same apparatuses as in the previous experiment (see Section 3.2.1).

Subjects were seated 1.35m from the display and were required not to rest their arms

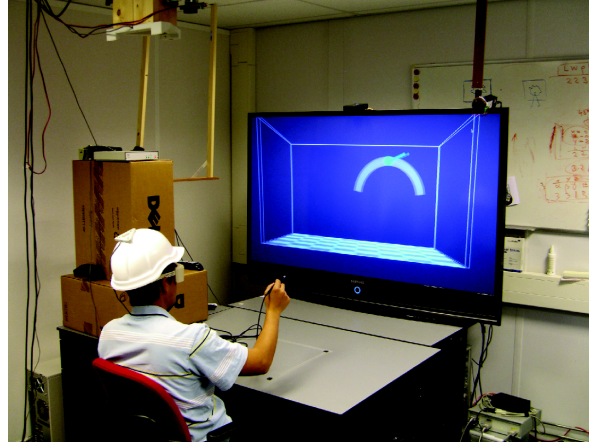


Figure 4.7: The experimental environment: a head tracked stereo display and a 6-DOF input stylus.

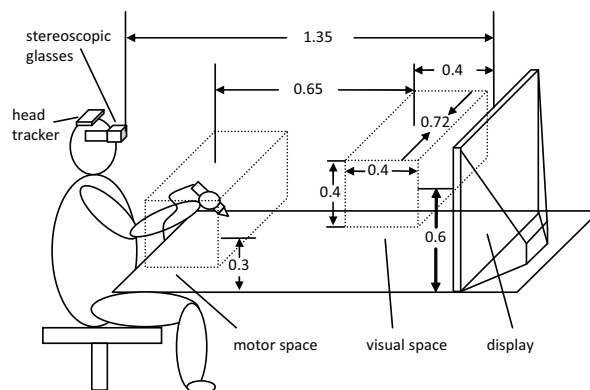


Figure 4.8: The experimental setup (units: meter): Motor and visual space were not co-located, *i.e.*, there is a horizontal offset of 0.65m and a vertical offset of 0.3m between the motor space and the visual space; C-D ratio was set to 1.

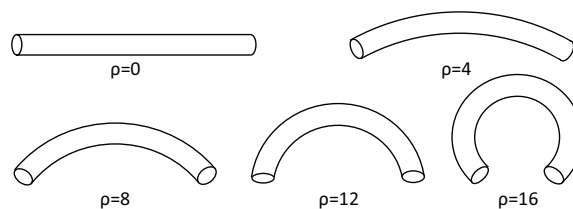


Figure 4.9: Experiment 1: 5 paths of different constant curvatures.

on the table (Figure 4.8). The experiments were designed with non-co-located setups. The origin of the visual space was 0.4m in front of the display and 0.6 above the desktop, while the origin of the motor space was set to 1.05m in front of the display and 0.3m above the desktop. The Control-Display ratio was always kept at 1.

As shown in Figure 4.7, the tunnel was drawn as a semitransparent 3D tube through which the cursor ball and target ball could easily be seen. To enhance the depth perception, we used the stereoscopic viewing, head tracking, head lighting and a $0.72\text{m} \times 0.4\text{m} \times 0.4\text{m}$ wire-frame box with a chessboard floor in the virtual world.

Subjects

All subjects invited to both experiments had normal or corrected to normal vision. None of them was stereo blind.

- *Experiment 1*

12 right-handed subjects voluntarily participated in the experiment. There were 2 females and 10 males, varying in age from 28 to 31. Half of the subjects had previous experience of working with virtual environments.

- *Experiment 2*

4 left-handed and 8 right-handed subjects voluntarily participated in the experiment. 10 of them have worked with virtual environments and 1 was female. They had a mean age of 29.8.

Procedure

- *Experiment 1*

Experiment 1 adopted a within-subject design, in which each independent variable was treated as a within-subject variable. For this, paths of varying length, width and curvature were used and placed in the xy -plane. They were perpendicular to the viewing direction, with the start of the path positioned at the origin. Curvature is defined as $\rho = 1/\text{radius}$, such that a path can be thought of as a segment on a circle of a given radius. The specific values chosen for each independent variable include:

- *Path length (L)*: 0.24m, 0.30m and 0.36m;
- *Path width² (W)*: 0.03m and 0.04m;
- *Path curvature (ρ)*: 0, 4, 8, 12 and 16 m^{-1} .

The lengths of the path were chosen in such a way that a participant needed to traverse a reasonable distance that was not too far to reach by hand. Locomotion with body was strictly prohibited. The five path curvatures (see Figure 4.9) selected were in arithmetic progression and corresponded to the circles with radii of inf (straight line), 0.2500m, 0.1250m, 0.0833m, 0.0625m. The maximum curvature was chosen such that no circular path overlapped when different path lengths were designated. In addition, the curvatures needed to provide a widespread representation and the difference between any two values must be big enough to significantly affect the movement time. Each combination of path

²The radius of the cursor ball was fixed to 0.005m, while the radius of the target ball was either 0.010m or 0.015m, resulting in two path widths, *i.e.*, 0.03m and 0.04m.

length, width and curvature were repeated three times, resulting in $5 \times 3 \times 2 \times 3 = 90$ trials (*curvatures* \times *lengths* \times *widths* \times *repeats*) per subject and 1080 trials in total.

As defined by Accot and Zhai's model, if C is a curved path and W is constant along the path, the index of difficulty for steering through this path is:

$$ID = \int_C \frac{ds}{W(s)} = \frac{L}{W} \quad (4.1)$$

Given 3 path widths and 2 path lengths, it leads to 6 different ID s in the experiment.

- *Experiment 2*

Experiment 2 was also a within-subject design, in which path orientation relative to the viewing direction was considered to be a within-subject variable. Other factors such as path length, width and curvature remained constant. We selected a representative path curvature from the previous experiment ($\rho = 8m^{-1}$, see Figure 4.9) for the current experiment. The choice of the specific path length and width makes sure that the path in this experiment is easier to steer through, *i.e.*, the path is of the smallest ID given L and W from the previous experiment. In this way, it can keep the time of fulfilling the task to a minimum while focusing on the effect of the independent variables under observation. The path was initially positioned in the xy -plane as shown in Figure 4.10, with one end of the path placed at the origin.

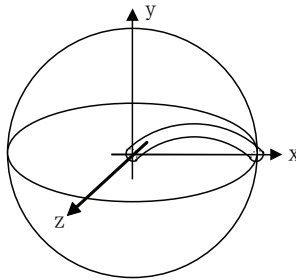


Figure 4.10: Initial position of the path: The path was placed in xy -plane, with one end positioned at the origin and the other end at the intersection of the equator and the x -axis.

then in turn rotated around the y -axis at an angle of β degrees (see Figure 4.11(a)) and the z -axis of α degrees (see Figure 4.11(b)) respectively. In fact, the definition of angle α and β is analogous to that of latitude and longitude in geography. Depending on the selection of α and β , theoretically, we can position the other end of the path at any point on the globe. To obtain a representative end point sampling, the globe was divided into 6 in latitude with 5 planes (see Figure 4.11(b), right) and further into 12 in longitude (see Figure 4.11(a), left) within each of the planes, except the two poles. The specific angles selected for the latitude and the longitude, and the chosen path length, width and curvature are shown below:

- Path length (L): 0.24m;
- Path width (W): 0.04m;
- Path curvature (ρ): $8m^{-1}$;
- Path orientation (latitude α): $-90^\circ, -60^\circ, -30^\circ, 0^\circ, 30^\circ, 60^\circ, 90^\circ$;
- Path orientation (longitude β): $0^\circ, 30^\circ, 60^\circ, 90^\circ, 120^\circ, 150^\circ, 210^\circ, 240^\circ, 270^\circ, 300^\circ, 330^\circ, 180^\circ$.

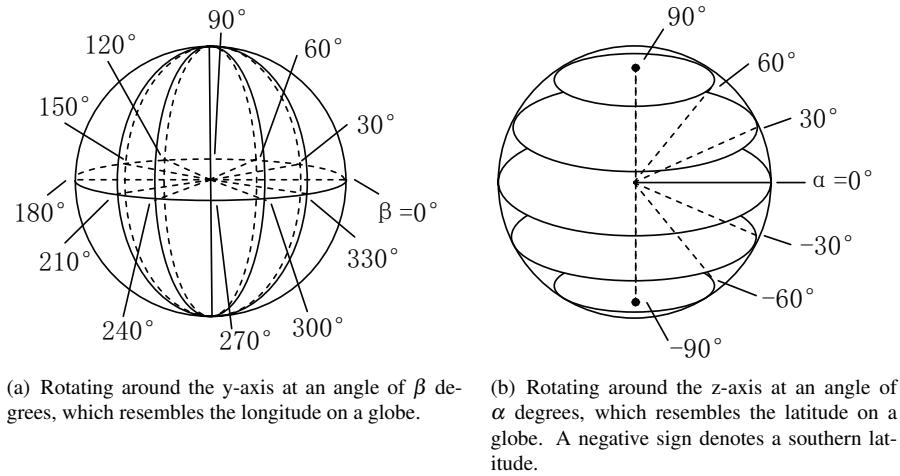


Figure 4.11: Two independent rotating orientations.

There were $12 \times 5 + 2$ points on the globe, which consequently corresponded to 62 path orientations in the 3D space. Each orientation was repeated three times, which made up 186 trials per subject and 2232 trials in total.

To counterbalance the interference and learning effect, the overall trials, including the repetitions, were given in a random order which was different between subjects in both experiments.

4.3.2 Results

This section specifies how the data collected from each of the experiments described above are analyzed using statistical methods, such as ANOVA and regression. Before any analysis, the raw data from both experiments have been logarithmically transformed³ to meet the requirements that are assumed by most of the statistical methods, *i.e.*, independence, normality and homoscedasticity of the data (see Appendix A for details).

Modeling Steering with Path Length, Width and Curvature,

The first observation from Experiment 1 is that about 60-70% of the trials could not be completed by the subjects in a single steering operation. It means that the subjects often did not succeed in keeping their pen within the boundary of the tunnel, and that they needed to spend time to correct for the failure by bringing the pen back within the boundary after wandering off too far. The instances where the pen entered and left the designated boundary were used to define *sub-trials*, and the sub-trial time is the actual steering time spent during the sub-trial.

Figure 4.12 plots the cumulative histogram of the sub-trial path lengths. The fact that path lengths rather than the ones specified in the experimental setup occur is the evidence for our above claim. For this, we have partitioned and analyzed the data such that the total time of a trial is equal to the total steering time of the sub-trials plus the correction time:

$$T_{total} = T_{steering} + T_{correction} = \sum T_{subtrials} + T_{correction} \quad (4.2)$$

³Unless explicitly stated otherwise, the implicit base of the logarithm used in this thesis is 10.

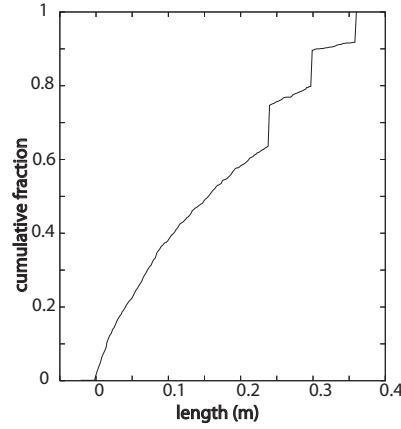


Figure 4.12: Cumulative histograms of path lengths for sub-trials.

Following the modeling methodology proposed in Section 1.3, the modeling procedure is broken into two steps. In the first step, we examine whether a linear relationship between $\log T$ and $\log(L/W)$ can be observed for each path curvature ρ , *i.e.*,

$$\log T = a + b \log\left(\frac{L}{W}\right) \quad (4.3)$$

Note that the coefficient b corresponds to the exponent in Stevens' power law (see Section 2.5). If it does not differ significantly from one, it can be considered as the evidence that the steering law is a statistically valid description of the data (as the steering law assumes the exponent to be one); If the power b is significantly different from one, it can be considered as the evidence that the steering law is not the optimal description of the data.

The second step aims to examine if and how the coefficients a and b of Equation 4.3 depend on ρ . Specifically, the following relationship needs to be verified through the experimental data:

$$a(\rho) = a_0 + a_1\rho \quad (4.4)$$

$$b(\rho) = b_0 + b_1\rho \quad (4.5)$$

A statistical model can be derived by combining Equation 4.3, 4.4 and 4.5:

$$\log T = (a_0 + a_1\rho) + (b_0 + b_1\rho) \log\left(\frac{L}{W}\right) \quad (4.6)$$

These two steps are implemented in terms of sub-trial analysis and complete trial analysis as below.

- *Sub-Trial Analysis*

As shown in Table 4.1, for each path curvature ρ , there is a strong linear correlation between $\log T_{subtrial}$ and $\log(L/W)$ ($p_{model} < 0.05$), and there is no evidence for lack of fit ($p_{lackofit} > 0.05$), which statistically indicates that $\log(L/W)$ is an adequate predictor of the sub-trial time without taking ρ into account. The relationship between the linear model and the empirical data is depicted in Figure 4.13, where (a) is the plot of averaging the effect of ρ and (b) is the plot of splitting the effect of ρ .

ρ	coef. a		coef. b		model		lack of fit	
	Est.	C.I.	Est.	C.I.	F	p	F	p
0	-.328	[-.39, -.26]	.603	[.46, .75]	483.357	< .01	-19.793	1.0
4	-.289	[-.39, -.19]	.582	[.46, .71]	948.835	< .01	-35.006	1.0
8	-.268	[-.35, -.18]	.572	[.36, .78]	636.840	< .01	-23.740	1.0
12	-.254	[-.29, -.22]	.613	[.47, .75]	636.840	< .01	-31.541	1.0
16	-.237	[-.31, -.17]	.590	[.39, .79]	778.111	< .01	-37.100	1.0

Table 4.1: Step 1 for sub-trial analysis: the statistical evidence for the linear regression between $\log T_{subtrial}$ and $\log(L/W)$ for each path curvature ρ (linear model accepted).

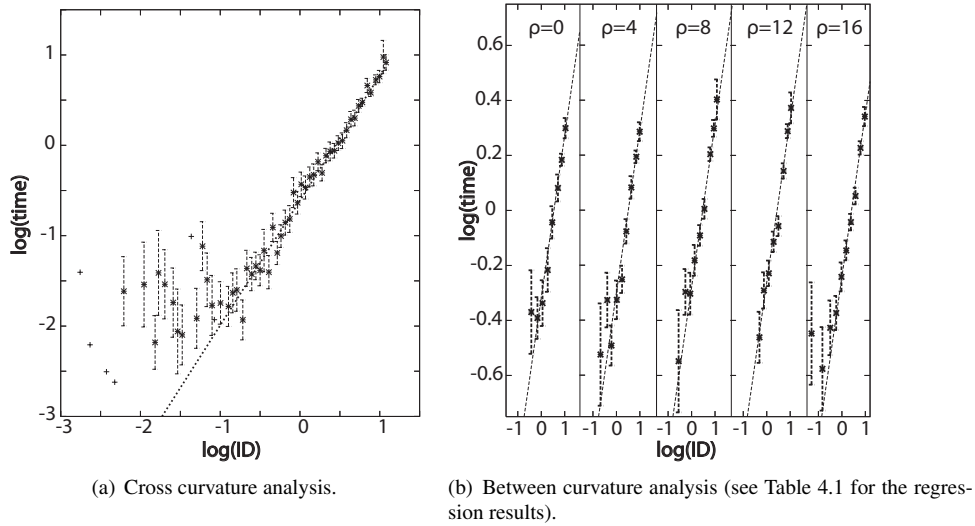


Figure 4.13: The linear regression between $\log T_{subtrial}$ and $\log(L/W)$ using Equation 4.3

The regression parameter estimates for Equation 4.4 and 4.5 are shown in Table 4.2 and 4.3, where all parameters but b_1 are evidenced to be of a significant effect. The interpretation is that the coefficient b is independent of curvature ρ , and only the coefficient a is curvature-dependent (linear). Figure 4.14 and 4.15 illustrate a and b as a function ρ with the 95% confidence intervals derived from the first step. It can be seen from Figure 4.14, both models (linear model and constant) cross all of the confidence intervals, which suggests that a constant model, independent of ρ , is already adequate to describe the data. This seems to contradict the statistical evidence in Table 4.2, where ρ is proven to have an effect. One reason can be attributed to the fact that the confidence intervals are derived from independent analysis. They may become smaller when more observations are provided. It is shown in Figure 4.14, the constant model may fail to cross some of the confidence intervals as the confidence intervals become smaller, while the linear model may still be valid. Figure 4.15 conclusively demonstrates that the coefficient b is independent of ρ .

Based on the analysis above, the model which describes the sub-trial time as a function

	Estimate	t value	P> t	[95% Conf. Interval]
a_0	-0.318	-46.872	< 0.001	[-0.340, -0.297]
a_1	0.005	7.782	0.004	[0.003, 0.007]

Table 4.2: Step 2 for sub-trial analysis: regression parameter estimates for Equation 4.4 (linear model accepted).

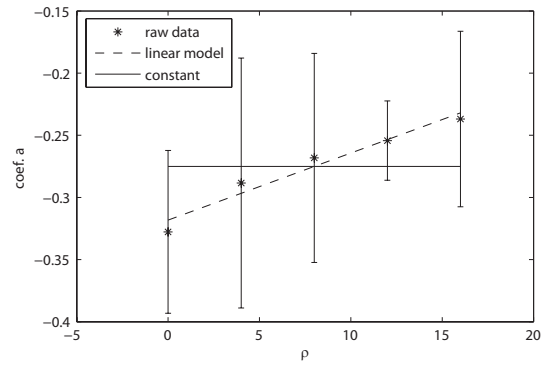


Figure 4.14: Step 2 for sub-trial analysis: the relationship between coefficient a and ρ .

	Estimate	t value	P> t	[95% Conf. Interval]
b_0	0.592	40.407	< 0.001	[0.545, 0.640]
b_1	< 0.001	0.072	0.947	[-0.005, 0.005]

Table 4.3: Step 2 of sub-trial analysis: regression parameter estimates for Equation 4.5 (linear model rejected).

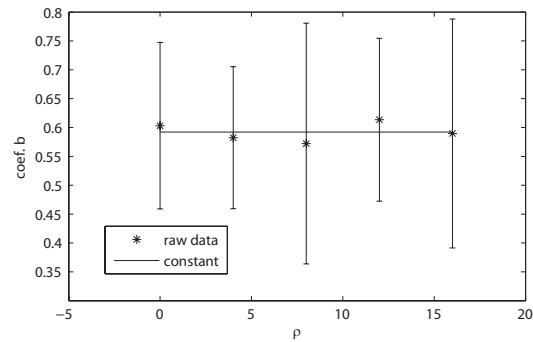


Figure 4.15: Step 2 of sub-trial analysis: the relationship between coefficient b and ρ .

of L/W and ρ can be expressed using the following form:

$$\log T = a_0 + a_1\rho + b_0 \log \frac{L}{W} \quad (4.7)$$

which can be further rewritten as:

$$\log T = a + b \log \frac{L}{W} + c\rho \quad (4.8)$$

In order to validate Equation 4.8, a direct regression using the empirical data is shown in Table 4.4. There is indeed a small, but statistically significant effect of the curvature ($p = 0.027$). According to Table 4.4, the coefficient b which corresponds to the the exponent of Stevens' power law is substantially different from one ($b = 0.592$). This indicates that the steering law proposed for Accot and Zhai's tasks is not a good description of the sub-trial time in the ball-and-tunnel tasks.

	Estimate	t value	P > t	[95% Conf. Interval]
a	-0.318	-27.757	< 0.001	[-0.337, -0.299]
b	0.592	56.930	< 0.001	[0.572, 0.612]
c	0.005	2.211	0.027	[0.004, 0.006]

Table 4.4: Regression parameter estimates on sub-trial time (fitting onto Equation 4.8).

- *Complete Trial Analysis*

First, we perform a two-way repeated-measures ANOVA to verify if $\log(L/W)$ and ρ have significant effects on $T_{steering}$. As shown in Table 4.5, both $\log(L/W)$ and ρ result in significant difference in $\log(T_{steering})$ ($p < 0.05$); no evidence is found for the effect of the interaction term $\log(L/W) \times \rho$. This suggests that the model should include both $\log(L/W)$ and ρ but not the interaction term, which is exactly the same as described in Equation 4.8.

source	SS	df	MS	F	p
$\log(L/W)$	3.080	5	0.616	233.407	< 0.0001
ρ	5.089	4	1.272	10.864	< 0.0001
$\log(L/W) \times \rho$	0.068	20	0.003	0.006	1
$\log(L/W) \times subject$	0.145	55	0.003		
$\rho \times subject$	5.153	44	0.117		
$\log(L/W) \times \rho \times subject$	120.284	220	0.547		

Table 4.5: Comparison between $\log(T_{steering})$: two-factor ANOVA with repeated measures on $\log(L/W)$ and ρ .

To show statistical evidence for the above claim, the two-step modeling methodology is also applied to the steering time for complete trial analysis. As shown in Table 4.6, for each path curvature, a linear relationship between $\log T_{steering}$ and $\log(L/W)$ can be evidenced and illustrated in Figure 4.16. Table 4.7 demonstrates the linear dependency of the coefficient a on ρ , while Table 4.8 shows that the coefficient b is independent of ρ . Therefore, the

ρ	coef. a		coef. b		model		lack of fit	
	Est.	C.I.	Est.	C.I.	F	p	F	p
0	-.630	[-.88, -.38]	.959	[.69, 1.23]	48.279	< .01	.072	.99
4	-.556	[-.82, -.29]	.926	[.64, 1.21]	41.736	< .01	.382	.82
8	-.548	[-.82, -.28]	.962	[.67, 1.25]	43.365	< .01	.286	.89
12	-.512	[-.77, -.25]	.943	[.66, 1.22]	44.171	< .01	.240	.92
16	-.521	[-.83, -.31]	1.04	[.76, 1.32]	52.945	< .01	0.238	.92

Table 4.6: Step 1 for complete trial analysis: the statistical evidence for the linear regression between $\log T_{steering}$ and $\log(L/W)$ for each path curvature ρ (linear model accepted).

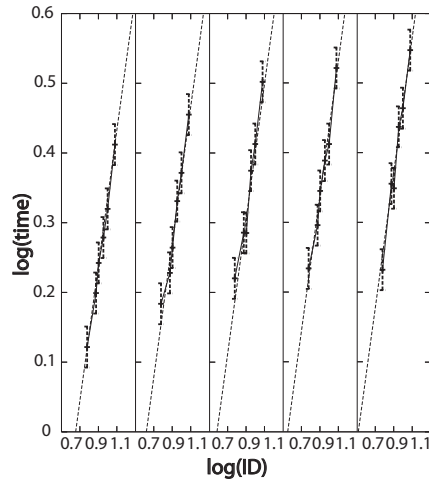


Figure 4.16: Step 1 of complete trial analysis: the linear regression between $\log T_{steering}$ and $\log(L/W)$ for each path curvature ρ (see Table 4.6 for the regression results).

	Estimate	t value	P > t	[95% Conf. Interval]
a_0	-0.606	-32.250	< 0.001	[-0.666, -0.546]
a_1	0.007	3.424	0.042	[0.001, 0.013]

Table 4.7: Step 2 for complete trial analysis: regression parameter estimates for Equation 4.4 (linear model accepted, a is a linear function of ρ).

	Estimate	t value	P > t	[95% Conf. Interval]
b_0	0.930	31.265	< 0.001	[0.835, 1.025]
b_1	0.005	1.481	0.235	[-0.005, 0.014]

Table 4.8: Step 2 for complete trial analysis: regression parameter estimates for Equation 4.5 (linear model rejected, b is independent of ρ).

model as described in Equation 4.8 can also be used to express the steering time for complete trials. Table 4.9 is the related parameter estimates. The fact that the steering law is not able to closely approximate the actual steering taking place during sub-trials doesn't exclude the possibility that the steering law might be an appropriate description for the total steering time. Indeed, the regression parameter estimates in Table 4.9 indicate that the coefficient b close to one provide a good description for the steering time of the complete trials.

	Estimate	t value	P> t	[95% Conf. Interval]
a	-0.634	-12.965	< 0.001	[-0.687, -0.580]
b	0.969	24.057	< 0.001	[0.913, 1.026]
c	0.008	-3.967	< 0.001	[0.007, 0.009]

Table 4.9: Regression parameter estimates on steering time in complete trials (fitting onto Equation 4.8).

- *Error-Free Trial Analysis*

The above analysis focuses on all the trials, including those with correction phase and those without. Since the trials without correction resemble Accot and Zhai' task definition, it would be interesting to examine if the value of the coefficient b is close to one in these trials. Table 4.10 demonstrates the regression parameter estimates using Equation 4.8. Although b in Table 4.10 for error-free trials is closer to one than in Table 4.4, statistical evidence still shows that it is significantly different from one (the 95% confidence interval does not include one).

	Estimate	t value	P> t	[95% Conf. Interval]
a	-0.569	-6.370	< 0.001	[-0.644, -0.493]
b	0.878	9.121	< 0.001	[0.804, 0.952]
c	0.005	3.296	0.001	[0.004, 0.006]

Table 4.10: Regression parameter estimates of Equation 4.8 for error-free trials.

Modeling Steering with Path Orientation

Analogous to the analysis for Experiment 1, users' data collected from Experiment 2 have also been divided into sub-trials due to the correction phase. Fitting the model described in Equation 4.3 onto the sub-trial time, we manage to find a power law with an exponent equal to 0.734 (95% confidence interval: [0.696, 0.772]). The corresponding data are illustrated in Figure 4.17. We have performed a two-way repeated-measures ANOVA on the steering time, taking α and β as two within-subject factors. The statistical result is shown in Table 4.11. Both α ($F(4, 44) = 9.0186, p < 0.0001$) and β ($F(11, 121) = 22.3765, p < 0.0001$) have significant effects on the steering time, but no evidence has been found for the interaction term $\alpha \times \beta$ ($F(44, 484) = 0.0111, p = 1.0000$). Hence, the influence of α and β is separately modeled as follows⁴.

⁴Since L/W for a complete trial does not vary in Experiment 2, the modeling methodology involved in this section is different from that described in Section 1.3

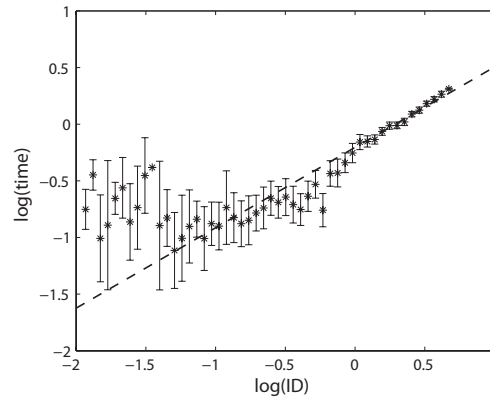


Figure 4.17: The linear regression on transformed sub-trial time using Equation 4.3 (cross curvature analysis).

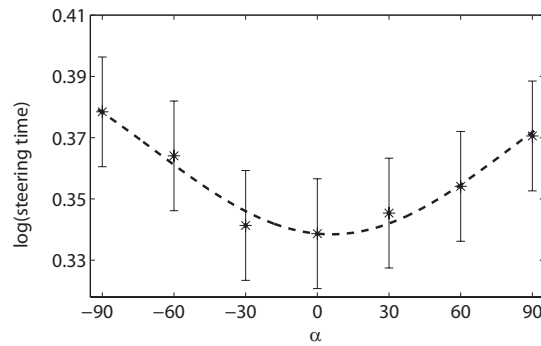


Figure 4.18: Comparisons between the steering time in terms of α (cross β and *subject* analysis). The asterisks represent the empirical data and the dotted curve is the model $\log(T) = a + b \cos(\alpha) + c \sin(\alpha)$.

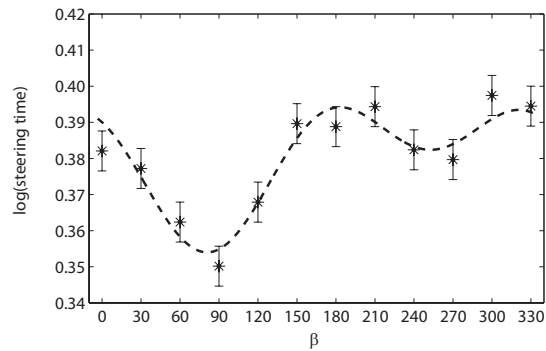


Figure 4.19: Comparisons between the steering time in terms of β (cross α and *subject* analysis). The asterisks represent the empirical data and the dotted curve is the model $\log(T) = a + b \cos(\beta) + c \sin(\beta) + d \cos(2\beta) + e \sin(2\beta)$.

source	SS	df	MS	F	p
α	0.3729	4	0.0932	9.0186	2.0748e-005
β	2.7750	11	0.2523	22.3765	0
$\alpha \times \beta$	1.3293	44	0.0302	0.0111	1
$\alpha \times \text{subject}$	0.4548	44	0.0103		
$\beta \times \text{subject}$	1.3642	121	0.0113		
$\alpha \times \beta \times \text{subject}$	1.3192e+003	484	2.7257		

Table 4.11: The two-way repeated-measures ANOVA table on the steering time.

- *Modeling the Effect of α*

To model the effect of α , β is treated as a cross-subject factor and different β values for the same α are averaged. The relationship between the $\log(\text{steeringtime})$ and α is shown in Figure 4.18. Generally speaking, there is a U-shaped influence where paths of rotating around the z-axis 0° (the paths whose end positions fall on the equator, see Figure 4.11(b)) always result in a valley in the steering time. The steering time increases as the end positions of the paths deviate from the equator and the further the deviation the longer the steering time. There also tends to be a symmetric effect when steering to the south and the north (negative and positive α values), implying that function $\cos(\alpha)$ may play an important role in modeling the effect of α on the steering time. We consider a model with the following form:

$$\log(T) = a + b\cos(\alpha) + c\sin(\alpha) \quad (4.9)$$

The related parameter estimates of Equation 4.9 is shown in Table 4.12, in which a and b are evidenced to have significant effects, but no statistical evidence is found for c .

	Estimate	t value	P > t	[95% Conf. Interval]
a	0.3794	21.3593	< 0.001	[0.3438, 0.4149]
b	-0.0495	-2.1574	0.0352	[-0.0954, -0.0036]
c	-0.0038	-0.5029	0.6169	[-0.0188, 0.0113]

Table 4.12: Regression parameter estimates of Equation 4.9.

- *Modeling the Effect of β*

A similar analysis has been done for the influence of β on the steering time. Figure 4.19 illustrates the relationship between the $\log(\text{steeringtime})$ and β . As depicted, the W-like curve derived from connecting all the asterisks tends to periodically oscillate up and down as β grows. In general, it has two significantly different local minima at paths of rotating around the y-axis 90° and 270° , *i.e.*, the paths parallel to the viewing direction (see Figure 4.11(a)), and two local maxima in the vicinity of 0° (or 330°) and 180° , *i.e.*, the paths perpendicular to the viewing direction. This seems to resemble the characteristics of the function $\sin(\beta)$ or $\cos(\beta)$ except for unequal peaks and valleys, which might be described optimally using a Fourier series expansion. Therefore, we consider a model with the following form:

$$\log(T) = a + b\cos(\beta) + c\sin(\beta) + d\cos(2\beta) + e\sin(2\beta). \quad (4.10)$$

Table 4.13 shows the regression parameters estimates for Equation 4.10. All parameters are evidenced to have significant effects on the $\log(\text{steeringtime})$.

	Estimate	t value	P> t	[95% Conf. Interval]
<i>a</i>	0.3434	101.6252	< 0.001	[0.3366, 0.3501]
<i>b</i>	0.0130	2.6345	0.0109	[0.0031, 0.0228]
<i>c</i>	-0.0285	-6.0908	1.1473	[-0.0378, -0.0191]
<i>d</i>	0.0221	4.3160	6.6742	[0.0119, 0.0324]
<i>e</i>	0.0142	3.1218	0.0029	[0.0051, 0.0233]

Table 4.13: Regression parameter estimates of Equation 4.10.

- *Modeling the Effect of α and β*

This section verifies how steering time is influenced by a combined effect of α and β , given that α and β can be independently used to model the steering time. We propose a model in Equation 4.11:

$$\log T = a + b \cos(\alpha) + c \sin(\alpha) + d \cos(\beta) + e \sin(\beta) + f \cos(2\beta) + g \sin(2\beta) \quad (4.11)$$

which includes a one-level Fourier series expansion on α and a two-level Fourier series expansion on β .

The regression parameter estimates using Equation 4.11 have been specified in Table 4.14. There is conclusive evidence for the term $\cos(\alpha)$, $\cos(\beta)$, $\sin(\beta)$, $\cos(2\beta)$ and $\sin(2\beta)$. It has

	Estimate	t value	P> t	[95% Conf. Interval]
<i>a</i>	0.4027	34.101	< 0.001	[0.3790, 0.4263]
<i>b</i>	-0.0517	-3.394	0.001	[-0.0823, -0.0211]
<i>c</i>	-0.0029	-0.572	0.569	[-0.0129, 0.0071]
<i>d</i>	0.0115	2.485	0.016	[0.0022, 0.0208]
<i>e</i>	-0.0296	-6.731	< 0.001	[-0.0385, -0.0208]
<i>f</i>	0.0225	4.664	< 0.001	[0.0128, 0.0322]
<i>g</i>	0.0146	3.421	0.001	[0.0061, 0.0232]

Table 4.14: Regression parameter estimates of Equation 4.11.

been verified that regressions with one of these terms dropped led to significantly worse results. No evidence, however, could be found for the term $\sin(\alpha)$ ($t = -0.572$, $p = 0.569$). This is predictable from the results demonstrated in Figure 4.18. Therefore, Equation 4.11 could be further simplified as follows:

$$\log T = a + b \cos(\alpha) + c \cos(\beta) + d \sin(\beta) + e \cos(2\beta) + f \sin(2\beta) \quad (4.12)$$

- *The Effect of Handedness*

In a pilot study [LMvL10], we have observed an asymmetric effect, *i.e.*, moving to the left requires more time than moving to the right. This might be attributed to the effect of handedness, since all subjects reported the right hands as the dominant hands and moving to the

right seems to be easier than to the left for the right-handed subjects. As both of the left-handed and the right-handed users were included in Experiment 2, it is possible to identify if handedness of the subjects plays a role in the steering time.

Instead of comparing the absolute steering time of moving to the left and right between the left-handed and the right-handed groups, we have performed a between subjects ANOVA for the steering time of moving to the left (the average steering time to navigate through paths of $\beta = 0^\circ, 30^\circ, 60^\circ, 300^\circ, 330^\circ$) divided by that to the right (the average steering time to navigate through paths of $\beta = 120^\circ, 150^\circ, 180^\circ, 210^\circ, 240^\circ$) to minimize the difference between individuals. Figure 4.20 illustrates the result between the left-handed and the right-handed groups. It can be seen that the mean ratio of the left-handed group is signifi-

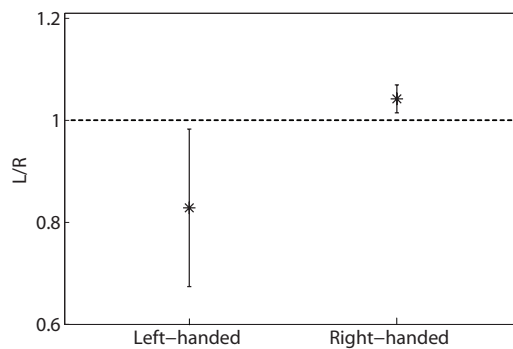


Figure 4.20: The between-subjects ANOVA result between the left-handed and the right-handed subjects on the ratio of left district steering time and right district steering time. The left district includes the paths of rotating around y-axis $0^\circ, 30^\circ, 60^\circ, 300^\circ, 330^\circ$, while the right district is composed of paths of $120^\circ, 150^\circ, 180^\circ, 210^\circ, 240^\circ$. See Figure 4.11(a) on the left.

cantly smaller than that of the right-handed group. Due to the small amount of left-handed subjects involved, the left-handed group shows a larger deviation from the mean. But the right-handed group tends to be closer to the mean and also demonstrates a small difference between steering to the left and right. In addition, the mean ratio of the left-handed group is significantly smaller than one (95% confidence interval does not include one), indicating that for the left-handed subjects steering to the left requires less time than steering to the right. On the contrary, the fact that the ratio is greater than one in the right-handed group specifies that it is faster to steer through paths diverting to the right than to the left. This is evidence showing that handedness⁵ plays a part in steering through paths of varying orientations.

4.4 Varying Path Properties

In Section 4.3, 3D steering on a path with fixed properties was studied. The effect of the path properties, such as path curvature and orientation, on the movement time was modeled under the condition where path properties were changed between the tasks. In practice, it is

⁵Right-handed and left-handed groups were both composed of participants from various countries. Therefore, the effect of handedness observed in our experiment should not be attributed to the influence of culture, though handedness for writing was found to be sensitive to cultural influences [LKHF95, MPME04].

quite usual to observe the case in which users need to perform the steering tasks on property-variable paths. For example, to navigate through 2D nested menus as shown in Figure 4.21(a), a user has to steer through a path of varying width and orientation. Entering and exiting a

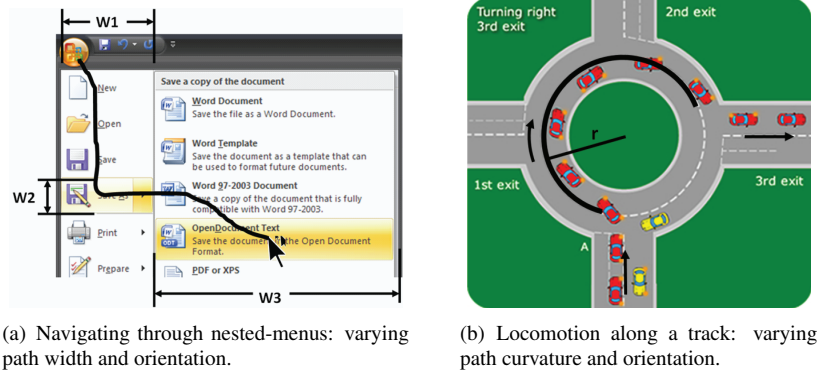


Figure 4.21: Two examples of steering tasks with varying path properties.

roundabout as shown in Figure 4.21(b) requires a user to change the steering orientation and curvature simultaneously. Gaining insight into such tasks, particularly understanding user's motion on each segment of the path and the joints where change occurs is of great practical significance. In addition, the paths as shown in Figure 4.21 can be considered to be a composition of three segments, each of which has different properties. If the number of such segments on a path tends to be infinite, the path can then be thought of as having continuous changes in path properties, *i.e.*, a general 3D path. Therefore, examining the motion under discrete changes in path properties is a move towards the study of steering on a general path.

In this section, we aim to understand the effect of discretely varying path properties, including path curvature, width and orientation, on users' steering performance. Instead of examining users' movement time, the focus is switched to the study of average velocity on a certain segment. The rationale behind the change is that the steering time depends heavily on the segment length which is not always the same within a steering task. To compare users' performance on such segments, it is necessary to introduce other measurements that are independent of segment length. The steering velocity is such a variable that is not constrained by the segment length but rather represents the difficulty of the same type of segments on a path. The comparison between the average velocities of several segments could disclose the difference between the segments. The focus of this section is to qualitatively illustrate the relationship between the path properties and the average velocity, but to derive a precise model is beyond the scope of this section.

4.4.1 Task Design

The *ball-and-tunnel* task was adapted to the current section. The tunnel through which the ball passes is made up of three connected semicircular paths (three segments), each of which can have different path properties. As shown in Figure 4.22, the scenarios include a case where no change was applied to the properties of the segments (scenario (a)), three cases of varying one of the segment properties (scenario (b) - (d)), three cases of varying each two properties (scenario (e) - (g)) and one case where change was applied to all three properties (scenario (h)).

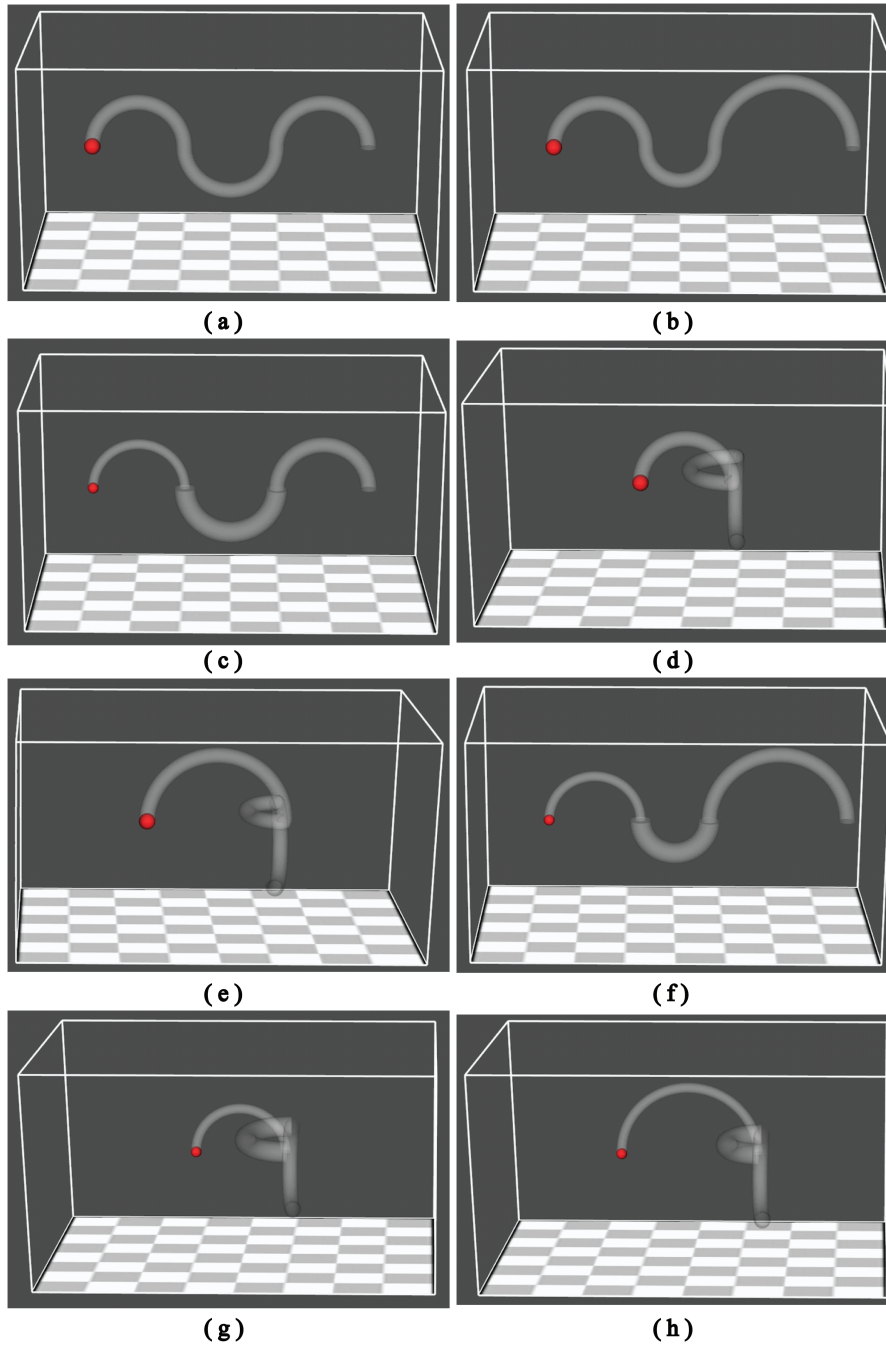


Figure 4.22: Eight scenarios: **(a)** fixed curvature, width and orientation; **(b)** change in curvature; **(c)** change in width; **(d)** change in orientation; **(e)** change in both curvature and orientation; **(f)** change in both curvature and width; **(g)** change in both width and orientation; **(h)** change in curvature, width and orientation.

Each of the path properties under observation can have three different values as shown below:

- path curvature: $8m^{-1}$, $12m^{-1}$ or $16m^{-1}$;
- path width: 0.03m, 0.04m or 0.05m;
- path orientation: xy-plane, yz-plane or xz-plane.

The specific property values in each scenario are demonstrated in Table 4.15

Scenarios	Seg.1			Seg.2			Seg.3		
	$\rho(m^{-1})$	$W(m)$	P	$\rho(m^{-1})$	$W(m)$	P	$\rho(m^{-1})$	$W(m)$	P
(a)	12	0.04	xy	12	0.04	xy	12	0.04	xy
(b)	12	0.04	xy	16	0.04	xy	8	0.04	xy
(c)	12	0.03	xy	12	0.05	xy	12	0.04	yz
(d)	12	0.04	xy	12	0.04	xz	12	0.04	yz
(e)	8	0.04	xy	16	0.04	xz	12	0.04	yz
(f)	12	0.03	xy	16	0.05	xy	8	0.04	xy
(g)	12	0.03	xy	12	0.05	xz	12	0.04	yz
(h)	8	0.03	xy	16	0.05	xz	12	0.04	yz

Table 4.15: The segment property values in each scenario (see Figure 4.22), where ρ , W and P represent path curvature, path width and the plane in which the path lies.

4.4.2 Experiment

The parameters involved in the experiment are:

- dependent variable: average velocity;
- independent variables: change of path curvature, width and orientation.

Subjects

8 right-handed and 4 left-handed users voluntarily participated in the experiment. There was 1 female and 11 males, varying in age from 26 to 35. 10 of them had previous experience of working with virtual environments.

Procedure

The experiment used a repeated-measures design, with $8 \times 5 \times 12$ trials (*scenarios* \times *repeats* \times *subjects*). Each of the 8 scenarios was repeated 5 times for a subject, which resulted in 40 trials per subject and 480 trials in total. Trials were given in a random order, which differed from subject to subject.

4.4.3 Results

Average Velocities between Scenarios

The movement instantaneous velocity is measured along the task axis. The average velocity of a trial on the complete path was calculated, resulting in 40 velocities per subject. For

each subject, the velocities were separated into eight groups based on the scenarios, and were further averaged in each of the groups. Among the eight scenarios, a repeated-measures ANOVA for the average velocities was performed. The corresponding results are illustrated in Figure 4.23.

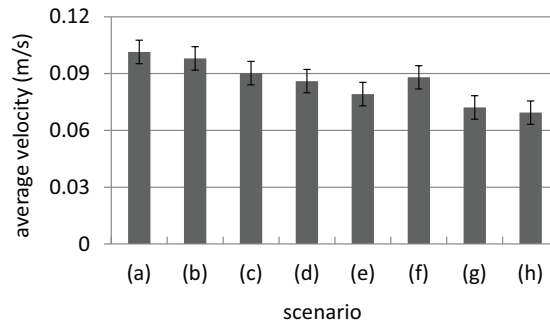


Figure 4.23: Comparisons between the average velocities of 8 scenarios. Scenario (a)-(h) correspond to those in Figure 4.22. A significant difference has been found between scenarios. The decrease in time from Scenario (b)-(h), compared to (a), is 3.3%, 8.95%, 11.11%, 13.52%, 10.44%, 16.86% and 18.59%, respectively.

As shown, there is a statistically significant difference between the average velocities of the eight scenarios ($F(7,77)=4.153$, $p=0.0006$). The average velocity of scenario (a) serves as a baseline for comparison. The results of scenarios (b) - (d) respectively represent the effect of varying path curvature, width and orientation on average velocities. It can be seen that the change in curvature does not significantly influence the average velocities, but those changes in path width and orientation strongly reduce the average velocities, especially the effect of varying path orientation in scenario (d), which leads to a valley in velocity among the three scenarios. The results of scenarios (e) - (h) are more complex due to the combination of varying two or more path properties at a time. The differences between the average velocities of scenario (a) and each of scenario (e) - (h) are all significant. Scenario (f) leads to a relatively high velocity due to the absence of effect of varying path orientation, while the most complex scenario (h) which combines varying all three properties has the largest reduction in velocity among the eight.

It is worth noting that in Figure 4.23, scenarios (d), (e), (g) and (h), each of which involves the effect of varying path orientation, rank the slowest four among the eight scenarios. Additionally, all of them have significantly different average velocities from the baseline (a), *i.e.*, the change in path orientation during a 3D path steering task has the most significant influence on the average steering velocity. Similarly, scenarios (c), (f), (g) and (h), influenced by the change in path width, also significantly slower than the baseline, *i.e.* the change in path width plays a role in affecting the average velocity as well. The fact that scenario (b) which corresponds to the scenario of varying path curvature during the steering tasks is the only scenario that is not significantly different from the baseline illuminates that varying path curvature during a steering task may have the least leverage in average velocity. Generally speaking, the average velocities of the scenarios tend to decline, as the difficulty of the steering task increases (from varying zero to three path properties).

Average Velocities within Scenarios: J vs. S

A 3D path consists of three segments, each of which is a semicircle of a given radius. To examine users' motion on different parts of the path, we partition the path into five segments (see Figure 4.24), in which J_1 and J_2 describe the joints where two semicircles connect and S_1, S_2 and S_3 are the rest of the segments on each semicircle. J_1 and J_2 are made up of

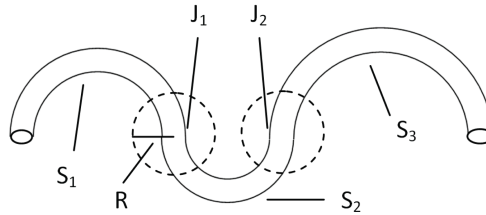


Figure 4.24: Path partition: A 3D path is divided into J ($J=J_1+J_2$) and S ($S=S_1+S_2+S_3$).

the units on the path whose distance are smaller than $R=0.04\text{m}$ from the joints. Figure 4.25 shows us a typical velocity profile for a complete path steering task, in which the subject made a correction while steering on segment S_2 . The fact that the instantaneous velocities

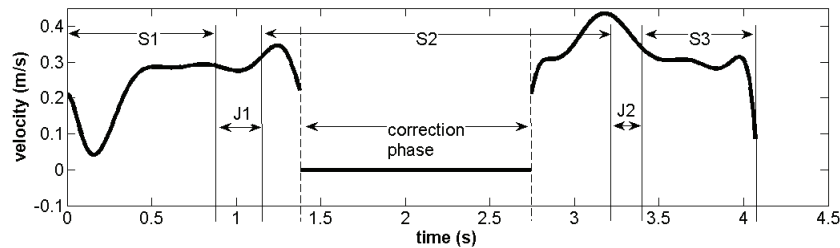


Figure 4.25: A typical velocity profile for a path steering task.

oscillate extensively, rather than slightly fluctuate around a constant indicates that Accot and Zhai's local steering law (Equation 2.4) is substantially not valid for the steering tasks in virtual reality. Due to the wide variety, instantaneous velocity can't be used to distinguish between segments/scenarios, while the average velocity on a segment/scenario represents the characteristic of the segment/scenario.

For the first step, we only focus on the difference between the average velocities on the segments without joints, denoted by S (see Figure 4.24) and those with joints, by J . Figure 4.26 shows the repeated-measures ANOVA results between J and S for each of the scenarios.

Statistically, the average velocities of navigating through J in scenarios (b), (d) and (e) are significantly slower ($F_b(1,11)=17.452$, $p_b=0.0015$; $F_d(1,11)=6.192$, $p_d=0.0301$; $F_e(1,11)=6.868$, $p_e=0.0237$) than that through S . It is the evidence that whenever there is a change in path curvature or orientation along the path, there is a dramatic reduction in velocity during the joint. However, there tends to be no significant difference between the average velocities of J and S if no change appears (scenarios (a), $p=0.2311$) or the path width varies during the steering (scenarios (c), (f), (g) and (h), $p > 0.05$). This may be due to the fact that the path width is an important variable that influences the velocity (see Equation 2.5).

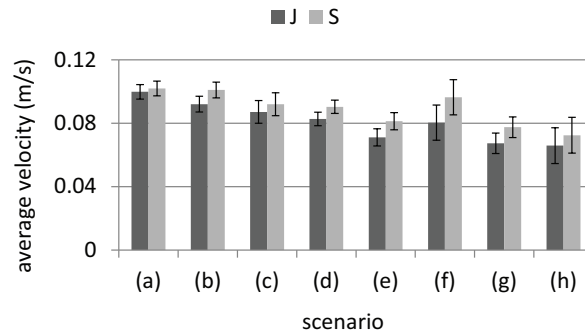


Figure 4.26: Comparisons between the average velocities of J and S. The figure shows eight independent comparisons, each of which focuses on the difference between J and S in the corresponding scenario. Hence, the confidence intervals between scenarios are not comparable. There is a significant difference in scenarios (b), (d) and (e), respectively.

The velocity difference between the three segments S_1, S_2 and S_3 may be significant larger than the difference between S and J .

Average Velocities within Scenarios: S_1, S_2 and S_3

To further specify the results, we split S into S_1, S_2 and S_3 , and J into J_1 and J_2 . A repeated-measures ANOVA between J_1, J_2, S_1, S_2 and S_3 for each scenario was performed and illustrated in Figure 4.27. As depicted, there are significant differences among the average veloc-

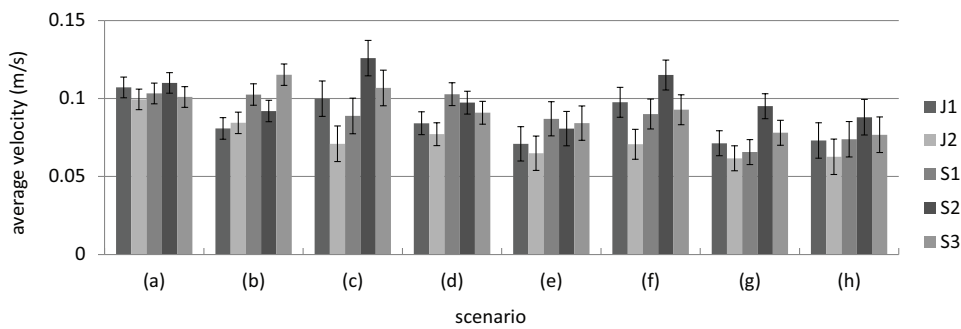


Figure 4.27: Comparisons between the average velocity of J_1, J_2, S_1, S_2 and S_3 in scenario (a) - (h). Significant difference takes place in scenario (b) - (h).

ities of J_1, J_2, S_1, S_2 and S_3 in all scenarios ($p < 0.05$), except in scenario (a) ($F(4,44)=1.521$, $p=0.2125$). In each of the scenarios, the average velocity on J_2 is always the smallest, compared to those on the corresponding J_1, S_1, S_2 and S_3 , except in scenario (b). In scenarios (a), (c), (f), (g) and (h), the average velocity on S_2 is the largest.

Leaving J_1 and J_2 behind, we only focus on S_1, S_2 and S_3 at this stage. It is found that the velocities on S_1, S_2 and S_3 in (b), (c) and (g) are significantly different from one another. In addition, it is possible to find out how the average velocity is influenced by path curvature, width or orientation separately on a certain segment by looking at scenarios (b), (c) and (d)

in Figure 4.27, since only one of the three path properties varies at a time in these scenarios.

In scenario (c), where only the effect of varying path width is included, the average velocities of S_1, S_2 and S_3 are significantly different from each other. Besides, the velocities on each of the segments ($V_1=0.089\text{m/s}$, $V_2=0.126\text{m/s}$, $V_3=0.107\text{m/s}$) are highly correlated to the corresponding path widths ($W_1=0.03\text{m}$, $W_2=0.05\text{m}$, $W_3=0.04\text{m}$), *i.e.*, the wider the segment, the larger the average steering velocity on that segment. This specifies that the path width is a key factor influencing the average velocity in path steering tasks, consistent with Accot and Zhai's local steering law in equation 2.5.

Similarly, scenario (b) in Figure 4.27 is the case that includes a single influence of varying path curvature. As shown, significant differences appear between the average velocities of S_1, S_2 and S_3 . Note that the average velocity of S_2 ($V_2=0.092\text{m/s}$) which has the largest curvature ($\rho_2=16\text{m}^{-1}$) is significantly smaller than that of S_3 ($V_3=0.115\text{m/s}$) which has the smallest curvature ($\rho_3=8\text{m}^{-1}$). It illustrates that the path curvature and the corresponding velocity are highly correlated, *i.e.*, the smaller the curvature of the segment, the larger the average steering velocity on that segment. Therefore, it is conclusive that path curvature is another important factor that influences average steering velocity, next to path width.

As evidence shows, however, there is no significant difference between the average velocity of S_1, S_2 and S_3 in scenario (d). The selection of the three orientations in the experiment (on xy -, yz - and xz -plane) may not be the optimal way to demonstrate the effect of varying path orientation so that more data are required to conclude the effect of varying path orientation on the average velocities of the segments.

4.5 Long Path Steering

The ball-and-tunnel task defined in Section 4.2 includes two movement phases, *i.e.*, a steering phase and a correction phase. A complete path steering task in the virtual environment was allowed to accomplish with correction behaviors when users went beyond the path boundary. In this section, we design a modified version of the ball-and-tunnel task, where the length of the path is prolonged and moreover users are not allowed to make any corrections during the steering task, *i.e.*, a 3D version of Accot and Zhai's 2D steering tasks. Once they cross the boundary of the path, the current trial fails and they need to start over. Under the circumstances, the resulting movements only contain one movement phase, *i.e.*, the steering phase. Our goal is to better understand users' steering performance in such tasks and verify if the model (Equation 4.8) proposed for two-movement-phase steering tasks can be used to model one-movement-phase steering tasks and if Accot and Zhai's steering law proposed for 2D one-movement-phase steering tasks can be used to model 3D one-movement-phase steering tasks that take place in virtual environments.

4.5.1 Experiment

The apparatuses and experimental setup were kept identical as described in Section 4.3.1. The specific experiment design is shown below:

- Independent variables involved:
 - path length: 0.2, 0.4, 0.6m;
 - path width: 0.04m;
 - path curvature: 0, 8m^{-1} .

- A repeated-measures design: $3 \times 1 \times 2 \times 5 \times 12$ (number of path lengths \times number of path widths \times number of path curvatures \times number of repetitions \times number of subjects)

4.5.2 Results

Our first focus is to verify if the steering model proposed in Section 4.3.2 (see Equation 4.8) is valid in describing the steering tasks defined in this section. Equation 4.13 gives the results of fitting Equation 4.8 onto the empirical data and Figure 4.28 plots the relationship, with the effect of path curvature separated.

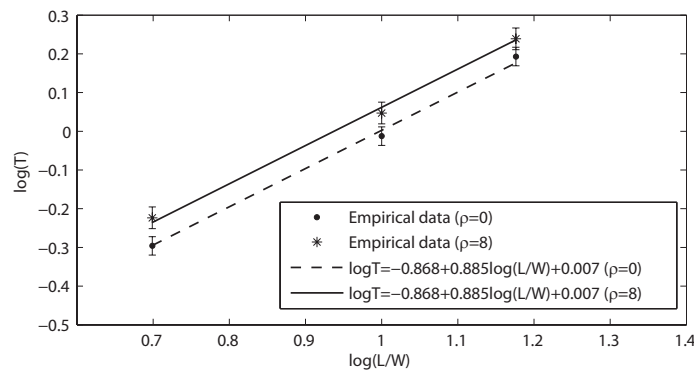


Figure 4.28: Fitting Equation 4.8 onto the empirical data with different path curvatures.

$$\log T = -0.8678 + 0.8847 \log \frac{L}{W} + 0.0074\rho \quad (4.13)$$

The effect of path curvature is once again evidenced by a percentage increase on the movement time. The corresponding regression parameter estimates are provided in Table 4.16. As

	Estimate	t value	P > t	[95% Conf. Interval]
<i>a</i>	-0.8678	-27.343	< 0.001	[-0.9686, -0.7670]
<i>b</i>	0.8847	27.822	< 0.001	[0.7837, 0.9857]
<i>c</i>	0.0074	4.388	0.022	[0.0019, 0.0130]

Table 4.16: Regression parameter estimates of Equation 4.8.

shown, both $\log(L/W)$ and ρ have a significant effect ($p < 0.05$) in modeling the steering time. It is verified that Equation 4.8 could also be used to predict the steering time for the tasks that do not allow the users to make any correction. The coefficient *b*, which corresponds to the exponent of Stevens' power law, is also significantly different from one. It indicates that the steering law proposed for 2D Accot and Zhai's steering tasks is a not a good description of the similar 3D steering tasks in the virtual environment.

The second focus is to examine how movement velocity varies in a 3D steering task. According to the steering law (see Equation 2.5), users' movement velocity should be approximately constant, since the width of the path does not change. Figure 4.29 demonstrates

two typical examples of the steering velocity profiles on a linear path and a circular path, respectively (more examples can be found in Appendix D). As shown, the movement velocity tends to fluctuate as the movement time increases, even if the path width remains constant.

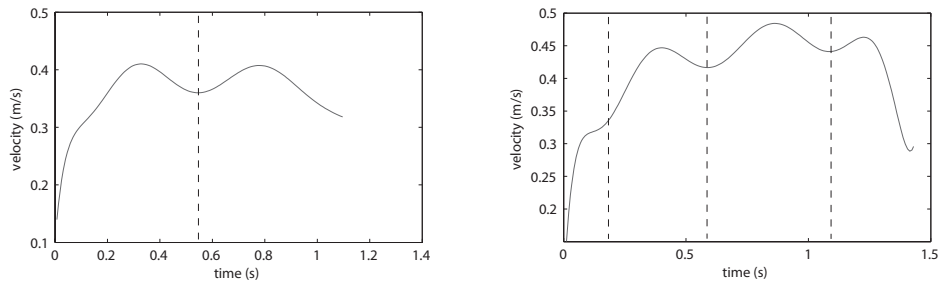


Figure 4.29: Two examples of the velocity profiles. Left: steering on a linear path ($\rho = 0$); right: steering on a circular path ($\rho = 8$).

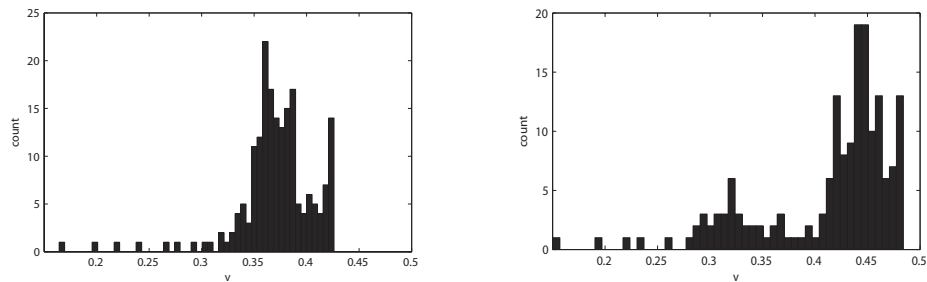


Figure 4.30: The histograms of the instantaneous velocities correspond to Figure 4.29.

Figure 4.30 shows the associate histograms for the instantaneous velocities in Figure 4.29, *i.e.*, the distribution of the instantaneous velocities. It can be seen that the velocities are spread out over a large range of values, rather than close to the population mean.

For each trial, the standard deviation of the instantaneous velocities over the mean σ/μ is calculated, which is used to indicate whether the velocities are of considerable diversity (big value represents rich diversity). The value is in between $[0.066, 0.534]$. We have also performed a two-way repeated-measures ANOVA on σ/μ , in an attempt to examine how L/W and ρ influence the diversity of the velocity. As shown in Table 4.17, there is a significant effect in ρ , indicating that the shape of the path significantly influences the change in the instantaneous velocity. As shown in Figure 4.31, navigating through more curved path results in significantly wider diversity in velocity. As path width always remained constant during the experiment, L/W actually represents the effect of path length on the velocity diversity. As shown in Table 4.17, no statistical evidence is found for L/W and the interaction term $L/W \times \rho$, *i.e.*, path length does not significantly affect the change in velocity. The assumption that users' velocity will tend to be constant as path length grows does not hold.

source	SS	df	MS	F	p
ρ	0.0607	1	0.0607	102.3286	6.5890e-007
L/W	6.4576e-004	2	3.2338e-004	0.3230	0.7274
$L/W \times \rho$	0.0351	2	0.0176	0.0261	0.9743
$L/W \times subject$	0.0220	22	0.0010		
$\rho \times subject$	0.0065	11	5.9285e-004		
$L/W \times \rho \times subject$	14.8103	22	0.6732		

Table 4.17: Comparison between the standard deviation of the instantaneous velocities over the mean σ/μ : two-factor ANOVA with repeated measures on L/W and ρ .

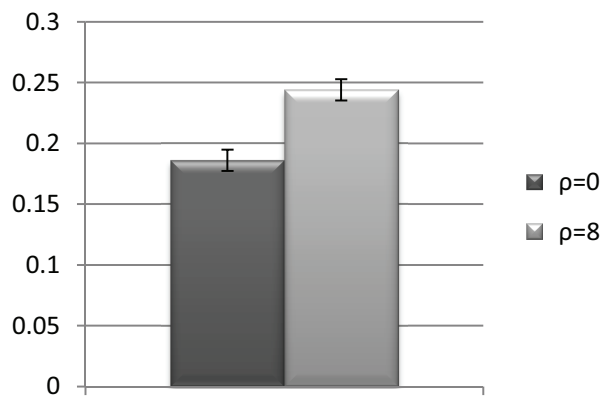


Figure 4.31: Comparison between σ/μ in terms of path curvature (cross *subject* and L/W analysis).

4.6 Haptic Path Steering

Haptic technology or haptics, which adds the sense of touch to the vision dominated solutions, is gaining widespread acceptance as a key part of virtual reality systems. It has been described as the study of “doing for the sense of touch as what computer graphics does for vision” [RDLT09]. Haptic feedback provided in current virtual reality systems mainly includes *force feedback* and *tactile feedback* [LD03, SGS⁺05]. The study of the haptic steering tasks in this thesis focuses on the force feedback.

Within the last decade, the benefit of utilizing force feedback in human-computer interaction has been investigated. As the experimental evidence shows, the adoption of force feedback significantly improved users’ efficiency and accuracy for completing pointing tasks [DY01, KMIH05, VNCL05]. The effect of force feedback on the performance of 2D steering tasks has also been extensively studied. For example, Dennerlein et al. [DMH00] obtained a 52% improvement in efficiency by introducing force feedback to a 2D tunnel steering task. Ahlstrom [Ahl05] used a combination of Fitts’ law and the steering law to model a 2D haptic nested menu selection task and derived that the force field they provided could decrease the selection time on average by 18%. However, the merits of introducing force feedback to 3D steering tasks have rarely been reported. It is not clear if the application of force guidance in 3D manipulation tasks may help us obtain the same improvement as in 2D tasks.

In addition, a large number of studies focused on how force feedback should be rendered to achieve efficient interaction (*e.g.*, [MPT99, AS99, MS94, SBM⁺95]), while few examined how much force feedback, *i.e.*, the amount of force magnitude, is appropriate for the interaction. Does a stronger force magnitude always result in better performance than a weaker magnitude? Is there an interval of force magnitude, within which user's performance can be significantly improved? Gaining knowledge of force magnitude in steering tasks may help us control the amount of force feedback exerted to the tasks and make better use of the haptic devices.

In this section, we introduce two types of force feedback to 3D steering tasks in two separate experiments. The first type of force is applied in such a way that any deviation from the tunnel center was proportionally pulled back based on the distance one deviated, while the second force is exerted in a more realistic way where users encounter the resistance as they push a ball through a tunnel and receive the assistance as being pulled or attracted by the ball. We aim to achieve a twofold goal in such haptic steering tasks. First, it is targeted at examining how users' performance when doing a steering task is affected in the presence of the first type of force feedback, and how the effect of force feedback can be determined by the available interaction models. In the second experiment, we focus on how the amount of force feedback influences users' performance in the presence of the second type of force feedback and how the amount can be modeled as an additional predictor for users' movement time.

4.6.1 Types of Force Feedback

The *ball-and-tunnel* task was again adapted to the current haptic steering task and two types of force feedback were implemented using the Novint Falcon.

Type 1: Force Guidance

Subjects are instructed to hold the Falcon's detachable grip, represented by the cursor ball in the virtual environment, to push the target ball through a 3D tunnel, which is always positioned in the x-y plane with depth of zero. Depending on the particular block, smooth and adaptable force feedbacks can be applied to the Falcon's grip, which are further passed to subjects' hands during the steering phase. The aim is to assist the subjects, to some extent, in keeping the cursor ball within the boundary of the path. In practice, subjects feel dragged slightly toward the center of the tunnel once they deviate from the center of the tunnel but are still within the tunnel. The magnitude of the force is proportional to the distance that the cursor ball deviates from the center of the tunnel (see Figure 4.32) and is computed by Hooke's law, $F = kD$, where k resembles a spring constant ($k = 50N/m$ in the experiment), D is the distance ($D \in [0, 0.015m]$) and F is the force magnitude. The direction of the force is from the center of the cursor ball to the nearest point on the center of the tunnel.

Type 2: Resistance and Assistance

As defined in Section 4.2, the steering task was divided into the steering phase and the correction phase. In the steering phase, Falcon generates smooth and continuous force feedbacks as subjects steer the cursor ball and push the target ball through the tunnel. As shown in Figure 4.33, the force (F3) is a resultant force that is composed of a reacting force (F1) exerted by the target ball, equal and opposite to the force that is exerted to the target ball by the cursor ball, and a friction (F2) at the point where the target ball and the cursor ball contact.

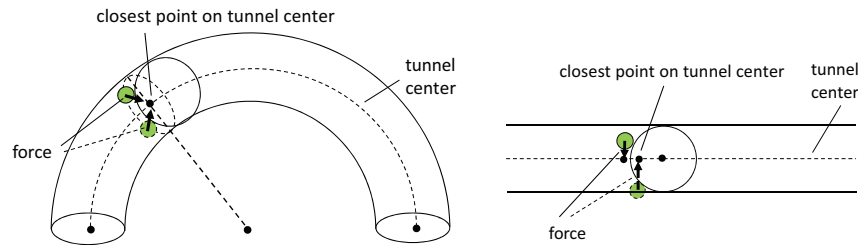


Figure 4.32: Type 1 force feedback provided during the steering phase. Left: 3D view of circular path; right: 2D view of linear path. Any deviation from the tunnel center was pulled back by a force that was proportional to the distance of the deviation.

The sense of touch is simulated in such a way that subjects feel almost realistic as they hold

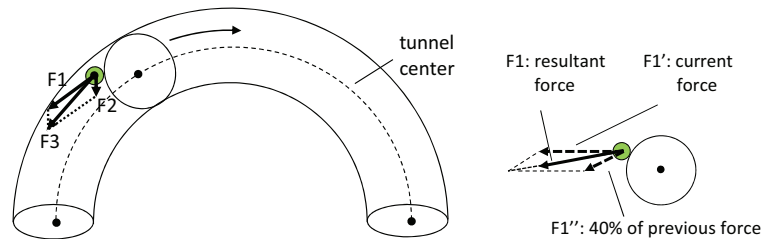


Figure 4.33: Type 2 force feedback provided during the steering phase. The figure on the right shows the damping factor introduced to the system, which takes 40% of the previous force ($F1''$) magnitude and direction into account and results in a new resultant force ($F1$) by combined with the current force ($F1'$). Note the magnitude of $F1$ remains the same.

the cursor ball and push against the target ball. A constant magnitude is applied to $F2$ and the direction is along the tangent line to the target ball at the point where the target ball and the cursor ball contact. Though the magnitude of $F1$ is fixed in one trial during the experiments, it can vary among different trials. The direction of $F1$ is determined by the movement direction of the cursor ball controlled by users. Our aim is to investigate how the movement time is affected by the magnitude of $F1$. To avoid jerky movements, a damping factor is introduced to the force, which also takes the previous force direction into account.

The effect of $F1$ in such a task simulates a resistance feedback. During the experiment, we also implement an assistance feedback that is equal but opposite to $F1$. Under the circumstances, the cursor ball and target seem to have the opposite magnetic fields and the cursor ball is attracted by the target ball such that users feel dragged along the tunnel.

4.6.2 Experiments

We have run two controlled experiments in which type 1 and type 2 force feedback were applied, respectively. In the first experiment, the tasks were accomplished in two blocks, one with force feedback (type 1) turned on and the other one turned off. The purpose was to collect users' data under the two conditions, which can then be used to make a comparison. In the second experiment, users were required to steer through paths of different length and width with varying force magnitude (type 2), in an attempt to model the steering tasks with force magnitude.

Apparatus and Environment

Both experiments were performed in the same desktop VR system as described in Section 3.2.1. All apparatuses were kept the same, except that the magnetic tracking system “Polhemus FASTRAK + tracked stylus” was replaced by a haptic device, the Novint Falcon. The Falcon was updated at around 700Hz⁶; the scene was rendered at 60Hz; the data were logged 120 times per second.

As illustrated in Figure 4.34, the experiments were conducted in a non-co-located virtual environment. There was a distance of 0.65m between the centers of the visual and motor



Figure 4.34: The experimental environment: non-co-located; C-D ratio = 1/5.

space. To compensate for the large distance between the subjects and the display and the small motor space of the device (around $0.10\text{m} \times 0.10\text{m} \times 0.10\text{m}$), the C-D ratio [CVBC08] was set to 1/5 to visually expand the scene.

Subjects

All subjects had normal or corrected to normal vision. None of them was stereo blind.

- *Experiment 1*

There were 14 right-handed subjects (11 males and 3 females) voluntarily taking part in the experiment. 10 of them had previous experience of working with virtual environments. The subjects aged from 25 to 38, with an average of 32.3.

- *Experiment 2*

12 right-handed subjects voluntarily participated in the experiment. 2 of them were females and 8 had previous experience of working with virtual environments. The subjects' age ranged from 28 to 35, with an average of 31.7.

⁶Our sense of touch is far more sensitive than our visual system. In graphics a refresh rate of 30fps is quite acceptable, while in haptics it is widely believed that a response frequency of 300-1000Hz is needed to ensure accurate interaction [De198, PLDA00]. In principle, the Falcon should have been approximately updated at 1000Hz. Due to other computing, *e.g.*, scene rendering, its frequency during the experiment was reduced to some 700Hz, which, however, still managed to provide a consistent and smooth sense of touch.

Procedure

- *Experiment 1*

Two blocks were conducted, including one with type 1 force feedback (block 2) and one without (block 1). We adopted the repeated-measures design in each block, introducing paths of different length, width and curvature. The specific settings for each property include:

- path length (L): 0.24, 0.30 and 0.36m;
- path width (W): 0.03 and 0.04m;
- path curvature (ρ): 0, 4, 8, $12m^{-1}$.

Each condition (a combination of path properties) was repeated 3 times, resulting in $3 \times 2 \times 4 \times 3$ trials ($L \times W \times \rho \times repetition$) per subject. There were in total 2016 trials for 14 subjects in two experiments.

To compensate the practice effect, half of the subjects were asked to do the experiment without force feedback (block 1) first, while the other half with force feedback (block 2) first.

- *Experiment 2*

The experiment was also of a repeated-measures design, in which paths of varying length and width, and type 2 force (F) with varying magnitude were introduced. The specific settings for each property included:

- path length (L): 0.24, 0.30 and 0.36m;
- path width (W): 0.03 and 0.04m;
- force magnitude (F): -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8, 1.0N.

A positive F actually indicates how much resistance is applied to the subjects while they are pushing the ball. The greater the value, the bigger the resistance. Intuitively, a negative F implies how much assistance is applied to the task. This is the effect of being attracted by the target ball. The greater the absolute value, the bigger the assistance. Each condition (a combination of path properties and force magnitude) was repeated 3 times, resulting in $3 \times 2 \times 9 \times 3$ trials ($L \times W \times F \times repetition$) per subject. There were in total 1944 trials for 12 subjects.

Subjects were required to practice an equal number of trials under each condition before starting both experiments. Trials were presented in a random order that differed from one subject to another. Subjects were allowed to have a break whenever they suffered from fatigue between trials. This was, however, strictly prohibited during a trial.

4.6.3 Results

Experiment 1: Comparing Non-Haptic and Haptic Steering

- *Users' Performance Comparison*

In this section, we compare users' performance between the two blocks of Experiment 1 in terms of the steering time, the correction time, the trial completion time, the error-free trial percentage, the number of corrections in a trial and the sub-trial time.

Three independent three-way ANOVAs with repeated measures on three factors, *i.e.*, F (on/off), L/W and ρ , have been performed to respectively compare the steering time, the correction time and the trial completion time between the two blocks. Table 4.18 demonstrates

the ANOVA result for the steering time. Significant effects have been found for F , L/W , ρ , $F \times L/W$ and $L/W \times \rho$. In particular, it is evidenced that the steering time is significantly affected by turning the type 1 force feedback on and off ($F(1, 13) = 90.935, p < 0.0001$).

source	SS	df	MS	F	p
F	3119.157	1	3119.157	90.935	< 0.0001
L/W	1048.323	5	209.665	97.371	< 0.0001
ρ	320.785	3	106.928	56.852	< 0.0001
$F \times L/W$	219.611	5	43.922	41.526	< 0.0001
$L/W \times \rho$	37.225	15	2.482	3.165	0.0001
$F \times \rho$	9.642	3	3.214	2.829	0.0509
$F \times L/W \times \rho$	9.497	15	0.633	0.874	0.5942
Error(F)	445.911	13	34.301		
Error(L/W)	139.961	65	2.153		
Error(ρ)	73.352	39	1.881		
Error($F \times L/W$)	68.750	65	1.058		
Error($F \times \rho$)	44.312	39	1.136		
Error($L/W \times \rho$)	152.912	195	0.784		
Error($F \times L/W \times \rho$)	141.273	195	0.724		
Between-Subjects	1778.374	13			
Within-Subjects	5840.353	658			

Table 4.18: Comparison between the steering time: three-factor ANOVA with repeated measures on F , L/W and ρ .

As shown in Figure 4.35, the steering task with force feedback has significantly decreased the steering time, compared to that without force feedback⁷. The presence of force guidance saves on average 56.25% (95% confidence interval: [44.45%, 68.05%]) of the time subjects spent on steering within the boundary of the path. The correction time with force feedback has also significantly been decreased ($F(1, 13)=24.828, p = 0.0003$) by 69.00% (95% confidence interval: [46.26%, 91.74%]). In addition, a significant difference has also been found between trial completion times ($F(1, 13)=67.730, p < 0.0001$). The total efficiency has been improved by 60.23% (95% confidence interval: [46.27%, 74.19%]) with force guidance.

We have collected data of 1008 trials in each block. As mentioned, a complete steering trial can be broken into several sub-trials due to the presence of the correction phases. This can be evidenced by the fact that in block 1 and 2, 919 and 731 out of 1008 trials contains more than one sub-trial. The opportunity of completing a task without any correction is only 8.83% and 27.48% (see Figure 4.36, left). The number of error-free trials in block 2 is more than 3 times of block 1. As shown, force guidance has dramatically increased the opportunity of error-free trials in the path steering tasks.

We have also performed a three-way repeated-measures ANOVA on the average number of corrections in a trial between the two blocks. As depicted in Figure 4.36 (middle), there is a significant difference between the two blocks ($F(1, 13)=43.030, p < 0.0001$) and the number of corrections in a trial dramatically falls by 78.27% (95% confidence interval: [96.93%, 59.61%]) with force feedback introduced. This specifies that the accuracy of performing the path steering tasks has been substantially improved with the force guidance.

⁷The ANOVA results shown in this section are derived from the transformed data, while the figures associated with the results are plotted with the raw data for intuitive demonstration.

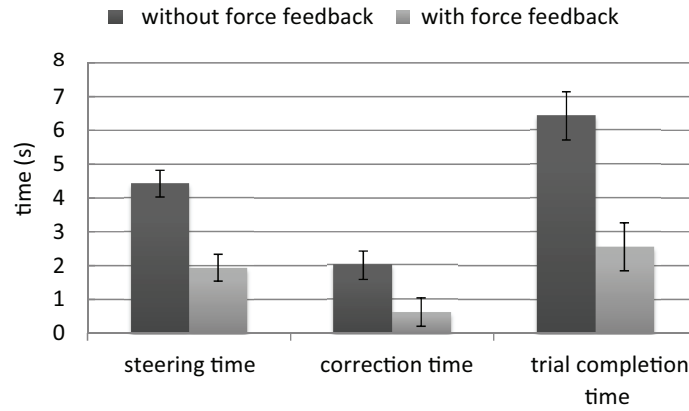


Figure 4.35: Comparisons between the non-haptic and haptic steering tasks in terms of the steering time, correction time and trial complete time (cross L/W , ρ and *subject* analysis).

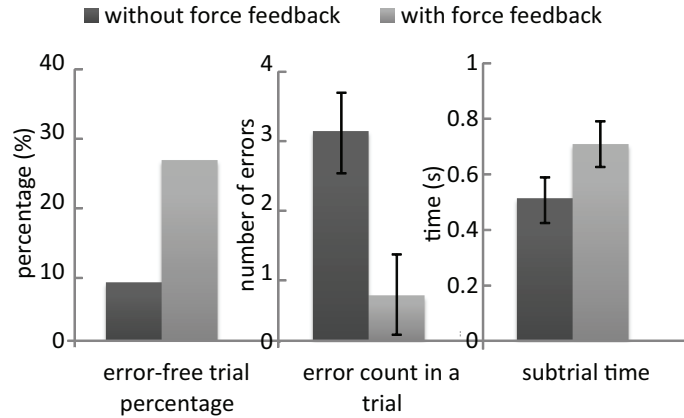


Figure 4.36: The error-free trial percentage and the average error count / average sub-trial time in a trial (cross L/W , ρ and *subject* analysis).

As shown on the right of Figure 4.36, an interesting observation is that the time subjects spent on steering a segment of the path without any correction in a trial (sub-trial time) grows significantly ($F(1, 13)=14.553$, $p = 0.0021$, increased by 27.48% [10.49%, 44.47%]) with force guidance. Since sub-trial time is the ratio of sub-trial length to sub-trial velocity, the growth of sub-trial time can be attributed to the fact that the increase in the sub-trial length exceeds that in the average sub-trial velocity.

- *Model Comparison*

In this section, we aim to verify the steering law (Equation 2.3) and its variation (Equation 4.8) for a haptic input device (Novint Falcon) and examine what the difference is, if any, between the models of non-haptic and haptic tasks.

The first step is to verify the steering law in both non-haptic and haptic scenarios. This implies that the path curvature is not considered as an independent variable and the steering time for navigating through paths of different curvatures needs to be averaged for the

same ID . Figure 4.37 demonstrates the result of fitting Equation 2.3⁸ to the empirical data. Statistically significant differences have been found between the steering⁹ time of various

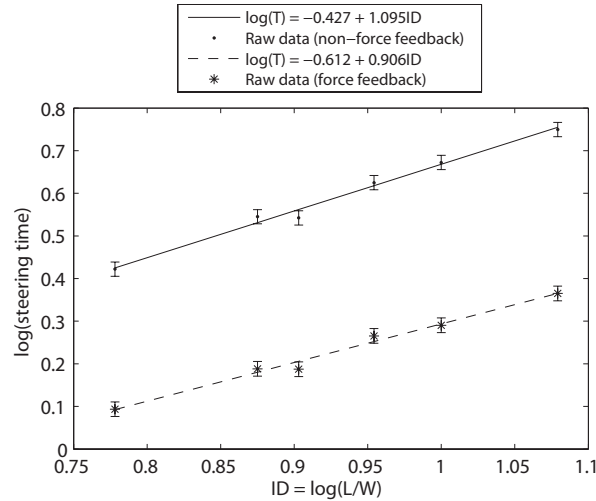


Figure 4.37: Classical T vs. ID plot, where $ID = \log(L/W)$. Solid line represents the model fitting on non-force-feedback data, while dotted line represents the model on force feedback data.

ID s in both blocks ($F_1(5, 65)=217.819$, $p_1 < 0.0001$; $F_2(5, 65)=140.865$, $p_2 < 0.0001$). As shown, the two lines, described by different intercept a and slope b , do not intersect within the observed range. For the same ID , the path steering task without force feedback is quantitatively illustrated to be more time-consuming than that with force feedback. Moreover, there is a strong linear correlation between $\log(\text{steeringtime})$ and $\log(L/W)$ in both blocks, which can be evidenced by the goodness of fit of the linear regression in block 1 (solid line: $R^2 = 0.9653$) and block 2 (dotted line: $R^2 = 0.9731$). It indicates that the steering law holds in both non-haptic and haptic scenarios and that the only difference is the coefficients for Equation 2.3.

Figure 4.38 illustrates the combined effect of $\log(L/W)$ and ρ on the steering time, which is also a result of fitting the variation of the steering law (Equation 4.8) on non-haptic and haptic data, respectively. As shown, the two lines in Figure 4.37 have been extended into two surfaces (planes described by different constants a , b and c shown in the legend of Figure 4.38), with all curvatures involved, and the fitted planes are approximately parallel to each other. According to the two models, steering with force guidance is more efficient than without (the plane with force feedback is lower), for a specific combination of $\log(L/W)$ and ρ . The improved efficiency of the steering tasks with force guidance, compared to the non-force-feedback case, can be quantitatively modeled as a (approximate) downward translation from the non-haptic plane to the haptic plane. The distance translated may strongly rely on the spring constant k selected in the experiment. A greater k , representing a bigger force, should result in a larger distance.

⁸ T and ID has been logarithmically transformed before fitted onto Equation 4.37. The transformation can be considered as data preprocessing since a direct regression between T and L/W strongly violates the assumption. In this section, the linear regression between $\log T$ and $\log L/W$ which corresponds to the first step of our modeling methodology is thought of as Fitts' law verification.

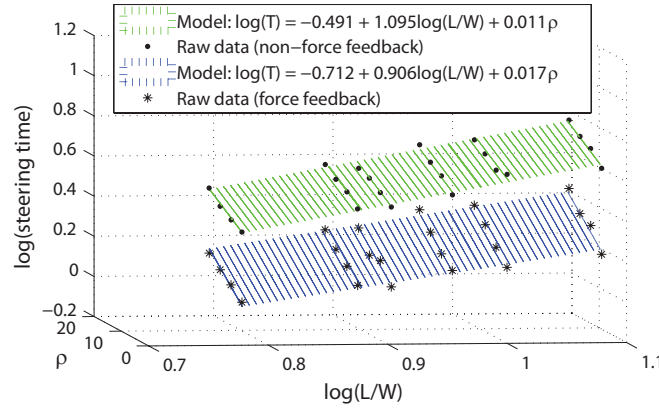


Figure 4.38: Equation 4.8 fits onto non-haptic (top green surface) and haptic feedback (bottom blue surface) data.

The model is strongly correlated to the data of both blocks. The goodness of fit of the two surfaces can be described by $R^2 = 0.9812$ and 0.9739 . Table 4.19 and 4.20 show the corresponding parameter estimates of the two surface. Conclusive evidence indicates ($p <$

	Estimate	t value	P> t	[95% Conf. Interval]
<i>a</i>	-0.491	-14.29	< 0.001	[-0.562, -0.419]
<i>b</i>	1.095	30.11	< 0.001	[1.019, 1.170]
<i>c</i>	0.011	13.73	< 0.001	[0.009, 0.012]

Table 4.19: Block 1: parameter estimates of Equation 4.8 fitting onto non-force-feedback data.

	Estimate	t value	P> t	[95% Conf. Interval]
<i>a</i>	-0.712	-17.64	< 0.001	[-0.796, -0.628]
<i>b</i>	0.906	21.22	< 0.001	[0.817, 0.995]
<i>c</i>	0.017	18.24	< 0.001	[0.015, 0.018]

Table 4.20: Block 2: parameter estimates of Equation 4.8 fitting onto force feedback data.

0.05) that the influence of the term $\log(L/W)$ and ρ is significant for modeling the steering time, indicating that the variation to the steering law can be applied to both the non-haptic and haptic steering tasks. It is also interesting to find out that with force feedback turned on, the coefficients *b* and *c* in Table 4.20 differ significantly from those in Table 4.19 (confidence intervals do not overlap), indicating that the coefficients *b* and *c* are dependent on the force feedback. There must be the effect for the interaction terms of $\log(L/W) \times F$ and $\rho \times F$.

Experiment 2: Modeling Haptic Steering with Force Magnitude, Path Length and width

The goal of Experiment 2 is to quantitatively describe the movement time, if possible, as a function of L , W and F . As the effect of L and W has been empirically verified in the steering law, we consider L and W as one independent variable ID ($ID = L/W$). The sub-goal is

therefore to quest for a relationship that satisfies $T = f(ID, F)$. This section specifies how such a relationship can be statistically derived. For each subject, the time stems from the same ID and F is averaged. ID , F and subject are all treated as within-subject factors in the following analysis.

- *Modeling the Effect of Path Length and Width*

Following the modeling methodology described in Section 1.3, the first step aims to find out if there is a linear relationship between $\log T$ and $\log ID$, *i.e.*,

$$\log T = a + b \log ID \quad (4.14)$$

Table 4.21 and 4.22 show the statistical evidence that under each force magnitude condition, such a linear relationship can be derived when $F < 0$ (assistance) and $F \geq 0$ (resistance), respectively. There is no statistical evidence for lack of fit ($p > 0.05$ in all cases), *i.e.*, for each force magnitude, L/W is an adequate predictor of the steering time. Figure 4.39 and 4.40 illustrate the related linear regression for each force magnitude. Note that the trend that the steering time increases as the ID grows can be statistically modeled by a linear function that passes through the confidence intervals.

- *Modeling the Effect of Force Magnitude*

The second step aims to examine the dependency of the coefficients a and b in Table 4.21 and 4.22 on the different force magnitudes F . We initially assume that a and b are linearly correlated to the force magnitudes, *i.e.*,

$$a(F) = a_0 + a_1 F \quad (4.15)$$

$$b(F) = b_0 + b_1 F \quad (4.16)$$

For the case where $F < 0$, the regression parameters estimates for Equation 4.15 are shown in Table 4.23 ($R^2 = 0.8246$, $F(1, 2) = 9.405$, $p = 0.0919$). No statistical evidence can be found for a_1 . As illustrated in Figure 4.41, the solid line which represents a constant crosses all of the 95% confidence intervals, *i.e.*, there is no conclusive evidence in the data that a (linearly) depends on F .

The linear correlation between the coefficient b and F can be evidenced by the regression parameter estimates for Equation 4.16 in Table 4.24 ($R^2 = 0.9273$, $F(1, 2) = 25.53$, $p = 0.0370$). As shown, there is a significant effect for b_1 . Figure 4.42 also illustrates that b is linearly dependent of F .

Therefore, the steering time in the case that $F < 0$ can be modeled as a function of $\log ID$ and F , which is of the following form:

$$\log T = a_0 + (b_0 + b_1 F) \log ID \quad (4.17)$$

Equation 4.17 can be further rewritten as:

$$\log T = a + b \log ID + c F \log ID \quad (4.18)$$

A similar analysis, *i.e.*, the dependency of coefficients a and b on F , has been performed in the case that $F \geq 0$. Table 4.25 shows the regression parameter estimates for Equation 4.15 ($R^2 = 0.9212$, $F(1, 4) = 46.79$, $p = 0.0024$). As can be seen, a_0 and a_1 have significant effects

F	coef. a		coef. b		model		lack of fit	
	Est.	C.I.	Est.	C.I.	F	p	F	p
-0.6	-.606	[-.96, -.25]	.862	[.48, 1.24]	11.589	< .01	.284	.89
-0.4	-.597	[-.88, -.32]	.901	[.60, 1.20]	12.815	< .01	.172	.95
-0.2	-.637	[-.82, -.46]	1.002	[.81, 1.19]	14.773	< .01	.067	.99
0.0	-.667	[-.27, -.61]	1.170	[1.11, 1.23]	20.058	< .01	.006	1.00

Table 4.21: Step 1: the statistical evidence for the linear regression between $\log T$ and $\log ID$ for each force magnitude ($F < 0$).

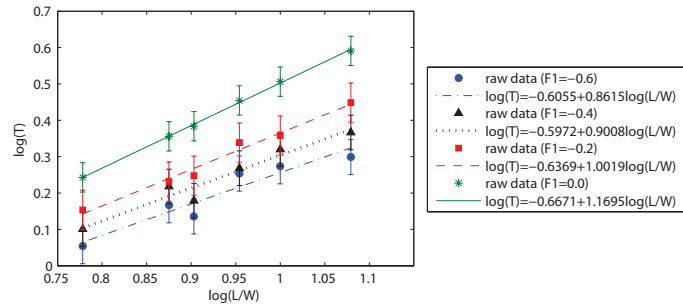


Figure 4.39: Step 1: the illustration of the linear regression between $\log T$ and $\log ID$ for each force magnitude ($F < 0$).

F	coef. a		coef. b		model		lack of fit	
	Est.	C.I.	Est.	C.I.	F	p	F	p
0.0	-.667	[-.72, -.61]	1.170	[1.11, 1.23]	20.058	< .01	.006	1.00
0.2	-.598	[-.89, -.31]	1.055	[.75, 1.36]	14.401	< .01	.141	.96
0.4	-.504	[-.75, -.26]	.968	[.71, 1.23]	13.981	< .01	.124	.97
0.6	-.430	[-.64, -.22]	.919	[.70, 1.14]	12.187	< .01	.088	.99
0.8	-.412	[-.63, -.20]	.964	[.74, 1.19]	14.316	< .01	.098	.98
1.0	-.398	[-.77, -.03]	1.011	[.61, 1.41]	12.613	< .01	.241	.91

Table 4.22: Step 1: the statistical evidence for the linear regression between $\log T$ and $\log ID$ for each force magnitude ($F \geq 0$).

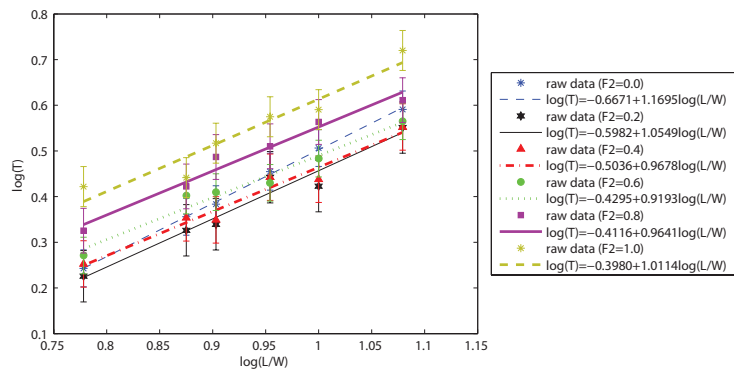
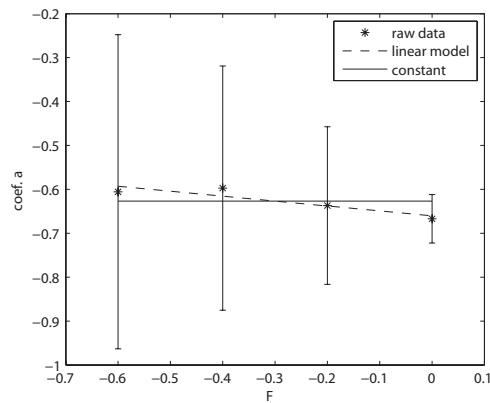
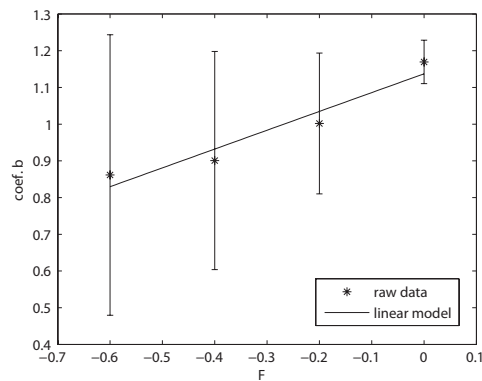


Figure 4.40: Step 1: the illustration of the linear regression between $\log T$ and $\log ID$ for each force magnitude ($F \geq 0$).

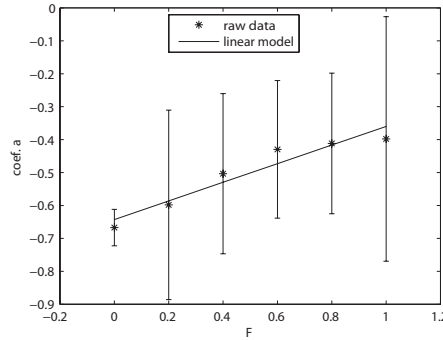
	Estimate	t value	P > t	[95% Conf. Interval]
a_0	-0.660	-48.218	< 0.001	[-0.719, -0.601]
a_1	-0.112	-3.067	0.092	[-0.270, 0.045]

Table 4.23: Step 2: regression parameter estimates for Equation 4.15 ($F < 0$).Figure 4.41: Step 2: the relationship between coefficient a and F ($F < 0$, linear model rejected).

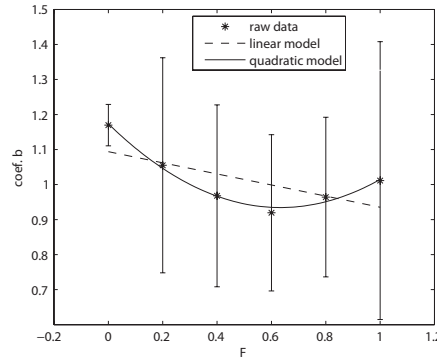
	Estimate	t value	P > t	[95% Conf. Interval]
b_0	1.137	29.960	0.001	[0.974, 1.301]
b_1	0.513	5.053	0.037	[0.076, 0.949]

Table 4.24: Step 2: regression parameter estimates for Equation 4.16 ($F < 0$).Figure 4.42: Step 2: the relationship between coefficient b and F ($F < 0$, linear model accepted).

	Estimate	t value	P> t	[95% Conf. Interval]
a_0	-0.643	-25.68	< 0.001	[-0.712, -0.573]
a_1	0.283	6.840	0.002	[0.168, 0.398]

Table 4.25: Step 2: regression parameter estimates for Equation 4.15 ($F \geq 0$).Figure 4.43: Step 2: the relationship between coefficient a and F ($F \geq 0$, linear model accepted).

	Estimate	t value	P> t	[95% Conf. Interval]
b_0	1.094	-20.471	< 0.001	[0.945, 1.242]
b_1	-0.159	-1.799	0.146	[-0.404, 0.086]

Table 4.26: Step 2: regression parameter estimates for Equation 4.16 ($F \geq 0$).Figure 4.44: Step 2: the relationship between coefficient b and F ($F \geq 0$, quadratic model accepted).

	Estimate	t value	P> t	[95% Conf. Interval]
b_0	1.174	98.230	< 0.001	[1.135, 1.211]
b_1	-0.756	-13.450	0.001	[-0.935, -0.577]
b_2	0.597	11.070	0.002	[0.425, 0.769]

Table 4.27: Step 2: regression parameter estimates for Equation 4.19 ($F \geq 0$).

in modeling the coefficient a , which is the statistical evidence in the data that a linearly increases as F grows. Figure 4.43 also illustrates this linear relationship, since the linear fit passes through all the 95% confidence intervals which can not be crossed by a constant.

The regression parameter estimates for Equation 4.16 is demonstrated in Table 4.26 ($R^2 = 0.4473, F(1, 4) = 3.237, p = 0.1464$). The p-value for the whole regression suggests that a linear regression is not an appropriate description of the data. There is no evidence for b_2 as shown in Table 4.26, either. A lack-of-fit test also rejects the linear model as opposed to a nonlinear one ($p = 0.0218$). A plot of the relationship between F and b (Figure 4.44) illustrates that neither a constant nor a linear model can properly represent the data. A polynomial regression, however, might be a better option. We first consider a quadratic relationship as below:

$$b(F) = b_0 + b_1F + b_2F^2 \quad (4.19)$$

The regression parameter estimates are shown in Table 4.27 ($R^2 = 0.9868, F(2, 3) = 112, p = 0.0015$), where all terms in Equation 4.19 have a significant effect. As shown in Figure 4.44, the quadratic model represented by the solid curve crosses all confidence intervals. A lack-of-fit test also indicates that the quadratic model can not be rejected as compared to the cubic model ($p = 0.8723$). Therefore, conclusive evidence shows that the coefficient b depends quadratically on F .

As a consequence, the steering time in the case that $F \geq 0$ can be modeled with the following form:

$$\log T = a_0 + a_1F + (b_0 + b_1F + b_2F^2) \log ID \quad (4.20)$$

which can be rewritten as:

$$\log T = a + b \log ID + cF + dF \log ID + eF^2 \log ID \quad (4.21)$$

In summary, type 2 haptic steering tasks (assistance ($F < 0$) and resistance ($F \geq 0$)) can be in general modeled using the following piecewise function:

$$\log T = \begin{cases} a_1 + b_1 \log ID + c_1F \log ID & (F < 0) \\ a_2 + b_2 \log ID + c_2F + d_2F \log ID + e_2F^2 \log ID & (F \geq 0) \end{cases} \quad (4.22)$$

Given $\log ID$, Equation 4.22 can be simplified as:

$$\log T = \begin{cases} a_1 + b_1F & (F < 0) \\ a_2 + b_2F + c_2F^2 & (F \geq 0) \end{cases} \quad (4.23)$$

Figure 4.45 illustrates how Equation 4.23 can be plotted given that $\log ID = 1$.

4.7 Discussion

The discussion is devoted to two issues. First, we discuss the issue of modeling path steering for 3D manipulation tasks. Then, we discuss the steering movement for the ball-and-tunnel task.

4.7.1 Steering Models

Can the Steering Law Be Used to Model the Ball-and-Tunnel Task?

The traditional way of verifying Accot and Zhai's steering law is to quest for a linear relationship between the time T and the index of difficulty ($ID = L/W$) using the empirical data.

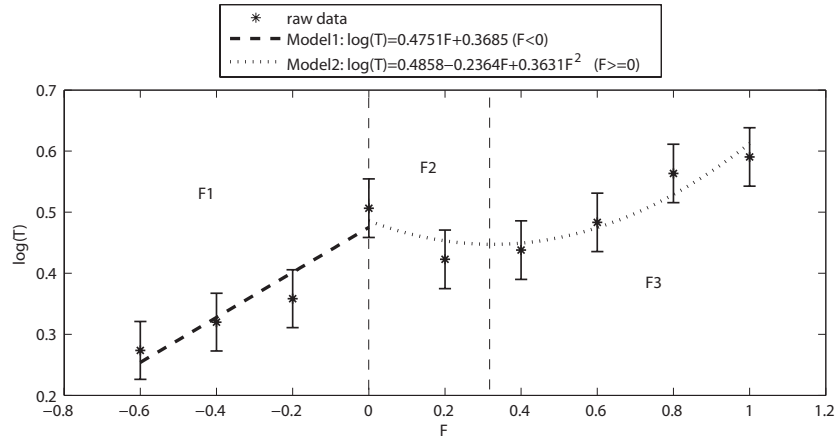


Figure 4.45: The relationship between $\log T$ and F , given that $\log ID = 1$. a_1, b_1, a_2, b_2 and c_2 have been empirically determined as shown in the legend. $\log T$ reaches its minimum when $F = 0.3255$ ($F \geq 0$).

In this chapter, we proposed a different way, where the verification of the linear regression is transformed to validate if the power b in Equation 2.8 is significantly different from one (see Appendix 2.5 for explanations). Table 4.28 summarizes the values of b derived from various conditions in this chapter. *Sub-trial* and *complete-trial* stem from the steering tasks,

	b value	[95% Conf. Interval]
sub-trial	0.592	[0.572, 0.612]
complete-trial	0.969	[0.913, 1.026]
no-correction	0.885	[0.784, 0.986]
haptic-1	0.906	[0.817, 0.995]
haptic-2	0.983	[0.5464, 1.420]
haptic-3	1.015	[0.786, 1.243]
steering-law	1	/

Table 4.28: Coefficient b 's (see Equation 4.3 and 2.8) values derived from various experiments. *Sub-trial*: steering through a segment of a path with varying curvatures (see Table 4.4); *complete-trial*: steering through a complete path with varying curvatures (see Table 4.9); *no-correction*: steering through a path without correction (see Table 4.16); *haptic-1*: haptic steering with type 1 force feedback (see Table 4.20); *haptic-2*: haptic steering with type 2 force feedback (assistance, see Table 4.21); *haptic-3*: haptic steering with type 2 force feedback (resistance, see Table 4.22); *steering-law*: Accot and Zhai's path steering task [AZ97].

in which users are allowed to make correction once they go beyond the boundary of the path. *Sub-trial* examines if the time of steering through each segment of a path (isolated by correction phases) can be modeled with the steering law, while *complete-trial* considers if the total steering time ($\sum \text{subtrials}$) can be modeled with the steering law. *No-correction* uses a different measure, where the ball-and-tunnel task does not allow for correction. Once the

input stylus goes out of the boundary, the current trial fails and users have to start over.

As shown in Table 4.28, the steering law can be applied to *complete-trial*, as its 95% confidence interval includes 1. However, the steering law is statistically evidenced not to be an appropriate description for the empirical data in *sub-trial* and *no-correction*.

Haptic-1, *haptic-2* and *haptic-3* represent the haptic steering with type 1 (force guidance) and type 2 (resistance and assistance) force feedback (see Section 4.6.1), respectively. As shown in Table 4.28, the steering law can be used to model the case with a more realistic force feedback (*haptic-2* and *haptic-3*), where haptic pushing and pulling feedback are rendered. However, it cannot model the case where deviation from the tunnel center is pulled back during the steering tasks (*haptic-1*).

Model Comparison with AICc

We have shown that in addition to path length and width, there are other independent variables that can significantly influence the ball-and-tunnel task. In particular, path curvature, orientation and force magnitude are all significant in modeling the steering time. The specific models are summarized below:

$$\log T = a + b \log \frac{L}{W} + c\rho \quad (4.24)$$

$$\log T = a + b \cos(\alpha) + c \cos(\beta) + d \sin(\beta) + e \cos(2\beta) + f \sin(2\beta) \quad (4.25)$$

$$\log T = \begin{cases} a_1 + b_1 \log ID + c_1 F \log ID & (F < 0) \\ a_2 + b_2 \log ID + c_2 F + d_2 F \log ID + e_2 F^2 \log ID & (F \geq 0) \end{cases} \quad (4.26)$$

In order to better compare the models proposed above and the steering law, we adopt the Akaike Information Criterion (AIC) [Aka74] which was developed to find the model, among a candidate set of models, that best explains the data with the fewest free variables. The AIC tends to penalize the models with large numbers of variables. It offers a relative measure of the information lost when a given model is used to describe the empirical data and hence, a smaller value indicates a better performance. The specific formula adopted is Akaike's second-order corrected Information Criterion (AICc), which is of the following form:

$$AICc = n \ln\left(\frac{RSS}{n}\right) + 2K + \frac{2K(K+1)}{n-K-1} \quad (4.27)$$

where K is the number of parameters in the model (including the constant term and the error term), n is the number of observations and RSS is the residual sum of squares. AICc is AIC with a greater penalty for extra variables. AICc is proved to be a better criteria than AIC, particularly when n is small or k is large [BA04].

Each of the models proposed in this chapter is compared to the steering law using AICc, except Equation 4.25. The reason is that in the experiment where Equation 4.25 was derived, the effect of L/W was not studied due to the impossibility of involving too many variables in one experiment. Therefore, the steering law cannot be applied to such tasks. Table 4.29 shows the differences of the AICc values in different conditions. As shown, in each condition, Equation 4.24 and 4.26 provide smaller AICc values than the steering law. This is the evidence that Equation 4.24 and 4.26 lose less information when dealing with ball-and-tunnel task of varying path curvature and force magnitude, and they outperform the steering law. The failure of taking path curvature into account significantly reduces the steering law's capacity when applied to different steering conditions.

	C1	C2	C3	C4	C5
ΔAICc	113.6	21.1	55.3	23.4	40.7

Table 4.29: AICc comparison between the steering law and our models (the AICc values of the steering law are always larger than those of our models). C1: the steering law vs. Equation 4.24 using the empirical data in Section 4.3.2; C2: the steering law vs. Equation 4.24 in Section 4.6.3 (Experiment 1: no force feedback); C3: the steering law vs. Equation 4.24 in Section 4.6.3 (Experiment 1: type 1 force feedback); C4: the steering law vs. Equation 4.26 in Section 4.6.3 (Experiment 2, $F \geq 0$); C5: the steering law vs. Equation 4.26 in Section 4.6.3 (Experiment 2, $F < 0$).

Model Usage

In the experiment described in Section 4.3, subjects were required to use a Polhemus FASTRAK connected by one 6-DOF input stylus to carry out the path steering tasks. In section 4.6, we asked the subjects to perform the same tasks with a Novint Falcon (both non-haptic and haptic steering). The models derived from the two separate experiments are illustrated in Figure 4.46. As shown, the three models do not intersect in the observed intervals.

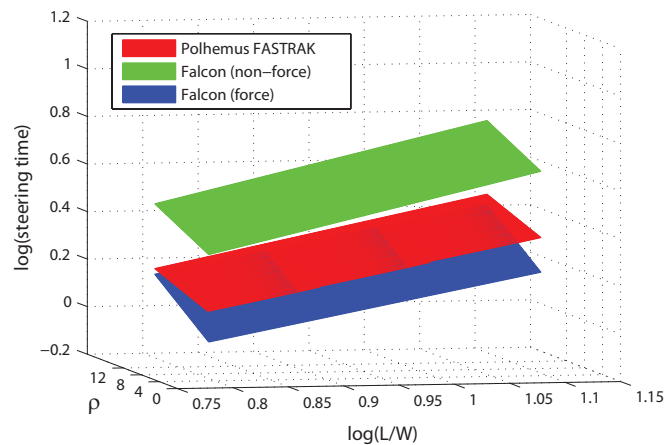


Figure 4.46: Models (Equation 4.8) derived from steering tasks using different input devices.

The time to steer through a tunnel using the FASTRAK stylus is in between that using the Falcon, with force feedback turned on and off, respectively. This is consistent with what was reported by the subjects, since most of the subjects participating in both experiments (10 subjects) claimed that the Falcon was more difficult to control in the absence of force feedback than the FASTRAK stylus and with force guidance the difficulty has decreased. It is evident that the model we proposed in Equation 4.8 can be used to compare the performance of input devices.

As shown in Figure 4.45, to analyze the effect of force magnitude, we divide F into three intervals (F_1 , F_2 and F_3) based on the monotonicity of the function. F_1 is the area where $F < 0$ and a certain amount of assistance is applied to the device. The minus sign only implies that the force exerted to the device is opposite to the direction it should have been exerted in

a ball-pushing task. The subjects are pulled toward the direction in which they operate the input device. The type of force actually facilitates the subjects by moving in the direction they were expecting and the absolute value of F indicates the amount of assistance. It is reasonable to find out that the steering time decreases as the amount of assistance increases.

$F2$ and $F3$ are the intervals when $F \geq 0$. This is a more realistic case when a resistance haptic feedback is provided as the subjects push a target ball through a tunnel. It is interesting to witness that the steering time falls with the growth of the resistance in $F2$. In a real world task, the fact may contradict our common sense, since the resistance usually leads to a longer steering time. In virtual reality, however, this is the supportive argument for utilizing haptic feedback in human-computer interaction. For our experiment, the steering time reaches its valley at point $F = 0.3255N$, after which it goes up again. It illustrates that without force feedback, the steering tasks in virtual reality can be time-consuming; with an appropriate amount of force guidance, the efficiency for the steering tasks can be improved; whereas with too much force feedback, the efficiency may be deteriorated. The model we proposed as in Equation 4.22 can be used to quantitatively determine the value of F , which can lead to the shortest steering time and serves as a guideline for choosing the appropriate force magnitude in the haptic steering tasks.

Limitations and constraints

Obviously, no model can be claimed to be complete and robust in any conditions. There does not exist a universal model for all kinds of human motion, either. The interaction models proposed in this chapter focus on the study of different spatial aspects that can influence users' temporal performance on 3D path steering tasks in virtual environments. The models are basically descriptive in that the formulation and verification rely on a statistical analysis of the experimental data. It is important to note, despite all of the models have been evidenced to be good descriptions of the experimental data, they do not incorporate the relationship between user actions and cognitive process and thus cannot be used to understand how users actually perform the steering tasks. Instead, they aim to address the question that given the spatial characteristics of a steering task in a virtual environment, how fast one may respond to the task.

As performing steering tasks in current imperfect virtual environments might be substantially different from that in the real world, the interaction models might only describe users' performance in the virtual environment. For a real world task, users performance might be influenced in a different manner or require different spatial characteristics.

The models are developed to describe users' steering performance on a regular shaped 3D tunnel, *i.e.*, the width, height and depth within the tunnel could all be expressed by one factor W . For an irregular shaped path, the effect of height and depth might be integrated into the proposed models.

The ball-and-tunnel task is designed such that the orientation of the input device does not play a role, which only makes use of 3 DOFs of the 6-DOF stylus. For a more complex 3D manipulation task (*e.g.*, the ring-and-wire task as described in Figure 4.1), the models might need to take the other 3 DOFs (roll, yaw and pitch) into account.

Model 4.24 can be used to predict users' steering time on path of constant width and curvature. However, it is beyond its scope to model a more general case where width and curvature vary along the path. A more sophisticated model can be expected for such tasks. Accot and Zhai' steering law is an attempt to address such a problem. However, the general law as shown in Equation 2.2 has barely been used in practice. On the contrary, the simplified version as described by Equation 2.3 has a widespread application.

Model 4.25 can be used to decide how quickly users may steer through the path placed in different orientations. One of the major issues is that the determination of the angles α and β might be difficult for irregular shaped path. In fact, to fully describe the orientation of a 3D path, a third angle γ which determines the rotation around the line from one end of the path to the other needs to be included. Together with α and β , γ makes sure that the path can be placed in any orientation in 3D space.

Model 4.26 describes two types of common force feedback. One is exerted perpendicular to the movement, while one mostly parallel to the movement. Though of less importance, there are many more force feedback that cannot be expressed by this model. Moreover, the model is used to describe a “constant” force feedback, *i.e.*, the force magnitude is (approximately) fixed during the task. Apparently, a dynamic force feedback in one trial cannot be precisely modeled without any revision.

4.7.2 Steering Movements

Should the Ball-and-Tunnel Task Be Modeled as An Infinite Number of Goal-Crossing Movements?

Figure 4.47 illustrates the power laws (Equation 2.8) derived from different steering conditions. Each curve represents a power law with a different power b that can be found in Table 4.28. As shown, Fitts’ law can also be converted into a similar form $T = \log_2(ID)^9$, which is very close to $T = ID^{1/3}$ that lies at the bottom among the curves. Accot and Zhai’s steering law which has the largest power ($b = 1$) is at the top. All other steering conditions studied in this chapter, *i.e.*, either using a 6-DOF stylus or a 3-DOF haptic device in the ball-and-tunnel task, can be modeled in between the steering law and Fitts’ law. It is important to note that the steering law and Fitts’ law describe two different types of movements. As shown in Chapter 3, the movement that can be described by Fitts’ law is composed of a ballistic movement and a correction movement, each of which is made up of several sub-movements. On the contrary, the movement that can be modeled by the steering law is postulated to consist of an infinite number of goal-crossing movements, which constitute a smooth and continuous movement. However, it is believed that under some circumstances (*e.g.*, when navigating through an extremely short or wide path [AZ97]), path steering tends to become a ballistic movement which follows Fitts’ law. According to Figure 4.47, it might imply that users’ steering movement for the ball-and-tunnel task should not be modeled as smooth and continuous goal-crossing movements. Instead, it might be better modeled as a series of small and jerky sub-movements (either ballistic or corrective), each of which covers a segment of the path.

It would seem that there are several intermediate destinations involved in between the starting point and the final destination on the path, which break the entire path steering into several sub-steering movements. An intermediate destination serves as a goal for one sub-movement, which is also a starting point for the next sub-movement. In this way, users temporarily transform a tiny segment of path steering into pointing-like movement such that a ballistic/correction movement is performed in between two adjacent intermediate destinations. The fluctuation that can be found in the velocity profiles in Figure 4.29 could be the evidence for such an argument. The peaks and the valleys appearing in the profiles are quite similar to those found in the pointing velocity profiles (*e.g.*, Figure 3.3), which might be used

⁹This is a simplified version of Fitts’ law which aims to show that the relationship between T and $\log_2(L/W)$ is linear. Note: ID is defined as L/W for the purpose of comparing to other power laws; the constant term is not taken into account.

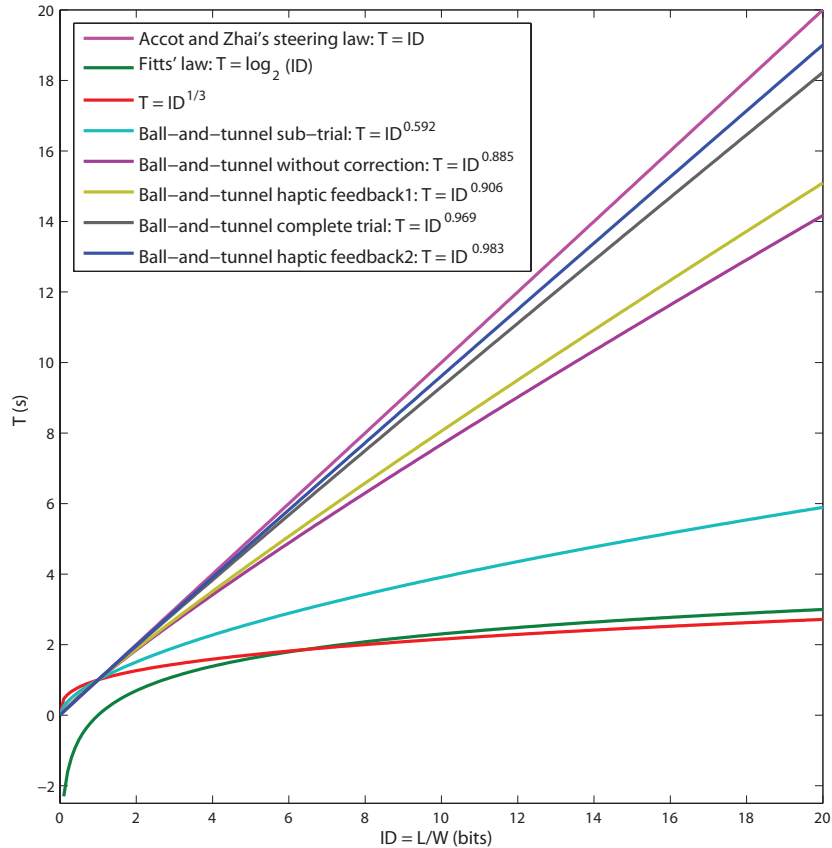


Figure 4.47: Power laws derived from various steering conditions.

to decompose a steering movement into sub-movements. Each sub-movement steers through a segment of the path, whose length might depend strongly upon the difficulty of the steering tasks. This can be proved by a comparison between Figure D.1 and D.2 in Appendix D. As steering through a linear path usually has a smaller difficulty than through a circular path, the number of sub-movements involved in Figure D.1 is on average smaller than that in Figure D.2, *i.e.*, the sub-movements on a linear path cross longer segments. This argument is demonstrated in Figure 4.48 and 4.49. In an ideal condition as shown in Figure 4.50, a continuous and smooth path steering with uniform velocity might be observed. This is exactly what the governing idea of the steering law claims and would require a path steering with extremely low index of difficulty.

The difference between the steering movements observed in our experiments and Accot and Zhai's experiments [AZ97, ZAW04] could be attributed to several reasons. First of all, it might be due to the difference between the steering tasks. The idea that the steering movement is composed of an infinite number of goal-crossing tasks was derived from 2D steering

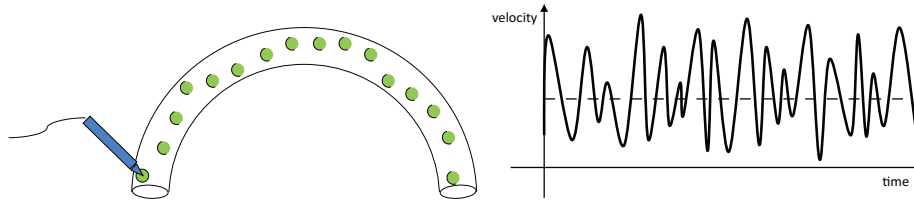


Figure 4.48: A virtual-world circular path steering movement. Left: sub-movements and intermediate destinations; right: the movement velocity profile. A sub-movement crosses the segment between any two adjacent intermediate destinations and corresponds to a concave up parabola observed in the velocity profile on the right.

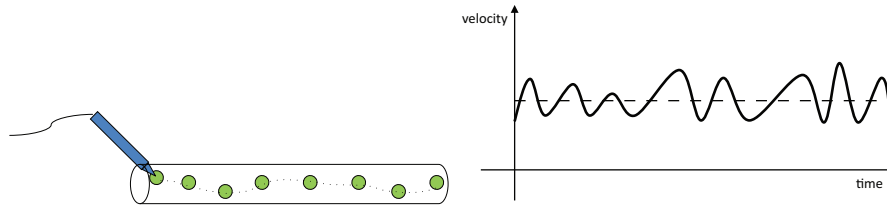


Figure 4.49: A virtual world linear path steering movement.

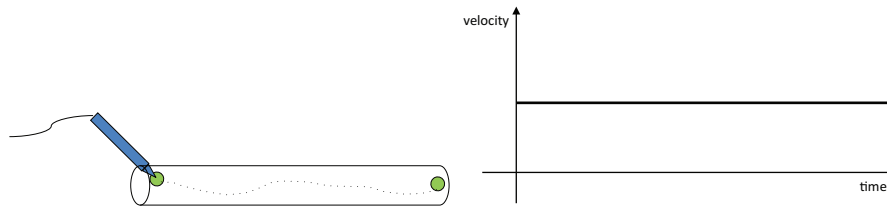


Figure 4.50: An ideal path steering movement assumed by the steering law.

tasks that do not provide users the opportunity for making corrections. Accot and Zhai have also validated the steering law in 3D locomotion tasks, utilizing this idea. However, 3D manipulation tasks as represented by the ball-and-tunnel task might be substantially different from these tasks. In particular, users have to manually control the input device in the 3D space, which increases the opportunity for introducing kinematic errors. This might stop the users from performing smooth movements.

The intrinsic difficulty of 3D interaction in the virtual environment might also be a reason. As discussed in Chapter 3, depth perception, visual quality and lack of multimodal cues in the virtual environment could all play a role. An interesting finding from Figure 4.47 is that as introducing haptic feedback to the path steering tasks, the power law is getting closer to the steering law, implying that the small and jerky sub-movements tend to be smaller or even disappear with the assistance of force feedback, particularly when the force feedback becomes more and more realistic. This might indicate that the steering law could be an appropriate description of the steering movement performed in the real world.

The reason could also be attributed to the technical issues on the experimental setup. For example, the VR system used for the experiment introduced latency, which to some extent leads to a lag between the visual and motor feedback. If the latency is large enough to be observed, users might have to deliberately slow down before the visual feedback is updated. However, it is reasonable to believe that the 80ms around latency measured in our VR system

should not cause such problems. In addition, the non-co-located setup could be a factor as well. Separating the motor space from the visual space might result in different steering behavior than a co-located setup. It is also important to note that the C-D ratio was set to 1/5 during the haptic steering tasks, which makes the users move faster in the virtual environment. A possible consequence could be that it increases the difficulty in controlling path steering accuracy.

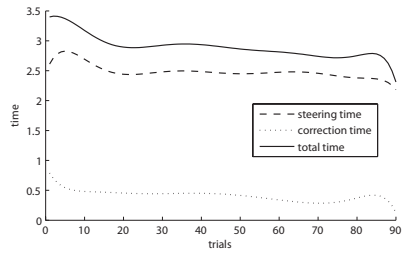
4.7.3 Learning Effect Analysis

All of the experiments involved in this chapter are designed with repeated measures. To avoid a learning effect on the different levels of a factor under observation, trials in an experiment are presented in a random order which differs from one subject to another. Subjects' movement time is averaged based on the trial numbers. Figure 4.51 demonstrates the variation of the movement time (the steering time, the correction time and the total time) as trial increases for all the experiments carried out in this chapter (the effect of *ID* is minimized through averaging the time). As shown, in most of the experiments, users demonstrate a small learning effect where the movement time tends to decrease as more trials are practiced. However, statistical evidence shows that the learning effect can only be statistically observed in Figure 4.51(d) and 4.51(e), which utilized the 3-DOF haptic device Novint Falcon as the input with type 1 force feedback (pulled toward the center of the tunnel) turned on and off, respectively. A learning effect can both be found in the correction time ($F_{off}(89, 1246) = 1.496, p = 0.0026; F_{on}(89, 1246) = 1.349, p = 0.0198$) and the total time ($F_{off}(89, 1246) = 1.562, p = 0.0009; F_{on}(89, 1246) = 1.360, p = 0.0171$). It seems that users are more familiar to the 6-DOF stylus and the room for improvement as trial increases is small. However, users demonstrate a relatively steep learning curve when using the haptic device. In particular, the time spent on the correction phase has significantly decreased as they practice. This might be either due to the fact that steering with Novint Falcon is difficult, or that users require some practice before they can make better use of the force rendered.

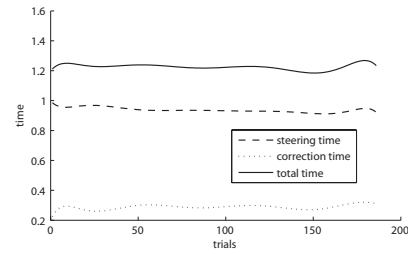
4.8 Conclusions

In this chapter, we have addressed the question “if the ball-and-tunnel tasks performed in the virtual environment could be modeled by Accot and Zhai's steering law”. As shown by the experimental results, the steering law can be applied in very limited cases. It is able to predict the overall steering time ($\sum subtrials$) for the ball-and-tunnel task if the task allows users to make a correction when going beyond the boundary. The time for path steering, in the presence of resistance and assistance feedback, can also be modeled by the steering law. However, statistical evidence shows that the steering law is not an appropriate description for the overall steering time when no correction phases are involved. It also fails to describe the steering time on a segment of the path (sub-trial). In the presence of a force guidance which is exerted to pull users back to the center of the path during the steering phase, the steering law was found to be invalid as well.

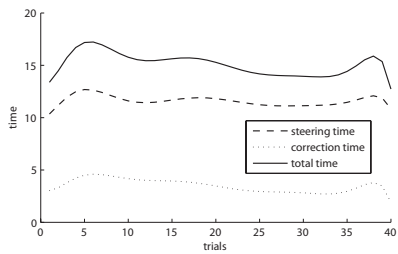
In parallel, we have proposed several new interaction models which can be used in various 3D manipulation tasks. In addition to path length and width, it has been statistically evidenced that path curvature and orientation can also significantly influence users' steering time. The effect of path curvature in 3D manipulation tasks can be accurately modeled as a percentage increase in time for increasing curvatures. According to AICc, the new law outperforms the steering law as it has a more accurate prediction power for 3D manipulation



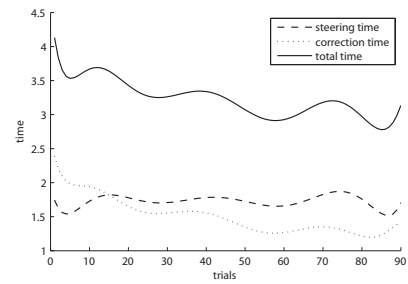
(a) Section 4.3, Experiment 1: varying curvature



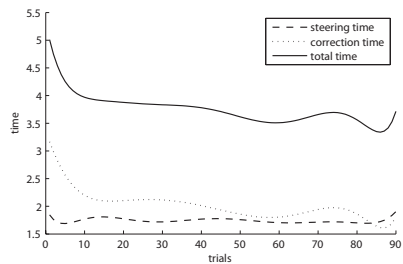
(b) Section 4.3, Experiment 2: varying orientation



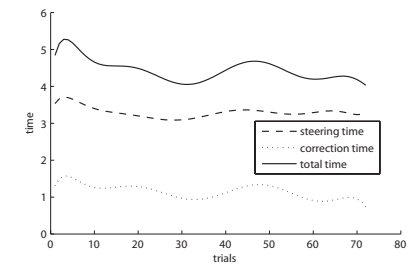
(c) Section 4.4: varying path property



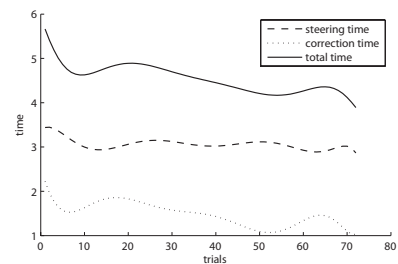
(d) Section 4.6, Experiment 1: without force feedback



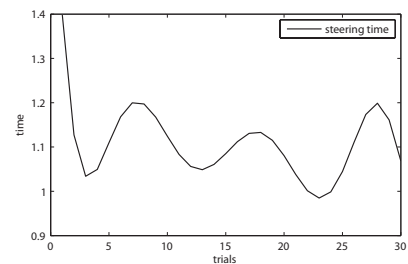
(e) Section 4.6, Experiment 1: type 1 force feedback



(f) Section 4.6, Experiment 2: assistance force feedback



(g) Section 4.6, Experiment 2: resistance force feedback



(h) Section 4.5: long path steering

Figure 4.51: Learning effect analysis in various steering tasks (cross *ID* and *subject* analysis).

tasks. Moreover, the empirical data indicate that it requires less time if the path steering direction is parallel to the viewing direction than perpendicular. The movement time can be statistically modeled as a periodic function of two independent orientation angles.

We have also focused on the effect of varying path properties on users' movement velocities. As the experimental results indicate, steering through the path of varying orientation or width results in a significant decrease in the average velocity. Within a task, the joint where a change in path curvature or orientation takes place also significantly reduces the velocity. Path width and curvature are highly correlated to the average velocity of steering through a segment, *i.e.*, the wider the segment (or the smaller the path curvature), the larger the average steering velocity on that segment.

Furthermore, the use of haptic feedback in the ball-and-tunnel task has also been studied. We experimentally implemented two types of force feedback. The first type is a force guidance, which aims to pull users back to the tunnel center when any deviation is detected. The second force is designed in a realistic way where users encounter the resistance as they push a ball through a tunnel and receive the assistance as being attracted by the ball. The experimental results showed that the steering time has improved significantly in the presence of the first type force feedback. The 60.23% improvement in efficiency is larger than the 52% improvement reported for similar 2D steering tasks. In addition, it is experimentally verified that the steering time can be statistically modeled using the second type force magnitude. The model provides an approach to determine an appropriate interval of force magnitude, within which the steering time can be significantly decreased.

A further study of the steering models proposed in this chapter suggests that users' steering movements for the ball-and-tunnel task utilizing a 6-DOF stylus or a 3-DOF haptic device should be described as a number of small and jerky sub-movements that are similar to those found in the pointing tasks, rather than a continuous and smooth movement as assumed by the steering law. As shown in Figure 4.47, the steering movement could be better modeled by the power laws that lie in between the steering law and Fitts' law. However, Figure 4.47 also demonstrates that as the difficulty of the tasks decreases (*e.g.*, input device: from 6-DOF to 3-DOF; multimodal cues: from non-haptic to haptic), the steering movement is getting closer to "an infinite number of goal-crossing movements".

Chapter 5

Object Pursuit

5.1 Introduction

Pursuit is widely referred to as the action of the eye in following a moving target [LMT87, BM83]. Given the parts of the human body that are used, pursuit can also be categorized into locomotion [CF07] and manipulation tasks [SBJ⁺97]. In this chapter, pursuit is employed to specifically describe a 3D manipulation task, which requires users to manually track a 3D moving target in a virtual environment.

To our knowledge, no research has been performed for modeling object pursuit and the study provided in this chapter is the first attempt. The goal is to identify the independent variables that can affect the object pursuit time and statistically formulate a model for such tasks. Specifically, this chapter aims to address the following questions:

- How can object pursuit tasks be modeled?
- What independent variables influence users' performance of object pursuit tasks?

These questions are approached through the following steps. First of all, we propose a way of measuring object pursuit tasks, which separates the phase when users lose the object (the correction phase) from the phase when users track the object within the boundary (the tracking phase). Then, several independent variables, including the object size, the velocity and the length of the path to be crossed by the object are statistically analyzed in each phase and an object pursuit model that quantitatively describes the movement time as a function of these variables are developed. We show that the model proposed can be used to determine an appropriate object velocity, with which the total movement time to track an object can be minimized. Finally, from a comparison between the models of pointing, steering and object pursuit, general rules for modeling interaction tasks are proposed.

5.2 Object Pursuit Task

As shown in Figure 5.1, a typical 3D object pursuit task is designed in a virtual environment. Users are required to hold an input stylus, represented as a 3D pen in the virtual environment, to trace a virtual 3D target (a 3D ball) that moves with a uniform velocity with their dominant hands. To enhance the visual feedback, 3D coordinate axes are attached to the tip of the

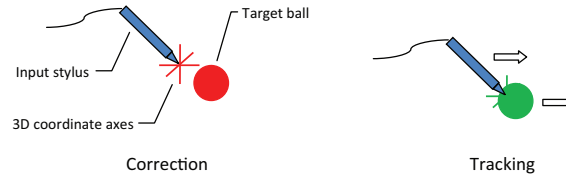


Figure 5.1: Two phases in an object pursuit task. Left: the correction phase; Right: the tracking phase.

pen. The space in which the 3D ball moves is encapsulated in a $0.72 \times 0.4 \times 0.4 m^3$ (*length* \times *height* \times *depth*) sized wire-frame box, whose floor is covered with a chessboard pattern.

At the beginning of the task, users are shown a stationary target ball and both the ball and the axes attached to the pen are colored red. The target ball starts to move with a uniform velocity, once the tip of the pen, *i.e.*, the origin of the 3D axes is inside the target ball (see Figure 5.1, right). Meanwhile, the ball and the axes turn to green. Users are asked to track the moving target by keeping the tip of the pen within the ball as well as they can. If they fail, the moving target stops, making the axes and the ball red again (see Figure 5.1, left). Users have to correct this movement by steering the pen back to the target ball and resuming the tracking where they left off. A task starts when the target ball leaves the initial position and proceeds until the target ball moves to a predefined destination that is not known to the subjects in advance.

The time when the tip of the pen is within the target ball and moves with it is defined as *the tracking time*, *i.e.*, the total time when the target ball remains green, while the time when the tip of the pen deviates from the target ball and the user makes a correction is defined as *the correction time*, *i.e.*, the total time when the target ball remains red. The time for completing an object pursuit task, *i.e.*, the sum of the tracking time and the correction time, is defined as *the total time*.

5.3 Experiment

A controlled experiment was carried out to study how users' movement time can be affected by the characteristics of the object pursuit tasks. The specific variables involved in the experiment are summarized below:

- independent variables: the object size, the object velocity, the length of the path to be crossed by the object and path curvature;
- dependent variables: the steering time, the correction time and the total time.

The same experimental apparatuses and setup as specified in Section 3.2.1 were employed.

5.3.1 Subjects

11 right-handed subjects voluntarily participated in the experiment. There were 2 females and 9 males, aged from 24 to 35. 6 of them were non-experienced VR users. All participants had normal or corrected to normal vision and none of them was stereo blind.

5.3.2 Procedure

We have conducted an experiment with a repeated-measures design, in which the target respectively moved on the linear (see Figure 5.2, left) and the circular paths (see Figure 5.2, right) that are placed on xy-plane. The target ball was designed to make uniform motions on



Figure 5.2: Two types of paths used in the experiment. Note that the paths were not shown to the subjects.

each type of path. The specific values selected for each independent variable are shown as follows:

- target size (W): 0.03 and 0.04m;
- path length (L): 0.24, 0.30 and 0.36m;
- target velocity (v): 0.10, 0.15, 0.20, 0.25 and 0.30m/s;
- path curvature (ρ): 0, $8m^{-1}$.

The length of the path to be crossed by the target was selected such that completing an object pursuit task only required the extension of the arm, keeping the body relatively still. The velocity of the moving target was tested to be suitable to track for both non-experienced and experienced subjects in a pilot study. Each combination of the above independent variables was repeated three times, resulting in $2 \times 3 \times 5 \times 2 \times 3$ (*target size* \times *path length* \times *motion velocity* \times *path curvature* \times *repeat*) trials for a subject and a total number of 1980 trials for 11 subjects. To compensate the practice effect, trials were presented in a random order that differed from one subject to another. Subjects were allowed to have a break whenever they suffered from fatigue between trials. This was, however, strictly prohibited during a trial.

5.4 Results

In order to model the object pursuit tasks, inspired by Fitts' law and the steering law, we chose to statistically explore the relationships between the temporal information of each phase and the characteristics of the tasks, *i.e.*, $T_{tracking} = f_1(L, W, v)$, $T_{correction} = f_2(L, W, v)$ and $T_{total} = T_{tracking} + T_{correction} = f_3(L, W, v)$. The tracking time, *i.e.*, the time when the target ball moves with a constant velocity, can be described as:

$$T_{tracking} = \frac{L_1}{v} + \frac{L_2}{v} + \dots + \frac{L_n}{v} = \frac{L}{v} \quad (5.1)$$

where L_n is the length of the n -th path segment crossed by the target ball without a correction. As shown, $T_{tracking}$ is constant if L and v are fixed.

Different from $T_{tracking}$, $T_{correction}$ and T_{total} cannot be precisely derived and thus require regression analysis. The collected data were logarithmically transformed to meet the requirements of the statistical methods. From a pilot study, it was confirmed that the correction time could be linearly related to the term L/W and v . The following data analysis aims at verifying the model of the form:

$$\log(T_{correction}) = a + b \log \frac{L}{W} + cv + dv \log \frac{L}{W} \quad (5.2)$$

where a, b, c and d are empirically determined constants.

For each subject, the correction time with the same L/W , v and ρ is averaged. We have performed a three-way ANOVA with repeated measures on factors L/W , v and ρ , each of which was treated as a within-subject variable. The related results are shown in Table 5.1, where statistical evidence indicates that all three factors have a significant effect on the correction time. There is also an effect of the interaction term $L/W \times v$, which is consistent with Equation 5.2.

source	SS	df	MS	F	p
L/W	39.873	5	7.975	38.311	< 0.0001
v	69.688	4	17.422	20.563	< 0.0001
ρ	10.770	1	10.770	11.294	0.0072
$L/W \times v$	7.261	20	0.363	3.254	< 0.0001
Between-Subjects	79.246	10			
Within-Subjects	249.336	649			
Error(L/W)	10.408	50	0.208		
Error(v)	33.889	40	0.847		
Error(ρ)	9.536	10	0.954		
Error($L/W \times v$)	22.312	200	0.112		
$L/W \times \rho$	0.070	5	0.014	0.078	0.9953
Error($L/W \times \rho$)	8.962	50	0.179		
$v \times \rho$	1.284	4	0.321	1.351	0.2680
Error($v \times \rho$)	9.499	40	0.237		
$L/W \times v \times \rho$	2.628	20	0.131	1.138	0.3132
Error($L/W \times v \times \rho$)	23.086	200	0.115		

Table 5.1: Comparison between the correction time: three-factor ANOVA with repeated measures on L/W , v and ρ .

Following the modeling methodology, we first examine if there is a linear relationship between $\log T$ and $\log(L/W)$, *i.e.*,

$$\log T = a + b \log \frac{L}{W} \quad (5.3)$$

Table 5.2 (the linear path) and 5.3 (the circular path) demonstrate that for each velocity, a linear regression between $\log T$ and $\log(L/W)$ can be statistically evidenced. There is no evidence for lack of fit ($p > 0.05$ in all cases), *i.e.*, given velocity, L/W is an adequate predictor of the correction time. This can also be confirmed by Figure 5.3 and 5.4, where the linear models pass through the associate confidence intervals.

The second step aims to verify if and how the coefficients a and b in Table 5.2 and 5.3 are influenced by the velocity v . We first assume that both a and b can be modeled as a linear function of v , *i.e.*,

$$a(v) = a_0 + a_1 v \quad (5.4)$$

$$b(v) = b_0 + b_1 v \quad (5.5)$$

For linear path pursuit tasks, the regression estimates of Equation 5.4 are shown in Table 5.4 ($R^2 = 0.8109, F(1, 3) = 12.86, p = 0.0371$). v is evidenced to have a statistical effect on a . As illustrated in Figure 5.5, the linear model crosses all of the confidence intervals. There

v	coef. a		coef. b		model		lack of fit	
	Est.	C.I.	Est.	C.I.	F	p	F	p
.10	-5.214	[-8.07, -2.25]	4.617	[1.47, 7.76]	8.590	< .01	.531	.71
.15	-2.672	[-4.08, -1.26]	2.596	[1.09, 4.10]	11.866	< .01	.179	.95
.20	-1.995	[-2.88, -1.11]	2.088	[1.14, 3.03]	19.517	< .01	.265	.90
.25	-1.770	[-2.48, -1.06]	2.025	[1.27, 2.78]	28.866	< .01	.219	.93
.30	-1.239	[-1.78, -0.70]	1.558	[0.98, 2.14]	29.069	< .01	.226	.92

Table 5.2: Step 1: the statistical evidence for the linear regression between $\log T$ and $\log ID$ for each velocity ($\rho = 0$).

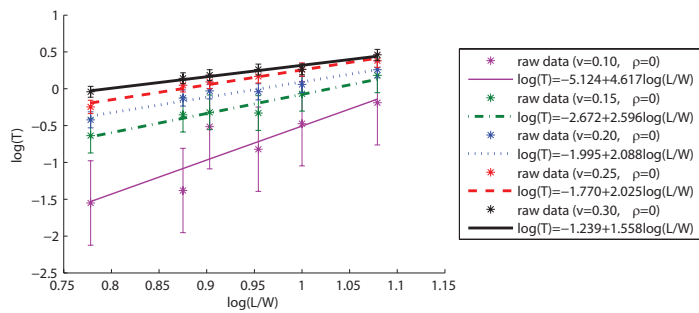


Figure 5.3: Step 1: the illustration of the linear regression between $\log T$ and $\log ID$ for each velocity ($\rho = 0$).

v	coef. a		coef. b		model		lack of fit	
	Est.	C.I.	Est.	C.I.	F	p	F	p
.10	-4.263	[-6.31, -2.21]	4.119	[1.93, 6.31]	14.142	< .01	.248	.91
.15	-2.554	[-3.59, -1.52]	2.779	[1.67, 3.88]	25.260	< .01	.643	.63
.20	-1.866	[-2.58, -1.16]	2.225	[1.47, 2.98]	34.283	< .01	.706	.59
.25	-1.315	[-1.76, -0.87]	1.757	[1.28, 2.23]	54.134	< .01	.313	.87
.30	-1.129	[-1.60, -0.66]	1.580	[1.08, 2.09]	39.099	< .01	.148	.96

Table 5.3: Step 1: the statistical evidence for the linear regression between $\log T$ and $\log ID$ for each velocity ($\rho = 8$).

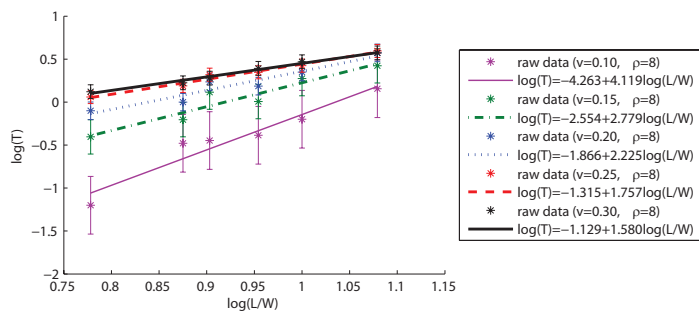
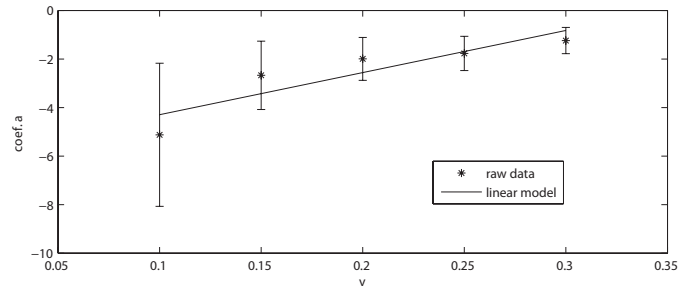
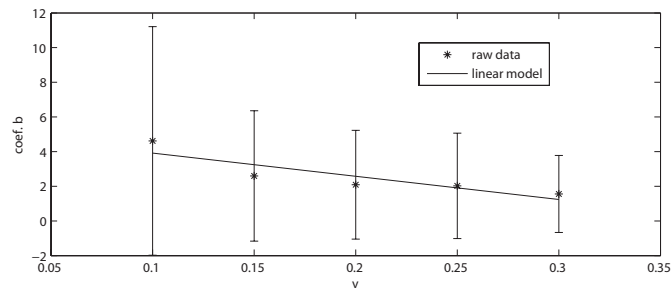


Figure 5.4: Step 1: the illustration of the linear regression between $\log T$ and $\log ID$ for each velocity ($\rho = 8$).

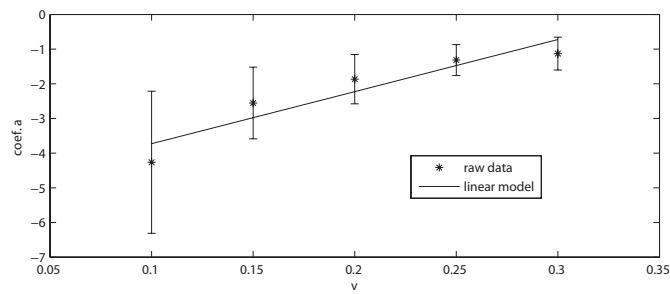
	Estimate	t value	P > t	[95% Conf. Interval]
a_0	-6.028	-5.877	0.010	[-9.293, -2.764]
a_1	17.341	3.586	0.037	[1.953, 32.730]

Table 5.4: Step 2: regression parameter estimates for Equation 5.4 ($\rho = 0$).Figure 5.5: Step 2: the relationship between coefficient a and v ($\rho = 0$).

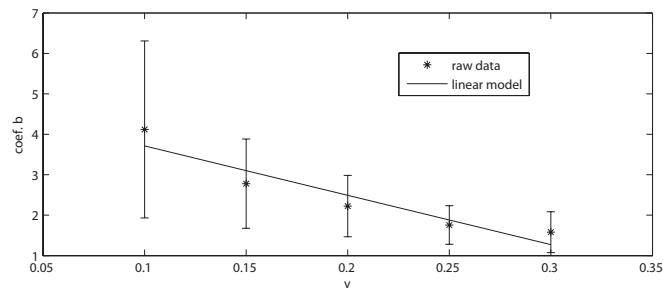
	Estimate	t value	P > t	[95% Conf. Interval]
b_0	-5.228	6.015	0.010	[2.473, 8.031]
b_1	-13.377	-3.250	0.047	[-26.48, -0.277]

Table 5.5: Step 2: regression parameter estimates for Equation 5.5 ($\rho = 0$).Figure 5.6: Step 2: the relationship between coefficient b and v ($\rho = 0$).

	Estimate	t value	P > t	[95% Conf. Interval]
a_0	-5.288	-7.621	0.005	[-7.411, -3.045]
a_1	15.010	4.643	0.019	[4.722, 25.310]

Table 5.6: Step 2: regression parameter estimates for Equation 5.4 ($\rho = 8$).Figure 5.7: Step 2: the relationship between coefficient a and v ($\rho = 8$).

	Estimate	t value	P > t	[95% Conf. Interval]
b_0	4.932	9.473	0.002	[3.275, 6.589]
b_1	-12.200	-4.971	0.016	[-20.010, -4.389]

Table 5.7: Step 2: regression parameter estimates for Equation 5.5 ($\rho = 8$).Figure 5.8: Step 2: the relationship between coefficient b and v ($\rho = 8$).

is conclusive evidence in the data that a can be modeled as a linear function of v . The statistical evidence in Table 5.5 and Figure 5.6 shows that b also linearly depends on v . The same results apply to the case when the object is pursued on circular paths (see Table 5.6, Figure 5.7, Table 5.7 and Figure 5.8.).

Hence, the correction time of the object pursuit tasks can be modeled with the following equation (both linear and circular paths):

$$\log T = a_0 + a_1 v + (b_0 + b_1 v) \log \frac{L}{W} \quad (5.6)$$

which is of exactly the same form as Equation 5.2.

5.5 Discussion

For the results presented above, several aspects need to be discussed. First of all, it is interesting to investigate if the total movement time ($T_{tracking} + T_{correction}$) of an object pursuit task can be minimized. Note that the tracking time of a pursuit task as shown in Equation 5.7

$$T_{tracking} = \frac{L}{v} \quad (5.7)$$

is constant only if L and v have been fixed. If v is treated as a variable, $T_{tracking}$ is a monotone decreasing function as v increases. On the contrary, the time for the correction phase as shown in Equation 5.8

$$\log(T_{correction}) = a + b \log \frac{L}{W} + (c + d \log \frac{L}{W})v \quad (5.8)$$

is a monotone increasing function as v increases. As a compromise, there must be a specific v , with which T_{total} of the following form

$$T_{total} = T_{tracking} + T_{correction} = \frac{L}{v} + 10^{a+b \log \frac{L}{W} + (c+d \log \frac{L}{W})v} \quad (5.9)$$

reaches its extremum. Figure 5.9 demonstrates the relationship expressed by Equation 5.9, given L, W, a, b and c . An optimal velocity can be derived by taking the derivative to the right side of Equation 5.9 in terms of v and making it equal to 0, *i.e.*,

$$\frac{d}{dv} \left(\frac{L}{v} + 10^{a+b \log \frac{L}{W} + (c+d \log \frac{L}{W})v} \right) = 0 \quad (5.10)$$

The solution of Equation 5.10 can be described as:

$$v = \frac{2 \text{ProductLog} \left[\sqrt{\frac{L(c \ln 10 + d \ln \frac{L}{W})}{2 \cdot 10^{\frac{a}{2}} (L/W)^{\frac{b}{2}}}} \right]}{c \ln 10 + d \ln(L/W)} \quad (5.11)$$

where $\text{ProductLog}(x)$ is the Lambert W function (also called Omega function or product logarithm) that gives the principal solution for w in $x = we^w$ (Complex-valued solutions for v have been discarded). It can be seen from Equation 5.11, if the target size W and the path length L are known, an optimal velocity to minimize the total pursuit time can be derived. This can be used as a guideline when designing user interfaces with moving targets. Assigning

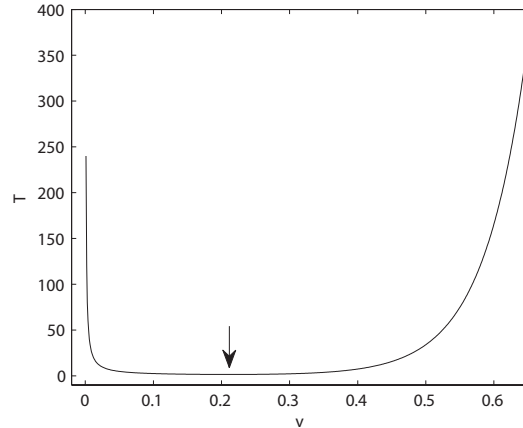


Figure 5.9: The relationship between v and T_{total} , given that $L = 0.24$, $W = 0.04$, $\rho = 0$, $a = -6.028$, $b = 5.252$, $c = 17.341$ and $d = -13.377$.

the experimentally derived constants a , b and c to Equation 5.11 ($L/W \in [6, 12]$), we manage to approximately derive that $v \in [0.188m/s, 0.212m/s]$ and $[0.173m/s, 0.196m/s]$, which minimize T_{total} for the object pursuit tasks on linear and circular paths, respectively.

Furthermore, the limitation of the model needs to be discussed. As the tracking time always equals to the actual time the target moves, $T_{tracking}$ is constant under such circumstances. Therefore, it is not necessary to give emphasis on the tracking time of an object pursuit task, but on the correction time. Depending on the target velocity (a uniform motion or a variable motion), $T_{correction}$ might be influenced in different manners. The reason is that the correction time may strongly rely on the extent to which the velocity and the trajectory of the moving target are predictable. In a variable motion, however, such a prediction is highly restricted. We may need the change of the velocity, *i.e.*, the acceleration, to fully address the problem. Accordingly, to model the tracking task of a variable motion is beyond the scope of the model.

Despite that pointing, steering and object pursuit are distinct in nature, the interaction models for describing these tasks, *i.e.*, Fitts' law, the steering law and the object pursuit law proposed in this chapter, share something in common. The way how the influence of path length (L) and target width (W) is combined (L/W) in object pursuit tasks very much resembles the way how the influence is combined in pointing and path steering tasks. This fact may be interpreted as the implication of involving such a combination in modeling a general interaction task. Although L and W may have different interpretations for different tasks, the underlying meaning for them is the same, *i.e.*, L always represents the distance that needs to be traversed by the movement, in parallel with the movement direction. W represents the extent to which the movement is constrained, not limited to a specific direction (see Figure 2.9). For a 3D movement, the constraint can also be extended with H and D , which stand for the constraint in height and depth, respectively. It is interesting to observe that the same combination of movement distance and constraints in width, height and depth (L/W , L/H and L/D) was used to model users' movement time [GB04, MB92, AZ03]. These combinations quantitatively indicate that decreasing the constraints to which the movement is subject and increasing the movement distance with the same factor have the same effect on the movement time. It seems that the effect of movement distance over movement constraint

is one of the most important indices of difficulty for an interaction task, which therefore deserves a further examination while modeling new interaction tasks.

One significant difference between the models of the three tasks stems from the additional variables that can intrinsically distinguish the task under observation from others. The selection of the variables must be representative and should avoid any details of the tasks. For example, the shape of a path, represented by path curvature, is a unique variable that can distinguish a path steering task from a pointing task. The velocity involved in the object pursuit model differentiates an object pursuit task from other interaction tasks, such as pointing and steering tasks. It would seem that the key link of modeling an interaction task is to identify the variables that can distinguish the task from others, next to the effect of L/W .

5.6 Conclusions

In this chapter, we introduced object pursuit as an interaction task that requires subjects to continuously track a 3D target that moves with a uniform velocity in a virtual environment. A complete pursuit movement was broken into a tracking phase and a correction phase. The tracking time is only governed by the path length crossed and the velocity with which the target moves, and is therefore fixed once the task has been established. For the correction phase, we developed a model that has been experimentally verified for object pursuit on linear and circular paths. The correction time can be modeled as a function of the path length, the object width and the velocity with which the object moves.

In addition, we have shown that an optimal target velocity, dependent of target width and path length, can be derived to minimize the total movement time. It can help us choose an appropriate velocity in designing user interfaces with moving objects. Interaction techniques that aim to facilitate object pursuit tasks can also be developed, taking the optimal velocity into account.

This is the first attempt at modeling object pursuit tasks in a virtual environment. An interesting finding is that the temporal characteristics of such an interaction task can partially be described as a function of the spatial properties of the tasks, particularly the use of the term L/W , which can also be found in Fitts' law and the steering law. In addition, we also propose that it is equally important to identify the variables that can distinguish the task from others. The term v , for example, is important for modeling object pursuit tasks.

Conclusions and Future Work

Research in virtual environment has been mostly active in two areas: one relates to the visual aspect that significantly influences users' sense of presence in the virtual environments, and one to the interactive aspect that determines the extent to which users can actively participate in the environments. The topics discussed in this thesis contributed to the interactive aspect that occurs between users and a desktop virtual environment. The objective was to model 3D pointing, steering and object pursuit tasks, and to gain an insight into users' movements through the analysis of interaction models and experimental results.

In this chapter, the main conclusions of the thesis are summarized, and suggestions for valuable extensions and future work are provided.

6.1 Conclusions

3D pointing, steering and object pursuit have been modeled in this thesis. Existing pointing and steering models have been employed and extended to 3D tasks in the virtual environment, and new steering and object pursuit models have been developed. This was achieved by carrying out formal user studies, verifying the interaction models through a statistical analysis on the empirical data, comparing the model performance according to the information loss (*e.g.*, AIC) and utilizing the models in obtaining a better understanding of the movements. The main conclusions are summarized below:

- The experimental results indicated that the 3D pointing movements collected from the real world and the virtual environment could both be modeled with the two-component model (see Figure 6.1), *i.e.*, the movements are composed of a ballistic phase and a correction phase. Using the two-component model, a comparison was made between real-world and virtual-world pointing movements. An interesting finding was that significant differences in pointing efficiency and accuracy were observed in each movement phase, particularly in the correction phase, where the time and the number of sub-movements involved were 84.06% and 92.59% higher in the virtual environment.
- In the study of 3D path steering, the steering law was evidenced to be able to model 3D manipulation tasks in very limited cases. The experimental results showed that a systematic effect of path curvature and orientation was missing from the steering law.

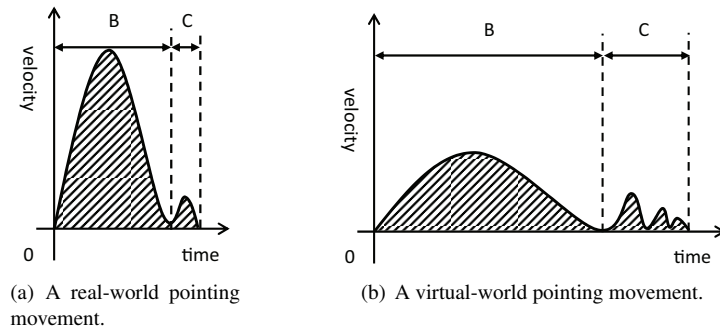


Figure 6.1: Pointing movements described with the two-component model. B: the ballistic phase; C: the correction phase. The shaded area represents the distance crossed during the movement.

By taking these variables into account, new steering models have been developed and experimentally proved to outperform the steering law. It was also shown that haptic feedback could be introduced to the path steering tasks as a means to improve the steering performance and that the steering time could be modeled as a function of the force feedback. The resulting model has been used to determine an appropriate force magnitude that optimizes users' performance.

A further analysis on the steering models showed that users' movements for the ball-and-tunnel task conducted with a 6-DOF stylus or a 3-DOF haptic device should not be described as an infinite number of goal-crossing movements as assumed by Accot and Zhai. Instead, the empirical data showed that they should better be modeled with the power laws that lie in between the Fitts' law and the steering law. As Fitts' law describes the pointing movements that consist of sub-movements and the steering law describes the steering movements as smooth and continuous movements, it implied that the ball-and-tunnel movements might be composed of a series of small and jerky sub-movements that are similar to the ballistic/correction movements observed in the pointing movements (see Figure 6.2). However, evidence showed that as haptic feedback is introduced to the task, the jerky sub-movements become smaller and the overall movement gets closer to "an infinite number of goal-crossing movements".

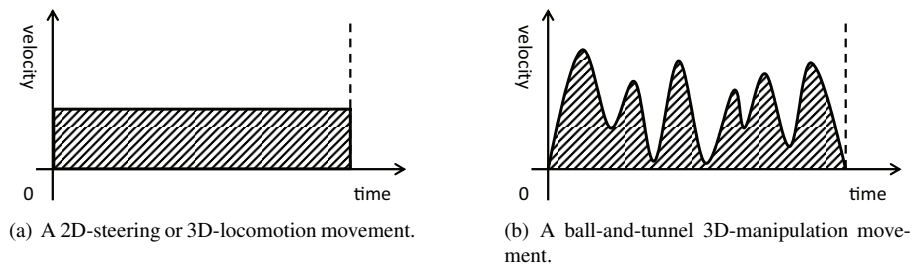


Figure 6.2: Steering movements for 2D, 3D locomotion and 3D manipulation tasks.

- A novel object pursuit law was developed, which models users' correction time of pursuing an object as a function of the object size, the velocity and the path length to be

crossed during the task. It has been shown that the model can be used to determine the optimal velocity of the object, with which the total pursuit time reaches its minimum.

- It was shown that Stevens' power law could be used to model the three interaction tasks involved in this thesis. Figure 6.3 summarizes the relationships between the total movement time (T) and the length to be traveled over the width of the constraint (L/W) for different interaction tasks, modeled with the power law. As illustrated, the power law can be used to identify different interaction tasks by examining the associate exponent and hence might serve as a general law for modeling interaction tasks.

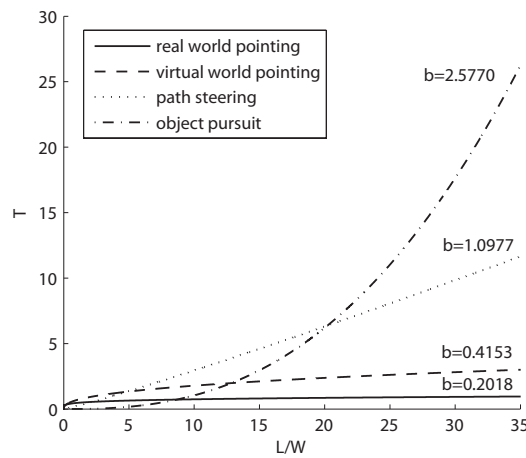


Figure 6.3: The relationships between T and L/W for different interaction tasks, described with Stevens' power law.

6.2 Future Work

There are several issues that have not yet been addressed in this thesis. A future extension of these issues may be of particular interest.

The measurement for the steering and object pursuit tasks decomposed the movements into two movement phases: one focuses on the actual time when users perform the interaction tasks and the other one focuses on the time when users fail to meet the requirements of the tasks and the corresponding corrections are made. Interaction models have been developed on the premise of such measurement choices. However, other ways of measuring the performance of path steering and object pursuit are also possible. For example, the sub-trials that were introduced for the path steering tasks were segmented by two adjacent correction phases. This may not be the optimal way to segment the overall trial, as a sub-trial can potentially consist of several sub-movements. A better way to actually resolve this problem is to adopt a more objective way of subdividing the (sub)trials, *e.g.*, using a similar parsing method as adopted for analyzing 3D pointing movements (see Appendix C.1 and C.2).

The 3D pointing movements in virtual reality were evidenced to be more time-consuming than their counterparts in the real world, particularly during the correction phase. In addition, the steering movements for the ball-and-tunnel task were found to be composed of a

series of small and jerky sub-movements. It is substantially different from Accot and Zhai's description, in which 2D path steering and 3D locomotion were modeled as continuous and smooth movements. It would be worth further investigating the reasons why these occur. In this thesis, we have discussed several aspects that might play a role. For example, it might be attributed to the intrinsic difficulty of 3D interaction in virtual reality, such as poor depth perception, low visual quality or lack of multimodal cues; it could be because of technical issues, such as the end-to-end latency of the VR system or the non-co-located experimental setup; it could be due to the learning effect, for which users need more practice in the virtual environment; or it might also be the way the interaction tasks were designed. Additional research is needed to substantiate these assumptions.

Another possible extension is to compare path steering that occurs in the real world with that in virtual reality, as what we have done for the pointing tasks. This work could help us better understand what the differences are between the real-world and virtual-world steering and what type of movement steering is in different environments (smooth movements or jerky movements), which may in return lead to new steering models.

Appendices

Appendix A

Data Processing and Model Analysis

A.1 Data Transformation

As most of the data analysis in this thesis involves ANOVA and regression, it is necessary to verify whether the raw data collected from the experiments meet the requirements of applying such statistical methods, *i.e.*, the assumptions of independence, normality and homoscedasticity of the data. If the requirements are not satisfied, data transformation needs to be performed before the statistical methods can be applied. This appendix gives an example of how data from Section 4.3.2 and 4.6.3 were transformed, and provides the arguments for these transformations [LMvL10].

A.1.1 Approach and Rationale

The logarithm transformation was employed as the main approach to preprocess the data in most of the experiments described in this thesis. The reasons for making such a choice are summarized below:

- **Methodology**

Movement time cannot be normally distributed as it is restricted to being positive. For example, the cumulative histograms of the sub-trial time in Figure A.1 show that the sub-trial time of Experiment 1 and 2 in Section 4.3.2 has asymmetrical distributions, which obviously deviate from the normality assumption made by the regression and ANOVA methods. A one-sample Kolmogorov-Smirnov test also evidences that the null hypothesis, *i.e.*, the empirical data have a normal distribution, was rejected ($D_1 = 0.361, p_1 = 0.002; D_2 = 0.622, p_2 = 2.96e - 07$) at the 5% significance level. In addition, the observed variance of sub-trial time increases as the time grows, which violates the assumption of homoscedasticity.

Taking the logarithm of time usually helps to bring the observed time distributions closer to normal distributions and makes the variance constant. As shown in Figure A.2, the empirical data after the transformation approximately follow a normal distribution, according to the cumulative histograms. This result has also been verified by the Kolmogorov-Smirnov test ($D_1 = 0.078, p_1 = 0.518; D_2 = 0.180, p_2 = 0.077$).

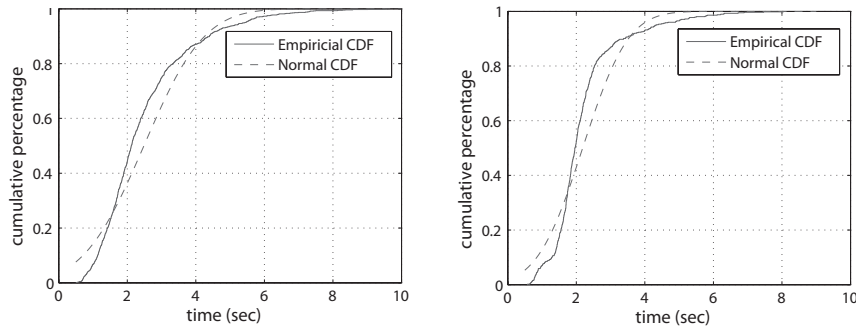


Figure A.1: Cumulative histograms of sub-trial time before the transformation. Left: Experiment 1 in Section 4.3.2; Right: Experiment 2 in Section 4.3.2. Both histograms (empirical CDF) show strong deviation from the Gaussian distribution fitting onto the data (normal CDF which has the same mean and standard deviation as the empirical data).

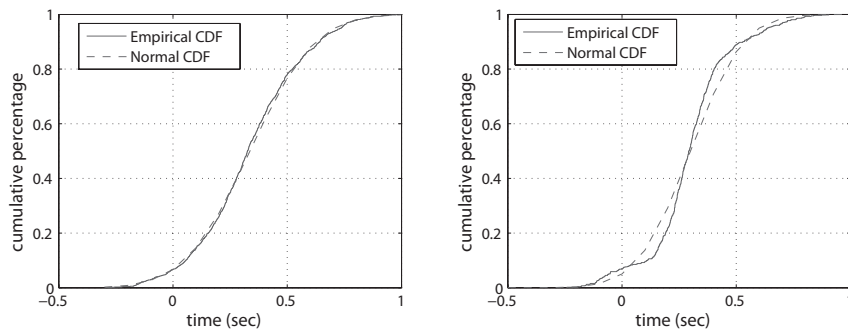


Figure A.2: Cumulative histograms of sub-trial time after the logarithmic transformation. Left: Experiment 1 in Section 4.3.2; Right: Experiment 2 in Section 4.3.2. The sub-trial time after the transformation approximately follows the Gaussian distribution fitting onto the data.

The studentized Breusch-Pagan test ($BP_1 = 0.4831, df_1 = 1, p_1 = 0.487; BP_2 = 2.0549, df_2 = 1, p_2 = 0.1517$) shows that there is not sufficient evidence against the constant variance assumption after the transformation. The way the experiments were designed guarantees the independence of the sub-trial time. Therefore, the assumptions of independence, normality and homoscedasticity are met after the logarithmic transformation.

Figure A.3 demonstrates another example by examining the data distribution before and after the transformation for Experiment 1 in Section 4.6.3. Block 1 and 2 represent the scenarios with and without force feedback, respectively.

• Modeling

Linear regression on $\log(\text{time})$ corresponds to a different class of functions (power functions that pass through the origin, see Section 2.5 for reasons) than linear regression on time. As the regression lines on time from our empirical data were observed to pass (approximately) through the origin, the power functions comprise a more general

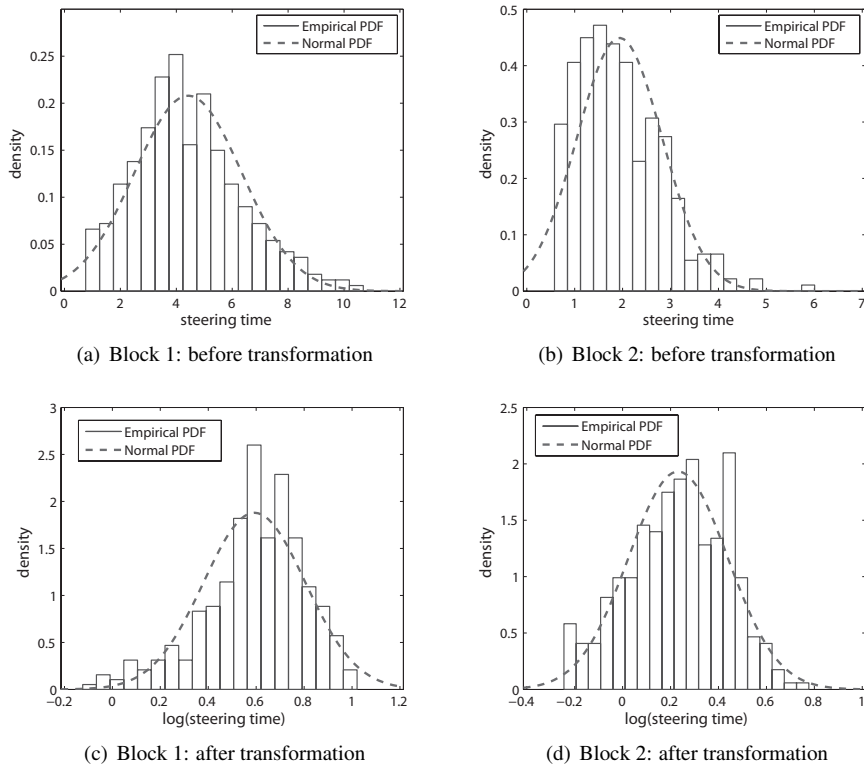


Figure A.3: The distribution of the steering time before and after the transformation.

class of models.

- **Interpretation**

In order to test the steering law, a test to determine whether or not time varies linearly with L/W is needed. By considering a larger class of models (power functions), we can translate this in a statistical test on the power, *i.e.*, testing whether or not the power is significantly different from 1.

- **Communication**

The different slopes that are found in linear regression lines on time can be translated into a shift for the regression lines on $\log(\text{time})$ (a percentage change, see Figure 4.13(b)). This is easier to communicate and moreover also shows that the effect of curvature is constant (in percentage) across *ID*s.

A.2 Confidence Intervals for Repeated-Measures Designs

The calculation of confidence intervals for repeated-measures designs uses the following equation as proposed in [LM94, ML03]:

$$CI = M_j \pm \sqrt{\frac{MS_{S \times C}}{n}}(t_{critical}) \quad (A.1)$$

where M_j is the mean for condition j , n is the number of observations associated with each mean and $MS_{S \times C}$ represents the mean square error for the interaction term (Subject \times Condition) in the repeated-measures ANOVA tables. The degrees of freedom for $t_{critical}$ are the degrees of freedom for the interaction term and the significance level selected throughout the thesis is $\alpha = 0.05$.

Appendix B

Path Steering as Goal-Crossing Movements

Accot and Zhai's steering law was formulated with the idea that a path steering task is composed of an infinite number of continuous goal-crossing tasks. This appendix briefly presents the procedure how the steering law was derived from this governing idea.

A goal-crossing task requires one to perform a rapid aimed movement from one goal to another by crossing both goals. Although a goal-crossing task by definition is different from a pointing task, researchers [AZ97, AZ02] have shown that goal-crossing tasks could also be modeled with Fitts' law. Thus, the index of difficulty (ID) of the goal-crossing task as shown in Figure B.1 (a) can be expressed as:

$$ID_1 = \log_2\left(\frac{L}{W} + 1\right) \quad (\text{B.1})$$

where L is the distance between the two goals and W is the height of each goal. If an additional goal is inserted in between the two goals such that it breaks the distance L into two equal lengths $L/2$, the task can be divided into two identical goal-crossing tasks, each of which has the same ID as shown below:

$$ID_{sub} = \log_2\left(\frac{L}{2W} + 1\right) \quad (\text{B.2})$$

Then, the ID for the overall task can be defined as the sum of the ID s for the two separate tasks, which is of form:

$$ID_2 = 2\log_2\left(\frac{L}{2W} + 1\right) \quad (\text{B.3})$$

By inference, if $N-1$ goals are inserted, the new ID should satisfy:

$$ID_N = N\log_2\left(\frac{L}{NW} + 1\right) \quad (\text{B.4})$$

Imagine the case when the number of goals tends to infinity, the task becomes a tunnel crossing (path steering) task which can be depicted by Figure B.1 (d). The ultimate ID can be described as:

$$ID_\infty = \frac{L}{W \ln 2} \quad (\text{B.5})$$

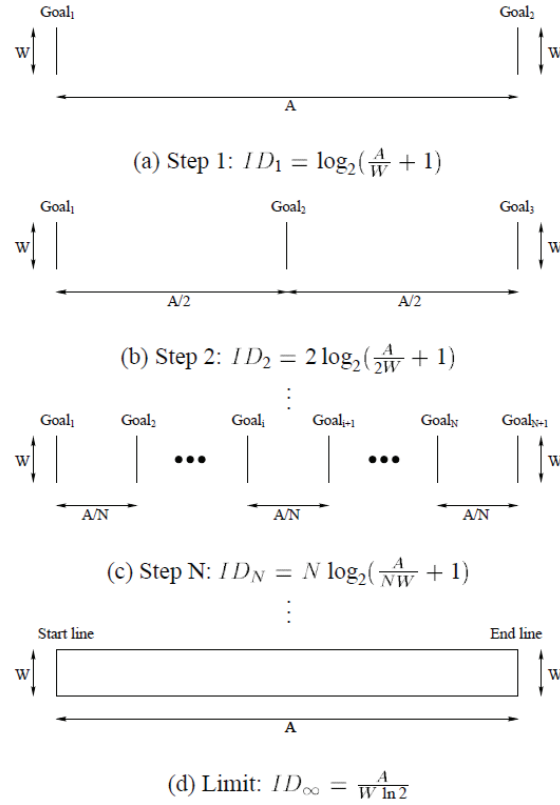


Figure B.1: The governing idea behind the steering law [AZ97]: a path steering task is composed of N Fitts'-law-describable goal-crossing tasks (N tends to infinity).

using a first order Taylor series expansion of $\log_2(1+x)$. The movement time of the tunnel crossing task can be consequently modeled as:

$$T = a + b ID = a + b \frac{L}{W} \quad (\text{B.6})$$

where the constant term $1/\ln 2$ is combined into the constant b . Equation B.6 is widely known as the steering law for the constant-width path steering tasks.

3D Movement Parsing

Meyer et al. [MAK⁺88] proposed a series of 1D movement parsing criteria (see Section 2.1.3) for dividing a 1D pointing movement into 3 basic types of sub-movements and assembling the sub-movements into phases. Although Meyer's criteria were considered powerful in breaking low-dimensional real-world movements into phases, their application in 3D virtual-world movement parsing could be difficult due to the additional dimension introduced to the movements and the differences between the real-world and virtual-world movements.

In this appendix, we [LvLNM09, LvL09] propose new parsing criteria that are especially developed for 3D pointing movements. The movement parsing criteria described in this appendix are composed of non-real-time parsing¹ and real-time parsing. The non-real-time parsing analyzes the movements as a post-processing step where the criteria are applied to the recorded movements, while the real-time parsing breaks the movements into phases in real time as users are performing the tasks.

C.1 Non-Real-Time 3D Movement Parsing

The recorded position coordinates (x , y , z) from the experiment described in Section 3.2.1 are filtered as a function of time to avoid spurious details as a result of taking derivatives of noisy signals. A Gaussian filter with a standard deviation of $25ms$ is used to preprocess the data.

The movement parsing starts by dividing movements into distinct movement intervals, which are separated by pauses. A pause is defined as an interval in between intervals with a zero velocity in which the speed of the cursor remains below 0.05 times the movement peak speed. For each of the detected movement intervals, it is determined whether or not it makes a considerable contribution to reaching the target. If the path length of a movement interval is more than 25% of the total path length, it is considered to be part of the ballistic phase. All movement intervals after the ballistic movement intervals are considered to be part of the correction phase.

The detected movement intervals are subsequently divided into sub-movements. One reason to divide movements into sub-movements is to use this division for a more detailed description of the movement performance. The other reason is to use this division to determine whether or not the last movement interval of the ballistic phase contains some corrective

¹This is a contribution of K. Nieuwenhuizen, included in the paper [LvLNM09].

sub-movements at the end. Meyer's criteria are adjusted such that they could be applied to velocity profiles based on path length. Sub-movements are detected according to the following three types:

- a type-1 sub-movement occurs when the speed increases from zero to a value that is above 0.05 times the movement peak speed (due to the way the intervals are defined, this only occurs at the beginning of a movement interval);
- a type-2 sub-movement occurs at a zero-crossing of acceleration from negative to positive (in combination with a positive jerk that exceeds 0.01 times the maximally observed jerk);
- a type-3 sub-movement occurs at a zero-crossing of jerk from positive to negative (in combination with a negative value of its derivative that exceeds 0.01 times the maximally observed value).

Movement Type	Meaning	Math
type 1	overshoot from target	velocity = 0
type 2	undershoot to target	acceleration = 0
type 3	decrease in deceleration	jerk = 0

Table C.1: Sub-movement types in association with physical and mathematical definitions.

The thresholds on the slopes of the zero-crossings are applied to avoid the detection of a sub-movement during the small involuntary tremor or the slow drift. The minimal requirements for a sub-movement proposed by Meyer et al. were specific for their 1D rotation task and also needed to be adjusted for the 3D interaction task. We chose as the minimal requirements for a sub-movement that it should traverse a distance of at least 3mm and last for at least 75ms, while the maximum velocity should exceed 0.05 times the maximally observed speed. Sub-movements that do not meet these requirements are combined with neighboring sub-movements. This criterion avoids detecting many small, but insignificant sub-movements.

The corrective sub-movements that occur during the last movement interval of the ballistic phase are considered to assist in positioning the pointer within the target area, *i.e.*, they are considered to belong to the correction phase. If the last movement in the ballistic phase consists of multiple sub-movements, the ballistic phase ends at the first type-2 sub-movement that occurs in the last 75%-95% of the traveled path length. Type-3 sub-movements are only considered to indicate subtle accuracy regulations and are therefore not used to indicate the end of the ballistic phase. If the last ballistic movement interval consists of only one sub-movement, the end of this movement coincides with the end of the ballistic phase (and the start of the correction phase).

C.2 Real-Time 3D Movement Parsing

Different from the non-real-time movement parsing, the absence of global overview of the complete movement makes it difficult to discriminate the correction phase from the ballistic phase in real time. For instance, the correction phase can only start after the global peak of a velocity profile has been detected; but in real time, it is not possible to distinguish between the global peak and a local peak of the velocity. In this section, we introduce a procedure which can parse 3D movement in real time. The entire procedure includes 5 steps:

Data Preprocessing

During data preprocessing, a velocity profile is constructed once every 1/120s after a sample of position has been received from the input tracker. The velocity profile is smoothed by taking the average of velocity values for every 10 continuous samplings. The acceleration and jerk profiles are then derived from the smoothed velocity.

Global Peak Detection

The global peak of a velocity profile is detected if all the following three conditions are met:

- A zero-crossing of acceleration from positive to negative is reached;
- The velocity is greater than a threshold a ;
- The time spent is longer than a threshold b .

The thresholds a and b are used such that small local peaks in the velocity profile are not considered as the global peak. The specific values used in the experiment are derived from a pilot study.

Sub-Movement Detection

Part of a movement is defined as a sub-movement when any of the three conditions is met at the end of the sub-movement:

- A zero-crossing of velocity from positive to negative is reached (type 1);
- A zero-crossing of acceleration from negative to positive is reached (type 2);
- A zero-crossing of jerk from positive to negative is reached (type 3).

The criteria above resemble Meyer's 1D movement parsing criteria.

End of Ballistic Phase Determination

The end of the ballistic phase is defined as the moment all the following conditions are satisfied:

- The global peak has been observed;
- A type 1, 2 or 3 sub-movement is detected;
- The current position is within a reasonable distance c from the target.

Threshold c is also experimentally determined by the pilot study.

Target Prediction

Once the end of ballistic phase is detected, the 'nearest neighbor' is introduced as the target prediction algorithm, since we assume the cursor has already entered the vicinity of the intended target at the moment. Therefore, there should be little chance that a wrong target is picked up. It might occur when the subject reaches a wrong target on purpose or has a dramatic change in behavior while passing through an unexpected target. But in practice this rarely happens.

Algorithm 1 shows the pseudo-code for the above steps.

Algorithm 1 Real-time movement parsing

```

for Every Motion Event do
  if GlobalPeak==0 and  $acc_{previous} \geq 0$  and  $acc_{now} < 0$  and  $v_{now} \geq a$  and  $T \geq b$  then
    GlobalPeak=1
  else if GlobalPeak=1 and ( $v_{previous} \geq 0$  and  $v_{now} < 0$  or  $acc_{previous} \leq 0$  and  $acc_{now} > 0$ 
or  $jer_{previous} \geq 0$  and  $jer_{now} < 0$ ) then
    for Every Target do
      Compute Distance between Cursor and Target
    end for
    Find the Nearest Target "A"
    if  $Dist_{CursorTargetA} \leq c$  then
      Apply Interaction Techniques
    end if
  end if
end for

```

C.2.1 Parsing Criteria Evaluation

The real-time movement parsing algorithm was applied to the $60 \times 3 \times 12$ trials in the experiment (see Section 3.3.2), independent of the interaction techniques. As the experimental results indicate, 5 trials out of 2160 were incorrectly predicted, resulting in an accuracy of 99.77%. According to the algorithm, only 142 out of 2160 pointing movements (6.57%) were completed without the correction phase.

The proportion of the types of sub-movements determined by the algorithm is depicted in Figure C.1, where type-3 dominates the sub-movement types with 50% and type-1 has the lowest percentage with 13%. It demonstrates that users have more opportunities to decrease in the rate of deceleration than to undershoot or overshoot (see Table C.1) when sub-movements occur.

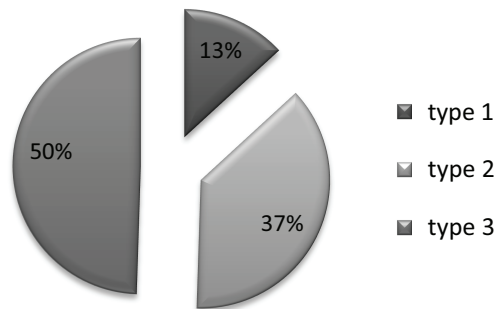


Figure C.1: Sub-movement types.

Appendix **D**

Velocity Profiles in Long Path Steering

This appendix demonstrates 24 examples of velocity profiles in long path steering tasks as conducted in Section 4.5. The velocity profiles are plotted as a function of movement time and are smoothed by a Savitzky-Golay Filter using a polynomial of the ninth degree. Each profile is a random sampling from one user. The goal for such an illustration is to show that users' steering behavior in long path steering tasks should not be modeled as an infinite number of continuous goal-crossing movements, as such behavior would result in a smooth uniform motion which is of constant velocities during the steering tasks. As shown in Figure D.1 and D.2, there are usually several local minima and maxima observed in the movement velocity profiles, which might be used to decompose the movement into sub-movements. Obviously, this contradicts what the steering law implies. On average, the number of sub-movements involved in linear path steering tasks as shown in Figure D.1 is smaller than that in circular path steering tasks as shown in Figure D.2.

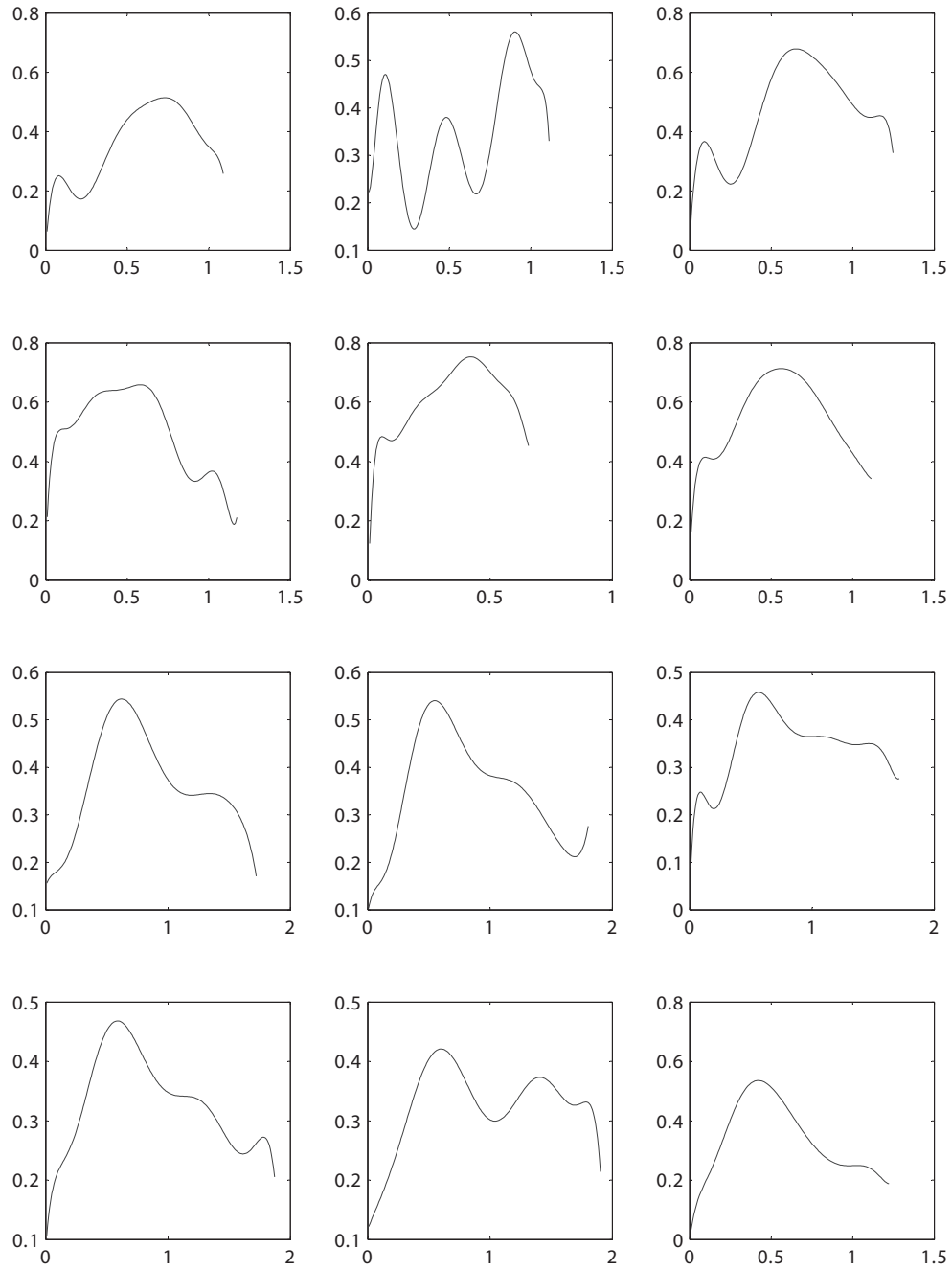


Figure D.1: Velocity profiles of linear path steering tasks. X-axis: time; y-axis: velocity.

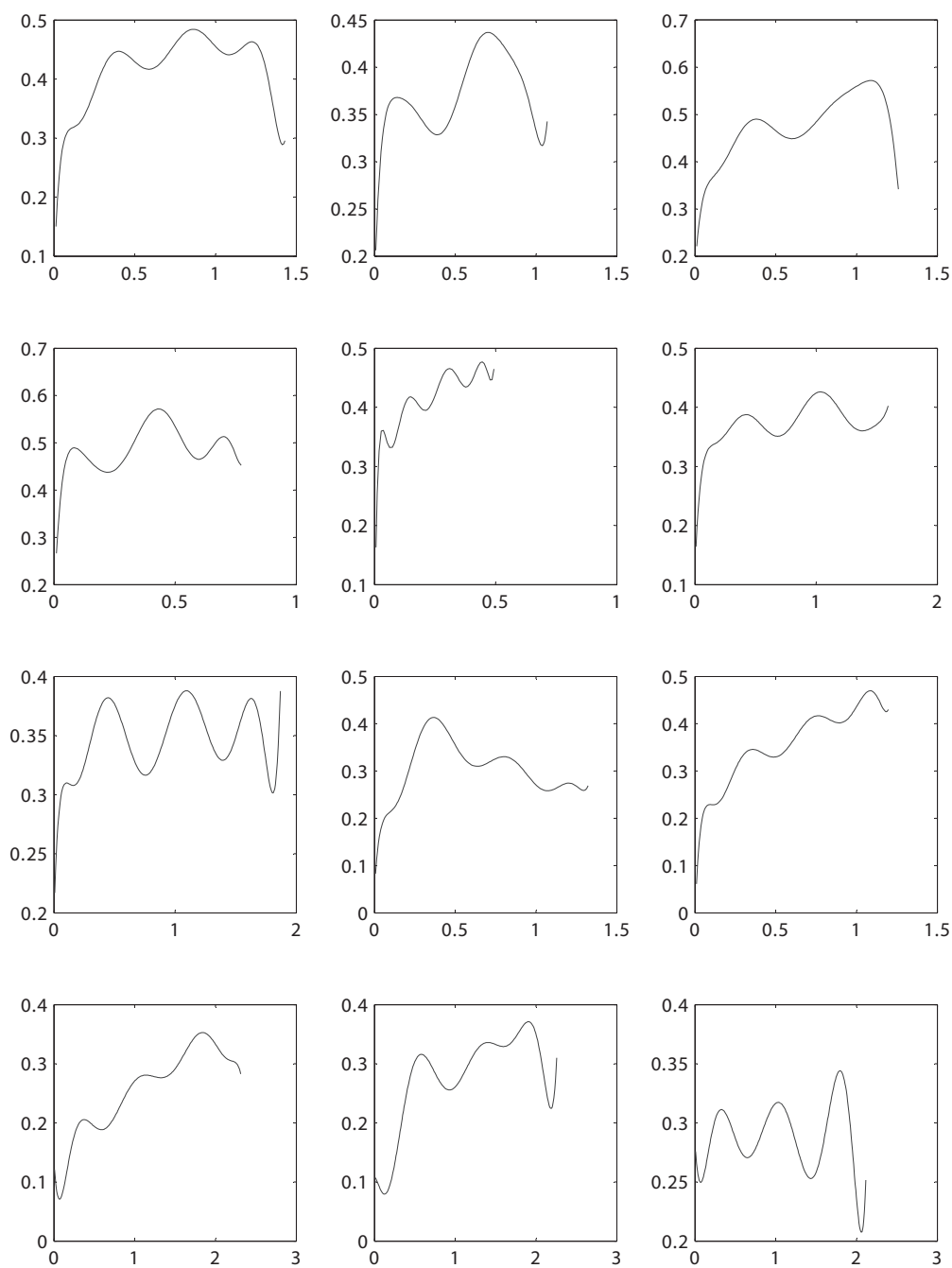


Figure D.2: Velocity profiles of circular path steering tasks ($\rho = 8$).

Bibliography

- [ACN02] Laura Lynn Arns and Carolina Cruz-Neira. *A new taxonomy for locomotion in virtual environments*. PhD thesis, Iowa State University Ames, 2002.
- [Ahl05] David Ahlstrom. Modeling and improving selection in cascading pull-down menus using Fitts' law, the steering law and force fields. In *Proceedings of the SIGCHI conference on Human factors in computing systems 2005*, pages 61–70, 2005.
- [Aka74] Hirotugu Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.
- [AS99] Ricardo S. Avila and Lisa M. Sobierajski. A haptic interaction method for volume visualization. In *VIS'96: Proceedings of the 7th conference on Visualization*, pages 197–204, 1999.
- [Atw06] Jeff Atwood. Fitts' law and infinite width. Available at <http://www.codinghorror.com/blog/2006/08/fitts-law-and-infinite-width.html>, 2006.
- [AZ97] Johnny Accot and Shumin Zhai. Beyond Fitts' law: Models for trajectory-based HCI tasks. In *CHI'97: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 295–302, 1997.
- [AZ99] Johnny Accot and Shumin Zhai. Performance evaluation of input devices in trajectory-based tasks: An application of the steering law. In *CHI'99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 466–472, 1999.
- [AZ01] Johnny Accot and Shumin Zhai. Scale effects in steering law tasks. In *CHI'01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1–8, 2001.
- [AZ02] Johnny Accot and Shumin Zhai. More than dotting the i's - foundations for crossing-based interfaces. In *CHI'02: Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*, pages 73–80, 2002.

- [AZ03] Johnny Accot and Shumin Zhai. Refining Fitts' law models for bivariate pointing. In *CHI'03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 193–200, 2003.
- [BA04] K. P. Burnham and D. R. Anderson. Multimodel inference: understanding AIC and BIC in model selection. *Sociological Methods and Research*, 33:261–304, 2004.
- [Bal04] Ravin Balakrishnan. “Beating” Fitts' law: Virtual enhancements for pointing facilitation. *International Journal of Human-Computer studies*, 61:857–874, 2004.
- [BB87] Ronald M. Baecker and William A. S. Buxton. *Readings in Human-Computer Interaction: A Multidisciplinary Approach*. Morgan Kaufmann Publishers, 1987.
- [BCR⁺03] Patrick Baudisch, Edward Cutrell, Dan Robbins, Mary Czerwinski, Peter T. Benjamin Bederson, and Alex Zierlinger. Drag-and-Pop and Drag-and-Pick: Techniques for accessing remote screen content on touch and pen-operated systems. In *Interact '03*, pages 57–64, 2003.
- [BGBL04] Renaud Blanch, Yves Guiard, and Michel Beaudouin-Lafon. Semantic pointing: Improving target acquisition with control-display ratio adaptation. In *CHI'04: ACM Conference on Human Factors in Computing Systems*, pages 519–526, 2004.
- [BGM⁺72] W. D. A. Beggs, J. C. Graham, T. H. Monk, M. R. W. Shaw, and C. I. Howarth. Can Hick's law and Fitts' law be combined? *Acta Psychologica*, 36:348–357, 1972.
- [BH70] W. D. A. Beggs and C. I. Howarth. Movement control in a repetitive motor task. *Nature*, 225(5234):752–753, 1970.
- [BH99] Doug A. Bowman and Larry F. Hodges. Formalizing the design, evaluation, and application of interaction techniques for immersive virtual environments. *Journal of Visual Languages and Computing*, 10(1):37–53, 1999.
- [Bie88] Eric A. Bier. *Snap-Dragging: Interactive Geometric Design in Two and Three Dimensions*. PhD thesis, EECS Department, University of California, Berkeley, May 1988.
- [BJH01] Doug A. Bowman, Donald B. Johnson, and Larry F. Hodges. Testbed evaluation of virtual environment interaction techniques. *Presence: Teleoperators and Virtual Environments*, 10(1):75–79, February 2001.
- [BL00] M. Beaudouin-Lafon. Instrumental interaction: An interaction model for designing post-WIMP user interfaces. In *CHI'00: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 446–453, 2000.
- [BM83] A. Terry Bahill and Jack D. McDonald. Smooth pursuit eye movements in response to predictable target motions. *Vision Research*, 23(12):1573–1583, April 1983.

- [Bow98] Doug A. Bowman. Interaction techniques for immersive virtual environments: Design, evaluation, and application. *Journal of Visual Languages and Computing*, 10:37–53, 1998.
- [Bro99] Frederick P. Brooks. What’s real about virtual reality? *IEEE Computer Graphics and Applications*, 19(6):16–27, 1999.
- [Bux83] William Buxton. Lexical and pragmatic considerations of input structures. *Computer Graphics*, 17(1):31–37, 1983.
- [Car81] L. G. Carlton. Processing visual feedback information in movement control. *Journal of Experimental Psychology: Human Perception and Performance*, 7:1019–1030, 1981.
- [CEB78] Stuart K. Card, William K. English, and Betty J. Burr. Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a CRT. *Ergonomics*, 21(8):601–613, 1978.
- [CF07] Noah J. Cowan and Eric S. Fortune. The critical role of locomotion mechanics in decoding sensory systems. *The Journal of Neuroscience*, 27(5):1123–1128, January 2007.
- [CHWS88] J. Callahan, D. Hopkins, M. Weiser, and B. Shneiderman. An empirical comparison of pie vs. linear menus. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, pages 95–100, 1988.
- [CMR90] Stuart K. Card, Jock D. Muckinlay, and George G. Robertson. The design space of input devices. In *CHI’90: Proc. SIGCHI conference on Human factors in computing systems*, pages 117–124, 1990.
- [Cro56] E. R. F. W. Crossman. *The measurement of perceptual load in manual operations*. PhD thesis, Birmingham University, 1956.
- [CrWGT03] Sarah H. Creem-regehr, Peter Willemsen, Amy A. Gooch, and William B. Thompson. The influence of restricted viewing conditions on egocentric distance perception: Implications for real and virtual environments. *Perception*, 34:191–204, 2003.
- [CVBC08] Gery Casiez, Daniel Vogel, Ravin Balakrishnan, and Andy Cockburn. The impact of control-display gain on user performance in pointing tasks. *Human-Computer Interaction*, 23(3):215–250, July 2008.
- [Del98] H. Delingette. Toward realistic soft-tissue modeling in medical simulation. *Proceedings of the IEEE*, 86(3):512–523, 1998.
- [DKM99] Sarah A. Douglas, Arthur E. Kirkpatrick, and Scott I. Mackenzie. Testing pointing device performance and user assessment with the ISO 9241, Part 9 standard. In *CHI’99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 215–222, 1999.
- [DMH00] Jack Tigh Dennerlein, David B. Martin, and Christopher Hasser. Force-feedback improves performance for steering and combined steering-targeting tasks. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 423–429, 2000.

- [Dru71] C. G. Drury. Movements with lateral constraint. *Ergonomics*, 14:293–305, 1971.
- [DY01] Jack Tigh Dennerlein and Maria C. Yang. Haptic force-feedback devices for the office computer: Performance and musculoskeletal loading issues. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 43(2):278–286, 2001.
- [EHC01] D Elliott, W F Helsen, and R Chua. A century later: Woodworth’s (1899) two-component model of goal directed aiming. *Psychological Bulletin*, 3:342–357, 2001.
- [EHG⁺10] Digby Elliott, Steve Hansen, Lawrence E. M. Grierson, James Lyons, Simon J. Bennett, and Spencer J. Hayes. Goal-directed aiming: Two components but multiple processes. *Psychological Bulletin*, 136(6):1023–1044, 2010.
- [EHMT04] Digby Elliott, Steven Hansen, Jocelyn Mendoza, and Luc Tremblay. Learning to optimize speed, accuracy, and energy expenditure: A framework for understanding speed-accuracy relations in goal-directed aiming. *Journal of Motor Behavior*, 36(3):339–351, 2004.
- [Epp86] Brian W. Epps. Comparison of six cursor control devices based on Fitts’ law models. *Human Factors and Ergonomics Society Annual Meeting Proceedings*, 30(4):327–331, 1986.
- [FIB95] George W. Fitzmaurice, Hiroshi Ishii, and William Buxton. Bricks: Laying the foundations for graspable user interfaces. In *CHI’95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 442–449. ACM Press, 1995.
- [Fit] http://en.wikipedia.org/wiki/Fitts's_law.
- [Fit54] Paul M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47(6):381–391, June 1954.
- [FKK07] Scott Frees, G. Drew Kessler, and Edwin Kay. PRISM interaction for enhancing control in immersive virtual environments. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 14(1), May 2007.
- [Flo78] K. Flowers. Some frequency response characteristics of parkinsonism on pursuit tracking. *Brain*, 101(1):19–34, 1978.
- [Fur06] Borko Furht. *Encyclopedia of Multimedia*. Springer, first edition, 2006.
- [FWC84] James D. Foley, Victor L. Wallace, and Peggy Chan. The human factors of computer graphics interaction techniques. *IEEE Computer Graphics and Applications*, 4(11):13–48, 1984.
- [GB04] Tovi Grossman and Ravin Balakrishnan. Pointing at trivariate targets in 3D environments. In *CHI’04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 447–454, 2004.

- [GBBL04] Y. Guiard, R. Blanch, and M. Beaudouin-Lafon. Object pointing: A complement to bitmap pointing in GUIs. In *Proceedings of Graphics Interface*, pages 9–16, 2004.
- [GHB⁺06] Tovi Grossman, Ken Hinckley, Patrick Baudisch, Maneesh Agrawala, and Ravin Balakrishnan. Hover widgets: Using the tracking state to extend the capabilities of pen-operated devices. In *CHI'06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 861–870, 2006.
- [GMK00] Diane C. Gooding, Meghan D. Miller, and Thomas R. Kwapil. Smooth pursuit eye tracking and visual fixation in psychosis-prone individuals. *Psychiatry Research*, 93(1):43–54, February 2000.
- [GOD85] J. GODTHELP. Precognitive control: open- and closed-loop steering in a lane-change manoeuvre. *Ergonomics*, 28(10):1419–1438, 1985.
- [Hal07] Kevin Hale. Visualizing Fitts's law. <http://particletree.com/features/visualizing-fittss-law/>, 2007.
- [HD92] R. Held and N. Durlach. Telepresence. *Presence: Teleoperators and Virtual Environments*, 1(1):109–112, 1992.
- [Hes81] R. A. Hess. Pursuit tracking and higher levels of skill development in the human pilot. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-11:262–273, 1981.
- [Hop91] Don Hopkins. The design and implementation of pie menus. *Dr. Dobb's Journal*, December 1991.
- [HS94] E. R. Hoffmann and I. H. Sheikh. Effect of varying target height in a Fitts' movement task. *Ergonomics*, 37(6):1071–1088, 1994.
- [HvDG94] Kenneth P. Herndon, Andries van Dam, and Michael Gleicher. The challenges of 3D interaction. *SIGCHI Bulletin*, 26(4):36–43, October 1994.
- [IMB⁺92] William G. Iacono, Margaret Moreau, Morton Beiser, Jonathan A.E. Fleming, and Tsung-Yi Lin. Smooth-pursuit eye tracking in first-episode psychotic patients and their relatives. *Progress in Neuro-Psychopharmacology and Biological Psychiatry*, 101(1):104–116, February 1992.
- [ISO98] ISO. Ergonomic requirements for office work with visual display terminals (VDTs) - Part 9: Requirements for non-keyboard input devices. *International Standard, International Organization for Standardization*, 1998.
- [JRWM80] R. J. Jagacinski, D. W. Repperger, S. L. Ward, and M. S. Moran. A test of Fitts' law with moving targets. *Human Factors*, pages 225–233, 1980.
- [KB95] Paul Kabbash and William Buxton. The “prince” technique: Fitts' law and selection using area cursors. In *CHI'95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 273–279, 1995.
- [KE88] B. H. Kantowitz and G. C. Elvers. Fitts' law with an isometric controller: effects of order of control and control-display gain. *Journal of Motor Behavior*, 20(1):53–66, March 1988.

- [Kee07] Daniel F. Keefe. *Interactive 3D Drawing for Free-Form Modeling in Scientific Visualization and Art: Tools, Methodologies, and Theoretical Foundations*. PhD thesis, Brown University, May 2007.
- [KFG98] Michael A. Khan, Ian M. Franks, and David Goodman. The effect of practice on the control of rapid aiming movements: Evidence for an interdependency between programming and feedback processing. *The Quarterly Journal of Experimental Psychology*, 51(2):425–443, 1998.
- [KGS07] Raghavendra S. Kattinakere, Tovi Grossman, and Sriram Subramanian. Modeling steering within above-the-surface interaction layers. In *Proceedings of the SIGCHI conference on Human factors in computing systems 2007*, pages 317–326, 2007.
- [KK78] B. H. Kantowitz and J. L. Knight. Testing tapping time-sharing: Attention demands of movement amplitude and target width. *Information processing in motor control and learning*, pages 205–227, 1978.
- [KLJ⁺11] Yongwan Kim, G. A. Lee, Dongsik Jo, Ungyeon Yang, Gihong Kim, and Jinah Park. Analysis on virtual interaction-induced fatigue and difficulty in manipulation for interactive 3D gaming console. In *ICCE'11: IEEE International Conference on Consumer Electronics*, pages 269–270, 2011.
- [KMIH05] Hilde Keuning, Tom K. J. Monné, Wijnand A. IJsselsteijn, and Adrianus J. M. Houtsma. The form of augmented force-feedback fields and the efficiency and satisfaction in computer-aided pointing tasks. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 47(2):418–429, 2005.
- [Kva80] T. O. Kvalseth. An alternative to Fitts' law. *Bulletin of the Psychonomic Society*, 16:371–373, 1980.
- [LbML11] Lei Liu, Jean Bernard Martens, and Robert Van Liere. Revisiting path steering for 3D manipulation tasks. *International Journal of Human-Computer Studies*, 69(3):170–181, March 2011.
- [LD03] S. D. Laycock and A. M. Day. Recent developments and applications of haptic devices. *Computer Graphics Forum*, 22:117–132, July 2003.
- [LKHF95] Kevin N. Laland, Jochen Kumm, John D. Van Horn, and Marcus W. Feldman. A gene-culture model of human handedness. *Behavior Genetics*, 25(5):433–445, 1995.
- [LM94] Geoffrey R. Loftus and Michael E.J. Masson. Using confidence intervals in within-subject designs. *Psychonomic Bulletin and Review*, 1:476–490, 1994.
- [LMT87] S. G. Lisberger, E. J. Morris, and L. Tychsen. Visual motion processing and sensory-motor integration for smooth pursuit eye movements. *Neuroscience*, 10:97–129, March 1987.
- [LMvL10] Lei Liu, Jean-Bernard Martens, and Robert van Liere. Revisiting path steering for 3D manipulation tasks. In *3DUI'10: Proceedings of the 2010 IEEE Symposium on 3D User Interfaces*, pages 39–46, 2010.

- [LvL09] Lei Liu and Robert van Liere. Designing 3D selection techniques using ballistic and corrective movements. In *JVRC'09: Joint Virtual Reality Conference of EGVE-ICAT-EuroVR*, pages 1–8, 2009.
- [LvL10] Lei Liu and Robert van Liere. The effect of varying path properties in path steering tasks. In *JVRC'10: Proceedings of Joint Virtual Reality Conference of EuroVR - EGVE - VEC 2010*, pages 9–16, 2010.
- [LvLK11a] Lei Liu, Robert van Liere, and K. J. Kruszyński. Comparing path steering between non-haptic and haptic 3D manipulation tasks: Users' performance and models. In *GRVR11: Proceedings of the IASTED International Conference on Graphics and Virtual Reality 2011*, 2011.
- [LvLK11b] Lei Liu, Robert van Liere, and K. J. Kruszyński. Modeling the effect of force feedback for 3D steering tasks. In *JVRC'11: Proceedings of Joint Virtual Reality Conference of EuroVR - EGVE 2011*, 2011.
- [LvLNM09] Lei Liu, Robert van Liere, Karin Nieuwenhuizen, and Jean-Bernard Martens. Comparing aimed movements in the real world and in virtual reality. In *VR'09: Proceedings of the 2009 IEEE Virtual Reality*, pages 219–222, 2009.
- [Mac89] I. S. MacKenzie. A note on the information-theoretic basis for Fitts' law. *Journal of Motor Behavior*, pages 323–330, 1989.
- [Mac91] I. Scott MacKenzie. *Fitts' Law as a Performance Model in Human-Computer Interaction*. PhD thesis, University of Toronto, 1991.
- [Mac92] I. Scott MacKenzie. Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction*, 7(1):91–139, 1992.
- [MAK⁺88] D. E. Meyer, R. Abrams, S. Kornblum, C. E. Wright, and J. E. K. Smith. Optimality in human motor performance: Ideal control of rapid aimed movements. *Psychological Review*, 95:340–370, 1988.
- [MB92] I. Scott Mackenzie and William Buxton. Extending Fitts' law to two-dimensional tasks. In *CHI'92: Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 219–226. New York:ACM Press, 1992.
- [MB93] I. Scott MacKenzie and William Buxton. A tool for the rapid evaluation of input devices using Fitts' law models. *SIGCHI Bulletin*, 25(3):58–63, 1993.
- [MBS97] Mark R. Mine, Frederick P. Brooks, and Carlo H. Sequin. Moving objects in space: Exploiting proprioception in virtual-environment interaction. In *SIGGRAPH'97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 19–26, 1997.
- [MCB⁺11] Liang Ma, Damien Chablat, Fouad Bennis, Wei Zhang, Bo Hu, and François Guillaume. Fatigue evaluation in maintenance and assembly operations by digital human simulation in virtual environment. *Journal Virtual Reality - Special issue on Virtual Manufacturing and Construction*, 15(1):55–68, 2011.
- [MH93] A. Modjtahedzadeh and R. Hess. A model of driver steering control behaviour for use in assessing vehicle handling qualities. *Journal of Dynamic Systems, Measurement, and Control*, 115:456–464, 1993.

- [MI01] Atsuo Murata and Hirokazu Iwase. Extending Fitts' law to a three-dimensional pointing task. *Human Movement Science*, 20(6):791–805, December 2001.
- [ML03] Michael E.J. Masson and Geoffrey R. Loftus. Using confidence intervals for graphically based data interpretation. *Canadian Journal of Experimental Psychology*, 57:203–220, 2003.
- [MM00] Maurice R. Masliah and Paul Milgram. Measuring the allocation of control in a 6 degree-of-freedom docking experiment. In *CHI'00: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 25–32, 2000.
- [MO98] I. Scott MacKenzie and Aleks Oniszczak. A comparison of three selection techniques for touchpads. In *CHI'98: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 336–343, 1998.
- [MPME04] S. E. Medland, I. Perelle, V. De Monte, and L. Ehrman. Effects of culture, sex, and age on the distribution of handedness: an evaluation of the sensitivity of three measures of handedness. *Laterality: Asymmetries of Body, Brain and Cognition*, 9(3):287–297, 2004.
- [MPT99] William A. McNeely, Kevin D. Puterbaugh, and James J. Troy. Six degree-of-freedom haptic rendering using voxel sampling. In *SIGGRAPH'99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 401–408, 1999.
- [MR76] G. K. Jr. Miller and D. R. Riley. The effect of visual-motion time delays on pilot performance in a pursuit tracking task. In *Visual and Motion Simulation Conference*, January 1976.
- [MS94] Thomas H. Massie and Kenneth J. Salisbury. The PHANTOM haptic interface: A device for probing virtual objects. In *Proceedings of the ASME Winter Annual Meeting Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 295–300, 1994.
- [MSB91] I. Scott MacKenzie, Abigail Sellen, and William A. S. Buxton. A comparison of input devices in element pointing and dragging tasks. In *CHI'91: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 161–166, 1991.
- [MW69] D. McRuer and D. Weir. Theory of manual vehicular control. *Ergonomics*, 12(4):599–633, 1969.
- [NbMLvL09] Karin Nieuwenhuizen, Jean Bernard Martens, Lei Liu, and Robert van Liere. Insights from dividing 3D goal-directed movements into meaningful phases. *IEEE Computer Graphics and Applications*, 29(6):44–53, 2009.
- [NCDT10] Abedeldjallil Nacéri, Ryad Chellali, Fabien Dionnet, and Simone Toma. Depth perception within virtual environments: A comparative study between wide screen stereoscopic displays and head mounted devices. *International Journal on Advances in Intelligent systems*, 3(12):51–64, 2010.

- [New94] William Newman. A preliminary analysis of the products of HCI research, using pro forma abstracts. In *CHI'94: Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*, pages 278–284, 1994.
- [NKK04] Satoshi Naito, Yoshifumi Kitamura, and Fumio Kishino. Steering law in an environment of spatially coupled style with matters of pointer size and trajectory width. *Computer Human Interaction 6th Asia Pacific Conference - APCHI 2004*, 3101:305–316, 2004.
- [Pas06] R. Pastel. Measuring the difficulty of steering through corners. In *CHI'06: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1087–1096, 2006.
- [PBWI96] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. The Go-Go interaction technique: Non-linear mapping for direct manipulation in VR. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, pages 79 – 80, 1996.
- [PKCR05] Jodie M. Plumert, Joseph K. Kearney, James F. Cremer, and Kara Recker. Distance perception in real and virtual environments. *ACM Transactions on Applied Perception*, 2:216–233, 2005.
- [PLDA00] G. Picinbono, J. Lombardo, H. Delingette, and N. Ayache. Improving realism of a surgery simulator: Linear anisotropic elasticity, complex interactions and force extrapolation. Technical report, INRIA Sophia Antipolis, France, September 2000.
- [Ras59] N. Rashevsky. Mathematical biophysics of automobile driving. *Bulletin of Mathematical Biophysics*, 21:375–385, 1959.
- [RBGA] Jannick P. Rolland, Christina A. Burbeck, William Gibson, and Dan Ariely. Towards quantifying depth and size perception in 3D virtual environments.
- [RDLT09] Gabriel Robles-De-La-Torre. Virtual reality: Touch / haptic. *Sage Encyclopedia of Perception*, 2:1036–1038, 2009.
- [RSB81] L. Reid, E. Solowka, and A. Billing. A systematic study of driver steering behavior. *Ergonomics*, 24(6):447–462, 1981.
- [RVL90] R. G. Radwin, G. C. Vanderheiden, and M. L. Lin. A method for evaluating head-controlled computer input devices using Fitts' law. *Human Factors*, 32(4):423–438, August 1990.
- [Sal01] D. Salvucci. Predicting the effects of in-car interface use on driver performance: an integrated model approach. *International Journal of Human-Computer Studies*, 55:85–107, 2001.
- [SBJ+97] P. Soliveri, R. G. Brown, M. Jahanshahi, T. Caraceni, and C. D. Marsden. Learning manual pursuit tracking skills in patients with parkinson's disease. *Brain*, 120(8):1325–1337, 1997.

- [SBM⁺95] K. Salisbury, D. Brock, T. Massie, N. Swarup, and C. Zilles. Haptic rendering: Programming touch interaction with virtual objects. In *SIGGRAPH'99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 401–408, 1995.
- [SECK00] N. Smyrnis, I. Evdokimidis, T.S. Constantinidis, and G. Kastrinakis. Speed-accuracy trade-off in the performance of pointing movements in different directions in two-dimensional space. *Exp Brain Res*, 134:21–31, September 2000.
- [SGS⁺05] Siriam Subramanian, Carl Gutwin, Miguel Nacenta Sanchez, Chris Power, and Jun Liu. Haptic and tactile feedback in directed movements. In *GOTH05: Proceedings of Guidelines On Tactile and Haptic Interactions*, pages 37–42, 2005.
- [Sha48] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 1948.
- [Sha98] Christopher D. Shaw. Pain and fatigue in desktop VR: Initial results. In *GI'98: Proceedings of the Graphics Interface*, pages 18–20. Canadian Human-Computer Communications Society, 1998.
- [She92] T. B. Sheridan. Musings on telepresence and virtual presence. *Presence: Teleoperators and Virtual Environments*, 1(1):120–125, 1992.
- [Smi96] Wanda J. Smith. *ISO and ANSI Ergonomic Standards for Computer Products: A Guide to Implementation and Compliance*. Prentice Hall, January 1996.
- [Ste57] S. S. Stevens. On the psychophysical law. *Psychological Review*, 64(3):153–181, 1957.
- [Ste08] Anthony Steed. A simple method for estimating the latency of interactive, real-time graphics simulations. In *VRST'08: Proceedings of the 2008 ACM symposium on Virtual Reality Software and Technology*, pages 123–129, 2008.
- [Sut65] Ivan E. Sutherland. The ultimate display. In *Proceedings of the Congress of the International Federation of Information Processing (IFIP)*, pages 506–508, 1965.
- [SZZ06] John J. Shaughnessy, Eugene B. Zechmeister, and Jeanne S. Zechmeister. *Research Methods in Psychology*. McGraw-Hill, 2006.
- [Tuc04] Allen B. Tucker. *Computer Science Handbook*. Chapman and Hall/CRC, second edition, 2004.
- [TWG⁺04] William B. Thompson, Peter Willemsen, Amy A. Gooch, Sarah H. Creem-Regehr, Jack M. Loomis, and Andrew C. Beall. Does the quality of the computer graphics matter when judging distances in visually immersive environments? *Presence: Teleoperators and Virtual Environments*, 13(5):560–571, 2004.
- [UI00] Brygg Ullmer and Hiroshi Ishii. Emerging frameworks for tangible user interfaces. *IBM Systems Journal*, 39(3&4):915–931, 2000.

- [VNCL05] Antonin Viau, Micheline Najm, C. Elaine Chapman, and Mindy F. Levin. Effect of tactile feedback on movement speed and precision during work-related tasks using a computer mouse. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 47(4):816–826, 2005.
- [vR06] Arjen van Rhijn. *Configurable Input Devices for 3D Interaction using Optical Tracking*. PhD thesis, Eindhoven University of Technology, 2006.
- [WB94] Colin Ware and Ravin Balakrishnan. Reaching for objects in VR displays: Lag and frame rate. *ACM Transactions on Computer-Human Interaction*, 1(4):331–356, December 1994.
- [WCTT07] Guy Wallis, Astros Chatziastros, James Tresilian, and Nebojsa Tomasevic. The role of visual and nonvisual feedback in a vehicle steering task. *Journal of Experimental Psychology: Human Perception and Performance*, 33(5):1127–1144, 2007.
- [Wel60] A. T. Welford. The measurement of sensory-motor performance: Survey and reappraisal of twelve years progress. *Ergonomics*, 3:189–230, 1960.
- [WL97] Colin Ware and Kathy Lowther. Selection using a one-eyed cursor in a fish tank VR environment. *ACM Transactions on Computer-Human Interaction*, 4(4):309–322, 1997.
- [WO90] Colin Ware and Steven Osborne. Exploration and virtual camera control in virtual three dimensional environments. *ACM SIGGRAPH Computer Graphics*, 24(2):175–183, 1990.
- [Woo99] R S Woodworth. The accuracy of voluntary movement. *Psychological Review*, 3:1–114, 1899.
- [WRMW95] John P. Wann, Simon Rushton, and Mark Mon-Williams. Natural problems for stereoscopic depth perception in virtual environments. *Vision Research*, 35(19):2731–2736, 1995.
- [WS98] Bob G. Witmer and Michael J. Singer. Measuring presence in virtual environments: A presence questionnaire. *Presence: Teleoperators and Virtual Environments*, 7(3):225–240, 1998.
- [WTN00] Steve Whittaker, Loren Terveen, and Bonnie A. Nardi. Let’s stop pushing the envelope and start addressing it: A reference task agenda for HCI. *Human-Computer Interaction*, 15(2):75–106, September 2000.
- [WWBH97] Aileen Worden, Neff Walker, Krishna Bharat, and Scott Hudson. Making computers easier for older adults to use: Area cursors and sticky icons. In *CHI’97: Proc. SIGCHI conference on Human factors in computing systems*, pages 266–271, 1997.
- [YIBB09] Xing-Dong Yang, Pourang Irani, Pierre Boulanger, and Walter F. Bischof. A model for steering with haptic-force guidance. In *Proceedings of INTERACT 1999: 12th IFIP TC 13 International Conference on Human-Computer Interaction: Part II*, pages 465–478, 2009.

-
- [ZAW04] Shumin Zhai, Johnny Accot, and Rogier Woltjer. Human action laws in electronic virtual worlds: An empirical study of path steering performance in VR. *Presence: Teleoperators and Virtual Environments*, 13(2):113–127, 2004.
- [ZBM94] S. Zhai, W. Buxton, and P. Milgram. The “silk cursor”: Investigating transparency for 3D target acquisition. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, pages 459–464, 1994.
- [Zha02] Haixia Zhao. Fitts’ law: Modeling movement time in HCI. <http://www.cs.umd.edu/class/fall2002/cmsc838s/tichi/fitts.html>, 2002.
- [Zha04] Shumin Zhai. Characterizing computer input with Fitts’ law parameters - The information and non-information aspects of pointing. *International Journal of Human-Computer Studies*, 61(6):791–809, December 2004.

Summary

A virtual environment is an interactive, head-referenced computer display that gives a user the illusion of presence in real or imaginary worlds. Two most significant differences between a virtual environment and a more traditional interactive 3D computer graphics system are the extent of the user's sense of presence and the level of user participation that can be obtained in the virtual environment. Over the years, advances in computer display hardware and software have substantially progressed the realism of computer-generated images, which dramatically enhanced user's sense of presence in virtual environments. Unfortunately, the same progress of user interaction with a virtual environment has not been observed.

The scope of the thesis lies in the study of human-computer interaction that occurs in a desktop virtual environment. The objective is to develop/verify 3D interaction models that can be used to quantitatively describe users' performance for 3D pointing, steering and object pursuit tasks and through the analysis of the interaction models and experimental results to gain a better understanding of users' movements in the virtual environment. The approach applied throughout the thesis is composed of three procedures, including identifying the variables involved in modeling a 3D interaction task, formulating and verifying the interaction model through user studies and statistical analysis, and applying the model to the evaluation of interaction techniques and input devices and gaining an insight into users' movements in the virtual environment.

In the study of 3D pointing tasks, a two-component model is used to break the tasks into a ballistic phase and a correction phase, and comparisons are made between the real-world and virtual-world tasks in each phase. The results indicate that temporal differences arise in both phases, but the difference is significantly greater in the correction phase. This finding inspires us to design a methodology with two-component model and Fitts' law, which decomposes a pointing task into the ballistic and correction phase and decreases the index of the difficulty of the task during the correction phase. The methodology allows for the development and evaluation of interaction techniques for 3D pointing tasks.

For 3D steering tasks, the steering law, which was proposed to model 2D steering tasks, is adapted to 3D tasks by introducing three additional variables, *i.e.*, path curvature, orientation and haptic feedback. The new models suggest that the 3D ball-and-tunnel steering movement should not be modeled as a continuous and smooth movement as assumed by the steering law, but a series of small and jerky sub-movements that are similar to the ballistic/correction movements observed in the pointing movements.

An interaction model is originally proposed and empirically verified for 3D object pursuit tasks, making use of Stevens' power law. The results indicate that the power law can be used to model all three common interaction tasks, which may serve as a general law for modeling interaction tasks, and also provides a way to quantitatively compare the tasks.

It has been demonstrated that the interaction models proposed in this thesis can be used

as general guidelines to quantitatively develop/compare/evaluate interaction techniques and input devices, which may substantially contribute to the improvement of current 3D user interfaces. The interaction models can also assist us in better understanding the interaction tasks, gaining an insight into user's movements and identifying the weaknesses and strengths of the 3D interaction that takes place in virtual environments.

Curriculum Vitae

Lei Liu was born on the 1st of October, 1981 in Jilin Province of China. He received his B.Sc. degree in Computer Science in 2004 from Jilin University of China, after which he began a double-Master-degree program in Artificial Intelligence between Jilin University of China and Vrije Universiteit (VU University), Amsterdam. His graduation project was performed under the supervision of Dr. Elena Marchiori and Prof. Dayou Liu, and involved dimensionality reduction on LC-MS dataset. He was awarded the Huygens Scholarship (HSP) and two M.Sc. degrees from the program in 2007. Immediately thereafter, he joined the Visualization and 3D Interfaces theme at the Centrum Wiskunde & Informatica (CWI), Amsterdam, in pursuit of a Ph.D. degree in Computer Science. There he performed the research described in this thesis under the supervision of prof.dr.ir. R. van Liere and as a member of the Quantitative Spatial Interaction Design (QUASID) project.