N° d'ordre : 4329

École doctorale Sciences Pour l'Ingénieur

Techniques d'interactions mixtes isotonique et élastique pour la sélection 2D et la navigation / manipulation 3D

THÈSE

présentée et soutenue publiquement le 19 Décember 2008

pour l'obtention du

Doctorat de l'Université des Sciences et Technologies de Lille
(spécialité instrumentations avancées)

par

PAN Qing

Composition du jury

Président :	Pr. Alain Derycke
Directeurs de thése :	Pr. Christophe Chaillou (LIFL – USTL) Dr. Géry Casiez(LIFL – USTL)
Rapporteurs :	Pr. Pascal Guitton Dr. Jean-Daniel Fekete
Examinateur :	Dr. Eric Lecolinet

Laboratoire d'Informatique Fondamentale de Lille — UMR CNRS 8022

© 2009 Tous droits réservés.

http://www.univ-lille1.fr/bustl

Thèse de Qing Pan, Lille 1, 2008

Mis en page avec la classe thloria.

© 2009 Tous droits réservés.

http://www.univ-lille1.fr/bustl

Acknowledgments

I want to express my grateful acknowledgments to the people who supported me to accomplish this thesis.

At first, I'm honored to express my deepest gratitude to my directeur de thèse, Prof. Christophe Chaillou for accepting me in the group ALCOVE to perform this thesis and giving me suggestions and criticisms. His kindness are greatly appreciated.

I'm also extremely grateful to my co-directeur de thèse, Dr. Géry Casiez for his guidance through this thesis, giving me valuable ideas, suggestions to and discussions about the problems that I encountered and correction of each document that I wrote.

I want to thank to my Chinese colleagues Dai Zheng, Wang Haibo, Xu Quan, Ding Li, Wei Yiyi in our group with them I shared much of my happy and hard time during my study. I also want to thank all the members of the group ALCOVE for their daily help.

I want to thank to Thomas Le Bris for his permanent support and encouragement to my work and his help to deal with each problem in my life. I want to thank my sister, Jie, for her patience to listen to me when I was in bad mood and cheering me up.

At last, I dedicate this thesis to my parents, they encouraged me to do my study aboard and finance my study and life during my master, without which, this thesis will be impossible.

© 2009 Tous droits réservés.

© 2009 Tous droits réservés.

Résumé

Le développement de interaction homme-machine aide les utilisateurs à travailler de manière plus efficace. Les technologies traditionnelles ne peuvent plus satisfaire les nouveaux besoins des applications variées. En environnement 2D WIMP, les périphériques d'entrée isotonique combinés avec un contrôle en position tels que les souris ou touchpad, souffrent de débrayages qui prennent du temps et rendent l'interaction moins lisse. C'est pire encore lors de l'utilisation d'un petit périphérique avec un grand écran. Les interactions 3D attirent l'attention de nombreux chercheurs. Toutefois, les techniques existantes qui permettent à la fois la navigation et de manipuler les objets ne sont pas naturelles ni suffisamment efficaces pour être totalement acceptées par les utilisateurs. De nouveaux périphériques d'entrée et des techniques d'interaction doivent être proposées afin d'améliorer la qualité de l'interaction.

Dans cette thèse, nous proposons deux périphériques d'entrée composé d'une zone isotonique avec un contrôle en position et une zone élastique avec un contrôle en vitesse; La première méthode *RubberEdge* est une méthode en 2D pour réduire le débrayage et la seconde méthode *Haptic Boundary* est une méthode 3D qui rend plus efficace les manipulations d'objets et l'exploration de l'environnement virtuel.

Pour *RubberEdge*, nous avons adopté la simulation de rotation d'un disque et un treatement mathématique dans sa fonction de transfert pour garantir un passage lisse du contrôle en position au contrôle en vitesse. Une évaluation d'une tache de sélection en 2D a été réalisée. Le résultat a montré que *RubberEdge* est 20% plus performant que le contrôle en position. Nous avons ensuite proposé deux modèles prédictifs pour le temps de sélection pour le contrôle en position ainsi que le contrôle hybride de *RubberEdge*. Nous avons présenté également une mise en œuvre de *RubberEdge* pour ordinateur portable.

Haptic Boundary permet des manipulations d'objets précises dans sa zone isotonique et la manipulation de la caméra sur sa paroi. Deux types de retour d'effort ont été adoptés pour fournir des mouvements de caméra plus riches et pour éviter un passage de mode explicite qui pourrait augmenter la charge mentale des utilisateurs. La taille et la forme de la zone isotonique ont été choisies avec soin afin de maximiser les avantages du contrôle en vitesse de deux types de retour d'effort. Une évaluation de la tache du montage d'une voiture virtuelle est réalisée. L'expérience a montré que *Haptic Boundary* est 50% plus performant que l'interface unimanuelle avec un changement du mode explicite. Après analyse des résultats, un mode d'*inspection en orbite* est mis en place pour améliorer l'usage de l'*Haptic Boundary*

Mots-clés: Haptique, Périphérique d'entrée, Interaction Human-Machine, Retour d'effort, Navigation, Metaphore, Evaluation.

© 2009 Tous droits réservés.

Abstract

The development of human-computer interaction technologies help people to work more efficiently. Meanwhile, traditional technologies could not fulfill the new-born requirements in diverse situations. In 2D WIMP environment the popular isotonic position control device, such as mouse, touchpad, suffers from clutching which is time-consuming and makes the interaction less smooth. It is getting worse when using a small input device to interact with a larger screen. 3D interactions attract attentions of many researchers. However, the existing techniques allowing both object and view manipulations are not natural or efficient enough to be totally accepted by users. New input devices and corresponding interaction techniques should be proposed to improve the interaction quality.

In this thesis, we propose two techniques based on isotonic-position and elastic-rate control spaces: 2D *RubberEdge* for reducing the clutching and 3D *Haptic Boundary* for efficient object manipulations and exploration in VE.

For *RubberEdge*, we adopted a simulation of the ration of a disc, and a mathematical treatment in its mapping function to guarantee a smooth switch between position and rate control. An evaluation of a 2D selecting task was performed. The result showed that *RubberEdge* outperforms position-only control by 20%. We then proposed two predictive models for selection time with position-only control and with the hybrid control of *RubberEdge*. We also presented the first *RubberEdge* prototype for laptop touchpad.

Haptic Boundary allows precise object manipulations inside its isotonic zone and camera manipulation on its boundary. Two kinds of force feedbacks were adopted to provide richer camera motions and to avoid the explicit mode switch which greatly increases user's mental load. The shape and size of isotonic zone is carefully chosen to maximize the benefit of the rate control with both two force feedbacks. An evaluation of car assembling task was performed. The result showed that *Haptic Boundary* outperformed the *unimanual interface with explicit switch* by 50%. After analyse the experiment result, the *orbiting inspection* is combined to enhance the applicability of *Haptic Boundary*.

Keywords: Haptic, Input Device, Human-Computer Interaction, Force Feedback, Navigation, Metaphors, Evaluation.

Title: Interaction techniques based on hybrid isotonic elastic feedback for 2D pointing tasks and 3D manipulation/navigation tasks

© 2009 Tous droits réservés.

Summary

© 2009 Tous droits réservés.

© 2009 Tous droits réservés.

Chapter 2
2D Isotonic Elastic Hybrid Device: *RubberEdge*

© 2009 Tous droits réservés.

Chapter 3

Manipulation and navigation device: *Haptic Boundary*

© 2009 Tous droits réservés.

http://www.univ-lille1.fr/bustl

© 2009 Tous droits réservés.

© 2009 Tous droits réservés.

List of Figures

© 2009 Tous droits réservés.

Thèse de Qing Pan, Lille 1, 2008

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

© 2009 Tous droits réservés.

© 2009 Tous droits réservés.

Thèse de Qing Pan, Lille 1, 2008

© 2009 Tous droits réservés.

http://www.univ-lille1.fr/bustl

List of Tables

© 2009 Tous droits réservés.

http://www.univ-lille1.fr/bustl

Thèse de Qing Pan, Lille 1, 2008

© 2009 Tous droits réservés.

http://www.univ-lille1.fr/bustl

Introduction

The works that we present in this thesis is a part of the project Alcove (action et collaboration sur des objets virtuels complexes) of INRIA and of LIFL(USTL-CNRS). This thesis is about new human-machine interaction techniques which benefit of hybrid isotonic-position and elastic-rate control input devices. We focus on the size, shape and force feedback of the device and its adopted transfer function which constitute our new techniques.

Nowadays, the 2D WIMP environment is still the mainstream of operating system. The isotonic position control input devices such as mouse, touch pad are popularly adopted in these operating systems. However with the diversification of application environments, the pure isotonic-position control devices became less and less satisfactory. For example, when such a device with small size interacts with a large display, the number of clutching will increase dramatically, which will affect the user performance.

3D virtual environment is still in the research stage, no mature operating system showed up, due to the complexity of 3D environment and lack of interaction techniques with ease, comfort, efficiency, simplicity and having compatible input device of rather low cost. A number of techniques are designed for one single task — one among object selecting, positioning or orienting and the view manipulation. Gathering them into a usable system requires extra effort to coordinate different modes and input devices used in the techniques for each task. All-in-one technique is possible but requires to be carefully designed.

New interaction techniques involving new input devices should be proposed to address these problems mentioned above. The idea of hybrid control which combines the isotonic-position and elastic-rate control is an interesting solution. Each control manner has its own advantages and disadvantages. Isotonic-position control is intuitive, easy to learn, but need the clutching mechanism to adjust user's hand position during the interaction, which makes the interaction less smooth, and bring certain fatigue. Elastic-rate control has the advantage of self-centering, without clutching suited for longer, smoother motions, however rate control requiring more learning time is less intuitive. Hybrid control combine them to maximize their advantages and minimize their disadvantages.

In this thesis, we propose two interaction techniques based on hybrid isotonic-position and elastic-rate control device to give new solutions to the problems mentioned above. We first studied the state of art in this and its related domain, and then described our two propositions — one for 2D target selection, another for 3D object and view manipulations. We also performed formal user evaluation for each technique.

In chapter one, we present the state of the art related to our work. We begin with the elemental interaction tasks including object selecting, positioning and orienting as well as

© 2009 Tous droits réservés.

Thèse de Qing Pan, Lille 1, 2008

the view manipulation task. Then, we present the input devices and transfer function, the relationship between them was also deeply studied. The interaction technique for both 2D WIMP and 3D virtual environment are reviewed. At last we focus on two groups of interfaces: bimanual interfaces and hybrid position-rate control interfaces. The former was interested by many researchers because it is close to the interaction in real life, but with existing techniques, it did not show an overwhelming advantage than unimanual interfaces, one of our proposed techniques is compared to an existing bimanual interface. Hybrid position-rate control interface is the base of our research work, we present several existing techniques including new input devices at the end of chapter one.

In chapter two, we describe our first proposed solution **RubberEdge** which is designed in order to reduce the clutching amount. RubberEdge is based on the 3D **Bubble** technique. RubberEdge overcome the discontinuity of cursor speed during the switch from the position control tro rate control which is an observable problem when applying the *Bubble* technique in the 2D environment. After analyses of the result of 2D pointing task, we give two performance models — clutching model and hybrid control model to predict the completion time to select a target of a given size in a given distance through the pure position control technique and RubberEdge technique. We also propose prototype for the laptop to replace the touch pad, including the calibration process for applying RubberEdge. An initial evaluation was also performed for this prototype.

In chapter three, we describe our second proposed technique **Haptic Boundary** designed for both object manipulations and exploration of virtual environment (view manipulation). It outperforms the traditional unimanual interface with explicit switching between different modes which increases the users' mental load. We investigate the workspace required for *Haptic Boundary*, analyses its size, shape and haptic force feedback properties to maximize the functionality, intuitive and efficiency. Based on the basic implementations, a few of improvement are proposed to increase the usability of *Haptic Boundary*. After a formal user evaluation of the car assembling task, we combine an extra feature, orbiting inspection, with *Haptic Boundary* to improve the object observing functionality.

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

Chapter 1

State of the Art

Contents

© 2009 Tous droits réservés.

Thèse de Qing Pan, Lille 1, 2008

1.1 Résumé

Dans ce chapitre, nous étudions l’état de l’art concernant notre travail. Nous partons des taches élémentaires de l’interaction — la sélection, le positionnement, l’orientation de l’objet et la manipulation de la vue, ainsi que leurs paramètres qui déterminent le degré de difficulté et qui favorisent certaines techniques plus que d’autres.

Nous étudions ensuite les périphériques d’entrée, y compris certaines termes importants — relatif et absolu, dicret et continu, l’intégralité et la séparabilité des degrés de liberté. Nous présentons ensuite trois résistances différentes pour le périphérique: isotonique, isométrique et élastique. Les retours d’haptiques passifs et actifs sont expliqués. Après cela, nous présentons les fonctions de transfert de base ainsi que la relation entre la résistance et la fonction de transfert. Pour le contrôle en position avec un périphérique isotonique, nous expliquons le *CD ratio* et le mécanisme de débrayage.

Pour avoir une bonne connaissance des techniques d’interaction, nous recueillons des techniques ciblées pour les taches élémentaires pour les environnements en 2D et en 3D. Nous donnons davantage de détails sur les techniques 3D. Nous présentons rapidement les techniques avec les périphériques d’entrée en 2D, et ensuite nous nous concentrons sur des techniques plus avancées. Dans un but de clarté, nous présentons la notion de technique d’interaction en taches croisées, suivie par les techniques d’éléments d’interaction commun, dont la désignation d’objet, l’indication de position, le contrôle de la direction du mouvement, la spécification de la vitesse de mouvement et la contrainte imposée sur des mouvements contrôlée. Basées sur ces composantes, nous présentons les techniques de sélection, de rotation et de positionnement des objets virtuels. Pour les manipulations de vue, nous présentons les métaphores communes, des techniques basées sur le zoom ou sur le widget, et puis nous parlons de plusieurs facteurs importants de la manipulation de la vue. Ces facteurs incluent le cadre de référence, la contrainte imposée sur le mouvement de point de vue, le retour et le changement de mode lorsque plusieurs techniques sont disponibles.

Ensuite, nous présentons les interactions bi-manuelles. Nous commençons avec la théorie de base, incluant le modèle de chaîne cinématique de Guiard, suivi par des recherches expérimentales, ainsi que les avantages des interfaces bi-manuelles. Nous présentons par la suite les techniques existantes dans plusieurs catégories: symétrique en 2D, asymétrique en 2D, symétrique en 3D et asymétrique en 3D, puis les interfaces

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

bi-manuelles pour les systèmes de bases sur table, ainsi que les périphériques d'entrée utilisés dans les interfaces bi-manuelles.

Enfin, nous présentons les interfaces hybrides combinant le contrôle en vitesse et le contrôle en position, incluant la technique *Bubble* ainsi que plusieurs périphériques avec des capteurs isotoniques élastiques qui peuvent être combinés avec un contrôle en position et en vitesse, tels que *GlobeFish*, *GlobeMouse*, *Two-4-six*, *YoYo* et *SquareBone*.

1.2 Introduction

In this chapter, we will review state of the art relative to our work. Firstly, we present the interaction tasks in the human-computer interaction in both 2D and 3D Environment. Then we talk about the important factors of interaction techniques. One factor is input devices we will discuss its basic properties, resistance and haptic feedback. Another factor is transfer function we examine its relation to the device resistance. Then we will examine the existing techniques for object and view manipulation in both 2D classic and 3D VE. We will also study the bimanual interfaces and hybrid position-rate control techniques.

1.3 Interaction Tasks

From its development history, the human computer interface (HCI) went through three major stages. In the early *MS-Dos* stage, the user communicates with computer through the predefined commands: the user types a command and presses "return", then the computer executes it. This primitive interface is not friendly enough to users due to the considerable command vocabulary to be memorized.

The first 2D Graphic User Interface (GUI) *Lisa* was developed by Steve Jobs, one founder of Apple Computer in 1983. Shortly after that, Apple published the second 2D GUI system *Macintosh*. In may 1990, Microsoft published Window 3.0, a more standardized GUI system. Up to today, the Windows series is still the most popular 2D GUI system occupying a more than 90% market share. The windows-based systems adopted the *desktop* metaphor, which is characterized as *Windows*, *Icons*, *Menu* and *Pointing* (*WIMP*). Here, *windows*, *icons*, *menus* are the computer-generated object for interaction and *pointing* is the interaction paradigm.

While the WIMP system is widely used in people's everyday lives and works, the 3D interaction became the focus of the entertainment industry and researches, such as video games, computer-aid design (CAD), scientific visualizations, simulator and medical surgery training and teleoperating. The goal of interaction technique is to create a VE similar to the real one, to make the interaction more intuitive to users and to perform the task that human can not in the real world.

In both 2D WIMP environments and 3D VEs, the principal interaction tasks can be abstracted as manipulation of the computer-generated objects and the manipulation of the viewpoint (virtual camera) which changes the manner user observes the computer generated environment. In this section we talk about the elemental tasks of object manipulation as well as their parameters. For view manipulations, we introduce the DOFs

© 2009 Tous droits réservés.

Thèse de Qing Pan, Lille 1, 2008

of the virtual camera and the different categories of travel tasks in VE.

1.3.1 Elemental Tasks of Object Manipulation

Based on the general assumption that the requisites of human efforts in all cases are composed of the same elemental tasks, Foley[FWC84] proposed the five basic direct object manipulation tasks for 2D GUI. These elemental tasks include: *selecting*, *positioning*, *orienting*, *text* and *qualifying*. *Selecting* means identifying an object[1], e.g., selecting an icon; *Positioning* means moving an object from an initial position to a desired final location, e.g., dragging an icon to a desire place; *Orienting* means changing the orientation of an object from an initial orientation to a desired final orientation; *Text* is the input of a string of characters; *Qualifying* is the input of a numerical value.

The object manipulation tasks in 3D VE are the same tasks performed in the real world. Based on Foley's taxonomy, most researchers support the three elemental VE interaction tasks: *selecting*, *positioning* and *orienting*. They have the same meaning as in 2D environment. Some researchers use manipulation to refer to *positioning* and *orienting*, *selecting* is listed separately. In most case, the object should be selected before the positioning and orienting is performed.

Below we talk about the parameters of each elemental task. These parameters define the complexity of a task, affect the user performance and can favor certain techniques than others.

1.3.2 Parameters of Basic Manipulation Tasks

Each elemental task has several parameters which should be considered during the design of interaction techniques. Poupyrev [PWBI97] summarized the principal parameters of the three elemental tasks from several literatures. We list them below.

Parameters of Selecting Task

The parameters of *selecting* include *the number of objects to be selected*, *distance to the target*, *size of the target*, *direction to the target*, *occlusion of target*, *density of objects around the target* and *dynamics of target*. Among them, The *number of objects to be selected* can be 1 in the simplest case. Task of selecting several objects is often referred to as a *browsing task* [Pla95]. According to Fitts' law, the ratio of the *distance to the target* and the *size of the target* determines the time required to select the target. The *direction to the target* determines which body parts and muscles are activated [McC70].

Parameters of Positioning Task

A *positioning* task can be decomposed into three distinct phrases: selecting of object, primary gross travel, and a corrective motion to place the object at the desired final location with the required accuracy. Its parameters include: *distance to the initial location of the manipulated object*, *direction to the initial location of the manipulated object*,

[1] Referenced also as a target acquisition task in [ZBM94]

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

Thèse de Qing Pan, Lille 1, 2008

distance to the final terminal location, *direction to the final terminal location*. Studies of positional movement have shown that the time and accuracy of movement depends on these four parameters above [McC70]. Other parameters include *visibility of terminal location*, *dynamics of terminal*, *occlusion of terminal*, *size of the manipulated object* as well as *required precision of positioning* which affects the corrective motion phrase of *positioning*.

Parameters of Orienting Task

Similar to the positioning task, the orienting task can be decomposed into three phrases: object selecting, gross orienting and the adjustment to achieve the desired orientation with a required precision . Its parameters include: *distance to the manipulated object*, *direction to the manipulated object*, *initial orientation*, *final orientation* and *required precision of orientation*.

1.3.3 View Manipulation

We discussed the object manipulation elemental tasks and their parameters, in this section we focus on the view manipulation tasks. We discuss the ones in 2D WIMP environment and in 3D VE.

In 2D WIMP environment, the user needs to move the view to different location and adjust the zoom scale to see a target at different detail level. Scrolling and zooming are two indispensable tasks in browsing large documents, such as webpages, map, image, and the resources manager in windows system which contains lots of file icons.

In 3D VE, the view manipulation can be performed in two different manners. The first one considers the view as a virtual camera and manipulating its parameters or DOF which will be discussed below. The second one regards the camera as an object which is moved around the VE. This process can be abstracted as navigation. Navigation is discussed in section 1.3.3.

DOF of Virtual Camera

The virtual camera has similar parameters or DOF to the real camera. These DOFs include dolly, truck, crane, pan, tilt which are defined in table 1.1 and marked out in figure 1.1.

Navigation in 3D VE

Instead manipulating each DOF, navigation emphasize on the behavior of camera as an object. According to Bowman [BKH98], navigations can be subdivided into the motor component *travel* and the cognitive component *wayfinding*. *Travel* refers to the motion of viewpoint motion. *Wayfinding* refers to the cognitive process to determine a path based on visual cues, knowledge of environment and aids.

© 2009 Tous droits réservés.

http://www.univ-lille1.fr/bustl

Cinematic Term	Definition
dolly	Translation of the viewpoint along the lateral axis
truck	Translation of the viewpoint along the view normal
crane	Translation of the viewpoint along the up vector
pan	Rotation of the camera about the vertical axis
tilt	Rotation of the camera about the lateral axis
roll	Rotation of the camera about the view normal
zoom	Changing the field of view of the camera

Table 1.1: Common DOF of camera.

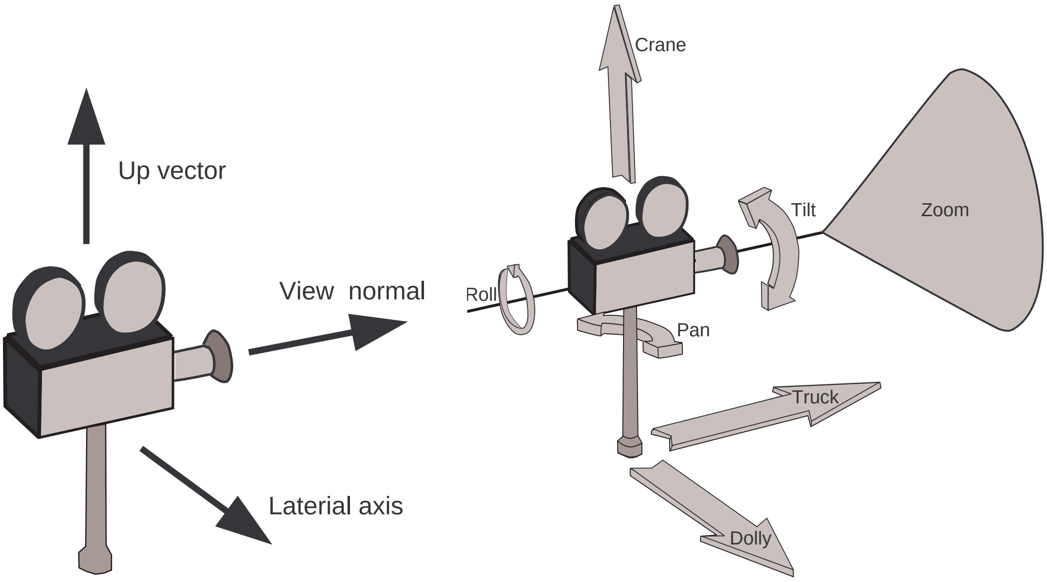


Figure 1.1: Left: Camera axis up vector, view normal, lateral axis; Right: Virtual camera's seven degrees of freedom: three for position: dolly, truck, crane; three for orientation: pan, tilt, roll; the last DOF is field of view(FOV) zoom.

Mckinlay et al.[MCR90b] classified the *travel* tasks into five categories. *General movement*[2] refers to travel without explicit target. *Targeted movement*[3] refers to moving viewpoint to a particular target location. *Specified coordinate movement*[4] refers to adjusting the viewpoint to a precise position and orientation. *Specified trajectory movement* refers to moving the viewpoint along a trajectory with predefined position and orientation, such as a cinematographic camera movement. Tan pointed out an additional category *inspection* which establishes and maintains a particular view of an object.

According to Bowman [Bow99] and Tan [TRC01], the parameters of travel include: *direction*, *velocity/acceleration* and *path or trajectory*. They are manipulated by different

[2] Referred as *exploration* by Bowman et al. [BKLP99]

[3] Referred as *searching* by Bowman et al.[BKLP99]

[4] Referred as "maneuvering" and considered characterized by short range and high precision by Bowman et al.[BKLP99].

© 2009 Tous droits réservés.

travel techniques.

1.4 Input Devices

The interaction technique are designed to facilitate the interaction tasks. There are several important factors to the interaction techniques. Input device is one of them. To obtain a high user performance, the interaction technique, interaction task and the input device should be compatible. Input device is any piece of computer hardware equipment it sends commands, position, motion or other informations to computer. And then the feedback informations are sent back through output devices. In this section, we discuss the input device from several aspects. We first talk about the several properties of input device, followed by the three kinds of resistance of input device — isotonic, isometric and elastic, which in fact serves as a feedback during interaction. A detailed haptic feedback is also discussed, involving active and passive haptic interfaces.

1.4.1 Properties of Input Device

Each input device has certain properties determined by its fabrications including the embedded sensor and its structure design, etc. These properties can make the input device suitable to certain tasks, and less to others. Below, we give the definitions of several common properties including discrete and continuous, absolute and relative as well as intergral and separable.

Discrete and *continuous* [Bux83] are use to describe the input data stream of devices. A *discrete* input has a distinct *start* and *stop*, such as a button. It is suitable for command-line interface and event-based architecture 2D WIMP application. A *continuous* input is a running stream of data, such as the mouse and position tracker.

Absolute [HJW04] input device refers to the device which reports its actual position rather than its relative movement, e.g. the touchscreen is absolute which senses the absolute position of finger on the screen. *Relative* [HJW04] input device refers to the one which reports the distance and direction of movement each time it is moved, but does not report its absolute position. Relative devices require a *clutching* mechanism to engage and disengage the link between control actions and movement of controlled graphics object. For example, the mouse is relative, lifting it from the mouse pad will stop the cursor.

The *integrality* and *separability* of DOFs were defined by Jacob [JS*94]. The device with *Integral* DOF allow users to move diagonally across DOF, e.g. user controls integrally the x and y DOF of mouse. The devices with *Separable* DOF allow users to move along only one dimension at a given moment, e.g. using two keys to control the cursor movement. The figure 1.2 shows two trajectories, one for separable DOF (left) which is called *stair-step* movement, one for integral DOF (right) which is called *euclidean* movement by Jacob [JS*94].

© 2009 Tous droits réservés.

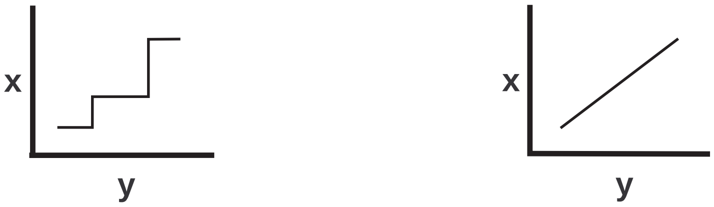


Figure 1.2: Left: Trajectory showing separable degree of freedom; Right: Trajectory showing the integral degree of freedom.

1.4.2 Device Resistance

Resistance of input devices is the constraint felt by user during the usage, it is determined by the structure and material for fabrication of input device, and provide the passive-haptic feedback. We talk about the devices of three kinds of resistances: *isotonic*, *isometric* and *elastic*.

Isotonic Devices

In Webster's Ninth New Collegiate Dictionary, the word *isotonic* means "of, relating to or being muscular contraction in the absence of significant resistance, with marked shortening of muscle fibers, and without great increase in muscle tone, compared with *isometric*. Isotonic devices have zero or constant resistances and it senses the position of the end effector. They are also called displacement devices, free moving devices or unload devices. The computer mouse, touch pad, 6 DOF tracker belong to this category. The figure 1.3 showed several isotonic devices.

Isometric Devices

In Webster's Ninth New Collegiate Dictionary, the word *isometric* means "of, relating to or being muscular contraction against resistance, without significant shortening the muscle fibers and with marked increase in muscle tone". Consisting with this definition, an isometric device senses the force applied by and does not perceptibly move. Its resistance can be infinite. Isometric devices are also called pressure devices or force devices. Some joysticks belongs to isometric devices category. The figure 1.4 showed two isometric devices.

Elastic Devices

The resistance of the elastic devices is between the infinite resistance of isometric devices and the zero or constant resistance of isotonic devices. Its resistance varies with the displacement of end effector. The *spacemouse* is an elastic devices. The figure 1.5 showed some other input devices.

The resistance of input device serves as some haptic feedback to user. Below, we discuss the haptic feedback issue in more details.

© 2009 Tous droits réservés.

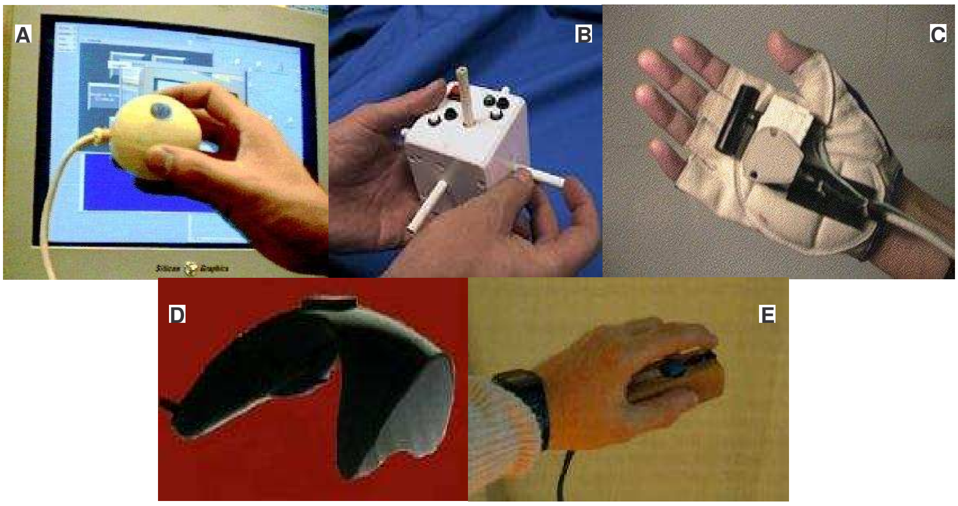


Figure 1.3: Isotonic devices. (A): the *Fingerball* designed by Zhai [Zha95]. (B): *Cubic Mouse* designed by B. Froehlich and J.Plate [FP00]. (C): *MITS Glove* designed by Zhai [Zha95]. (D): *Cricket*™, commercialized by Digital Image Design Inc. — a free moving device consisting of a tracker inside of the handle. (E): *Bat* designed by C. Ware [War90], consisting of a Polhemus™tracker and a handle.

1.4.3 Haptic Feedback of Input Devices

The word "*haptic*" comes from the Greek, it means "touch". Haptic sensing is very important for human being to interact with the world. The parts of and the parts around the hand play a major role for receiving the feedback information. The studies of neuron physiologist have shown that various parts of the human body have different proportion of brain cortex allocated to their control, the finger tips and hands take a relative large proportion than body parts. The receptors and nerve endings in the skin, joints, tendons and muscles inside the hand structure can catch the external mechanical thermal and chemical stimuli with different rapidity and transit electrical impulses via the neural network to the central nervous system in the brain which sends back commands to the muscles to reaction.

The feedback informations caught by the hand during contact with an object can be mainly divided into two kinds: the *tactile* information and the *kinesthetic* (equivalently, *proprioceptive*) information.

- *Tactile* refers to the information about the contact surface (texture, roughness), temperature. The fine properties such as shapes, textures and rubber-like compliances of the object within the contact region and whether the object is slipping relative to the skin are sensed by tactile sensor in the skin.

- *Kinesthetic* refers to the information about the position, motion of limbs along with the associated forces. The rough properties such as large shapes or spring-like compliance that require hand and arm motion are sensed by the kinesthetic sensors.

The net force can be sensed by both two kinds of human sensors.

© 2009 Tous droits réservés.

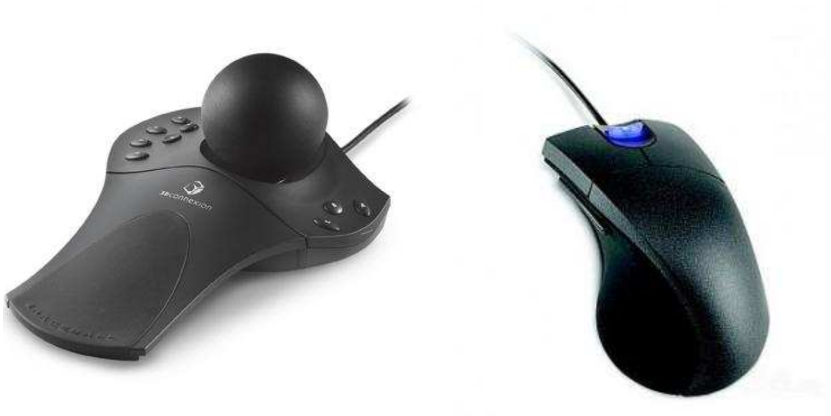


Figure 1.4: Isometric devices. Left: The Spaceball™, manufactured by Labtec®. Right: *ScrollPoint*® mouse of IBM combined with 2 degree of freedom isometric joystick.

The haptic interface groups tactile and the kinesthesis feedback to get as much information as possible. When the user interacts with VEs through an haptic interface, the tactual informations are returned to the user by appropriately stimulating users' tactile and kinesthesis sensory. According to the employed devices, the haptic interfaces can be divided into active-haptic and passive-haptic interfaces.

Active-haptic Interface

The active-haptic interfaces use the active-haptic devices that produce the output stimuli fed calculated according to the sensed position of end effector and the information of simulated environment to the user through the electric, electronic, or mechanical hardware. They are the input and the output device simultaneously. A number of active-haptic devices were developed, such as *Wingman Force Feedback* of Logitech, the *PHANToM* of sensable co., the *Virtuose6D*, the *DELTA Haptic Device*, the *DigiHaptic*.

PHANToM, a series of most successful haptic input devices, have an armature with a stylus at its distal end, providing the designer with a programmable force feedback via OpenHaptic SDK. The user can feel virtual objects through the stylus. During the usage, the *PHANToM* can sense the position of the tip of the stylus, and generate the calculated force which resist user's motion, according to the shape and stiffness of virtual objects. In our research work, the programmable force feedback feature is used to simulate the physical workspace of different shape and size.

DigiHaptic [CP*03, Cas04] is a new 3D separable multi-finger force feedback device.It consists of three levers actuated by motors and controlled by thumb, ring finger and index finger respectively. Each lever can provide active-haptic force feedback generated by computer. DigiHaptic is shown in figure 1.7.

© 2009 Tous droits réservés.

Thèse de Qing Pan, Lille 1, 2008

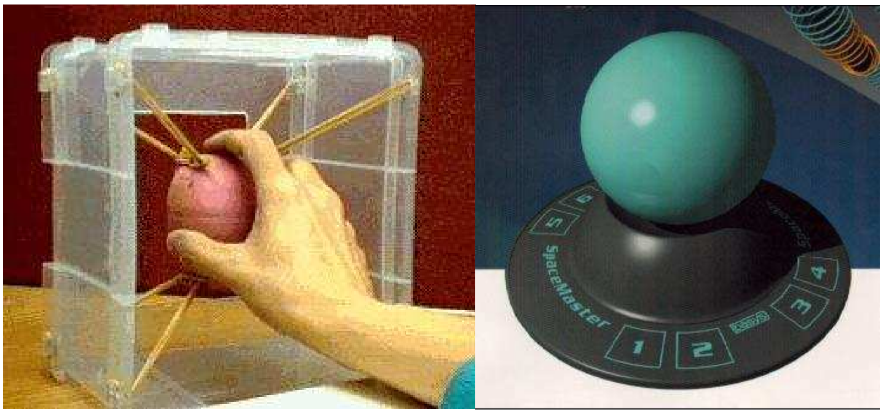


Figure 1.5: Two elastic devices. Left: EGG designed by S.Zhai[Zha95] consists of an Ascension Bird®mounted within an elastic frame. Right: SpaceMaster™patented by DLR, the German aerospace research establishment, manufactured by Space Control Company, Malching, Germany and marketed by Logitech, Gremont, CA, USA.

Haptic Display

Haptic display is used to describe the update of the active haptic feedback in real-time. It has the same implementation principle as the visual display. To achieve a realistic interaction, the frequency of visual display should be above than 60 Hz, the one of haptic display should be above 1000 Hz.

Passive-haptic Interface

Another kind of haptic interface is passive. IT provideS feedbacks through the shape, texture, elasticity, resistance or other inherent properties of the parts of or the whole input device. This feedback is not controlled by the computer, instead, it comes from the device itself. In fact, every input devices provides more or less passive haptic feedbacks. Hinckley et al. used a passive-haptic "*props*" as input devices in neuronsurgical visualization application, a doll representing the volumetric data of patient's brain in non-dominant hand, and a cutting-plane or a stylus in the dominant hand to interact intuitively.[RGK94]. The props are shown in figure 1.8. The commercial product Magellan SpaceMouse (elastic) and SpaceBall (isometric) shown in figure 1.8. Their resistance provide an useful self-centering passive haptic feedback for controlling the movement speed.

1.5 Transfer Functions

We have discussed the interaction tasks and input devices in the previous sections. Now, we begin to talk about the interaction techniques for both 2D and 3D environments. This section talks about the transfer function, which transform the input of device into a property of an object in VE. Transfer function is another key factor of interaction techniques. Position control and rate control are two transfer functions most commonly

© 2009 Tous droits réservés.

Thèse de Qing Pan, Lille 1, 2008

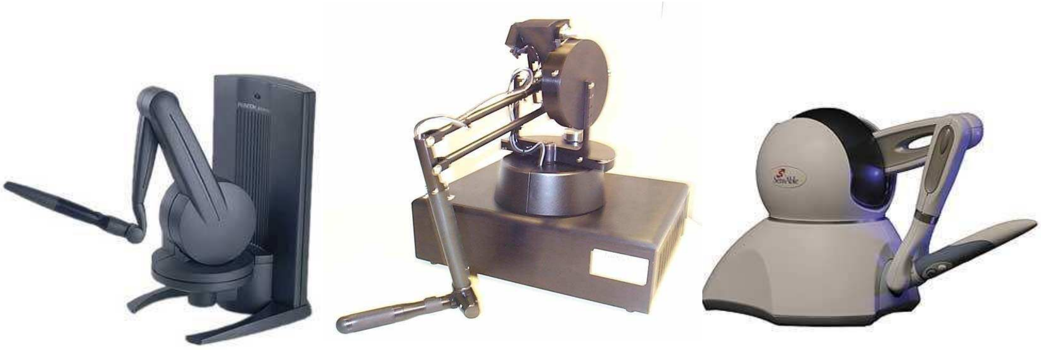


Figure 1.6: Three products of the *PHANToM* series made by sensable co.

used. After the description of them, we talk about the relationship between the transfer function and device resistance. And then we discuss in details the isotonic position control involving its two factors: *CD ratio* and *clutching*.

1.5.1 Basic Transfer Functions

Position Control

Position Control refers to the control mechanisms by which the human operator directly controls the object position / displacement. It is also called *zero order* transfer function. In the simplest case, position control is a linear relationship between input and the object position / displacement with a gain constant. The non-linear transfer functions were also used.

Rate Control

Rate Control transforms the sensor input into the velocity of the controlled object.It is also called *first order* transfer function, because the transformation consists of an integral operation.

The performance difference between position and rate control is not obvious. The works have been done in comparing these two transfer function in 1 and 2 DOF tasks. Most of them concluded that rate control is inferior to position control, because the position control is more direct, requires little mental transformation in generating control actions. However, with position control, the operating range is limited unless clutching is adopted, whilst rate control has an effectively unlimited operating range. Another advantage of rate control is the resulting movement is smoother comparing to position control.

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

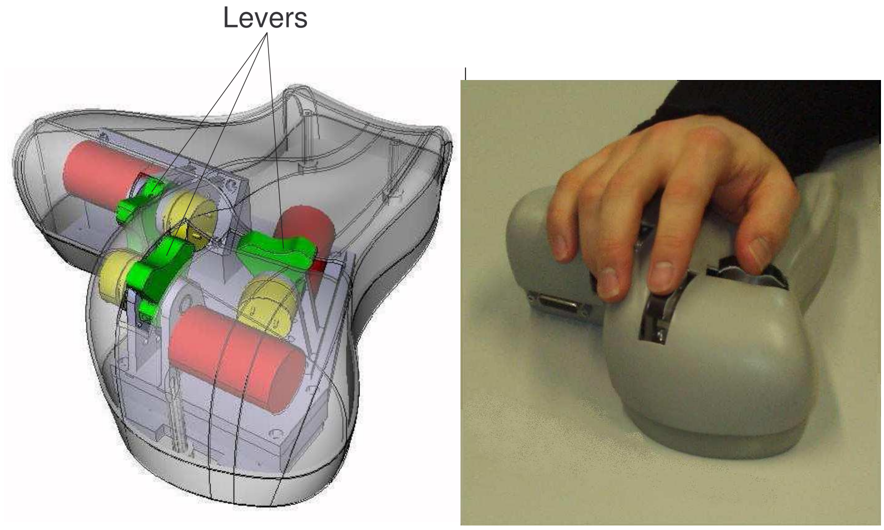


Figure 1.7: Left: The DigiHaptic with its three levers actuated by motors; Right: The way the user puts his hand on it.

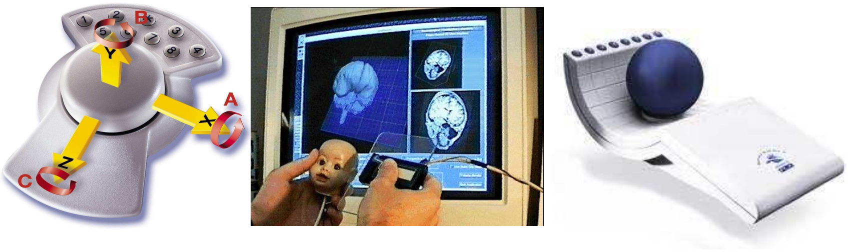


Figure 1.8: Two passive-haptic input dvices: *SpaceMouse* (left) with 6 DOF elastic sensor. *Spaceball* (right) with 6 DOF isometric sensor, and the props in Hinckley's medical visualization application,

High Order Transfer Function

High Order Transfer function refers to the transfer function with higher order than one, for example, mapping the input onto this acceleration of virtual object.Some research works have demonstrated that both position and rate control are superior to higher order control during interaction [Wic92, Pou74], because the latter lack of intuitiveness, thus are rarely used in human computer interactions.
Zhai summarized the position, rate, acceleration control transfer functions in figure 1.9.

1.5.2 Position/Rate control and Resistance of Input Devices

The position control and rate control transfer function can be combined with input device of different resistance. Some combinations outperform other combinations. Zhai's thesis[Zha95] provided useful informations about this subject. Zhai pointed out that

© 2009 Tous droits réservés.

Thèse de Qing Pan, Lille 1, 2008

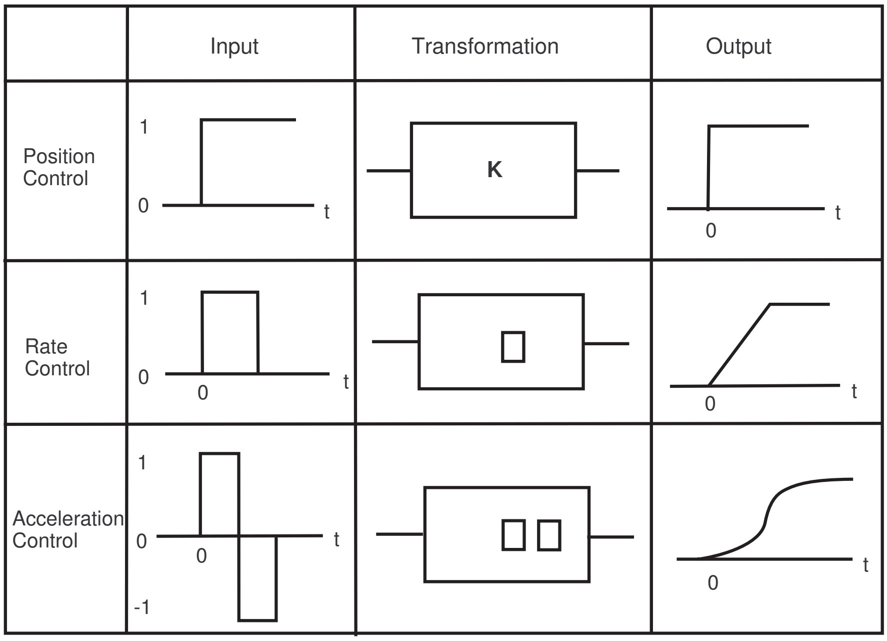


Figure 1.9: Idealised control inputs (left column) for obtaining step changes in output level (right column) for position, rate and acceleration control (Zhai [Zha95])

the controller transfer function should be compatible with the characteristics of physical devices. He performed a 6 DOF docking experimentation where 4 combinations of the transfer function (position control, rate control) and the resistance (isotonic, isometric) were compared. The result showed that the isotonic-position control outperformed the isometric-position control and the isometric-rate control outperformed the isotonic-rate control because of the strong self-centering effect of isometric devices which is helpful to understand the rate control interaction pattern. He also showed that in 6 DOF docking task and 6DOF tracking task, the elastic-rate control is easier to learn than isometric-rate control due to the richer kinesthetic/proprioception feedback provided by elastic devices. Elastic device performs as well as isometric-rate control because elastic has also the advantageous self-centering feature.

Zhai et al. proposed a taxonomy of 6DOF input [ZM94] summarizing certain important properties of input devices to facilitate the input device design, shown in figure 1.10.

1.5.3 Isotonic Position Control

In this section, we present more details about the isotonic position control. Control-Display ratio is the key factor of isotonic position control. Another factor, clutching mechanism is generally adopted to break through the restriction of operating range. These two factors will be discussed below.

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

Thèse de Qing Pan, Lille 1, 2008

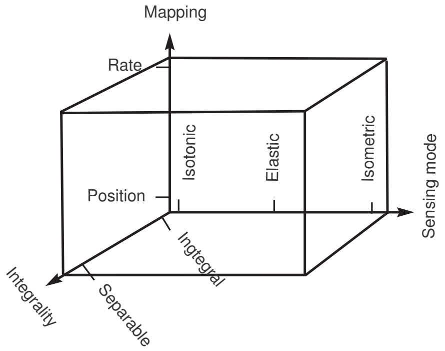


Figure 1.10: Input Taxonomy(Zhai[ZM94])

Control-Display Ratio

Control-Display ratio (*CD ratio*) is the ratio of the controller (input device) motion to the graphical object motion. For example, if a mouse moves 1cm on the mouse pad (the control) to move the cursor 2 cm on the screen (the display), the *CD ratio* is $\frac{1}{2}$. *CD gain* is multiplicative inverse of *CD ratio*.

The *CD ratio* can be constant, e.g. the touch screen using a constant *CD ratio* of 1, the cursor is always under the user's finger. Jellinek et al. [JC90], Accot et al. [AZ01] and Zhai [Zha95] found that performance degraded at low and high *CD ratio* levels. This may be partially due to the limits of human motor control [BM97] (e.g. the magnified hand tremor), the limited device resolution preventing selection of one pixel with a high *CD ratio*, and increased clutching with a low *CD ratio* [Zha95, MO98]. Unfortunately, the effect of *CD ratio* is not conclusive [JC90, AZ01, Joh94, LCF76].

The *CD ratio* can be changed dynamically, *Pointer Acceleration* is such a technique. *Pointer Acceleration* is used in Windows XP operating system [5] (shown in figure 1.11). If the user activates "improve the precision of mouse" option in mouse's properties panel, the *CD ratio* will vary with the velocity of mouse: when the mouse moves quickly, the cursor will move even more quickly; while when the mouse move slowly, the cursor will move even more slowly . The goal of this technique is to provide users with both the rapidity and the precision during the interaction.

Clutch Mechanism

Clutching is a mechanism to engage and disengage the link between input and controlled item. For the mouse, the clutching is performed naturally: when reaching the limit of movement, the user lifts the mouse, replaces it at a more comfortable location, usually closer to the user himself, and continues the interaction. Clutching allows the input device with limited operating system to perform a motion without limit.

The amount of clutching is dependent on the input area of input device, target distance and the transfer function. A small input area causes more clutching with large target distances, A higher *CD ratio* thus reduces the clutching amount but with a potential

[5] "*Pointer Ballistics*" [Poi] in Windows operating system is a *Pointer Acceleration*.

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

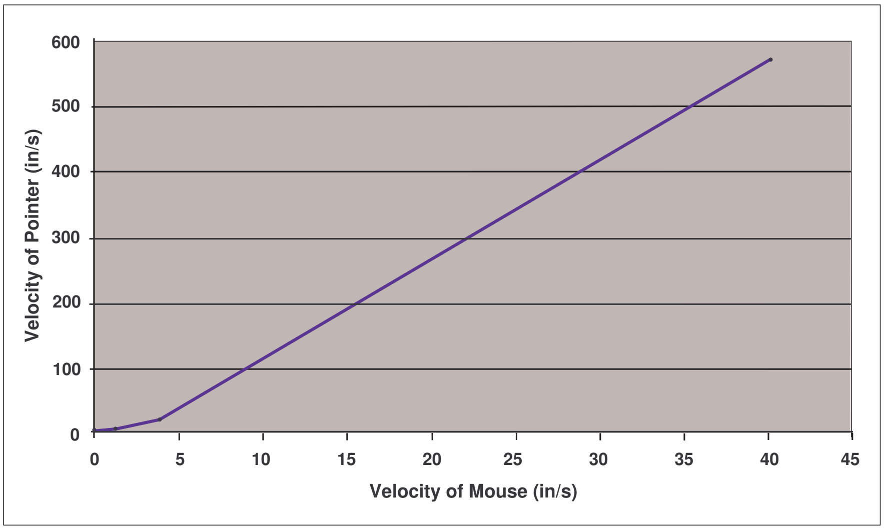


Figure 1.11: *Pointer Acceleration* (pointer ballistics) transfer function in Windows XP.

performance cost.

Clutching is necessary for both translation and rotation with rate control. A 3D rotation with a tracker-based input device needs a clutching mechanism to perform a large angle rotation, due to the kinesthetic constraints of the wrist and the structure of input device,

Clutching can make the interaction seem unusually reducing the feeling of engagement [Han97]. The clutching interface should be well-designed. A poor design can jeopardize the usefulness of the spatial input. For example, an "ill-placed" or "hard-to-press" clutch button will make user fatigued in as little as five minutes of use. If a clutching button is mounted at a fixed location on 3D input device, the user must have a fixed grip on the input device to keep his finger in a position to press the button. Due to the constraints of the wrist, a fixed grip limited the possible rotation. This makes a large rotation awkward. Using voice for clutching does not seem to work very well. A separated clutch button from the input device can be considered, such as foot pedal used in [RGK94].

We have discussed the two factors of interaction techniques — input devices and transfer function. In following sections we will review some of existing techniques.

1.6 Interaction Techniques in 2D Environments

In this section, we review the interaction techniques in 2D environment, including the selection techniques for facilitating the small target selection or for reducing the selection time; the techniques for rotating and positioning items in the tabletop system, and the view manipulations techniques based on scrolling and zooming.

© 2009 Tous droits réservés.

1.6.1 Selection in 2D WIMP

The selecting action in 2D WIMP applications can be described as *point-and-click*. A number of techniques are proposed to facilitate the 2D selecting. To ease the selecting small targets, the techniques based on zooming were used. In the traditional "*ZoomPointing*", the user firstly zooms in the area around the target with a dragging gesture, and then selects the target in the zoomed-in view. However, his zooming action forces the user to leave from the target. "*Dual Finger stretch*" proposed by Bneko [BWB06] is a bimanual multi-fingers technique designed for tabletop interaction, where the primary finger sets the initial anchor location around which the visual area is scaled, the secondary finger controls the zooming scale by moving closer or further away from the primary finger.

Pointing lenses [RC*07] temporarily enlarges both the visual and motor area under cursor. Guiard proposed *Semantic pointing* [BGB04] which dynamically adjusts *CD ratio* as the cursor approaches the target, with the level of adaptation depending on the importance of the target. Albinsson [Alb03] proposed several techniques to finely adjust the position of cursor. *Cross-Key* used the intersection of two "rubber-bind" lines to select the target where the adjustment is done by dragging the handles at the both end of the two rubber-bind lines. *Precision handle* consists of a handle, a pivot point, and a tip with a cross-hair to tip to point the target, the adjustment is performed by dragging the handle. For these two techniques, the selection is confirmed by clicking the activation circle over the target. However, the author's formal experimental study showed that their techniques did not outperform the traditional *ZoomPointing*, which may be due to the additional steps required for the fine adjustments.

Grossman proposed "*bubble cursor*"[GB05] to help the 2D target selection by dynamically change the size of the cursor area, depending on the proximity of surrounding target, so that only one target is encompassed in the cursor area at any time. However with the increase of the density of targets, the performance of *bubble cursor* degrades. Additionally, both *bubble cursor* and *Semantic pointing* require prior aware about the size, locations and the importance of the targets.

1.6.2 2D Manipulation Techniques

In the 2D WIMP environment, the major manipulation task is positioning an icon which can be performed by dragging and dropping.More complex 2D interactions are required in the Graphic editing task, such as positioning, rotating and scaling. "*GEdit*" system [KB91] used *gestures* of 2D mouse to invoke the manipulation commands. For example, the user draws a circle around one or a group of 2D shapes (objects) and points out the destination. These gestures can invoke a positioning command which moves the circled objects to the destination. "*Hand-to-rotate*" is used in *Microsoft power point*, paint software to perform rotations.

In the tabletop system, such as *Virtual WorkBench*, *Responsive Workbench* (usually for the multiuser collaborative interaction), a number of techniques are proposed for icon rotation.With *corner to rotate* the user selects an icon, touches its corner and turns the icon about its center. *Relocate and reorient*[RS*04] uses the finger's position as input, allows translating and rotating simultaneously, similar to *corner to rotate*. A special task

© 2009 Tous droits réservés.

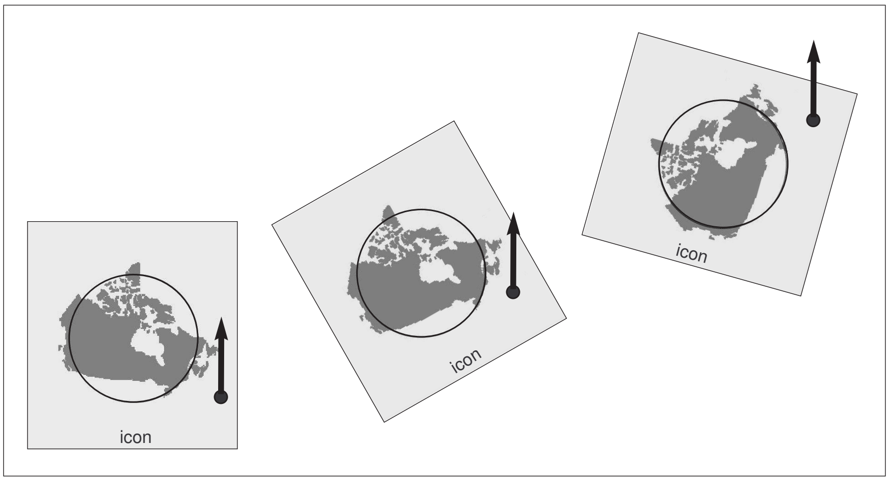


Figure 1.12: RNT. A control point is selected in lower right, and the icon rotates counterclockwise while it is translated upward [KCT05]

in tabletop system environment is "passing a document or image to someone", where the document icon should be translated and rotated so that the icon reaches at and orients toward the receiver. *R'NT* [KCT05] (figure 1.12) is a technique for such a task, when the user clicks and drags outside of the central "circular translation only region", the icon begins to translate and rotate according to the user's click position on the icon, as if there is some fluid opposite to the icon movement direction. "*Drag*" technique [Mit03] is similar to *R'NT* by using the virtual friction to produce changes in orientation.

The *TNT* [LP*06] techniques (figure 1.13) rotate the document icon by turning the open hand with a tracked "finger sleeve" worn on the index finger, which is called "*TNT hand*". "*TNT-block*" uses a tracked cylindrical "block" to perform large rotation by turning the block.

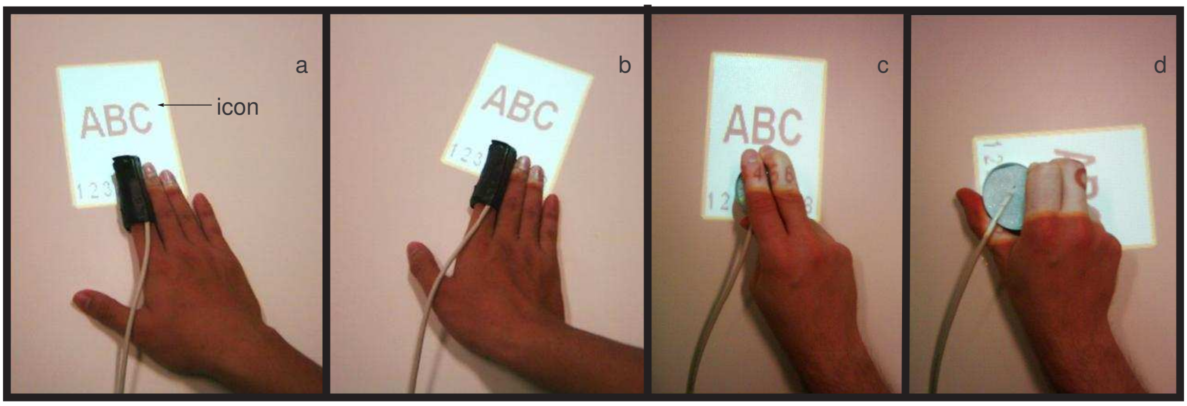


Figure 1.13: TNT techniques. a,b: TNT-hand, the orientation of hand controls the one of the virtual icon, which is similar to the rotation in real life. c,d: TNT-Block. c: sensor positioned on artifact. d: Large rotation angle by twisting block. [LP*06]

© 2009 Tous droits réservés.

Thèse de Qing Pan, Lille 1, 2008

Several techniques automatically rotate the icon on the tabletop space for the goal of "give the right person a right view.", such as the *InfoTable* [RS99] when a user drags an icon to his own side of the table, the icon automatically rotates toward the closest table edge; In *STARS* [MSP03] the icon is automatically rotated toward the active player; *ConnecTable* [TP*01] automatically rotates the passed object to the right side of the receiver when touched by the receiver. Similar technique appears in *DiamandSpin* [RS*04], "*InteracTable*"[SG*99]. These techniques requires exploring the locating of the person manipulating the icon, which is not always appropriate.

1.6.3 View Manipulation Technique in 2D Environment

The fundamental view operations in WIMP applications is *scrolling* and *zooming*. *Scrolling* is used to displace the viewport to see the information beyond the current window. *Zooming* is used to change detail scale of view.

Traditionally, scrolling is performed by dragging the virtual scrolling bar on the screen with a pointing input device like a mouse or a pen. The small modifications of mouse were proposed to improve the scrolling performance. The mouse IBM *Scrollpoint* is with an isometric joystick atop for rate control scrolling. Most mice are equipped with a isotonic scrolling wheel at the same position to allow position control scroll. Additionally, Hinckley et al. proposed a *wheel acceleration algorithm* [HC*02b], based on the observation that when reading or moving short distances, users move the wheel slowly in a controlled, line-by-line manner; when covering longer distances, users rapidly flick the wheel to get to the destination quickly. They produced a graph of distribution of the wheel speed (shown in figure 1.14) and found two picks on this graph, one for rapid wheel motion, one for slower "reading". The acceleration algorithm adopted a continuous exponential transformation to rapid wheel movement, to make the scrolling even more rapid. Their formal experiments result showed this accelerated wheel scrolling outperformed the joystick and wheel scrolling. "*Navipoint*" [KI98] uses an elastic "trackpoint" to perform the scrolling for browsing the information with small mobile input device such as a PDA.

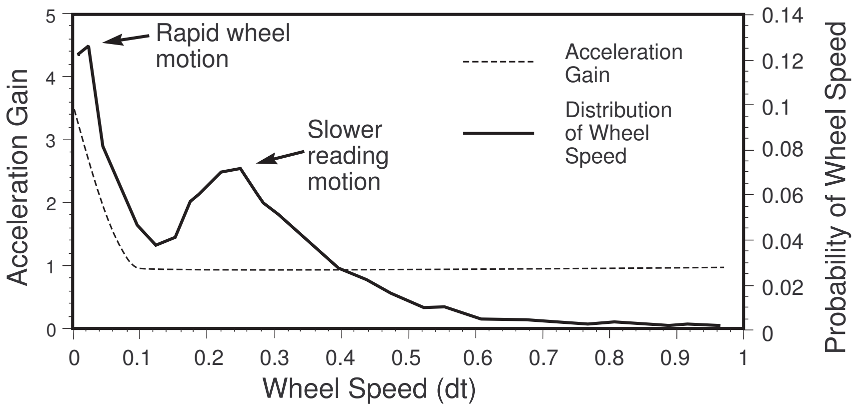


Figure 1.14:

Several techniques are proposed to improve zooming performance. Igarashi[IH00] pro-

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

posed the *Speed-dependent automatic zooming* to give user a global view of documents as the scrolling rate increased. This technique allows users to precisely transit between overview and the local view of specific target and to inspect particular target.

Zooming operation is often combined in view manipulation techniques for special purpose, for example,to emphasize the interested area of a large document by distorting the view using different zooming scale. "*Fishview*"[Fur86] enlarge the interested area of large document. "*Document lens*"[RM93] zooms in the part of document inside the rectangle lens controlled by user and remains the rest of the document zoomed out to give the user a global context. The "*perspective wall*" [MRC91] is similar to "*document lens*". However, the view distortion may put additional cognitive load on the user to mentally re-map the spatial relations. Several techniques discussed in 1.6.1 for facilitating the selection are also based on zooming.

We have reviewed the interaction techniques in 2D environment. In next section, we review the interaction techniques in 3D VE.

1.7 Interaction Technique in 3D VEs

The interaction techniques in 3D environment can be realized with both 2D and 3D input devices. We classified the techniques in the two categories.

1.7.1 Perform 3D Interaction with 2D Input Device

The techniques for selection, positioning, orienting and traveling techniques in 3D with 2D input are reviewed below.

3D Position / Translation using 2D Input Device

The mouse-ray based positioning techniques are commonly used in 3D modeling systems. It involves two key elements, the first one is the ray emitted from the eye point through the pixel currently under mouse cursor to find an intersection with the scene. The second one is the constraint during positioning or translation. Usually, the manipulated object is snapped to and move along the surface intersected [Bie86], which can be the boundary of the scene, or other object surfaces. However the manipulated object appears to jump when it is snapped to different surfaces (shown in figure 1.15 a,b). The movements become unpredictable. To address this problem, Oh proposed a method using the *foremost hidden surface* behind manipulated objects[OS05] (the closest surface behind the manipulated object) combining with the collision detection to prevent the manipulated object from penetrating into the objects in front of it (shown in figure 1.15 c-f). However this technique does not allow placing the object on the invisible surfaces, the user has to move the viewpoint to see the desired position, and then move the object. Another inconvenience is that if the manipulated object occupies a large part of the screen, the moving action is difficult to perform, the user should move the camera further away.

© 2009 Tous droits réservés.

Thèse de Qing Pan, Lille 1, 2008

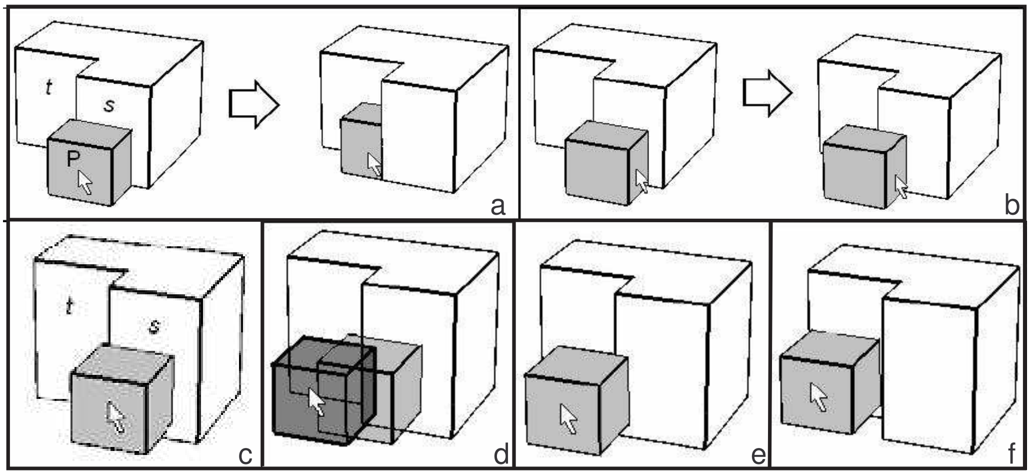


Figure 1.15: a,b: Techniques that use the surface directly interacted with the mouse ray suffer from non-predictability, as the same mouse-motion may generate different results, because the different location of cursor. c-f: Object slides on the *foremost hidden surface* [OS05]. The cursor translates the object to left, for the moment of c, d, e, the closest surface is s, for f, the closest surfact is t. With this technique, manipulated object behavior can be predicted.

3D Rotation using 2D Input Device

A number of techniques for 3D rotation with 2D input were proposed. The first *Virtual Sphere* technique is proposed by Chen[CMS88], where a 3D sphere located behind the viewpoint, displayed as a circle around the object to be rotated, shown in figure 1.16 left. Up-down and left-right cursor movements inside the circle produce the rotations about the x and y axis. Cursor movements along (or completely outside) the edge of the circle produce rotations about z. The amount of rotation is adjusted so that a full sweep of cursor across the circle rotates the object 180 degrees; a full circle around the outside rotates the object 360 degrees. This technique is used in *Quickdraw3D*. However it has some drawbacks. The first one is the "*hysteresis effect*", more concretely, after several manipulations, the user returns the cursor to the beginning point, but the orientation of the object may not return to its initial orientation. The second one is that the *Virtual Sphere* depends on the incremental mouse movement and can behave badly if sampling is slow or coordinates are noisy. The third one is the distinction of control modes between the inside and outside of the circle may disturb users.

The *Arcball* (figure 1.16 right) proposed by Shoemake[Sho92] improved the Chen's technique in the problems motioned above. The user rotates the object by drawing an arc on the screen projection of the *virtual sphere*. The arc is a great arc computed from the initial and current position during the dragging, due to which, *Arcball* avoids the problem caused by discontinuous input. The direction and the amount of rotation are those of half length arc model which results in a fixed *CD ratio* and is free of *hysteresis effect*. With "half length arcs". a sweep across *Arcball*'s inner circle rotates the virtual object 360

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

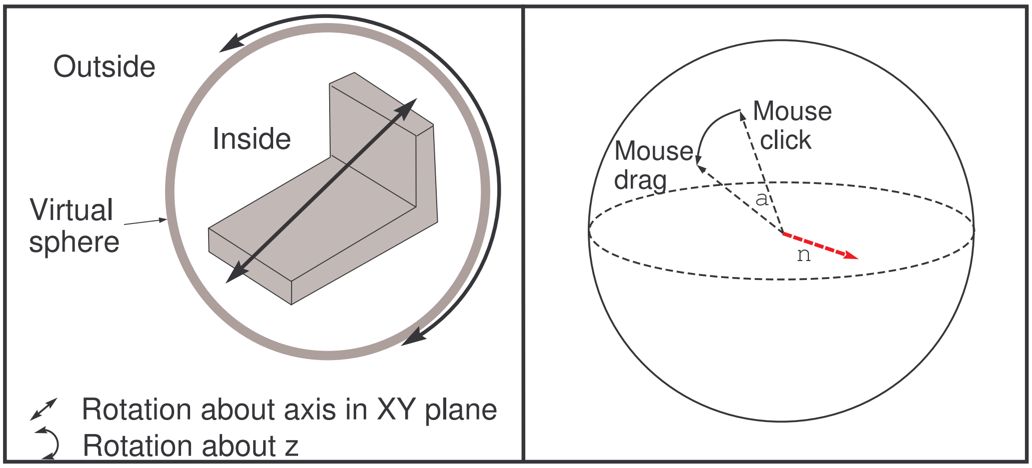


Figure 1.16: Left: Chen's *Virtual Sphere* and Right: Shoemake's *Arcball*

degrees (180 degrees in *Virtual Sphere* of Chen.). And with the *Arcball* the user does not have to change their behaviors between inside and outside of the circle representing the *Virtual Sphere*. Additionally, *Arcball* has some constraint mechanisms to achieve the constrained rotation such as rotation about the principal axis of objects. *Arcball* has also virtual feedbacks to help user to understand the technique. However, Henriksen[HSH04] pointed out that the *Arcball* is inhomogeneous and discontinuous with the consequences for usability.

Henrison[HSH04] also re-analyzed the *Virtual Sphere* of Bell, and pointed out that the Bell's *virtual sphere* and *Arcball* have different function mapping points in 2D screen plane to 3D points: the *Arcball* maps the points onto a sphere, while Bell maps them onto a surface composed of a sphere and a hyperbola. Compared with *Arcball*, Bell's *Virtual Sphere* produces smoother rotation because of the orientation of the rotation axis is continuous which is a function of the screen coordinate.

However, all the *Virtual Sphere* techniques separated the rotation into 2D + 1D control. However, some psychologist's study showed that the mental model of rotation is an integrated manipulation which is opposite to the 2D + 1D control of *Virtual Sphere* techniques [HTP97]. Another obvious problem of these techniques is "too slow", Chen measured 17.5 seconds to make complex rotations with virtual sphere[CMS88], and Hinckley measured for 28 seconds for both *Virtual sphere* and *Arcball*[HTP97]. While in the real life, a rotation will take around 1 second [WM*98, WR99]. This may due to the fact that the translation DOF are used to control the rotation DOF. Hinckley's experiment shows that the 6DOF tracker is 36% faster than *Virtual Sphere* techniques.

View Manipulation using 2D Input Device

For view manipulations, the 2D input device has the advantage of precision than the devices with higher DOF. "*WebSpace*" (later named "Cosmo player"), the first VRML browser[MM*96] maps the mouse movement onto different translation and orientation DOF according to different mode selected by clicking on appropriate menu bar. The

© 2009 Tous droits réservés.

available modes include "walk"(z-y translation in rate control), "pan"(x-y translation in rate control), "turn" (x-y rotation in rate control),"roll" (z-rotation in rate control), study (x-y rotation of the world in position control). Its drawback is obvious — mode switching increase the user's mental load. Mouse-based 3D navigations commonly used in video games such as Doom3, Halflife. "*Speed-coupled flying with orbiting*"[TRC01] uses the mouse movement control the inspection of *point of interest* (*POI*). Mouse, joystick, walking around can be used as input of orbital viewing.

We have talked about the techniques using 2D input devices. Below, we talk about the techniques with higher DOF input. These techniques for different tasks have the common components. The cross-task conception describe this in more detail.

1.7.2 Cross-task Interaction Technique

The major tasks in 3D VE — travel, selection, manipulation have some common points between them. For example, travel can be considered as specification of position and orientation of the viewpoint, and the manipulations do the same thing to the objects. This means that a technique designed for one task can be a solution technique to another task. The manipulation techniques can be used for travel tasks by making user represented as an avatar in VE, and setting the position and orientation of the avatar; The selection techniques can be used for manipulation, for example, place the object attached to the cursor, next to the selected object. Another example is Pierce's "*image-plane*" technique can be used for selection, manipulation and travel. In manipulation mode, the selected object is moved relative to the scene while in travel mode, user's viewpoint is moved relative to the scene or to the selected object. These techniques are called *cross-tasks* interaction techniques, because they cross the boundaries between the tasks [Bow99].

The advantage of cross-tasks techniques is that the same metaphor may be used for multiple tasks, which will increase the consistency and reduce the complexity of the interface.

1.7.3 Components of Interaction Techniques

Inspired the cross-task techniques of Bowman motioned above, and summarized the opinion of Mine [Min95], Bowman[Bow99] and others, we consider several common components in interaction technique: *indication of object*, *indication of position*, *motion direction control*, *specification of velocity of motion*, *Constraint* is also discussed. At end of this section, the *virtual control* represented by widget-based techniques are reviewed.

Indication of Object

Indication of object is an essential step of object selection, also can be a step of object positioning and travel tasks. Bowman[Bow99] summarized the technique for object indication in his taxonomy of selection technique.

The first group of techniques is "*touching-based*" indication, where the object is indicated by touching it, shown in figure 1.17 left. The major design factors include input

© 2009 Tous droits réservés.

device, mapping between the position, orientation of the real hand and the ones of the virtual hand (representing the cursor). The basic technique is "*Classic Virtual hand*" which maps the direct mapppingIndication of object is done by placing the cursor in the selection zone of an object, e.g. its bounding box. The drawback of the "*Classic Virtual hand*" is the reach virtual hand is limited by the physical reach of user's arm. The *scaling factor* (equivalent to *CD ratio*), is difficult to be determined, because most applications either do not know the furthest distance to reach or require the user to work with the object in both short and long distance. The object at distant may be difficult to touch due to the magnified tremor movement of user's hand.

To address the problem of limited reach, Poupyrev proposed non-linear mapping *touching* techniques "*Go-Go*" [PWI96]. The virtual hand moves normally when close to the user, and extends rapidly when it reaches into the VE. Its mapping function is shown in figure 1.17 right. *Go-Go* requires a user to place his arm at a certain position during the selection, which may bring fatigues to user.

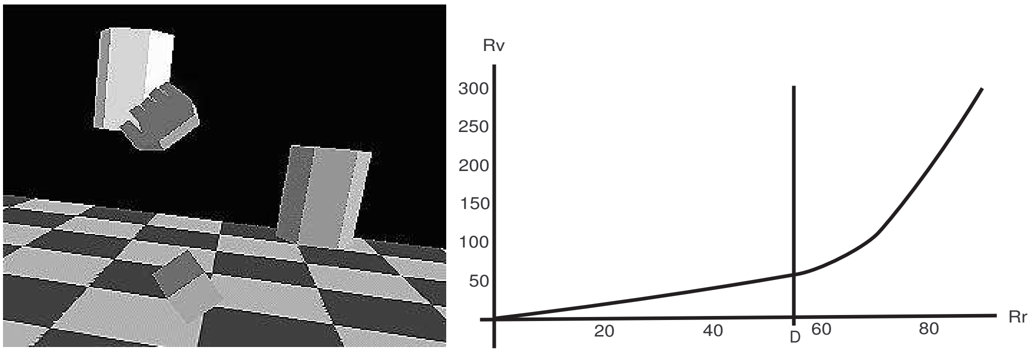


Figure 1.17: Left: *Touching*-based object indication. Right: Mapping function for the *Go-Go* technique, *Rr* represents physical hand distance, *Rv* represents virtual hand distance, *D* represents the boundary separating the closer range from the rest of physical space.[PWI96]

Touching-based techniques may require the data glove or 3D/6D trackers as input devices in the implementations.

The second group of techniques for object indication is *ray-casting*[6], where a ray is used to indicate an object. Its major design factors involves the definition of origin and direction of the ray, shape of the cursor and the disambiguating method. The ray's origin can be the position of user's hand, user's eyes (gazed-based indication in immersive VE), the virtual hand, cursor, viewpoint even at level of hip (immersive VE), etc. The direction can be pointed by user's hand, the cursor, or the gaze direction, etc. *Ray-casting* indication is shown in figure 1.18 left.

Ray-casting can be implemented with a 2D input devices such as a mouse (mouse-ray). Like in [BK99] the indication of target was done by moving the 2D cursor in the screen plane such that the cursor was over the target (in the line of sight) and clicking the left

[6] Noted also as pointing

© 2009 Tous droits réservés.

mouse button. This 2D controlled *ray casting* is as same as the selection in *WIMP*, so takes less time to learn it. In Bowman's taxonomy it is called *2D pointing*.

The advantage of *ray-casting* is non fatiguing. The user's hand does not need raise too high.With *Ray-casting* the user controls the direction of the ray via 2DOF, which is easier than the *touching* techniques requiring 3 DOF controlled. *Ray-casting* is specially effective for the larger, closer objects, but it's difficult to indicate the smaller object at distance which requires certain precisions.

To address the problem for small distant objects, the *Aperture*-based[7] techniques were proposed [FHZ96], which define a cone with its apex at the eye point and a circle (aperture) floating parallel to the screen plane, shown in figure 1.18 right. By moving the cone along the screen plane, the direction of ray is changed; By moving the cone along the depth direction, the cone's width is adjusted. The objects whose projection fall within the conic volume are indicated.

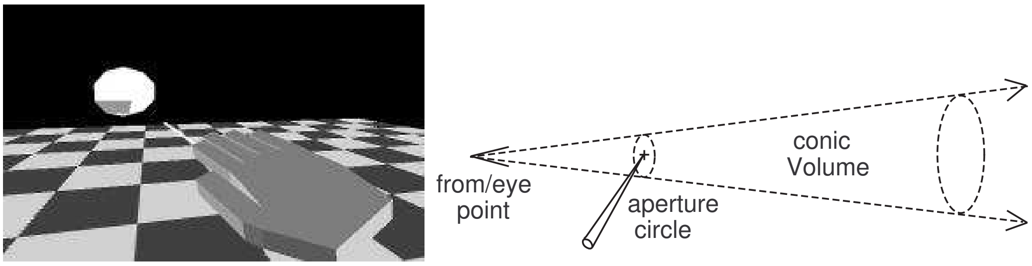


Figure 1.18: Left: *Ray-casting*-based object indication. Right: The conic volume of aperture technique is described by the eye point and aperture geometric

The third group is *Image-Plane*-based technique[8]. *Image-plane*-based techniques proposed by Pierce[PF*97] are similar to *ray-casting*. It works on the object's projection on the 2D image plane (screen or view plane). Pierce proposed four *image-plane*-based techniques: *Sticky Finger* where the user indicate objects by touching its projection. *Head Crusher* where the user indicates object by placing his forefinger and thumb so that the projection of object is between them. *Framing hand* where the user indicates objects by making a frame around its projection. The frame is formed by the forefinger and thumb of both hands, or through the motion of input device such as a mouse dragging. *Lifting Palm* where the user indicates objects by placing the hand or cursor under its projection. Like *ray-casting*, these *image-plane*-based techniques required 2DOF controlled by user. Some studies [BJH99, WT*05] showed that the *image-plane*-based techniques have a similar performance time to *ray-casting*. But they are more fatiguing, because the user can not completely put down his hand during interaction.

Additionally, "*Scaled-world*" technique scales the VE down so that the desired object is inside of user's reach. *WIM* [SCP95] is such a technique, the user can indicate any object by touching their "avatar" in the *WIM*. However, the small objects can be hard to access. Usually the *scaled world* is visualized in a separated view port, which brings some

[7] Aperture-based technique is called also spotlight-based technique in the literatures.

[8] In Bowman's taxonomy *Image-Plane*-based techniques are noted as "Occlusion/framing".

© 2009 Tous droits réservés.

Thèse de Qing Pan, Lille 1, 2008

occlusions or reducing the available display for the normal visualization of VE, and forces the user to switch his attention between these two views.

Indication of Position

Indication position is a requisite part of any positioning technique of both view and object. According to Bowman[Bow99] the position indication can be implemented as discrete target specification and as the continuous specification. Among the discrete target specification, all the techniques for the object indication discussed above can be adopted. Among the continuous specification, the position is continuously controlled by user's hand, by the position of end effector of input device with lower or higher DOF, the gaze, by physical input device such as wheel steering, and even by virtual control.

Motion Direction Control

Motion direction should be specified during object translating or view traveling. It can be controlled via *ray-casting*, or *image-plane*-based techniques with hand, gaze as input, in both continuous specification manner and discrete target specification manner. The "*crosshairs*" technique used the hand as input where the motion direction is determined by the vector from the user's head (immersive VE) or from the virtual eye position, through the *crosshairs* cursor. *Gaze directed* uses the user's view direction as the motion direction, the user moves himself or an object towards the direction he is currently looking. *Gaze direct* motion is easy to understand, but it eliminates the possibility of looking around during the motion.

Other techniques adopting physical input devices such as joystick, trackball, button, slider are used to *steer* the motion direction.

The direction of motion can be influenced by *driven objects* in VE. These objects include *autonomous vehicle*, *attractor* and *repellor*. An *autonomous* is an virtual object which can automatically move another object or the user to a new location in VE, such as an elevator. An *attractor* draws the controlled objects towards itself, such as the planet with a simulated gravity. A *repellor* pushes the controlled object away from itself, extends radically from the center of the repellor. *Attractor* and *repellor* can be *autonomous* or under control. However they require an additional interaction to control their position.

The motion direction can be *Goal Driven*: Select the destination from an pre-defined list or icons or an virtual map , the motion is achieved automatically with or without an animation. *World-scaled* technique in *WIM* can also be used to indicate the motion direction.

Specification of Velocity of Motion

A specified motion direction and a specified velocity or acceleration form a motion. The velocity can be specified by system controlled *constant velocity*, *constant acceleration*, or *hand control*, *physical control*, and *virtual control* by users.

With *Constant Velocity*, the motion is alway at the same rate. This velocity is usually adjusted so that user can traverse the whole space in a reasonable time. However the constant motion velocity could lead to jercky, sudden start/stop and overshoot.

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

Thèse de Qing Pan, Lille 1, 2008

With *Constant Acceleration*, the motion begins at a low velocity and grows exponentially. It can be used in very large VE. However, the acceleration rate is difficult to determine.

With *Hand control*, the motion velocity is determined by how far user's hand is extended from user or how far the end effector of hand hold input device leaves its zero position. The mapping from the input to the motion velocity can be linear or exponential. However, the fatigue of having to hold the hand at the required distance may degrade user performances. And the range in which different control mode is applied is limited.

With *Physical control*, Control the velocity through the physical input devices, such as keyboard, voice, acceleration pedal, slot-car controller, treadmill, dial or slider on the hand-hold input devices.

Constraint

The *constraint* commonly used in 3D interaction especially in modeling systems, such as *JDCAD*[LG93], *Skitters and Jacks*[Bie86], "*ARCADE*"[SM97]. The principal constraints include:

"*Snap to*" where the cursor can be snapped to the different type of elements: vertices, edges, curve, surfaces, object, according to the selected mode. *Grid snapping* snaps the cursor or manipulated elements to the grid in *JDCAD*[LG93], *ARCADE*[SM97].

"*Constrain for manipulation*" constrains the axis for translation, rotation, scaling and aligning the modeling elements to the principal axes, edges, surfaces. The SKETCH[ZHH96] constrained the placement of object to the position in contact with the existing objects or geometries. Teather [TS07] proposed a design guide that when the object is far from the view plane, its movement is constrained on the background surface.

"*Physical-based constraints*" such as the "*collision detection*" to avoid two solid objects from interpenetrating[OS05] or to avoid object from floating in VE.

Constraints can be provided via the virtual indications such as the modeling elements edges, vertexes, surfaces or the widget. Constraint can also be provided is via the physical props. In *TabletPC* [SG97a], a *personal interaction panel* is used as a 2D interface that controls the 3D environment. Chen [CNP05] has the similar idea. This kind of techniques usually require the non-dominant hand to hold the secondary physical props.

If the force feedback input device is used, the haptic constraints can be provided, for example, the collision detection mentioned above, can be hapticly displayed. Another example is the "*Virtual Fixture*" in [SJP96] which employs the force feedback to guide user in carrying out manual and supervisory control task. Komerska combined the force feedback with his 3D interaction widget [KW03]: during the rotation, the cursor is constrained to move along a circle ring around the widget; during the uniform scaling, a *restoring force* towards the scaling center is sent back to users; during the translation, a small amount of inertia can be felt by user. *Snap to* is also in haptic, where the tip of *PHANToM* (user's input device) is attracted by the 3D grid surface *battymetric surface*, the bounding edge, 2D planar surface, the ring menu and widget.

Constraints reduce the number of DOF necessary for interaction. In some situation, it reduces the complexity of the interaction, allow more precise control using a multi-DOF input devices. Some constraints technique requires object-specific constraints to be de-

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

signed a priori, lack of generality.

We have talked the cross-task techniques, component of interaction techniques, based on them the existing interaction for object selecting, object manipulation and view manipulation are talked about in following sections.

1.7.4 Selecting Techniques

For object selection, the basic *touching*-based technique is intuitive, and the shift into the manipulation following the selection is more natural than *ray casting*. However, the *touching*-based techniques require users to match 3 DOF of the cursor with the target, which causes fatigues. The object at distance may be difficult to select due to the magnified tremor and the fatigue, which may make the selection signal (pressing a button) difficult to achieve, because this fine movement may drag out the cursor from the object's selection zone. The *ray casting* selection requires only 2 DOF to be matched but has some difficulties for task requiring precision.

Some techniques provide aid for 3D object selection. For *touching*-based selection, *Silk* cursor [ZBM94] a volumetric semi-transparent cursor was proposed by Zhai to enhance the depth-cue for user. It is shown in figure 1.19. A similar selection manner in "ARCADE"[SM97] uses a transparent *pick-sphere* cursor. The *snap to* technique mentioned in section 1.7.3 are good aids for selection. Other similar techniques were proposed to facilitate the basic selection technique: Wingrave proposed *snap to angle* technique [WBR02] to enhance *ray-casting* and *image-plane*-based selection technique. When the objects are within the *snap-to angle* scope of user's hand, for *ray-casting*, a *snap-to ray* is emitted from the origin of the original selection ray, pointing to the closest one among all objects within the *snap-to angle*, shown in figure 1.20 left. For *image-plane*[9], a "*bull's-eye*" is attached to the hand representing the point through which the ray emitted from the eye passes and extends into the VE, shown in figure 1.20 left. When the object is in its

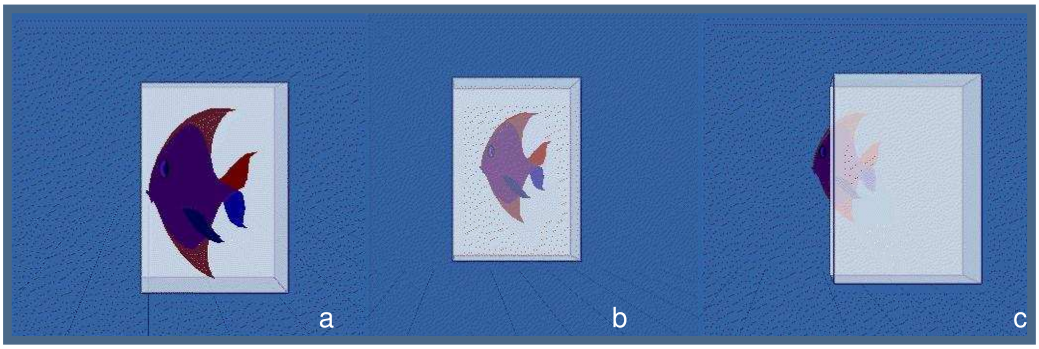


Figure 1.19: Silk cursor. The transparent cube is the silk cursor. a: the object is in front of the silk cursor; b: the object is inside of the silk cursor. c: the object is behind the silk cursor.

[9] Belong to occlusion technique

© 2009 Tous droits réservés.

snap-to angle scope, the bull's eye changes color. Olwal 's flexible pointer [OF03] bends the ray around the occluding objects. Hauptmann [Hau89] proposed the voice aid for eye-gazed selection, to disambiguate the eye-gaze and gesture information.

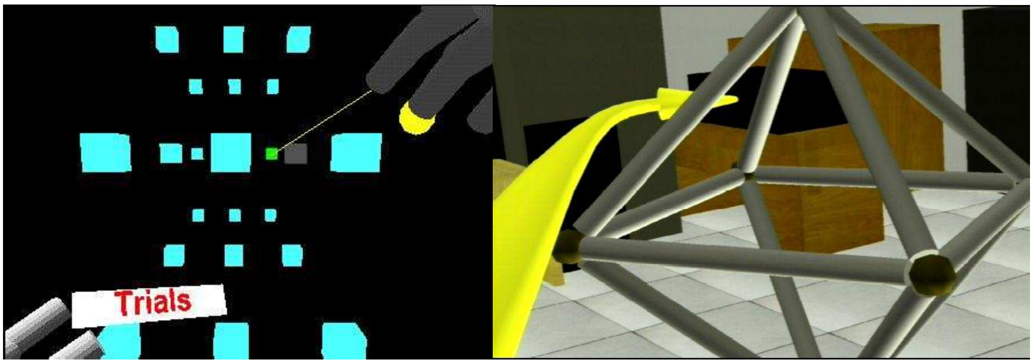


Figure 1.20: Two technique for selection facilitating. Left: Snap-to ray and bull's-eye. [WBR02] Right: The flexible pointer selecting a partially obscured object, without visually interfering with the occluding object. The pointer also gains access to a larger selectable area than is visible to the user[OF03].

1.7.5 Manipulation Techniques

In this section we talk about the interaction techniques for object manipulations including positioning, orienting.

Positioning Techniques

The techniques components discussed in section 1.7.3 can be used for object positioning. Each of them has its own advantages and disadvantages [PW*98, BH97]: *ray-casting* is efficient for constant distance positioning task, but not suitable for orienting and translating the object along the ray, because only the DOF perpendicular to the ray may be independently controlled, the selected object can be translated along or rotate about this DOF. All other DOF are dependent on one or more of the others. Bowman [BH97] proposed *fishing reel* to enhance *ray-casting* by allowing user reel the object closer and further away through two mouse buttons after selection. However the user performance of *fishing reel* is not clear.

Another reason for manipulation difficulty with *ray-casting* is pointed out as "not hand centered", comparing to the *touching*-based selection technique. Bowman proposed *HOMER* (Hand-centered Object Manipulation Extending *Ray-casting*) technique, where after the selection with *ray-casting*, the user's virtual hand extends so that it touches the selected object, and carries out manipulations directly with the virtual hand until releases at that position, the virtual hand returns to its natural position. Cursor-based ray-casting combines the *ray-casting* and 3D cursor for the same reason [Gri99]. However the precise orientation is always difficult.

© 2009 Tous droits réservés.

The fine-grained object manipulations via the scaled-down world such as *WIM* is difficult.

Rotation Techniques

We have discussed about the 3D rotation with 2D input device in section 1.7.1, and concluded them as time-consuming, we now talk about rotation techniques with high DOF trackers. Most of them use the direct mapping (1:1), a clutching mechanism is necessary to overcome the rotation limit of human's finger, hand and wrist. Poupyrev proposed "*non-isomorphic*"[PWF00] rotation where the input rotation is magnified to reduce the clutching.

The rotation using higher DOF input, in fact, are not rapid enough either. In a experimental of rotation task[HTP97], Hinckley reported that the small ball shaped 6 DOF tracker takes 10.7 for male subjects and 14.9 seconds for female subjects. Ware reported[War90] 13.5 seconds when speed was emphasized and 28 seconds when accuracy was emphasized in the object rotation with a 6 DOF hand-held device. Zhai reported 18 seconds with a glove device and 13 seconds with a small ball devices, declining with practice to 11 and 5 seconds.

Ware [WA04] pointed out that one of the factors affecting the rotation performance is the mismatch of reference frame displayed on the monitor and the reference frame of user where the input device is held. The device held at user's side should be co-located with the object being manipulated. Ware reported that a large mismatch (90 degrees) will be extremely disturbing and result in four or five rotation times longer than the normal. and that the smaller mismatch angles have a relative smaller impact. He argued that a "fish tank"[WAB93] arrangement of VE, shown in figure 1.21, is preferable to place the hand inside of the workspace. If the "fish tank" can not be provided, the input device should be placed as close as possible to the center of the display.

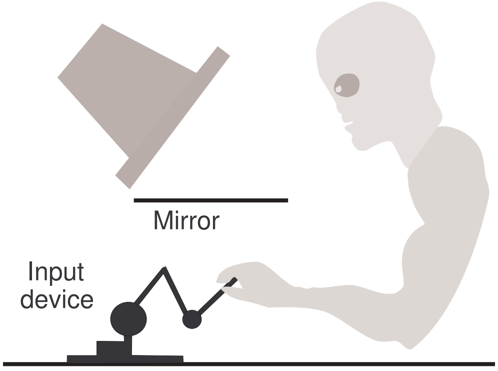


Figure 1.21: Fish Tank arrangement VE. [WAB93]

Constraints can be also applied to rotation task, such as the gridded increment[LG93]. In [SCP95], this value is set to 30 degrees based on the observation that a gridded increment of less than 15 degrees or more than 45 degrees, users report that the rotation behaves as if there is not the increment.

© 2009 Tous droits réservés.

Thèse de Qing Pan, Lille 1, 2008

We have discussed the techniques for selecting, positioning and orienting an object. In next section we talk about the 3D view manipulation techniques with higher DOF input.

1.7.6 View Manipulation Techniques

In this section, we first review the common metaphors in travel techniques followed by a further discussion about techniques based on zooming and widget. We also talk about the two references frame, constraint and feedback in view manipulation, as well as the mode switch when more than one techniques are available.

Common Metaphors

Travel is the most common manner of navigation. Bowman pointed out the five categories of metaphor of travel technique: *physical movement*, *manual viewpoint manipulation*, *steering*, *target-based travel* and *route planning*. Among them, *physical movement*, *manual viewpoint manipulation* and *steering* can be used for exploration task in VE, *target-based travel* and *route planning* for the *target movement* and *specified trajectory movement* tasks.

- *Physical movement*: using the motion of the user's body to travel through the VE, to enhance the sense of presence and the proprioceptive feedback. *Physical movement* can provide a much larger travel range comparing with the manual viewpoint manipulation techniques. For example,"*walking*" metaphor used in *JDCAD* system [LG93] which used a tracker to move the viewpoint via the real movement of user's body. "*Leaning*" metaphor in *Head-butt zoom* [MBS97], the viewpoint moves in the direction of the body lean. "*Shopping cart*" metaphor, proposed by Brooks[Bro88] requires user walking on the *treadmill*, and steering with the handle bar, which is similar to pushing a trolley around in the supermarket. The *treadmill* can be replaced by other devices bicycle, footpad, roller skate [IM92].

- *Manual viewpoint manipulation*: The user's hand motions and gestures are used to affect the travel. For example the "*VPL*" navigation method uses the data glove to recognize a pointing gesture to indicate the travel direction. With the "*Virtual walking*" in "*SmartScene*"[Mul], the view traveling is based on the hand gesture "climbing a rope". "*Eyeball in hand*", proposed by Ware [WO90], maps the orientation and position (or their variations) of the input device onto the viewpoint. It is implemented using the spaceball in *i3D* system [BG95]

- *Steering*: Rate control with continuous specification of the motion direction, such as "*flying*" and "*vehicle*" metaphor, *Flying* without constraint is not natural for most user. "*Flying vehicle*" makes the velocity proportional to the cube of the displacement of the input device. Hand controlled steering technique allows user to look around during flying. Brooks proposed "*Helicopter metaphor*" in the *Walk Through* architecture visualization system, using a *joystick* to simulate a real helicopter steering [Bro88]. With the "*head-mounted display vehicle*" in *DIVE* system, the view

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

follows the movement of a head tracker, user's hands are free to do other manipulations. "*Magic carpet*"[ZK*99] is a bimanual 3D navigation interface using a dual joystick, which seems like a special flying simulator.

- *Target-based travel*: the user specifies the destination, the system handles the actual movement. Several strategies are possible. "*Teleportation*" places the user immediately to the new location, which may make user disoriented. Mackinlay proposed [MCR90b] *rapid controlled movement* technique which moves the user towards a *POI* with a speed logarithmically related to the distance from it allowing user to rapid approach the POI and then slow down when the *POI* is near to avoid the collection. The *Navidget* [HD*08] uses a circling gesture to select the *POI* and the invoke the flying along the simple linear interpolated straight line movement toward the POI. "*Orbital viewing*" [DMH96] and "*HoverCam*" [KK*05] allow user to inspect a single object or point in VE by mapping the rotation of user's head onto the movement along an imaginary sphere around the *POI*. "*Slide on ball*" of *i3D*[BG95] is a similar technique using spaceball as input. In The 2D gesture camera control technique *UniCam* [ZF99], clicking in the border region activates the orbiting around the view center; clicking in the border region followed a dragging and releasing cursor over any object in scene activates orbiting about a specific point. *Navidget* [HD*08] allows user to control the view angle and the distance from the target by interacting with a semi-sphere and the border ring widget, shown in figure 1.22.

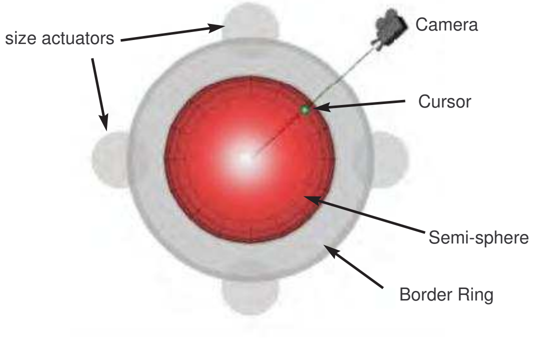


Figure 1.22: Navidget [HD*08]: the view angle is controlled by moving the camera on the semi-sphere surface. The distance from the *POI* is controlled by adjusting the size of the semi-sphere with the four size actuators located at the top, bottom, left, right of the ring border.

- *Route planning*: the user specifies the path that should be taken through the VE, the system handles the actual movement. The user may plane a route by placing directly the icons or markers in the VE, by moving the markers or by drawing a path on a map.

The following metaphor are not included in Bowman's summary.

© 2009 Tous droits réservés.

- *Physical theory based metaphor*: Some view manipulation techniques consider the viewpoint as a "virtual camera object" and applying a physical model using Newtonian mechanics[TR*91] during the view manipulation. Similar technique has been included in *ALIAS 3D modeling package* and used in "Zashiki-Warashi" system[YNS94], [Ali94]. "*Inertial-based landing*" [TRC01] was proposed to address the problem of landing at the end of flying that when the user carried out the landing action by e.g. releasing the button, user felt uncomfortable to fall straight down to the ground, and it's easy to overshoot or undershoot the destination. With *inertial-based landing*, the camera glides forward and eventually lands at the point situating at the bottom of view frustum.

- *Prop-based metaphor*: Some techniques use special physical prop, such as the *WIM*[SCP95]. It is a props-based "*scene in hand*" technique. "*SSWIM*"(*Scaled Scrolling World-in-Miniature*) [WHB06] is based on "*WIM*" and allows user to pan and zoom the *WIM*, in which scrolling is done by dragging the user's representation towards the edge of the *SSWIM*, a large arrow is displayed as the visual feedback; zooming is done through the scrolling wheel of a wireless mouse held in non-dominant hand, shown in figure 1.23.

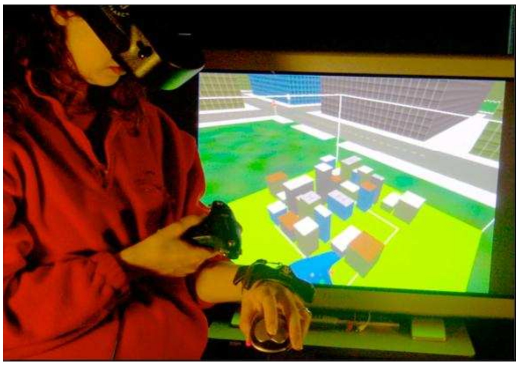


Figure 1.23: The *SSWIM* technique enables immersed users to scale and scroll a *WIM* in VE [WHB06].

- *Possession Metaphor* allows users to see the VE from the view of another actor or object.

For the metaphor using position control such as "*eyeball in hand*", "*scene in hand*", a clutching mechanism is required for a large rotation angle, or translation range which surpass the movement limit of the human's limbs.

Below we present the view manipulation techniques based on zooming, followed by the ones based on widget.

© 2009 Tous droits réservés.

Zooming-based View Manipulation

Zooming is a commonly used 2D view manipulation technique for change the detail level of browsing content. In 3D VE, zooming appears in a number of techniques. Bederson uses zooming as the principal means of navigation of the interface build in Pad++[BH*96]. "*Ephemeral world compression*" [TRC01] is similar to "scaled world" allow user to see all the object to facilitate target searching. Similar to the 2D view technique "*Speed-dependent Automatic Zooming*"[IH00], Ware presented the "*cyclopean scaling and depth modulated flying*"[WF97] technique, in which the scaling velocity is based on the various depth of the object.

Tan[TRC01] use a inverse technique "*inverse scaling*" to reduce the amount of occlusion. The user controls the radius of a sphere centered on his view. The objects falling inside are made scaled down, according to their distance from user.

Widget-based View Manipulation

The technique using widget for the viewpoint manipulation appeared in *Haptic-GeoZui3D* [KW03] system which allows user to yaw, pitch, scale and translate the view in VE by manipulating the widget located at the center of view. It is shown in figure 1.24.

In the Xerox 3D ROOMs system [MCR90a] widget was used to control the travel around, clicking and dragging the "*four way arrow icon*" translates the viewpoint in the virtual body with the speed corresponding to the vector between the icon and the user cursor, "rubber-binding" virtual feedback is offered. Clicking and dragging "*eye icon*" rotates the gaze. This mouse, widget based navigation is involved in *DIVE* VR system[CH93] and VRML browser such as *Place* and *VRweb*.

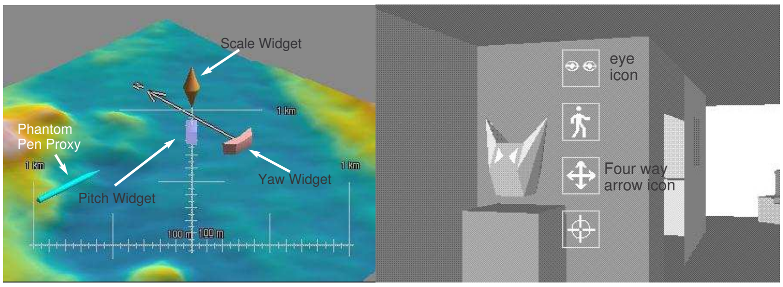


Figure 1.24: Left:Scene navigation widget with *PHANToM* pen proxy in Haptic-GeoZui3D [KW03]. Right: Navigation Icons in the VRweb VRML Browser

In following sections we present several factors in view manipulation techniques, including the reference frame, constraint, feedback and the mode switch if there are more than one view manipulation techniques is available.

Reference Frame of View Manipulation

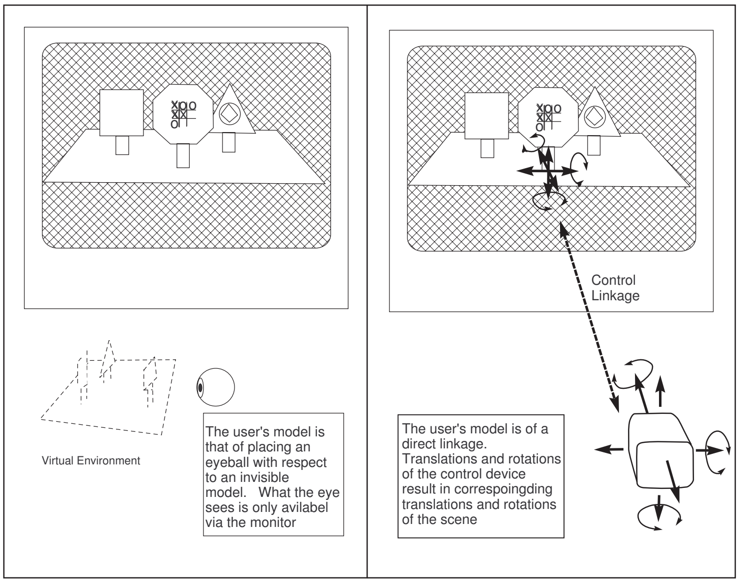
There are two different reference frames during the view manipulation: egocentric and exocentric.

© 2009 Tous droits réservés.

Thèse de Qing Pan, Lille 1, 2008

Egocentric travel is also called the *first person travel* which refers to moving the viewpoint through the VE which creates the sense of being in that space in the user. The metaphor "*eyeball in hand*" and "*Flying vehicle*" proposed by Ware[WO90] are two techniques for egocentric travel.

Exocentric travel, is also called the *third person travel* which gives a feeling of looking in VE from outside, a "bird-eye" or "God's eye". "*Scene in hand*" [WO90] is an exocentric viewing technique, which maps the movement of the 3D input device onto the virtual world, so that the exploratory viewing is performed by keeping the viewpoint still, and moving the world around. "*Scene in hand*" is used in *3-Draw* [SRS91] and "*Ball and mouse*" in the sculpting system of LeBlanc with a spaceball[LK*91]. "*Eye in hand*" and "*Scene in hand*" are shown in figure 1.25.



The user's model is that of placing an eyeball with respect to an invisible model. What the eye sees is only avilabel via the monitor

The user's model is of a direct linkage. Translations and rotations of the control device result in correspoingding translations and rotations of the scene

Figure 1.25: Left: Eye in hand. Right: Scene in hand

Egocentric frames provide a local view while exocentric frames provide a global view. Some techniques combine the two reference frames. For example in [PB95] the user's view can be temporarily detached from the egocentric frame to give a more global view of environment.

Constraints for View Manipulation

We have discussed the constraint technique in section 1.7.3. In this section we focus on the technique using constraint of view manipulation, which is a way to help the user to succeed the navigation. The non-constraint camera techniques such as *eyeball in hand*, *scene in hand* can lead to user's disorientation or position inside of an object. Commonly used constrains include the constrained height, terrain following which can be considered as a collision detection between the user's avartar and the scene. "*Object inspection*" navigation around a 3D object, such as in *HoverCam*[KK*05], keep the camera a fixed distance from the surface and relatively normal to the surface of the object being observed.

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

Hanson used the *hiding surfaces*[HW97] to constrain the view motion: each point of the constrained surface has an associated viewpoint dynamically generated in such a way that users do not miss important objects while navigate near them.

Bares [BL99] proposed the constraints applied directly on the camera's parameters: *vantage angle*, *viewing distance* and textslposition. In "*speed coupled flying*" technique the flying speed of the view is coupled with the height and titl of viewpoint, to allow a smooth transition between the local view and the global view.

Constraints could bring better performance. Subramanian carried out a navigation / selection experiment on his *hybrid 3D and 2D interface*. The result showed that the mode allowing the user to control only the position or rotation at a given time is the fastest. The free camera movement without constraint is the slowest.

Feedback during Viewpoint Manipulation

The feedbacks such as visual feedbacks, locomotions, kinesthesis feedbacks and audition feedbacks are important to help user judging the direction, egocentric orientation, relative orientation of the body parts, to help user to know where he is, how fast he is moving, more importantly, help user to understand the scene as a whole.

Vestibular feedback is the feedback information about movement and balance. It is originally produced by the vestibular system or balance system which located in the vestibulum in the inner ear of human. *Veribular* and *proprioceptive* cue can be provided by locomotion. An artificial method is proposed by Maeda[MA*05], where an electronic current is send to between ears through a helmet to make the user feel disequilibrium towards left or right.

Passive feedbacks are also useful. Such as the "*Virtual Treadmill*"[SSU93] tracks the head movement to detect when the user is walking. This action is translated into locomotion within VE, ensuring a certain amount of kinesthetic correspondence.

Switch between View Manipulation Techniques

A number of systems use more than one view manipulation techniques, a good switching should be considered to allow smooth interactions. Some switching are inherent. For example *Speed-coupled flying with orbiting* [TRC01], use *speed-couple* flying for navigation through the VE, "*orbiting*" is used to inspect the object intersected with it. When the user drags on an object, the orbiting inspecting is activated, while when the user drags on the free space, flying is activated.

We have reviewed a number of interaction techniques in a general manner. In following sections we focus on two kind of interfaces — bimanual interfaces and hybrid position-rate control interfaces.

1.8 Bimanual Interactions

In every day life, people use both two hands to interact with the real world. Compared with the traditional unimanual interfaces, bimanual interfaces allow users to transfer the

© 2009 Tous droits réservés.

ingrained interaction skills, thus are more intuitive, easier to learn, and may provide better performance on certain tasks and reduce the training time. The bimanual interactions break down the tasks into a series of sequential operations, each hand controlling fewer DOF, and perform operations simultaneously. In this section, we first talk about the fundamental theories of bimanual interfaces — *Kinematic Chain Model* proposed by Guiard [Gui87], and some relative experimental researches as well as the advantages of bimanual interface. Then we list some techniques including symmetric and asymmetric interfaces for 2D and 3D environment, as well as the ones for table-based system. At last we summarize the input devices used these techniques

1.8.1 Theories of Bimanual Interactions

Guiard's analysis of human skilled bimanual actions [Gui87] provided an insightful theoretical framework for classifying and understanding the roles of both hands. Guiard pointed out that the every day manual activities falls into three classes: unimanual activities, such as brushing one's teeth; symmetrically bimanual activities where the two hands play the same role in tasks, such as such as weight lifting, climbing a rope, rowing, jogging, or swimming; and asymmetrically bimanual activities where the two hands play different roles, such as writing and drawing. For asymmetric bimanual interaction, Guiard proposed this *Kinematic Chain Model*(*KC*).

Guiard's Kinematic Chain Model

1. Motion of the dominant (*D*) hand typically finds its spatial references in the results of the non dominant *(ND)* hands. The D hand articulates its motion relative to the reference frame defined by the ND hand. This principle is also called *right-to-left* principle. For example, when writing, the ND hand controls the position and orientation of the page, while the D hand performs the actual writing by moving the pen relative to the ND hand.

2. The D and ND hands are involved in asymmetric temporal spatial scales of motion. The movements of the D hand are more frequent and more precise than those of the ND hand. During handwriting, the movements of the ND hand adjusting the page are infrequent and coarse in comparison with the high-frequency, detailed works are performed by the D hand.

3. The contribution of the ND hand starts earlier than that of the D hand. The ND hand precedes the D hand: in the writing example, the ND hand first positions the paper, then the D hand begins to write.

Experimental Researches

A number of experimental researches were carried out according to Guiard's study. In earlier research, Buxton measured the use of two hands in positioning, scaling task and scrolling to known parts of a document. This study found that unimanual interface is inferior to the bimanual interface which splits the compound tasks in parallel by both

© 2009 Tous droits réservés.

Thèse de Qing Pan, Lille 1, 2008

hands and that the user naturally adopts parallel use of both hands and could scroll faster with the left (ND) hand [BM86].

Balakrishnan and Hinckley[BH99] have shown experimental evidence that even if the hands work in two separate kinesthetic frames, each with its own independent origin, the principle of the *right-to-left* spatial reference still applies.

Casalta [CGB99] tested asymmetric bimanual, symmetric bimanual and unimanual interface with a rectangle specifying task. The preliminary results revealed better performances and a higher degree of bimanual parallelism with the symmetrical than asymmetrical options. They concluded that in the absence of a marked scale contract between the bimanual movements, the *right-to-left* principle does not hold.

Kabbash, Buxton and Sellen [KBS94] argued that bimanual interfaces could be worse than unimanual interfaces if an inappropriate interaction technique is employed, particularly when cognitive load is increased. This seem to be verified by the experiments carried by Balakrishnan[BK99]. He performed experiments of selection and docking to compare the bimanual interfaces with two mice and unimanual interface with one mouse. The result from his object docking experiment showed no significant difference in average task completion time between the bimanual and unimanual interface. He suggested that the performance of the bimanual interface may have suffered because the interaction style deviated from the Guiard's KC model — the interface encouraged a parallel symmetric style of interaction.

Some other experimental studies didn't find the better performance of bimanual interfaces. Dillion, Edey and Tombaugh[DET90] compared a single cursor method of menu and task manipulation to a dual cursors method. The result showed no significant improvement in bimanual actions. Myers[MLY00] compared a bimanual and a unimanual interface for web page scrolling tasks where the user scrolls the web page and select the target link with the mouse controlled by D hand. In bimanual interface, the D hand control a PDA sized touch sensitive "*WordPad*" , scrolling is done by using the buttons, slider, rate scroller. Mayers reported that the completion time is improved by the bimanual interfaces in earlier similar experimentation, the unimanual interfaces are faster than the bimanual interface.

Advantage of Bimanual Interfaces

Bimanual interfaces provide a better proprioception feedback. Hinckley, Pausch and Profitt[HP*97] experimented with a bimanual user interface on an object-alignment task in a 3D environment. They concluded that the use of both hands forms a hand-relative-to-hand reference frame that can help the user to gain a better sense of space in which he is working in. Hinckley pointed out that this proprioception is independent of the virtual feedback [HCS98].

Some studies[BK99, LZB98] pointed out that the bimanual interfaces are useful not only for actions which are directed towards a task goal, but also for actions which facilitate cognition. For example, Balakrishnan [BK99] noted that in docking task experiment, when using the bimanual interaction technique, the users tended to move the camera around more, in order to enhance their perception of the 3D scene. Owen [KF*98] observed several advantages of bimanual interaction. Firstly, bimanual reduce and externalize the

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

load of planning / visualization in unimanual input, because bimanual is natural that user no longer need to compose, "thinking", and plan the elemental step task; Secondly, rapid feedback of manipulation results in a higher level of task, the user immediately sees the result of action in relation to the goal state; Thirdly, bimanual supports epitemic actions: the user may take advantage of the two-handed input and perform action of an epistemic nature in addition to those of a pragmatic nature. Leganchuk [LZB98] pointed out that the bimanual interfaces perform tasks in parallel, providing better mental representation. Buxton has another opinion, he proved that bimanual interactions do not significant affect the cognitive load [BM86].

Owen[OK*05] investigated the relationship between the bimanual interaction and the cognitive aspect of the task integration, divided action and epidemic actions. An empirical study compared the bimanual technique versus the unimanual technique for a curve matching tasks where the user should control two curve vertices of the responsive curve to match the target curve. This study found that the bimanual technique resulted in better performance than unimanual technique, and as the task becomes more cognitively demanding, the bimanual technique exhibited even greater performance benefits.

1.8.2 Bimanual Interfaces

After analysis of the performance of bimanual interfaces, we summarize some existing techniques. We list the bimanual techniques of four categories: symmetric 2D bimanual interfaces, asymmetric 2D bimanual interfaces, symmetric 3D bimanual interfaces, asymmetric 3D bimanual interfaces.

Symmetric 2D Bimanual Interface

The symmetric bimanual motions are useful to specify the scale or extent. A bimanual technique for sweeping out a rectangle proposed by Kreuger [Kru91].

Latulipe[Lat04] investigated the symmetric bimanual interaction for 2D drawing application. In her research, there were two cursors on the screen, controlled respectively by two hands. The users can control two parameters during interaction simultaneously. For example, "*symDraw*" controls the size and position of a circle."*SymDrive*" shown in figure 1.26., controls the scaling / translation or rotation / translation of an selected object, where when two cursors move in the same direction with the same speed, the selected object translate, as soon as one of the mice starts to move in diverging direction or with diverging velocities the selected object will rotate.The similar approach is applied to "*symSpline*",where the two cursors control the two end of the tangent of a control point of the curve. The simultaneous translation and rotation allow users to magnify the shape of the curve, implicitly combine two functionalities into a single gesture. In Latulipe's following work[LM*06], she compared the *symSpine* with two asymmetric dual-mouse techniques and a standard single-mouse technique. The *symSpline* outperformed the two asymmetric dual-mouse techniques and was the most preferred by user.

Bimaunal interfaces are commonly used in the tabletop system with the touch sensing technique involved to detect the position of finger(s) or hand(s) on the tabletop. *VIDEODESK* [Kru91] uses a camera and image processing to track 2D hand position

© 2009 Tous droits réservés.

Thèse de Qing Pan, Lille 1, 2008

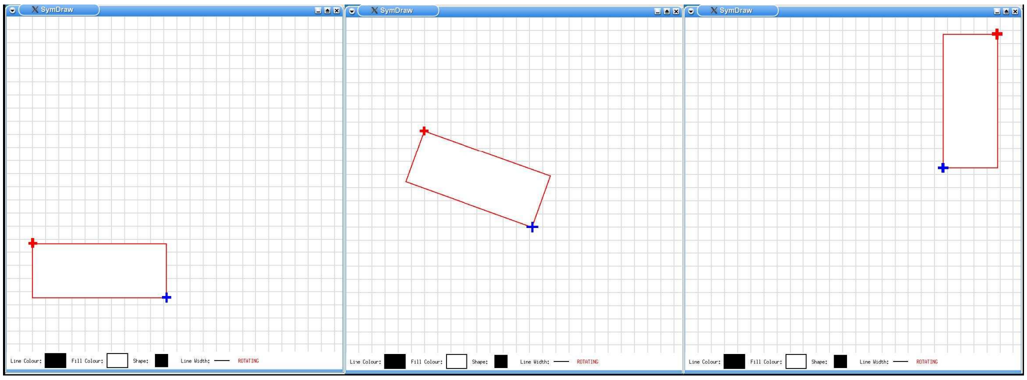


Figure 1.26: SymDrive.

and to detect image features such as hands, fingers. The index finger and thumb of both hands simultaneously manipulate four control points along a spine curve.

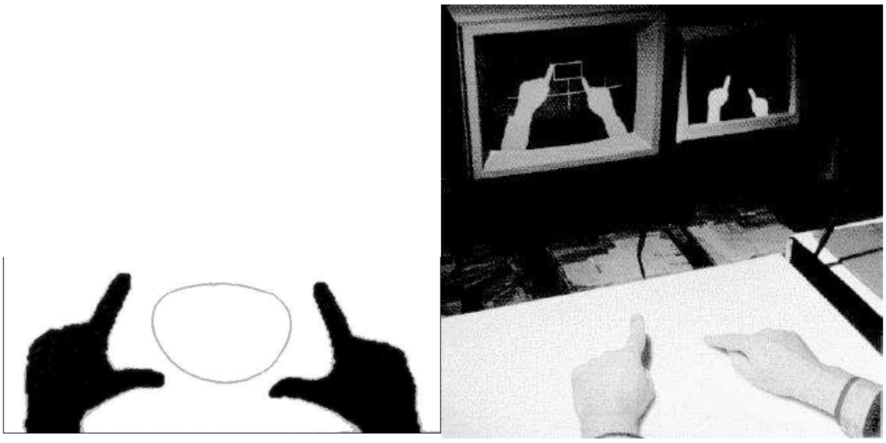


Figure 1.27: Example VIDEODESK applications using two hands [Kru91]

Asymmetric 2D Bimanual Interface

Toolglass and *magic lenses* proposed by Bier[SP*93] consist of a semi-transparent menu which the user places superimpose upon a target using a trackball in ND hand, the D hand moves the mouse to the target and click through the menu to apply an operation.

"*MacPaint*" of Buxton [Mac] uses the D hand to paint, and the ND hand to pan when the user wants to continue the painting beyond the screen via a trackball. Balakrishnan's study[BP98] uses gesture drawn with the ND hand on the small touchpad of "PadMouse" to invoke commands or modifiers. This technique is aimed at facilitating the selection and positioning. In some bimanual interfaces the ND hand plays a supporting role for D hand, such as controlling other drawing tools "*T3*" [Kur97] a high-end artwork tools using tablet, two-handed, transparent, and adjusting translation and scaling in [ZSS97b, BM86]

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

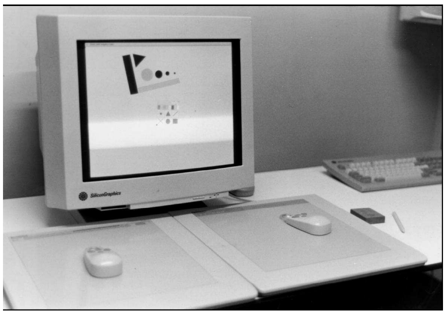


Figure 1.28: Configuration of *T3*: Tablet, two-handed as input, transparent widget.

Symmetric 3D Bimanual Interface

In the immersive architectural design application of Mine[Min], the separation of the hands indicates the magnitude of object displacement, and to select options from the *mode bar* extending between the hands. Mapes' "*Polyshop*" [MM95] uses symmetric bimanual tools to perform rotation, scaling, stretching the object. Zhai's "*Bulldozer*"[ZK*99] shown in figure 1.29 uses two trackpoints controlled by both hands as input. Pushing both trackpoint forward moves the view forward. Pulling both trackpoints backward, moves the view backward. Pulling the left hand and pushing the right hand turn the view to left; pulling the right hand and pushing the left hand turn the view to right. Pushing the two trackpoints in opposite direction move the view up (outward) and down (inward). This technique allows simultaneously turning and moving forward as in a real *bulldozer*. A dead zone is added to reduce possible coupling between DOF. An example of DOF coupling is that when the user intents to move forward so pushes both trackpoints forward, but may be not parallel, so produces an unintended up or down movement. The dead zone can filter that out. The formal study showed the *bulldozer* outperformed the mouse in navigation, due to not requiring the mode switching and the capability between the trackpoint and rate control. "*Two handed flying*" [MBS97] uses both hands to control the flying, where the travel direction is specified by the vector between user's two hands, the travel speed is proportional to the separation between the hands. A dead zone is added to stop the travel when two hands are brought together. *Two handed flying* is a two handed steering technique.

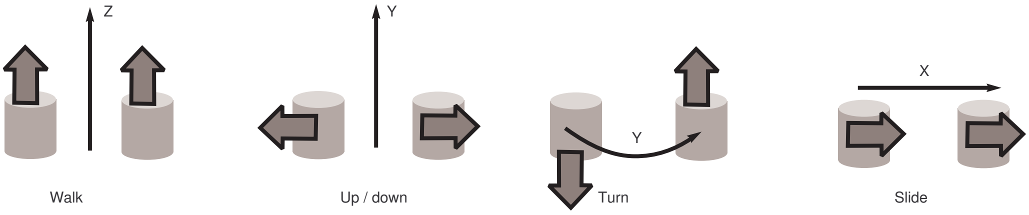


Figure 1.29: Bulldozer metaphor [ZK*99]

© 2009 Tous droits réservés.

Thèse de Qing Pan, Lille 1, 2008

Asymmetric 3D Bimanual Interface

A hybrid 3D and 2D interface is proposed by Subramanian [SAM03] is an asymmetric bimanual interface where the *rigid intersection selection props* (*RISP*)" is used to get the cross-section image through the model by moving the *RISP*. The digital pen controls "*Virtual Pins*" which can be placed at any point in the cross-section image, and the intersection image is pinned to the location within the 3D model correspond to that point. The RISP performs 3D movement, but the digital pen's movement is constrained on the 2D surface RISP.

"*Ball and mouse*" metaphor in the sculpting system[LK*91]. In this system, the SpaceBall held in the ND hand orients the sculpture, the mouse in the D hand is used to deform the vertices making up sculpture.

"*THRED*"[SG97b], a polygonal surface design, uses the ND hand to indicates the constraint axis, to perform the menu selections and to orient the scene; the D hand performs all detailed manipulation. The original research did not carry out the formal behavioral study.

Maarten[Maa99] use two "*frogs*" — a 6D hand-held tracker embedded device, to control two 3D cursor. The N hand cursor can select and drag more than one objects. The ND cursor can position and orient objects that are not being dragged by the D hand cursor. The ND can aid the selection by moving the object closer to the D hand cursor. His experiment study carried out an object assembly task. The result showed that the bimanual interface is 17.5% faster than the unimanual interface.

Zeleznik[ZFS97] uses two cursors to perform object rotation, object scaling, and navigation. The rotation process is shown in figure 1.30 where the D hand cursor selects a point D on the surface of the object, the line between D and the object center C define the ration axis. The ND hand cursor selects another point P to control the amount of rotation. Object Scaling is performed in a similar way. In the navigation mode, the D hand performs the navigation as in the VRML browser, the ND hand controls the elevation of camera from the ground. In flying mode the ND hand controls the elevation and the roll of camera.

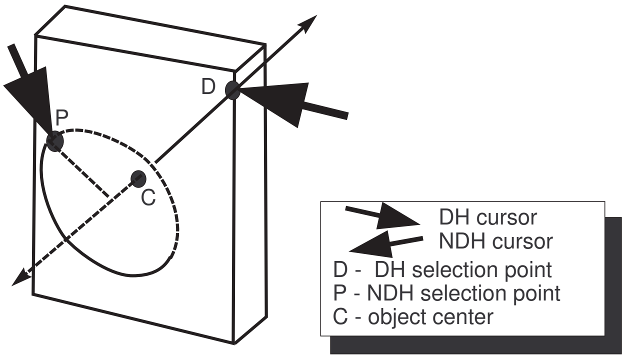


Figure 1.30: D Hand defines virtual sphere through point D. ND hand rotates object around the axis CD.

"*3-Draw*" [SRS91] is a CAD tool facilitating the sketching of 3D curve, the user holds

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

Thèse de Qing Pan, Lille 1, 2008

a stylus in D hand to draw and to edit the curve of an object and ND hand holds a tablet which is used to view the object. This study found that motions between hands requires less concentration of user than a fixed single object and one freely moving mouse.

Mine [MBS97] proposed "*hand-held widget*" attached to user's ND virtual hand. This study showed that it's easier to find the tools comparing with the technique attaching widget to the object. "*Object in hand*" [DC*04] uses the user's ND hand to grab a selected object. After that, the user interacts with the object through "*virtual hand*" metaphor. When releasing the object, it automatically moves back to its original position.

The ND hand can provide motion constraints for the object controlled by the D hand. In the editing of features of the object, such as the vertices, edges, surfaces. For example, the ND cursor is positioned over a surface of the polygonal object to define an interaction plane, the D hand cursor selects another feature, and moves it in a plane parallel to the surface selected th the ND hand cursor.

Tabletop-based System Combining with Bimanual Interface

A number of bimanual interfaces are proposed for the tabletop-based system. "*Virtual WorkBench*" [PS96] display a 3D image on an opaque mirror in front of the user's face. Both two hands move physical tool handles which can have different virtual effectors attached to them, depending on current mode. However, Poston did not perform the formal experimental evaluation.

"*Responsive workbench*" [CFH97] uses bimanual manipulation on the stereoscopic, rear-projected tabletop VR display and allows bimanual zooming via setting focus point with ND hand, and D hand move forth and back to zoom. For rotation, the ND hand specifies the rotation axis, and the D hand controls the rotation. Another rotation mode is "*steering wheel*" symmetric rotation where both hands grab the model and rotate. *Virtual WorkBench* and *responsive workbench* are used in medical training applications.

Other similar systems include *metaDesk* [UI97], *reactive workbench* [Fak], *ImmersaDesk* [Ele] with large display filling wide visual angle, and stereoglass worn by users.

Input Devices in Bimanual Interfaces

The input devices used in bimanual interfaces are varied. Some interfaces provide similar devices for each hand. Maarten[Maa99] use two "*frogs*" — a 6D hand-held tracker embeded device, to control two 3D cursor. Cline [Cli] use two input devices similar to EGG for each hand. *VIDEODESK* [Kru91] uses the fingers as the input.

Some interfaces use a board shaped prop in ND hand, and a different device in D hand. "*3-Draw*" [SRS91] uses a tablet and a stylus as input. Veigl [VK*02] proposed a wrist-mounted augmented reality panel made from a simple 2D touchpad for ND hand, the D hand performs the gestures including pointing, grabbing, stretching for interaction. *Hybrid 3D and 2D interface* of Subramanian [SAM03] use the RISP in ND hand, and digital pen in D hand.

Some interfaces adopted physical props as input devices. Hincklay used the doll prop in a neurosurgical visualization system [RGK94], Bricks[FIB95].

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

[BM86] used a pair of 1D touch sensitive strip for ND hand, and a mouse for D hand. "*Ball and mouse*" [LK*91] uses a spaceball and a mouse as input. In *Reachin* technique[Rea] a spacemouse in ND hand and a PHANToM controlled by D hand are used.

Froehlich proposed several input devices handed and manipulated with two hands. "*Cubic Mouse*" [FP00] a cube-shaped 6DOF tracker embedded with 3 rods which can be pulled and pushed to constrain the DOF. The cube is held in ND hand, the rods are manipulated by the D hand. *Yo-Yo* and *SquareBone* consist of two spacemouse sensors, are manipulated by both hands.

With 6 DOF input device in each hand, the 12 DOF available is more than necessary, which provides certain flexibility but can also make some tasks more difficult. Because users are allowed more DOF than what is strictly required for the task, constraints are necessary during interactions. A better solution is the bimanual interface combining a 3D device in one hand, and a 2D device in the other.

1.9 Hybrid Position-Rate Control Interface

We have generally talked about the interaction techniques in the previous sections. In this section, we focus on one hybrid position-rate control interfaces.Hybrid position-rate control interfaces combine both control modes into one device. This can be done simultaneously with two different physical position and rate controls mapping each to different outputs [ZSS97c]; however controlling the object movement with two controls simultaneously is not feasible. A more general solution is to make the device bimodal, using either position or rate control, and always controlling the object movement directly. A common example is used in many applications using scrolling windows: when dragging and selecting items, the input switches from position to rate control as the cursor crosses the boundary of the visible window. In practice, it is difficult to move in arbitrary 2D directions and rate control is difficult, because without haptic feedback the position-to-rate transition point is difficult to perceive and self-center [Zha95].

In virtual environments, Bowman and Hodges' "*Stretch Go-Go*" technique [BH97] improved Pouyrev's *Go-Go* to make the user's range be infinite. It used visual feedback to help controlling the rate and self-center. The virtual hand is controlled with position control but the arm length is expanded or contracted with constant rate control when the hand enters circular near or far regions. The use of circular zones allows rate-control movement in any direction.

Tactile 3D [Tac] is a commercial 3D file browser using hybrid position-to-rate control with visual and audio feedback. Rate control is used to rotate the camera with rotation speed proportional to the distance from the circular zone.

Synaptics touch pads include a hybrid technique called "*EdgeMotion*™ " [Edg]. At the edge of the touch pad, an isometric rate control mode is activated by switching to a downward pressure. In practice, transitioning from horizontal movement to vertical pressure for rate control may not be intuitive. Also, because of the rectangular shape of the position control zone, continuing pointer movement in the same direction in the rate control mode is difficult. No user evaluations have been reported for *EdgeMotion*.

© 2009 Tous droits réservés.

1.9.1 “Bubble” Technique

Donminjonet al. proposed a 3D hybrid position-rate control technique, uses elastic feedback with a large Virtuose 6 DOF force feedback device [DL*05]. *Bubble* technique is shown in figure 1.31 A spherical volume is simulated in physical space and visualized as a transparent sphere on the display. When the end effector is inside the volume, cursor movement is in position control with constant *CD ratio*. When the end effector is moved beyond the spherical volume, the device uses rate control with elastic feedback. To our knowledge the authors have not conducted any sound user evaluation, and there is no satisfactory theoretical basis.

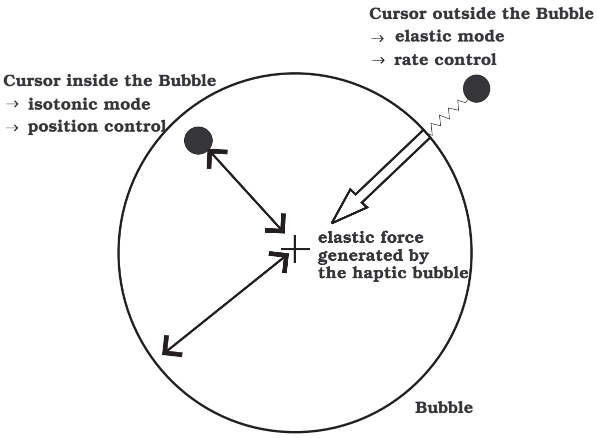


Figure 1.31: Control mode in *Bubble* technique [DL*05]

The “*BubbleCam*” is a technique based on “*Bubble*” [DL*05] where the virtual camera is attached to the center of *Bubble* permitting to manipulate and navigate in large VE with a limited input device workspace. The two techniques proposed in this thesis *RubberEdge*, *Haptic Boundary* are based on this technique.

1.9.2 Isotonic-Elastic Input Devices

In this section, we briefly present some hybrid input devices embedded both isotonic sensors elastic sensor resistances, on which the position and rate control are applied to provide a better user performance. The devices are presented including: GlobeFish[Fro06], GlobeMouse[Fro06], Two-4-Six. The User can use one hand to hold and manipulate them. Then we present the two handed hybrid devices.

GlobeFish and GlobeMouse

GlobeFish[Fro06] is a 6 DOF input device that combines elastic translational input with isotonic rotational input. It consists of a 3D trackball embedded in a spring-load frame. For translation, user move the trackball in any direction using force; for rotation, the user turn the trackball as the traditional one. This device can be used to place and rotate

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

sequentially, not simultaneously. GlobeMouse is similar to GlobeFish. It consist a 3 DOF trackball on the top, and a SpaceMouse at the bottom as the frame.

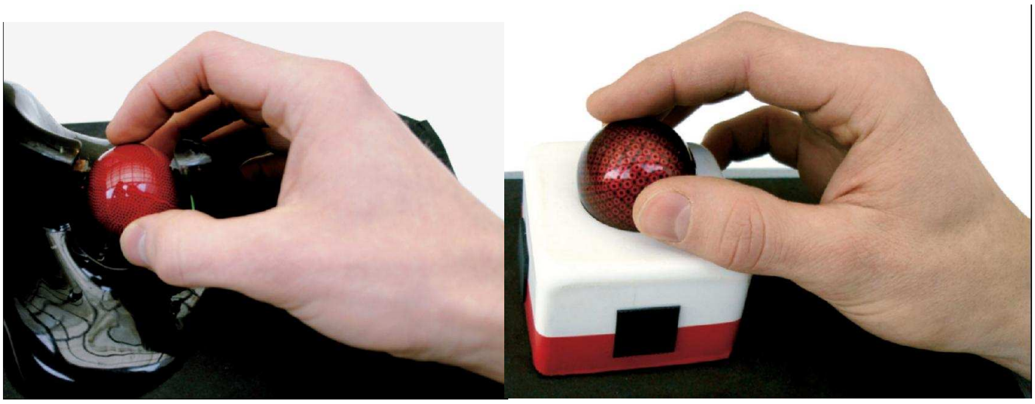


Figure 1.32: GlobeFish (left) and GlobeMouse (right)

Two-4-six

Two-4-six [KBF06] is an input device designed for navigation-centered application. It consists of a gyroscopic sensor for indicating the travel and pointing directions. For rate control of travel velocity, the user manipulate a rocker-lever in the front part of the handle with the index and middle finger. The touchpad atop the device senses the absolute position of the fingertip, which is used for rotation using Shoemake's "arcball" technique. The elastically suspended ring framing the touch pad offers two more DOF.

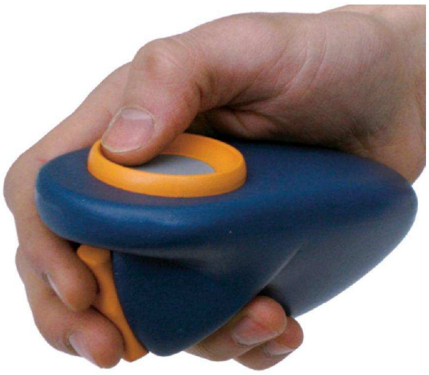


Figure 1.33: The two-4-six device is designed to navigation

Two Handed YoYo and SquareBone

YoYo [SF03] consists of two symmetric elastic 6 DOF SpaceMouse sensors attached to a grip in the middle with a 6 DOF embedded tracker. The device is operated with the

© 2009 Tous droits réservés.

Thèse de Qing Pan, Lille 1, 2008

caps of the SpaceMouse sensor using rate-control. The left sensor is typically used for navigation, the right one for object manipulation.The 6 DOF tracker in the grip is used to compensate for rotation of YoYo against the world coordinate system. This tracker may also used for isotonic input and position controlled rotation or translation. The *SquareBone* is based on YoYo, the caps of the spacemouse sensor are replaced by square handles providing a distinct tactile coordinate system. As for the YoYo, the handles of *SquareBone* are manipulated with the fingertips. Contrary to the YoYo, both handles of *SquareBone* can be operated simultaneously. The SquareBone has its grip on the outside and rests comfortably in the user's hands.

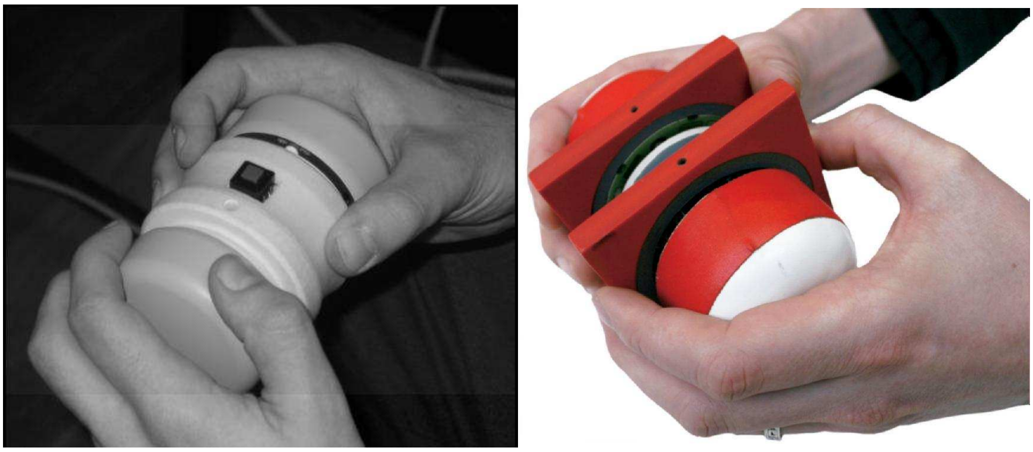


Figure 1.34: YoYo (left) and SquareBone (right)

1.10 Conclusion

In this chapter, we first presented the elemental interaction tasks in both 2D and 3D environment including object selecting, positioning, orienting and view manipulation as well as the parameters of each elemental task which define the complexity of task and affect the user performance. These interaction tasks are what many research work devoted to, so does this thesis.

Then, we presented one of important factors of interaction techniques — the input device, including its properties: discrete and continuous, relative and absolute, DOF integrality and separability, resistance and the haptic feedback of input devices. The properties of input devices are the foundation of the interaction technique design and the applicable tasks.

Then we present another factor of interaction techniques — the transfer functions. The common ones include position control and rate control. Zhai's study of relationship between the device resistance and transfer functions was presented.

We collected the interaction techniques for object and view manipulation techniques for each kind of tasks.

© 2009 Tous droits réservés.

http://www.univ-lille1.fr/bustl

Thèse de Qing Pan, Lille 1, 2008

We then focus on the bimanual interfaces which became the focus of a number of research work, we presented its theory foundation Kinematic Chain Model proposed by Guiard and the bimanual techniques based on this theory. Although the Guiard's theory demonstrate that the bimanual interface should outperform the unimanual interface, but the working condition is not easy to respected. An experimental comparison between our technique *Haptic Boundary* and an existing bimanual technique is described in later of this thesis.

At last we introduced some existing techniques and the input devices combining more than one transfer functions. Our propositions are also belong to this category. Some presented techniques gave us a lot of inspirations.

Base on these knowledges, we proposed the two techniques: "*RubberEdge*" for 2D object selecting task and *Haptic Boundary* for object manipulation and navigation the 3D VE. They are described in following chapters.

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

Chapter 2

2D Isotonic Elastic Hybrid Device: *RubberEdge*

Contents

© 2009 Tous droits réservés.

Thèse de Qing Pan, Lille 1, 2008

2.1 Résumé

Dans ce chapitre, nous présentons *RubberEdge* — un technique pour réduire le débrayage.Il est basé sur le concept hybride avec un contrôle isotonique en position et avec un contrôle élastique en vitesse, conçu pour les taches de sélétion en 2D.

Pour la sélection en 2D, le problème du périphérique isotonique combiné avec un contrôle en position, comme la souris, est le débrayage. Il augmente de façon exponentielle quand on utilise un petit volume de travail pour interagir avec un grand écran ou un écran à haute résolution. Trop de débrayage dégrade fortement la performance d'utilisateur. L'accélération du pointeur qui change le *CD gain* en fonction de la vitesse de mouvement de la souris, est proposée pour réduire le nombre de débrayage, mais sur un petit volume de travail, son avantage s'avére limité. Une solution alternative est un périphérique élastique avec un contrôle en vitesse, mais c'est une méthode moins intuitive.

RubberEdge est basé sur la technique *Bubble*. *RubberEdge* évite la discontinuité de vitesse du curseur pendant la transition entre modes grâce à une nouvelle fonction de transfert. Cette amélioration est obtenue en tournant et translatant la frontière entre la zone isotonique et le bord élastique après la transition entre modes et par un lissage des vitesses avant et après la transition. Nous simulons ensuite *RubberEdge* sur le *PHANToM*.

Une évaluation formelle de la tache de sélection en 2D est effectuée. Le résultat montre que, pour le temps de sélection, *RubberEdge* est de 20 % plus rapide que le contrôle en position. L'avantage en performance de l'accélération du pointeur est également perçu au travers des résultats. Nous avons constaté que les participants n'ont utilisé que 75% de l'espace disponible pendant chaque débrayage. Plusieurs participants ont préféré *RubberEdge* pour sa rapidité à couvrir les distances longues, d'autres participants ont préféré le contrôle en position pour son exactitude.

Nous avons observé de l'expérience que *la loi de Fitts* n'est plus valable pour le contrôle en position avec des débrayages nombreux, elle cesse aussi d'être valable pour la technique hybride combinant plusieur fonctions de transfert. Nous avons mis au point deux modèles de performance pour le control en position avec débrayage et la technique hybride afin de prédire le temps de sélection. Nous avons trouvé une bonne correspondance entre les temps prédits des modèles et les temps mesurés par expérience.

Egalement, par les deux modèles, nous avons observé que pour les cibles proches, le contrôle en position n'a besoin que d'un ou deux débrayage, il est plus rapide que la technique hybride. Pour offrir plus de souplesse aux utilisateurs, le contrôle en position avec débrayage et le contrôle en vitesse sont tous les deux présents dans le nouveau prototype basé sur le Touchpad que nous avons fabriqué pour les ordinateurs portables. Une calibration de deux étapes a été conçue pour les nouveaux utilisateurs. Une première évaluation des taches de Windows a été effectuée. Ce nouveau prototype est facile à comprendre pour les participants. Les participants préfèrent le contrôle en vitesse au

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

Thèse de Qing Pan, Lille 1, 2008

niveau du bord élastique pour parcourir des longues distances de façon continue et ils préfèrent aussi le contrôle en positon dans la zone isotonique pour les taches qui demandent plus de précisions comme dessiner.

2.2 Introduction

We have reviewed a number of techniques for the basic interaction tasks. In this chapter, we focus on the selection task in 2D environment, we adopt the *hybrid position-rate control* to this task. For the most WIMP operating systems, the *relative position control* devices, such as the mouse, are commonly used. In most of time, they will perform better than a rate control device, such as a joystick [CEB78, DM94]. However, a potential issue with *position control* devices is when *clutching* — the momentary recalibration to avoid running out of input area — becomes more frequent, taking additional time [JC90, MO98]. Recently the resolution of digital displays has increased significantly, while the input area remains fixed, making *clutching* more of an issue. For example, laptops are available with 38cm (15") displays with resolutions in excess of 1400 x 1050 pixels, yet the touch pad input space remains at about 4cm. With wall-sized displays, the difference is even greater.

Clutching can be reduced by increasing the ratio of display movement to control movement (Control-Display gain, or *CD gain*), but high *CD gain* can hurt performance [AZ01, JC90, Joh94, Zha95]. An alternative is to dynamically adjust *CD gain* based on the input velocity, called *Pointer Acceleration* [JC90, Poi]. This technique uses low *CD gain* at low velocity to improve precision and high *CD gain* at high velocity to cover large distances with minimal *clutching*.

Clutching can be avoided altogether by using a rate control device such as the *TrackPoint* [Zha95]. This may increase performance for long distance movements, but for shorter movements, where a *position control* device could be used without *clutching*, performance will suffer [DM94].

To preserve the benefits of medium-distance *position control* and still accommodate long movements without *clutching*, simple *hybrid position-rate control* techniques have been proposed [BH97, Edg]. But without any haptic feedback, the transition between position and rate mode is difficult to distinguish and the rate is difficult to control. Zhai found that elastic feedback is well suited for rate control [Zha95] and Dominjon et al. used elastic feedback for 3D *hybrid position-rate control* [DL*05]. However, their mapping function has trajectory and velocity discontinuities when transitioning from position to rate control, further highlighting the challenges in designing a usable *hybrid* device.

RubberEdge is a 2D *hybrid position-rate* control technique using elastic feedback. Unlike past work, we designed a mapping function which enables a smooth transition from position to rate control. We conducted an experiment to evaluate its performance and explore the interaction of *CD gain* and *Pointer Acceleration*. We found that our *hybrid control* technique outperforms position-only control by 20% with a small input area similar to a laptop touch pad. We derive two predictive models for selection time with *position control clutching* and *hybrid position-rate control*. Finally, we discuss a class of *RubberEdge* devices (figure 2.1) and present our first physical *RubberEdge* prototype device for laptop touch pads, with initial user feedback.

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

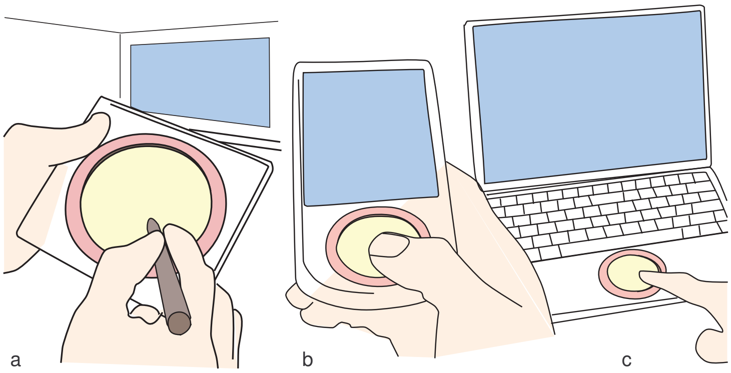


Figure 2.1: Design concepts for *RubberEdge* devices: a: handheld pen tablet for a large display; b: PDA with touch pad; c: laptop touch pad

2.3 RubberEdge

The *Bubble* that we have described in chapter 1 is a 3D position-rate control technique with elastic force feedback. However, when the *Bubble*'s straight forward mapping functions is adapted to 2D, it exposed trajectory and velocity discontinuities at the transition point affecting its usability. Through an analysis of Dominjon et al.'s [DL*05] straight forward mapping functions, we were able to determine the reason for erratic behavior when transitioning to rate control. This motivated our design for improved *RubberEdge* mapping functions with a smooth transition from *isotonic position control* to *elastic rate control*.

2.3.1 Problem of Straightforward Mapping Functions

Dominjon et al.'s [DL*05] mappings (equations 2.1, 2.2) introduce trajectory and speed discontinuities when transitioning from isotonic to elastic surrounding. In equation 2.1, the feedback force $\overrightarrow{F}$ is proportional to the distance between end effector P and the isotonic-to-elastic boundary N given spring stiffness k. The force direction is always radial with $\vec{r}$, the radial direction from the center of the isotonic circle to P. In equation 2.2, the input control rate $\overrightarrow{V}$ is a third degree polynomial with a scaling constant K. Dominjon et al.'s implementation set $k = 200N.m^{-1}$ and $K = 0.03N^{-3}.s^{-1}$.

$$\overrightarrow{F} = -k \cdot (P - N) \cdot \overrightarrow{r} \tag{2.1}$$

$$\overrightarrow{V} = K \cdot F^3 \cdot \overrightarrow{r} \tag{2.2}$$

This formulation introduces a trajectory discontinuity as long as the the end effector trajectory inside isotonic zone is not radial to the isotonic circle. The cursor will jump to the radial trajectory defined by equation 2.2 the moment it enters the elastic rate control zone, regardless of its initial path (figure 2.2). A speed discontinuity also occurs because according to equation 2.1, the initial force in the elastic zone will be zero, and thus the

© 2009 Tous droits réservés.

velocity will be set to zero with equation 2.2. Continuity of speed is important, since a noticeable drop could affect the pre-planned trajectory, impairing user performance [PA97].

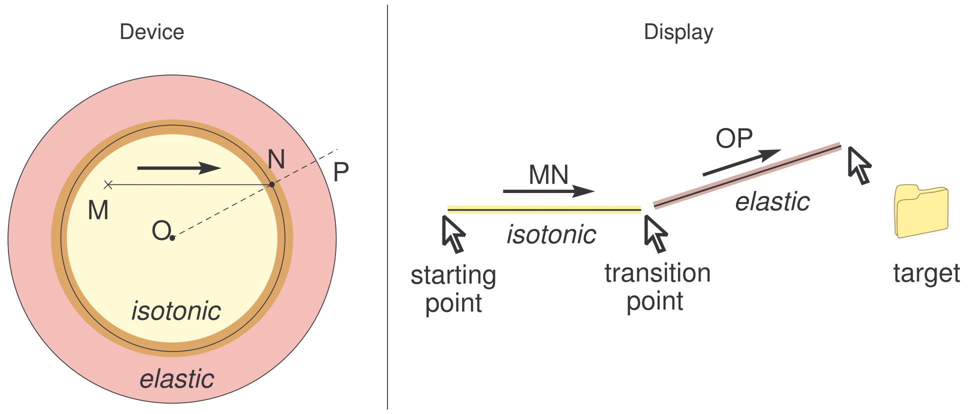


Figure 2.2: Trajectory discontinuity with straightforward mapping function. Left: Using the device to select a distant target, the user moves from position M to N in the isotonic zone, then transitions to the elastic zone; Right: On the display, the cursor will deviate from its trajectory of MN at the transition point, instantly changing to OP because the elastic zone always uses a direction vector radial from O through transition point N to an end effector P.

2.3.2 RubberEdge Mapping Functions

Our mapping functions enable a consistent trajectory by rotating and translating the isotonic-to-elastic boundary[10] after transition and we smooth the transition velocity by mixing pre- and post-transition velocities. At first it appears that simply using the same isotonic direction vector (MN in figure 2.2) as the direction vector $\vec{r}$ for the rate in equation 2.2 is the solution. However, the pre-planned trajectory direction in the isotonic zone is not always correct and the user may want to adjust it in the elastic zone. This could be done by saving the exit point N and translating the cursor according to NP. However any change in P produces an important variation in the cursor direction.

To create a consistent trajectory we translate and rotate the isotonic-to-elastic boundary zone as the user penetrates the elastic zone. We do this smoothly, by giving mass and inertia to the boundary zone using a simple physical simulation to align it with the isotonic direction vector. The intuition behind this technique is to consider how a real circular object, like a dinner plate, would rotate and translate when pulled by a string attached to its edge (figure 2.3). When the user exits the isotonic zone, the exit point N is saved. In the elastic zone, the vector from N to the end effector P gives the force direction applied by the user on the plate. By applying angular momentum, N rotates smoothly to N' and the force direction vector becomes radial to O, the center of the isotonic-to-elastic

[10] The isotonic-to-elastic boundary is not translate and rotate physically, we consider it is rotated and translated in algorithm level.

© 2009 Tous droits réservés.

Thèse de Qing Pan, Lille 1, 2008

boundary. Past user interface researchers have utilized similar physics-based rotation and translation functions, but for rotating graphical objects with direct manipulation [KCT05] and smoothly rotating or peeling back GUI windows [Bea01].

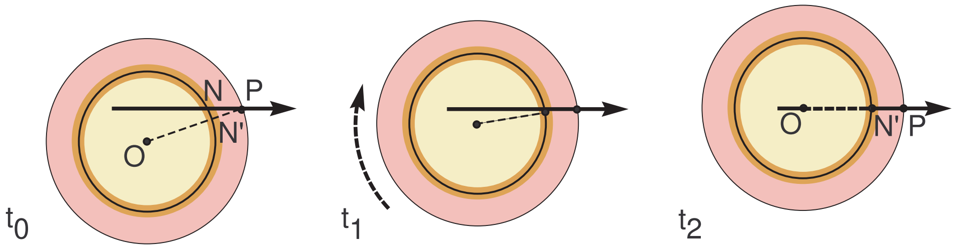


Figure 2.3: Rotation and translation with momentum over time: like pulling a dinner plate with a string.

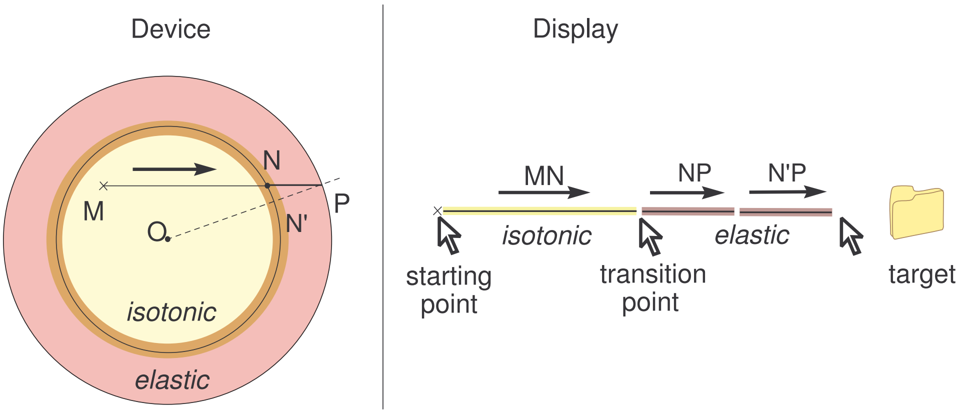


Figure 2.4: Continuous trajectory with enhanced technique: Left: Using the device, the user moves from M to N in the isotonic zone, then transitions; Right: On the display, the initial trajectory NP smoothly changes to $N'P$ by applying angular momentum.

The angular speed of the isotonic zone is computed by the theorem of angular momentum (equation 2.3). ω is the rotation vector of the isotonic zone and J is its moment of inertia with a friction term (μ) added to avoid instability. The translation is proportional to the vector NP. We found that using a mass of 1 Kg and a friction coefficient of $3*10^{-3}N.s.rad^{-1}$ smoothes out the trajectory nicely without the sharp direction changes.

$$J\frac{d\omega}{dt} - \mu\vec{\omega} = \overrightarrow{ON} \wedge \overrightarrow{NP} \tag{2.3}$$

To smooth the transition velocity, we mix the pre-transition velocity in the isotonic zone V_0 with the input control rate computed in the elastic zone. Equation 2.4 find the mixed velocity V_t where t is the time after the isotonic exit and A is a constant to adjust the mixing time ($A = 0.3s$).

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

Thèse de Qing Pan, Lille 1, 2008

$$V_t = (V_0 \cdot e^{-t\frac{t}{A}} + K \cdot NP \cdot (1 - e^{-\frac{t}{A}}))\frac{\overrightarrow{NP}}{NP} \qquad (2.4)$$

While iterating this design we found the technique worked well except when the displacement is tangent to the boundary circle. Then, the elastic force is near zero, making control difficult. However, the friction coefficient makes this occurrence rare.

2.4 Experiment

Since no previous work has demonstrated a benefit for *hybrid position-rate control*, we conducted an experiment comparing *RubberEdge* 2D *hybrid position-rate control* to pure *position control*. Note that after our theoretical analysis established that the 2D adaptation of Dominjon et al.'s unproven technique has control discontinuities, it can not be considered state-of-the art and an empirical comparison would be of limited value.We focused instead on a more general investigation of *hybrid position-rate control* since no previous work has demonstrated a benefit, and little work has investigated the effect of *clutching*. We included *Pointer Acceleration* as an experimental variable since it should reduce *clutching*, and could negate the benefit of *hybrid control*.

2.4.1 Goals and Hypotheses

Since the motivation for *hybrid position-rate* control is to eliminate (or at least reduce) *clutching*, *clutching* is a key factor in the experiment. We experimentally manipulated *clutching* by holding the device operating range constant and adjusting the distance from the target. Since the transfer function affects the device operating range, we included conditions for both constant *CD gain* and *Pointer Acceleration*. Unlike past work [JC90], we use the more aggressive Windows XP/Vista "Pointer Ballistic" function [Poi]. We give following hypothesis:

H1: The *hybrid* technique will outperform pure *position control* when there is *clutching*. *Clutching* with a *position control* device takes time because the cursor movement stops as the user recalibrates their position, whereas with the *hybrid* technique the cursor continues to move in the direction of the target. For long distances, we expect the inclusion of an elastic zone to be an advantage for the *hybrid* device in spite of the lower performance of pure elastic devices. This is because *hybrid control* still enables isotonic control for fine adjustment near the targets.

H2: *Pointer Acceleration* improves *position control* performance by reducing *clutching*. A dynamic transfer function uses high *CD gains* at high speeds which should increase the effective operating range and reduce the amount of *clutching*, but without hurting low speed precision.

2.4.2 Apparatus

To avoid device related confounding factors in the experiment, we simulated both 2D *position control* and 2D *hybrid control* on a *PHANToM Omni haptic device*. The *PHAN-*

© 2009 Tous droits réservés.

http://www.univ-lille1.fr/bustl

Thèse de Qing Pan, Lille 1, 2008

ToM uses a stylus connected to a force feedback armature to produce haptic feedback. By simulating both techniques with a single device, we were able to compare them without introducing extraneous intra-device differences such as ergonomics, size and sensitivity. The *PHANToM* also enabled rapid prototyping — we could iterate the *RubberEdge* technique and parameters with synthesized haptic feedback.

To ensure that *position control* performance is not adversely affected when using the *PHANToM*, we conducted a 4 participant pilot experiment comparing it to the mouse. We used 3 target distances (70, 140, 280 *mm*) and a constant *CD gain* of 2. With these settings, no *clutching* was needed by either device (we found constraining the maximum mouse operating range difficult, so did not compare it with clutching). A keyboard key was used for target selection, since the *PHANToM* and mouse have different buttons. We found mean movement times of 1.24s for the *PHANToM* and 1.23s for the mouse. Fitts's Law analysis gave similar regression coefficients: $T = -0.03 + 0.24ID$ for the *PHANToM* ($R^2 = 0.97$) and $T = 0.09 + 0.22ID$ for the mouse ($R^2 = 0.99$). This is consistent with previous mouse results [MO98]. Although not definitive, the results of this pilot bolstered our confidence that using the *PHANToM* would be comparing *position control* comparable to a mouse, perhaps the best performing *position control* device.

2.4.3 Simulating the Techniques on the PHANToM

For both techniques, the *PHANToM* stylus moves on a simulated haptic surface 1cm above the desk. The size of the isotonic area in each technique was constrained to a circle 40mm in diameter by simulating a vertical wall around the perimeter (figure 2.5). This size was selected to be similar to typical laptop touch pads. For the *hybrid control* technique, the elastic zone was accessed beyond the perimeter wall by pushing against a simulated radial spring with a stiffness of 60 $N.m^{-1}$. This setting was chosen after running a pilot experiment testing different stiffness values and it approximates the elasticity of a typical thick rubber band. For the *position control* technique, the simulated perimeter wall was rigid, and *clutching* was performed by lifting the pen above the simulated surface. To avoid instability when selecting a target with the button on the *PHANToM* stylus, participants instead pressed a keyboard key with their non-dominant hand.

Our experiment was conducted on a 3G Hz PC with dual 19 inch, 85 DPI LCD monitors, see figure 2.6. Our C++ software displays the stimulus at 60 Hz. The *PHANToM Omni* has a 450 DPI nominal resolution with 1000 Hz haptic rendering.

2.4.4 Task and Stimuli

The task was a reciprocal two dimensional selecting task, requiring participants to select round targets back and forth in succession. The positions of the targets were randomly pre-computed using the position of the previous target and the current distance. When participants correctly selected a target, the target disappeared and the next one appeared on the other side of the screen. If a participant missed a target, a sound was heard and an error was logged. Participants had to successfully select the current target before moving to the next one, even if it required multiple attempts. This prevented participants from "racing through the experiment" by clicking anywhere. To avoid using the edges to

© 2009 Tous droits réservés.

http://www.univ-lille1.fr/bustl

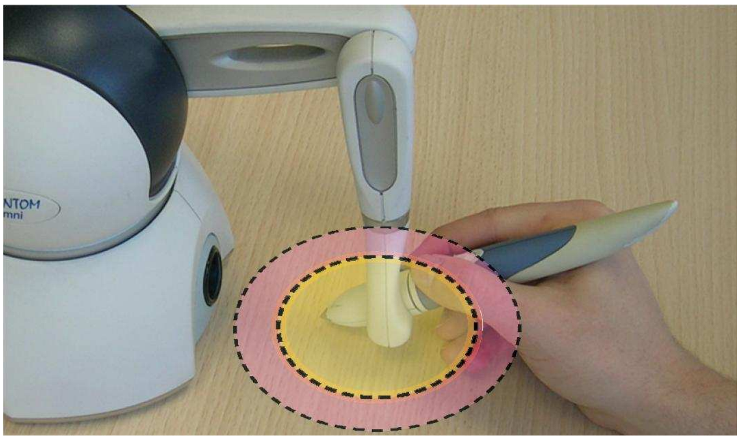


Figure 2.5: Simulating the *RubberEdge hybrid* technique with the *PHANToM* haptic device. A 2D simulated haptic surface constrains the pen movement and the elastic zone is created using force feedback.

assist in target acquisition, the cursor was not constrained to the bounds of the screen. Participants were encouraged to take breaks between sets of trials.

2.4.5 Participants

Eight people (5 male, 3 female) participated with a mean age of 26.3 ($SD = 1.5$). Three participants used Windows XP/Vista *Pointer Acceleration* exclusively, two did not, and the remaining used both.

2.4.6 Design

A repeated measures within-subjects design was used. The independent variables were *Technique* (*Position* control and *Hybrid control*), *Transfer Function* (*CG* — constant gain and *PA* — *Pointer Acceleration*), target *Distance* (D_L — 688mm, D_M — 344mm, D_S — 172mm), and target Width (W_L — 8mm, W_M — 4mm, W_S — 2mm). The nine *Distance-Width* combinations give five Fitts' indices of difficulty (*ID*) [Mac92] ranging from 4.5 to 8.4. We selected long distances to promote *clutching*, so our *ID* range is high. With short distances (and corresponding low *IDs*) the *Hybrid* and *Position control* techniques are equivalent since the elastic zone is not needed in the *Hybrid* technique.

For the *CG Transfer Function* we used a constant *CD gain* of 2 to encourage *clutching*. For the *PA Transfer Function*, we used the default Windows XP/Vista setting [Poi]. Using this setting, the *CD gain* increases continuously with the speed, from about 1.6 for low speeds to 7.3 for high speeds.

The presentation order of the 2 *Techniques* and 2 *Transfer Functions* was fully counterbalanced across participants. For each *Technique* and *Transfer Function* combination, participants completed a training period of approximately 5 minutes. Each *Distance-Width* combination was repeated 36 times with 4 *Blocks* of 9 *trials* each. *Distance-Width*

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

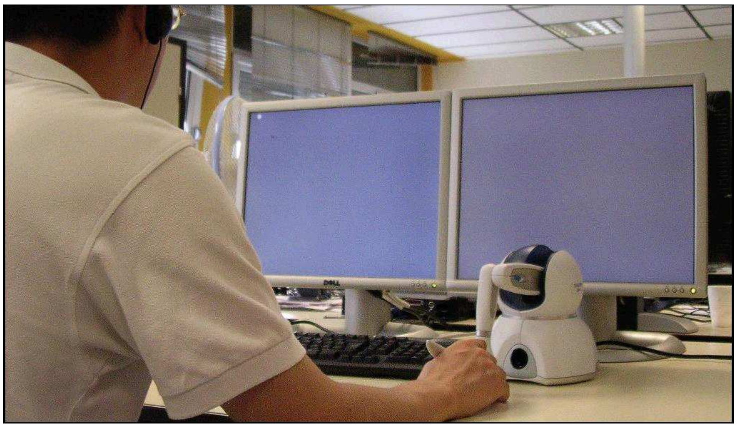


Figure 2.6: Experimental Apparatus: Simulating the RubberEdge *hybrid* technique and *position control* with a *PHANToM* haptic device

combinations were presented in ascending order of *ID* within a single *block* allowing participants to leverage repetitive, ballistic movements while steadily increasing task difficulty.

After all *blocks* were completed for a *Technique* and *Transfer Function* combination, a short questionnaire asked participants to compare it to the previous combination. At the end of the experiment, a final questionnaire asked for an overall ranking of the four *Technique* and *Transfer Function* combinations. The experiment lasted approximately 120 minutes.

In brief, the experimental design was: 8 Participants ×
2 *Techniques* ×
2 *Transfer Functions* ×
4 *Blocks* ×
3 *Distances* ×
3 *Widths* ×
9 repetitions
= 10,368 total trials.

2.5 Results

The dependent variables were movement time, error rate, and measurement of *clutching* and elastic zone usage.

2.5.1 Error Rate

Participants had an overall mean error rate of 1.5%, and a repeated measures analysis showed no significant effect of the different independent variables on error rate. In this type of experiment, a 4% error rate represents a good trade combination of speed and accuracy, so our lower error rate suggests greater emphasis on accuracy. As a result, movement times were somewhat higher and we computed the effective width for our

© 2009 Tous droits réservés.

Fitts' Law analysis [Mac92].

2.5.2 Selection Time

Selection time is the time from the beginning of the trial until the first target select attempt. Targets that were not selected on the first attempt were marked as errors, but were still included in the timing analysis (the analysis was run with and without error trials and the same significant effects were found with similar F and p values).

Repeated measures analysis of variance showed that the presentation order of *Technique* and *Transfer Function* had no significant effect on movement time, indicating that a within-participants design was appropriate. No significant effect for *Block* was found indicating that there was no learning effect present. There was a significant main effect for *Technique* on *selection time* ($F_{2,14} = 16.0$, $p < 0.005$) with the *Hybrid control* technique outperforming the *Position* control technique by 20.6%. As expected in a target selection experiment, there were also significant main effects for *Distance* ($F_{2,14} = 471.0$, $p < 0.0001$) and *Width* ($F_{2,14} = 231.0$, $p < 0.0001$) on *selection time*. The significant interactions for *Technique*×*Distance* ($F_{2,14} = 50.2$, $p < 0.0001$) and *Technique*×*Transfer-Function*×*Distance* (F2,14 = 9.7, $p < 0.017$) are perhaps most relevant. These show that the *selection time* increases with *Distance* at different rates given the *Transfer-Function* (figure 2.7). Pair-wise comparisons found no significant difference between the two *Techniques* for the smallest *Distance* D_S but significant differences for D_M ($p < 0.017$) and D_L ($p < 0.001$) with 16% and 29% improvements for *Hybrid control* over *Position control* respectively. The high *selection times* for distant targets with the *Position* control technique are due to heavy *clutching*, which we discuss in detail below.

Pair-wise comparisons revealed a significant difference between the two *Transfer Functions* for the *Position control* technique and D_L ($p < 0.032$). *PA* reduces the *selection time* by 7.5% compared to *CG*. It appears that participants were able to harness the higher speeds for distant targets and thus use higher *CD gains* to avoid *clutching*.

2.5.3 Fitts' Law Analysis

The significant interaction between *Technique*, *Transfer-Function* and *Distance* leads us to a *Fitts' Law* analysis. We aggregated the *Distance-Width* combinations for each *Technique* and *Transfer-Function* and computed the effective width since the error rate is not equal to 4% [Mac92].

Unlike many past studies, we found poor regression fitness suggesting that *Fitts' Law* may not hold in the presence of significant *clutching* or for a technique combining two different control mappings. The *index of difficulty* (*ID*) is expressed as a ratio between target distance and width, giving the same importance to each. Therefore, *Fitts' Law* predicts that any *Distance-Width* combination with the same *ID* will yield the same *selection time*. However, in looking at a plot of *ID* and *selection time* (figure 2.8), it appears that distance alone affects *position control clutching* or encourages *hybrid control* mode switching. We explore this somewhat surprising result in the discussion section.

© 2009 Tous droits réservés.

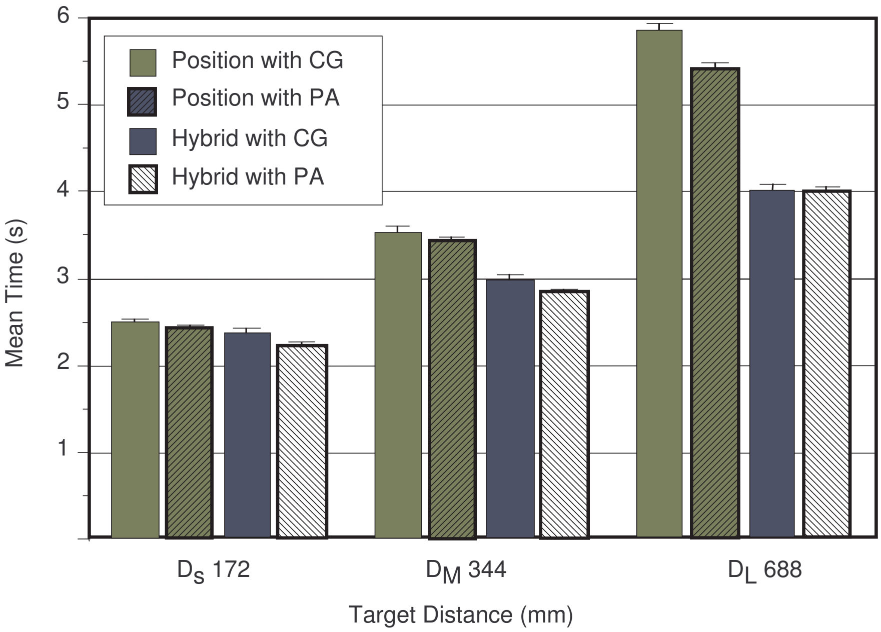


Figure 2.7: *Technique*×*Transfer Function*×*Distance* interaction on selection time (error bars 95% CI).

2.5.4 Usage of Clutching and Elastic Zone

To help characterize technique usage and to investigate possible explanations for the poor conformance to *Fitts' Law*, we analyzed the amount of time spent on *clutching* in *Position control* or on using the elastic zone in *Hybrid control*, as well as the number of invocations for each. Since cross-technique statistical comparisons of these two measures would not be meaningful, separate ANOVAs were used for each technique.

Technique	*a*	*b*	r^2
Position with *CG*	-3.9	1.1	.73
Position with *PA*	-2.4	0.9	.70
Hybrid with *CG*	-1.0	0.6	.86
Hybrid with *PA*	-1.0	0.6	.74

Table 2.1: *Fitts' Law* regression values for *Technique* and *Transfer Function*: *a* is the intercept of the regression line, *b* is the slope, r^2 is the fitness.

Clutch Time and Elastic Zone Time

Clutch time is the total time in which the stylus was lifted during a trial, and *elastic zone time* is the time spent outside the isotonic zone. For the *Position* technique, there is a significant main effect of *Distance* ($F_{2,14} = 165.5$, $p < 0.0001$) showing that *clutch time* increases as *Distance* increases. Pair-wise comparisons revealed that the total *clutch time*

© 2009 Tous droits réservés.

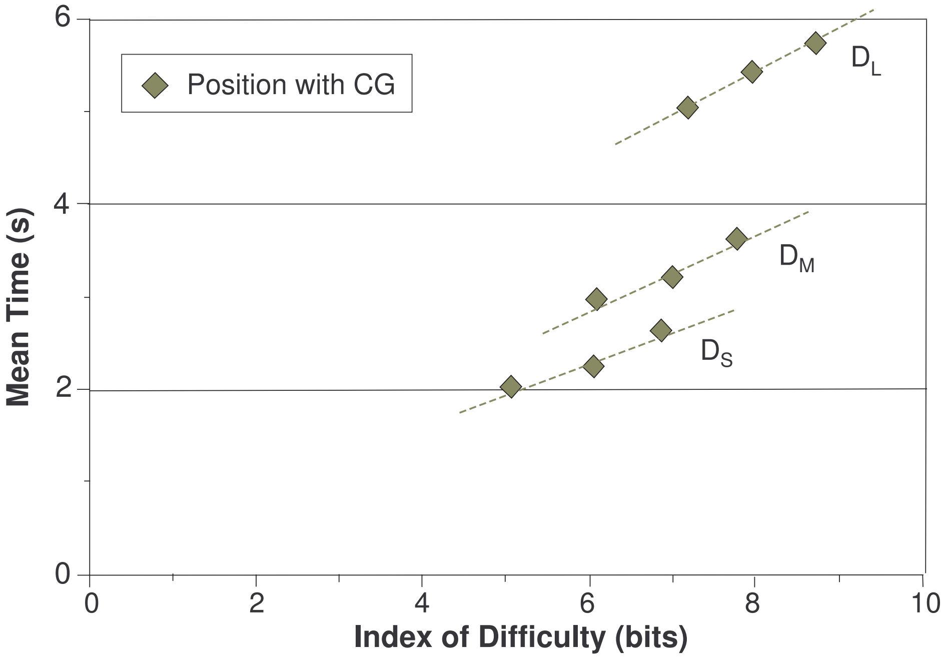


Figure 2.8: Selection times given *Distance-Width* combinations for the Position technique and Constant transfer function. Trend lines for *Distance-Width* groups are highlighted, suggesting *Distance* has a greater effect on *selection time*. Plots of other *Technique* and *Transfer Function* combinations are similar.

increased from 0.7s for D_S to 1.2s for D_M and 2.4s for D_L for *Position control* and *CG* (all $p < 0.0001$). Results for *PA* are similar (figure 2.9 left). A *Transfer Function*×*Distance* interaction ($F_{2,14} = 21.9$, $p < 0.0001$) and pair-wise comparison showed that *clutch time* is reduced by 10% with the *PA Transfer function* for the largest distance D_L ($p = 0.018$). For the *Hybrid* technique, we found a significant main effect for *Distance* ($F_{2,14} = 956.7$, $p < 0.0001$) showing that *elastic zone time* increases with *Distance*.

Clutch and Elastic Zone Invocations

Comparing the mean number of *Position clutch* invocations to *Hybrid* elastic zone transitions can help characterize technique usage. With 1.3 and 7.5 invocations respectively, we see that frequent *clutching* actions can be replaced often with a single transition to the elastic zone.

For the *Position* technique, a significant main effect was found for *Distance* ($F_{2,14} = 430.6$, $p < 0.0001$), but the *Transfer Function*×*Distance* interaction ($F_{2,14} = 19.1$, $p < 0.0001$) is more relevant. Pair-wise comparison revealed that the number of *Position clutch* invocations for D_L is dependent on *Transfer Function*: 12.4 for *CG* and 11.6 for *PA* ($p = 0.013$) (figure 2.9 Right). *Hybrid* invocations were near 1 regardless of *Distance* or *Transfer Functions*.

Table 2.2 compares the actual *clutching* invocations with an ideal minimum number calculated by dividing the floor of target distance by maximum device operating range

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

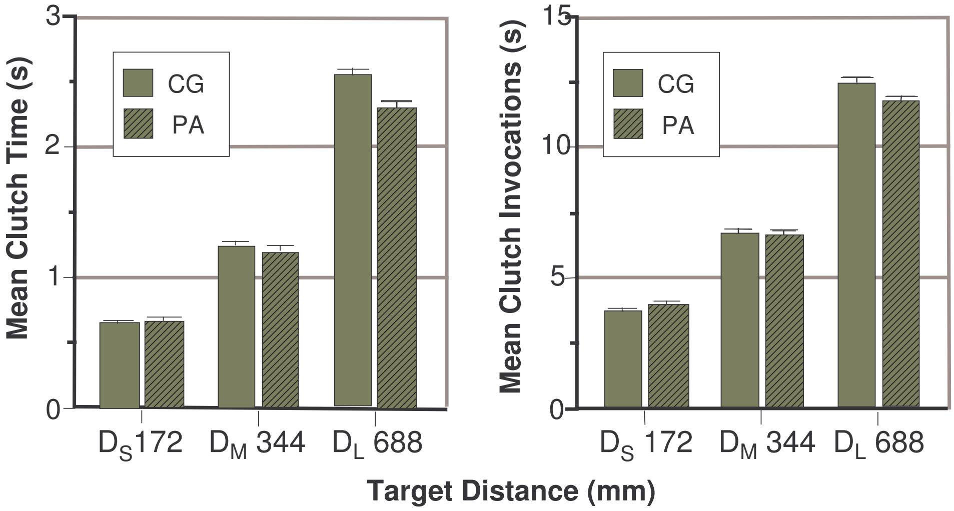


Figure 2.9: *Position* Control *Transfer Function*×*Distance* interaction on. Left: clutch time; Right: clutch invocations

(accounting for *CD gain*). We can see that the ratio remains constant around 0.75, indicating that participants did not use the maximum available operating range for each clutch.

Distance	D_S	D_M	D_L
Ideal	3	5	9
Actual	3.75	6.7	12.5
Ratio	0.8	0.75	0.72

Table 2.2: Comparison of actual and ideal numbers of invocations used in the *Position* technique.

2.5.5 User Feedback

Overall preference for *Technique* was split. Those preferring *Hybrid* found it faster for long-distance targets and disliked the repetitive *clutching* motion with *Position*. Those preferring *Position* found it acceptable to use conventional *clutching* for short and medium distances, and felt that it was difficult to accurately exit the elastic zone and fine-tune their selection. Our observations during the experiment reinforced this last comment: when participants exited the elastic zone having undershot the target, any movement back in the direction of the target transitioned them back to the elastic zone. This left no room for isotonic movement to fine-tune the selection. One way to address this is to allow conventional *clutching* in the isotonic zone of the *Hybrid* technique. Later, we present a prototype device which does exactly that.

© 2009 Tous droits réservés.

2.5.6 Discussion

Our experiment confirmed our two hypotheses and illustrated a negative performance impact with *clutching*.

Our results confirmed hypothesis *H1*: a *hybrid position-rate control* technique has a performance advantage over pure *position control* when faced with significant *clutching*. Our experimental design intentionally provoked *clutching*, and overall we found *Hybrid control* improved performance by 20.6%. Specifically, *Hybrid control* improved performance by 16% and 29% over *Position* control for D_M and D_L respectively. Increased *clutching* with *position* control appears to be the reason. Clutch times went from 0.7s at D_S to 1.2s at D_M and 2.4s at D_L with the *CG* transfer function. One reason why *clutching* may be slower than *Hybrid*, is the number of invocations required. Users had to clutch more than 12 times to reach D_L.

We confirmed hypothesis *H2*: *Pointer Acceleration* reduces *clutching* with *Position* control. The effect is somewhat slight; we saw it only at the longest distance where *PA* clutch time was 10% lower, requiring an average of 11.6 invocations compared to 12.4 for *CG*. This suggests that participants are able to utilize the high *CD gain* levels with quick ballistic movements. Our results differ from past researchers who did not see an effect [JC90]. We attribute this to using a more aggressive *Pointer Acceleration* function and a task requiring significant *clutching*.

The experiment demonstrates the negative impact of *clutching* on user performance and shows that *selection times* with significant *clutching* do not conform to *Fitts' Law*. With *clutching*, task difficulty appears to be primarily dependent on distance, rather than the ratio of distance to width as in *Fitts' Law*. Past researchers have not reported this [JC90, Mac92], perhaps because their experiments did not promote significant *clutching*.

2.6 Performance Models

We developed two formal models to predict *position control* performance with *clutching*, and performance with a *hybrid position-rate control* technique.

2.6.1 Clutching Model

We base our model for *position control* movement with *clutching* on a two-part, idealized movement. In the first part, the user clutches several times to bring the cursor within a “clutch-free” distance to the target. In the second part, the user completes the movement without *clutching*, and selects the target. The total movement time T is the sum of the time for the first part T_1 and the second part T_2:

$$T = T_1 + T_2 \tag{2.5}$$

T_1 is dependent on the number of clutches N and the time for each clutch T_C, which we assume to be constant. We also assume the time the cursor is engaged between two clutches to be equal to T_C, hence the factor 2:

$$T_1 = 2 \cdot N \cdot T_C \tag{2.6}$$

© 2009 Tous droits réservés.

N is dependent on the target distance D (in mm) and the effective device operating range d_e:

$$N = \lfloor \frac{D}{d_e} \rfloor \tag{2.7}$$

The effective operating range of the device, d_e is calculated from the physical device operating range d (in mm), the *CD gain* (CD), and a corrective parameter c. Recall that our experimental results showed that in practice, only a portion of the operating range is actually used:

$$d_e = c \cdot d \cdot CD \tag{2.8}$$

The movement time in the second part, T_2, can be calculated using *Fitts' Law* [Mac92] with the remaining target distance D_2 and Fitts' device parameters a_i and b_i.

$$T_2 = a_i + b_i \log_2(\frac{D_2}{W} + 1) \tag{2.9}$$

Where D_2 is equal to:

$$D_2 = D - N \cdot d_e \tag{2.10}$$

By substituting equations 2.6 to 2.10 into equation 2.5 and simplifying, we have a model which accounts for *clutching* when predicting target *selection time*:

$$T = 2NT_C + a_i + b_i log_2(\frac{D - N \cdot c \cdot d \cdot CD}{W} + 1) \tag{2.11}$$

2.6.2 Hybrid Control Model

A similar idealized model exists for a *hybrid position-rate control* technique. Similar to *clutching*, the movement has two parts (equation 2.5). T_1 is the time to move to the isotonic-elastic boundary and T_2 is the remaining time in the elastic zone to the target. Note that if the target is within reach of isotonic movement, then $T_2 = 0$ and T_1 can be predicted by *Fitts' Law* with the parameters a_i and b_i of an isotonic device. Otherwise, we suppose the movement distance in the first part is equal to the effective device operating range and $T_1 = T_C$. T_2 can then be determined using *Fitts' Law* with parameters a_e and b_e for an elastic device, and the remaining distance D_2 to the target:

$$T_2 = a_e + b_e \log_2(\frac{D_2}{W} + 1) \tag{2.12}$$

Where D_2 is simplified:

$$D_2 = D - CD \cdot d \tag{2.13}$$

By substituting equations 2.12 and 2.13 into equation 2.5 and simplifying, we have a model for hybrid movement predicting target *selection time*:

$$T_a = T_C + a_e + b_e \log_2(\frac{D - CD \cdot d}{W} + 1) \tag{2.14}$$

© 2009 Tous droits réservés.

Thèse de Qing Pan, Lille 1, 2008

2.6.3 Comparison to Experimental Results

To test the validity of our models, we compared their predicted *selection times* with the results of our experiment. The following model parameters were used $d = 40mm$, $CD = 2$, $c = 0.75$, $T_C = 0.2s$ (T_C from our experiment). The *Fitts' law* parameters were from the literature $a_i = 0$, $b_i = 1/4.5$ [MSB91], $a_e = 0$, $b_e = 1/2.0$ [DM94]. Considering the simplicity of the model, we found good fitness (figure 2.10). The root mean square (RMS) is 0.4s for *clutching* and 0.2s for *hybrid* (The RMS for *Fitts' law* are respectively 1.1s and 0.6s). At D_S, the *clutching* model was 25% lower, likely due to the floor in equation 2.7, while the experimental data presents a mean value. For example, at D_S, the predicted number of clutches is 3, but the experimental data is 3.75.

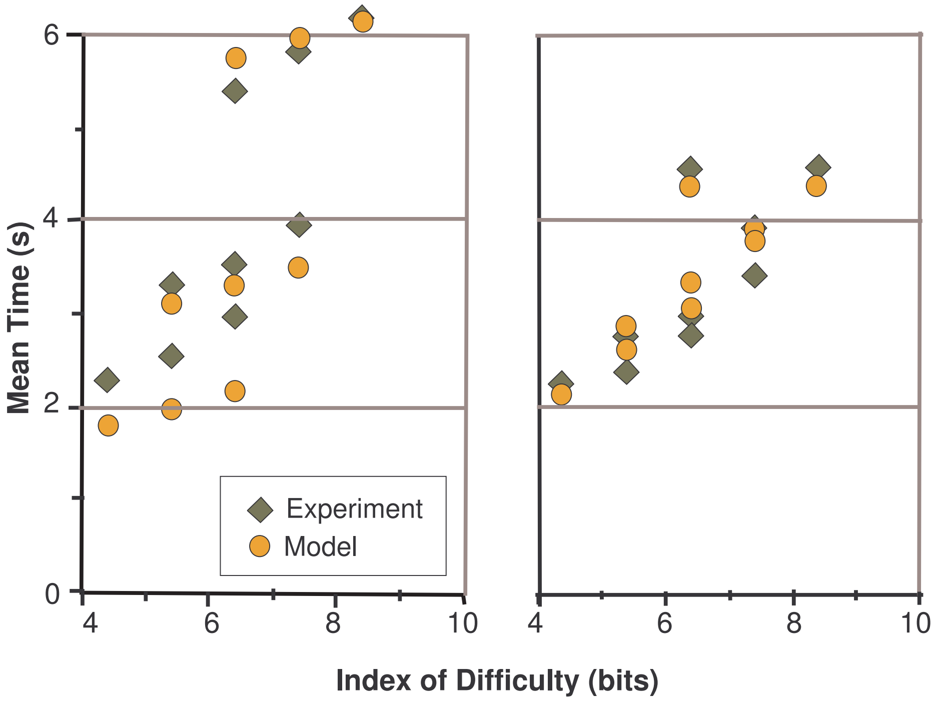


Figure 2.10: Predicted Model Time vs. Actual Time for: *Clutching* Model (left); *Hybrid* Model (right).

When is Hybrid Control Advantageous?

We can use the models to predict when *hybrid control* will have a performance advantage over *position control*.

For example, consider a laptop with a 38cm (15"), 1400×1050 pixel display, and 4cm touch pad. Using our theoretical model, *RubberEdge hybrid control* will outperform *position control* when targets are more than 30cm apart (nearly three-quarters of the maximum possible target distance) (figure 2.11 right). A second example is the HP iPAQ hx4700 PDA which has a 1cm touch pad and a 10cm (4") display. Here our model predicts an advantage for *hybrid control* above 5 cm (half the display distance) (figure 2.11 left).

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

Thèse de Qing Pan, Lille 1, 2008

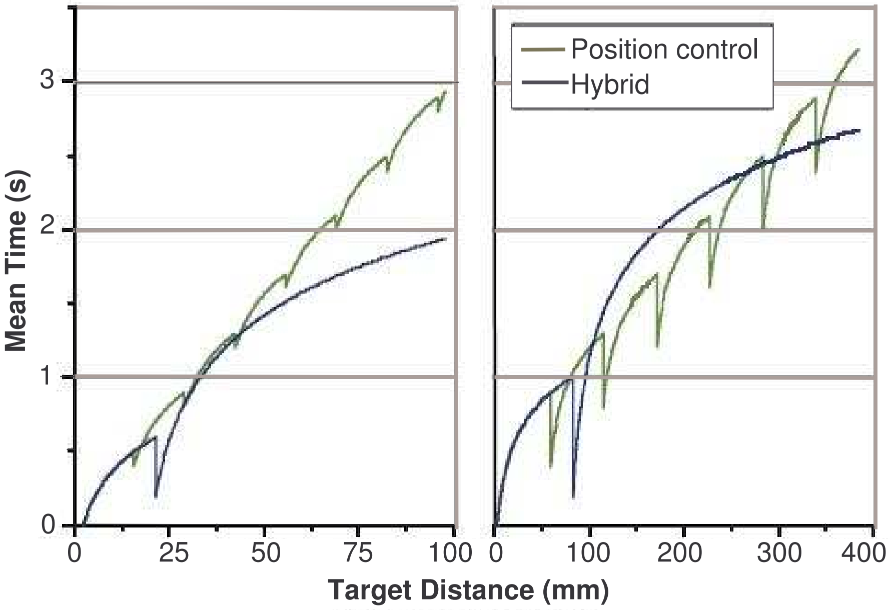


Figure 2.11: Theoretical Comparison for W=4mm: Touch Pad PDA: d =10mm, CD=2 (left); High Resolution Laptop: d=40mm, CD=2 (right).

The examples in figure 2.11 also illustrate a potential drawback with *hybrid control*. Depending on the operating range size, one or two manual clutches can be faster than using the rate control zone. In our model this is attributed to the lower performance of elastic devices, but there may be other factors not accounted for, such as a constant mental transition time. Regardless of the reason, it appears that a *hybrid* device should allow standard isotonic *clutching* as well as an elastic zone. This way, the user can develop their own optimized strategy for reaching near or far targets.

2.7 Prototype Device

With this more flexible *hybrid* model in mind, we built a device prototype that enabled a mix of isotonic *clutching* and elastic rate control. Our initial requirements were:

- cheap and compatible with current notebooks.
- support for high resolution absolute input to measure the elastic zone penetration and compute cursor velocity.
- support for relative position input and *clutching* in the same way as an existing device.

We found that modifying a standard laptop touch pad fulfilled these requirements. Creating an elastic zone with the right feel and stiffness similar to the *PHANToM* required

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

some trial and error. We experimented with elastic materials like latex gloves, balloons, and elastic fabrics mounted on different types of frames. Eventually, we converged on a simple design using a $1mm$ thick plastic frame cut from a old phone card. The frame has a $40mm$ hole with a plastic ring suspended by four rubber bands for elastic feedback (figure 2.12). The ring is $36mm$ in diameter leaving $2mm$ for elastic movement. Plastic lets the ring slide easily on the touch pad surface and has just enough tactile feedback to define the boundary of the $30mm$ isotonic zone. We would have preferred creating a larger isotonic zone, but the borders of the frame had to support the elastic force. Adding physical constraints in this way is reminiscent of Wobbrock et al. EdgeWrite [WMK03].

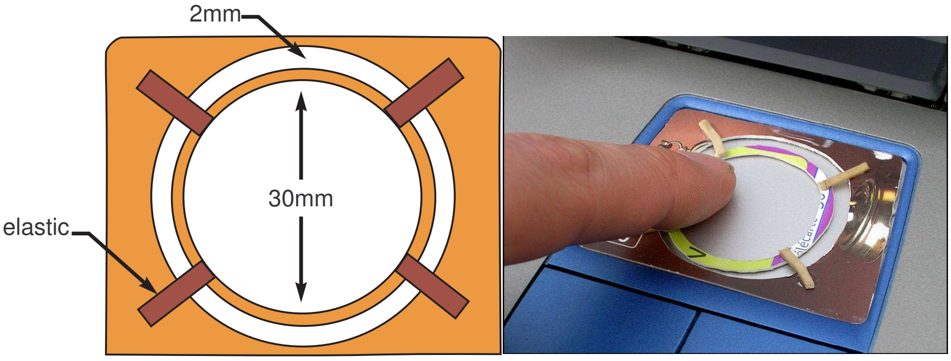


Figure 2.12: *RubberEdge* prototype device: Schematic (left). implementation (right).

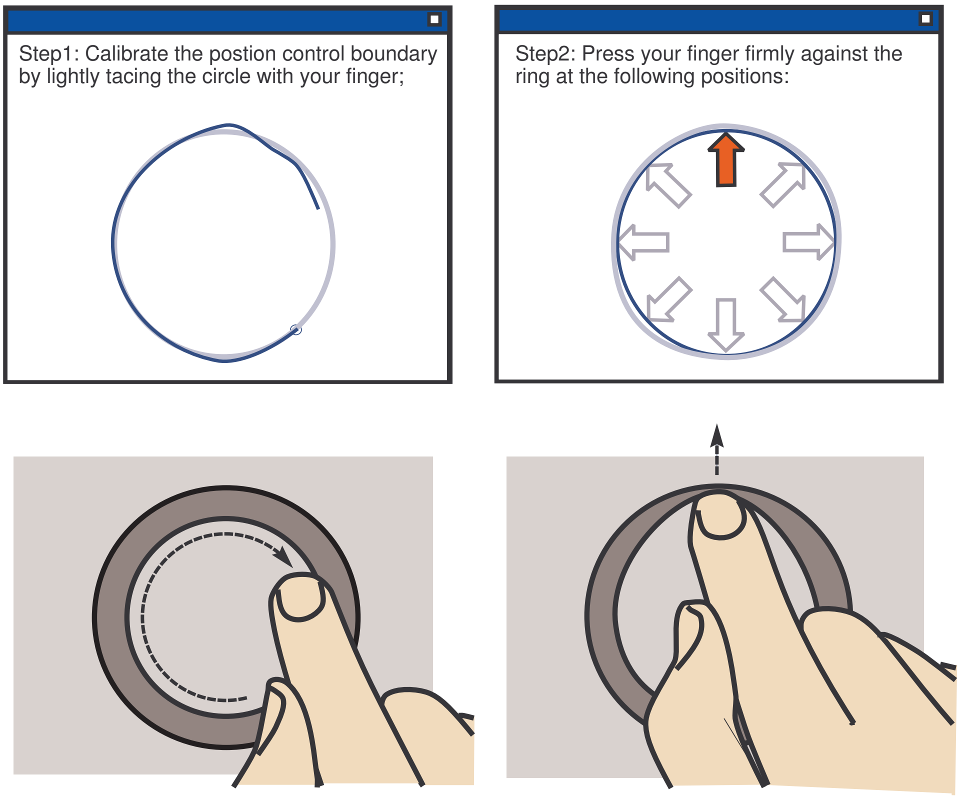
Our driver uses the Synaptics SDK [Syn] to measure the absolute finger position at 2000 DPI. In the isotonic zone, the cursor behaves like a standard Windows touch pad. When the finger enters the elastic zone, we transition using the *RubberEdge* mapping functions and compute the cursor velocity (equations 2.3 2.4). Our driver works like a standard Windows pointing device with any application.

Early designs revealed that isotonic-elastic boundary accuracy is critical since there is only $2mm$ of movement. Imperfections in our fabrication and the non-uniform way in which a finger contacts positions around the ring led us to develop a two-step calibration (figure 2.13). First, the boundary is defined by tracing around the perimeter of the ring. Then, to calibrate the maximum force (penetration distance) in each direction, the user pushes into the elastic zone at eight radial positions. We interpolate between these measurements when computing elastic rate control.

2.7.1 Initial Evaluation

We ran a pilot study with four participants to gather initial feedback about our prototype. We targeted people with touch pad experience since we were interested in the usability of our device, not touch pads in general; 3 participants used a touch pad daily and the fourth occasionally. All were right-handed. For approximately 20 minutes, participants used the device with common Windows tasks: file browsing, viewing PDF documents, painting, and web browsing. The tasks included selecting, window scrolling and steering through menus.

© 2009 Tous droits réservés.



Step1: Calibrate the postion control boundary by lightly tacing the circle with your finger;

Step2: Press your finger firmly against the ring at the following positions:

Figure 2.13: Two Step Prototype Calibration addresses caused by finger angle and fabrication: Left: Calibrating the boundary of the isotonic zone by tracing the finger clockwise around the perimeter; Right: Calibrating the maximum force by pushing the finger into the elastic zone at eight radial positions.

To grow accustomed to the device's reduced operating range, participants used only the isotonic zone with the rate control disabled for the first 2 minutes. We then enabled the elastic zone using a generic calibration, and gave no explanation or instructions. Participants immediately grasped that the cursor moved in two different ways depending whether you were pushing into the ring. The most difficulty was with elastic rate control: participants would at first overshoot the target, then sometimes overcompensate with hesitant and slow rate control. Past researchers have found that elastic rate control has a steep learning curve [Zha95]. Two of the participants used overshooting as a kind of strategy: in the elastic zone, they shot the cursor as fast as possible past the target, then moved back to the target under isotonic control. However, after more practice, participants generally moved the cursor more accurately and with less hesitation using the elastic zone. Overall, participants said they liked using rate control for continuous movement of the cursor over far distances and appreciated the ability to use the isotonic zone for tasks like drawing.

© 2009 Tous droits réservés.

http://www.univ-lille1.fr/bustl

2.8 Conclusion

RubberEdge hybrid position and rate control enables users to reach distant targets without *clutching*, yet still maintains benefits of *position control* for precise movements. Our mapping functions eliminate trajectory and velocity discontinuities when transitioning from *isotonic position control* to elastic rate control. The results of our controlled experiment found that *hybrid control* outperforms pure *position control* by 20% when there is significant *clutching*. This advantage is in spite of our related finding that a *Pointer Acceleration* transfer function will decrease *clutching*. We present theoretical performance models for *position control clutching* and *hybrid position rate control*, enabling designers to determine when *hybrid control* is beneficial. Based on our experimental and theoretical investigations, we developed a *RubberEdge* hybrid device for laptops which revealed design considerations such as construction, material, and calibration. With promising initial user feedback, we plan to further iterate our current prototype and investigate other types of *RubberEdge* hybrid devices.

© 2009 Tous droits réservés.

Thèse de Qing Pan, Lille 1, 2008

© 2009 Tous droits réservés.

http://www.univ-lille1.fr/bustl

Chapter 3

Manipulation and navigation device: *Haptic Boundary*

Contents

© 2009 Tous droits réservés.

http://www.univ-lille1.fr/bustl

Thèse de Qing Pan, Lille 1, 2008

3.1 Résumé

Dans ce chapitre, nous présentons *Haptic Boundary*, technique conçu pour les manipulations d'objets et la navigation en environnement virtuel pour les ordinateurs de bureau.

Pour l'environnement virtuel en 3D, il n'y pas encore de système d'interaction largement accepté qui permette à la fois la manipulation des objets et l'exploration de l'environnement. Les techniques basées sur les widget décomposent l'interaction en actions séquentielles, ce qui augmentent de charge cognitive des utilisateurs. Certains périphériques d'entrée permettent aux utilisateurs d'effectuer différentes interaction et basculent entre les différents modes de contrôle avec une méthode explicite, ceci apporte différentes réponses à une même entrée, ce qui augmente la charge cognitive des utilisateurs. A priori, les interfaces bimanuelles peuvent fournir de meilleures performances, mais les déplacements asymétriques des deux mains qui fournissent une bonne performance, ne sont pas toujours respectés dans les systèmes de bureau 3D.

Haptic Boundary permet une manipulation d'objets intuitive avec un contrôle en position à l'intérieur de la zone isotonique, et la manipulation du point de vue avec un contrôle en vitesse lors de l'interaction avec la paroi de la zone isotonique. *Haptic Boundary* est basée sur le *BubbleCam*, et l'a amélioré en permettant la rotation de la caméra et en fournissant des retours d'effort plus riches. Ces retours d'effort permet un changement implicite entre les modes de contrôle.

Nous expliquons un algorithme qui fournit les relations entre l'effecteur physique et le curseur 3D dans l'environnment virtuel. Cet algorithme calcule la position du curseur, avec la position de l'effecteur et la caméra controlée via *Haptic Boundary*.,

Nous examinons deux types de retour d'effort: une force simulant un mur et une solution "collante". Après avoir analysé leurs avantages et inconvénients, nous décidons d'utiliser la force simulant un mur pour la translation pure et la solution "collante" pour la rotation et la translation.

Nous étudions ensuite la taille et la forme de la zone isotonique par deux expériences. La taille est fixée à une dimension d'environ 40mm * 40mm * 40mm. Basé sur cette taille, la forme choisie est un semi-ellipsoïde avec plafond et plancher. La zone isotonique a un espace utilisable suffisamment grand, sa surface convexe et trois plans avantagent le contrôle vitesse avec la force simulant le mur et la force collante.

Au début, *Haptic Boundary* a étbasé sur des principes physiques de mécanique, mais après l'application sur la caméra, nous avons observés des rotations de caméra non désirées ainsi qu'une certaine difficulté de se déplacer vers l'avant. Nous avons utilisé ensuite un vecteur unitaire (0,0,-1) pour calculer la rotation. Pour permettre la rotation de la caméra autour d'un point fixé et la translation latérale, nous déterminons un vecteur dans la base (1,0,0), (-1,0,0) et (0,0, -1) comme direction de translation selon l'emplacement du point collant et du désir de l'utilisateur. Ce vecteur est aussi utilisé pour calculer la rotation. Nous distribuons la force appliquée par l'utilisateur entre la rotation et la translation. Nous présentons ensuite un système d'interaction, simple et complet, dans lequel *Haptic Boundary* est mis au point pour produire contrôler la caméra. Nous présentons quelque complements à ce système d'interaction. Nous avons conçu une interface avec des retours visuels pour les nouveaux utilisateurs, afin de faciliter la compréhension du *Haptic Boundary*.

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

Thèse de Qing Pan, Lille 1, 2008

Nous avons effectué une évaluation formelle à une tache d'assemblement une voiture virtuelle avec 3 participants. Elle montre que *Haptic Boundary* est de 50 % plus rapide que l'interface *uni-manuelle avec le changement explicite.* L'un des trois participants a préféré *Haptic Boundary*, les deux autres ont préféré l'interface bi-manuelle. Mais l'*Haptic Boundary* apporte plus de confort. L'observation que les participants n'ont pas profités pleinement de l'*Haptic Boundary* pour effectuer la rotation autour de la voiture, nous a fait ajouter un mode d'*inspection en orbite* avec *Haptic Boundary* afin de faciliter cette rotation.

3.2 Introduction

Most of techniques discussed in chapter 1 are devoted to one task such as orienting an object, positioning an object or navigation. A usable interaction system should provide interaction techniques for both object manipulations and view manipulations.

Some systems do not provide view manipulation, instead, they allow rotating an object to view it from different angle using orienting techniques such as Arcball [Sho92]. It works in the systems containing only one central object or a group of objects gathered at the scene center, frequently seen in scientific visualization and CAD systems. The "*model inspection*" in Bornik's volumetric medical data manipulation interface [BB*06] is an example. However, another essential navigation style — environment exploration can not be performed though object orienting.

A versatile technique commonly used in interaction systems is *widget*.The DIVE VR system [CH93], VRML browser Community Place and VRweb as well as the Haptic-GeoZui3D [KW03] (shown in figure 1.24) are such systems. *Widget*-based techniques decomposed interactions into sequential actions, which can burden users with heavier cognitive process.

A user-friendly interaction system requires delicate designs not only in software but also in hardware. The input device is the foundation of interaction metaphors and also a key factor for the user performance . A number of studies focused on input devices. Steed [SS95] proposed a 5 DOF *Desktop Bat* (3DOF elastic torque + 2DOF isotonic position) for both object and view manipulations. The two tasks are performed in a similar way. Pressing the buttons embedded in the device body switches between the object and view manipulations. This "multi-mode + switch" style also increase users' mental load [ZK*99]. A possible awkward situation is that you move the bat to reposition the cursor, but the cursor does not respond, instead, the camera is moved. Then you realize that you are not in the right mode, so you press the switch button, and then you have to redo what you have done. Although this situation become less frequent after a longer-term training, the multi-mode + switch combination is still an obstacle to smooth interaction. It causes inconsistent response to the same input. Consistent input / output mapping make the human information behavior becoming an "automatic process" which requires little central capacity, attention or effort. While the inconsistent input / output mapping make human behavior became a "controlled process" which requires effort, attentive resource and central capacity [SS77a, SS77b].

Noyo proposed by Simon et al. [SD04] is a joystick device with a 6 DOF elastic rate

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

controller designed for travel and object manipulation. The author said "pushing and twisting on the elastic controller cap result in viewer or object motion in the same absolute direction". It seems that the object is attached to the view and they are manipulated together. They did not explain any pure object manipulation in their works.

Two handed input devices such as *Cubic Mouse* [FP00], *YoYo* [SF03], *SquareBone* [FH*06, HCP05] were proposed for 3D interaction. According to Guiard's kinematics chain model, the non-dominant (ND) hand provides the frame of reference for the Dominant (D) hand motion. In Huckauf's extended docking experiment [HCP05], the ND hand holding part is used to navigate, while the D hand holding part is used for object manipulation. With *Cubic mouse*, the cube body with an embedded track sensor serves as the ND handed navigating controller. It has been designed to be used in exocentric navigation (moving the world)[11], according to Huckauf. He also reported that exocentric navigation[12] is faster when presenting the manipulated object in front of a black background compared to being presented inside a room. The rotation of room disorients users. The better performance with a black background can be explained by the fact that the user feels that he is manipulating the object not the world, because he only sees the object. The limit of applicability of exocentric navigation affects the applicability of *Cubic mouse* to certain extend. Another imperfection of *Cubic mouse* is that pushing, pulling and rotating three rods with the D hand separately control the motions along three principal axes, which may increase the complicity of the tasks requiring integrated movement, such as 3D positioning and orienting, due to the incompatibility between the integrality of DOF of the tasks and the separability of input device[JS*94]. The *YoYo* and *SquareBone* are very similar devices. They have two elastic sensors locating at the two ends of tube shaped body with an embedded 6 DOF tracker, held by two hands. The ND handed elastic sensor controls the navigation, the D handed elastic sensor controls the object manipulation. Both *YoYo* and *SquareBone* outperformed the *Cubic mouse* in Huckauf's extended docking experiment [HCP05]. *YoYo* and *SquareBone* are also based on hybrid position-rate control conception. The elastic controllers were used for most interactions which may be less accurate for fine positioning compared to isotonic position controller; although the embedded tracker also compensate for some interactions but requires extra input actions.

In some systems, each hand holds a device. Similarly, the ND hand manipulates the view, and the D hand manipulates the objects. For object manipulation, a pointing device such as stylus or mouse is commonly used. The ND handed device is more diverse. 3-Draw[SRS91], WIM [SCP95]. and Haan's VR system [HKP02] used a board shaped device. Hinckley's medical data visualization system used a rubber head prop [RGK94]. Balakrishnan's unimanual and bimanual comparison experimentation [BK99] used mice. The *ReachIn* Technology [Rea] use *SpaceMouse* held in ND hand to control the scene rotation, and a *PHANToM* in D hand for object manipulation. According to Guiard's *Kinematic Chain Model*, this view-object interaction should be performed through asymmetric motions of two hands. However in desktop system it is not always respected, unintentionally. With each hand controlling one device supported by desk, they work

[11] see section 1.7.6 in chapter 1 for details of exocentric navigation

[12] see section 1.7.6 in chapter 1 for details of egocentric navigation

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

independently, which easily result in less asymmetric motion between two hands. Here is an example, in the docking experiment of Balakrishnan [BK99], the result didn't show a significant difference in average completion time between the unimanual and bimanual interface. Each-mouse-in-each-hand may unintentionally encouraged the parallel hand motions.

As we discussed above, object orienting techniques does not satisfy the VE exploration requirement; the *widget*-based techniques decomposed the interactions, make them more complex; one handed input devices with explicit switch use inconsistent input / output mappings requiring more efforts of users to perform interactions. The elastic sensor devices are not suited for fine positioning task; Desktop bimanual interface with one device in each hand should perform well in both object and view manipulations, but its performance is sensible to the parallelism degree of two hand motions.

To address these problems, we *Haptic Boundary* which is designed for the one-hand input device which allows both object manipulations and VE explorations for desktop systems. Touching the boundary of its workspace performs an implicit switches between different interaction modes. The force feedback input device is used to simulate a workspace with a haptic boundary. The object manipulation is performed intuitively inside this workspace using isotonic position control[13]; effective view manipulation is performed using elastic rate control [14] when the end effector reaches the boundary. The *Haptic Boundary* provides different force feedbacks corresponding to different camera movements. The user feels this force and is aware of in which mode they are using. We believe that the implicit switch, different user actions and different force feedbacks for view and object manipulation will reduce the user's mental load. Additionally, unlike the technique which fixed the cursor relative to the camera, *Haptic Boundary* allows the cursor to move within the virtual workspace in front of the camera. Our technique is based on *BubbleCam* [DL*05] technique and improves it by adding the view orientation and by providing more rich force feedback.

In the following sections, we will firstly give the definitions which will be used after. We then explain our *Haptic Boundary* technique including the property of its boundary, its isotonic zone, implementations as well as the visual feedback for novice user. In section 3.5 we describe a car assembling experimentation involving object and camera manipulations to evaluate the user performance of *Haptic Boundary*. Following the analysis of this experiment, we introduce the orbit inspection functionality to *Haptic Boundary* to enhance its usability in object-centered applications.

3.3 Definition, Terminology and Convention

To clear our explication, we give some definitions and conventions of the vocabulary to be used in the following sections, to avoid confusing with their meanings in other literatures.

- *Physical Workspace* is the space in which the end effector of input devices can move. For example, the desktop is the workspace of computer mouse; the workspace of

[13] see section 1.5.2 for details about isotonic position control

[14] see section 1.5.2 for details about elastic rate control

© 2009 Tous droits réservés.

Thèse de Qing Pan, Lille 1, 2008

PHANToM is the 3D space that its stylus can reach.

- *End Effector* is the last link in a joint-and-link chain held by user, to interact with the system. *PHANToM*'s end effector is its stylus. Its position and orientation are measured by build-in sensor or calculated by driver level software and then sent to the application in real time. Its absolute or relative position and orientation are used to control the movements of objects in VE, eg. the cursor.

- *Cursor* is the virtual representation of the position and / or the orientation of the end effector on the screen.

3.3.1 End effector System and Camera Cursor System

In this section we describe an algorithm which maps the end effector position in physical workspace onto that of the cursor in VE. It was used in several techniques such as [OL01] and also in our *Haptic Boundary*. First, we give some notations:

- End effector position in the physical workspace is noted as point P_e (figure 3.1 a).

- In VE, we have a virtual workspace corresponding to the physical workspace. Its center position in VE is noted as point C_w, its orientation is noted as quaternion Q_w.

- Camera position and orientation in VE are respectively noted as point P_c and quaternion Q_c (figure 3.1 d).

- Cursor position in virtual workspace are noted as point P_{pw} (figure 3.1 c); cursor position in VE is noted as point P_p (figure 3.1 d)

- The camera is at a fixed position P_{cw} relative to the virtual workspace. The orientation of the camera and that of the virtual workspace in VE are identical (see equation 3.1). P_{cw} is an offset between them (figure 3.1 c). P_{cw} could be predetermined and adjusted so that the cursor's movement in the whole virtual workspace can be seen.

$$Q_w = Q_c \tag{3.1}$$

We care about three items: the orientation and the position of camera in VE: P_c, Q_c and the cursor position in VE: P_p. They are the basic elements of the interaction system. Our *Haptic Boundary* technique will determine the first two elements about camera: P_c and Q_c. The virtual workspace is the link between the camera and the cursor. To calculate P_p, we should firstly calculate the orientation Q_w and position C_w of virtual workspace using equation 3.1-3.2.

$$C_w = P_c - Q_w \cdot P_{cw} \cdot Q_w^{-1} \tag{3.2}$$

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

For the cursor position, to have the consistency between the movement of the end effector and the one of the cursor in virtual workspace, the P_e is directly mapped into P_{pw} with a scale factor CD, see equation 3.3. Then the equation 3.4 is used to transform the cursor position P_{pw} in local virtual workspace into its global position P_p in VE.

$$P_{pw} = P_e \cdot CD \tag{3.3}$$

$$P_p = C_w + Q_w \cdot P_{pw} \cdot Q_w^{-1} \tag{3.4}$$

The last basic element P_p can be determined by combining all the equations 3.1 to 3.4The figure 3.1 shows the system of the end effector in the physical workspace and the camera cursor system in VE.

This algorithm discussed above is used to calculate the cursor position if given the position / orientation of the camera, which will be determined by *Haptic Boundary* that we will explain below.

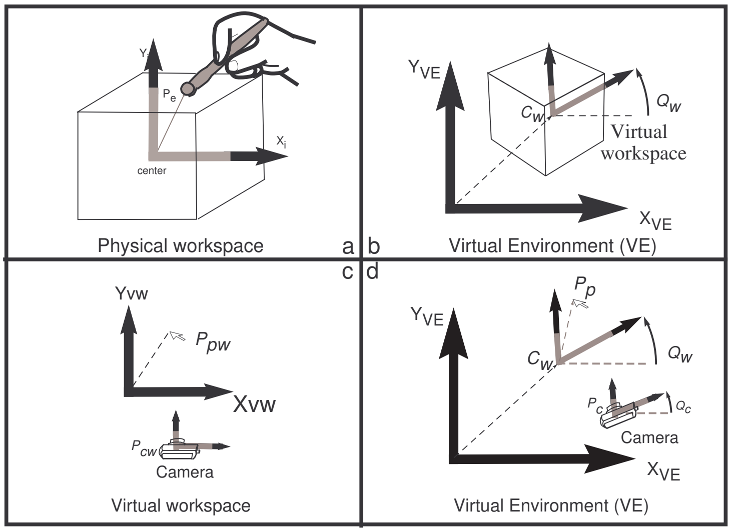


Figure 3.1: End effector and Camera Cursor system. (a) The end effector P_e in the physical workspace. (b) In VE, the virtual workspace locates at point C_w with an orientation Q_w ; (c) In the virtual workspace, camera locates at P_{cw}, the cursor locates at P_{pw}. (d) In VE, camera locates at P_c with its orientation Q_c which is identical to the one of the virtual workspace Q_w. Cursor position is represented as P_p

3.4 Haptic Boundary

In this section, we describe our *Haptic Boundary*. We begin with the global idea, followed by the details of the technique including the force feedback, workspace — isotonic zone, and its implementation. Then, we describe a simple usable system which use the *Haptic Boundary* combining with the algorithm presented above to control the camera and cursor. At last we describe the visual feedback for novice user to help them better understand the *Haptic Boundary*.

© 2009 Tous droits réservés.

3.4.1 Global Idea

Our *Haptic Boundary* combines isotonic position control and elastic rate control into one input device. It defines a workspace, represented as the cube in figure 3.2 left. We call it *isotonic zone*, because no force feedback is involved. Inside of it, position control is used. The effector directly manipulates the cursor. The virtual camera is anchored in the VE (figure 3.2 upper right). When the end effector reaches the boundary of isotonic zone, rate control is activated, which is used to translate and rotate the camera. At the same time, the user has certain haptic feedback, similar to the *extended camera in hand* [DC02] and *BubbleCam* [DL*05] technique, but in the *Haptic Boundary*, the feedback force can be a *sticky force* or a *wall force* according to the different camera motions. The figure 3.2 bottom shows a camera translation example.

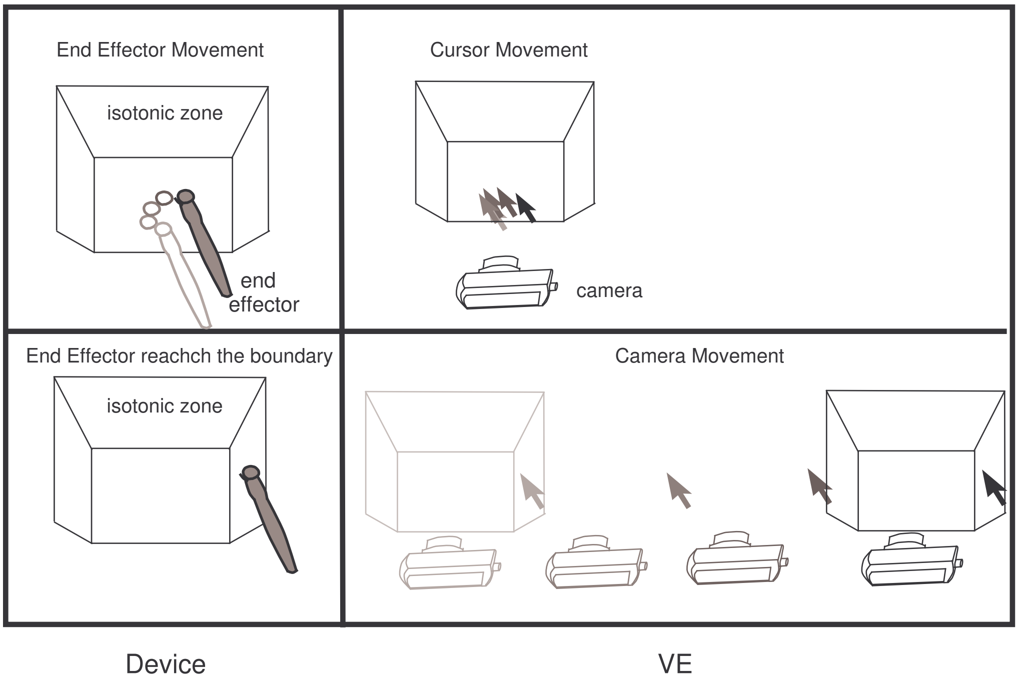


Figure 3.2: Top: the end effector moves inside the isotonic zone , the camera does not move, the cursor is directly controlled by the end effector. Bottom: the end effector reached the boundary, camera begin to translate, the cursor moves with it.

3.4.2 Haptic Property of Boundary

To provide suitable feedback to different camera motions, we examine two kinds of force which can be involved in our *Haptic Boundary* technique — *sticky force* and *wall force*. We first give the description of them and then analyze their advantages and disadvantages to perform different camera movements, and decide their role in *Haptic Boundary* based on this analysis.

© 2009 Tous droits réservés.

- The *sticky force* $\overrightarrow{F_s}$ felt by the user is directed from the end effector noted as P_e to a sticky point noted as M on the contact surface (figure 3.3 left). The user feels that the end effector is stuck to M. For that reason we call it *sticky force*. The equation 3.5 is used to calculate $\overrightarrow{F_s}$, where K_s is the coefficient representing the stickiness of the contact surface. The figure 3.3 left shows *sticky force* between the handed end effector (represented as a stylus) and the sticky point M.

$$\overrightarrow{F_s} = K_s \cdot (M - P_e) \tag{3.5}$$

- The *wall force* $\overrightarrow{F_w}$ felt by the user is orthogonal to the supporting surface, i.e. directed to the projection of the end effector on the supporting surface, noted as E_{pj} (figure 3.3 right). The user feels that the effector can slide on the surface. The equation 3.6 is used to calculate $\overrightarrow{F_w}$. Figure 3.3 right shows a *wall force* when the user is pressing the end effector on the supporting surface.

$$\overrightarrow{F_w} = K_w \cdot (E_{pj} - P_e) \tag{3.6}$$

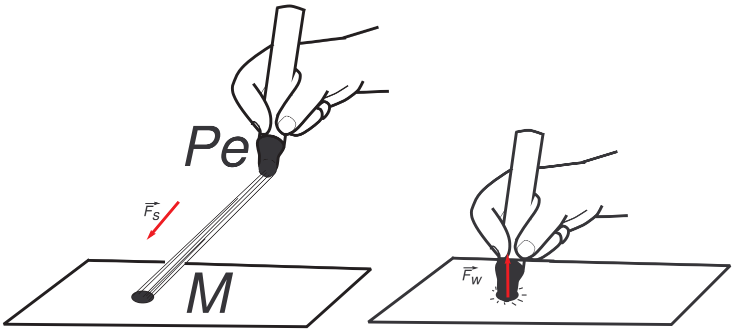


Figure 3.3: Left: *Sticky force*; Right: *Wall force*

Usability in Haptic Boundary

The force applied by user $\vec{F}$ is the counterforce of the force feedback provided by *Haptic Boundary* which can be *sticky* or *wall force*. This applied force is used to rate control the camera translation or rotation. To ease the control of camera, we keep a close relationship between the translation / rotation direction and the applied force direction. In addition, the norm of the translation / rotation speed are also closely related to the norm of applied force. These close relationships maximize the consistency between the user controlled action and the resulting camera motions.

Wall force rate control object translation makes user feel like pushing it. The *Haptic Boundary* simulated by a force feedback device should be composed of several walls (planes with *wall force*), which allows user to translate the virtual camera by pushing these walls with the end effector. Another advantage of *wall force* surface is that users can feel the isotonic zone shape by sliding along its boundary. This is important, especially to

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

novice users. Knowing the shape makes the technique easier to understand, and makes the interaction more intuitive. With the sticky force, it is impossible to get that feeling, because the user feels stuck to the sticky point and can not slide along the boundary.

For *sticky force* rate control, the advantage is that it can provide the translation in any direction. Additionally, for rotation control, *sticky force* can help user to perceive the rotation direction. The user feels rotating the object by "dragging" a rope attached to the surface of the object. The user can change the rotation by simply moving the end effector. While, for the *wall force* its direction is always perpendicular to the contact surface. This monotonicity of the feedback results in the monotonicity of the controlled movement — only in one single direction. If using *wall force* to rate control rotation, the user should move to another contact point even another contact surface (if the current surface is a plane) in order to turn to another direction. Obviously, it is not very convenient, especially for the tasks which requires continuous variation of rotation such as following a winding path. Additionally, comparing to *wall force* controlled translation, the *wall force* controlled rotation has less consistency between the control action (pushing) and the resulting trajectory. The resulting trajectory of the former is straight in the pushing direction. For the latter, the resulting trajectory is winding, less consistent with pushing action. For these reasons, we preferred to involve the *sticky* surface in *Haptic Boundary* to perform the camera rotation.

In the table 3.1 we summarize the advantages and drawbacks of both *wall force* and *sticky force* as the haptic property of the boundary of the isotonic zone to perform the rate control movement.

	User's feeling	*Advantages*	*Drawbacks*
Wall Force	1.Pushing the object. 2.Sliding along the boundary.	1.Easy to feel the shape of isotonic zone. 2.Intuitive rate control translation.	1.Limited movement direction. 2.Less consistency between control action and resulting rotation.
Sticky Force	1.Dragging the object. 2.Stuck to a point.	1.Providing rich translation directions. 2.Providing rate control rotation and translation.	1.Impossible to feel shape of isotonic zone. 2.Translation is less intuitive.

Table 3.1: The summary of the user feeling, advantage and drawback of the *wall force* and of the *sticky force* when applied on the isotonic zone boundary.

3.4.3 Workspace: Isotonic Zone

In this section we discuss issues of the isotonic zone of the *Haptic Boundary*, including its size and shape. The size of isotonic zone should be adjusted so that the user does not feel much constraint during the position control interactions, at the same time, the boundary should be easy to reach to perform rate control interactions. The shape of isotonic zone should also be well-designed to benefit the advantages of both *wall force* and *sticky force*

© 2009 Tous droits réservés.

rate control. At last we clarify the haptic property of each surface of the isotonic zone boundary.

Size of Isotonic Zone

To get an optimal size of the isotonic zone, we performed an informal thinking aloud pilot experiment with several participants, using our first prototype with a cube shaped isotonic zone (figure 3.6 a). The front side of the cube [15] can stick the end effector, and the other sides of the cube act as *walls*. The figure 3.4 showed the cross section of the isotonic zone and marked each side with the force feedback property (*wall* or *sticky*) and the corresponding functionality (rotation or translation). The participants were asked to perform several individual manipulations such as moving an object, orienting an object with position control, translating the camera by pushing the *walls*, and rotating the camera by touching the front surface and then dragging the end effector. During participants' manipulation, we adjusted the size of cube according to their preference by asking them whether their workspace is too small or too large, and whether they can easily reach the boundary.

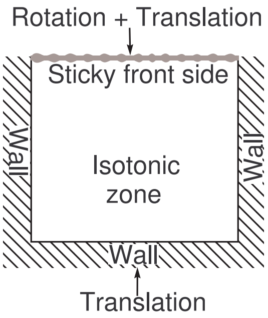


Figure 3.4: The cross section of cube-shaped isotonic zone for size determining experience. The front *sticky* side is used for rotation and translation. The other sides are *walls* which are only used for translation towards the corresponding direction.

After the pilot experiments, a range between 30mm and 50mm in width was generally accepted. For naive users, 50mm is preferred. A smaller workspace made them feel bounded, while the experienced users preferred a smaller size of 30mm. This range fits also to the height and the depth of isotonic zone. We used the middle of this range (40mm) in the following implementation.

Shape of Isotonic Zone

To find the optimal shape of the isotonic zone, we should know what properties this shape should have. Firstly, this shape should be simple, such as cube, semi-sphere, semi-cylinder to ease user's comprehension, learning and use. We use the force feedback device

[15] If you put the cube between you and the monitor screen, its front side is the nearest one to the screen.

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

PHANToM to simulate the isotonic zone. The shape can be felt by user's hand, but can not be seen by users' eyes. Unfortunately, the appearance of input device, such as the workspace, is very important to users, it gives a lot of information about how to use it. For example, when using a spacemouse, users see its cap and will be very quickly conscious of interacting by manipulating this cap. But for the *Haptic Boundary* simulated with *PHANToM*, the isotonic zone is invisible, and lack of the information from the appearance make the complex shape infeasible. So the simplest shape is the best. Secondly, the usable volume of the isotonic zone should be maximized limited by the dimension of 40mm * 40mm * 40mm [16] in order to make users to be able to easily stay in the isotonic zone as they want. At last the chosen shape should favor both the *wall force* and *sticky force* rate control to make the camera control easy to be performed. Further more, we always use a *wall* at the back of the isotonic zone, which can facilitate the backward movement and makes the cursor keep a fixed distance from the camera when it slides along the back wall of the corresponding virtual isotonic zone. We tested five shapes shown in figure 3.6.

- Cube is the first shape we tested. As in the precedent pilot experiment for determining the size, the cube front side is assigned the *sticky* property, other five sides are *walls*, for which the translations in corresponding directions are intuitive. Here, we chose the front side as *sticky* surface because it is easier to drag towards desired direction to have an intuitive rotation compared to the other sides. The cube maximize the isotonic zone volume with 64 cm^3. Its drawback is that when interacting with the *sticky* front side, the end effector easily and unintentionally falls into the isotonic zone, due to its flat surfaces (compared to a convex surface), especially when the *sticky force* is almost parallel with this front side. We can see from figure 3.5,that for the convex surface, the range in which the end effector moves towards the surface without reaching it, represented by d_2 is more important than the one for flat surface, represented by d_1 for the same motion. The more convex the boundary surface is, the less possible the end effector falls unintentionally into the isotonic zone.

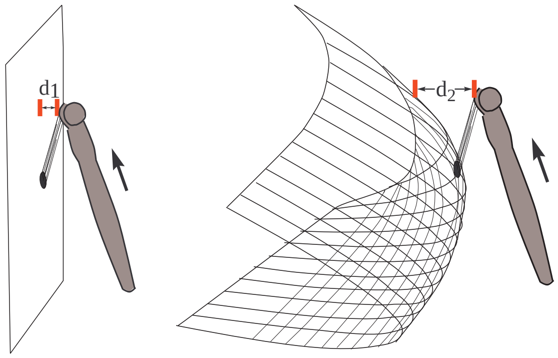


Figure 3.5: *Sticky force* applied on flat and convex surfaces. The movement range of end effector without touching the surface represented by d_1 and d_2. $d_1 < d_2$.

[16] This dimension was determined in the previous pilot experiment

© 2009 Tous droits réservés.

- Semi Sphere: Sphere is a 3D homogeneous convex surface, giving a homogeneous performance. Here we take only semi sphere surface with *sticky* property and add a vertical *wall* at its back. Users interact with it front convex part. The homogeneous convex surface can reduce the chance of falling into the isotonic zone. Its disadvantage is that with this shape, the isotonic zone volume is less than 17 cm^3, it is not enough to move the end effector, especially in depth direction (the distance from the back *wall* to the farthest point on the sphere surface is 20mm, only a half of the depth of the cube shaped zone.).

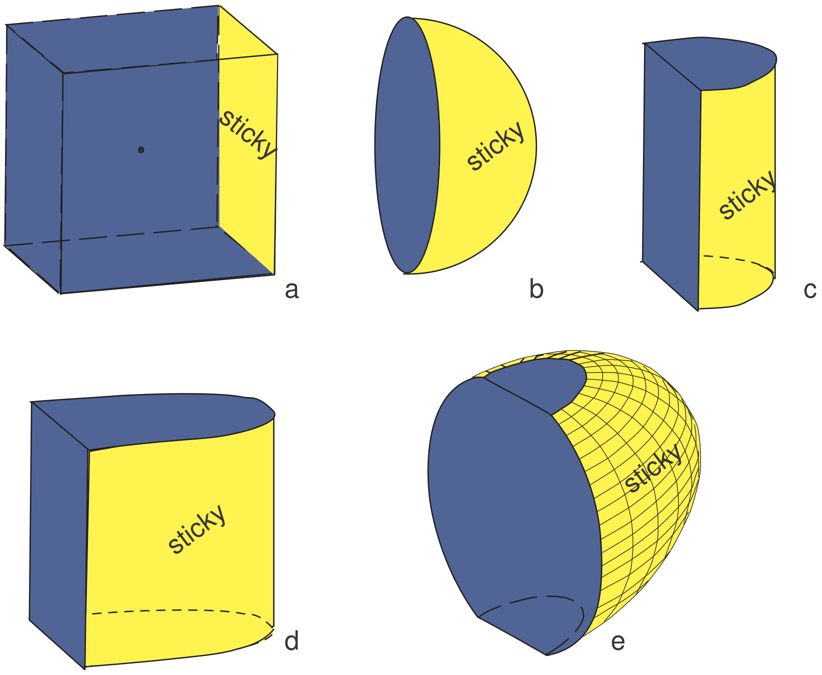


Figure 3.6: The five isotonic shape options. (a) cube, (b) semi sphere, (c)semi cylinde, (d) semi ellipse cylinder (e) semi ellipsoid with ceiling and floor.

- Semi Cylinder: its convex surface is homogeneous in XZ level, which works as *sticky* surface. Comparing to semi sphere, the end effector is easier to fall into the isotonic zone when it moves up and down in the area close to the convex surface. The added vertical *wall* behind and the original ceiling and floor planes help the backward, upward, downward translation. It has the same disadvantage as the semi sphere: insufficient workspace of about 25 cm^3, especially in depth direction. To have more available workspace, we thought about to stretch the convex surface along depth direction, we obtained the shape below.

- Semi Ellipse cylinder: this shape has the similar haptic properties configuration and manipulation advantage as the semi Cylinder. Additionally, it has more available workspace (about 50 cm^3) than semi sphere and semi cylinder.

© 2009 Tous droits réservés.

- Semi Ellipsoid with ceiling and floor: it has a 3D convex *sticky* surface for rotation and three *walls* for the translations. We predict this shape could outperform cube, perform as well as the semi-sphere in *sticky force* rate control rotation due to its 3D convex and outperform semi sphere with its advantage of its rather spatial workspace (about $35cm^3$) and its *wall force* translation.

Usability Qualitative Evaluation for Shape of Isotonic Zone

To evaluate the different shapes described above, we performed a second pilot experiment with three participants. They were asked to perform following motions: rotating around the camera [17], rotating around an object, translating along x, y, z axis of the camera. After that, the participants were asked to rate the different shapes with a 7 points likert scale (1 point means very difficult, and 7 means very easy) for flowing items: ease of rotation around camera, ease of rotation around an object, ease of translation along x,y, z axis, directness of camera control, ease of staying in camera control mode, ease of staying in object manipulation mode, and ease of moving the camera in any desired direction.

Among the averages of user rates obtained from the evaluation, no difference is found for the ease of translation along z axis, the ease of staying in navigation mode and the ease of staying in object manipulation mode. The average user rate for the other items are shown in the figure 3.7, 3.8 , 3.9.

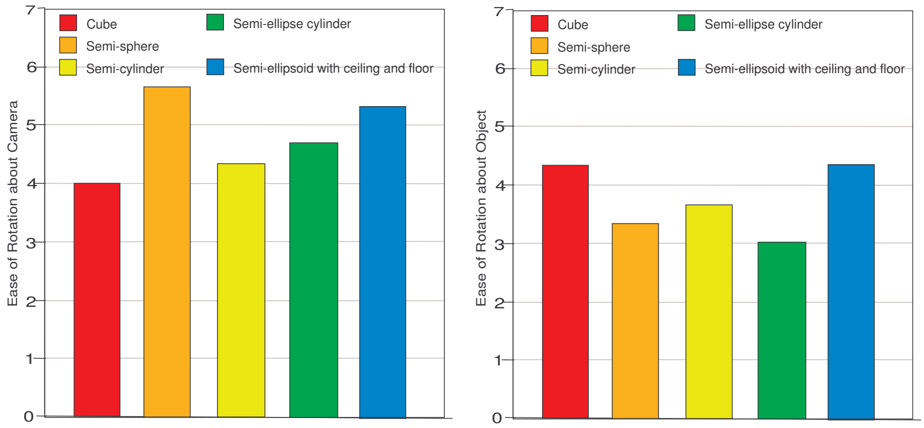


Figure 3.7: User rate average. Left: Ease of rotation around Camera; Right: Ease of rotation around an object.

Let's look at this preliminary result. The two 3D convex shapes, semi-sphere and semi-ellipsoid with ceiling and floor, outperformed others in rotation around the camera (figure 3.7 left). Sphere did not perform well in other manipulation. The cube and semi ellipsoid with ceiling and floor perform the best in rotation around an object (figure 3.7 right). Semi cylinder and semi ellipse cylinder were outperformed by the cube or semi ellipsoid in several rating items (figure 3.7). Additionally the participants also remarked

[17] rotate around the camera is the camera motion which allows looking around without change the camera location

© 2009 Tous droits réservés.

Thèse de Qing Pan, Lille 1, 2008

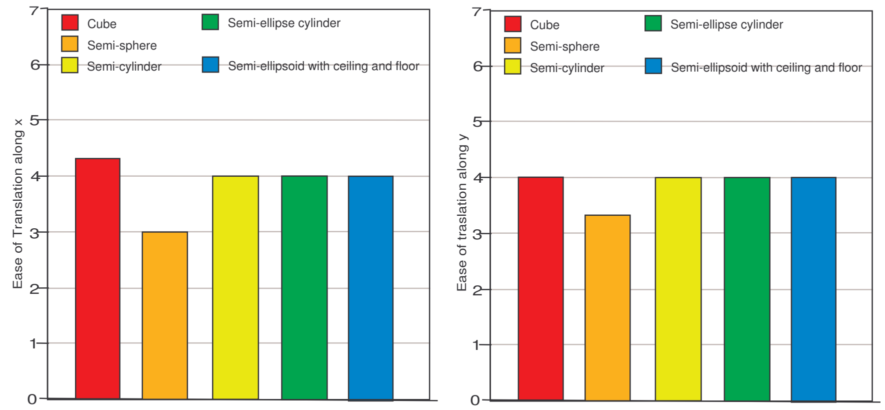


Figure 3.8: User rate average. Ease of translation along *x*, *y* axes.

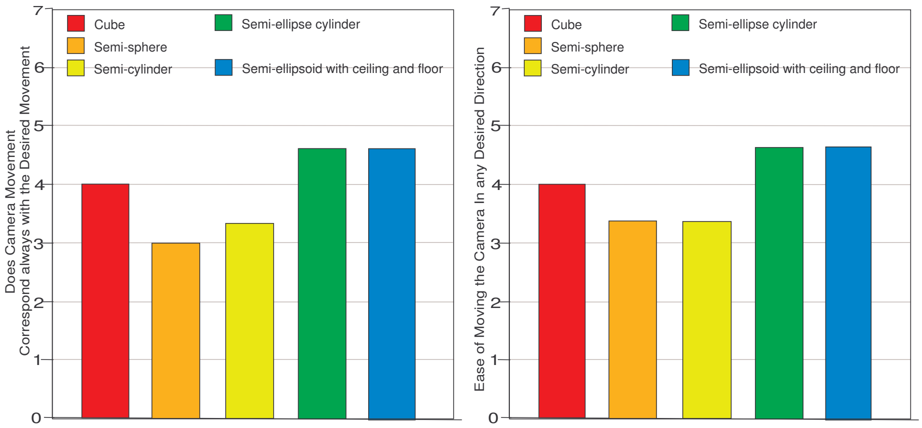


Figure 3.9: User rate average. Left: Does camera movement correspond always with the desired movement; Right: Ease of moving the camera in any desired direction.

the insufficient workspace with the semi-sphere and semi-cylinder . The cube has obtained a good rate, however the end effector unintentionally falling into the isotonic zone is really a problem and can disturb the user. That's why it performed the worst in rotation around the camera. At last, we choose the ellipsoid with ceiling floor as the shape of isotonic zone, we believe that after training, the advantage of its convex for sticky force will make it better outperform the cube.

At last we clarify again the haptic property and corresponding functionality of the isotonic zone having a shape of semi-ellipsoid with ceiling and floor. When the end effector touches the three planes boundary: a wall force calculated by using equation 3.6 will be returned to user, its counterforce applied by user is used to rate control the camera translation, no rotation of camera will be applied in this situation.

$$\overrightarrow{F} = \overrightarrow{F_w} \tag{3.7}$$

When the end effector touches the convex surface of isotonic zone, a sticky force

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

calculated with equation 3.5 will be returned to user, its counterforce applied by user is used to rate control the rotation and translation of camera.

$$\overrightarrow{F} = \overrightarrow{F_s} \tag{3.8}$$

3.4.4 Explore in VE with Haptic Boundary

As we motioned above, *Haptic Boundary* determines the camera position and orientation, which allows the user to explore in the VE, then the cursor position can be calculated. In this section, we explain the camera movement with *Haptic Boundary*.

If the end effector touches any of the three planes, ceiling, floor and back wall, the camera will be pushed upward, downward and backward respectively. The translational speed $\overrightarrow{v_t}$ is controlled by the force applied by user $-\overrightarrow{F_w}$, the counterforce of the $\overrightarrow{F_w}$, using the equation 3.9, where Kv is a constant coefficient to adjust the result speed.

$$\overrightarrow{v_t} = -\overrightarrow{F_w} \cdot Kv \tag{3.9}$$

When the end effector touches the convex surface boundary, the force applied by the user $-\overrightarrow{F_s}$, the counterforce of *sticky force* $\overrightarrow{F_s}$, is used to rotate and translate the camera. An example is shown in figure 3.10. The user moves the end effector from inside to outside of the isotonic zone, touches and passes the convex surface boundary along a rather horizontal trajectory, at last stopped by the *sticky force* (figure 3.10 left). During each loop, the camera translates and rotates a certain amount in its local reference frame. The resulting camera trajectory in VE is shown at right of figure 3.10.

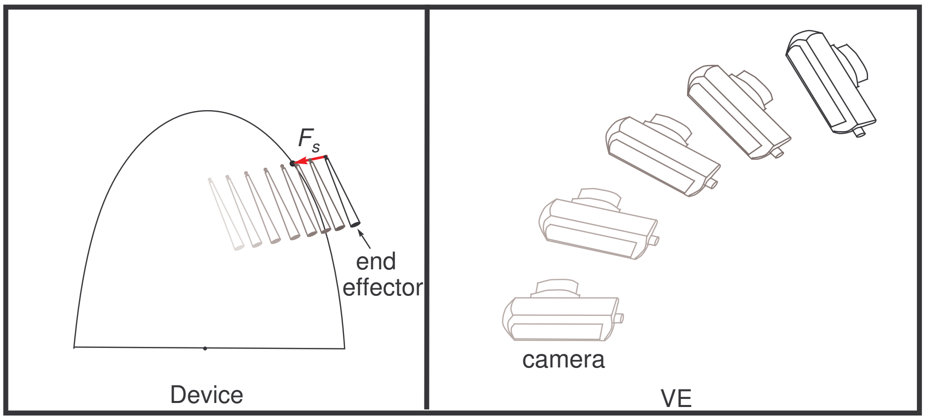


Figure 3.10: Left: The end effector moves out of the isotonic zone. Right: the camera translates and rotates simultaneously.

To develop this idea, we have passed through an evolution of algorithm: we began from a basic idea based on mechanic theory and add modifications to resolve the problems we encountered to ease the interaction. We explain principal ones below.

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

Camera Rotation based on Mechanics

To simplify the calculation of rotation, we use an ellipsoid shaped solid which covers the isotonic zone and their convex surfaces are superposed, shown in figure 3.11. The user applies $-\overrightarrow{F_s}$ on the common convex surface at the right part of the ellipsoid. The left part is prevented from being accessed by the back wall of isotonic zone between them. The $-\overrightarrow{F_s}$ rotates the ellipsoid solid about its center. The ellipsoid solid has x, y, z axes symmetry, can simplifies the calculation of angle speed $\overrightarrow{\omega}$ using the Euler dynamic equations (3.10, 3.11, 3.12) [Phy].

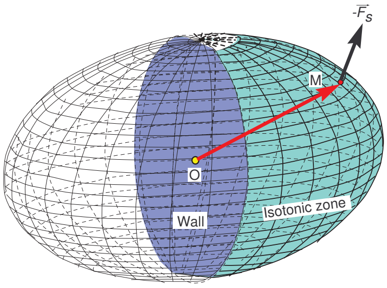


Figure 3.11: The ellipsoid solid used to calculate the rotation which covers the isotonic zone. Their convex surfaces are superposed. O is the center of ellipsoid solid, the *sticky force* $-F_s$ applied at M will rotate it about O.

$$J_x \cdot \frac{d\omega_x}{dt} + (J_z - J_y) \cdot \omega_y \cdot \omega_z = Moment_x \tag{3.10}$$

$$J_y \cdot \frac{d\omega_y}{dt} + (J_x - J_z) \cdot \omega_z \cdot \omega_x = Moment_y \tag{3.11}$$

$$J_z \cdot \frac{d\omega_z}{dt} + (J_y - J_x) \cdot \omega_x \cdot \omega_y = Moment_z \tag{3.12}$$

Where ω_x ω_y ω_z are the three components of the angle speed $\overrightarrow{\omega}$, they rotate the ellipsoid solid about the axis x or y, z. The $Moment_x$, $Moment_y$, $Moment_z$ are the components of $\overrightarrow{Moment}$ applied by user, it is calculated using 3.13.

$$\overrightarrow{Moment} = \overrightarrow{OM} \wedge -\overrightarrow{F_s} \tag{3.13}$$

J_x, J_y, J_z are the moment inertia about x, y, and z axes, which are decided by the shape, mass of the solid. J_x, J_y, J_z of the ellipsoid solid are shown in equation 3.14, 3.15, 3.16.

$$J_x = m\frac{b^2 + c^2}{5} \tag{3.14}$$

© 2009 Tous droits réservés.

$$J_y = m\frac{c^2 + a^2}{5} \tag{3.15}$$

$$J_z = m\frac{a^2 + b^2}{5} \tag{3.16}$$

where a and b are the equatorial radii (along the x and y axes) and c is the polar radius. $a = 20mm$ $b = 20mm$ $c = 40mm$. With these parameters, the J_x and J_y have the same value (equation 3.17), we simplify equation 3.12 to the equation 3.18. Here you can see how the ellipsoid object simplified the angle speed calculation.

$$J_x = J_y \tag{3.17}$$

$$J_z \cdot \frac{d\omega_z}{dt} = Moment_z \tag{3.18}$$

Considering that the resulting angle speed is to apply to camera, a rotation around the view vector (-z axis) is unwanted, which will make the user dizzy. We added some constrained moment to the ellipsoid solid to make the ω_z zero. With this constraint we once again simplify the equations of angle speed calculation. We obtain equation 3.19, 3.20, 3.21. Similar to rotation calculations in chapter 2, we add friction terms μ_x, μ_y to avoid instability.

$$J_x \cdot \frac{d\omega_x}{dt} - \mu_x \cdot \omega_x = Moment_x \tag{3.19}$$

$$J_y \cdot \frac{d\omega_y}{dt} - \mu_y \cdot \omega_y = Moment_y \tag{3.20}$$

$$\omega_z = 0 \tag{3.21}$$

With what we discussed above, the rotation of camera is calculated by simulating rotating an ellipsoid shaped solid with a *sticky force* applied on its convex surface and with a constraint on the rotation about the view direction. However when using these calculated rotations to the camera, some problems come out. Users want a camera to be easier operated than such a solid. Below, we describe the modifications addressing the specific problems that we encountered. These modifications did not completely follow the mechanic law, but make the interaction easier.

Modification for Avoiding Incompatible Rotation and Facilitate Forward Movement

The $\overrightarrow{OM} \wedge -\overrightarrow{F_s}$ is used to produce the rotation of the ellipsoid solid. However, it produces some inconveniences. One problem is shown in figure 3.12 left, where the ellipse represents the cross section along the y, z axes of the solid. We examine an interaction happening in this plane: the user applies $-\overrightarrow{F_s}$ at the sticky point M. The extension line of $\overrightarrow{OM}$

© 2009 Tous droits réservés.

divides the space around M into two areas marked with blue and yellow color. We focus on the part of yellow area above the horizon line: using current $\overrightarrow{OM} \wedge -\overrightarrow{F_s}$ [18], the produced $Moment_x$ is a negative value which will tilt the solid downward. However, the user is applying -$\overrightarrow{F_s}$ having a upward component, what he wants is tilting up the object, opposite to what he obtained.

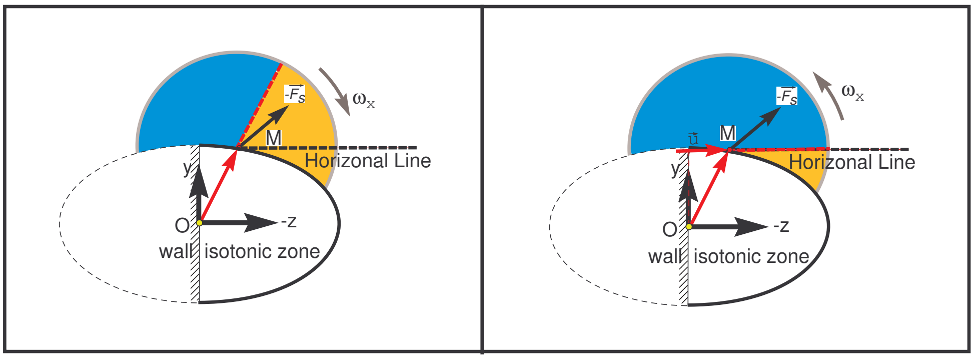


Figure 3.12: Left: Inconvenience of $OM \wedge -F_s$, $-F_s$ has a upward component, but produce a tilting down rotation. Right: Using $u(0,0,-1)$ to replace OM in *Moment* calculation, see equation 3.22, the F_s with an upward component always tilts up.

Another observed problem is the unwanted rotation when the user only wants to move forward. According to the current implementation, to move forward, the user should touch and interact with the front middle part of the convex boundary, represented as the red area in figure 3.13. However, the user is not always conscious of which point on the convex boundary he touched. He just moves forward the end effector. If he touched somewhere outside the red area, for example, M' in figure 3.13 right, the resulting moment of $\overrightarrow{OM} \wedge -\overrightarrow{F_s}$ will observably produce a rotation applied to the camera, which is not what the user wants.

To address these two problems, we modified $\overrightarrow{Moment}$ calculation. We use the equation 3.22, the $\overrightarrow{OM}$ is replaced by $\overrightarrow{u}$ which is the unit vector $(0,0,-1)$. With this calculation, the two problems motioned above are resolved. Any $-\overrightarrow{F_s}$ with upward component applied by user will rotate the camera upward, shown in figure 3.12 right. The user action of moving forward the end effector and then passing the boundary will produce a forward camera movement. In fact any $-\overrightarrow{F_s}$ collinear with the $\overrightarrow{u}$ will only move forward the camera without rotation.

$$\overrightarrow{Moment} = \overrightarrow{u} \wedge -\overrightarrow{F_s} \tag{3.22}$$

Modification to Facilitate the Camera Rotation about a Point and Lateral Translation

With the modified implementation discussed above, the camera can be moved around in the VE. However, the camera rotation around a fixed point which can be the location

[18] $\overrightarrow{OM}$ and $-\overrightarrow{F_s}$ are marked up in the figure 3.12

© 2009 Tous droits réservés.

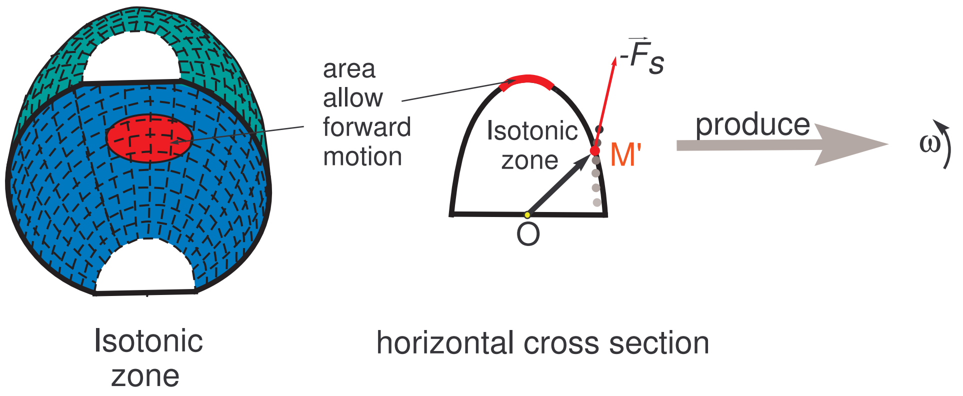


Figure 3.13: Left: The red round area on convex boundary is the area that the end effector should interact with to produce forward translation. Right: when the user moved forward the end effector, touched the convex boundary and stuck at M', a point outside the red area, $OM' \wedge -F_s$ will produce a unwanted rotation, represented by ω, opposite to user's desired forward translation. Only the horizontal cross section of isotonic zone is shown at right, the points with the color from grey to black represent the trajectory of the end effector before and after touched the convex boundary.

of an object, is not easy to perform. Rotation about an object is a common interaction in 3D VE and should be included in *Haptic Boundary* to increase its applicability. To have such a rotation, the camera needs to translate laterally and keep looking at the fixed point, it needs a translation and a rotation in opposite direction i.e., leftward translation with turning right or rightward translation with turning left. Figure 3.14 top showed an example of the latter. However, with $\overrightarrow{Moment} = \vec{u} \wedge -\vec{F_s}$, the lateral force will produce a turning to the translation direction. Like in figure 3.14 bottom, rightward $-\vec{F_s}$ produces a rightward translation and a turning to the right. The rotation around a fixed point can not be performed. Additionally, a pure lateral translation without rotation can not be performed due to the calculation of $\overrightarrow{Moment}$.

We proposed a solution to these problems, where we choose one from three discrete vectors $\overrightarrow{u_0}$ (0,0,-1), $\overrightarrow{u_1}$ (-1,0,0) and $\overrightarrow{u_2}$ (1,0,0) to replace the $\overrightarrow{u}$ in the $\overrightarrow{Moment}$ calculation equation 3.24, and also use this chosen vector as the direction of translation speed $\overrightarrow{v_t}$. The three vectors are represented as $\overrightarrow{u}$ in the equations, which vector is assigned to $\overrightarrow{u}$ depends on if the user desires to move forward and on where sticky point locates. User's desired movement direction can be extracted from the trajectory of the end effector during the period just before and just after touching the convex boundary, it is represented as the points with a color from grey to black in figure 3.15 left. We obtain the vector of the trajectory and test its approximation to the "forward" vector (0, 0, -1). If the included angle between the two vectors is less than 5 degrees [19], like *trajectory 1* in figure 3.15, we are pretty sure that the user wants to move forward, so assign $\overrightarrow{u_0}$ to $\overrightarrow{u}$. Then, if the user

[19] The value "5 degrees" is chosen through a test of an expert user. She was asked to perform 100 times forward movement by touching any part of the convex boundary she wants. 95% of collected trajectories fell into this range.

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

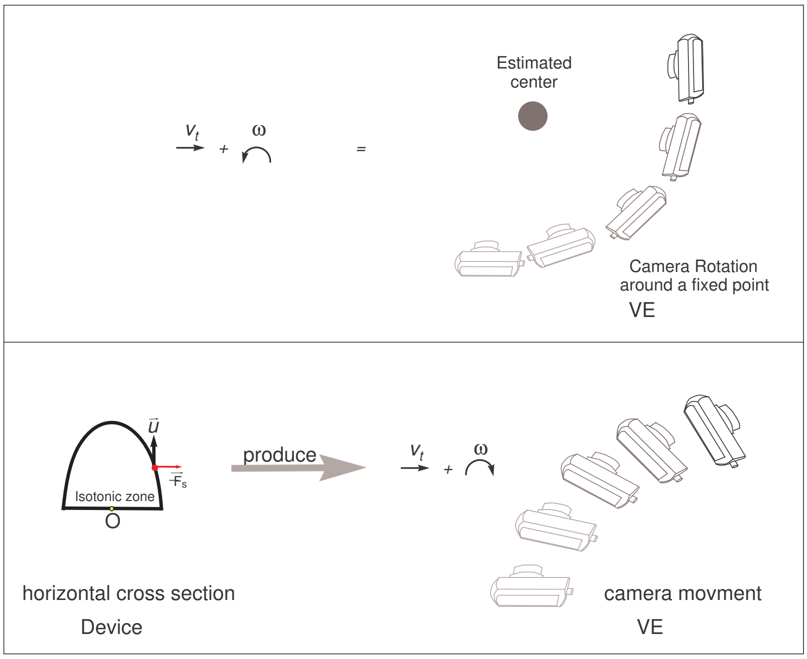


Figure 3.14: Top: The rotation around the approximate center shown at right should be composed of a rightward translation and a leftward rotation. Below: rightward $-F_s$ produces a rightward camera rotation and rightward translation in camera reference system.

drags the end effector away from $\vec{u_0}$, applies a *sticky force* as the $-\overrightarrow{F_{s1}}$ in figure 3.15, the camera will rotate to the right and translate forward, the result camera motion is shown in figure 3.15 top-right. If the vector extracted from the end effector trajectory is not closer enough to the vector (0,0,-1), as the *trajectory 2* in figure 3.15 left, we think that the user does not want to move forward, $\vec{u_1}$ or $\vec{u_2}$ is assigned to $\vec{u}$ according to whether the sticky point (represented by the red point in figure3.15 left) locates at left or right side of the convex boundary (marked with green and blue in figure3.15 left) which makes the camera to translate laterally. If the applied force $-\vec{F_s}$ has a forward component like the $-\overrightarrow{F_{s2}}$ shown in figure 3.15 left, the camera will rotate around a center, its trajectory is shown in figure 3.15 right middle. If the applied force $-\vec{F_s}$ is rightward as $-\overrightarrow{F_{s3}}$ in figure 3.15 left, a pure rightward translation of camera is produced, shown in right-bottom of figure 3.15.

The equation calculating the $\overrightarrow{Moment}$ does not change, only the $\vec{u}$ is assigned different value as we explained above.

Distribution of $|-F_s|$ between Rotation and Translation of Camera

We have determined the translation direction $\vec{u}$ as one of three vectors $\vec{u_0}$, $\vec{u_1}$, $\vec{u_2}$. In this section, you will see the modification to the translation speed norm to ease the control of the trajectory curvature. For the plane boundary of isotonic zone — ceiling, floor and back wall, the translation velocity is proportional to the applied force (equation 3.9). While for the sticky convex boundary, the applied sticky force $-\vec{F_s}$ produces a motion

© 2009 Tous droits réservés.

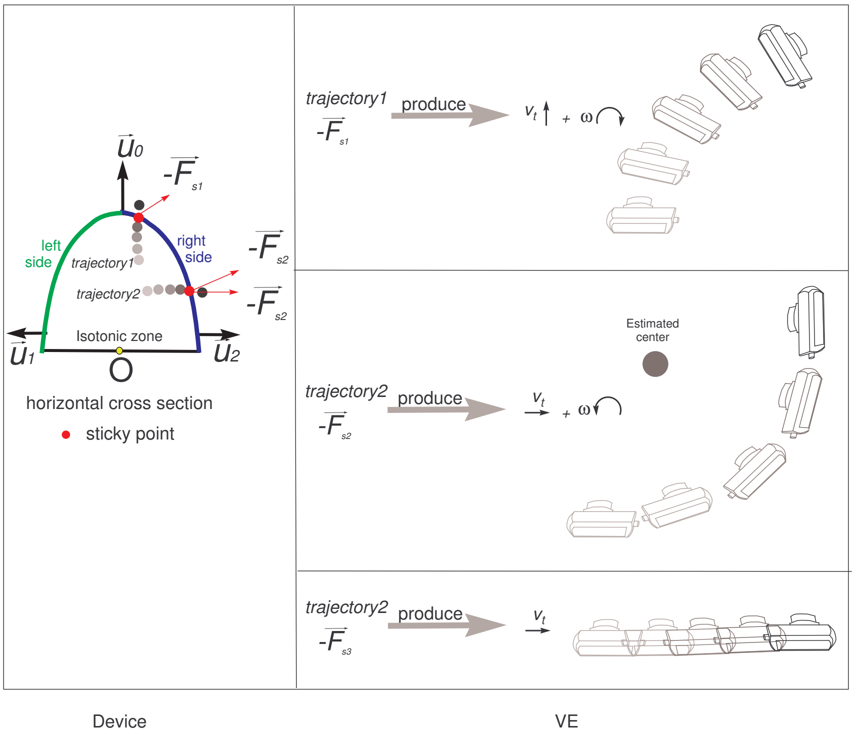


Figure 3.15: Left: The horizontal cross section of the isotonic zone marked with left green side and right blue side. One of three vectors u_0, u_1, u_2 is assigned to u to calculate the rotation and translation. User's desired movement direction is extracted from the trajectory of end effector which is represented by the series of points with color from gray to black. Right: $trajectory1$ indicates that the user wants to move forward, u_0 is assigned to u, *sticky force* $-F_{s1}$ produces a camera moving forward with a rotation (top); $trajectory2$ indicates that the user does not want to move forward, u_2 is assigned to u, *sticky force* $-F_{s2}$ produces a camera rotation around a point marked as the estimated center (middle). $-F_{s3}$ produce a pure rightward camera translation without rotation (bottom).

combining translation and rotation. A translation velocity proportional to the applied force make it not easy to control the camera trajectory curvature.When being stuck on the convex boundary, the user applies a rather stable force, and concentrates on varying the direction of $-\vec{F_s}$ to control the camera movement direction. The translation velocity $|\vec{v_t}|$, proportional to $-\vec{F_s}$, keeps stable unless the user intentionally places the end effector closer or farther to the sticky point to reduce and increase the $|-\vec{F_s}|$. The problem comes out when the user wants to change the trajectory curvature. For example, when the user makes a sharp turn, he needs a great angle speed, so he should perform a gesture of dragging the end effector to make the $-\vec{F_s}$ deviate from the $\vec{u}$ to increase the $|\overrightarrow{Moment}|$; he also needs a small translation velocity to make a turn really sharp, so he should perform another gesture of placing the end effector closer to the sticky point. Performing the two gestures simultaneously requires more effort and attention from user.

To ease the control of trajectory curvature, we distribute the norm of applied force

© 2009 Tous droits réservés.

$|-\overrightarrow{F_s}|$ onto the rotation and the translation of camera according to θ, the included angle between the $-\overrightarrow{F_s}$ and $\overrightarrow{u}$. For the rotation, the norm of $\overrightarrow{Moment}$ is $|-\overrightarrow{F_s}| \cdot sin\theta$. It seems that only a component of $-\overrightarrow{F_s}$ is used by rotation. The other component $|-\overrightarrow{F_s}| \cdot cos\theta$ could be distributed to translation. The equation 3.23 is used to calculate $\overrightarrow{v_t}$ in its local reference frame, where K_v is a coefficient to adjust the resulting speed norm. A similar coefficient K_a should be applied to $\overrightarrow{Moment}$ calculation, equation 3.24. Then, the angle speed of camera ω is calculated by using equation 3.19, 3.20. Additionally, θ can vary between 0° and 180°. With the increase of θ, the $|\overrightarrow{v_t}|$ decreases and $|\overrightarrow{\omega}|$ increases till $\theta = 90°$, then $|\overrightarrow{v_t}|$ became a negative value and ω decrease. To avoid both, we add a constraint that if $\theta > 90°$, $|\overrightarrow{v_t}| = 0$, and ω keeps its maximum value $-\overrightarrow{F_s} \cdot K_a$ and there is only rotation at top speed in this situation. The user can control the distribution of $|-\overrightarrow{F_s}|$ between rotation and translation: the closer is $-\overrightarrow{F_s}$ to the translation direction $\overrightarrow{u}$, the faster the camera translate, the slower the camera rotates; reversely, the camera rotates faster, but the camera moves slower.

$$\overrightarrow{v_t} = \vec{u} \cdot |-\overrightarrow{F_s}| \cdot cos\theta \cdot K_v \tag{3.23}$$

$$\overrightarrow{Moment} = \overrightarrow{u} \wedge -\overrightarrow{F_s} \cdot K_a \tag{3.24}$$

In figure 3.16, sharp turn example is explained. When having a forward translation direction and the user applies a rightward force $-\overrightarrow{F_s}$ to turn to right, with and without distribution of $-\overrightarrow{F_s}$ produce two different camera trajectories. With the distribution, the translation speed is nearly zero, thus the camera turns around without translation, the red point is its trajectory (figure 3.16 middle). Without the distribution, the camera translation speed proportional to $|-\overrightarrow{F_s}|$, its trajectory (figure 3.16 right), represented by the red line, is much longer, the user may miss his destination.

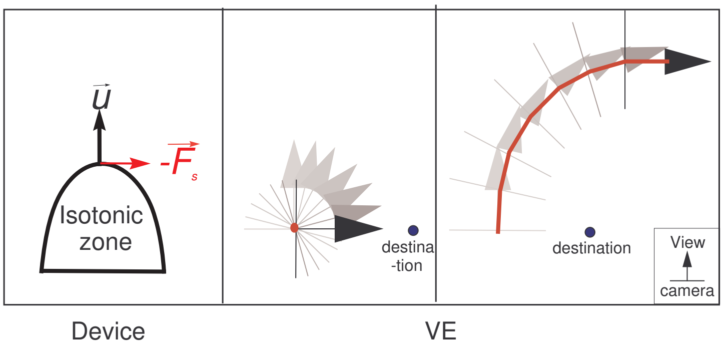


Figure 3.16: Trajectory of camera with and without the distribution of $|F_s|$. Left: Camera is translating along u, then the user applied a rightward force to turn right. Middle: the trajectory of camera with distribution of applied force. Right: the trajectory of camera without distribution of applied force.

We have shown you the major modifications to improve the performance of *Haptic Boundary*, which allows avoiding the incompatible rotations to the applied force with the

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

Thèse de Qing Pan, Lille 1, 2008

initial mechanic implementation and easing the camera motions including the forward movements, lateral movements, rotation around object and easing the control of the camera trajectory curvature. With current implementation, users have only three translation directions $\overrightarrow{u_0}$, $\overrightarrow{u_1}$, $\overrightarrow{u_2}$ which constrains the camera motions to some certain comparing to the previous implementation. However, from another standpoint, the current solution allows users to easily perform the more standard motions, and make the camera trajectory more regular. Users can easily move forward, leftward, rightward and after some training, the user could make the camera rotate about any object along a regular circle trajectory and can change the distance from the object by adjusting the $-\overrightarrow{F_s}$ direction. These regular camera motions may be preferred by users.

3.4.5 Interaction System

In this section, we describe a simple but complete interaction system in which the *Haptic Boundary* is applied. The figure 3.17 shows the global view of the system. It includes several parts: *calculator*, *input device*, *interaction*. The main part is the *calculator* which determines the position and orientation of the camera as well as the position of cursor. *Calculator* obtains the effector position P_e from the *input device* and determine the value of force feedback according to the haptic property of isotonic zone and return it to the *input device*. If the end effector is inside the isotonic zone, the process directly enters the *camera cursor system* [20] subpart without moving the camera. Otherwise, the process enters the *Explore* subpart. The *Explore* subpart produces the position and orientation of camera, P_c and Q_c, by applying *Haptic Boundary*. These two values and P_e are passed on to the *camera cursor system* subpart to calculate the cursor position P_p. Then, the *collision* subpart gathers the information about the scene from the *interaction* to check whether the camera and cursor will collide with the existing scene or not if P_c and P_p are assigned to them. The *interaction* part observes whether the user clicks to select or drag on any object and attach or detach the object to cursor.

Object Manipulations

The touching-based selection is adopted in this system. With Haptic Boundary, the user can get closer to object, and then select or manipulated. When the end effector is inside isotonic zone, the camera does not move. The cursor with the end effector. The combination of equations 3.1-3.4. is used to calculate P_p, we obtain :

$$P_p = (P_c - P_{cw}) + Q_c \cdot (P_e \cdot CD) \cdot Q_c^{-1} \tag{3.25}$$

$$\overrightarrow{F} = 0 \tag{3.26}$$

The movement of cursor is used to precisely manipulate the selected object. When the user drags on an object, it is attached to the cursor, the cursor position determines the object position. The position control inside the isotonic zone allows a precise movement. The rate control when reaching the boundary allows rapid movement in coarse grain.

[20] *camera cursor system* adopts the classic algorithm discussed in section 3.3.1

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

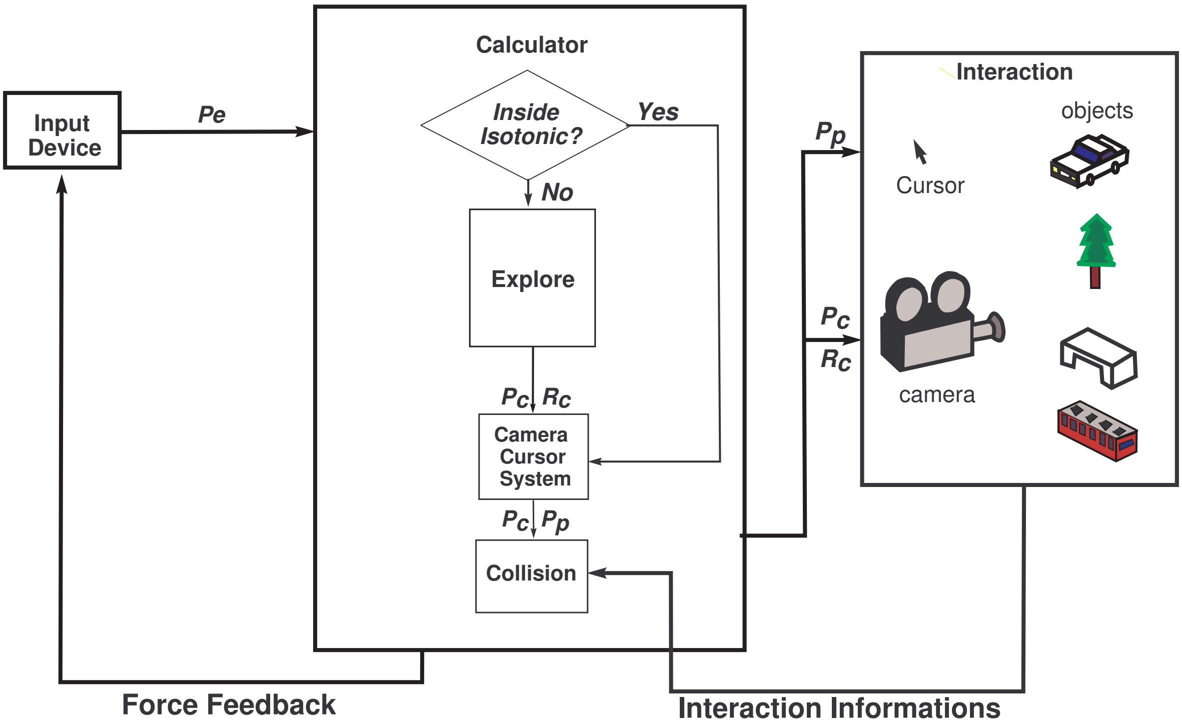


Figure 3.17: Interaction System

For the object orienting, several traditional method can be adopted. A possible implementation can be that the user first clicks on the desired object, then this object is highlighted. When the cursor is near enough to the selected object, for example inside the object's bounding box, the cursor changes to a "hand" shape to indicate that the orienting is allowed, and then dragging movement is interpreted as the rotation of object. If the orientation DOFs are available in the input device, such as *PHANToM*, a direct or non-isomorphic [PWF00] mapping can be used to orient an object. To avoid unpredictable resulting motion in elastic zone, the object rotation is allowed only when the end effector in the isotonic zone.

Collision test

The goal of collision test is to check whether the calculated position of camera and cursor are in collision with any other object in the scene. If this is true, the correction will be made. For camera, an extra step is applied which constrains it in the scene. The two steps are performed before the update cursor and camera.

3.4.6 Visual feedback for Novice Users

Haptic Boundary PHANToM to simulate the isotonic zone including the *sticky force* and *wall force*. But for a user who never touched such a device, it's difficult to understand haptic display. In an earlier pretest, some users can not feel the isotonic zone shape or

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

forget it very quickly. Additionally, when interacting with the *sticky force*, they were not aware of the direction of *sticky force* through their hand. It's a big barrier to understand and master our technique.

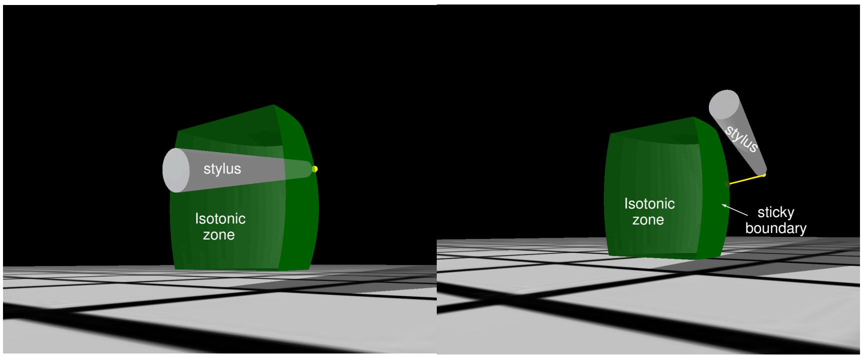


Figure 3.18: Aid to understand *Hptic Boundary*. Left: The wall force is activated for the whole boundary, the user can feel the shape of isotonic zone by sliding the end effector along its boundary. At the same time the user can see the isotonic zone displayed on the screen. That makes the isotonic zone easier to understand. Right: After known about the shape of isotonic zone, the sticky convex boundary is activated, the stylus is stuck on convex boundary, the yellow line gives this information.

Aiming at this problem, in the learning stage, we display the isotonic zone and the stylus on the screen, and *PHANToM* haptic display the whole isotonic zone surface as *wall*, and let user see and feel its shape. After the user has remembered this shape, we recover *sticky* property of the convex surface. The *sticky force* is represented by a yellow line on the screen, to give the user an impression that something sticks the end effector to the convex surface.

This learning aid helps the naive user to understand the haptic feedback by helping him to build the mental image of the simulated input device using *PHANToM*, help understanding *Haptic Boundary*.

3.5 Car Assembling Experiment

To evaluate the performance of *Haptic Boundary*, we performed this experiment and compared *Haptic Boundary* hybrid technique to bimanual manipulation navigation technique, and unimanual interaction technique with an explicit switch between object and viewpoint manipulations.

3.5.1 Goals and Hypotheses

Our *Haptic Boundary* technique is designed to perform object and viewpoint manipulations without explicit switch by combining the isotonic position control and elastic rate control with rich force feedback, which will require less mental load than the technique with an explicit mode switch. The bimanual techniques allow the user to control the

© 2009 Tous droits réservés.

camera in one hand, and to manipulate objects in another hand, which, a priori, will facilitate the interaction, but its user performance is very sensible to the parallelisms of two hand motions. We give following hypotheses:

H1: The *Haptic Boundary* technique will outperform the unimanual interaction technique with explicit switch (*Explicit Switch*). which will require users to think about switching between view object manipulations, translation and rotations during the interaction. With *Haptic Boundary*, switch is performed implicitly when touching the boundary and returning to the isotonic zone. The *sticky force* and *wall force* will make user be aware of what motion they are controlling. Users receive the information about different modes, do not have to perform the switch intentionally.

H2: The *Haptic Boundary* may require more learning time than bimanual interaction technique (*Bimanual interaction*), after certain period of learning, the *Haptic Boundary* will performs as well as the *Bimanual Interaction*. The later is more natural for users because people work with two hands everyday.

3.5.2 Techniques to be Compared

Explicit Switch Technique

We chose Otaduy's user centric viewpoint computation technique for haptic exploration and manipulation [OL01] (*User Centric*) as the base of the *Explicit Switch* technique to be compared.This technique use the haptic feedback device *PHANToM* as input. So does *Haptic Boundary*. This could avoid the difference of performance bought by input device, and enhance the accuracy of comparison result. In *User Centric* technique, the position / orientation of the stylus of *PHANToM* to calculate the position and orientation of the main camera and cursor. "position mode" and "velocity mode" are distinguished. The camera is anchored in virtual workspace and cursor position / orientation related to the virtual workspace is mapped directly from the ones of the *PHANToM* stylus. In "position mode" the virtual workspace remains still; In "velocity mode" the difference between the current position/orientation of the stylus and its last position/orientation in the "position mode" is used to rate control the rotation and translation of the virtual workspace. In [OL01], the switching between modes is not explained, but explicit switch is a logical choice between "position" and "velocity" modes. A familiar "pressing CTRL key" was chosen to do this job in our experiment.

Bimanual Interaction Technique

We chose the configuration of the bimanual interface in *Reachin* technology [Rea] to compare with. *PHANToM* stylus in dominant hand is used to interact with the objects in VE and the *spacemouse* to rotate the whole VE. but in our implementation we use all 6 DOF of *spacemouse* to control the orientation and position of camera to allow a navigation with more extent.

© 2009 Tous droits réservés.

Thèse de Qing Pan, Lille 1, 2008

3.5.3 Implementation of Techniques with PHANToM

All of the three techniques are implemented using *PHANToM* Desktop, controlled by Dominant (D) hand. The *Bimanual Interaction* using a *spacemouse* in non-dominant (ND) hand. The *wall* planes of isotonic zone has a stiffness of 60 $N.m^{-1}$. The *sticky force* has a springiness of 10 $N.m^{-1}$. These settings were chosen after running a pilot experiment testing different intensity values to allow user to feel the difference between zero force feedback, *sticky force* and *wall force* without having too much fatigue. For the *Explicit Switch* and *Bimanual Interaction*, a cube shaped workspace with a dimension of 40mm*40mm*40mm is simulated by the *wall force*, which limits the movement of *PHANToM* stylus, to give similar workspace size for each technique. The object manipulation is performed by dragging with the button of the *PHANToM* stylus pressed. 'CTRL' key on keyboard is used to switch between "position mode" and "velocity mode" in the *Explicit Switch* technique to avoid the confusion with the selection actions. A spring force of 10 $N.m^{-1}$ springiness is activated during the "velocity mode".

For all three techniques, the input devices were placed near the center of screen, to minimize the mismatch between the reference frame displayed on the monitor and the one of user's input. For *Bimanual Interaction spacemouse* and *PHANToM* were placed close to each other (figure 3.19 right), to minimize the parallelism motions of two hands.

Our experiment was conducted on a 3G Hz PC with 19 inch, 85 DPI LCD monitor. Our C++ software displays the stimulus at 60 Hz. The haptic display on *PHANToM* Desktop is updated at 1000 Hz.

3.5.4 Task and Stimuli

The task was assembling a car body with four wheels. The car body is positioning at the center of our virtual environment: a garage. The four wheels locates at different distances from the car. Participants were asked pick up one wheel and move to the corresponding target position indicated by a box with colorful edge. The four wheels and

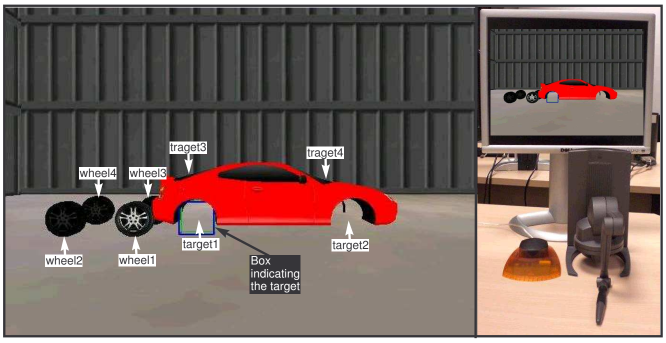


Figure 3.19: Left: Selection experiment interface: the four wheels 1,2,3,4 should be placed to target 1,2,3,4. After each trial, the camera is reset to the position which give the view in this figure left. Right: Experimental Apparatus, PHANToM is used for *Haptic Boundary* and *Explicit Switch* technique, for *Bimanual Interaction*, *spacemouse* and *PHANToM* are used.

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

Thèse de Qing Pan, Lille 1, 2008

their corresponding target positions are marked in figure 3.19 left. The wheel 1 should be placed at target 1 position, the wheel 2 at target 2, and so on. The current target position is marked with a wire-frame indicating box. At a given time, only one wheel is manipulatable and displayed with a normal appearance and the other wheels can not be interacted with, are displayed with more darken color (figure 3.19 left). When the current wheel is positioned within the tolerance of target, it is highlighted, which served as the visual feedback for participants. After the wheel is dropped within this tolerance, it became immobile and the next wheel becomes available; the indicating box appears at the next target position. And the camera is reinitialized at the pre-defined position which gave the view shown in figure 3.19 left. This makes the four wheel assembling trials have different camera manipulation requirements: for first two wheels the camera only need to be translated. For wheel 3 and 4 the camera should be rotated and translated to another side of the car to see the target position. Participants must place the camera at the same side of car as the target before positioning the wheel, which forces participants to perform more camera manipulation. Participants were told to perform as quickly as possible. The figure 3.19 is the experiment interface.

3.5.5 Participants

3 right-handed volunteers participated this experiment with a mean age of 25.6 (SD = 0.94). All of them have less or more experience of 3D VE interaction, such as video games. Two of them have little experience of haptic input device.

3.5.6 Experiment Design

A within-participant repeated measures design was used. The independent variables are techniques(*Haptic Boundary*, *Explicit Switch* and *Bimanual Interaction*), the target position which decides the shortest trajectory distance from wheel to target as well as the orientation difference between the camera initial orientation and the orientation necessary to see the target. These two variables define the minimum required amount of manipulation.

All participants performed the experiments with all three techniques. The presentation order of three techniques was counterbalanced across participants. For each technique, the participant completed a training period of 2 sessions. During training, participants were given some suggestions and they can manipulate the wheels and the camera with the strategy they choose[21] and did not have any time limit, the purpose is to learn and master the techniques as well as they can. And then, each participant performed 4 sessions of moving all four wheels to their target positions one after one with a pre-defined order which began with the wheel-target pair requiring the less manipulation, and came with the one requiring the most. The participants were allowed breaks between sessions.

[21] User can interact with their own strategy. For example, they can firstly adjust the camera to have a good view of the target position, then take the wheel and place it at the target position. Or, they can firstly take the wheel and adjust the camera to see the target with the wheel attached to the cursor and place it.

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

Thèse de Qing Pan, Lille 1, 2008

After all 4 sessions of each technique, a short questionnaire asked participants about their strain and dizziness, and asked them to compare the technique to the previous one. At the end of the experiment, a final questionnaire asked for an overall ranking of the three techniques. The experiment last for about 60 minutes.

In brief, the experimental design was: 3 participants ×
3 techniques (*Haptic Boundary*, *Bimanual Interaction*, *Explicit switch*) ×
4 sessions of trials for each technique ×
4 trials per session (4 wheel-target pair)
= 144 total trials.

3.5.7 Results

The dependent variables were *Completion Time* and the *Inefficiency* which is based on the inefficiency quantification proposed by Zhai [ZM98]. We examined *Translation Inefficiency* I_{trans} and the *Rotation Inefficiency* I_{rot}. Although we have only three participants in our experiment, we still performed the ANOVAs for each dependent variable to have a rather reliable analysis result.

Completion Time

The completion time of one trial is the time from when the participant begin to move the end effector until the wheel was dropped inside the tolerance of the target. Repeated measures analysis of variance showed that no effect of the presentation order of *technique* was found on completion time, indicating a within-participants design was appropriate. No significant effect for *session* indicating that there was no learning effect present. There was a significant main effect for *technique* ($F_{2,4} = 14.64$, $p < 0.05$) and *trial* ($F_{3,6} = 22.83$ $p < 0.05$) on completion time. No significant effect of interaction was found between *technique*, *session* and *trial*.

Pair-wise comparisons revealed a significant difference between technique *Explicit Switch* and the two other techniques ($p = 0.035$ for *Haptic Boundary* , $p = 0.027$ for *Bimanual Interaction*), *Explicit Switch* was 50% slower than the two others (figure 3.20 left). Pair-wise comparison also showed that the *trial* 1 and 2 significantly take less time than *trial* 3 and 4 ($p = 0.03$), which depends on the required amount of manipulation of each trial (figure 3.20 right).

Although the effect of interaction between the *technique* and *trial* is not significant, We give the figure 3.21 for additional informations on Completion Time.

Translation Inefficiency

Translation Inefficiency I_{trans} is defined as the ratio $\frac{l-d}{d}$ where l is the length of the trajectory covered by the cursor from picking up the wheel until positioning it at the target, d is the shortest distance between the target and the wheel initial position. In our experiment, participants is allowed to move freely and manipulate the wheel and camera as they want. For example they can take the wheel and then translate and rotate the camera to face the target position, the wheel moves with the camera, which results a much

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

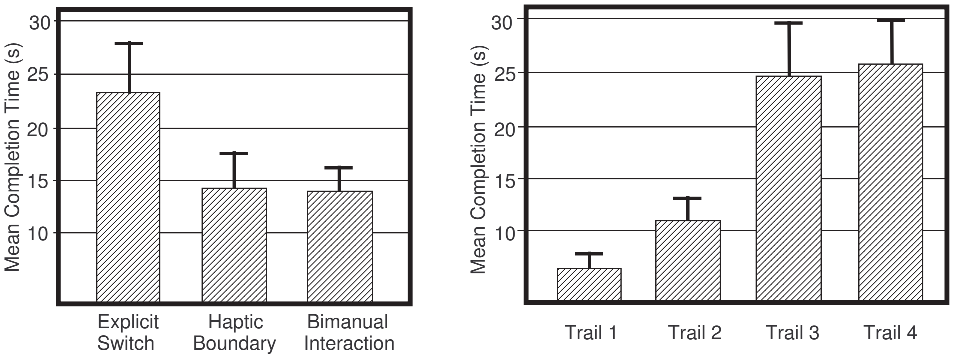


Figure 3.20: Mean Completion Time for different technique (left) and different trial (right). (error bars 95%)

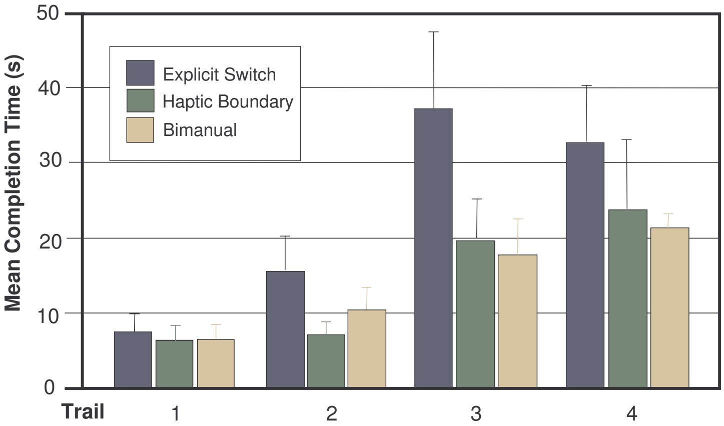


Figure 3.21: Mean Completion Time for different technique and different trial.(error bars 95%)

longer trajectory comparing to the strategy where the participants adjust the camera to face to target at first, and then pick up the wheel and begin the trajectory. So, the I_{trans} can be easily affected by individual participant's interaction strategy and vary in a very large range (90 percents of I_{trans} value fall in the interval [15%, 2000%]). May be for this reason, we did not obtain a significant difference for the variable of experiment design *technique*, *session*, *trial*. However we can still see some tendency from the figure 3.22 which shows the mean translation inefficiency for *technique* and *session*. For the trial 1 where the wheel and the target position are very close to each other, the I_{trans} is almost the same for all three techniques. For the other trials requiring more translation and curve trajectory of camera, I_{trans} mean for the *Haptic Boundary* and *Bimanual Interaction* are close, and the *Explicit Switch* has a higher *Translation Inefficiency*.

© 2009 Tous droits réservés.

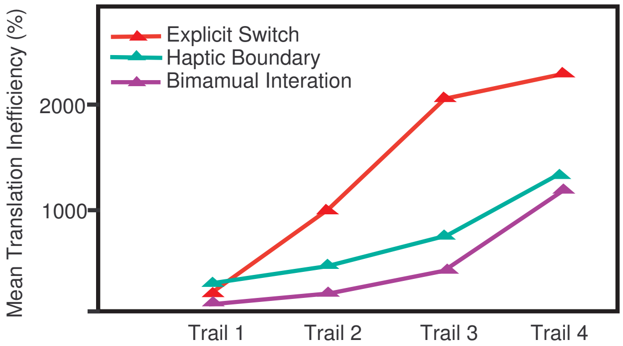


Figure 3.22: Mean inefficiency of translation I_{trans} for each technique, each trial.

Inefficiency of Rotation

The inefficiency of rotation I_{rot} is similarly defined as the ratio $\frac{\theta - r}{r}$ where θ is the actual camera rotation performed by the participant, r is the difference between the initial orientation of the camera and the orientation to face the target. In our experiment, the rotation of camera is not necessary for trial 1 and 2. The I_{rot} is only calculated for trial 3 and 4. There is no significant difference found for *technique*, *session*, *trial*. The figure 3.23 showed the mean of I_{rot} of all three techniques. We can see, globally, the *Rotation Inefficiency* for *Bimanual Interaction* technique is higher than for the two others.

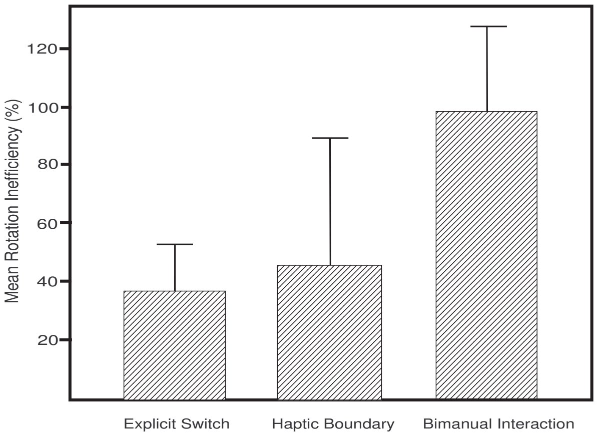


Figure 3.23: *Rotation Inefficiency* I_{rot} for three techniques (error bars 95%)

User Feedback

Among the three participants, one preferred the *Haptic Boundary*, he said that with the *Haptic Boundary*, he did not have to learn the synchronization between two hands as in *Bimanual Interaction*. But others participants said that it takes certain time to learn and master *Haptic Boundary*, and it may be inefficient for application like video games. Two participants preferred the *Bimanual Interaction*. They said that the collaboration

© 2009 Tous droits réservés.

of two hands facilitates the interaction and the camera control with *spacemouse* is easy to understand. However one participant felt that sometimes *spacemouse* is less precise than *Haptic Boundary*. That may be due to pure elastic property of *spacemouse*. No one preferred the *Explicit Switch*. Participants said that it was more difficult to master, and need more training; some gestures make the hand uncomfortable. However one participant said that the *Explicit Switch* is intuitive. Table 3.2 showed the ranking of technique given by three participants using a note between 1 to 3: 1 means most preferred; 3 means least preferred.

Participant No	01	02	03
Explicit Switch	3	3	2
Haptic Boundary	1	2	3
Bimanual Interaction	2	1	1

Table 3.2: User ranking for three techniques

Additionally we asked participants to rate their arm strain, hand strain dizziness after four sessions of each technique. The note is between 1 and 7, 1 means no strain, or no dizziness, 7 means very strong strain or very strong dizziness. Table 3.3 showed the result. From this table we can see the *Haptic Boundary* is the one bring less hand strain and dizziness. The *Explicit Switch* bring the most of all three comfortlessness.

	Explicit Switch	*Haptic Boundary*	*Bimanual Interaction*
Arm strain	2.66	1.66	1.66
Hand strain	2.33	1.33	2.33
Dizziness	2.66	1.33	1.66

Table 3.3: User rate of arm strain, hand strain and dizziness

3.5.8 Discussion

Our experiment confirmed our two hypothesis. For *H1*, the *Haptic Boundary* technique has a performance advantage of 50% over the *Explicit Switch*. We can see from figure 3.21, for the trial 2, 3, 4 which require more camera movement, the difference is even more visible. The implicit switch and *sticky force* camera manipulation made *Haptic Boundary* more efficient than the *Explicit Switch* which brought more mental load and the stylus manipulation of *User Centric* may be awkward for user's hand. For the *H2*, after enough training, the *Haptic Boundary* can perform as well as the *Bimanual Interaction*. No significant difference of completion time between the two techniques is found, although the mean value showed a slight advantage for *Bimanual Interaction*. According to the user feedback, two participants preferred *Bimanuam Interface*, one preferred *Haptic Boundary*, and the *Haptic Boundary* brought less hand strain and dizziness than *Bimanual Interaction*.

© 2009 Tous droits réservés.

http://www.univ-lille1.fr/bustl

We have observed that participants had still some difficulties to rotate the camera around the car, because they did not make full use of the solution described in section 3.4.4. To rotate around the car, the user should give a non-forward trajectory of end effector before touching the boundary, and the *sticky force* $-\overrightarrow{F_s}$ with a forward component. This is a gesture requires a little more training before master it. In the following section, we combine another method *Orbiting Inspection* to ease the rotation about an object.

3.6 Enhancing Haptic Boundary with Orbiting Inspection

In this section, we combine the *Orbiting Inspection* functionality with our *Haptic Boundary* technique to facilitating the camera rotation around an object. When the user doubleclick an object, the *Orbiting Inspection* of this object is activated. This object serve as *Point of interest* (*POI*), and is indicated with a bounding box. Just after this doubleclick, an animation is applied to the camera, to make the camera face to the *POI* object. Because object inspection should keep certain distance D from *POI* object, the camera will be pull away from the *POI* to respect this distance if the camera is too close to the *POI*. In our implementation, we use the half of the diagonal of the *POI* object's bounding box as D. After this animation, the *POI* object is displayed at the center of screen, the camera is ready to inspect it. The begin and the end of the animation is shown in figure 3.24.

During *Orbiting Inspection* the applied force $\vec{F}$ ($\overrightarrow{F_s}$ or $\overrightarrow{F_w}$) is used to move the camera. The camera is moved along a sphere around the *POI* object. The z component F_z is used to push toward or pull away from the *POI*, change the sphere radius, the x and y components F_x F_y move the tangential line direction of the sphere surface. The camera is always looking at the *POI* object. The schema is shown in figure 3.25. When user double click again, the interaction exits from the *Orbiting Inspection* mode, *Haptic Boundary* is recovered.

3.7 Conclusion

In this chapter, we described our *Haptic Boundary* combining the isotonic-position and elastic-rate control which allows user to perform object manipulation and exploration in VE. The *wall force* and *sticky force* feedbacks during rate control enable the intuitive translations and rotations of camera. The size and shape of isotonic zone were carefully designed and adjusted to maximize user comfort, user performance and the profit from two force feedbacks. We described in detail the implementation for exploration functionality, in the final version, the user could perform the regular camera movements such as forward, leftward, rightward, rotation about camera itself and rotation around object. Although the gestures need certain training before master it, the *Haptic Boundary* outperformed the unimanual interaction technique with explicit switch by 50% in completion time in the of car assembling experiment and no significant difference is found between *Bimanual Interaction* technique and our *Haptic Boundary*. To make easier the rotation around an object, we introduced the *Orbiting Inspection* to combine with *Haptic Boundary*.

© 2009 Tous droits réservés.

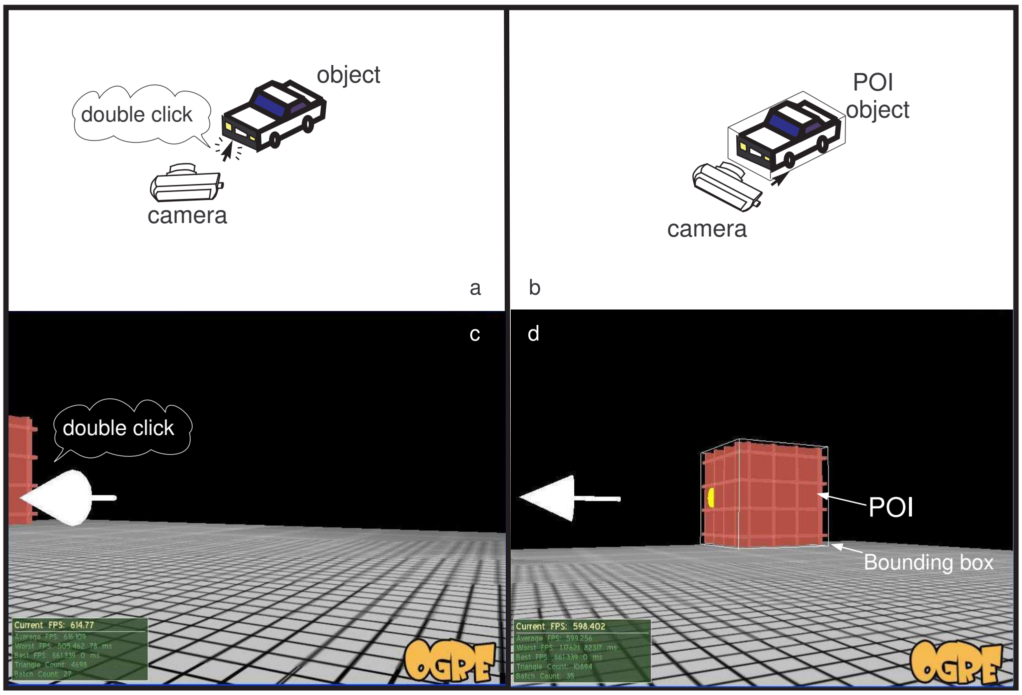


Figure 3.24: When the user doubleclicks on an object, the *Orbiting Inspection* is activated. An animation is launched. a: the user doubleclick an object (the car), it became the POI object; b: the fin of the animation. c, d show the view from the camera, at the moment of a and b. The *POI* is the pick cubic.

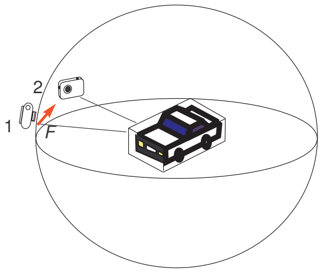


Figure 3.25: *Orbiting inspection* of the *POI* object. Force F pull the camera for position 1 to position 2.

© 2009 Tous droits réservés.

Thèse de Qing Pan, Lille 1, 2008

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

Conclusion

The goal of our work is to improve the user performances of both 2D classic interface and 3D virtual environments by adopting the interaction techniques based on hybrid isotonic position and elastic rate control.

To achieve this goal, we started with the state of the art of interaction tasks. Facilitating them is the goal of every interaction technique. We emphasis on each elemental task and their parameters which determine how difficult the task is and which may favor certain techniques than others.

Then we studied the input devices with the focus on their resistance and their haptic feedback which can be either passive, corresponding to the resistance, or active, provided by the motors embedded in some devices. Haptic feedback play an important role in our proposed techniques. After this we studied the basic transfer functions and their relationship with the resistance of device.

To have a good knowledge of the existing interaction techniques, we spent a little more ink in summarizing techniques for both 2D WIMP and 3D virtual environment, to learn the common key factors, try to find out their limitations, and try to avoid these limitations in our techniques. We then studied the bimanual interaction techniques including their theory foundation and existing bimanual interfaces. One of them was compared to our second proposed technique. Then we studied a few of existing technique based on hybrid position rate control. They inspired the techniques proposed in this thesis.

Although there are numerous tasks that can be applied the hybrid conception, we choose two of them to investigate — the 2D selecting task and 3D manipulation / navigation task, we proposed *RubberEdge* and *Haptic Boundary* to facilitate each of them. Both techniques are based on the hybrid isotonic position and elastic rate control conception to benefit the intuitiveness of the former as well as the self-center property, saving input area and smoother trajectory provided by the latter.

For the 2D selecting task, the major problem of the traditional isotonic position control input is the indispensable clutching which could increase dramatically when using a small input area, as the one of touch pad, to interact with a rather large or high-resolution screen. Too many clutching can badly affect the user performance. The *Pointer Acceleration* which changes the *CD gain* according to the mouse motion speed is proposed to reduce the clutching number, but with a small input area, its benefits is not clear. Another existing solution "elastic rate control" is less intuitive, but fundamentally avoids the clutching and

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

allows using a small input area. The techniques based on hybrid position-rate control were also proposed but the majority of them did not provide the force feedback, so make it difficult to distinguish the two control modes.

Our *RubberEdge* is an input device prototype with a small input area. It is based on the *Bubble* technique which has same isotonic position and elastic rate control foundation. However *Bubble* used the straightforward mapping function which produces a discontinuity of cursor speed during the mode transition form the isotonic position to elastic rate. Our Rubber Edge avoids the discontinuity of both direction and norm of speed by rotating and translating the isotonic-to-elastic boundary after the transition according to the theorem of angular momentum and smooth the velocity by mixing pre- and post- transition velocities. We simulated our *RubberEdge* design on the commercial input device *PHANToM* whose programmable force feedback property allows rapid prototyping.

To evaluate our *RubberEdge* technique, we performed a two dimensional selecting task. The result showed that *RubberEdge* outperformed the *Position*-only control by 20% on selection time with a small input area. We also found that the *Pointer Acceleration* outperformed the constant *CD gain* by 7.5%, which was not seen by past researchers. However, a poor regression fitness about *Fitts' law* is found which suggests that *Fitts' Law* may not hold in the presence of significant clutching and for a technique combining two different control mappings. It seems that the distance between the cursor and the target alone affects the selection time, the effect of target width is not significant. We then examined the number of clutch in *Position* control, and observed that the users only used about 75% of the maximum available operating range for each clutch. According to user questionnaire, a half of participants preferred the *RubberEdge* prototype for its rapidity to cover the long distances. Other participants preferred *Position* for its accuracy, and they felt difficult to accurately exit the elastic zone and fine-tune their selection with *RubberEdge*.

It seems that the *Fitts' law* was not held in the evaluated techniques — *position* control with a significant clutching number as well as the hybrid position-rate control of *RubberEdge*. Under this condition, the user performance can not be estimated by using *Fitts' Law*. To address this problem, we developed two formal models to predict the selection time with *position + clutching* and the one with a hybrid position-rate control technique like *RubberEdge*. We then tested their validity by comparing the selection time predicted through models with the measured time in our experiment. The parameters in the models are determined by the experiment configuration and the data from literatures.

We found that our two models have a better fitness than the Fitts' Law. With the two models, we can sum up in what situation the hybrid control of *RubberEdge* outperforms *Position* control. After analyzing the models with different parameter values, we discovered a potential drawback of hybrid control, which is the lower performance elastic rate control will degrade the one of hybrid control for the closer target where the *Position* control which only needs one or two clutching performs better. The standard clutching should be provided to allow more flexibility.

Based on this idea, we made a low cost prototype having a 2D round shaped isotonic zone of a $30mm$ diameter and a ring-shaped elastic zone of $2mm$ width to improve the performance of touch pad of notebook computers. Both clutching inside the isotonic zone and the elastic rate control are provided with this prototype. A two-step calibration is per-

© 2009 Tous droits réservés.

formed for new user to minimize the imperfection of our fabrication and the non-uniform way in which each user's finger contacts positions around the boundary. We performed an initial evaluation of common Windows tasks with four participants for this prototype. Participants immediately grasped the interaction method, they like using rate control for continuous movement of cursor over far distances and appreciated the ability to use the isotonic zone for tasks like drawing.

For the 3D virtual environment, there is not yet a widely acceptable interaction system which provides both object manipulations and environment explorations. The widget-based techniques allow multiple functions but decomposed the interactions into sequential actions which can increase user's mental load. Several existing input devices allow users to perform different interaction in a similar manner and use embedded buttons or other explicit switch mechanisms to distinguish different modes. This explicit mode switching causes inconsistent responses to the same input which requires more user mental efforts. Several two handed input devices were proposed, 3 separate DOF device *Cubic mouse* only perform well in exocentric navigation, which limits its applicability. *YoYo* and *SquareBone* principally use the elastic sensor to interact, which can be less accurate. The system used an input device in each hand is based on Guiard's kinematic chain model, and should produce higher user performance. However the necessary asymmetric model of two hand motions is not always respected in the desktop system. That is a reason why some researches comparing bimanual to unimaual interfaces did not demonstrate the advantage performance of the former.

Our *Haptic Boundary* is technique designed for a one-handed device of small size allowing both object manipulation and environment exploration for desktop system. It is based on the *BubbleCam* version of *Bubble* technique which does not allow rotation of camera. *Haptic Boundary* improved *BubbleCam* on this point. *Haptic Boundary* allow intuitive object manipulation with position control inside the central isotonic zone, and allows effective view manipulations when interacting with the boundary using rate control, different force feedbacks are provided to help users to distinguish different control modes. These force feedbacks serve as the implicit switch between control modes, additionally, users perform different actions in different modes, so will not get confused.

To well explain the development of this primary idea, we first gave some necessary definitions and a classic algorithm which clarified the relationship between the end effector in physical workspace and the cursor in the camera system within the virtual environment. This algorithm determines the cursor position when given the camera position/orientation and end effector position. *Haptic Boundary* focus on the camera position and orientation, which are then used to determine cursor position.

To provide suitable feedbacks to different camera motions, two kinds of force feedbacks, *sticky force* and *wall force*, were examined, their advantages and disadvantages when performing translation and rotation motions were analyzed. Based on this analysis, we decided to use *sticky force* to perform rotation and translation of camera, and the wall force to perform pure translation of camera. We then apply the two kinds force feedbacks on different surface of *Haptic Bounday*.

Then we investigated the size and shape of isotonic zone. To determine the size, we performed an informal thinking aloud pilot experiment of elemental tasks in 3D VE, with

© 2009 Tous droits réservés.

several participants. The size is fixed at a dimension of about $40mm * 40mm * 40mm$ according to our choosing criterion: the user does not feel much constraint when inside the isotonic zone, and the user can easily reach the boundary to perform translation and rotation of camera. For the choice of shape of isotonic zone based on the determined volume size, we performed an analysis and carried out an experiment to compare several simple shapes. We finally chose the semi ellipsoid with ceiling and floor as the shape of isotonic zone, because it gives enough usable workspace based on the fixed dimension, and more importantly, its convex surface and three planes can respectively favor the sticky and wall force rate control motion. We then clarify the haptic property of each surface of the isotonic zone.

After determined isotonic zone, we described the implementation of *Haptic Boundary*. To give an optimal exploration performance, the *Haptic Boundary* is carefully developed, via an evolutions of several improvements of algorithm. We started from a similar idea to the one of *RubberEdge* — mechanics. To simplify rotation calculation, we used an complete ellipsoid of same equatorial radii and polar radius to cover the isotonic zone. The rotation of the complete ellipsoid produced by the force applied on the convex surface is calculated via the Euler dynamical equations. The translation varies with the applied force. However, when the calculated rotation is applied to camera, some unwanted motions comes out: In certain area of the convex surface, an applied force containing upward component produces a downward camera rotation. Additionally, the forward translation can be performed only within the front middle area of the convex surface, outside this area, the camera will rotate. However, the user just moves forward the end effector in order to get a camera forward movement, without being conscious where he is interacting. The two reasons make the forward movement difficult. To address this problem, we use the cross product of unit vector (0,0,-1) and the applied force to calculate the camera rotation.

With current implementation, the user can navigate in virtual environment, but the rotating camera around an object and the lateral translations can not be easily performed. In the new solution, we limited the camera translation in three directions in three vectors (1,0,0), (-1,0,0) and (0,0,-1), according to the sticky point location and the user's forward movement desire extracted from the trajectory of end effector. The translation direction vector was also used in rotation calculation. Both object around rotations and lateral translations are provided in this new implementation.

We then distribute the norm of applied sticky force between rotation velocity and translation velocity of camera, which allows users to easily control the curvature of the camera trajectory by simply moving the end effector, without worrying about the distance between the sticky point and the end effector.

We then developed a simple but complete interaction system in which the *Haptic Boundary* is applied. This system is composed of several parts: *calculator*, *input device*, *interaction*. In the *calculator* part, the *Exploration* subpart updates the camera orientation and position by applying the *Haptic boundary* technique; the *camera cursor system* subpart produces the cursor position.

We also designed the virtual feedback for the novice users who are not familiar with the haptic feedback input device and have some difficulty to perceive the isotonic zone.

To evaluate our *Haptic Boundary* technique, we performed an formal evaluation of car

© 2009 Tous droits réservés.

resembling task with 3 participants. The result showed that *Haptic Boundary* outperformed the *Unimanual explicit switch* interface by 50% in task completion time. Additionally, *Haptic Boundary* is more efficient than *Unimanual explicit switch* in translation and more efficient than *Bimanual Interface* in rotation.

Among the three participants, one preferred the *Haptic Boundary*, he said that with *Haptic Boudnary*, he did not have to learn the synchronization between two hands like with *Bimanual Interface*. The two others preferred the *Bimanual Interface*, they like the collaboration between two hands, and felt that it needs a lot of training to master the *Haptic Boundary*. The participants rated *Haptic Boundary* as the one which brought least hand strain and dizziness.

An observation from the usage of participants is that they still have some difficulty to rotate the camera around the object, may be due to the two sequential gestures required to perform such motions. So we consider to combine another mode — *Orbiting Inspection* with the *Haptic Boundary* to facilitate the rotation around any object. When the user doubleclicks on an object, this object is selected as the *point of interest* (*POI*), *Orbiting Inspection* is activated, an animation is launched to turn the camera towards the *POI* object. After that any lateral and longitudinal component of applied force will move the camera along the sphere surface surrounding the *POI* object, any posteroanterior component changes the distance between camera and *POI* object. The camera will keep orienting towards the *POI* object till the user exits orbiting inspection mode by doubleclicking anywhere and returns to *Haptic Boundary* mode.

Future Work

The work presented in this thesis brought several interesting possibility of further work.

For the *RubberEdge*, the further iteration of our current prototype should be carried out to make it suited to more diverse applications. The two performance models for *Position* control and *hybrid position-rate* control could be optimized by finding other independent factors.

For the *Haptic Boundary*, more experiments should be performed including the one requiring more camera displacement to test the exploration performance of *Haptic Boundary*. For example with a task which asks the participants to start navigation from the initial room, pass though a tunnel and find a specified object in another room and then return to the initial room. More studies should be performed to well combine the *Haptic Boundary* and *Orbiting Inspection*.

Other hybrid isotonic position elastic rate control could be investigated. Hybrid conception can be combine with two handed input devices, for example, connecting your two wrist through a rubber band. The two hands can move freely and comfortably in a certain range. But when they are far enough away from each other, the rubber band is stretched, an elastic feedback is provided. The isotonic positioning control and elastic rate control can be applied to different separation between hands. The camera and object

© 2009 Tous droits réservés.

manipulation technique can be designed in this condition. The advantage of this idea is that it benefits the intuitiveness of bimanual interface which is supported by Guiard's Kinematics Chain Model, and may enhance the performance of hybrid conception.

© 2009 Tous droits réservés.

http://www.univ-lille1.fr/bustl

Thèse de Qing Pan, Lille 1, 2008

Appendix A

Quaternion

A.1 Definition

Quaternion is first described by Irish mathematician William Rowan Hamilton in 1843. Quaternion is used for 3D camera control, compact storage and smooth 3D interpolation in 3D computer graphics. Quaternion is a hyper-complex number with one real part and three imaginary part. Quaternion can be represented in the form of equation A.1,

$$q = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k} \tag{A.1}$$

where w is the real part, $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ are imaginary numbers that form the vector basis of the quaternion q, x, y and z are the coefficient of each items. $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ can be considered as vectors or variables. They have following relationships:

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{i} \times \mathbf{j} \times \mathbf{k} = -1 \tag{A.2}$$

$$\mathbf{i} = \mathbf{j} \times \mathbf{k} = -\mathbf{k} \times \mathbf{j} \tag{A.3}$$

$$\mathbf{j} = \mathbf{k} \times \mathbf{i} = -\mathbf{i} \times \mathbf{k} \tag{A.4}$$

$$\mathbf{k} = \mathbf{i} \times \mathbf{j} = -\mathbf{j} \times \mathbf{i} \tag{A.5}$$

A.2 Arithmetic Operations of Quaternion

The mathematics of addition, multiplication, inverses and so on of quaternion are all the same as in standard complex number theory, but with more elements.

Quaternion Multiplication

Given:

$$p = w_0 + x_0\mathbf{i} + y_0\mathbf{j} + z_0\mathbf{k} = w_(0) + \overrightarrow{p_v}q = w_1 + x_1\mathbf{i} + y_1\mathbf{j} + z_1\mathbf{k} = w_(1) + \overrightarrow{q_v} \tag{A.6}$$

where $\vec{p_v}$ and $\vec{q_v}$ the vectors representing the imaginary part of p and q.

$$p * q = (w_0 * w_1 - (\vec{p_v} \cdot \vec{q_v})) + (w_0 * \vec{q_v} + w_1 * \vec{p_v} + \vec{p_v} \times \vec{q_v}) \tag{A.7}$$

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

Thèse de Qing Pan, Lille 1, 2008

Quaternion Conjugate

Given

$$q = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k} = w + \vec{p_v} \tag{A.8}$$

the conjugate of q, q^* is:

$$q = w - x\mathbf{i} - y\mathbf{j} - z\mathbf{k} = w - \vec{p_v} \tag{A.9}$$

The Quaternion-Conjugate Product

$$q * q^* = w^2 + x^2 + y^2 + z^2 \tag{A.10}$$

Norm of Quaternion

$$|q| = \sqrt{w^2 + x^2 + y^2 + z^2} = \sqrt{q * q^*} \tag{A.11}$$

The multiplicative inverse of the quaternion q is

$$q^{-1} = q^*/|q|^2 \tag{A.12}$$

If $|q| = 1$, $q^{-1} = q^*$. Most of time, we assume that all quaternions are unit quaternions, in order to use this simplified form.

A.3 Applications of Quaternions

Unit quaternion $q = cos(\frac{\theta}{2}) + sin(\frac{\theta}{2}) * \vec{V_q}$ defines both the axis of rotation and the angle θ to rotate about the axis $\vec{V_q}$.

Rotation a Vector

Given a vector $\vec{v} = x, y, z$, we can transform it into its quaternion form: $v = 0 + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$. To rotate v with the unit quaternion q:

- clockwise rotation in right-handed system $v' = q^* * v * q$
- counter-clockwise rotation in right-handed system $v' = q * v * q^*$
- clockwise rotation in left-handed system $v' = q * v * q^*$
- counter-clockwise rotation in left-handed system $v' = q^* * v * q$

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

Thèse de Qing Pan, Lille 1, 2008

Euler Angles to Quaternion

Given pitch angle x_θ, yaw angle y_θ, roll angle z_θ, the corresponding quaternion to these euler angle are :

$$q_{x\theta} = cos(\frac{x_\theta}{2}) + sin(\frac{x_\theta}{2})\mathbf{i} \tag{A.13}$$

$$q_{y\theta} = cos(\frac{y_\theta}{2}) + sin(\frac{y_\theta}{2})\mathbf{j} \tag{A.14}$$

$$q_{z\theta} = cos(\frac{z_\theta}{2}) + sin(\frac{z_\theta}{2})\mathbf{k} \tag{A.15}$$

The rotation quaternion q_{final} can represented in 6 forms:

$$q_{final} = q_{x\theta} * q_{y\theta} * q_{z\theta} \tag{A.16}$$

$$= q_{x\theta} * q_{z\theta} * q_{y\theta} \tag{A.17}$$

$$= q_{y\theta} * q_{x\theta} * q_{z\theta} \tag{A.18}$$

$$= q_{x\theta} * q_{z\theta} * q_{x\theta} \tag{A.19}$$

$$= q_{z\theta} * q_{x\theta} * q_{y\theta} \tag{A.20}$$

$$= q_{z\theta} * q_{y\theta} * q_{x\theta} \tag{A.21}$$

Matrices to Quaternion

Given a homogeneous matrix

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{A.22}$$

- Calculate the trace of the matrix $t = a_{11} + a_{22} + a_{33} + 1$
- If the trace is above zero, to calculate the quaterion $q = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$

$$s = 0.5/\sqrt{t} \tag{A.23}$$

$$x = (a_{32} - a_{23}) * s \tag{A.24}$$

$$y = (a_{13} - a_{31}) * s \tag{A.25}$$

$$z = (a_{21} - a_{12}) * s \tag{A.26}$$

$$w = 0.25/s \tag{A.27}$$

- If the trace of the matrix is under zero, according to the column in which locates the greatest value element of the diagonal:
  Column 1:

$$s = 2 * \sqrt{1 + a_{11} - a_{22} - a_{33}} \tag{A.28}$$

$$x = 0.5/s \tag{A.29}$$

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

$$y = (a_{21} + a_{12})/s \tag{A.30}$$
$$z = (a_{13} + a_{31})/s \tag{A.31}$$
$$w = (a_{32} + a_{23})/s \tag{A.32}$$

Column 2:

$$s = 2 * \sqrt{1 + a_{22} - a_{11} - a_{33}} \tag{A.33}$$
$$x = (a_{21} + a_{12})/s \tag{A.34}$$
$$y = 0.5/s \tag{A.35}$$
$$z = (a_{32} + a_{23})/s \tag{A.36}$$
$$w = (a_{13} + a_{31})/s \tag{A.37}$$

Column 3:

$$s = 2 * \sqrt{1 + a_{33} - a_{11} - a_{22}} \tag{A.38}$$
$$x = (a_{13} + a_{31})/s \tag{A.39}$$
$$y = (a_{32} + a_{23})/s \tag{A.40}$$
$$z = 0.5/s \tag{A.41}$$
$$w = (a_{21} + a_{12})/s \tag{A.42}$$

© 2009 Tous droits réservés.

Appendix B

Empirical Research in HCI

Empirical research is observation-based investigation seeking to discover and interpret facts, theories, or laws. Empirical research is trying to answer the following questions about a new or existing UI design or interaction technique:

- Is it viable?
- Is it as good as or better than current practice?
- Which of several design alternative is best?
- What are its performance limits and capabilities?
- What are its strengths and weaknesses?
- Does it work well for novices, for experts?
- How much practice is required to become proficient?

The testable question should be multiple narrow and cover the range of outcomes influencing the broader untestable questions, for example, a technique that is faster, is more accurate, takes fewer steps, is easy to learn and is easy to remember (testable) is generally better (untestable).

B.1 Procedure

We want to know if the measured performance on a dependent variable (e.g., speed) is different between test conditions, so

- We conduct a user study and measure the performance on each test condition over a group of participants.
- For each test condition, we compute the mean score over the group of participants.
- Answer the questions: Is there a difference? Is the difference large or small? Is the difference significant or is it due to a chance. The third question can be answered by using the statistical tool — the analysis of variance (anova).

© 2009 Tous droits réservés.

Analysis of Variance

Analysis of variance is used to determine if there is a significant difference between the means. “Significant” implies that in all likelihood the difference observed is due to the test conditions. “No significant” implies that the difference observed is likely due to a chance.

B.2 Terminology

Participant

Participants are the people participating in an experiment.

Independent Variable

An independent variable, also called factor is a variable that is selected or manipulated through the design of the experiment. For example, device, feedback mode, button layout, visual layout, gender, age, expertise, etc.

Test Conditions

Test conditions are the values or settings fro an independent variable (levels or treatments)

Dependent Variable

A dependent variable is a variable representing the measurement or observations on an independent variable. Example include task completion time, speed, accuracy, error rate, throughput.

Confounding Variable

A confounding variable is any variable that varies systematically with an independent variable. For example, if 3 devices are always administered in the same order, participant performance might improve due to practice. “Practice” varies systematically with “device” , so it is a confounding variable.

Within Subjects, Between Subjects

The administering of levels of a factor is either within subject or between subject. If each participant is tested on each level, the factor is within subjects. The terms repeated measures and within subjects are synonymous. If each participant is tested on only one level, the factor is between subjects. In this case, a separate group of participants is used for each condition.

Repeated measures advantage is that the variance due to participants’ pre-dispositions should be the same across test conditions. Between subjects advantage is that it can avoid interference effects.

© 2009 Tous droits réservés.

Thèse de Qing Pan, Lille 1, 2008

Experiment Design

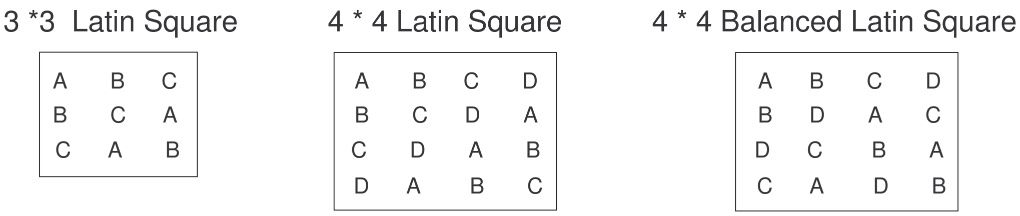
Experiment Design is general term referring to the organization of factors, levels, procedure, etc., in an experiment. For example, "3×2 repeated-measures design" refers to an experiment with two factors, having 3 levels on the first and 2 levels on the second. There are 6 test conditions in total. Both factors are repeated measures, meaning all participants were tested on all test conditions.

Counterbalancing

For repeated measures designs, participants' performance may tend to improve with practice as they progress from one test condition to the next. To compensate, the order of presenting conditions should be counterbalanced: Participants are divided into groups, and a different order of administration is used for each group. The order is best governed by a Latin Square.

Latin Square

The defining characteristic of a latin square is that each condition occurs only once in each row and column. Figure B.1 give some examples.



3 *3 Latin Square

A	B	C
B	C	A
C	A	B

4 * 4 Latin Square

A	B	C	D
B	C	D	A
C	D	A	B
D	A	B	C

4 * 4 Balanced Latin Square

A	B	C	D
B	D	A	C
D	C	B	A
C	A	D	B

Figure B.1: Latin Square for counterbalancing

© 2009 Tous droits réservés.

http://www.univ-lille1.fr/bustl

© 2009 Tous droits réservés.

Bibliography

[Alb03] Albinsson, P. and Zhai, S. (2003). High precision touch screen interaction. ACM CHI Conference on Human Factors in Computing Systems. p. 105-112.

[Ali94] Alias Reserach, Inc. ALIAS Reference Guide Toronto Ontario, Canada,1994

[AZ01] Accot, J., and Zhai, S. Scale effects in steering law tasks. In Proc. of CHI'01, 2001, pp. 1-8.

[BB*06] A. Bornik, R. Beichel, E. Kruijff, B. Reitinger, and D. Schmalstieg. A hybrid user interface for manipulation of volumetric medical data. In Proceedings of IEEE Symposium on 3D User Interfaces 2006. IEEE, March 2006. in print.

[Bea01] Beaudouin-Lafon, M. Novel interaction techniques for overlapping windows. In Proc. of UIST'01, 2001, pp. 153-154.

[BG95] Balaguer, J. F., Gobbetti, E.: i3D: A High-Speed 3D Web Browser. Proc. of VRML'95 Symposium, San Diego, 1995. (http:// www.crs4.it/ gobetti/)

[BGB04] Blanch, R., Guiard, Y. and Beaudouin-Lafon, M. (2004). Semantic pointing: improving target acquisition with control-display ratio adaptation. CHI, 519-526.

[BH97] Bowman, Doug A. Hodges, Larry (1997) "An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments" Proceedings of the 1997 symposium on Interactive 3D graphics, pp. 35.

[BH99] R. Balakrishnan and K. Hinckley. The role of kinesthetic reference frames in two-handed input performance. In Proceedings of UIST'99, pp. 171-178, 1999.

[Bie86] Eric Allan Bier, Skitters and Jacks: Interactive 3D Positioning Tools 1986

[BH*96] Bederson, B., Hollan, J., Perlin, K., Meyer, J., Bacon, D., and Furnas, G. Pad++: A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics, Journal of Visual Languages and Computing, 7 (1996), pp.3-31.

[BJH99] D. Bowman, D. B. Johnson, and L. F. Hodges, Testbed Evaluation of Virtual Environment Interaction, Proc. of ACM VRST'99, 1999, pp. 26-33.

[BK99] Ravin Balakrishnan and Gordon Kurtenbach, Exploring Bimanual Camera Control and Object Manipulation in 3D Graphics Interfaces, 1999.

© 2009 Tous droits réservés.

Thèse de Qing Pan, Lille 1, 2008

[BKH98] D. Bowman, D. Koller, L. Hodges (1998) A methodology for the evaluation of travel techniques for immersive virtual environments. Virtual Reality: Research, Development, and Applications 3, pp. 120-131.

[BKLP99] Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola, Jr., Ivan Poupyrev, An Introduction to 3D User Interface Design. 1999

[BL99] BARES W. H., LESTER J. C.: Intelligent Multi-Shot Visualization Interfaces for Dynamic 3D Worlds. In IUI'99: Proceedings of the 4th international conference on Intelligent user interfaces (New York, NY, USA, 1999), ACM Press, pp. 119-126.

[BM86] Buxton, W. and Myers, B. "A Study in Two-Handed Input," in Proceedings S1GCHI'86: Human Factors in Computing Systems. 1986. Boston, MA: pp. 321-326.

[BM97] Balakrishnan, R., and MacKenzie, I. S. Performance differences in the fingers, wrist, and forearm in computer input control. In Proc. of CHI'97, 1997, pp. 303-310

[Bow99] Bowman, Interaction Techniques For Common Tasks In Immersive Virtual Environments Design, Evaluation, And Application, thesis ,1999

[BP98] Balakrishnan, R. and Patel, P. "The PadMouse: Facilitating Selection and Spatial Positioning for the Non-Dominant Hand," in Proceedings SIGCHI'98: Human Factors in Computing Systems. 1998. Los Angeles, CA: pp. 9-16.

[Bro88] Brooks, F., "Grasping Reality Through Illusion: Interactive Graphics Serving Science," CHI'88.

[BS*92] D. Brookshire Conner, Scott S. Snibbe, Kenneth P. Herndon, Daniel C. Robbins, Robert C. Zeleznik, and Andries Van Dam. Three-dimensional widgets. In Proceedings of ACM Symposium on Interactive 3D Graphics (I3D), pp. 183-188, 1992.

[Bux83] Buxton, W. Lexical and Pragmatic Considerations of Input Structures. Comp. Graphics, 17,1 (1983), pp.31-37.

[BWB06] Benko, H., Wilson, A.D. and Baudisch, P. (2006). Precise selection techniques for multi-touch screens. CHI, pp.1263-1272.

[Cas04] G. Casiez, "Contribution à l'étude des interfaces haptiques - le DigiHaptic : un périphérique haptique de bureau degrés de liberté séparé|s." PhD Thesis. Lille, University of Lille, 2004.

[CEB78] Card, C., English, W., and Burr, B. Evaluation of mouse, rate controlled isometric joystick, step keys, and text keys for text selection on a crt. Ergonomics, 21 (1978), pp.601-613.

[CFH97] L. D. Cutler, B. Froehlich, and P. Hanrahan. Two-handed direct manipulation on the responsive workbench. 1997 Symposium on Interactive 3D Graphics, 1997.

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

[CGB99] Didier Casalta, Yves Guiard, and Michel Beaudouin-Lafon. Evaluating two-handed input techniques: Rectangle editing and navigation. ACM CHI'99 Conference (Extended Abstracts), pp. 236-237, 1999.

[CH93] C. Carlsson and 0. Hagsand. "DIVE-a Multi-User Virtual Reality System." In Proceedings of IEEE VRAES, pages 394-400, 1993.

[Cli] Michael Cline, Higher Degree-of-Freedom Bimanual User Interfaces for 3-D Computer Graphics

[CMS88] Michael Chen, S. Joy Mountfurd, Abigail Sellen, A Study in Interactive 3-D Rotation Using 2-D Control Devices, Computer Graphics, Volume 22, Number 4, August 1988

[CNP05] J. Chen, M.A. Narayan, and M.A. Perez-Quinones. "The Use of Hand-held Devices for Search Tasks in Virtual Environments," In Proc. of the IEEE Virtual Reality 2005 workshop on New Directions in 3DUI, pp. 15-18, March 2005

[CP*03] CASIEZ G., PLENACOSTE P., CHAILLOU C., SEMAIL B.: The DigiHaptic, a new three degrees of freedom multi-finger haptic device. In Virtual Reality International Conference (May 2003), pp. 35-39.

[DC*04] Joan De Boeck, Erwin Cuppens, Tom De Weyer, Chris Raymaekers, and Karin Coninx. Multisensory interaction metaphors with haptics and proprioception in virtual environments. In Proceedings of NordiCHI 2004, Tampere, FI, October 2004.

[DC02] Joan De Boeck, Karin Coninx, Haptic Camera Manipulation: Extending the Camera In Hand Metaphor, 2002.

[DET90] Dillion, R. F., Edey, J. D., Tombaugh, J. W. Measuring the true cost of command selection: Techniques and Results. Proceedings of the CHI'90 Conference on Human Factors in Computing Systems (pp. 19-25). New York ACM, 1990

[DL*05] Lionel Dominjon, Anatole Lécuyer, Jean-Marie Burkhardt, Guillermo Andrade-Barroso, Simon Richir, The "Bubble" Technique: Interacting with Large Virtual Environments Using Haptic Devices with Limited Workspace, Proceedings of World Haptics Conference (joint Eurohaptics Conference and Haptics Symposium), Pisa, Italy, 2005

[DM94] Douglas, A. and Mithal, A. The effect of reducing homing time on the speed of a finger-controlled isometric pointing device. In Proc. of CHI'94, 1994, pp. 411-416.

[DMH96] David Koller, Mark Mine, and Scott Hudson. Head-tracked orbital viewing: An interaction technique for immersive virtual environments. In Proceedings of the ACM Symposium on User Interface Software and Technology (UIST) 1996, Seattle, Washington, USA, 1996.

[DS93] Darken, R. and Sibert, J. (1993), A Toolset for Navigation in Virtual Environments, in Proc. UIST 93', ACM Press, pp. 157-165.

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

[Edg] Synaptics, EdgeMotion. http://www.synaptics.com.

[Ele] Electronic Visualization Laboratory at the University of Illinois at Chicago. Creators of the Immersadesk, demonstrated at SIGGRAPH in 1994. Information available at URL: http://evlweb.eecs.uic.edu/EVL/VR/systems.html as well as URL: http://www.eecs.uic.edu/ mczernus/Immersadesk/immersadesk.html

[Fak] Fakespace, Inc. URL: http://www.fakespace.com/.

[FH*06] B. Frohlich, J. Hochstrate, A. Kulik, A. Huckauf, "On 3D Input Devices". IEEE Computer Graphics and Applications, vol. 26, no. 2, pp. 15-19, March/April 2006.

[FHZ96] A. Forsberg, K. Herndon, and R. Zeleznik. Aperture based selection for immersive virtual environment. In Proceedings of UIST96, pages 95-96 1996.

[FIB95] Fitzmaurice, G., Ishii, H., Buxton, W., "Bricks: Laying the Foundations for Graspable User Interfaces," Proceedings of CHI'95: ACM Conference on Human Factors in Computing Systems, 1995, pp. 442-449.

[FP00] Frohlich, B. and Plate, J, The CubicMouse: A new device for 3D input, Proceedings ACM CHI, 2000.

[Fro06] B. Froehlich et al., The GlobeFish and the GlobeMouse,Proc. Computer-human Interaction 2006, ACM Press, to appear in Apr. 2006.

[Fur86] Furnas, G. Generalized Fisheye Views. Proceedings of CHI'86, ACM Press, pp.16-23.

[FWC84] Foley, J. D., Wallace, V. L. Chan, P. (1984).The Human Factors of Computer Graphics Interaction Techniques. IEEE Computer Graphics and Applications 4(11), pp.13-47.

[GB05] T. Grossman and R. Balakrishnan. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In CHI'05: Proceeding of the SIGCHI conference on Human factors in computing systems, pages 281-290, New York, NY, USA, 2005. ACM Press.

[Gri99] Maarten Wim GRIBNAU. Two-handed interaction in computer supported 3D conceptual modeling. Thesis 1999

[Gui87] Guiard, Y., "Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model." Journal of Motor Behavior, 1987.19(4): pp. 486-517.

[Han97] Hand, C., "Survey of 3-D Interaction Techniques", Computer Graphics Forum, vol. 16, pp. 269-281, 1997.

[Hau89] Hauptmann, A., "speech and Gestures for Graphic Image Manipulation," Proc. CHI'89, pp. 241-245.

© 2009 Tous droits réservés.

[HC*02a] Ken Hinckley, Edward Cutrell, Steve Bathiche, and Tim Muss, Quantitative Analysis of Scrolling Techniques,CHI 2002, April 20-25, 2002, Minneapolis, Minnesota, USA, 2002.

[HC*02b] Hinckley K, Cutrell E, Bathiche S, Muss T, 2002 “Quantitative Analysis of Scrolling Techniques”, in Proceedings of CHI’2002 Conference on Human Factors in Computing Systems (Minneapolis, Minnesota, 20–25 April) pp 65-72

[HCS98] Ken Hinckley, Mary Czerwinski, and Mike Sinclair. Interaction and modeling techniques for desktop two handed input. In UIST’98 Proceedings, 1998.

[HCP05] A. Huckauf, M.F. Costabile and F. Paterno, eds., Evaluation of 12-DOF Input Devices for Navigation and Manipulation in Virtual Environments, Human-Computer Interaction: Proc. Interact, 2005.

[HD*08] Hachet, M. Decle, F. Knodel, S. Guitton, P. Navidget for Easy 3D Camera Positioning from 2D Inputs, 3D User Interfaces IEEE Symposium 2008 pp. 83-89, Reno, NE

[HD*95] Hinckley, K., Pausch, R. Downs, J.H., Proffitt, D., and Kassell, N.F, The Props-Based Interface for Neurosurgical Visualization, 1995.

[HKP02] Haan, G., Koutek, M., Post, F. 2002, Towards Intuitive Exploration Tools For Data Visualization In Vr, Proceedings of the ACM symposium on Virtual reality software and technology, 105-112

[HJW04] K. Hinckley, R.J.K. Jacob, and C. Ware, “Input/output Devices and Interaction Techniques,” in The Computer Science Handbook, Second Edition, ed. by A.B. Tucker, pp.20.1-20.32, Chapman and Hall/CRC Press (2004).

[HP*97] Hinckley, K., Pausch, R., Proffitt, D., Kassell, N., ”Attention and Visual Feedback: The Bimanual Frameof-Reference,” ACM/SIGGRAPH Symposium on Interactive 3D Graphics, 1997, 121-126.

[Hou92] Stephanie Houde, Iterative Design of an Interface for Easy 3-D Direct Manipulation, 1992 ACM CHI

[HSH04] Knud Henriksen, Jon Sporring, and Kasper Hornbæk, Virtual Trackballs Revisited,Ieee Transactions On Visualization And Computer Graphics, VOL. 10, NO. 2, MARCH/APRIL 2004

[HTP97] Hinckley, K., Tullio,J., and Pausch, R. Usability analysis of 3D rotation techniques. UIST’97 (1997), pp. 1-10.

[HW97] Hanson, A. and Wernert, E. (1997), Constrained 3D Navigation with 2D Controllers, in Proc. Visualization 97’, IEEE Press, pp. 175-182.

[IH00] Igarashi, T., Hinckley, K. Speed-dependent Automatic Zooming for Browsing Large Documents, Proceedings of UIST 2000, ACM Press, pp.139-148.

© 2009 Tous droits réservés.

[IM92] Iwata, H. And Matsuda, K. 1992. Haptic walkthrough simulator: Its design and application to studies on cognitive map. In the 2nd International Conference on Artificial Reality and Tele-existence. ICAT 92. pp. 185-192.

[JC90] Jellinek, H. and Card , S. Powermice and user performance In. Proc. of CHI'90, 1990, pp. 213-220.

[Joh94] Johnsgard, T. Fitts' Law with a virtual reality glove and a mouse: Effects of gain. In Proc. of Graphics Interface Conference, 1994, pp. 8-15.

[JS*94] Robert J. K. Jacob, Linda E. Sibert, Daniel C. Mcfarlane, And M. Preston Mullen, Jr. Integrality and Separability of Input Devices. ACM Transactions on Computer-Human Interaction, Vol. 1, No. 1, March 1994

[KB91] Gordon Kurtenbach and William Buxton. Issues in combining marking and direct manipulation techniques. Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'91, S. Carolina, November), ACM, 1991, pp. 137-144.

[KBF06] Kulik, A.,Blach, R.,Frohlich, B., "two - 4 - six" - A Handheld Device for 3D-Presentations, 3D User Interfaces, 2006. 3DUI 2006. IEEE Symposium on Publication Date: March 25-26, 2006 On page(s): pp.167-170

[KBS94] Kabbash, P., Buxton, W., Sellen, A. (1994). Two handed input in a compound task. Proceedings of the CHI'94 Conference, pp.417-423, ACM.

[KCT05] Kruger, R., Carpendale, S., Scott, S.D., Tang, A. Fluid integration of rotation and translation . Proceedings CHI 2005.

[KF*98] Owen, R., Kurtenbach, G., Fitzmaurice, G., Baudel, T. and Buxton, W. (1998), Bimanual Manipulation in a Curve Editing Task. Unpublished manuscript. Toronto: Alias|Wavefront Inc.

[KI98] Kawachiya, K., and Ishikawa, H. NaviPoint: An input device for mobile information browsing, In Proceedings of the CHI '98 Conference on Human Factors in Computing Systems. New York: ACM, 1998, pp. 1-8

[KK*05] Azam Khan, Ben Komalo, Jos Stam, George Fitzmaurice, and Gordon Kurtenbach. HoverCam: Interactive 3D navigation for proximal object inspection. In Proceedings of ACM Symposium on Interactive 3D Graphics (I3D), 2005.

[Kru91] Krueger, M. Artificial Reality II. Addison-Wesley, Reading, MA, 1991.

[Kur97] Kurtenbach, G., et al. "The Design of a GUI Paradigm based on Tablets, Two-hands, and Transparency," in Proceedings, CH1'97: Human Factors in Computing Systems. 1997. Atlanta, GA: ACM. pp. 35-42.

[KW03] R. Komerska et C. Ware, "Haptic Task Constraints for 3D Interaction", Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, Los Angeles, California, U.S.A., 2003.

© 2009 Tous droits réservés.

[Lat04] Celine Latulipe. Symmetric interaction in the user interface. In UIST 2004 Companion Proceedings. ACM Press, 2004.

[LCF76] Langolf, G. D., Chaffin, D. B. and Foulke, J.A.. An investigation of Fitts' law using a wide range of movement amplitudes. Journal of Motor Behavior, 8 (1976), pp. 113-128.

[LG93] Liang, J. , Green, M., "JDCAD: A Highly Interactive 3D Modeling System:" 3rd International Conference on CAD and Computer Graphics, Beijing, China, Aug. 1993, pp. 217-222.

[LK*91] LeBlanc A, Kalra P, Magnenat-Thalmann N, Thalmann D (1991) Sculpting With the "Ball and Mouse" Metaphor - Proc. Graphics Interface '91, Calgary, Canada

[LM*06] C. Latulipe, S. Mann, C. S. Kaplan, and C. L. Clarke. symSpline: symmetric two-handed spline manipulation. In CHI'06, ACM Press, April, 2006.

[LP*06] Liu, J., Pinelle, D., Sallam, S., Subramanian, S. Gutwin, C. (2006, June 7th). TNT: improved rotation and translation on digital tables. Proceedings of the 2006 Conference on Graphics interface (Quebec, Canada, 09). ACM International Conference Proceeding Series, 137, pp. 25-32. Canadian Information Processing Society, Toronto, Ont., Canada.

[LZB98] A. Leganchuk, S. Zhai, and W. Buxton. Manual and cognitive bene

ts of two-handed input: An experimental study. ACM Transactions on Computer-Human Interaction, 5(4):326-359, 1998

[Maa99] Maarten W. Gribnau. Two-handed interaction in computer supported 3D conceptual modeling. PhD thesis, Netherland, 1999.

[MA*05] T. Maeda, H. Ando, T. Amemiya, M. Inami, N. Nagaya, and M. Sugimoto, "Shaking the World: Galvanic Vestibular Stimulation as a Novel Sensation Interface", ACM SIGGRAPH, Emerging Technologies, 2005.

[Mac] Claris Corporation, MacPaint. Version 2.0.1985-1987.

[Mac92] MacKenzie, I.S. Fitts' law as a research and design tool in human computer interaction. Human Computer Interaction, 7 (1992), 91-139.

[McC70] McCormick, E. Human factors engineering. 3d edition. McGraw-Hill, 1970, pp. 639

[MBS97] Mine, M., Brooks, F., and Sequin, C. (1997). Moving objects in space: exploiting proprioception in virtual environment interaction. ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques. pp. 19-26.

[MCR90a] Mackinlay, Jock, Card, Stuart K., and Robertson, George G. A Semantic Analysis of the Design Space of Input Devices. Human-Computer Interaction, 5:145-190, 1990.

© 2009 Tous droits réservés.

[MCR90b] Jock D. Mackinlay, Stuart K. Card, and George G. Robertson, Rapid Controlled Movement Through a Virtual 3D Workspace, 1990.

[Min] Mine, M., http://www.cs.unc.edu/ mine/chimp.html.

[Min95] Mine, M. (1995). Virtual Environment Interaction Techniques. UNC Chapel Hill Computer Science Technical Report TR95-018.

[Mit03] Mitchell, G. D. Orientation on Tabletop Displays. M.Sc. Thesis, Simon Fraser University, Burnaby, BC, 2003.

[MLY00] Myers, B., Lie, K., Yang, B. (2000). Two-Handed Input using a PDA and a Mouse. CHI 2000. pp.41-48.

[MM95] D. P. Mapes and J. M. Moshell. A two-handed interface for object manipulation in virtual environments. Presence, 4(4): pp.403-416, 1995

[MM*96] M. Mohageg, R. Myers, C. Marrin, J. Kent, D. Mott, P. Issacs, A user interface for accessing 3D content on the World Wide Web. In: Proceedings of the CHI'96. ACM/SIGCHI, New York, pp. 466-472.

[MO98] MacKenzie, S. and Oniszczak, A. A comparison of three selection techniques for touch pads. In Proc. of CHI'98, 1998, pp. 336-343.

[MR94] MacKenzie, S. and Riddersma, S. Effects of output display and control-display gain on human performance in interactive systems. Behavr. and Info. Tech., 1,3 (1994), pp.328-337.

[MRC91] Jock D. Mackinlay, George G. Robertson, and Stuart K. Card. The perspective wall: Detail and context smoothly integrated. In Proceedings of ACM Conference on Human Factors in Computing Systems (CHI), pp. 173-176, 1991.

[MSB91] MacKenzie, I. S, Sellen , A., and Buxton, W. A comparison of input devices in element pointing and dragging tasks. In Proc. of CHI'91, 1991, pp. 161-166.

[MSP03] Magerkurth, C., Stenzel, R. and Prante, T. STARS - a ubiquitous computing platform for computer augmented tabletop games. In Extended Abstract of UbiComp'03, Springer, (2003), 267-268.

[Mul] MultiGen Inc. "SmartScene™." for more information see: http://www.multigen.com/smart.htm

[NO86] Gregory M. Nielson and Dan R. Olsen Jr. Direct Manipulation Techniques for 3D Objects Using 2D Locator Devices, 1986

[OF03] Olwal, A., and Feiner, S. (2003). "The Flexible Pointer: An Interaction Technique for Augmented and Virtual Reality." Proceedings of ACM Symposium on User Interface Software and Technology, pp. 81-82.

© 2009 Tous droits réservés.

[OK*05] R. Owen, R. Kurtenbach., G. Fitzmaurice, T. Baudel, and B. Buxton, When it gets more difficult, use both hands: exploring bimanual curve manipulation. In Graphics Interface, pages 17-24, 2005.

[OL01] Miguel A. Otaduy and Ming C. Lin, User-Centric Viewpoint Computation for Haptic Exploration and Manipulation, 2001.

[OS05] J.-Y. Oh, W. Stuerzlinger, Moving objects with 2D input devices in CAD Systems and Desktop Virtual Environments, Graphics Interface 2005, 141-149.

[PA97] Plamondon, R., and Alimi, A. M. Speed/accuracy tradeoffs in target-directed movements. Behavioral and Brain Sciences, 20, 1997, pp. 279-349.

[PB88] Cary B. Phillips, Norman I. Badler, Jack: A Toolkit for Manipulating Articulated Figures, 1988

[PB95] Randy Pausch and Tommy Burnette,Navigation and Locomotion in Virtual Worlds via Flight into Hand-Held Miniatures,1995.

[PBG92] Cary B. Phillips, Norman I. Badler, John Granieri, Automatic Viewing Control for 3D Direct Manipulation,1992.

[PF*97] Image Plane Interaction Techniques In 3D Immersive Environments, Jeffrey S. Pierce, Andrew Forsberg, Matthew J. Conway, Seung Hong, Robert Zeleznik, Mark R. Mine,1997.

[Phy] The cambridge handbook of physics formulas, Handbook of Physics,Rraham Woan, Department of Physics and Astronomy, University of Glasgow

[Pla95] Plaisant, C., Carr, D., Shneiderman, B. Image-Browser taxonomy and guidelines for designers. IEEE Software, Vol. 12, N. 2, March 1995, pp. 21-32

[Poi] Pointer ballistics for Windows XP/Vista. http://www.microsoft.com/ whdc/ device/ input/ pointer-bal.mspx.

[Pou74] Poulton, E. C. (1974). Tracking skill and manual control. New York: Academic Press. 1974

[PS96] Poston, T., Serra, L., “Dextrous Virtual Work,” Communications of the ACM, 39(5), pp. 37-45, 1996.

[PWBI97] I. Poupyrev, S. Weghorst, M. Billinghurst T. Ichikawa (1997) A framework and testbed for studying manipulation techniques for immersive VR. In: Proceedings of the ACM Symposium on Virtual Reality Software and Technology, pp. 21-28.

[PW*98] I. Poupyrev, S. Weghorst, M. Billunghurst, and T. Ichikawa. Egocentric object manipulation in virtual environments; empirical evalutaion of interaction techniques. Computer Graphics Forum, 17(3): pp. 30-41, 1998.

© 2009 Tous droits réservés.

http://www.univ-lille1.fr/bustl

[PWF00] Poupyrev, I., Weghorst, S., and Fels, S., "Non-isomorphic 3D rotational techniques ", in Proceedings of the SIGCHI conference on Human factors in computing systems The Hague, The Netherlands ACM Press, 2000 pp. 540-547

[PWI96] Poupyrev, I. Billinghurst, M. Weghorst, S. Ichikawa, T. (1996) "The Go-Go Interaction Technique: Non-Linear Mapping for direct Manipulation in VR", Proceedings of the ACM Symposium on User Interface Software and Technology, pp. 78-80.

[RC*07] Gonzalo Ramos, Andy Cockburn, Ravin Balakrishnan, Michel Beaudouin-Lafon, "Pointing Lenses: Facilitating Stylus Input through Visual- and Motor-Space Magnification," submitted to CHI2007

[Rea] http://www.reachin.se/

[RGK94] Hinckley, K., Pausch, R., Goble, J., Kassell, N., "Passive Real-World Interface Props for Neurosurgical Visualization," Proceedings of the ACM CHI '94 Conference on Human Factors in Computing Systems, (1994), pp. 452-458.

[RM93] Robertson, G., MacKinlay, J. The Document Lens. Proceedings of UIST'93, ACM Press, pp.101-108.

[RS99] Rekimoto, J. and Saitoh, M. Augmented surfaces: a spatially continuous work space for hybrid computing environments. In Proceedings of CHI '99, ACM Press, (1999), 378-385.

[RS*04] Ringel, M., Ryall, K., Shen, C., Forlines, C., Vernier, F. Release, relocate, reorient, resize: fluid techniques for document sharing on multi-user interactive tables, CHI'04 extended abstracts, pp. 1441-1444.

[SAM03] Subramanian, S., Aliakseyeu, D., Martens, J. B., Empirical Evaluation of Performance in Hybrid 3D and 2D Interfaces. in Human Computer Interaction - Interact'03, (Zurich, Switzerland, 2003), OS Press, (c) IFIP.

[SCP95] Richard Stoakley, Matthew J. Conway, Randy Pausch, Virtual Reality on a WIM: Interactive Worlds in Miniature, 1995.

[SD04] Simon, A., Doulis, M., NOYO: 6DOF Elastic Rate Control for Virtual Environments. Proc. ACM VRST 2004, ACM. (2004) pp. 103-106

[SDL99] Marilyn C. Salzman, Chris Dede, R, Bowen Loftin, VR's Frames of Reference: A Visualization Technique for Mastering Abstract Multidimensional Information, CHI'99

[SF03] Simon, A. and Froehlich, B., The YoYo: A Handheld Device Combining Elastic a Isotonic Input, Interact,2003.

[SG97a] Z. Szalavari, M. Gervautz, The Personal Interaction Panel, A Two-Handed Interface for Augmented Reality, Eurographics'97, pp. 335-346, September 1997.

© 2009 Tous droits réservés.

[SG97b] Chris Shaw and Mark Green. THRED: A two-handed design system. Multimedia Systems, 5:126-139, 1997.

[SG*99] Streitz, N., GeiBler, J., Holmer, T., Konomi, S., Muller-Tomfelde, C., Reischl, W., Rexroth, P., Seitz, P. and Steinmetz, R. i-Land: an interactive landscape for creativity and innovation. In Proceedings of CHI'99, ACM Press, (1999), 120-127.

[SH*05] Anthony J. Sherbondy, Djamila Holmlund, Geoffrey D. Rubin, Pamela K. Schraedley,Terry Winograd, Sandy Napel, Alternative Input Devices for Efficient Navigation of Large CT Angiography Data Sets, 2005.

[Sho92] Ken Shoemake. ARCBALL: a user interface for specifying three-dimensional orientation using a mouse. In Proceedings Graphics Interface'92, pages 151-156, May 1992.

[SJP96] Z. Stanisic, E. Jackson, and S. Payandeh, "Virtual Fixtures as an Aid for Teleoperation, "Proceedings 9th Canadian Aeronautic and Space Institute Conference, 1996.

[SM97] A. Stork, M. Maidhof: Efficient and Precise Solid Modelling using a 3D Input Device. Proceedings of ACM Siggraph Symposium on Solid Modeling and Application, Atlanta, Georgia, May, 1997.

[SP*93] Bier, E. A., Stone, M. C., Pier, K., Buxton, W. and DeRose, T. D. (1993). Toolglass and magic lenses: The see-through interface. Proceedings of SIGGRAPH'93, Annual Conference Series (Anaheim, CA), Computer Graphics 27, 73-80, New York: ACM.

[SRS91] Sachs, E., Roberts, A., Stoops, D., "3-Draw: A Tool for Designing 3D Shapes," IEEE Computer Graphics and Applications, Nov. 1991, pp. 18-26.

[SS77a] Schneider, W. Shiffrin, R.M. Controlled and Automatic Human Information Processing: I Detection, search and attention, Psychological Review, 1977, 84, pp.1-66.

[SS77b] Shiffrin, R.M. and Schneider, W. Controlled and Automatic Human Information Processing: II Perceptual learning, automatic attending and a general theory, Psychological Review, 1977, 84, pp.127-190.

[SS95] A. Steed and M. Slater, 3D Interaction with the Desktop Bat, Computer Graphics Forum, 14(2), 1995.

[SSS02] Stanislav L. Stoev, Dieter Schmalstieg, and Wolfgang StraBer, Through-The-Lens Techniques for Motion, Navigation, and Remote Object Manipulation in Immersive Virtual Environments,2002.

[SSU93] M. Slater, A. Steed, and M. Usoh. The virtual treadmill: A naturalistic metaphor for navigation in immersive virtual environments. In M. Gobel, editor, First Eurographics Workshop on Virtual Environments, Polytechnical University of Catalonia, September 7, pp. 71-83, 1993.

© 2009 Tous droits réservés.

Thèse de Qing Pan, Lille 1, 2008

[Syn] Synaptics, SDK. http://www.synaptics.com/support/dev_support.cfm

[Tac] Tactile 3D. http://www.tactile3d.com

[TP*01] Tandler, P., Prante, T., Muller-Tomfelde, C., Streitz, N. and Steinmetz, R. ConnecTables: dynamic coupling of displays for the flexible creation of shared workspaces. Proceedings of UIST'01, ACM Press, (2001), pp. 11-20.

[TR*91] Turner, Russell, F.Balaguer, E.Gobbetti, D. Thalmann. (1991). Physically-Based Interactive Camera Motion Control Using 3D Input Devices. Computer Graphics International.pp. 135-145.

[TRC01] Desney S. Tan, George G. Robertson, Mary Czerwinski, Exploring 3D Navigation: Combining Speed-coupled Flying with Orbiting. CHI'2001

[TS07] R. Teather and W. Stuerzlinger. Guidelines for 3D positioning techniques. Proceedings of Futureplay'07, 61-68, 2007.

[WA04] Colin Ware , Roland Arsenault, Frames of reference in virtual object rotation, Proceedings of the 1st Symposium on Applied perception in graphics and visualization, August 07-08, 2004, Los Angeles, California

[UI97] Ullmer, B., Ishii, H., "The metaDESK: Models and Prototypes for Tangible User Interfaces," Proceedings of the ACM UIST'97 Symposium on User Interface Software and Technology, 1997, pp. 223-232.

[VK*02] S. Veigl, A. Kaltenbach, F. Ledermann, G. Reitmayr, D. Schmalstieg, "Two-Handed Direct Interaction with ARToolKit", In First IEEE International Workshop on ARToolKit (ART02), IEEE, September, 2002.

[WAB93] Ware, C., Arthur, K., Booth, K. S., "Fish Tank Virtual Reality," INTERCHI'93, pp. 37-41.

[War90] Ware, C. (1990). "Using hand position for virtual object placement." The Visual Computer, 6, pp.245-253.

[WBR02] WINGRAVE, C. BOWMAN, D. RAMAKRISHNAN, N. 2002. TOWARDS PREFERENCES IN VIRTUAL ENVIRONMENT INTERFACES. EIGHTH EUROGRAPHICS WORKSHOP ON VIRTUAL ENVIRONMENTS, 63-72.

[WF97] Ware, C., Fleet, D. Context Sensitive Flying Interface. Proceedings of the 1997 Symposium on Interactive 3D graphics, pp.127.

[WHB06] Wingrave, C., Haciahmetoglu, Y. and Bowman, D. (2006) Overcoming World in Miniature Limitations by a Scaled and Scrolling WIM. submitted to the 1st IEEE Symposium on 3D User Interfaces at IEEE Virtual Reality, 2006.

[Wic92] Wickens, C. D. (1992). Engineering Psychology and Human Performance. HarperCollins Publishers.

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl

[WM01] Wenbi Wang and Paul Milgram, Dynamic Viewpoint Tethering for Navigation in Large-scale Virtual Environments,Proceedings Of The Human Factors And Ergonomics Society 45th Annual Meeting-2001

[WM*98] WANG, Y., MACKENZIE, C., SUMMERS, V.A., AND BOOTH, K.S. 1998. The structure of object Transportation and Orientation in Human-Computer Interaction. Proceedings of ACM CHI'98 , pp. 312-319.

[WMK03] Wobbrock, J. O., Myers, B. A., and Kembel, J. A. EdgeWrite: a stylus-based text entry method designed for high accuracy and stability of motion. In Proc. of UIST '03, 2003, pp. 61-70.

[WO90] Ware, C. and S. Osborn (1990). Exploration and Virtual Camera Control in Virtual Three Dimensional Environments. Proc. 1990 Symposium on Interactive 3D Graphics, Snowbird, Utah, ACM Press.

[WR99] WARE, C. AND ROSE, J. 1999. Rotating Virtual Objects with Real Handles, ACM Transactions on CHI. 6(2) 162-180.

[WT*05] Wingrave, Ryan Tintner, Bruce N. Walker, Doug A. Bowman Larry F.Hodges, Exploring Individual Differences in Raybased Selection: Strategies and Traits Chadwick A. Proceedings of the IEEE Virtual Reality 2005 (VR'05)

[YNS94] T.Yoshimura, Y. Nakamura and M.Sugiura. "3D direct manipulation interface: Development of the zashiki-warashi system", Computers and Graphics 18(2) pp.201-207, 1994

[ZBM94] Zhai, S., Buxton, W., Milgram, P. The "Silk cursor": investigating transparency for 3D target acquisition. In Proc. of CHI'94, 1994, pp.459-464

[ZF99] Robert Zeleznik, Andrew Forsbergt, UniCam - 2D Gestural Camera Controls for 3D Environments Symposium on Interactive 3D Graphics Atlanta GAUSA, 1999

[ZFS97] R. C. Zeleznik, A. S. Forsberg, and P.S. Strauss, Two pointer input for 3D interaction. In Symposium on interactive 3D Graphics (SI3D'97). ACM Press, 1997.

[Zha95] Zhai Shumin, Human Performance in Six Degree of Freedom Input Control, Thesis 95 University Toronto

[Zha04] Zhai, S. Characterizing computer input with Fitts' law parameters - the information and non-information aspects of pointing. Intl. Jrnl. of Human-Computer Studies, 61,6 (2004), pp.791-809.

[ZHH96] R. C. Zeleznik, K. Herndon, and J. Hughes. SKETCH: An interface for sketching 3D scenes. in ACM SIGGRAPH 1996 Conference on Computer Graphics and Interactive Techniques. 1996. New York, NY pp. 163-170.

[ZK*99] S. Zhai, E. Kandogan, B. Search, and T. Selker. In search of the "magic carpet", design and experimentation of a 3d navigation interface. Journal of Visual Languages and Computing, 10(1), 1999.

© 2009 Tous droits réservés.

Thèse de Qing Pan, Lille 1, 2008

[ZM94] Zhai, S., Milgram, P., "Input Techniques for HCI in 3D Environments," Proceedings of the ACM CHI '94 Conference on Human Factors in Computing Systems, (1994).

[ZM98] S. Zhai and P. Milgram. Quantifying coordination in multiple dof movement and its application to evaluating 6 dof input devices. In Proceedings of the SIGCHI conference on Human factors in computing systems, pages 320-327. ACM, 1998.

[ZMD93] Shumin Zhai and Paul Milgram and David Drascic, An evaluation of four 6 degree-of-freedom input techniques,CHI '93: INTERACT '93 and CHI '93 conference companion on Human factors in computing systems, pp. 123-125, Amsterdam, The Netherlands, ACM Press, New York, NY, USA,1993.

[ZSS97a] Shumin Zhai, Barton A. Smith, Ted Selker, Improving Browsing Performance: A study of four input devices for scrolling and pointing tasks, Proceeding of INTERACT97: The Sixth IFIP Conference on Human-Computer Interaction, Sydney, Australia, July 14-18, pp.286-292, 1997

[ZSS97b] Zhai, S., Smith, B.A., and Selker, T. "Improving Browsing Performance: A Study of Four Input Devices for Scrolling and Pointing," in Proceedings of Interact97: The Sixth IF1P Conference on Human-Computer Interaction. 1997. Sydney, Australia: pp. 286-292.

[ZSS97c] Zhai, S., Smith, B. A., and Selker, T. Dual stream input for pointing and scrolling. In Proc. of CHI '97 Ext. Abstracts, 1997, pp. 305-306.

© 2009 Tous droits réservés. http://www.univ-lille1.fr/bustl