6DOF MOUSE INTEGRATION FOR
SCIENTIFIC AND MEDICAL
VISUALISATION

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER
FOR THE DEGREE OF MASTER OF PHILOSOPHY
IN THE FACULTY OF ENGINEERING AND PHYSICAL SCIENCES

2016

By
Mario Sandoval Olivé
School of Computer Science
# Contents

Abstract 9

Declaration 10

Copyright 11

Acknowledgements 12

1 Introduction 14
   1.1 Project Context 14
   1.2 Project Aims and Objectives 16
   1.3 Project Scope 16
   1.4 Overview 17
   1.5 Structure of the Thesis 19

2 Background and Related Work 20
   2.1 VR in Scientific Visualisation 20
      2.1.1 Virtual Reality 21
      2.1.2 Scientific Visualisation 21
   2.2 Input Devices 26
      2.2.1 Properties of the Input Devices 26
      2.2.2 Isotonic and Isometric Input Devices 28
      2.2.3 Elastic and Viscous Input Devices 29
   2.3 Evaluation and Analysis of Input Devices 30
      2.3.1 Fitts’ Law and Hick’s Law 30
      2.3.2 The Steering Law and Minimum-Jerk Law 31
      2.3.3 GOMS Analysis and the Keystoke Level Model (KLM) 31
   2.4 Related Work 32
      2.4.1 Advantages and Disadvantages of Input Devices 32
      2.4.2 Applications 32
      2.4.3 Evaluation: Personal Experience 35
3 System Architecture of the Wing

3.1 The Wing: Beyond the 2D Mouse

3.1.1 A Bit of Background

3.1.2 Applications for the Wing

3.2 Mechanical Design and Operation

3.3 Wing 3D functions

3.4 Calibration and extrema points

3.4.1 Data Collection and Data Analysis

3.4.2 Delta Change Values Range from (-1, +1)

4 Programming Language Implementation

4.1 Criteria for the Selection of the Programming Language

4.1.1 Daemon Details and Selection Criteria

4.1.2 Characteristics of the Selected Languages

4.1.3 Ease of Learning

4.1.4 Ease of Understanding

4.1.5 Speed of Development

4.1.6 Fit-for-purpose

4.1.7 Final Decision

4.2 Criteria for the Selection of the Volume Visualisation

4.2.1 Volume Visualisation and Rendering in Medical Imaging

4.2.2 Characteristics of the Visualisation Volume Packages

4.2.3 Choosing a Visualisation Application Package

4.3 Development and Testing of the Python Daemon

4.3.1 Socket Function

4.3.2 Setting the Extrema Points and the Centre Point

4.3.3 Pygame Function

5 The User Evaluation Study

5.1 The User Group

5.2 Experimental Settings

5.2.1 Experimental Materials and Tasks

5.2.2 Participants Selection and Experimental Treatment

5.3 Experiment Details and Results

5.3.1 The Questionnaire and the Written Comments

5.3.2 Usability and Likeability of the Wing

5.3.3 Utility of the Wing

5.3.4 Cost and Acceptability of the Wing

5.4 Discussion of the Evaluation Results

6 Conclusions and Future Work

6.1 Conclusions

6.2 Limitations
# List of Tables

2.1 Comparative table between three input devices: Mouse vs. Isometric Joystick vs. Touch-screen .......................................................... 33

3.1 Table with the values for $\sigma$ and for the rise to determine the centre point and the range that the Wing can operate efficiently. .............................................. 56

4.1 Number of seconds taken to complete where the code grows text string by adding another string in cycle until it grows up to 1280 Kb. ............................................. 64

4.2 Language selection criteria. Python was selected for getting 3:4 compared to the other languages in terms of ease of learning, ease of understanding and fit-for-purpose. .... 67

4.3 Python libraries used in this project. ........................................................ 74

5.1 Table of the satisfaction questions for the experimental tests. .................... 85

5.2 Correlations of the satisfaction questions with p-value = 0.468 and $\alpha = 0.05$. The blue colours are for significant positive correlations and the red for significant negative correlations. .................................................. 86

5.3 The paradigm of usability and related concepts. ......................................... 86

C.1 Answer to the question of three things the Wing can do better compared to the common mouse. ................................................................. 122

C.2 General comments of the participants about the Wing. ......................... 123

C.3 Comments of the improvements to do on the Wing. ................................ 124

C.4 Answers to open questions about the Wing. ......................................... 125
List of Figures

1.1 A sample of input devices for 6DoF manipulation. .......................... 15
1.2 6DoF motion sensing: translational movement in three perpendicular axes (forward/back, up/down and left/right) and rotational movement in three perpendicular axes (pitch, roll and yaw). ................................................................. 17
1.3 Image of the evolution of the joystick into a sophisticated game controllers able to manipulate objects using some functions of 6DoF. a) Xbox 360 Wireless Controller for Windows; b) Sony Playstation 4 controller; c) Nintendo Wii and Nunchuck controller . 18
2.1 Visualisation pipeline. ................................................................. 23
2.2 Medical imaging examples: A) The results of a CT scan of the human head; B) Magnetic resonance imaging (MRI) machine to generate a magnetic field around a patient; C) Positron emission tomography (PET) scans which create images of active blood flow and physiologic activity of the organ or organs targeted by the use of radiopharmaceuticals; D) Example of the use of ultrasound technology to monitor pregnancies. . . . . 24
2.3 An example presentation package that includes a user interface. .............. 25
2.4 Volumetric visualisation processing pipeline. .................................... 26
2.5 Example of volumetric visualisation processing for medical imaging. ....... 27
2.6 Example of an isotonic and an isometric input device. A) The mouse is a hand-held device pushed over a horizontal surface where the user can freely change the position of the cursor, with more or less constant force; B) With the joystick, the user controls the stick by varying the amount of force they push with and the position of the stick remains more or less constant. ............................................................... 29
2.7 The movement through a tunnel constraint is modelled by the steering law. . . 31
2.8 Different mouse designs. From left to right: Travel Mouse, Desktop Mouse, Gaming Mouse. ................................................................. 34
2.9 Personal experience with input devices in 3D VE. .................................. 36
3.1 Wing shell and base parts part-machined. ........................................ 38
3.2 Wing 3D computer input device. .................................................. 39
3.3 Wing 3D control features. ......................................................... 39
3.4 The Wing with Hexcam UAVs. .................................................... 40
3.5 The principal components of the Wing are all CNC machined from aluminium. . 41
3.6 Wing principal dimensions. The measurements are in millimetres (mm). . . . 42
3.7 Wing principal components. ......................................................... 43
3.8 Roll-pitch-yaw representation. A series of three ordered right-handed rotations is needed to coalesce the reference frame with the rigid body. .......................... 45
3.9 Pygame noise detected in the Wing on the pitch, roll, yaw and Z-axis where: blue-roll, purple-yaw, red-pitch and green-Z axis. .................. 48
3.10 Noise detected in the pitch side of the Wing. ............................... 49
3.11 Noise detected in the yaw bar side of the Wing. ........................... 49
3.12 Example of the use of functions to control the joystick using C++. .. 50
3.13 Input mapping sensitivity. .......................................................... 51
3.14 Normal distribution in roll axis to calculate the centre point. ............ 52
3.15 Normal distribution in yaw axis to calculate the centre point. ............ 52
3.16 Normal distribution in roll axis to calculate the first extrema point (+1). .. 53
3.17 Normal distribution in pitch axis to calculate the first extrema point (+1). 53
3.18 Normal distribution in yaw axis to calculate the first extrema point (+1). 54
3.19 Normal distribution in roll axis to calculate the second extrema point (-1). 54
3.20 Normal distribution in pitch axis to calculate the second extrema point (-1). 55
3.21 Normal distribution in yaw axis to calculate the second extrema point (-1). 55
3.22 Analysis of the input values to set the centre and extrema points .......... 56
3.23 Digital signal processing of input values on axis X of the Wing. ........ 57
3.24 Digital signal processing of input values on axis Y of the Wing. ........ 58
3.25 Digital signal processing of input values on axis Z of the Wing. ........ 58

4.1 Execution speed graph ............................................................... 64
4.2 Drishti visualisation example. ..................................................... 71
4.3 Drishti demos example. Simple example to illustrate transfer functions: human tooth; Stag Beetle. Example of clipping planes; Cochlear. Demonstrate lighting; Reticulate. Example of animation; Human Torso. ................................. 72
4.4 ParaView visualisation example. .................................................. 73
4.5 TomViz graphical user interface for 3D visualisation of tomographic data. 73

5.1 Materials prepared for the experiment. ......................................... 81
5.2 Comparative table of the time per participant to complete the experimental task with the computer mouse and the Wing ..................... 83
5.3 Average annotation time per user to complete the task. ..................... 84
5.4 Correlation between the level of effectiveness in completing the tasks is proportional to the interest the users shown in use the Wing for their work and home use. 87
5.5 Correlation between the level of effectiveness in completing the tasks and users’ interest in buying the Wing. ........................................ 87
5.6 Pie chart where the participants answered the question regarding the effectiveness of the Wing. ....................................................... 88
5.7 Pie chart showing that over half of the participants felt satisfied with the ease of completing the required tasks. .............................. 89
5.8 Correlation between the level of effectiveness and confidence by using the Wing. .... 90
5.9 Correlation to show that the rating of the Wing depends on the user satisfaction feeling with the ease of completing the required tasks. ......................................................... 90
5.10 Correlation between questions 4 and 7 where it is stated that the well-integrated function of the Wing affects positively on the engagement of user with this product. ............. 91
5.11 Pie chart showing that over half of the participants felt confident with the ease of completing the required tasks. ................................................................. 92
5.12 Correlation between question 6 and 8 where it is stated that the preference in using the Wing also depends on how intuitive the Wing can be. ............................. 92
5.13 Correlation between question 6 and 11 where it is stated that the more intuitive the Wing is, the probability to buy it is higher. ................................................. 93
5.14 Correlation between the level of confidence and the easiness to learn how to operate the Wing. ................................................................. 93
5.15 Pie chart of the learning curve of the Wing. ................................................. 94
5.16 Pie chart of the level of intuitiveness of the Wing. ....................................... 94
5.17 The correlation between questions 7 and 12 shows that the rate of this product also depends on the well integrated functions found in this device. ................. 95
5.18 Histogram of the utility of the Wing. ......................................................... 96
5.19 Pie chart of the utility of the Wing. ......................................................... 97
5.20 Pie chart of the interest in the participants to buy the Wing .......................... 98
5.21 Histogram of participants’ acceptability and estimated cost of the Wing. ......... 98
5.22 Pie chart of the overall rating of the Wing. ................................................. 99
5.23 Histogram of the general comments of the Wing were classified in 4 groups for its study. 101
5.24 Histogram of the “improvements” needed to do to the Wing, based on the participant comments. ................................................................. 101

6.1 Digital signal processing of input values on axis X of the Wing with $N_{steps} = 123$ steps. 105
6.2 Digital signal processing of input values on axis X of the Wing with $N_{steps} = 7$ steps. 106

C.1 Pie chart of the level of experience of the participants of the user evaluation study. . 126
C.2 Pie chart of the level of knowledge of the participants of the user evaluation study. . 126
Abstract

6DoF Mouse Integration for Scientific and Medical Visualisation
Mario Sandoval Olivé
A thesis submitted to the University of Manchester
for the degree of Master of Philosophy, 2016

With advances in medical imaging over the years, medical diagnosis, especially X-ray based examinations and ultrasonography, have become increasingly reliant on a range of 3D digital imaging data systems for navigation, reference, diagnosis and documentation. Medical imaging, used in the field of Scientific Visualisation (SV), is used to explore results and extract meaning from complex multidimensional visual representations of the interior of a human body for clinical analysis and medical intervention. The implication of this is that an effective and intuitive input device is needed that allows the information entered into the computer can be encompassed to different imaging modalities and processes for diagnostic and treatment purposes. Traditional input devices such as the computer mouse are limited to explore these data sets, creating difficulties and inefficiencies, which in turn can entail potential medical complications as, in public health and preventive medicine as well as in both curative and palliative care, effective decisions depend on correct diagnoses. Therefore, using this principle this work investigates human-computer interactions in relation to a specific Six Degree of Freedom (6DoF) input device called the Wing and reports the results of a brief satisfaction questionnaire to analyse the effectiveness and intuitiveness of this device in 3D visualisation volume software for SV and medical imaging, the mental processes involved in performing control actions on this device and the effect of proprioception of the test users. Finally, it concludes with a discussion of the evolution of the Wing based on the results and new possibilities for 3D interactions in other fields.
Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.
Copyright

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the “Copyright”) and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made only in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii. The ownership of certain Copyright, patents, designs, trade marks and other intellectual property (the “Intellectual Property”) and any reproductions of copyright works in the thesis, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see http://documents.manchester.ac.uk/DocInfo.aspx?DocID=487), in any relevant Thesis restriction declarations deposited in the University Library, The University Library’s regulations (see http://www.manchester.ac.uk/library/aboutus/regulations) and in The University’s policy on presentation of Theses.
Acknowledgements

First and foremost, I would like to thank God, the Almighty, for having made everything possible by giving me strength and courage to do this work. I give thanks to you Lord, because you have made known to me the paths of life and you have filled me with happiness and joy. In the process of putting this thesis together I realised how true this gift of learning and understanding visualisation topics and developing code is for me. You have given me the power to believe in my passion and pursue my dreams. I could never have done this without the faith I have in you Jesus, the Almighty, True and Living God, to whom I will be always grateful.

In addition, I would like to express my sincere gratitude to my supervisor Dr. Tim Morris for the continuous support of my MPhil. study and related research; for his patience, motivation and immense knowledge. His guidance helped me throughout both the research and writing of this thesis. I could not have imagined having a better supervisor and mentor for my MPhil. study.

Besides my supervisor, I would also like to thank Dr. Martin Turner for his insightful comments and encouragement, but also for the hard questions which motivated me to widen my research from various perspectives.

My sincere thanks also go to Sam Worthington and Worthington Sharpe Ltd., who provided me an opportunity to use their most recent novelty invention, the Wing, and who gave access to all the information and documentation regarding this input device. Without their precious support it would not be possible to conduct this research.

Furthermore, I am highly indebted to the School of Computer Science of the University of Manchester and its academic staff for providing the laboratory and research facilities for evaluating the Wing. Thanks for your guidance and constant supervision as well as for providing necessary information regarding the project and support in completing this project.

I thank my fellow labmates for the stimulating discussions, for the sleepless nights as we were working together before deadlines and for all the fun we have had in the last four years. Also I thank my friends in the Manchester Science Park and Methodist International House. In particular, I am grateful to Dr. Kepa Koldo Burusco, Meike Weis, Jeong Hyeon Lee, Andreas Tomsa, David Perretta Tejedor and Aranza Gomez for supporting me in the initial stages of my research and for challenging my way of thinking by helping me question assumptions and view issues from multiple perspectives, as well as seeing variety as a chance, not as the barrier.

To my friends and housemates, thank you for listening, offering me advice and supporting me through this entire process. Special thanks to Habersheel Acevedo, Hugo Aguilera, Darren Fitzpatrick, James Edward Perkins, Luca Vitaloni, Garry Collier, Raul Sanse and Adrian Byrne. The debates, dinners and game nights as well as editing advice, rides to the airport and general help and friendship were
all greatly appreciated. To my friends scattered around the country, thank you for your thoughts, well-wishes/prayers, phone calls, e-mails, texts, visits, editing advice and being there whenever I needed a friend.

Last but not the least, I would like to thank my family; this journey would not have been possible without their support. I thank my parents Griselda Olive and Carlos Sandoval for supporting me spiritually throughout my life. To my sisters Gris and Hilda, many thanks for being there supporting me, your prayers helped me to succeed. I always knew that you believed in me and wanted the best for me. Thank you for teaching me that my job in life was to learn, to be happy and to know and understand myself; only then could I know and understand others.
Chapter 1

Introduction

“We should look on the display as a window into a virtual world. Improvements of image generation will make the picture look real. Computers will maintain the world model in real time. Immersion in virtual worlds will be provided by new displays. Users could directly manipulate virtual objects. The objects will move realistically. Virtual world will also sound and feel real.” Ivan Edward Sutherland [Sut70]

1.1 Project Context

Advances in computer graphics and virtual reality (VR) technologies have kick started an increase in the creation of devices that allow users to interact with 3D objects [BKLJP04]. These technologies involve the use of input devices that have an arbitrary number of degrees of freedom (commonly abbreviated as DoF), and include gloves, position trackers and hand controllers (as can be seen in figure 1.1).

Pressure from movement and twists of the user’s fingers can generate related movements in the $x$, $y$ and $z$ axes as well as rotations along these three axes. These then represent a standard 6DoF input device and versions are now commonly available for desktop 3D applications to manipulate virtual objects [BKLJP04] [Zha95].

One of the best known and most used VR applications are vehicle simulators, mostly for aeroplanes, cars or ships but VR applications are also used by engineers for virtual prototyping to design, develop and evaluate new products and in the entertainment industry within specific games, virtual rides and interactive story telling such as museum exhibitions. Other serious application areas [ZPL15] include:

- Architectural, planning and environmental design.
- Aerospace simulation and modelling.
- Naval and military training as well as military operations.
- Medical applications e.g. psychiatric treatment, surgical training and biopsy.
- Visualisation of scientific data from medicine, chemistry, pharmacy, geology, meteorology and other applied sciences.
A single nDoF input device often does not replace the mouse; rather, they are used in conjunction with it. For example while one hand is on the motion controller positioning the object in 3D space the other hand with the mouse can simultaneously select menu items and edit the object.

Today’s scientific simulations and data sensing/measuring systems produce enormous amounts of data and a practical way to gain insight is to use data visualisation [SCG+15] [KT15]. Users not only have to choose the software with the features necessary for a visualisation but also need to select the device which helps the user explore the 3D object in an appropriate manner. A sample list of 6DoF input devices are shown in figure 1.1.

Figure 1.1: A sample of input devices for 6DoF manipulation.

Figure 1.1 show a sample list of input devices for 6DoF manipulation:

(a) The VKB Black Mamba™ joystick is made for simulators and designed to last long under heavy usage without degrading in precision (https://flightsimcontrols.com/).
(b) The Haptic Workstation™ is a fully integrated simulation system providing right and left whole-hand haptic feedback, immersive 3D viewing and easy-to-use CAD model manipulation and interaction software (http://www.gizmag.com/go/1606/).

(c) Kensington Expert Mouse Trackball is an input device which requires far less wrist movement than a mouse to provide a pain-free computing experience (http://www.kensington.com/).

(d) 3Dconnexion SpaceExplorer™ is a 3D Navigation device which allows the user to command six-degrees of freedom to manipulate 3D objects on screen (http://www.3dconnexion.co.uk/service/drivers.html).

(e) The CyberGlove™ motion capture system has been used in a wide variety of real-world applications, including digital prototype evaluation, virtual reality biomechanics and animation (http://www.cyberglovesystems.com/).

1.2 Project Aims and Objectives

The goal of this project is the evaluation of a specific six degree of freedom input device called the Wing in a 3D visualisation volume package called Drishti, used for scientific visualisation (SV) and medical imaging. For this, we defined a set of experimental tests and evaluations to consider the product’s effectiveness, efficiency and satisfaction in a specified context of use. These results were then compared with the use of a standard mouse, a 2DOF device, in the same scene. The research methodology used within this project is a combination of experimentation, literature comparison and technical development of the hardware and software device drivers. Conclusions are drawn based upon statistical analysis of the experimental results that are discussed in relation to the literature.

1.3 Project Scope

The scope of the project focuses on testing the Wing and involves the following tasks:

- Design and building the low level driver (tracking software daemon) integrated with the Wing.
- Analysis of the statistical features, specifically the asymmetrical bias in the signal and noise including the latency incorporated in the drivers.
- Software integration, including programming a calibration method, for use within the visualisation system’s pipeline.
- Linking the interface values to arbitrary 1D/2D (controller parameters) and 3D (observational modes) interface control for scientific visualisation modes:
  - Development of new methods for rotation and zoom within indirect volume visualisation so adding new interrogation exploration modes.
  - Development of new methods for direct volume visualisation through gradient magnitude transfer function manipulations.
Human-computer interaction (HCI) user evaluation for these different modes.

The software daemon designed and used in this project is developed using Python and Pygame and described in chapter 4.

1.4 Overview

Today, there are many input devices that are used in desktop 3D user interfaces. Many of these devices have been used and designed for traditional 2D desktop applications such as modelling and computer games. Some desktop input devices have also been developed for 3D interaction and the most common input devices are:

- **Keyboards**, a classic example of a traditional desktop input device that contains a set of discrete components (i.e. a set of buttons).

- **Mice and track balls** are another classic example of desktop input devices, widely used in traditional 2D and 3D input tasks and come in many different varieties.

- **Joysticks**, input devices come with a combination of a manually continuous 2D locator with a set of discrete components such as buttons and switches.
CHAPTER 1. INTRODUCTION

Figure 1.3: Image of the evolution of the joystick into a sophisticated game controllers able to manipulate objects using some functions of 6DoF. a) Xbox 360 Wireless Controller for Windows; b) Sony Playstation 4 controller; c) Nintendo Wii and Nunchuck controller

- **nDoF devices for the desktop**, input devices that allow the user to interact with 3D objects.

Many different characteristics can be used to describe input devices. One of the most important is the degree of freedom that an input device affords. A degree of Freedom refers to the movement of a rigid body inside any space. It could be explained as “the minimum number of independent variables required to completely describe a body”.

A device such as a tracker generally captures three position values and three orientation values for a total of 6DoF (see figure 1.2). Any possible movement of a rigid body through input, no matter how complex it is, can then be expressed as a combination of this basic 6DoF. For instance when someone hits a tennis ball with a racket, the complex position of the racket can be expressed as a combination of translations (where the body is free to translate in 3DoF: forward/back, up/down, left/right) and rotations (where the body can also rotate with 3DoF: pitch, roll and yaw) [AT87].

Many console video game systems make use of different joystick designs in their game controllers, including vibration motors, analogue sticks, pressure-sensitive directional pad, analogue triggers and multiple buttons (see figure 1.3), mainly because the manipulation of 3D objects requires the efficient handling of 6DoF.
CHAPTER 1. INTRODUCTION

1.5 Structure of the Thesis

The rest of this thesis is organized in the following manner:

- Chapter 2 deals with the background knowledge and related research that is required to be gathered before doing this project.

- Chapter 3 covers the system architecture of the Wing; including the principal components, functions and calibration.

- Chapter 4 covers the implementation, apparatus and methods employed. This includes the choice of programming language and volume visualisation software to be used.

- Chapter 5 elaborates on the evaluation process used describing the methodologies chosen.

- Chapter 6 covers the conclusions, achievements and future works.
Chapter 2

Background and Related Work

“Many people are familiar with the use of mouse devices or joysticks in 3D/2D environments mainly because of the relentless popularity of video games stem.” Chris Melissinos [Mel12]

At first glance, it may seem that there is no difference using either a mouse or a joystick device within a 3D interface. This initial perception, however, fails to take into account an important aspect of nDoF navigation that throughout this chapter will be described.

This chapter presents the background knowledge that is used in this project and the literature review describes the previous related work. This addresses several related areas of 3D user interface design (including 3D manipulation) and compares different input devices and virtual reality (VR) display technologies. The rest of this chapter is organised as follows:

- Section 2.1 VR in Scientific Visualisation and their relation with input devices
- Section 2.2 A brief taxonomy of Input devices
- Section 2.3 Evaluation and analysis of input device based on descriptive models of human movement
- Section 2.4 Related work with 6DoF input devices.

2.1 VR in Scientific Visualisation

Development of 3D user interfaces (UI) is a subject of study for human-computer interaction (HCI), because the users must communicate actions, commands, intents, request, questions and goals to the system. During the last couple of decades, HCI has seen an influx of new input devices that aim at providing a heightened level of realistic interaction between users and computers. Therefore, HCI experts have developed general principles for good interface design and models that explain how humans process information when interacting with systems.
2.1.1 Virtual Reality

“Virtual reality is the use of computer technology to create the effect of an interactive 3D world in which the objects have a sense of spatial presence. The primary difference between conventional 3D computer graphics and VR is that in Virtual Reality we are working with things instead of pictures of things.” 

Bryson, S. [Bry94]

VR refers to any computer generated graphical simulation, in which a user is practically immersed in a 3D environment (VE). Immersion can be defined as an experience of being included in the simulation and the capability of interacting with the VE. Thus, VR allows viewers to be participants immersed in the data rather than passive observers watching from a distance.

Immersion is made up of two main components: depth of information and breadth of information. When a user is using simulations and interaction information is exchanged between the user and the virtual environment. Depth of information includes anything and everything starting from the resolution of the display unit, the graphics quality, the effectiveness of the audio and video, etc. Breadth of information is defined as a number of sensory dimensions presented simultaneously.

Some of the components used to build a VR system include:

- Visual displays
- Tracking systems
- Input devices
- Haptic devices
- Sound systems
- Graphics and computing hardware

Immersive VR provide a good environment for scientific visualisation because the data are often high-dimensional, can be represented in a three-dimensional (3D) volume and the visualisation of the phenomena associated with this data often involves 3D structures. Also, shapes and relations of 3D structures are often extremely important that VR can highlight. The next section briefly explains Scientific Visualisation and how it can be used in VR.

2.1.2 Scientific Visualisation

“Scientific visualisation is the use of computer graphics to create visual images which aid in the understanding of complex, often massive numerical representations of scientific concepts or results.” 

McCormick, B.H. [McC87]

Scientific visualisation (SV) is the representation of data graphically as a means of gaining understanding and insight into the data. It is sometimes referred to as “visual data analysis” and has an aim to allow researchers gain new insights into the system.

Scientists and engineers run experiments and simulations that can produce huge amount of data, which they wish to interactively navigate through and query various points via SV. This involves research
in many areas including computer graphics, image processing and high performance computing and exploits the same tools that are used for VR.

Visualisation Techniques and Methods

The process of data visualisation can be described as a sequence of fundamental processing steps that [Hab90] [Nie90] termed the visualisation pipeline:

- **Simulation**: results of numerical simulations (or data sensing / measurement) are the input of the visualisation pipeline.

- **Data selection and filtering**: relevant regions of the raw data are selected, then filtered and enhanced. Techniques such as enrichment and enhancement, data cropping, down-sizing, noise filtering, segmentation and feature extraction can be used.

- **Visualisation mapping**: the processed data are mapped / transformed into graphical primitives such as; points, lines, planes / surfaces (triangle meshes), or icons, as well as their properties such as; colour, texture, or opacity.

- **Rendering**: finally, the graphical primitives are rendered as images, which are then displayed on the screen.

On conventional computers (see figure 2.1) the users are provided with the keyboard and mouse interface and can interact with the different stages of the visualisation pipeline. If a visualisation system provides a good interactive control over the processing elements (not just changing the view on the data in the rendering module, but also changing visual representations and providing further options), then it can become a powerful research tool.

For example, scientists can interpret potentially huge quantities of laboratory or simulation data to aid reasoning and cognition, becoming an ideal tool to build and test hypotheses related to that field.

The field of **medical imaging** is a huge application domain for SV that we will consider in more detail [HR03]. Through these images, physicians can now diagnose or monitor treatment for conditions such as tumours, heart problems, malformations of the blood vessels and tracking the gestation of a foetus in the womb of a pregnant woman [HR03].

**Medical Imaging**

Medical imaging involves different imaging modalities and processes to image the human body for diagnostic and treatment purposes. More recent medical imaging technologies produce data that is integrated and analysed by computers to produce three-dimensional images or images that reveal aspects of body function (see figure 2.2). Therefore, medical imaging plays an important role in initiatives to improve public health for all population groups.

Over the years, different sorts of medical imaging have been developed, each with their own advantages and disadvantages. Figure 2.2 shows some of the most common applications of medical imaging, including Computed tomography (CT) for X-ray based examinations, magnetic resonance imaging (MRI), positron emission tomography (PET) and ultrasonography, all of them crucial in a variety of
medical settings and at all major levels of health care. Medical imaging has transformed medicine because innovations can be made faster, more precise and less invasive.

Below is a brief explanation of the four applications mentioned:

- **Computed tomography (CT)** is a non-invasive imaging technique that uses computers to analyse several X-rays radiographs in order to reveal minute details about structures in the body (figure 2.2 A).

- **Magnetic resonance imaging (MRI)** is a non-invasive medical imaging technique which uses a
CHAPTER 2. BACKGROUND AND RELATED WORK

Figure 2.2: Medical imaging examples: A) The results of a CT scan of the human head; B) Magnetic resonance imaging (MRI) machine to generate a magnetic field around a patient; C) Positron emission tomography (PET) scans which create images of active blood flow and physiologic activity of the organ or organs targeted by the use of radiopharmaceuticals; D) Example of the use of ultrasound technology to monitor pregnancies.

A powerful magnetic field, radio frequency pulses and a computer to produce detailed pictures of organs and soft tissues (figure 2.2 B).

- **Positron emission tomography (PET)** is a medical imaging technique involving the use of so-called radiopharmaceuticals, substances that emit radiation that is short-lived and therefore relatively safe to administer to the body (figure 2.2 C).

- **Ultrasonography** is an imaging technique that uses the transmission of high-frequency sound waves into the body to generate an echo signal that is converted by a computer into a real-time image of anatomy and physiology (figure 2.2 D).
CHAPTER 2. BACKGROUND AND RELATED WORK

Figure 2.3: An example presentation package that includes a user interface.

Graphics Libraries and Presentation Packages for SV and Medical Imaging

Data presentation can be beautiful, elegant and descriptive. There is a variety of conventional ways to visualise data; tables, histograms, pie charts and bar graphs that are being used every day (see figure 2.3). Medical imaging also focuses on the factors that affect the VR experience required for multidimensional and volumetric datasets.

Volumetric visualisation comprises four main processing stages (see figure 2.4), which are similar to the visualisation pipeline but these stages create/operate upon voxels. Whereas for 2D images we were managing pixels, a voxel is the short word for volume pixel which is the smallest distinguishable volume element of a three-dimensional space. The four main processing stages of volumetric visualisation are:

- **Data formation.** Stage where the simulated data is produced by a computer simulation or voxelization. Voxelization is the process of adding depth to an image using a set of cross-sectional images known as a volumetric dataset. Here, the original data are enhanced through standard image processing techniques and then the 2D slices are stacked together to generate 3D volume data sets.

- **Data classification.** Stage where the measured data are represented as raw data.

- **Data manipulation.** Stage where the 3D voxel set can be enhanced using 3D processing techniques, including geometric transformations, volume merging, for example.
Figure 2.4: Volumetric visualisation processing pipeline.

- **Data viewing.** The final stage where the 3D volume is projected onto 2D using volume shading and output image composition.

A typical result of volumetric visualisation processing is shown in figure 2.5. In chapter 4, we will discuss in more detail the specific Volume Visualisation Software used for this project.

### 2.2 Input Devices

Input devices sense physical properties of people, places, or things. They allow users to communicate and feed instructions and data to computers for processing, display, storage and/or transmission. The most commonly used input devices on a computer are the keyboard and mouse. However, there are dozens of other devices that can also be used to input data into the computer. Examples include the data glove and its variants, three-dimensional joysticks and voice recognition systems.

This section enumerates some of the properties of input devices, as well as models and theories that help to evaluate interaction techniques.

#### 2.2.1 Properties of the Input Devices

A few important key properties [HW02] that can characterise most input devices are:
CHAPTER 2. BACKGROUND AND RELATED WORK

27

Figure 2.5: Example of volumetric visualisation processing for medical imaging.

- **Sampling.** In sampling (also called *polling*), the application program determines how often measurements from the physical sensors embedded in the input device are taken and sent to the computer system. Increased sampling rates produce finer control over the input.

- **Resolution.** Resolution is a metric of the number of unique measurements the input device can send to the computer. In this context, it refers to the conversion of an analog voltage to a digital value in a computer.

- **Latency.** Latency, or lag, is the time that elapses between the physical actuation of the input device and the resulting on-screen feedback. Latency greatly affects how usable and enjoyable electronic as well as communication devices are. For example, in internet gaming, low latency is critical for best gameplay and enjoyability.

- **Noise.** The IEEE Standard Dictionary of Electrical and Electronic Terms defines noise (as a general term) as *unwanted disturbances superposed upon a useful signal that tend to obscure its information content*. Noise is the result of sensing errors due to hardware malfunctions or design inadequacies and increased noise leads to sampling problems and loss of accuracy. Noise limits the performance of any communication system.

- **Position mode.** The position mode can be either absolute or relative.
  - For an absolute positioning 2D interface, each position on the sensing surface corresponds to a specific location on the screen. For example, a touch screen is often an absolute input
device because each location on the touch screen corresponds to a location on the computer screen.

- In a relative positioning for 2D interfaces, each input is a functional translation from the current point. For example, a mouse is a relative input device because it detects relative motion using a mechanical ball or light reflections. It transmits the direction and the amount of motion, which are then used to move a 2D pointer.

- **Gain.** Gain is also referred to as the Control-Display (C/D) ratio. C/D is the ratio of the distance that the on-screen cursor moves in relation to the physical movement of the input device. If the C/D ratio is high, then the control sensitivity is low and vice-versa. For example, the movement of a mouse should have a reasonable C/D control so users can minimize the effort when they try to move their mouse to reach a target.

- **n Degrees of freedom.** Degrees of freedom is a measure of the number of dimensions that the input device senses. Navigation and data analysis in VR environments often require multiple degrees of freedom. 6 dof are commonly used to implement this. Not all applications incorporate all six degrees of freedom and input devices with fewer degrees of freedom might be more appropriate for specific tasks.

- **Direct and indirect input devices.** Indirect and direct input refers to how data or commands are entered into a system [Jac96]. If the input surface is also the display surface, then the input device is direct. An example of such a device is a touch-screen. Most other input devices are indirect in that the on-screen cursor is controlled through an intermediary device such as a mouse or joystick.

- **Footprint.** Footprint refers to the amount of space that is required for the device. The theory suggests that a small footprint is better than a big one since desk space is a finite resource. The footprint of the input devices depends on the sequence of action in its applications. For example, a mouse has a large and variable footprint, whereas a trackball has a smaller but fixed footprint.

- **Device acquisition time.** Data acquisition systems capture real-world signals and convert them into a format that a computer can understand. Device acquisition time refers to the time it takes to grasp the input device before the sample circuit settles to the new input voltage.

### 2.2.2 Isotonic and Isometric Input Devices

Using his hands, a user can send and receive information through either force/torque or displacement/rotation. Input devices are able to translates this analogue signal into a digital one for interpretation by the computer application and from HCI studies these input devices can be classified into two groups: isotonic and isometric devices. Figure 2.6 displays two examples of these kind of input devices.

- **An isotonic device** is a displacement or free moving device which should have zero or constant resistance (e.g. the mouse) allowing the user to be connected to the machine through movement.

- **An isometric device** senses the angle of deflection. In other words, this device senses force but does not perceptibly move (e.g. a desktop joystick); as a result users are connected to the machine through force/torque.
CHAPTER 2. BACKGROUND AND RELATED WORK

Figure 2.6: Example of an isotonic and an isometric input device. A) The mouse is a hand-held device pushed over a horizontal surface where the user can freely change the position of the cursor, with more or less constant force; B) With the joystick, the user controls the stick by varying the amount of force they push with and the position of the stick remains more or less constant.

There are some discussions highlighting advantages and disadvantages of isometric versus isotonic devices [BKLJP04] [HW02] [Zha95]. However, this is not conclusive and the definitive answer may depend upon dimensions of the controllers other than resistance and also on the tasks they are used for. For example, the response speed (also known as bandwidth) and extent of feedback have been the two major underlying factors that researchers have believed to account for the theoretical differences between isometric versus isotonic devices.

Different studies have concluded that, with regards to the response speed, human response with an isometric device is faster than with an isotonic device, since no transport of limb or device is needed [HW02] [Zha95]. With regards to the feedback, different researchers disagree on which device actually provides stronger feedback.

2.2.3 Elastic and Viscous Input Devices

As with touch tables, mice and trackballs, the force displacement relationship of a joystick specifies the displacement of a joystick that result from applying a specific force. This force is characterised by its extreme force-displacement relationship: on one hand, the displacement of the input device is completely independent of the force applied. At the other extreme, it is possible to design input devices that offers little -or insignificant- resistance to displacement.

Most displacement input devices, however, offer at least some resistance to displacement and in this case there are two common relationships between the force applied to the input device and its resulting displacement: elastic and viscous resistance.

When a device’s resistive force or torque increases with displacement we say that the device is elastic, whereas when the resistance increases with velocity of movement, the device is viscous.

In elastic devices, the application of a constant force to the device results in the constant displacement. Increasing the force applied to the input device increases its displacement. This direct relationship between applied force and resulting displacement can be used by the operator to enhance positioning accuracy. One advantages of this device is the capability to automatically return to its center position upon release.

When the input device has viscous resistance, the application of a constant force to the device results
in a constant rate of movement. Increasing the force applied to the input device increases the rate of dis-
placement. Viscous damping helps in the execution of smooth control movements and it is appropriate if maintaining a constant rate of cursor movement or acquiring a cursor location at a particular time are important aspects of the task.

2.3 Evaluation and Analysis of Input Devices

A number of techniques to the study of input devices exist. In this section, we will focus on describing formal analysis using Fitts’ Law and Hick’s Law, the Steering Law and the Keystroke-Level Model.

2.3.1 Fitts’ Law and Hick’s Law

Fitts’ law \cite{Fit54} is a model of human psycho-motor behaviour developed in 1954 that states that the time required to move to a target is a function of the target size and distance to the target. This law has been widely applied to the comparison and optimization of pointing devices. Pointing movements (such as moving a mouse over a link and clicking it) typically begin with a quick movement toward the target (ballistic movement) followed by fine-tuning movements (homing movements) over the target. Homing movements usually take the most time and are generally responsible for most causes of any error. Fitts’ law applies to remarkably diverse task conditions, including:

a) Rate-controlled devices
b) Area cursors
c) Scrolling
d) Zooming

Fitts’ law gives us some ideas for how we can do that and can be expressed mathematically as:

$$ T = a + b \log_2(1 + D / W) $$  \hspace{1cm} (2.1)

Fitts’ law is typically used for input device studies where the movement time \((T)\) is a function of distance \((D)\) and width \((W)\) and that the relationship is logarithmic. The logarithmic relationship means that after some point Fitts’ law will have diminishing returns. The constants \(a\) and \(b\) are the coefficients that fit to the average of all observed \(T\) for each combination of \(D\) and \(W\) in the experiment.

Another law that has implications in navigation with input devices and asking people to take action is Hick’s law which states that the time it takes to make a decision increases as the number of alternatives increases. Hick’s law can be expressed mathematically as:

$$ T = b \log_2(N + 1) $$  \hspace{1cm} (2.2)

Here, the time it takes to make a decision \((T)\) is dependent on the number of choices \((N)\) and that once again this relationship is logarithmic.
2.3.2 The Steering Law and Minimum-Jerk Law

Trajectory-based interactions, such as navigating through nested-menus, drawing curves and moving in 3D worlds, are becoming common tasks in modern computer interfaces. Users’ performances in these tasks cannot be successfully modelled with Fitts’ law as it has been applied to pointing tasks. A steering through a tunnel task is used to represent such interactions. Therefore, the importance of the Steering Law and the Minimum-Jerk Law.

The Steering Law predicts the movement time through a particular space with constraints, such as a straight or a narrowing tunnel. This law describes the time required to navigate, or steer, through a 2D tunnel. A tunnel is the path or trajectory that has an associated thickness or width. The goal of a steering task is to navigate from one end of the tunnel to the other as quickly as possible, without touching the boundaries of the tunnel (see figure 2.7). [KMS05]

The Minimum-Jerk Law states that unconstrained human movement trajectories tend to minimize the derivative of acceleration (jerk). One of its implications is that there is a two-thirds power law linking tangential velocity and path curvature. [YGHHN12]

2.3.3 GOMS Analysis and the Keystoke Level Model (KLM)

GOMS is a modelling technique that analyses the user complexity of interactive systems. It is used by software designers to model user behaviour. The user’s behaviour is modelled in terms of Goals, Operators, Methods and Selection rules - therefore its name [Hoc02].

This model consists of methods that are used to achieve goals. A method in GOMS is a sequential list of operations that the user performs and goals that must be achieved. If there is more than one method which may be employed to achieve a goal, a selection rule is invoked to determine what method to choose, depending on the context.

On the other hand, the Keystroke Level Model (KLM) is an engineering and analysis tool that can be used to predict the way an interface will behave. It is a predictive model and it is a simplified version of GOMS. KLM estimates the overall task execution time by counting all of the physical operations and then summing the mental acts required. The KLM-Operators [Dau11] are:
a) K for keystroke which is estimated to be between 0.12 and 1.2 sec.
b) P for pointing which is estimated in 1.1 sec.
c) B for pressing or releasing the mouse button, also estimated in 0.1 sec.
d) H for home hands to keyboard or mouse, estimated in 0.4 sec.
e) M for routine thinking or perception, estimated in 1.2 sec.
f) W for waiting for the system to respond where t must be determined.

In this project, as we are evaluating a device with different participants, we use this technique to predict the time it will take for the user to carry out different goals.

2.4 Related Work

Thirty-one years after their first break onto the public stage with the introduction of the Apple Macintosh in 1984, 2D mouse devices are, until now, the ubiquitous pointing devices. In shops today we can find laser mice, bluetooth mice, even specialised game mice, since aiming, targeting, slashing, attacking are some of the most important actions someone takes in a game and all of them can happen with the click of a mouse.

Nevertheless, the latest input devices have given way to much more elaborate controllers with several buttons, triggers and dials mainly because the mouse, despite it being a universal tool for computers, it is limited to certain tasks.

A large number of studies have shown that the mouse is often faster than other input devices, especially the isometric joystick. A possible explanation for this is that all of these studies focus on characteristics of movement such as target width, target distance and movement time, which are the variables in the Fitts’ law equation [MD96]. However, it is not necessarily true that being faster is the same as being efficient.

Information about the research related to this project is presented in the following three sections; the advantages and disadvantages of current input devices, the application chosen and the evaluation process.

2.4.1 Advantages and Disadvantages of Input Devices

Table 2.1 shows some of the advantages and disadvantages of three popular input devices: the computer mouse, the isometric joystick and the touchscreen. This comparative table was created based on studies performed on 3D VE [FWSB07] [DSC07].

2.4.2 Applications

Isotonic Mouse

Today the market is filled with different pointing devices, with the most common being the computer mouse, which is called ”mouse” because of the cord attached to the rear part of the device resembling
### Advantages

<table>
<thead>
<tr>
<th>Mouse</th>
<th>Isometric Joystick</th>
<th>Touchscreen</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Ideal for use with desktop computers.</td>
<td>• Great for 2D or 3D VE systems such as fighting games, shooting games and flight simulators.</td>
<td>• Often used for user-friendly selection from a menu of items for non-specialist users.</td>
</tr>
<tr>
<td>• Works well in conjunction with a keyboard for data entry.</td>
<td>• More comfortable on user’s hands and more appropriate for immersion.</td>
<td>• Fast to respond.</td>
</tr>
<tr>
<td>• Fast to respond.</td>
<td>• More intuitive when playing a video game. Allow users to grip the device with their hands for more control.</td>
<td>• Very intuitive and easy to learn.</td>
</tr>
<tr>
<td>• Can be used right or left handed.</td>
<td>• Compatibility with any OS.</td>
<td></td>
</tr>
<tr>
<td>• Compatibility with any OS.</td>
<td>• Relatively inexpensive</td>
<td></td>
</tr>
<tr>
<td>• Relatively inexpensive</td>
<td>• Improve enjoyment and user experience for games.</td>
<td></td>
</tr>
</tbody>
</table>

### Disadvantages

<table>
<thead>
<tr>
<th>Mouse</th>
<th>Isometric Joystick</th>
<th>Touchscreen</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Can contribute towards carpal tunnel syndrome and repetitive strain injury.</td>
<td>• Their use is limited for some game genre and simulators.</td>
<td>• The screen has to be big enough to be able to touch the buttons without missing.</td>
</tr>
<tr>
<td>• Takes up a lot of space on the desk and they need a flat surface to be used.</td>
<td>• Compatibility with all gaming systems.</td>
<td>• They usually cost more than ordinary devices.</td>
</tr>
<tr>
<td>• They are not intuitive to use as other input devices such as touchscreens or joysticks.</td>
<td>• Standard joysticks commonly operate in one of two ways: position-controlled or rate-controlled. Therefore, they only will allow a four directional movement that of left, right, up (forward) and down (backward), which is a problem for games using 360 degree access.</td>
<td>• If they crash, recovery could be difficult due to the lacking of buttons.</td>
</tr>
<tr>
<td></td>
<td>• A good joystick is not cheap.</td>
<td></td>
</tr>
</tbody>
</table>

Table 2.1: Comparative table between three input devices: Mouse vs. Isometric Joystick vs. Touchscreen
CHAPTER 2. BACKGROUND AND RELATED WORK

Figure 2.8: Different mouse designs. From left to right: Travel Mouse, Desktop Mouse, Gaming Mouse.

a mouse’s tail. It was invented by Doug Engelbart in the early 1960’s at Stanford Research Institute (SRI) and its main use in 2D environment is to control the motion of a pointer in a graphical user interface (GUI). The mouse can be used in 3D VE, where the mouse’s motion often translates directly into changes in the virtual objects’ or camera’s orientation.

There are different kinds of computer mice in the market: desktop mice, the typical one which are made for everyday use and have the standard right and left buttons and a scroll wheel for navigating in web browsers or PC applications; Travel mice, designed for use on the road, are usually much smaller, made to fit easily into the pocket of a laptop bag or backpack and connect quickly without a tangle of wires; Gaming mice, which offer reliable connectivity, smooth and responsive tracking and buttons, controls and scroll functions for PC games. (See figure 2.8)

Isometric Joystick

The joystick was invented by C. B. Mirick at the United States Naval Research Laboratory (NRL) in 1926 [Lab11]. In the beginning, the joystick was originated as a controller for radio-controlled pilotless aircraft. Later on, between 1967 and 1972, the advance and proliferation of different types of computers and video machines has been accompanied by the development of associated joysticks, which were able to control the horizontal and vertical position of a spot displayed on a screen.

Since then, joysticks have been used extensively to play games as they can control the movement of objects quickly and accurately. In some cases joysticks are used to control the movement of specific devices such as video surveillance cameras.

Some researchers estimate that joystick can replace the computer mouse based on its multiple features. Joysticks employ a vertical rod mounted on a base with one or two buttons. Rather than controlling games, these joysticks are able to control the pointer on the computer screen. In addition, these devices are appropriate for 3D VE mainly because joysticks have two degrees of freedom, three if we count yaw control by way of twisting them compared to the computer mouse. Projects, such as the HERL Isometric Joystick, has proved to be a user-friendly clinical tuning interface that allows a user or clinician to customize it using a program that displays on a computer screen, even to control interfaces for electric
CHAPTER 2. BACKGROUND AND RELATED WORK

powered wheelchairs [DSC+07].

Touchscreen

Direct-touch or touchscreen have been presented as another way to interact with the PC. They can be defined as screens through which data can be input just by touching it with a finger. Items are selected just as they would be with a mouse pointer or a light pen.

The primary touchscreen advantage are the multiple ways the UI can be designed compared to a set of fixed physical buttons. Moreover, people with low or no computer literacy find using touch screens easy and motivating because they have proved to be faster, more intuitive to interact with the PC and they do not need too much training. A study directed by Clifton Forlines [FWSB07] indicates that users may be better off with their fingers for bimanual input when working on a large, horizontal display. As a result, today they are often found in public places such as ATM, ticket collection terminals at cinemas or airport and interactive museums.

However, as candidates to replace the computer mice, they are not suitable for inputting large amounts of data, as they are not very accurate. In the study led by Clifton Forlines [FWSB07], people felt more comfortable using a mouse for unimanual input - tasks executed with one hand and this study showed that a direct-touch input modality do not lead to greater performance in terms of speed and accuracy for unimanual tasks.

Limitations of touch-screens are mainly in their speed and in the level of information, the depth of commands and arm fatigue. Also, selecting detailed objects with fingers on a touchscreen can be strenuous since it is difficult to accurately point at objects that are smaller than one’s finger [FWSB07].

2.4.3 Evaluation: Personal Experience

The popularity of video games had impacted the use of most of the input devices. It is common to see in multiples iPads, mobile phones, even on some arcade consoles novelty gamepads or touchscreen applications, where the users are able to control objects by the use of their fingers or through tilting their hands.

The number of input devices for providing multi-dimensional spatial input data to a computer is large. A touchscreen sounds to be an ideal candidate for navigation in 3D since in SV, we can find specific visualisation software such as VISIONAIR or Drishti Prayog [GSM+14].

In August 2015, I had the chance to manipulate 3D objects with one of these touchscreen devices in the Natural History Museum in London (see figure 2.9). Based on the 16 exhibits I was able to participate with I had a good experience mainly because I was able to see high resolution images, to grab 3D objects with my finger, zoom and rotating objects, understand and participate in the activities proposed by the interactive display. Speaking of the personal experience, it was difficult to navigate the 3D VE in some of the exhibits in the Museum, because the area to place my fingers were dirty or they were not accurate enough, causing fatigue.

Therefore, joysticks appear to be the second best option to immerse the user within the VR. Analogue sticks generally allow for greater precision in a 3D VE. Well known as multi-axis input devices, joysticks have evolved for controlling computers, controlling computer graphics applications in the field
of CAD, computer games and for control of machines such as construction equipment, robotic manipulators, vehicles and the like. A multi-axis input transducer apparatus, such as the one invented by Henry Obermeyer, Fritz Obermeyer and Leslie Obermeyer provides 6DoF by using light sensors and having a sufficiently small size to permit easy finger-tip manipulation of the active grip [OOO05]. However, research has shown users can find this technology challenging to master and rather difficult to operate, as a result, taking some considerable time to becoming an expert.

Hence, most users decide to keep using the ubiquitous computer mouse. Perhaps, its fast enough respond and the fact of being easy to grasp and manage make this device so popular.

These limitations have led some researches to find alternatives for navigation in 3D VE. A study conducted by Tom Tian’s multi-dimensional computer mouse [Tia00], pretend to operate and control objects in a three-dimensional fashion with the mouse. In order to provide the third-dimensional control of a computer mouse, Tom Tian added a joystick to the computer mouse. Nevertheless, due to limitation imposed on by physical size of a computer mouse this device is difficult to operate [Tia00].

Topics that are used in this project have been described including VR, SV, medical imaging, input device classification and some examples of applications for 3D manipulation. The next chapter presents the full 3D architecture, capabilities and evaluation process of the specific Wing 6DoF input device.
Chapter 3

System Architecture of the Wing

“After working on something for so long it can become a little bit difficult to let it go. My main purpose for a long time now has been getting the engineering design finished. Sure, there was a lot of other things to do, but it was all about the product.” Sam Worthington, about the pre-production prototype of the Wing.

In the previous chapter we talked about VR, the technology that aims to provide almost real and believable experience in a synthetic way. SV is now one of the most common applications of VR for research of large amount of data with a major impact in medical imaging. Good choice of input devices allow the user to “step inside” the synthetic world and in the previous chapter the advantages and disadvantages of three popular input devices were described: the computer mouse, the isometric joystick and the touchscreen.

This chapter presents the full 3D capabilities of pitch, roll and yaw to control the 3D objects in immersive environments by using the Wing, a 6DoF input device created by Worthington Sharpe Ltd.

Over the rest of this chapter, the system architecture of the Wing is presented as follows:

- Section 3.1 The description of the Wing
- Section 3.2 The Wing’s mechanical design and operation
- Section 3.3 The Wing’s principal 3D functions and capabilities to navigate in 3D VE.
- Section 3.4 The Wing’s calibration and extrema points
3.1 The Wing: Beyond the 2D Mouse

“Really the best optimal design for games is minimal input for maximum output - that’s the way that games work best. When you watch people playing with a mouse and keyboard, you see them barely moving their fingers and hands but on screen you see crazy movement and all kinds of stuff.” John Romero, designer and developer in the video game industry. [Bri14]

Figure 3.1: Wing shell and base parts part-machined.

Sometimes people are initially surprised that with today’s touch screens, gesture recognition and VR products that provide fast access to any and all types of digital media, people are still able to find a large number of computer mice in the market and in many different topics of research.

However, the truth is that despite its rather humble nature, the computer mouse is extremely well suited to fast, accurate input and all-day use. This led Sam Worthington to the creation of a precision pointing mouse for the plotting and adjusting of way points, and the inclusion of 3D functions for manual aircraft control as required, or for camera gimbal control: the Wing.

3.1.1 A Bit of Background

Worthington Sharpe Ltd. (WSL) was founded in 2005 as a mechanical engineering design consultancy in Lancashire, UK [Wor14a]. Much of WSL’s early work was with Unmanned Aerial Vehicles (UAVs) - often called drones, and working on projects in consumer electronics, industrial equipment and sub-sea oil and gas.

WSL visualised an idea for an input device to replace the computer mouse but also to offer full 3D capabilities, following John Romero’s way of thinking, “minimal input for maximum output”. Hence, WSL developed the Wing, a new product that combines full 3D control incorporating the precision of a professional laser mouse, ideal for any application requiring the speed and accuracy of a computer mouse, as well as intuitive 3D interaction [Wor14a].

Sam Worthington mentioned that in the beginning it was not easy manufacturing an input device such as the Wing because of the first 3D printed prototype which “gave some level of confidence but it
was needed to do much more”. Thus, WSL went from a machined plastic design to aluminium in order to ensure that even the most delicate movements can be transferred directly from the user to the sensors.
Figure 3.2 shows the final design for the second generation of the Wing, which has a similar form to a computer mouse. This design is the used in this project and not only can it be moved on a desk like a normal mouse, but it also has an upper part which can be pitched, rolled and moved vertically, relative to the lower part. Finally, the Wing has a “yaw bar” positioned underneath the front of the upper part, which adds a further control axis. Figure 3.3 shows the control features of the Wing as well as all its possible movements.

3.1.2 Applications for the Wing

Applications for the Wing are where full 3D control navigations in immersive environments and the precision of a computer mouse are needed. Some examples are described below:

- Manoeuvre UAVs or drones. The Wing has been used to operate an UAV created by HexCam - a drone filming company based in Norwich (see figure 3.4). During this test, the user “felt more...”
than comfortable with the Wing as he was able to manoeuvre the aircraft adeptly” [Wor14b].

- PC Ground Station Features such as DJI (http://www.dji.com/) and Ardupilot (http://ardupilot.com/) [Wor14b].
- Professional R/C helicopter flight simulator, also for multicopters, tricopters, quadrocopters, hexacopters, and 3D quadrocopters, such as Heli X5 [Wor14b].
- 3D design software for entertainment, natural resources, manufacturing, engineering, construction and civil infrastructure. The Wing has been tested in software such as PTC Creo 2, Creo 3, and Autodesk, which are design solutions enabling companies to innovate their product design and development process. [Wor14a].

3.2 Mechanical Design and Operation

Figure 3.5: The principal components of the Wing are all CNC machined from aluminium.

The principal parts of the Wing are machined from aluminium and anodised ceramic, as we can see in figure 3.5. Anodising is a process for producing decorative and protective films on articles made from aluminium and its alloys. It is essentially a process where a thick film of aluminium oxide is built up on the surface of the aluminium through the use of a direct current electrical supply, and commonly there is a Type I anodising process, which uses chromic acid and a Type II, which uses sulphuric acid [Vel03]. Anodising processes produce surfaces which are hard, resistant to corrosion and aesthetically pleasing.

The main reason for having the parts of the Wing made of anodised ceramic is to give a scratch resistant and robust finish to this input device. Therefore, the Wing uses an anodising process called Diamondyze™. Dyamondyze improves the surface finish of aluminium by around 30%, creating a hard highly wear resistant surface compared to any hard anodising coating process and making the
external growth so small that it is almost unnoticeable. In addition, Dymondize has high corrosion and chemical resistance, being an excellent ceramic thermal barrier and reducing hotspots [doi13].

The secondary buttons are vacuum cast polyurethane. Cast polyurethanes are a diverse and versatile group of materials that are known for abrasion resistance, chemical resistivity, stability in water, ease of processing, high strength, ageing resistance and low cost. Also, cast polyurethanes are a good substitute for a number of materials where improved performance is required [CD96].

Each aspect of the Wing is designed to seamlessly transfer movement to the sensors. The principal dimensions and components are shown in Figure 3.6 and Figure 3.7 [Wor14a]. As we can see in figure 3.7, the key engineering features of the Wing are:

- The lower part contains the laser mouse sensor:
  
  - **Three Hall-Effect sensors.** Three Hall-Effect sensors are used in combination with two neodymium rare-earth magnets to create a signal recognisable to the PC as a mouse and 4-axis joystick.
  
  - **Fully Ball-Raced Axes.** Dual ball-bearings are fitted to each axis in order to take full advantage of the sensors. This ensures that all movements from the user are immediately transferred through to the sensors and the software with minimal loss of accuracy and without unwanted movements.
The upper part attaches via a precision slide and pivot mechanism to provide pitch and roll and $z$-height movement.

- The pitch and roll movement is achieved through a pair of gimbals similar to that on a gyroscope mounting (see figure 3.7 bottom). Springs provide a force that securely holds the upper part in the neutral position. This feature is critical in preventing any unwanted movements.

- A yaw-bar positioned towards the front with side paddles. The yaw bar is machined from solid aluminium for rigidity and the pads are ergonomically shaped and rubber-coated for maximum purchase. It forms the primary grip of the Wing and enables not only the yaw but also the pitch and roll functions to be operated without interfering with the main left and right buttons or the scroll wheel.

- Side buttons on both sides of the device, include two primary buttons, two buttons for the mouse sensor resolution and a scroll wheel.

### 3.3 Wing 3D functions

a) Mouse functions. The mouse functions of the Wing are operated in the typical manner and the Wing can be used as a direct replacement for the standard mouse. The design is ambidextrous and intended to be operated by the user’s principal hand. The other hand is then free to operate the keyboard.

b) Primary buttons. The primary buttons work as normal left and right mouse buttons.
c) **Mouse sensor resolution.** Two small buttons behind the scroll wheel change the resolution of the laser mouse sensor, effectively changing its sensitivity. The change in sensitivity is indicated by the colour of the scroll wheel light.

d) **Scroll wheel.** The scroll wheel has two operating modes: Either as a normal scroll wheel; alternatively, pressing the upper-left side button disables the z-axis sensor and outputs from the scroll wheel act as the joystick throttle. Pressing the scroll wheel in scroll-wheel throttle mode centres the output. Pressing the upper-left side button again resumes normal scroll wheel and z-axis operation.

e) **Side buttons.** Two buttons on either side are designed to be thumb operated. They are provided on both sides for ambidextrous use only and we suggest the buttons opposite the thumb should not be used.

f) **Joystick functions.** The upper body can be pitched, rolled and pressed down relative to the lower body and the yaw bar twisted. This provides 4-axis joystick functionality. By grasping the two yaw bar paddles between the thumb and the fourth and fifth fingers, it is possible to easily manipulate the upper body and yaw bar without inadvertently clicking the primary buttons.

g) **Roll, pitch and yaw angles** are routinely used to represent the orientation of rigid bodies in aerospace, navigation and robotics because they minimize the dimensionality of the control problem [AT87] (see figure 3.8). Motion about the longitudinal axis is termed **roll** and in 3D objects, it determines the rotation around the front-to-back axis. Motion about the perpendicular axes is called **yaw** and, it determines the rotation around the side-to-side axis. Finally, motion about the lateral axis is called **pitch** and it’s a measure of how far a 3D object is tilted up or down. Based on this, the Wing can do:

   (a) **Pitch.** Pressing the front of the paddles down pitches the upper body down; while pressing towards the back of the upper body pitches up. This is interpreted as the standard joystick Y-Axis within Windows. (See figure 3.3, of section 3.1.1)

   (b) **Roll.** Grasping the paddles and rocking the upper body left or right imparts a rolling motion, which is interpreted within Windows as a standard joystick X-Axis. (See figure 3.3, of section 3.1.1)

   (c) **Yaw.** Grasping the side paddles and twisting them imparts a yawing motion. This is interpreted within Windows as the rudder axis. (See figure 3.3, of section 3.1.1)

h) **Z-Axis.** The whole of the upper body can be pressed down relative to the lower body. Windows interprets this as a standard joystick throttle. Pressing the upper-left side button disables the sensing of this function as described in the scroll wheel section.

   **Note:** in this project we refer to the rudder axis as “Z-axis” due to the current limitation of this axis in this second generation of the device and the improvement of the Z-Axis in the new generations of the Wing.
Figure 3.8: Roll-pitch-yaw representation. A series of three ordered right-handed rotations is needed to coalesce the reference frame with the rigid body.
3.4 Calibration and extrema points

Definition of Calibration

According to ISA’s The Automation, Systems, and Instrumentation Dictionary, the word calibration is defined as “a test during which known values of measurand are applied to the transducer and corresponding output readings are recorded under specified conditions.” [ITIS03]

An interpretation of the definition would say that a calibration is a comparison of measuring equipment against a standard instrument of higher accuracy to detect, correlate, adjust, rectify and document the accuracy of the instrument being compared.

Calibration has four important characteristics [ITIS03]:

- **Calibration Tolerance.** Every calibration should be performed to a specified tolerance.
  - **Accuracy.** The ratio of the error to the full scale output or the ratio of the error to the output, expressed in percent span or percent reading, respectively.
  - **Tolerance.** Permissible deviation from a specified value; may be expressed in measurement units, percent of span, or percent of reading.

- **Accuracy Ratio.** Describes the relationship between the accuracy of the test standard and the accuracy of the instrument under test.

- **Traceability.** All calibrations should be traceable to a nationally or internationally recognized standard.

- **Uncertainty.** Parameter, associated with the result of a measurement that characterizes the dispersion of the values that could reasonably be attributed to the measurand.

Every instrument has at least one input and one output so calibration and ranging are two tasks associated with establishing an accurate relationship between the instrument’s input signal and its output signal. On one hand, to calibrate an instrument means to check and adjust (if necessary) its response so the output accurately corresponds to its input throughout a specified range. On the other hand, to range an instrument means to set the lower and upper range values so it responds with the desired sensitivity to changes in input.

Calibration is important because it guarantees that an instrument is capable of measuring to the specifications for which it is rated. Many instruments lose their calibration, and hence their accuracy, over time, therefore it is necessary to recalibrate them on a regular basis. If the calibration process is designed properly, the results will be traceable to standards of measurement.

**Extrema points**

Principally, extrema points are the maxima and minima points where a function reaches the highest or lowest value. As we described before, the purpose of calibration is to present a sequence of points where the input device reacts accurately to the stimulus that it is applied to. The minimum jerk trajectory describes how a system should move from rest to a target location in a desired time. Therefore, extrema

---

1 A quantity intended to be measured; term commonly used in engineering.
points should be chosen to provide a sufficiently large coordinate range to allow the Wing to interact with the user’s Point of Regard (POR)\(^2\) to manipulate the 3D object between those extrema points.

In section 3.2, we described the Wing’s three Hall-effect sensors\(^3\), which are used in combination with two neodymium rare-earth magnets and a laser mouse sensor, to create a signal recognisable to the PC as a mouse and 6-axis joystick. The Wing uses a single USB port to provide the interface to the computer whereas the operating system interprets this as two separate devices: a standard mouse and a standard 6-axis joystick [Wor14a]. As the Wing is associated to an analogue circuit to be recognised as a 6-axis joystick, we have recorded their internal values to monitor both the location of the hand and the target to ensure that the current desired change in hand location was always such that it brought the hand in a minimum jerk path to the target.

Figure 3.9 describes a graph that maps the level of noise that Pygame [McG07] detects on the Wing when no force is applied to it. The blue line represents the roll control, the red line represents the pitch control, the green line represents the Z-axis and finally the purple line represents the yaw control. Based on the graph, there is considerably more noise on the roll and the yaw than on the pitch axis. The Z-axis will not be considered as part of our study as it is in the process of being improved. Nevertheless, we can see some peaks of instability on the Z-axis. Therefore, although this analysis won’t be included in this project, it will be included in the future work on this input device conducted by Sam Worthington and WSL.

Based on the results shown in figure 3.9, we can see different peak values on the axes, but the measurements on the roll and yaw are higher than the zero value of the graph. We magnified the blue and purple line in figure 3.10 and figure 3.11, in order to do a deeper analysis on the roll and yaw. As we can see, there are different peak values repeated in a time interval, which result in different possible delta values (\(\Delta\)). Delta represents an incremental change in a variable, as (\(\Delta\)) or (\(\sigma\)). Thus, those deltas values correspond to the noise which is not filtered at the moment to convert an analogue signal to a digital density. The next sections describe the process where these delta values are used to create an automatic standard calibration routine in Python and Pygame [McG07].

According to the Pygame’s values of the daemon, the Wing shows a slight instability on the roll (X) and yaw (Z). An explanation of this behaviour lies in the construction of the upper part of this input device. The upper part attaches via a precision slide and pivot mechanism to provide pitch and roll movement [Wor14a]. This ensures that all movements from the user are immediately transferred through to the sensors. Therefore, since the yaw is linked to the roll internally, if one of these axes is not stable, the other will be unstable as well.

The data collection shows that the pitch control has a bigger instability than the yaw bar. Figure 3.10 shows that the instability on roll control is more on the right side than on the left. Figure 3.11 shows the instability in the yaw is on the left side of the yaw bar.

**Standard Windows Calibration Method**

Most modern joysticks should not require any sort of calibration. However, every isometric joystick and game controller, such as the Xbox Controller 360 for example, should come with the software used to

\(^2\)The point on the retina at which the rays coming from an object regarded directly are focused.

\(^3\)Hall Effect Sensors are devices which are activated by an external magnetic field used in all sorts of widely available gadgets and products.
install on the operating system. After the joystick or game controller has been installed in Windows, it can be tested through the **Set up USB game controllers** utility in Windows.

The process of calibrating any isometric joystick for a Windows-based computer consists of typing `joystick` or `joy.cpl` in the **Run** or **Search text field** or at the **Windows Start Screen**, and then tapping or clicking **Game Controllers** in the search results. This will launch a pop-up window with a settings tab where the user can follow the steps in the device calibration wizard.

There are some cases where it is necessary to perform additional configuration steps to set up the joystick or game controller to work with games on the computer. This depends on the kind of input device which is being used. In order to do this, the user should check the settings in **game and look for a configuration menu or option for configuring the game controller**.

Moreover, there are cases where advanced users need to code their own subroutines to calibrate their input devices. To achieve this, Microsoft has created a series of functions and messages that support joysticks, as well as other input devices that track positions within an absolute coordinate system, such as a touch screen, digitizing tablet, and light pen. These functions allow the user to perform tasks on the input device such as getting the driver capabilities, capturing the joystick input and processing the device messages (see figure 3.12).
CHAPTER 3. SYSTEM ARCHITECTURE OF THE WING

Figure 3.10: Noise detected in the pitch side of the Wing.

Figure 3.11: Noise detected in the yaw bar side of the Wing.
In addition, extended capabilities provide support for rudder and other devices that use up to 6DoF. In a more recent publication of Microsoft (https://msdn.microsoft.com/en-us/library/windows/desktop/dd757116(v=vs.85).aspx), the joystick API has been replaced by DirectInput as it has more capabilities for modern input devices.

3.4.1 Data Collection and Data Analysis

Analogue Circuits Limitations

While most joysticks, especially USB based ones should just work without additional packages, some may require a setup process to solve most of their limitations caused by the use of analogue circuits, allowing them to give better results and a more pleasant experience to the user.

Although analogue circuits are a fairly inexpensive way to create input devices and can easily generate a digitised signal, analogue components and circuits can have some inherent problems. These include:

- **Drift.** Drifting means that, through no action of the user, the signal would change as temperature and other electromagnetic signals can affect the device’s resistance.

- **Range Calibration.** The range of values could be very different from one device to another. This means that developing games for particular input devices could be very difficult and it produces inconsistency when using different devices on the same software.

- **Centre Calibration.** Most joystick devices have a centre or dead zone where the input device should be at rest. If this zone failed to be at the centre as far as a game is concerned (or drifted for no apparent reason) it will drastically affect the gaming or navigation experience in a negative way.

  - **Dead zone** is the set radius around the centre of the controller axis where user input is ignored. Depending on how sensitive the input device is, holding it in a neutral, centred
position can sometimes return movement to the 3D rigid body where the user normally wouldn’t want any. In this way, adjustments on the dead zone can prevent this unwanted movement in the neutral position.

There are two basic features in the analogue device calibration, minimum range calibration and maximum range calibration for each axis, and a centre calibration. The process that the setup installation program performs typically involves a series of prompts for the user to move the input device, storing the resulting values, and then using the stored values to scale and offset any raw input values so that they matched the desired behaviour (see figure 3.13).

Peaks Analysis and Results

Most of the modern controllers for PCs and especially for new game consoles use digital circuits and components that give consistent performance with no need for calibration or having the risk of drifting. The Wing, however, uses digital and analogue circuits explaining the behaviour shown on the graphs in the previous sections (figure 3.9, 3.10 and 3.11), which correspond to the response of the Wing while it is held in a neutral position.

The peaks on the graphs suggest a reduction in the sensitivity of the Wing to prevent unnecessarily fast control surface movement. Also, it could help a beginner user who is unfamiliar with the Wing and is just starting out or it may counter the issues of an over-sensitive controller (mainly because high or full sensitivity is of most benefit to more experienced joystick users than novice ones). Therefore, we need to work on the dead zone of the Wing.

Pygame [McG07] includes libraries and functions ideal for detecting controller axes and their level of sensitivity in order to set the values for the dead zone. Thus, an extra calibration stage is performed after the Standard Windows Calibration method for calibrating the Wing. The Wing axes values range from (-1, 1), with a value of 0 being centred. During the extra calibration stage, minimum change delta (Δ) values were recorded, that can be used for smoothing and improving the user experience. With these values, a daemon program was created to set the values for the dead zone to determine the range and centre of the Wing.
3.4.2  Delta Change Values Range from (-1, +1)

Based on the recording values, we identified the centre point for each axis and with two extrema to calculate and analyse the values from the range (-1, +1) of the Wing, where -1 represents the maximum movement in one direction and +1 represent the maximum movement in the other. (See figures 3.14, 3.15, 3.16, 3.17, 3.18, 3.19, 3.20 and 3.21)

![Figure 3.14: Normal distribution in roll axis to calculate the centre point.](image1)

![Figure 3.15: Normal distribution in yaw axis to calculate the centre point.](image2)
CHAPTER 3. SYSTEM ARCHITECTURE OF THE WING

Figure 3.16: Normal distribution in roll axis to calculate the first extrema point (+1).

Figure 3.17: Normal distribution in pitch axis to calculate the first extrema point (+1).

Figure 3.22 shows the Wing calibration curves on the roll, pitch and yaw axes (called X, Y and Z respectively), highlighting the calibration centre point and the minima and maxima (extrema points) rotating possible values of the Wing. We can assume from the curve on figure 3.22 that the standard deviation is constant over the calibration range in the axis. Thus, we can calculate the uncertainty based on the standard deviation values to set the values for the centre calibration (note: if a method’s calibration curve is linear and stable, recalibration can be simplified greatly). In this way, we can detect variances and irregularity defects beforehand and make the Wing more precise and even repeatable.

The uncertainty of the Wing is determined by the number of steps from the minima to the maxima.
CHAPTER 3. SYSTEM ARCHITECTURE OF THE WING

Figure 3.18: Normal distribution in yaw axis to calculate the first extrema point (+1).

Figure 3.19: Normal distribution in roll axis to calculate the second extrema point (-1).

extrema point, crossing through zero. The formulae used to calculate the uncertainty of the Wing and to ensure repeatability and reproducibility in the Wing are described below:

\[ N_{Steps} = \frac{\text{max} - \text{min}}{2\sigma} \]  \hspace{1cm} (3.1)

\[ \text{rise} = 2\sigma \]  \hspace{1cm} (3.2)
Figure 3.20: Normal distribution in pitch axis to calculate the second extrema point (-1).

Figure 3.21: Normal distribution in yaw axis to calculate the second extrema point (-1).

\[ \text{steps} = \frac{2}{N_{steps}} \]  \hspace{1cm} (3.3)

Where \( N_{steps} \) in 3.1 are the number of steps for the uncertainty; \( \text{max} \) is the maximum value recorded from Pygame; \( \text{min} \) is the minimum value recorded from Pygame; And \( \sigma \) is the sigma value based on the standard deviation. In 3.2, \( \text{rise} \) corresponds to the distance between the steps. In equations 3.1 and 3.2 the the multiplier is 2 as we are calculating the minimum and maximum extrema points. Finally, in 4.1, \( \text{steps} \) correspond to the full-scale output range of the uncertainty.
Figure 3.22: Analysis of the input values to set the centre and extrema points

Given the current position of the hand with respect to target and the hand’s velocity and acceleration, the above expressions provide a method for calculating the change in hand position, velocity and acceleration, so that the hand arrives at the target in an optimally smooth way.

Moreover, by using the formulae described before, the values for $\sigma$ and for the rise were calculated and presented in the table 3.1. These values are required to calculate the uncertainty of the Wing and to set a centred calibration. As seen in figures 3.23, 3.24 and 3.25, the repeatability and reproducibility of the steps suggest a symmetric probability distribution of 15 steps uniformly distributed on the unit interval [-1, 1] on each axis of the Wing.

<table>
<thead>
<tr>
<th>Axis</th>
<th>$\sigma$</th>
<th>rise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roll (X-axis)</td>
<td>0.057818</td>
<td>0.028909</td>
</tr>
<tr>
<td>Pitch (Y-axis)</td>
<td>0.054186</td>
<td>0.027093</td>
</tr>
<tr>
<td>Yaw (Z-axis)</td>
<td>0.054186</td>
<td>0.027093</td>
</tr>
</tbody>
</table>

Table 3.1: Table with the values for $\sigma$ and for the rise to determine the centre point and the range that the Wing can operate efficiently.
Device measurements may not be reproducible due to operator inefficiency. Therefore, the operator efficiency is just as important as instrument accuracy and precision. This is why training is an important determining factor to ensure we are getting precise measurements and we are correctly using the device.

![Quantization Noise and Signal-to-Noise on axis X](image)

Figure 3.23: Digital signal processing of input values on axis X of the Wing.

In closing, the architecture of the Wing (the second version of this product), its mechanical design and operation, as well as its main components and its 3D functions on VE were described. Additionally, the data collection and analysis of the internal values of the Wing, as well as the calibration method based on models closely related to Fitt’s and Hicks’ law and the calculation of standard deviation on each axis of the Wing to determine its extrema and centre point values were described.
CHAPTER 3. SYSTEM ARCHITECTURE OF THE WING

Figure 3.24: Digital signal processing of input values on axis Y of the Wing.

Figure 3.25: Digital signal processing of input values on axis Z of the Wing.
Chapter 4

Programming Language Implementation

“When someone says: ‘I want a programming language in which I need only say what I wish done’, give him a lollipop.” Alan J. Perlis [PSS81]

In the previous chapter we explained in detail the architecture of the Wing, describing its principal components and the technology involved in its construction. We also talked about the data collection of the internal values of the Wing and their analysis, which led us to the calibration method required to set the centred point and range of the Wing based on its extrema points and their standard deviations.

This chapter presents the details behind the implementation of the Python daemon program, created to calibrate the centre point, to set the range values of the Wing and to link the physical movements of the Wing with the Volume Visualisation Software Drishti. Furthermore, the criteria for the selection of Python and Drishti in this project over other similar products is described.

Covering the programming language implementation, the rest of this chapter is organised as follows:

- Section 4.1 The criteria for the selection of the programming language used in this project.
- Section 4.2 The criteria for the selection of the Volume Visualisation Package used in this project.
- Section 4.3 The development and testing of the daemon.

4.1 Criteria for the Selection of the Programming Language

“No programming language is perfect. There is not even a single best language; there are only languages well suited or perhaps poorly suited for particular purposes.” Hebert Meyer [CP]

Comparison of programming languages is a common topic of discussion among software engineers because they need to take into consideration the advantages and disadvantages of each language. Multiple programming languages are designed, specified, and implemented every year in order to keep up with the changing programming paradigms and hardware evolution.
In this section we present a comparative analysis between Python, Matlab, C/C++ and Java with respect to the following criteria: programmer’s skill, operating system dependence and fit-for-purpose.

4.1.1 Daemon Details and Selection Criteria

At the moment of choosing the required language, we analysed three important factors:

- The first overriding factor is the last one in the preceding subsections: fit-for-purpose.
- The second factor is platform choice. We analysed the possibility of our application to integrate with other applications. The daemon should be easier to implement with other applications.
- Finally, the third major factor is the existing skills base of the programmer.

The practical approach to comparing programming languages which led to the decision to use Python in this project are described as follows.

4.1.2 Characteristics of the Selected Languages

Python

**Python** is an interpreted, object-oriented programming language similar to Perl\(^1\), which is easy to learn, handy to read and write, and extremely flexible. Thanks to its high-level interactive nature and its maturing ecosystem of scientific libraries, it is an appealing choice for algorithmic development and exploratory data analysis as it has gained popularity because of its clear syntax and readability \[PVG^+ 11\]. Yet, as a general-purpose language, it is increasingly used not only in academic settings but also in industry.

Python was conceived in the late 1980s, and its implementation was started in December 1989 by Guido van Rossum, a former resident of the Netherlands, who chose that name from one of his favourite television shows, *Monty Python’s Flying Circus* (http://python-history.blogspot.co.uk/2009/01/brief-timeline-of-python.html). Python allows users to write quick scripts or back-end web development codes and the source code is freely available and open for modification and reuse. Scripts written in Python (.PY files) can be parsed and run immediately. They can also be saved as compiled programs (.PYC files), which are often used as programming modules that can be referenced by other Python programs.

Matlab

**Matlab**\(^2\) is a high-level language and interactive environment used by millions of engineers and scientists worldwide to explore and visualise ideas and collaborate across disciplines including signal and image processing, communications, control systems and computational finance.

Matlab stands for MATrix LABoratory and the software is built up around vectors and matrices. This makes the software particularly useful for linear algebra but Matlab is also a great tool for solving algebraic and differential equations and for numerical integration. Matlab has powerful graphic tools

\(^1\)Perl is a high-level programming language used especially for developing Web applications that borrows features from other programming languages including C.

\(^2\)Matlab is a registered trademark of The MathWorks, Inc.
and can produce both 2D and 3D images. It is also a programming language and is one of the easiest programming languages for writing mathematical programs.

Matlab comes with a set of useful, sophisticated, analytical and statistical tools right out of the box, including libraries for making plots or creating numerical models.

C/C++

C is a powerful system programming language, and C++ is an excellent general purpose programming language. C is a high-level programming language that was developed in the mid-1970s. It was originally used for writing Unix programs but is now used to write applications for nearly every available platform. C++ is a programming language that was built off the C language. The syntax of C++ is nearly identical to C but it has object-oriented features, which allow the programmer to create objects within the code.

Both languages deal with high performance computing and/or high-frequency trading and collection of commands. They are ideal to work with existing infrastructure already written in these languages.

Java

Java is a programming language and computing platform first released by Sun Microsystems in 1995. Java syntax is fairly similar to C++ and borrows ideas from Mesa, Objective-C and Modula-3. Some features of Java are inheritance, exception handling, modularity, strong type checking, garbage collection and polymorphism. The class in java is the fundamental structure component. Java standard library includes extensive I/O facilities, date/time support, cryptographic security classes, distributed computation support, GUI toolkit and system interfaces. Additionally to normal application development by Java, it is used to develop embedded programs (also known as “applets”) for web browsers and other Java enabled platforms. This capability is an important part of Java, and the standard library packages include a security manager to restrict the capabilities of Java applets.

There are lots of applications and websites that will not work unless Java is installed. Java is used for Android devices, game consoles, scientific supercomputers and mobile phones for being fast, secure and reliable.

4.1.3 Ease of Learning

When we are in front of the computer ready to code, we need to consider that the language we decided to pick up first has a lot to do with what we’re trying to learn, what we want to do with our skills and where we want to eventually go from there. Still, some languages are easier to pick up than others, have a community dedicated to teaching or offer more useful skills once you learn them. The easier that the programming language is to learn, the quicker that programmers become productive. Based on our analysis, we conclude that:

- Oracle’s Java is one of the web’s longest standing, persistent and influential programming languages. Java is a lot easier to learn than C/C++ and, while Java isn’t a perfect programming language many schools and classes start with C or C++ because Java gets a lot of its syntax from those earlier languages. The flip-side to Java is that for all of its portability and applicability, it can be quite difficult to grasp, and quite difficult to program effectively and efficiently.
• C and C++ are both amongst the most foundational languages in computer science and programming. However, the code redundancy and the debugging application make C/C++ complex. C++, as an Object-oriented programming (OOP) language, relies on creating and destroying objects constantly, but has no organised approach to memory management; it is up to the programmer to take care of memory management to avoid memory leaks and dangling references. Most other OOP languages abstract this by using a garbage collector, therefore taking this out of the hands of the programmer, but not C++. The requirement to constantly be aware of memory allocation and deallocation, to make sure that every object is freed once and only once, and to never keep a pointer to a freed object, makes C++ a much more challenging experience than most other languages. Therefore, C/C++ are much less programmer-friendly than most of other languages.

• Matlab is easy, maybe even trivial, and can be learned in an hour. Learning to program well using abstraction (functions, structures, etc.) is not. Matlab has no named arguments to functions and Matlab’s functions are called with inconsistent syntax across and (within) toolboxes. In Matlab this usually involves reformatting the data and a totally different function call that generally one has to look up from the instruction manual.

Matlab has many program development tools to make the program easy to use. They include an integrated editor/debugger, on-line documentation and manuals, a workspace browser and extensive demos.

• Python is quite simple to learn compared to the other languages. The official tutorials are easy to follow, useful and practical, making it the easiest language to learn compared to Matlab, Java and C/C++.

In summary, Python and Matlab have the advantage over the other languages in terms of being easy to learn because of their multiple features, adaptation and flexibility.

4.1.4 Ease of Understanding

Most code is written once and read many times, to focus on a particular point (for instance, to fix a bug). Thus, it is important that the reader quickly grasp the essence of what’s happening. Some languages can usually be read easily; but, because it is verbose, we have to read many lines of code to get anywhere. According to this concept, we conclude that:

• C/C++ do not have this problem, they can be read easily; but C/C++ can still be hard to understand. Moreover, C/C++ provided more than 35 operators and almost all operators can be overloaded, a term that beginner programmers can find difficult to address.

• The same thing happens to Java as Java has many features and quite a complicated syntax for a beginner programmer that they can get lost easily.

• Matlab is easier to understand compared to the previous two languages, but Matlab is full of various tools and packages that may confuse programmers, especially those who are not familiar with this language.
• The syntax of Python is readable and clear and allow the programmer to write more compact code because of code reuse. The way it is organized, it imposes some order to programmers. Beginners and experts can easily understand the code and everyone can become quite productive in Python very rapidly.

In summation, Python is demonstrated to be easier to understand compared to its competitors for having fewer “dialects” than the other popular languages.

4.1.5 Speed of Development

There are many reasons that may be considered for choosing a language X over a language Y: ease of learning, ease of understanding, help with enforcement of correct code, performance of compiled code, supported platform environments, portability and fit-for-purpose. However, one important factor to consider is the speed of development. Thus, at the moment of evaluating the speed of development, we need to take into consideration:

• **Interpreted languages.** Some languages are interpreted rather than compiled, for example Python and Matlab. That means that Python code doesn’t compile to machine code, but rather is interpreted on-the-fly. It is possible to compile Python and Ruby to a virtual machine. Interpreted is generally slower than compiled code for various reasons. Not only can interpretation itself be slow, it is also harder to do optimizations. However, if the code spends most of the time on library functions, performance won’t suffer and these languages can be a better option compared to others.

• **Virtual machine.** Some languages are compiled into bytecode, an invented machine code which is then interpreted. Primarily, the bytecode converts each generalised code line into a specific machine instruction that the computer’s processor can understand. Java and C/C++ are examples of compiled languages. While bytecode can be converted to machine code on the fly, the code will probably still run slower. In the case of Java, a virtual machine is used for portability. In the case of C/C++, there might be other concerns such as security.

• **Overhead.** Some languages trade efficiency for security. For example, some versions of Pascal would check that an array is well constructed and releases memory after being used. On the other hand, C/C++ requires that when allocated memory is no longer used, the programmer has to deallocated, which is hard because it relies on the memory of the programmer.

The table 4.1 which is shown below provides execution speed of two version of Python (Python and Python3), two versions of Java (from open source and Sun), C and C++. The figure 4.1 displays the results of table 4.1 on a graph where we can see that C and C++ are the faster languages and Java is the slower.

The overall result is that low level languages such as C/C++ and Java take longer to write, they can take much less memory but they can run much faster. Writing a program in Python will usually be faster than writing the same program in C/C++, Java. For example, in a simple iteration pattern, for-loop or list comprehension, Python can be more succinct compared to the other languages here mentioned. Not
Table 4.1: Number of seconds taken to complete where the code grows text string by adding another string in cycle until it grows up to 1280 Kb.

<table>
<thead>
<tr>
<th>Line size Kb</th>
<th>Python</th>
<th>C</th>
<th>C++</th>
<th>Python3</th>
<th>Java(openJDK)</th>
<th>Java(Sun)</th>
</tr>
</thead>
<tbody>
<tr>
<td>256</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>17</td>
<td>39</td>
<td>38</td>
</tr>
<tr>
<td>512</td>
<td>32</td>
<td>26</td>
<td>8</td>
<td>81</td>
<td>162</td>
<td>157</td>
</tr>
<tr>
<td>768</td>
<td>78</td>
<td>60</td>
<td>19</td>
<td>201</td>
<td>381</td>
<td>371</td>
</tr>
<tr>
<td>1024</td>
<td>144</td>
<td>107</td>
<td>34</td>
<td>373</td>
<td>711</td>
<td>696</td>
</tr>
<tr>
<td>1280</td>
<td>232</td>
<td>167</td>
<td>53</td>
<td>598</td>
<td>1161</td>
<td>1145</td>
</tr>
</tbody>
</table>

In conclusion, based on the previous analysis C/C++ have proven to be faster in their execution than Python.

4.1.6 Fit-for-purpose

Programming languages are islands, each disconnected from the rest. We choose a language for a task and, for better or worse, stick with it. Communicating between programs written in different languages is such a slow, arduous, task that we avoid doing it whenever possible.

Deciding if a programming language is well equipped or well suited for its designated role or purpose is not an easy task. Thus, we conclude:

- While Java is a good language and is highly portable, it is unsuitable for some purposes, such as some game programming and system programming. Also, Java requires more time to compile a large program than a C/C++ compiler.

- Game programming often requires fast access to the screen-display hardware. This can be done by using DirectX, which is available only in C++ or C. However, those languages could make the
program hard to read and to develop and despite of being useful for games development, the code may not be portable on other operating systems.

- Matlab can run on almost all computers and operating systems. For instance, in Microsoft Windows, Linux and Mac OS, Matlab source code, written on one is completely compatible with the others, as Matlab is independent of computers and operating systems. Unfortunately, in Matlab, GUIs (Graphical user interface) seem to be somehow limited compared to other languages. In order to make Matlab suitable for game development, the programmer needs to buy the Matlab compiler which can be extortionate.

The daemon we are looking to build requires a programming language familiar with game interfaces; a programming language easy to learn and use and a programming language able to run fast. Also, we need a programming language which can help us to analyse all the results we can get from the experimental section. Python fits those requirements because:

- The standard library of Python comes with a number of modules that can be used both as modules and as command-line utilities. Furthermore, besides incorporating modules, Python includes exceptions, dynamic typing, very high-level dynamic data types and classes [Ros95].

- Python has interfaces to many system calls and libraries, as well as to various Windows-based systems, making any Python’s file highly portable between operating systems. Like Perl, Python offers a portable set of bindings that supports portable GUIs across Unix, MacOS, and Windows [Ros95].

- Python packages integrate with other scientific computing libraries [Ros95]. Some of them, which we will use to analyse the effects of the Wing on the participants, are:

  - Matplotlib is a Python-based plotting library with full support for 2D and limited support for 3D graphics, widely used in the Python scientific computing community. It provides hardcopy of the plots in multiple formats such as portable network graphics (PNG), portable document format (PDF) and scalable vector graphics (SVG), as examples.

  - SciPy is an open source Python library used by scientists, analysts, and engineers doing scientific computing and technical computing. The SciPy library depends on NumPy, which provides convenient and fast N-dimensional array manipulation. The SciPy library is built to work with NumPy arrays, and provides many user-friendly and efficient numerical routines such as routines for numerical integration and optimization. Together, they run on all popular operating systems, are quick to install, and are free of charge.

  - NumPy is the fundamental package for scientific computing with Python. It contains among other things, a powerful N-dimensional array object; sophisticated tools for integrating C/C++ and Fortran code; useful linear algebra, Fourier transform, and random number capabilities. Besides, NumPy can also be used as an efficient multi-dimensional container of generic data.

- Perhaps, the most important feature of Python which makes it the right choice for this project, it is that Python has a library called Pygame.
Pygame

Pygame is a cross-platform library designed to make it easy to write multimedia software, such as games, in Python which is proven to be a popular language for writing game simulations [Idr13]. In that way, Pygame is a Python wrapper for SDL (Simple DirectMedia Layer) and it was written by Pete Shinners.

SDL is a cross-platform development library designed to provide low level access to audio, keyboard, mouse, joystick, and graphics hardware via OpenGL and Direct3D [CDF+94]. By using Pygame, a programmer can write games or other multimedia applications in Python that will run unaltered on any of SDL’s supported platforms (Windows, Unix, Mac OS, and others), become a highly portable file among any operating system.

As a convenience, most of the top-level variables in Pygame have been placed inside a module named pygame.locals [Idr13]. This is meant to be used with from pygame.locals import *, in addition to import pygame. By using import pygame, all available Pygame submodules are automatically imported. Furthermore, pygame.init() initialises all imported Pygame modules [Idr13] and an example of how to use it is displayed below.

```python
import pygame
from pygame.locals import *
...
def main():
    pygame.init()
    hScreen=pygame.display.set_mode((256,240))
...
```

Moreover, Pygame provides user input handling (mouse, keyboard, joystick) and game output via screen (shape drawing, image blitting, font rendering) and speakers (effects and music)[Idr13]. In order to do this, Pygame uses pygame.joystick, a module for interacting with joysticks, gamepads, and trackballs.

Each instance of the Joystick class represents one gaming device plugged into the computer. If a gaming pad has multiple joysticks on it, than the joystick object can actually represent multiple joysticks on that single game device [Idr13]. The quick way to initialise the joystick module is by using the following code:

```python
pygame.joystick.init()
```

Last but not least, timing is crucial in Pygame. Without timing control, any application in Pygame will run as fast as it possibly can on whatever platform it happens to be on, losing the control or having...
undesired movements. Thus, a timing control is required to be added and for this Pygame has the module called `pygame.time.Clock()` which represents the time in milliseconds (1/1000 seconds), as it is shown in the following code:

```python
clock = pygame.time.Clock()
FRAMES_PER_SECOND = 30
deltat = clock.tick(FRAMES_PER_SECOND)
```

Most platforms have a limited time resolution of around 10 milliseconds. Therefore, Pygame has a module called `pygame.time.get_ticks()` that returns the number of milliseconds since `pygame.init()` was called [Idr13]. This module instructs the clock object to pause until 1/30th of a second has passed since the last call to tick. This effectively limits the number of calls to tick to 30 per second [Idr13]. The example below shows the syntax and use of this module.

```python
if pygame.time.get_ticks() - last_fps > 1000:
```

In summary, these feature were an asset for our daemon, as we developed an interface between the Wing and Drishti able to record the Wing internal values in different intervals of time to set the range values for the Wing and send the instructions to rotate the 3D objects in Drishti.

### 4.1.7 Final Decision

Based on the results displayed in the table 4.2, we decided to implement the daemon in Python for being a language that satisfied our demands for this project, also for being a language easy to learn and use, for being a language that provides the tool to synchronise the movements of the Wing with Drishti via `pygame.joystick` and, last but not least, for being a language comfortable for the programmer.

<table>
<thead>
<tr>
<th>Language</th>
<th>Ease of Learning</th>
<th>Ease of Understanding</th>
<th>Speed of Development</th>
<th>Fit-for-purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>C/C++</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Java</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Python</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Matlab</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.2: Language selection criteria. Python was selected for getting 3:4 compared to the other languages in terms of ease of learning, ease of understanding and fit-for-purpose.

In addition, as we can see on the table 4.2, some honourable mentions are C/C++ and Matlab to be good candidates to create the daemon to link the Wing with Drishti. Nevertheless, the license of Matlab is expensive whereas to build a code C/C++ take some considerably time, which in the case of this project, time is limited. Building a code in Python doesn’t take too much time and Python is open source, as result, Python is for free.

In summation, so as to create the daemon, **Pygame** was used to manage the Wing on a computer and send the instructions to rotate the 3D objects, whereas **Python** was used to write code that generates the packages for the movement, send these instructions through the port used by the volume visualisation software and generate the user interface behaviour.
Last but least, for commercialisation purposes, which is one of the main aims of Sam Worthington and WSL\textsuperscript{2} as they are looking to commercialise the Wing in large scale, it is highly recommended to implement the daemon for the extrema and centre points in C/C++, primarily for being faster compared to the Python’s compiler.

4.2 Criteria for the Selection of the Volume Visualisation

A volumetric dataset is a discrete representation of a spatial object, whose geometry is often too complex to be defined explicitly. For instance, a sequence of slices obtained from computed tomography (CT) is a typical volumetric dataset. Volumetric datasets usually require large amounts of storage space, describe both the exterior and interior of objects, and contain both solid and amorphous components. This results in an information overload in each dataset and thus difficulties in its visual presentation.

In this section we will talk about the visualisation software used for the experimental test which principal objective is to extract meaningful information from volumetric data using computer graphics techniques. Our task is to integrate the dual-usb 6DoF mouse device with a visualisation volume software. The possible options of visualisation volume were Drishti, ParaView and TomVis. Before choosing which visualisation package to use, we need to take into consideration:

- The person operating the the visualisation software must be concerned with the following topics:
  - The software and hardware requirements necessary to run a particular package.
  - The software category: turnkey application or application builder (these concepts will be described below).
  - The implementation of the user interface and its ease of use.
  - The support for application building and functional extension.
  - The support for distributive processing and computational steering.
  - The documentation and help facilities (tutorials, user’s manual, etc.).

- Furthermore, the visualisation software’s specific topics should include:
  - Internal data types.
  - Support for data manipulation.
  - Support for general data formats.
  - Available visualisation techniques.

- Finally, the the visualisation software’s graphics specific topics should include:
  - Different rendering.
  - Viewing options.

Therefore, we can divide the visualisation software in two groups:

\footnote{Worthington Sharpe Ltd.}
• **Turnkey** applications enable the user to select operations and options by pushing buttons (keys) only. Hence, it does not require a lot of knowledge to operate them. However, these applications do not allow the user to extend the set of available operations.

• **Application builders**, in contrast to turnkey visualisation software, they allow extension by user-written operations and they provide a much greater flexibility in the processing and visualising the data. The user builds an application for the visualisation of his data by connecting modules, each performing a specific action on the data. The connections between these modules represent the data flow between them.

In summation, the volume techniques we select to visualise volume data depend on what type of data we have and what we want to learn from it. As we are aiming medical imaging and scientific visualisation (SV) in this project, the visualisation package we select should be able to provide these applications with a more elegant technical foundation and a set of more powerful software tools. The next section described the main features of the volume packages proposed for this project.

### 4.2.1 Volume Visualisation and Rendering in Medical Imaging

The advances in volume visualisation over the past decades, coupled with the rapid increase in computer power, suggest that volume visualisation may be developed into volume graphics [KCY93] [CKY12] as a general purpose graphics technology. The noticeable advances in this field include:

• **Voxelisation** — Algorithms and software tools for converting various surface-based representations to volumetric representations.

• **Constructive volume geometry (CVG)** — Theory and methods for modelling interactions of volume objects in a way similar to constructive solid geometry (CSG). The fundamental differences between the two include that (a) CSG operates in the Boolean domain, while CVG operates in real domain; (b) CSG handles geometrical properties but not physical properties (e.g. colour and opacity), while CVG deals with physical as well as geometrical properties.

• **Interactive volume sculpting** — Methods and software tools for constructing and manipulating volume models, using surface-based operations such as cutting and drilling as well as volumetric operations such as burning and painting.

• **Domain-based modelling and rendering** — Methods for transforming spatial representations of volumetric datasets into other domains, such as compression, frequency and wavelet domains, and algorithms for rendering the data in such a domain directly.

• **High performance computing** — Researches in areas of parallel and distributed computation and special purpose hardware for volume rendering.

• **Photo-realistic rendering** — Methods for synthesising images involving shadows, global illumination, reflection, refraction and solid- and hyper-textures in volume rendering.

• **Non-photo-realistic rendering** — Methods for simulating painting techniques (such as line drawing and oil painting) in volume rendering.
• **Image-based rendering** — Methods for rendering a volumetric scene into a set of images according to some predefined views, and using these images to form a virtual environment where a viewer can navigate around in real time. Some image-based rendering techniques store the pre-rendered images in the form of 4D and 5D volumetric datasets.

• **Irregular grids** — Methods of manipulating and rendering volumetric datasets defined upon irregular grids which are commonly used in scientific computation.

• **Volume deformation** — Methods for controlling the deformation of volumetric datasets for applications such as correction of data acquisition errors, volume morphing, and extraction of internal structures (e.g. ray reflectors).

• **Volume-based animation** — A new development attempts to model the kinematics and dynamics of actors in volumetric representations.

### 4.2.2 Characteristics of the Visualisation Volume Packages

As it was stated in the previous section, when choosing a data visualisation tool, we must consider the volume of data that we can represent and the format of incoming data. This section contains an overview of the possibilities of the 3 visualisation packages proposed to work with the Wing and Python. These visualisation packages are: Drishti, ParaView and TomViz, and they were selected as they provided tools to manipulate 3D datasets and their authors were responsive.

**Drishti**

Drishti is open source software created by Ajay Limaye at the ANU (Australia National University); Drishti stands for vision or insight in Sanskrit, an Indian language. This product was written for visualising medical imaging such as CT data and/or electron-microscopy data. The central idea of Drishti is that scientists can use it to explore volumetric datasets without extensive training (see figure 4.2).

Drishti is available for Microsoft Windows, Macintosh, and some Linux flavours. Also it is available as source code and provides a number of features for 3D visualisations (see figure 4.3):

• All rendering performed on the GPU.

• 2 dimensional transfer functions: uses both the gradient and the voxel value in the shading model.

• Enables diffuse / specular shading and shadows.

• Consists of two main programs, a data importer and the render.

• Importer supports various commonly found volumetric file formats, resampling volumes, cropping, and filtering.

• Powerful keyframe based animation of (almost) all rendering and camera controls.

• Arbitrary clipping planes.

• A number of ways to tradeoff quality for interactive performance.

• Able to combine / superimpose up to 4 volumes.
ParaView

ParaView is an open-source, multi-platform data analysis and visualisation application. ParaView was developed to analyse extremely large datasets using distributed memory computing resources. It can be run on supercomputers to analyse datasets of petascale size as well as on laptops for smaller data.

ParaView possesses two interaction modes: interactive mode based on GUI and batch mode employing Python scripting (see figure 4.4). ParaView has an open, flexible, and intuitive user interface. Furthermore, ParaView is built on an extensible architecture based on open standards. ParaView runs on distributed and shared memory parallel as well as single processor systems and has been successfully tested on Windows, Linux, Mac OS X, IBM Blue Gene, Cray XT3 and various Unix workstations and clusters.

TomVis

Tomviz is a branded ParaView-based application tailored for visualising electron tomography data. It can utilize the large quantities of memory and processing resources required to render, manipulate and analyse voluminous 3D tomography. The platform provides a robust graphical interface where objects can be rendered as shaded contours or volumetric projections. Multiple datasets, colormaps and other visualisation settings can be used in combination and these objects can be rotated, sliced, animated and saved as image or video files. Data collected can be further analysed through histograms, multi
correlative statistics and filters to name a few. The results will be open source, meaning that new algorithms can be readily implemented through its Python scripting interface or the core C++ application (see figure 4.5).

4.2.3 Choosing a Visualisation Application Package

For the final decision, all features of these volume visualisation packages were analysed. These visualisation tools can make simple charts and graphs, and offer a robust set of additional capabilities that we can use to learn more from the data. However, as TomViz is in its experimentation stage, it has limited
resources and lacks 3D capability to support large datasets, thus this software was discarded.

Other factors analysed were the scalability and customisability of the visualisation software because some visualisation tools have a large number of features that are confused to operators. Therefore, the software for this project should be flexible and intuitive to be tailored to the needs and knowledge of
operators as they are from different areas as it will be explained in chapter 5. In this way, Drishti fitted in this context as, compared with ParaView, it was easy to use, its author was very responsive to make queries about the software, provided tools appropriate for medical imaging and SV exploration and its features were appropriate to link the Wing within it through the use of some commands written in Python.

Note: the current version of this software, Drishti version 3 is not a production version, but only to test rendering for massive volumes. Therefore, in this project was used the latest version, Drishti 2.5.1, available for download from the releases section on https://github.com/AjayLimaye/drishti for Windows, Mac and Linux.

4.3 Development and Testing of the Python Daemon

While The Python Language Reference describes the exact syntax and semantics of the Python language, for our daemon we added two important libraries: socket and Pygame. From Pygame, we imported all the functions to access and manipulate joysticks input devices [Idr13], as the Wing has a dual behaviour as explained in chapter 3. Below, it is shown the libraries used in this daemon:

```python
# LIBRARIES
import socket
import time
import pygame
from pygame.locals import *
import os
```

Where:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>import socket</td>
<td>Module that implements an interface to the socket communication layer.</td>
</tr>
<tr>
<td>import time</td>
<td>Module that provides various time-related functions.</td>
</tr>
<tr>
<td>import pygame from pygame.locals import *</td>
<td>Imports all the available Pygame modules into the pygame package</td>
</tr>
<tr>
<td>import os</td>
<td>Module that provides a portable way of using operating system dependent functionality.</td>
</tr>
</tbody>
</table>

Table 4.3: Python libraries used in this project.

The daemon used in this project is divided in three sub-modules:

- A sub-module to creates a socket, bind it to a port and then accept connections.
- A sub-module to set the extrema and centre points of the Wing.
- A sub-module to send instructions through the open socket in order to map any movement coming from the Wing, thus moving the 3D image displayed on Drishti.

For further details, the daemon code is included in appendix A of this thesis.
CHAPTER 4. PROGRAMMING LANGUAGE IMPLEMENTATION

4.3.1 Socket Function

A socket is one endpoint of a two-way communication link between two programs running on the network [Sri97]. Python provides two levels of access to network services [Ros95]. At a low level, we can access the basic socket support in the underlying operating system, which allows us to implement clients and servers for both connection-oriented and connectionless protocols [Ros95]. Below it is shown the fragment of code of the socket function:

```
# Symbolic name, meaning all available interfaces
HOST = 'localhost'
# Arbitrary non-privileged port
PORT = 7755
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
print 'Socket created'
# Bind socket to local host and port
try:
    s.bind((HOST, PORT))
except socket.error as msg:
    print 'Bind failed. Error Code : ' + str(msg[0]) + 'Message' + msg[1]
sys.exit()
print 'Socket bind complete'
```

Visibly, the `socket.socket()` function in the socket module was used to create the socket with the general syntax. As we selected Drishit version 2.5.1, the port used was the port number 7755 and it was used localhost as the hostname. Localhost refers to the local computer that a program is running on, in this case, as we are running an application on our computer, then our computer is considered to be the localhost and the port is a virtual identifier that defines a service endpoint.

This fragment of code says that the socket will be created, bound and then put into listening mode, waiting for a client to make a connection request [Ros95]. The Pygame function knows the hostname of the machine on which the server is running and the port number on which the server is listening (in this case, localhost and port 7755). Then, the Pygame function tries to engage with the server on the server’s machine and port and if everything goes well, the server accepts the connection, sending the packages and rotating the images displayed on Drishti.

4.3.2 Setting the Extrema Points and the Centre Point

In order to record the values for each axis, at the moment of connecting the Wing to the computer, the daemon runs and set the values for the extrema and centre points. It was used the module `pygame.joystick` to create a new joystick object to access the Wing. Then, the sub-module `joystick.get_axis` was used to return the current position of a joystick axis, which will range from -1 to 1 with a value of 0 being centred.
# This gets the position of the axis on the game controller.
# It returns a number between −1.0 and +1.0

x_axis_pos = wing_joystick.get_axis(0)
y_axis_pos = wing_joystick.get_axis(1)
z_axis_pos = wing_joystick.get_axis(3)

The values used in the sub-module wing_joystick.get_axis for the roll, pitch and yaw axes were 0, 1 and 3 respectively. Then, the daemon recorded the minimum and maximum possible value on each axis and calculated the centre point. This can be seen in the following fragment of code:

# FUNCTION TO SAVE VALUES. USED FOR THE CALIBRATION STAGE

```python
def saveAxisValues(x, y, z, v):
    xValues.append(x)
    yValues.append(y)
    zValues.append(z)
    vStpwatch.append(v)
```

Then, the mean and standard values were calculated for each axis using the code below:

```python
def mean(values):
    return sum(values)*1.0/len(values)

def stanDev(values):
    length = len(values)
    m = mean(values)
    total_sum = 0

    for i in range(length):
        total_sum += (values[i]−m)**2

    under_root = total_sum*1.0/length

    return math.sqrt(under_root)
```

Once having the value of the standard deviation and the mean of each axis, the value for delta of each axis was calculated, using the modules SciPy and NumPy, as it is shown below:

```python
maxVal = np.amax(JOYSTICK_AXIS)
minVal = np.amin(JOYSTICK_AXIS)
delta = (stanDev_MaxVal + stanDev_Centre + stanDev_MinVal)/3
```

3wing_joystick.get_axis(2) was not used as it belonged to the Z-axis of the Wing.
CHAPTER 4. PROGRAMMING LANGUAGE IMPLEMENTATION

After having these delta values, the daemon uses the formulae, particularly:

\[
\text{stepsize} = \frac{2}{N\text{Steps}}
\]  

(4.1)

This will give us the range of values for the extrema and centre points of the Wing. The following code implements the previous formula once the function \texttt{AdjustmentAxes} is called in the main function:

```python
# SET THE EXTREMA AND THE CENTRE POINTS
x_axis_pos = AdjustmentAxes(axes["x"], x_axis_pos, "x")
y_axis_pos = AdjustmentAxes(axes["y"], y_axis_pos, "y")
z_axis_pos = AdjustmentAxes(axes["z"], z_axis_pos, "z")

# FUNCTION TO SET THE CENTRE POINT
def AdjustmentAxes(limit, axis, type_axis):
    varAux = limit
    sigma = sigmas[type_axis]
    varAux = varAux + sigma
    if axis < 0:
        varAux = -1 * varAux
    if axis > 0 and axis <= varAux:
        axis = 0
    elif axis < 0 and axis >= varAux:
        axis = 0
    elif axis == 0:
        axis = 0
    return axis
```

4.3.3 Pygame Function

Previous to set the extrema and centre points, the movements of the Wing were synchronised through the module \texttt{joystick} to consider the time required to reach at the target object (see Fitts’ and Hick’s law in chapter 2). As the 3D object followed a particular path, the distance was considered for finding the movement time and those values were recorded for their further analysis. Below, we can see the implementation of Pygame in the daemon.

```python
while not done:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            done = True
            clock.tick()
        if pygame.time.get_ticks() - last_fps > 1000:
            last_fps = pygame.time.get_ticks()
```
# THE VALUES ARE MULTIPLIED BY 10 TO SPEED UP
# THE MOVEMENT.
rotr = rotr + (x.axis_pos * 10)
rotp = rotp + (y.axis_pos * 10)
roty = roty + (z.axis_pos * 10)

# Saving the values in an array and display
# time.time() - st 1 / 0.01 = time taken to complete a round trip to a server using TCP in min.
if (rotr!=0 or rotp != 0 or roty != 0) and ((time.time() - start) > 0.15):

    rotAngle = abs(round((rotr + rotp + roty)))
    saveAxisValues(x.axis_pos, y.axis_pos, z.axis_pos,
                   pygame.time.get_ticks())
    # 1 is DEG2RAD(angle): angle * PI / 180.0
    multipacket = 'addrotation %s %s %s %s' %(rotr, rotp, roty, (rotAngle))
    s.send(multipacket.encode('utf-8'))
    rotr = 0
    rotp = 0
    roty = 0
    start = time.time();
clock.tick(60)
pygame.quit()

Through this fragment of code, all the possible movements of the Wing were mapped using the
pygame.joystick.event and addrotation4 commands. As a result, the more the user pressed any of the
axes of the Wing (roll, pitch or/and yaw), the more speed of the movements on the 3D object they got.
Consequently, the responses were smoothed to be tested with the participants of the user evaluation
study.

In summary, this chapter described the criteria for selecting Python and Drishti over different similar
options. The main features of the Python as well as Pygame and Drishti and the main characteristics of
the daemon used to synchronise the events sent by the Wing to Drishti were described.

---

4 addrotation is a Drishti clip planes command to rotate a 3D object by an angle about the axis defined by vector x, y, z from its current orientation [Lov11]
Chapter 5

The User Evaluation Study

“True genius resides in the capacity for evaluation of uncertain, hazardous and conflicting information.”
Winston Churchill

This thesis started explaining the importance of Scientific Visualisation (SV) and Medical Imaging in Virtual Reality (VR) and it justified the purpose of the evaluation of 6DoF input devices in these areas. Then, for this project, it was selected the Wing as the product to evaluate and some related and previous works were described. Moreover, it was explained in detail the main functions of the Wing in 3D Virtual Environments (VE) and the reasons of needing a daemon to calibrate the range of values to set the extrema and centre points of the Wing. Finally, in the previous chapter the development and implementation of the Python daemon as well as its main functions were described.

In this chapter, the methods and the results of the evaluation of the Wing with different users are presented. The comments and suggestions thorough the evaluation are discussed. This evaluation chapter is organised as follows:

- Section 5.1 Describes the background profile of the participants.
- Section 5.2 Focuses the task to be performed by the participants.
- Section 5.3 Focuses on the observations, recordings of the evaluation, the questionnaire applied to the evaluation and the comments of the participants regarding the task they performed.
- Section 5.4 Discusses the results of the evaluation.

5.1 The User Group

The participants in our study test came from two backgrounds. The first group, the “experts” group, consisted of participants that have backgrounds in visualisation areas whereas the second group, the non-expert group, consisted of normal participants from other backgrounds. We asked 18 participants (6 female, 12 male) to undergo our experiment and then to complete a satisfaction questionnaire after the test in order to analyse the results from different perspectives. Out of these 18 participants, 9 were from the experts group while the remaining 9 belonged to the non-expert group. The participants’ age ranged
from 20 to 37 years with a mean age of 25.66 years. Moreover, over half of the participants had plenty of experience using CAD (Computer-aided) design packages and 3D games, and only 1 participant had previously worked with the volume visualisation package, Drishti, chosen for this project.

5.2 Experimental Settings

The purpose of this study is to investigate human-computer interactions in relation to the use of the Wing in 3D VE, specifically in the Drishti Volume Visualisation Package, to analyse user’s perceptions. For the experimental test on the Wing, a combination of evaluation methods was selected: user task performance testing included observation and audio recording followed by an interview including a satisfaction questionnaire.

In the first stage, we used a controlled experiment to test the Wing. For that purpose, we formulated the following hypotheses for measuring the performance of the Wing on navigation and manipulation of 3D objects in the Drishti:

- **H1**: The Wing is an effective, efficient and easy to learn input device to manipulate 3D objects in VE.
- **H2**: In a short time, expert users will master the Wing and will have a pleasant experience at manipulating 3D objects. This pleasant experience will lead them to buy the Wing for their work and/or home use.
- **H3**: Non-expert users will have a pleasant experience at manipulating 3D objects and will be interested in buying the Wing for their work and/or home use.

In the second stage, we have used a survey (using a satisfaction questionnaire) to test the users’ perception on the effect of using the Wing in 3D VE. In the following subsections, a detailed description of the experimental design is presented.

5.2.1 Experimental Materials and Tasks

The main instrumentation for the experiment was a 6DoF input device prototype, the Wing, demonstrated in chapter 3 and the visualisation volume software described in chapter 4. For the experiment, Drishti was loaded with a sample dataset of a padlock. In addition to that the following materials (see in figure 5.1) were prepared for the experiment:

(a) HP EliteDesk 800 G1 SFF desktop computer with Intel Core i7-4700 processor and Windows 7 Enterprise OS, with the Python daemon installed. See figure 5.1 a.

(b) HP optical laser mouse. See figure 5.1 b.

(c) Drishti version 2.5.1 for 64-bit Windows 7. See figure 5.1 c. In addition, the figure shows the dataset used in this evaluation.

(d) The Wing version 2.0. See figure 5.1 d.

The experiment consisted of three main tasks, which are as follows:
• **Understandability Task.** In order to ensure that all users performed the test in a similar manner they were given the same instructions and information, with engagement within a 10 minute training period. During this training period a volume dataset was loaded in Drishti and the participants were instructed to manipulate the 3D object through translations or rotations by only using the Wing.

• **Navigation and Manipulation Task.** In this task, through a series of 5 minute directed instructions, participants were asked, by using the Wing and then the mouse, to match an image given to them.

• **Post Task Satisfaction Questionnaire.** Upon completing the experiment participants were asked to perform a post task satisfaction questionnaire. The satisfaction questionnaire consisted of eighteen statements in which the users had to state their opinion, using a Likert scale and answer some open questions which could be correlated to the task results.

### 5.2.2 Participants Selection and Experimental Treatment

As we stated in the previous section, the participants involved in the experiment were a mix of Bachelor and Master students in Computer Science, Pharmaceutical and Chemical areas, and even from other fields such as Graphic Designing and Linguistics. The benefit of using student participants is that they
form a homogeneous group with respect to their academic background, ages and experience. Moreover, the experimental tasks did not require high levels of professional experience, which justifies our selection of participants.

At the beginning of the experiment, the participants were given a short list of written instructions describing the experiment. The experimental test started with a training session, where experiment mentors demonstrated how the Wing could be used to manipulate 3D objects in Drishti. Once participants were familiar with the Wing, the actual experiment was conducted. The participants were given the materials, responses from the 18 participants were received and all the data collected was considered for analysis. During the experimental test we did not provide feedback regarding the accuracy of the provided measurements. Participants were allowed as much time as desired to complete their tasks. The answers of this post-experimental survey provided by the participants are included in appendix C.

5.3 Experiment Details and Results

All the participants completed the tasks that consisted in matching the image given to them with the image created in Drishti, first rotating all the axes of the 3d object with the Wing and then repeating the same but with the computer mouse. The elapsed time for individuals test subjects varied between 3 to 9 minutes excluding training. Subsequently, the Python daemon, using the Keystroke Level Model (KLM) explained in chapter 2, decomposed the experimental task into seconds' level actions: it counted all of the physical operations and added the mental acts where required, recording the overall task execution time. Figure 5.2 shows the time taken by the user to rotate the 3D object with the mouse and the Wing, according to KLM. Visibly, the task was done faster with the computer mouse than the Wing, mainly because users were familiar with the computer mouse and they didn’t need time to learn how to use it in 3D VE.
Figure 5.2: Comparative table of the time per participant to complete the experimental task with the computer mouse and the Wing

From the Wing distribution graph showed in figure 5.2, the graph on figure 5.3 correspond to the histogram of the time taken by each participant to complete the experimental task with the Wing. Based
on the results, the mean and the median are close together - 2.96 and 2.40 minutes respectively. Figure 5.3 histogram shows there are two areas for further investigation. First of all, the values for user P5 (8.16 minutes) and user P11 (6.63 minutes) are higher than the others and should be looked at in more detail. These users stated that they were familiar with visualisation software; however, both mentioned “feeling confused by the rotation of the object” as “their movements do not correspond to the rotation of the object with the Wing,” showing a preference using the computer mouse.

There appears to be a trend that the users P1, P4, P6, P7, P13, P15, P17 and P18 performed faster than the rest of the participants. From these 8 participants, 5 of them had computer science background and 6 belonged to the non-expert group. In particular, the participants who took longer, except for P5, were mostly the participants from the expert group.

5.3.1 The Questionnaire and the Written Comments

After completing experimental test, participants were asked to fill out a satisfaction questionnaire to provide responses regarding their experience after having used the Wing and the computer mouse in the same scene. The satisfaction questionnaire consisted of eighteen statements. The first three questions asked about the characteristics of the users\(^1\). The user’s answers were given on a Likert\(^2\) scale from 1 (None) to 5 (A Lot). The remaining questions were about the Wing, the user’s answers were either given on a Likert scale from 1 (Strongly Disagree) to 5 (Strongly Agree) or open questions. The questions are presented in table 5.1.

\(^{1}\)The results of the first questions of the satisfaction questionnaire as well as the answers to open questions are included in appendix C

\(^{2}\)A scale used to represent people’s attitudes to a topic.
CHAPTER 5. THE USER EVALUATION STUDY

<table>
<thead>
<tr>
<th>Description of the question</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
</tr>
<tr>
<td>Time of the tasks performed by the users</td>
</tr>
<tr>
<td>Q1: Do you have any experience in using 3D user interfaces (e.g. CAD design, 3D games etc.)?</td>
</tr>
<tr>
<td>Q2: Do you have any knowledge about 3D Volume Visualisations?</td>
</tr>
<tr>
<td>Q3: I was able to effectively complete the required tasks.</td>
</tr>
<tr>
<td>Q4: I am satisfied with the ease of completing the required tasks.</td>
</tr>
<tr>
<td>Q5: I believe I understand better how 3D Volume Visualisation works.</td>
</tr>
<tr>
<td>Q6: I would like to use this input device (Wing) for my future work and home use.</td>
</tr>
<tr>
<td>Q7: I found the various functions in this input device (Wing) were well integrated.</td>
</tr>
<tr>
<td>Q8: I would imagine that most people would learn to use this input device (Wing) quickly.</td>
</tr>
<tr>
<td>Q9: I felt confident when using this input device (Wing).</td>
</tr>
<tr>
<td>Q10: I needed to learn a lot of things before I could get going with this input device (Wing).</td>
</tr>
<tr>
<td>Q11: Based on your experience with Wing, if this was available to buy now, would you buy one?</td>
</tr>
<tr>
<td>Q12: Overall how would you rate the Wing?</td>
</tr>
<tr>
<td>Q13: How easy was it to grasp the Wing device?</td>
</tr>
<tr>
<td>Q14: How easy was it to manipulate the Wing device to do rotation or a zooming?</td>
</tr>
</tbody>
</table>

Table 5.1: Table of the satisfaction questions for the experimental tests.

The table 5.2 shows the correlation matrix between the questions with a p-value of 0.468 and a significance level of 5% (\(\alpha = 0.05\)) for two-tailed test to identify if the mean is significantly greater than or significantly less than the p-value, highlighting the statistically significant positive and negative correlations. Some correlations are self-explanatory, e.g. between questions 1 and 2. According to these questions, the Wing was evaluated in terms of utility, usability (confidence, efficiency, effectiveness, engagement and easiness to learn), likeability, cost and acceptability. The results of this evaluation are described in the next sections.

5.3.2 Usability and Likeability of the Wing

At the moment of making decisions about systems, users’ decision depends not upon usability but upon an assessment balancing various factors: how useful the system will be, if they feel comfortable using it, if they would like to use it and how much it will cost. In this way, it is proposed the paradigm shown in table 5.3 to evaluate the usability and acceptability of the Wing.

Effectiveness and Efficiency Results

First, we need to differentiate effectiveness from efficiency as they are not the same: efficiency is concerned primarily with how quickly a task can be completed, while effectiveness considers how well the work is done.

Having established this definition, figure 5.4 shows the correlation between questions 3 and 6 that corresponds to the level of effectiveness in completing the tasks and the interest of the users in having the Wing for their work and home use. The results show a positive reaction on the preference of the users for the Wing mainly because of the sensation of confidence built each time they successfully completed the experimental task.

---

3The p-value is a number between 0 and 1 representing the probability of the data.
### CHAPTER 5. THE USER EVALUATION STUDY

#### Table 5.2: Correlations of the satisfaction questions with p-value = 0.468 and $\alpha = 0.05$. The blue colours are for significant positive correlations and the red for significant negative correlations.

<table>
<thead>
<tr>
<th>Measure</th>
<th>Time</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q5</th>
<th>Q6</th>
<th>Q7</th>
<th>Q8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q1</td>
<td>0.122</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q2</td>
<td>0.188</td>
<td>0.81</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q3</td>
<td>-0.246</td>
<td>0.042</td>
<td>-0.163</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q4</td>
<td>-0.125</td>
<td>-0.483</td>
<td>-0.456</td>
<td>0.222</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q5</td>
<td>0.055</td>
<td>-0.177</td>
<td>-0.39</td>
<td>0.134</td>
<td>0.226</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q6</td>
<td>-0.192</td>
<td>-0.107</td>
<td>-0.216</td>
<td>0.730</td>
<td>0.528</td>
<td>0.517</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q7</td>
<td>-0.018</td>
<td>-0.162</td>
<td>-0.032</td>
<td>0.180</td>
<td>0.597</td>
<td>0.226</td>
<td>0.548</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Q8</td>
<td>0.094</td>
<td>-0.247</td>
<td>-0.239</td>
<td>0.540</td>
<td>0.389</td>
<td>0.290</td>
<td>0.653</td>
<td>0.138</td>
<td>1</td>
</tr>
<tr>
<td>Q9</td>
<td>-0.199</td>
<td>0.119</td>
<td>0.052</td>
<td>0.839</td>
<td>0.095</td>
<td>-0.069</td>
<td>0.490</td>
<td>0.186</td>
<td>0.298</td>
</tr>
<tr>
<td>Q10</td>
<td>0.514</td>
<td>-0.185</td>
<td>-0.214</td>
<td>-0.435</td>
<td>-0.049</td>
<td>0.239</td>
<td>-0.185</td>
<td>0.205</td>
<td>0.192</td>
</tr>
<tr>
<td>Q11</td>
<td>-0.243</td>
<td>-0.1</td>
<td>-0.17</td>
<td>0.619</td>
<td>0.558</td>
<td>0.431</td>
<td>0.941</td>
<td>0.516</td>
<td>0.615</td>
</tr>
<tr>
<td>Q12</td>
<td>-0.238</td>
<td>-0.424</td>
<td>-0.41</td>
<td>0.476</td>
<td>0.707</td>
<td>0.311</td>
<td>0.576</td>
<td>0.637</td>
<td>0.441</td>
</tr>
<tr>
<td>Q13</td>
<td>0.1645</td>
<td>-0.218</td>
<td>-0.201</td>
<td>0</td>
<td>0.454</td>
<td>0.090</td>
<td>0.116</td>
<td>0.030</td>
<td>0.387</td>
</tr>
<tr>
<td>Q14</td>
<td>-0.085</td>
<td>-0.296</td>
<td>-0.341</td>
<td>0.330</td>
<td>0.712</td>
<td>0.245</td>
<td>0.469</td>
<td>0.5312</td>
<td>0.163</td>
</tr>
</tbody>
</table>

#### Table 5.3: The paradigm of usability and related concepts.

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>UTILITY</strong></td>
<td>- will it do what is needed functionally?</td>
</tr>
<tr>
<td>+</td>
<td></td>
</tr>
<tr>
<td><strong>USABILITY</strong></td>
<td>- will the users actually work it successfully?</td>
</tr>
<tr>
<td>+</td>
<td></td>
</tr>
<tr>
<td><strong>LIKEABILITY</strong></td>
<td>- will the users <em>feel</em> it is suitable?</td>
</tr>
<tr>
<td><strong>COST</strong></td>
<td>- what are the capital and running costs?</td>
</tr>
<tr>
<td></td>
<td>- what are the social and organisational consequences?</td>
</tr>
<tr>
<td><strong>ACCEPTABILITY</strong></td>
<td>- on balance the best possible alternative for purchase</td>
</tr>
</tbody>
</table>

Furthermore, figure 5.5 shows that the level of effectiveness is proportional to the interest the users have in buying the Wing.
Based on the results of figures 5.4 and 5.5, the interest of the users in the Wing depends on the success in completing a task on 3D VE and the more success on that task, the more satisfied the users...
are, increasing their desire in getting this product and developing, as a consequence, their interest in acquiring it. To sum up, the Wing has proven to be a more effective input device for 3D VE than the computer mouse. Despite being less efficient than the computer mouse, users were interested in buying the Wing because of its effectiveness. For SV, most engineers require, at a minimum, basic charting and graphing capabilities. Thus, the Wing fits in those environments where charting and graphing capabilities are required, e.g., medical imaging, as doctors can take a look inside human body for clues about a medical condition.

**Proof:** Previously, we could see that the computer mouse responded faster compared with the Wing. From this initial evaluation, the computer mouse proved to be more efficient than the Wing. Furthermore, the users who took longer to complete the task, in their majority from the expert group, felt frustrated and some of them hardly completed the experimental task (P11 and P10).

However, over half of the users (experts and non-experts) expressed their satisfaction after completing the experimental task (see figures 5.6 and 5.7) and most of the comments from both groups show the interest in having the Wing as they “would love to have such a model for gaming” and they believed that “the Wing device is really helpful for 3D visualisation”. This statement in conjunction with the time in completing the experimental tasks explains the linear relation between the variables of figures 5.4 and 5.5. The implication of this information is that effectiveness was more important for the users than gains in speed (efficiency).

---

**Q3. “I was able to effectively complete the required tasks.”**

<table>
<thead>
<tr>
<th>Strongly Disagree</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Pie chart](image)

Figure 5.6: Pie chart where the participants answered the question regarding the effectiveness of the Wing.
Figure 5.7: Pie chart showing that over half of the participants felt satisfied with the ease of completing the required tasks.

Confidence and Engagement Results

An input device is engaging if it is pleasant and satisfying to use. Moreover, confidence is extremely valuable for any usability test. In respect to these previous definitions, the correlation between questions 3 and 9 shows that the more effective users were in completing the task, the more confident they feel. See figure 5.8.

In addition to this, the correlation between questions 4 and 12 displayed on figure 5.9 states that the rating of the Wing depends on the user satisfaction feeling with the ease of completing the required tasks. According to this, the relationship between effectiveness and confidence is established: a user feels confident when they know that using an effective device will help them to control all the actions for their work. As the Wing is an effective input device, users will feel confident to use it because the Wing allows them to complete their 3D task.

Finally, figure 5.10 show that users found the various functions in the Wing well integrated and because of this, they felt confident to complete with their task. Thus, users are engaged to the Wing because its features are appropriate for the demands of their task in 3D VE. With advances in medical imaging over the years, surgical procedures have become increasingly reliant on a range of digital imaging systems for navigation, reference, diagnosis and documentation. Hence, doctors engaging with the Wing may feel confident to have better insight of the exploration they are performing to determine which disease or condition explains a person’s symptoms and signs.

Proof: The Wing received generally positive reviews from participant. Over half of the users confirmed that they felt confident (56% of 18 participants, see figure 5.11) to know that things were under their control as they stated that “it is easier for orbiting in 3D, it make so easy experience/visualisation better,” thus effectiveness had a important role in the engagement of this product.
Figure 5.8: Correlation between the level of effectiveness and confidence by using the Wing.

Figure 5.9: Correlation to show that the rating of the Wing depends on the user satisfaction feeling with the ease of completing the required tasks.

Despite having neutral responses from 5 participants, over half of participants engaged with the Wing. Most of the comments agreed that the Wing “provides more axes of rotation which enables better manipulation of objects in the image.”
Figure 5.10: Correlation between questions 4 and 7 where it is stated that the well-integrated function of the Wing affects positively on the engagement of user with this product.

The Wing did not escape criticism, with reviews pointing to not feeling engaged with the Wing because “it is a cheap, mouse integrated joystick” and expressed their predilection for the computer mouse. In conclusion, the Wing received praised from most the participants as the well-integrated functions of the Wing were the reasons that users felt confident and could engaged with the Wing.

Easiness to Learn Results

One of the biggest objections to usability comes from people who fear how much time it will take to learn to operate a new product. An input device which is easy to learn allows users to master it on their own in a short time.

The correlation between questions 6 and 8 (see figure 5.12) suggests a linear and directly proportional relation between the likeability and the intuitiveness of the Wing. This is translated as the decision of users to choose the Wing for their work or home use not only depends on the effectiveness of the Wing, but also depends on how quickly people learn to use it. Then, users will engage with the Wing if the Wing proves to be efficient and easy to learn.

Also, the correlation between questions 6 and 11 showed in figure 5.13 states that the interest of the users in buying this product is related to the capacity of learning of the user. If users learn how to master the Wing quickly, the probability to buy it is higher. Note figure 5.13 indicated that the data is strongly correlated.

The inversely proportional relation between questions 9 and 10 shown in the correlation graph of figure 5.14 determines that the more confident users were in using the Wing, the fewer things they needed to learn of the Wing. In consequence, the more intuitive the Wing is, the more useful for 3D navigation it is because it is compatible with the tasks as the Wing’s buttons are in the obvious places.
and they, together with the rest of the controls, do the things the user thinks they ought to do.

To conclude, **the Wing is an intuitive input device which doesn’t require expert knowledge**
CHAPTER 5. THE USER EVALUATION STUDY

Figure 5.13: Correlation between question 6 and 11 where it is stated that the more intuitive the Wing is, the probability to buy it is higher.

Figure 5.14: Correlation between the level of confidence and the easiness to learn how to operate the Wing.

or a steep learning curve to operate it. The Wing is appropriate to those environments that employ direct-manipulation user interfaces and simplify the programmer’s task, as the Wing makes volume
visualisation packages more accessible to scientist and engineers. Then, the Wing is an asset for scientist and engineers as it would help them to extend and customise the software for their individual needs as they would be able to learn in short time how to master the Wing.

Proof: Figure 5.15 and 5.16 show the opinion of the users about the Wing as an intuitive input
device. Figure 5.15 shows that over half of the participants agreed that the Wing was fast to learn. Also, users assumed that they would be able to complete the task without assistance pointing to the nature of the task and because the well-integrated functions were self-explanatory.

The user evaluation study reported that 61% of 18 surveyed participants gave the Wing a positive review regarding intuitiveness (see figure fig:PieCurvLearning), whereas 77% of 18 surveyed participants stated that they learned quickly how to use the Wing (see figure fig:Intuitive). Some of the positive comments stated that performing the same task with the mouse was “harder” because “they can’t see” the 3D object, the computer mouse “was not responsive” and they were “unable to manipulate the object with it,” but declared that “the Wing was really intuitive.” Also, some participants “found hard to use the mouse with Drishti” as they “cannot move freely” compared with the Wing.

Criticism was directed to the design of the Wing as any time the participant moved forward “it felt so uncomfortable because of the buttons of the mouse.” According to this “it takes a lot to get familiar with the axes on the screen”. Despite this, they agreed that the participant was more into the details of the image than the device and the Wing was more useful than the mouse, it only takes a time to understand how to use it. Therefore, using this principle it can be shown that, the Wing is not only an effective input device but also an intuitive tool ideal for 3D visualisations.

5.3.3 Utility of the Wing

![Correlation between Q7 and Q12](image)

Figure 5.17: The correlation between questions 7 and 12 shows that the rate of this product also depends on the well integrated functions found in this device.

**Utility** is the ease with which the object can be used. This suggests that whatever the object is there is an advantage in using it. Figure 5.17 shows the linear and directly proportional relation between the rate of this product and the how good participants find the functions of the Wing. The Wing received
positive reviews from participant as some of them stated “the Wing would be better for controlling movements such as the flight of a plane in game the axes could be used as shortcuts within an operating system” and “it could be very beneficial for exploring 3D datasets.”

Additionally, figure 5.18 shows the histogram of the utility of the Wing compared with the computer mouse. The Wing received mostly positive reviews of its utility in 3D visualisation navigation. Over half of the participants agreed that “the 3D rotations and manipulation of 3D objects were better with the Wing”, pointing to the fact that “the Wing is appropriate for 3D visualisations.”

Conversely, some participant from the expert group denounced those same attributes, saying, “the design looks and feels flimsy, the button placement is not optimal and the buttons could be analogue and get some speed control.” Figure 5.18 shows that 24% of 18 participants considered that the “mouse was better than the Wing for 3D visualisations.”

Other reviews were more mixed. The non-expert group consensus reads: “I think that once you got the grasp of it is quite easy to manipulate the device. I’m satisfied with it, but some improvements could be really handy.” Also, they stated that “the wing could be really practical but is not well correlated with the computer, the feedback can be improved, and also there is a discrepancy between the input and output.”

The implications of this information are that, although some functions of the Wing can be corrected and/or adjusted to improve performance, the Wing is appropriate to interact with emerging 3D imaging technologies as it can manipulate 3D objects better than the mouse. The Wing offers new possibilities for imaging tests as the doctor has control over the 3D image to see inside a particular organ, for example.

Proof: Figure 5.19 shows that the Wing received highly positive reviews as over half of the participants (66% of 18) agreed of having found the various functions in the Wing well integrated.
Figure 5.19 shows that 39% of 18 participants, based on their experience in 3D VE, were interested in acquiring this product. Negative response from participants (17% of 18) were caused mainly because of problems with the design of the Wing as “there was an issue with the input and output of the device; the device is lagging at some point, staggers and is difficult to figure out the sensitivity of the device”.

Furthermore, most of the opinions of the participants were biased because the experience they had with the Wing. 24% of 18 participants, the majority being from the expert groups, shown an inclination for the computer mouse and considered it was better than the Wing to perform the task.

In conclusion, critical reaction was generally positive to very positive regarding of the use of the Wing for 3D interfaces as the Wing is “quite easy to use, it was almost natural” and the Wing “manipulates things more easily in 3D versus 2D.” Therefore, using this principle it can be shown that the previous conclusion is valid.

**Figure 5.19: Pie chart of the utility of the Wing.**

### 5.3.4 Cost and Acceptability of the Wing

Figure 5.20 shows that 39% of 18 participants showed interest in buying the Wing based on their own experience in Drishti as they were “happy” of having had a good experience with the Wing. 44% of 18 participants considered that the Wing needs some improvement, but they did not discard the possibility to buy the Wing for a reasonable price.

Figure 5.21 shows the participants’ acceptability, with a median cost for the Wing of £35.00 and a mean of £43.53. Some comments stated that because the Wing is new, they would not pay more than £10.00. But if the grasp is adjusted I would pay even £50.00.”

Furthermore, figure 5.21 shows that over half of the participants were willing to pay up to £35.00 for the Wing; whereas the real cost of the Wing is over £100.00. Only two participants agreed to pay up
Figure 5.20: Pie chart of the interest in the participants to buy the Wing
to £100.00. Criticism was directed to the “difficulty in not clicking buttons when tilting downward/forward.” However, participants suggest that “if they were graphic designers, they would pay £40.00 for
this device.” A participant from the non-expert group stated that “if the device does not slowdown on its reaction, and works quicker I would buy it, let’s say approximately £21.50.”

Commonly, a good computer mouse doesn’t cost a ton of money as it is included at the moment to buy desktop machine. In fact, the price of the mouse is included in the price of the equipment. However, because it has been 30 years since its invention, new generations of mouse are built from cheaper materials and their response on computer have been improved since then [MD96], making their cost almost negligible.

The relevance of this theory to the question is that as the Wing is similar in design to the computer mouse, the participants expected to have the same value in terms of cost. Some participants went on to describe the Wing as a “cheap, mouse integrated joystick,” suggesting to “implement analogue integrating control (like in analogue controllers) to make it worthwhile.” However, the Wing size was praised as “because of its size it would be a novel.” Therefore, the Wing was estimated in a lower cost (from £7.00 to £35.00).

The implication of this information is that Sam Worthington and WSL must consider this study for the new generations of the Wing, as this project suggests that with an adequate calibration system it is possible to build an effective input device as the Wing from cheaper materials, making its price more accessible to people.

5.4 Discussion of the Evaluation Results

The Wing has received critical acclaim and is seen as an excellent product as is shown in figure 5.22. The pie chart shows a preponderance of the participants (39% of 18) rating the Wing as an excellent

---

4 Worthington Sharpe Ltd.
product. Only 17% of the participants, based on their experience with the experimental task, stated that the wing is a poor product.

The experimental and usability tests revealed that the Wing, compared with the computer mouse, is an effective and intuitive input device with well integrated function, appropriate to interact with emerging 3D imaging technologies and able to offer a pleasant experience to novice users. Additionally, the evaluation revealed that the Wing is preferred to be used by people whose background knowledge in computer science is not strong and the Wing fits in 3D VE where charting and graphing capabilities are required, such as SV and medical imaging.

The Wing received mixed reviews from participants. Results from the evaluation showed that most of the participants from both groups agreed that they felt confident in doing the experimental task with the Wing and having a pleasant experience. However, they complained about the design of the device as well as having some difficulties in understanding how to rotate the image in the desired direction. Criticism from 9 of the 18 participants stated that “the buttons’ position were not optimal” although they did not state that this made their experience unpleasant “with the movement freedom the Wing offered by providing extra buttons for zooming and rotation.”

In a positive review, users were generally impressed by the design of the Wing and by the fact that they could feel they were rotating the object with their own hand. One of the users stated that “at first it was quite difficult” when he “had to incline it to the front”, but then it “became quite natural.” Figure 5.23 shows the comments of the participants were classified in 4 groups for its study. 26% of the comments was about the user experience and 15% to about improvements to the Wing. However, figure 5.23 also shows that 22% of the comments were regarding the design of the Wing, whereas 37% were related improvements to do on the Wing. Additionally, some comments stated that the right and left rotation were fine but forward and backward rotation were not, having “a bit unfamiliar feeling” at the moment of grasping the device and rotate the 3D object.

Furthermore, 5 of the improvements that were suggested for the Wing, according to participants, were to change the two primary buttons’ position, to fix the directions of movement, to improve the speed in the execution time of the Wing, to fix the calibration of the Wing and to change the design of the Wing. One of the participants from the non-expert group declared that she had “smalls hands, so it was hard to lean forward without touching the mouse button.” As seen in figure 5.24, suggestions of changing the two primary buttons’ position and fixing the directions of movement were the highest scores.

Moreover, 11% of 18 participants claimed of having had some “lags” when they operated Drishti with the Wing and they considered that “if the speed problem is corrected, they would have a better experience.” These “lags” problems are attributed to the daemon and the version of the software used and not to the Wing. The errors on the daemon were corrected and the new version of Drishti corrected all the bugs presented in this version (https://github.com/AjayLimaye/drishti).

Finally, some participants stated that the Wing was easier to use when compared with “other input devices” as “the Wing is a single device for all tasks presented for 3D rotations.” Unfortunately, the participants did not mention which other devices they were referring to.

To conclude, this chapter provided the details of the user evaluation study used in this project. The performance of the evaluation has favourable results and provides evidence enough to prove if the hypotheses stated in the beginning of this chapter are correct and true. The results are presented as
Figure 5.23: Histogram of the general comments of the Wing were classified in 4 groups for its study.

Figure 5.24: Histogram of the “improvements” needed to do to the Wing, based on the participant comments.

follows:

- \textbf{H1:} The Wing is an effective, efficient and easy to learn input device to manipulate 3D objects in VE. The evaluation results confirm this statement.
• **H2:** In a short time, expert users will master the Wing and will have a pleasant experience at manipulating 3D objects. This pleasant experience will lead them to buy the Wing for their work and/or home use. In particular, expert users felt more comfortable with the computer mouse and some of them expressed their dislike for the Wing. However, they stated that they had a pleasant experience and agreed that the Wing is intuitive and appropriate for 3D visualisations. Also, they recommended to do some improvements on the Wing’s functions to test it again.

• **H3:** Non-expert users will have a pleasant experience at manipulating 3D objects and will be interested in buying the Wing for their work and/or home use. Non-expert users demonstrated master the Wing faster than the expert group and they showed interest in buying the Wing for their personal use. However, they also recommended to do some improvements on the Wing’s functions to improve their user experience.

The next chapter presents the achievements of this thesis, according to the hypotheses presented. Also, an analysis of the strengths and limitations of this project is provided. Finally, it provides the recommendations for future work to be carried on to expand on the objectives of this project.
Chapter 6

Conclusions and Future Work

“On TV, stories and events are finalized in 30 or 60 minutes, or neatly tied up after a season or two. The best stories are the ones that force us to come to our own conclusions and to explain why we believe in our conclusions.” Lurlene McDaniel

In the previous chapter we presented the results of the evaluation of 18 participants manoeuvring the Wing in Drishti, a 3D volume visualisation package for Scientific Visualisation (SV) and medical imaging. The experimental tasks were described and we discussed the evaluation results, developing initial conclusions that we presented according to three hypotheses.

In this chapter we present all the achievements of this thesis, according to the hypotheses presented in chapter 5, as well as an analysis of the strengths and limitations of this project.

The sections for the rest of this chapter, conclusions and future work, are as follows:

- Section 6.1 Conclusions
- Section 6.2 Limitations
- Section 6.3 Future work

6.1 Conclusions

In this project, we evaluated the utility, usability, likeability, cost and acceptability of a combined hybrid isometric, isotonic and viscous 6DoF input device called the Wing and its applications for scientific visualisation (SV) and medical imaging.

Therefore, using these factors it can be shown that, despite being less efficient compared with the computer mouse, the Wing is a more effective and intuitive input device appropriate to interact with 3D volume visualisation packages for SV and medical imaging. It was demonstrated that users will rapidly engage with the Wing because of its well-integrated 3D functions (pitch/roll and yaw) and, after using the Wing, they will have a pleasant experience and they will feel more confident after having completed their tasks in 3D Virtual Environments (VE) by manoeuvring the Wing, compared with the computer mouse.
Additionally, the Wing’s 3D integrated functions are well defined so that there is no need for a steep learning curve or expert knowledge to master them. The results show that over half of the participants felt they were able to effectively complete the required task and, despite some of them having difficulties managing the Wing, they felt satisfied with the ease of completing the required task. Thus, the Wing is particularly easy to use for those with limited knowledge in computer science or in sophisticated and more elaborated input devices. The Wing provides control on a range of digital imaging systems for navigation, reference, diagnosis and documentation.

The design of the Wing has proven to fit and meet the requirements of users of 3D interfaces as it is easy to grasp, allows the users to manipulate the objects at their own convenience and users can do rotation and zooming easily compared with the traditional 2D input devices. However, modifications on the design of the Wing are highly recommended. Despite having advantageous attributes, the primary buttons’ position are not optimal for 3D navigation as most of the users commented that they had difficulties trying not to click on them when they moved forwards/backwards (when they operated the pitch axis).

Furthermore, the stability of the Wing was tested and, based on the results, we created (and then upgraded) an additional auto-calibration daemon to compliment the standard calibration. The results of this research showed that there is little inconsistency in two axes of the Wing (roll and yaw). The function of the daemon is to compensate the noise and non-zero centre values in the Wing, setting a valid range of values for the Wing. Future generations of the Wing should consider these values to improve user experience.

Finally, in terms of cost, we demonstrated that the Wing, with a good auto-calibration program, can be built from cheaper materials for its commercialisation. Nevertheless, if Sam Worthington and WSL

\footnote{Worthington Sharpe Ltd.}

are looking to commercialise this product, it is highly recommended to implement the daemon for the extrema and centre points in C or C++ to be faster compared with Python.

In summary, the Wing functions were evaluated and the results show that, compared with the computer mouse and in terms of effectiveness and intuitiveness, the Wing is a superior product for the exploration and navigation through 3D image data where charting and graphing capabilities are required. The results of the evaluation presented here also suggest that the Wing is ideal for all types of users, from those who are familiar with 3D interfaces to those who aren’t and, ultimately, users can engage with it in a short time.

### 6.2 Limitations

There are some limitations in this project that we split into four sections: the Wing, the Python daemon, the SV software used in this project and the experiment test.

#### 6.2.1 Review of the Wing

In this project, the second generation of the Wing was used and at first it presented sub-optimal results for the Z-axis. Operator and team members tried to use this axis but it was hard to press it down, make it still and this axis behaved akin to an off-set button than a smooth controller. New versions of the Wing
have been developed and this axis has been improved; as a result, new generations of the wing have a more comfortable design for a user’s hand. It is proposed that future work includes the evaluation of this axis in conjunction with the roll, pitch and yaw of the Wing.

### 6.2.2 Review of the Python Daemon

#### Number of steps

In chapter 3, the uncertainty of the Wing was determined by the number of steps from the minima to the maxima extrema points, crossing through zero as was explained in chapter 3. The initial result for the number of steps ($N_{Steps}$) was an adjustment of 123 steps (see figure 6.1) per axis. However, the results of the test were not favourable as the Wing movements were extremely sensitive, making it impossible to control the 3D objects on the screen. Therefore, most of the reviews from participants asked to reduce the level of sensitivity of the Wing as “lower sensitivity when stopping to move” was desirable.

The next approach was to make the margin of error for the centre point wider, reducing the number of steps to 7 (see figure 6.2). However, the resulting value was too big, making the operation of the Wing complicated as it was behaving as an off-set device and not providing smooth rotations.

The last upgrade of the Python daemon fixed these problems with the centre point of the Wing by setting 15 steps for the uncertainty of the Wing to have smoother movements and therefore a better user experience.

![Quantization Noise and Signal-to-Noise on axis X](image)

Figure 6.1: Digital signal processing of input values on axis X of the Wing with $N_{Steps} = 123$ steps.
CHAPTER 6. CONCLUSIONS AND FUTURE WORK

106

Figure 6.2: Digital signal processing of input values on axis X of the Wing with $NSteps = 7$ steps.

Daemon Time

Another limitation found during the evaluation test was the execution time of the daemon. Initial participants felt there was a lag on the Wing because the movements didn’t correspond to the movement of their hands. This problem was caused because there was a slight delay in the time parameters used to send the instructions through the port, explaining the mismatch of the movements in Drishti. Nevertheless, the next command line fixed the problem and it was included in the last upgrade of the daemon:

$$\text{(time.time()} - \text{st1}) > 0.15$$

6.2.3 Review of Drishti

Drishti is one of the most complete and stable products for SV and medical imaging. It has multiple features to control 3D objects and manipulate them for exploration, navigation and documentation. However, the main problem found in this project was the rotation of the objects in Drishti. Participants were confused as they expressed their discomfort of not knowing the orientation of their rotations.

This problem was caused because Drishti works on local spaces, meaning that the coordinate system of the object is from the object’s point of view and not from the user’s point of view.

6.2.4 Review of the Experimental Tests

One of the biggest limitations was that we had a dearth of input devices to compare with the Wing. This project only focussed on the comparison with the computer mouse as it is the most common pointing
device. However, it is desirable for future work to compare the Wing with other input devices in terms of usability, utility, likeability, cost and acceptability.

In addition, another limitation was the nature of the experimental tasks as the tasks presented here were intuitive and easy to complete. It is preferable to test the Wing in real scenarios of SV and/or medical imaging to analyse the result and compare them with utility and usability of current devices used in those areas.

Last but not least, the time given to complete this project was also a limitation. We consider that a year is a short time to analyse all the advantages and disadvantages of the Wing and explore all the possibilities of the Wing and 6DoF in different scenarios and fields.

6.3 Future Work

The second generation of the Wing was evaluated in this project. It is proposed that future work focuses on new generations of the Wing in terms of utility of the Z-axis and its applications, efficiency compared with the computer mouse and other input devices and stability according to their extrema and centre points. It is also worth noting that Sam Worthington and WSL have modified the Z-axis for new generations of the Wing, thus more capabilities and its applications in various areas besides SV and medical imaging in 6DoF can be studied.

Using the Wing allows the user to control 3D objects with only a hand. Future work can focus on the study of the possibility to manage the mouse and the Wing at the same time - one hand to manipulate via roll/pitch/yaw 3D objects and the other to select options from menu bars on the screen or operate a keyboard. This can be useful for interactive exhibits where pushing a button with one hand and operating the Wing in the other hand can enable the communication between the participants and the computer to allow social interaction among visitors, involve physical activity and give the audience flexible goals.

In medical imaging, the Wing may also be beneficial to surgeons, for example, when carrying out some types of biopsies or surgeries; whereas the surgeon with one hand rotates and zooms the scanned object by using the computer mouse with their other hand they can make a medical diagnosis, identify many other conditions or, in case of surgeries, indicate where the incision should be made.

Furthermore, the operation of two Wings at the same time can be of interest in this field. Future work can focus on the analysis of the results of the interaction of users with the operation of two Wings and determine if users are able to manoeuvre them at the same time. This can enable new possibilities for how scientists and engineers interact with the 3D volume visualisation packages and how they can use both their hands to manipulate 3D object. This can be useful for medical imaging using X-rays such as computed tomography (CT) or mammography, as the specialist can rotate the 3D object with one
hand and, with the other hand, do a 3D cropping, having a better interpretive and analytic use of the images and thus, create more accurate diagnosis.

Moreover, future work can explore various applications of the Wing in architecture, planning and environmental design, as the Wing can be useful to make 3D animations and models and its performance can be assessed in 3D computer graphics software such as Reddit, Autocad and/or Autodesk 3D Studio Max. Some participants in the user evaluation study of this project were from these areas and they expressed their interest in using the Wing for their work. They mentioned that the Wing will be an asset for them as they believe they could get more details of their designed images with the Wing as they needed a tool able “to get a perfect match of the image.”

Last but not least, another use of 6DoF with Drishti is the colour selection and direction for new light sources. With 6DoF, the user will be able to rotate the colour palette of Drishti. Consequently, more functions from this visualisation volume software can be stored as a list of operators, replacing the traditional grabbing and picking of 2D devices, improving the user experience in 6DoF.
Bibliography


Sam Worthington. The Wing for UAV control. *Beyond the Mouse. The development of the Wing - A revolutionary 3D input device for control in real and virtual environments.*, September 2014.


Appendix A

Daemon in Python

# BEGINNING OF THE DAEMON
# LIBRARIES
import socket
import time
import pygame
from pygame.locals import *
import os

# VARIABLES USED
st1 = time.time()
i =
rotr = 0
rotp = 0
roty = 0
axisx = 0
axisy = 0
ejez = 0
sigmaX = 0.115637 # Value calculated based on Std. Dev.
sigmaY = 0.108371 # Value calculated based on Std. Dev.
sigmaZ = 0.119617 # Value calculated based on Std. Dev.
sigmas = {'x': sigmaX, 'y': sigmaY, 'z': sigmaZ }
done = False
clock = pygame.time.Clock()
joystick_count = pygame.joystick.get_count()
start = time.time();

# CONNECTING TO THE PORT USED BY DRISHTI
s = socket.socket( socket.AF_INET, socket.SOCK_DGRAM )
host = 'localhost'
port = 7755
s.connect((host, port))
pygame.init()

# FUNCTIONS
def AdjustmentAxes(limit, axis, type_axis):
    varAux = limit
    sigma = sigmas[type_axis]
    varAux = varAux + sigma
    if axis < 0:
        varAux = -1 * varAux
    if axis > 0 and axis <= varAux:
        axis = 0
    elif axis < 0 and axis >= varAux:
        axis = 0
    elif axis == 0:
        axis = 0
    return axis

# FUNCTION TO SAVE VALUES. USED FOR THE CALIBRATION STAGE
def saveAxisValues(x, y, z, v):
    xValues.append(x)
    yValues.append(y)
    zValues.append(z)
    vStpwatch.append(v)

# MAIN FUNCTION
if joystick.count == 0:
    # No joysticks!
    print("Error, no se detecto ningun joystick!")
else:
    # Use joystick #0 and initialize it
    wing_joystick = pygame.joystick.Joystick(0)
    wing_joystick.init()

    if wing_joystick.get_init() == 1: print("Joystick detectado")

while not done:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            done = True
            clock.tick()
        if pygame.time.get_ticks() - last_fps > 1000:
last_fps = pygame.time.get_ticks()

if joystick_count != 0:
    # This gets the position of the axis on
    # the game controller. It returns a number
    # between −1.0 and +1.0
    x_axis_pos = wing_joystick.get_axis(0)
    y_axis_pos = wing_joystick.get_axis(1)
    z_axis_pos = wing_joystick.get_axis(3)

if i == 0:
    axisx = axisx + x_axis_pos
    axisy = axisy + y_axis_pos
    axisz = axisz + z_axis_pos
    axis = { 'x': axisx, 'y': axisy, 'z': axisz }
    i = i + 1

    # SET THE EXTREMA AND THE CENTRE POINTS
    x_axis_pos = AdjustmentAxes(axis["x"], x_axis_pos, "x")
    y_axis_pos = AdjustmentAxes(axis["y"], y_axis_pos, "y")
    z_axis_pos = AdjustmentAxes(axis["z"], z_axis_pos, "z")

    # THE VALUES ARE MULTIPLIED BY 10 TO SPEED UP
    # THE MOVEMENT.
    rotr = rotr + (x_axis_pos * 10)
    rotp = rotp + (y_axis_pos * 10)
    roty = roty + (z_axis_pos * 10)

    # Saving the values in an array and display
    # time.time() − start) / 0.01 = time taken to complete a
    # round trip to a server using TCP in min.
    if (rotr!=0 or rotp != 0 or roty != 0) and
        ((time.time() − start) > 0.15):

        rotAngle = abs(round((rotr + rotp + roty)))
        saveAxisValues(x_axis_pos, y_axis_pos, z_axis_pos,
                       pygame.time.get_ticks())

        # 1 is DEG2RAD(angle): angle * PI / 180.0
        multipacket = 'addrotation %s %s %s %s' %(rotr, rotp, roty, (rotAngle))
        s.send(multipacket.encode('utf-8'))
        rotr = 0
rotp = 0
roty = 0
start = time.time();
clock.tick(60)
pygame.quit()
# END OF THE DAEMON
Appendix B

Evaluation’s Questionnaire and Interview Questions

B.1 Satisfaction Questionnaire

Name: ____________________________________________________________

Please circle the most appropriate selection:

Age Range: 18-24 25-34 35-44 45-54 55-64
Gender: Male Female

What is the highest level of education you have completed?

Instructions:

a) Your first task is to use the WING input device and to rotate the object on the screen.

b) Your second task is to open the volume data set of the mobile phone in DRISHTI and using rotation create an image.

c) Your third task is to

General Information

<table>
<thead>
<tr>
<th>Do you have any experience in using 3D user interfaces (e.g. CAD design, 3D games etc.)</th>
<th>None</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>A lot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do you have any knowledge about 3D Volume Visualisations (e.g. medical, materials science etc.)</td>
<td>None</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>A lot</td>
</tr>
<tr>
<td>Do you have any knowledge or experience of using Drishti or other similar visualisation software?</td>
<td>None</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>A lot</td>
</tr>
</tbody>
</table>

Ease of manipulation.
How easy was it to grasp the WING device?

---------------------------------------------------------------------------------------------
How easy was it to manipulate the WING device to do a rotation or a zooming?

Usability Scale

Please answer the following questions by ticking the correct box:

<table>
<thead>
<tr>
<th>Statement</th>
<th>Strongly D</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Strongly A</th>
</tr>
</thead>
<tbody>
<tr>
<td>I was able to effectively complete the required tasks.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I am satisfied with the ease of completing the required tasks.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I believe I understand better how 3D Volume Visualisation works.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I would you like to use this input device for my future work and home use.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I found the various functions in this input device were well integrated.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I would imagine that most people would learn to use this input device quickly.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I felt confident when using this input device</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I needed to learn a lot of things before I could get going with this input device</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Please share with us three things the WING could do better compared to the common mouse?

Please answer the following questions by ticking the correct box:

<table>
<thead>
<tr>
<th>Statement</th>
<th>Strongly D</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Strongly A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Based on your experience with WING, if this was available to buy now, would you buy one?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Please answer the following questions by ticking the correct box:

<table>
<thead>
<tr>
<th>Statement</th>
<th>Poor</th>
<th>Fair</th>
<th>Neutral</th>
<th>Good</th>
<th>Excellent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall how would you rate WING?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

How much would you be prepared to pay for this device (in pounds)?

Comments:

---

---

---

---
APPENDIX B. EVALUATION’S QUESTIONNAIRE AND INTERVIEW QUESTIONS

B.2 Interview Questions

Ask the participants to register themselves on the attendance lists

Participants arrive
"Hello everyone, please be seated"

Wait until the time limit for registration. Let the participants to enter to the waiting room. Make groups of four participants and ask them to enter to the lab. Once inside the lab and while the participant sit down

"Welcome. You are invited to take part in a research project to gain information about an evaluation tool for 3D volume visualisation. The purpose of this research is to evaluate a 6 degree of freedom input device called WING with an image displayed on a software program called DRISHTI and to examine the effects on the users."

"Your participation will involve one session which will last approximately 5 minutes, during which you will be asked to rotate an image by using the 6DoF input device and a computer mouse. If, at any time during the test you would like to stop, you may do so by raising your hand and notifying this to me."

"As part of this test, I am going to share a brief introduction and a discovery session to familiarize you with the software and hardware for the test. Degree of Freedom (commonly abbreviated as DoF) refers to the movement of a rigid body inside any space. It could be explained as the different basic ways in which an object can move in a particular space by means of combinations of computer programs and input devices."

"Those spaces are defined in terms of computer graphics models which can be conceived as drawing pictures on computers, also called rendering. The pictures can be photographs, drawings, movies, or simulations (pictures of things which do not yet exist and maybe could never exist)."

"For this experiment, we are going to use the image of a cell phone or mobile phone device which will be render in a Scientific Visualisation Volume Software called DRISHTI. DRISHTI was written for visualizing tomography data and has been used for a number of purposes, such as volumetric visualisation of various computer tomography datasets."

Showing the WING
"The input device to evaluate is WING. The WING is a 6DoF input device with a similar form to a computer mouse. With this device, the user is able to move the camera in the X, Y or Z directions by gently pushing the controller left and right, forward and backward or up and down. It also allows the user to rotate the camera around the X, Y or Z axes by gently tilting or twisting the controller."

"You will be call in the order of your arrival. Please wait here until I mention your name. I appreciate your time and patience."

Call the first participant. Let him/her enter the room. Participant sits down face to you. Past out the consent form and participants’ copy
"Before continue, please read this form and if you agree, sign it. If you have any question, please let me know. If, at any time during the test you would like to stop, you may do so by raising your hand and
If once past out the consent form, the participant refuses to sign the form
"I understand and I appreciate you time and I respect your decision. May I ask you the main reasons for refusing signing this consent form?"

This discussion should be documented. Gently, walk the participant out the lab and lead him/her to the way out

Once having the consent form signed, gently lead the participant to the experimental area. The participant sits down
"I am now going to distribute the instructions of the experimental tasks and a questionnaire to be answered after the test. Please answer the questions as accurately as possible. If you have any question or any technical difficulties, please let me know by raising your hand."

Past out the questionnaire and the pens
"There is currently no enforced time limit for the questionnaire. However, it is estimated to complete the tasks in less than 5 minutes. I will also give you the pen that you need to write with. You may now turn over your sheet and begin reading the instructions."

Time 5 min
"Please complete the questionnaire here and wait until the last member of your room complete the experiment."

Wait for participant to complete their responses and wait for participants to finish their questionnaire

After all data is collect and the experiment is over

Proceed with the interview questions
"Thank you for participating in this experiment. I have some finals questions for you."

Interview Questions
Tasks: How did the Wing device perform compared to the computer mouse from a scale of 1 to 5 where 1 is Miserable and 5 is Delightful?

<table>
<thead>
<tr>
<th>Overall quality</th>
<th>Miserable</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Delightful</th>
</tr>
</thead>
<tbody>
<tr>
<td>Look and feel</td>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>Delightful</td>
</tr>
<tr>
<td>Initial experience</td>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>Delightful</td>
</tr>
<tr>
<td>Final experience</td>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>Delightful</td>
</tr>
</tbody>
</table>

1. Did you enjoy using this system?
2. What do you think should be different added or taken away from this system?

3. Any other comments?

"Do you have any questions?"

Answer any questions

"Thank you for your help. If interested, you can find out the result of the study by contacting me, the researcher Mario Sandoval, after date October, 2015. I can be contacted in room 2.94 at Kilburn building, University of Manchester."
## Appendix C

### Evaluation Data

<table>
<thead>
<tr>
<th>Codes</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td></td>
</tr>
<tr>
<td>P2</td>
<td></td>
</tr>
<tr>
<td>P3 R</td>
<td>3D rotations, single device for all tasks presented</td>
</tr>
<tr>
<td>P4</td>
<td>No</td>
</tr>
<tr>
<td>P5 R</td>
<td>The rotation thing, easy to grip</td>
</tr>
<tr>
<td>P6</td>
<td></td>
</tr>
<tr>
<td>P7 I</td>
<td>It could be a bit more sensitive</td>
</tr>
<tr>
<td>P8 I</td>
<td>Improve the lag and adjust the input commands. Be a little more user friendly. More precise to have good grasp of commands.</td>
</tr>
<tr>
<td>P9 I</td>
<td>Lower sensitivity when stopping to move</td>
</tr>
<tr>
<td>P10 R V U</td>
<td>It is easier for orbiting in 3D. It make so easy experience/visualization better.</td>
</tr>
<tr>
<td>P11 R M</td>
<td>It provides more axes of rotation which enables better manipulation of objects in the image.</td>
</tr>
<tr>
<td>P12 V M</td>
<td>It works better with 3D interfaces, better for designing. Could be used in the airport security for better scanning.</td>
</tr>
<tr>
<td>P13 M</td>
<td>Manipulates things more easily in 3D versus 2D</td>
</tr>
<tr>
<td>P14 I</td>
<td>1) Current mouse buttons and Wing arragement give the feeling of colliding; 2) Better if the Wing device was fixed to a table (surface); 3) Confusing while rotation the angles (euler angles) so once rotated the object it’s difficult to keep track of the axis to continue rotating the object.</td>
</tr>
<tr>
<td>P15 I U</td>
<td>I think the wing could be really practical but is not well correlated with the computer, the feedback can be improved, also there is a discrepancy between the input and output</td>
</tr>
<tr>
<td>P16 R M</td>
<td>Rotation, ease of managing pitch/roll/yaw</td>
</tr>
<tr>
<td>P17 R V U</td>
<td>Show full features of items; Allow free movement; Provide better idea of real dimension</td>
</tr>
<tr>
<td>P18 V M R I</td>
<td>Wing would be better: for controlling movements such as the flight of a plane in game the axes could be used as shortcuts within an operatin system (this would be cool?) it could be very beneficial for exploring 3D datasets</td>
</tr>
</tbody>
</table>

Table C.1: Answer to the question of three things the Wing can do better compared to the common mouse.

**Note:** the theme code used in this table is described as follows: R for “The 3D rotations was better”, V for “The Wing is appropriate for 3D visualisations”, I for “Mouse is better than the Wing”, M for “Manipulation of 3D objects was better” and U for “I have a good experience with the Wing.”
<table>
<thead>
<tr>
<th>Codes</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>FW</td>
</tr>
<tr>
<td>P2</td>
<td>IW DW</td>
</tr>
<tr>
<td>P3</td>
<td>IW</td>
</tr>
<tr>
<td>P4</td>
<td>IW</td>
</tr>
<tr>
<td>P5</td>
<td>IW</td>
</tr>
<tr>
<td>P6</td>
<td>UE</td>
</tr>
<tr>
<td>P7</td>
<td>UE</td>
</tr>
<tr>
<td>P8</td>
<td>UE</td>
</tr>
<tr>
<td>P9</td>
<td>FW UE</td>
</tr>
<tr>
<td>P10</td>
<td>FW IW DW</td>
</tr>
<tr>
<td>P11</td>
<td>IW UE</td>
</tr>
<tr>
<td>P12</td>
<td>IW DW</td>
</tr>
<tr>
<td>P13</td>
<td>IW FW</td>
</tr>
<tr>
<td>P14</td>
<td>UE DW</td>
</tr>
<tr>
<td>P15</td>
<td>UE IW DW</td>
</tr>
<tr>
<td>P16</td>
<td>UE IW DW</td>
</tr>
</tbody>
</table>

Table C.2: General comments of the participants about the Wing.

*Note: the theme code used in this table is described as follows: UE for “User Experience”, DW for “Design of the Wing”, FW for “Possible scenarios for the Wing” and IW for “Improvements to do on the Wing.”*
### Table C.3: Comments of the improvements to do on the Wing.

<table>
<thead>
<tr>
<th>Codes</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>P2</td>
</tr>
<tr>
<td>B S D</td>
<td>The design looks and feels flimsy. Button placement is not optimal. Buttons could be analog and get some.</td>
</tr>
<tr>
<td>P4</td>
<td>P5</td>
</tr>
<tr>
<td></td>
<td>S M</td>
</tr>
<tr>
<td>P7</td>
<td>P8</td>
</tr>
<tr>
<td></td>
<td>P10</td>
</tr>
<tr>
<td>M B C</td>
<td>The Wing device is really helpful for 3d visualization. Some adjustments can be made to the calibration so it reacts as similarly as possible to the hands movements. Maybe shift the clicking button on the Wing device lower.</td>
</tr>
<tr>
<td>P12</td>
<td>P13</td>
</tr>
<tr>
<td>D B</td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>Difficult not to click buttons, when tilting downward/forward. If I were a graphic designer I would pay £40.00 for this.</td>
</tr>
<tr>
<td></td>
<td>P15</td>
</tr>
<tr>
<td>M</td>
<td>The design of the device is really neat and user friendly. But there is an issue with the input and output of the device; the device is lagging at some point, staggers and is difficult to figure out the sensitivity of the device</td>
</tr>
<tr>
<td></td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>It is a cheap, mouse integrated joystick. Implement analog integrating control (like in analog controllers) to make it worthwhile. Then, because of its size it would be a novel.</td>
</tr>
<tr>
<td>M</td>
<td>P18</td>
</tr>
<tr>
<td></td>
<td>Using gest the axes in solation without making the mouse was challenging (but I do not know how common this requirement would be). Biggest challenge here was using local axis y model which is controlled by using V.S. the axis relative to the screen.</td>
</tr>
</tbody>
</table>

**Note:** The theme code used in this table is described as follows: B for “Button placement is not optimal”, S for “Speed”, C for “Calibration”, M for “Movement direction is not according the wrist movement” and D for “It’s not the appropriate design.”
### Table C.4: Answers to open questions about the Wing.

<table>
<thead>
<tr>
<th>Question</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
<th>P6</th>
<th>P7</th>
<th>P8</th>
<th>P9</th>
<th>P10</th>
<th>P11</th>
<th>P12</th>
<th>P13</th>
<th>P14</th>
<th>P15</th>
<th>P16</th>
<th>P17</th>
<th>P18</th>
</tr>
</thead>
<tbody>
<tr>
<td>How easy was to grasp the Wing?</td>
<td>Easy</td>
<td>Easy</td>
<td>Fairly easy</td>
<td>A bit unfamiliar</td>
<td>Quite easy</td>
<td>At first it was quite difficult when I had to incline it to the front, but then it became quite easy.</td>
<td>It was easy, since I had the movement freedom the device offers, by providing extra buttons for the right and left rotation is fine but forward and backward rotation is not.</td>
<td>Quite easy, it was almost natural</td>
<td>A bit difficult at start, I think that once you got the grasp of it is quite easy to manipulate the device. I'm satisfied with it, but some improvements could be really handy.</td>
<td>First it was kind of hard to get a feeling for it, but after some time I figured it out.</td>
<td>Yes</td>
<td>Very easy but sometimes not in the planned direction.</td>
<td>Yes</td>
<td>It takes a lot of time to get familiar with the axes on the screen. The Wing device shows more control over the object. I would like to use it in Redit.</td>
<td>Yes</td>
<td>I found hard to use the mouse with Drishti. I cannot move freely.</td>
<td>Yes</td>
<td>Using the mouse it's hard because I can’t see the object and it’s not responsive.</td>
</tr>
</tbody>
</table>
APPENDIX C. EVALUATION DATA

Figure C.1: Pie chart of the level of experience of the participants of the user evaluation study.

Figure C.2: Pie chart of the level of knowledge of the participants of the user evaluation study.