The University of Birmingham

Human Factors Implications of the use of Technical Aids During Real and Virtual Search Tasks

by

Robert Guest

A thesis submitted to
The University of Birmingham
For the degree of
DOCTOR OF PHILOSOPHY (Ph.D.)

School of Engineering
Department of Electronic, Electrical & Computer Engineering
The University of Birmingham

October 2011

The University of Birmingham

Abstract

Department of Electronic, Electrical and Computer Engineering

Doctor of Philosophy

By Robert Guest

With the advent of robust remote communication technologies, the potential for the remote operation of Unmanned Vehicles (UVs) has increased. Recent examples of Remotely Operated Vehicles (ROVs) include underwater port security, police forensic investigations, airborne drones in warfare situations and robots for the disposal, or “render-safe”, of explosive devices. In particular, UVs are often used in situations where there is search activity in a hostile environment.

UV operation is a prime candidate for training by simulation, since, with present-day software and hardware technologies, the virtual representations of remote stimulus can be presented to the user in a manner that is almost identical to the real situation (through video feeds and interactive controls). In the present study, we investigate the simulation of UV operation, especially related to search tasks, using low-cost Commercial Off The Shelf (COTS) development tools, so-called “serious games”. The work is broken into two parts, which are then brought together in a final over-arching investigation.

The first area of investigation is specific to the search task. In particular we investigate whether the cognitive processes of search, recall and spatial awareness of the user are the same when “technical aids”, such as problem-specific equipment readings, are available and when they are absent. The presence of technical aids is an important aspect of UV simulation, since nearly all examples of current systems, both in the field and experimental versions of real world systems feature some kind of non-visual data. The findings show that there is no significant difference in a person's ability to recall location or environmental features when performing a task in the real or virtual world. This would suggest that simulated search can be as effective as physical search.

The second area of investigation evaluates whether simulation with "serious games" techniques is mature enough to effectively represent tasks typically carried out with a UV. This thesis focuses particularly on comparisons between control of an underwater UV/ROV, in simulation and in reality. The impact of fidelity on the usefulness of simulation and whether a user’s prior familiarity with a games environment influences their performance in a simulation scenario is also investigated. There was a clear positive significant difference in performance for those participants that had received virtual training, this validated the simulation as having the appropriate psychological fidelity required to make training effective. However, the participants' previous game playing experience did not prove a significant factor in their performance.

Finally, the two main areas of study are linked by simulating ROV search tasks with technical aids. The impact of high fidelity simulation on the dependency of technical aids is investigated. In particular, the increase in dependency on display characteristics is considered; when the fidelity is increased and whether or not the use of different technical aids affects the search strategy employed. The findings suggest a significant increase on the dependency of using additional technical aids when the simulation fidelity was increased. There is also significant evidence to show that the increased fidelity did affect the search strategy employed.

Publications

Part of the initial research into unmanned vehicles and user interfaces was published as conference proceeding at a NATO workshop in 2006.

> *Serious Gaming Technologies Support Human Factors Investigations of Advanced Interfaces for Semi-Autonomous Vehicles.* Stone, R. J., Guest, R., Ch'ng E., McCririe, C., Collis, C., Mannur, R., & Rehmi, I. (2006). In Proceedings of Virtual Media for Military Applications; NATO RTA HFM-136 Workshop; US Military Academy; West Point, NY; 13-15 June.

The development of "Virtual Scylla" simulation has been published in (Stone *et al.* 2009) and has formed the basis of the "Virtual Scylla" heritage project conducted in collaboration with the National Marine Aquarium.

> *The Virtual Scylla: an exploration of "serious games", artificial life and simulation complexity (2009),* Stone R. J., White D., Guest R. and Francis B., Virtual Reality Volume 13, Number 1, 13-25.

Acknowledgements

I could not have completed thesis without the assistance of many people who I have consulted and deliberated with.

Firstly, I would like to thank my supervisor Professor Robert J Stone. Who has shown constant support and guidance throughout this long process. Secondly, I would like to thank my mother, Penelope Cox, for being willing to stay up into the early hours of the morning proof reading.

I would like to express my thanks to several members of the Electronic and Electrical Engineering Department, particularly, Dr James Knight, who assisted in the statistical analysis and Professor Chris Baber, who helped with experimental design. I would also like to thank my close friends Dr Greg Reynolds for his assistance in mathematical modelling and Dr Sashah Eftekhari for assistance in performing experiments.

Finally, I would to thank all the staff National Marine Aquarium (particularly Deborah Snelling) for their assistance in performing ROV trials and for the use of the ExplorOcean Exhibit.

Table of Contents

Chapter One

Chapter Two

Chapter Three

Chapter Four

Chapter Five

Chapter Six

Chapter Seven

Chapter Eight

Chapter Nine

Chapter Ten

Chapter Eleven

References

Appendices

List of Illustrations

Chapter Two

Chapter Three

Chapter Four

Chapter Five

Chapter Six

Chapter Seven

Chapter Eight

Chapter Nine

Chapter Ten

Chapter Eleven

List of Tables

Chapter Three

Chapter Five

Chapter Six

Chapter Seven

Chapter Eight

Chapter Ten

Chapter One

This chapter outlines the initial problem statement and defines the overall scope of the thesis. It presents the aims and objectives of the work as well as stating the contributions made.

1.1 Introduction

The concept of virtual reality was presented by Ivan Sutherland in 1965; his goal was to make a virtual world "look real, sound real, feel real, and respond realistically to the viewer's actions" (Sutherland 1965). Research into virtual reality and synthetic environments has shown that for training purposes the accuracy of the representation in terms of 'physical' and 'psychological' fidelity effects the performance of the user and what skills are adequately trained (Alexander *et al.* 2005; Waller, Hunt, and Knapp 1998; Kozlowski *et al.* 2004).

A simulation's physical fidelity describes the degree to which objects within the virtual world operate and appear compared to their real world counterparts. Psychological fidelity deals with the accuracy of the behaviours reproduced when performing simulated tasks (Miller and Research 1954). In terms of training, it is the psychological fidelity which is typically associated with positive transfer of skills than that of a simulation's physical fidelity.

The visual fidelity of a simulation, a subset of physical fidelity, can be defined as the visual accuracy of the reproduction from the original (Winkler 2001) or the degree to which visual features in the Virtual Environment (VE) conform to visual features in the real world (Hendrix and Barfield 1995). Both of these definitions are, however, difficult to measure quantitatively.

Assessing the appropriate level of fidelity required for any given training simulation has been an important stage in the development process to ensure the transfer of the required skills, however there is no universal method for defining this (Tsang and Vidulich 2002).

Task performance has traditionally been used to establish the evaluation metric for assessing the required level of fidelity for any given virtual simulation. Comparing a participant's performance in the real world to that of a virtual one provides an indicator as to the level of fidelity that is needed to successfully train for the particular task (Lok *et al.* 2003; Ferwerda 2003).

One of the most common methods for assessing the effect of fidelity within a virtual environment is to ask participants to perform a simple "search and recall" task which addresses their ability to locate and identify targets. It has been shown that, during such a task, individuals construct a mental representation of the space they are searching (Moray *et al.* 1976). It has generally been accepted that any deficits within the simulated world, particularly in the depiction of spatial information, will lead to a poorer mental representation of the scene (Ruddle, Payne, and Jones 1997). This in turn can affect the ability of participants to perform recall tasks (Dinh *et al.* 1999).

Research into human search processes has also revealed two other important factors. Firstly, an item's distinctiveness when compared to the surrounding environment is a greater factor in search duration than the number of items in the search array (Bacon and Egeth 1991). Secondly, it has been found that accuracy of recall falls as the total amount of information displayed increases (Teichner and Krebs 1972).

The amount of information displayed (or environment complexity) has also been shown to be a significant factor affecting the search strategy employed (Crundall, Underwood, and Chapman 1998). With increasing complexity, recall becomes more focused on an object's location rather than the object's identity (Beck, Peterson, and Vomela 2006). This would imply a strong relationship between object location and recall, which suggests that a more accurate understanding of spatial information has a profound effect on recall performance.

Most research addressing the comparison between recall performance in both the real and simulated environments is typically based on a simple search task. These tasks tend to involve a participant, usually in a fixed position, searching a static virtual environment and memorising the target objects and their location. The target objects are usually not obscured and can been seen with the naked eye without requiring any interaction with the environment (Pausch, Proffitt, and Williams 1997). Whilst this method of passive searching has been established as a good metric to test visual fidelity, immersion and engagement, (Robertson, Czerwinski, and Van Dantzich 1997), it is not realistic to the act of search in the real world.

Search tasks in the real world, as one might witness in support of bomb detection, deep-sea surveying and medical/surgical examinations, require a more active approach; users must physically move around a local or remote environment and engage with it using tools or sensor information to uncover hidden targets. Much of modern search and surveillance is now conducted by unmanned vehicles which has become a growing area for search training (Walton *et al.* 1993). Unlike manned vehicles, there is no window or canopy to look through and users are limited to video screens that typically suffer from distortions and degraded imagery due to environmental factors (Calhoun *et al.* 2005).

Image degradation can be described as the reduction of the inherent optimum potential of individual sensor systems caused by error in sensor operations, processing procedures or data transfer errors (Wang and Bovik 2002). It is also often the case that a reduction in quality is caused by environmental factors (i.e. atmospherics, snow, dust, etc.). In unmanned vehicles image degradation is present in the transmitted video and environmental factors reduce or partly obscure the images presented to the user.

This is of particular importance to remotely operated submersibles (ROVs – V for vehicle) where the pilot must contend with extremely limited views of the environment caused by particulate matter, "fish-eye" lens distortions from the remote observation dome and back scattering lighting effects (Li *et al.* 1997). Because of the often limited views, ROV pilots are presented with numerous technical aids such as Global Positioning Systems (GPS), sonar and other sensors, information from which are used for navigation and search tasks. It may be reasonable to conclude that the use of additional technical aids would increase during a search task if the incoming video signals were significantly degraded. However, there is currently a lack of research in this area and it is unclear whether a participant's behaviour, performance or approach to search will differ as the visual fidelity of the virtual environment varies.

The lack of research may be due to three factors. Firstly, high fidelity simulations of real world environments are no easy task. Only recently have the necessary hardware and software development tools become readily available. Secondly, much of the data regarding Unmanned Air Vehicles (UAVs) and ROVs in terms of equipment usage, interface design and simulated training, is based on the assumption that it is the same as flight simulation (Dinadis and Vicente 1999; Zhang *et al.* 2002; Wilson 2002) Very few studies have been conducted that relate directly to UAV and ROVs. Finally, data regarding military search may often be restricted or be of a highly sensitive nature. Due to the above factors, there is currently little evidence to show the relationship between real and virtual search when using additional technical aids. As a consequence of this, it is also unknown what the effect of the simulation's visual fidelity is and how it alters the participant's behaviour when multiple information resources are available.

It is argued within this thesis that simple search tasks fail to address the issues of "real-world" search and that task performance alone may not be the most suitable metric for simulation fidelity. This is particularly appropriate to tasks where the user is being presented with multiple information sources from additional technical aids. It is proposed that the level of fidelity in such tasks may not only affect a participant's performance but also the approach and cognitive strategy they employ.

1.2 **Aims and Objectives**

As discussed above, even though significant studies have been performed on both real and virtual search tasks; little work has been carried out on the impact on search performance when using technical aids. In addition, there appears to be a significant gap in knowledge with regards to remote searching using unmanned vehicles and how onboard technical aids are used. Therefore, it is the aim of this thesis to demonstrate that suitably accurate virtual simulations, created using low cost off-the-shelf development software, can be used to investigate the human factors aspects of technical aid-based search tasks. To this end, a number of specific objectives are now defined.

The first objective is to evaluate current real-time 3D development tools to assess their suitability for development of unmanned vehicle simulation. Each development tool will be chosen based on set criteria, such as the tool's ability to import 3D assets, ease of use, visual quality and user control.

The second objective is to examine whether results from previous real-world studies of the mental search processes of memory and recall, when using technical aids such as metal detectors, still holds true in a virtual simulation of the task.

The third objective is to demonstrate that a suitably accurate virtual simulation of a remotely operated vehicle can be developed, again using low cost off-the-shelf development software. Its ability to simulate the visual and control characteristics of its real world counterpart accurately will be evaluated through a detailed skills transfer study. Sub-objectives of this are to determine the effects of the level of the fidelity of the simulation and to investigate how a participant's previous gaming experience can affect performance.

The fourth and final objective is to develop a virtual simulation of a typical real-world ROV search task as a test-bed to study the use of onboard technical aids. It is hypothesised that increasing the visual fidelity, such as realistic fogging and particulate matter, will increase the user's dependency on technical aids. Consideration will also be given as to whether different search strategies are employed depending on the technical aids available.

1.3 Thesis Organisation and Contributions

Broadly, this thesis is divided into four parts. The first part, chapter two and chapter three, reviews the field of virtual reality and compares it to the emerging field of "serious games" in the development of modern simulations. These chapters will also review current real-time development tools and evaluate them on their suitability for developing remotely operated vehicle search task simulations. Chapter three was, in part, the basis for a serious games engine evaluation commissioned by the Defence Technology Centre (DTC) and elements of the unmanned vehicle research from chapter two were published (Stone *et al.* 2006).

The second part, chapter four, builds on the work presented in a recent study that investigates the cognitive processes, such as memory and recall, involved when performing simplified search tasks using technical aids (Houghton, Baber, and Knight 2009). The study concludes that separate cognitive processes are evident in search tasks for recognising the location of targets and for recalling their surrounding features. Chapter four extends this work by investigating whether the same conclusions can be drawn when the search task is performed within a virtual simulation. The results of this can be seen in chapter five.

The third part, comprising chapters six and seven, describes the development of a virtual simulation of a remotely operated submersible using an appropriate simulation engine (based on the findings in chapter three). The simulation is evaluated though a detailed skills transfer study performed at the National Marine Aquarium (NMA) in Plymouth. This evaluation is required to assess the simulators suitability to becoming a test bed for evaluating the use of technical aids during ROV search tasks.

The fourth part, comprising chapters eight, nine and ten, combines the findings of the previous work to develop a virtual simulation of a more typical real-world ROV search task. As well as the virtual ROV video feed, the user will also be presented with various on-screen technical aids, such as sonar information, current depth and distance to target. The aim of this chapter is to investigate the use of technical aids when searching a realistic environment using ROVs. The development of this simulation has been published in (Stone *et al.* 2009) and has formed the basis of the "Virtual Scylla" project, again conducted in collaboration with the National Marine Aquarium.

Chapter Two

This chapter defines the differences between traditional virtual reality and “serious games”, as well as reviewing the effects of search and recall in previous studies. It also reviews the current research in the area of remotely operated vehicles and current limitations of commercial systems.

2.1 History of Simulation

Simulation is the embodiment of the principle of "learning by doing" (Garris, Ahlers, and Driskell 2002). From an early age children simulate the world around them, and their interactions with other objects, using simple toys and figures (Piaget 1973). Today, 3D computer simulation can be thought of as an digital equivalent of this form of play, where virtual avatars replace plastic toy soldiers.

When speaking about 3D computer simulations, many people cannot help but conjure up images of bulky headsets and instrumented gloves. These images originate from the technology driven heyday of "virtual reality" in the late 1980s and '90s. However, the roots for computer simulation lie much deeper than that. During the 1960s, Ivan Sutherland began theorising his concept for an "Ultimate Display" (Sutherland 1965) and by 1966 he had begun conducting trials of a new head-mounted display that could deliver strong cues to depth and distance through simple stereoscopic images produced from a computer simulation. The years that followed saw the military, in particular the United States Air Force, becoming a driving force behind the technology that by late 1970s was beginning to be used for combat flight training. Whilst the military was interested in the benefits of the interactivity of such simulations, the entertainment industry was beginning to push the boundary of the visual imagery to satisfy the needs of an increasingly demanding audience. Films like *Star Wars* and *Tron* began to show the power of computer generated imagery (CGI). By the early 1980s computer simulations were also beginning to prove themselves in the aviation industry. During the following years computer usage was becoming more and more widespread among the masses and no longer required hundreds of square feet of air conditioned office space to service the racks and racks of analogue and early digital platforms. In 1988, Jaron Lanier coined the term "virtual reality" (Lanier and Heilbrun 1988) to describe the "immersive" experience made possible by the use of the latest generation of helmet-mounted stereoscopic goggles, gloves, body tracking systems and related technologies. There was a general feeling that VR could be used as a universal solution to training and a thriving community of virtual reality companies began to emerge to sell this great new era of computing (Lanier 1992). Towards the end of the 1990s, however, *real* reality began to sink in. It was becoming apparent that VR had become over-hyped in a technology-driven market. Developers of VR training rarely considered the learning outcomes of their systems, as they continuously became caught up in pushing the ever-advancing technology. Many VR systems were very expensive and the community provided few universal standards for interfacing and coding, let alone strong human factors evidence that the technology actually delivered real benefits in training, design and scientific visualisation, to mention but three examples. Researchers and developers began to learn that technology itself was not sufficient to immerse the user, but rather the quality of the content of the simulation or virtual environment that the technology was trying to deliver (Ryan 1999; Stone 2003) .

In the late 1990s, computer games were becoming big business and their visual fidelity was now easily surpassing that of traditional VR systems. It did not take long for developers to realise that the same technology could be used to deliver the interactive 3D benefits that VR failed to achieve.

2.2 Serious Games

"Serious games" is a term which has been adopted in recent years to describe games with educational benefits (Blackman 2005). A growing number of buzz words have been further used to attempt to categorise different styles of education gaming, such as 'edutainment' (adopted from the VR era of the '90s), 'game-based learning' and 'immersive learning environments' (Sawyer and Smith 2008). In truth, serious gaming is not a new phenomenon, and, in fact, computer games have been exploited for training applications ever since their emergence in the early 1980s. However, the one major change in the last few years is the development of the tools necessary for users to create their own content. This allows for popular entertainment games to be altered by a wide range of end users (or "modders"), incorporating training and educational elements at almost no cost.

"Battlezone", developed in 1980 by Atari, is often considered to be one of the first serious games. "Battlezone" allowed players to drive a virtual tank and engage wireframe adversaries and this military theme began to open up the possibilities of using such a system for training. Despite this early interest, it was not until recently that off-the-shelf computers games were used to develop serious training applications. In 2002 a serious game called "America's Army" was developed using the "Unreal" game engine. It was not designed to train any particular task but was heavily used to improve the US Army's public relations and to assist in recruitment (Shaffer *et al.* 2005). Since then the term "serious games" has grown in use, there being today an estimated global market in excess of 20 million pounds per year (Van Eck 2006).

The distinction between *objective* representation and *subjective* representation is made clear by a consideration of the differences between simulations and games. A simulation is a serious attempt to accurately represent a real phenomenon in another, more malleable, form. A game is an artistically simplified representation of a phenomenon (Garris, Ahlers, and Driskell 2002). Games do not (in the main) aim to reproduce an accurate representation of the world but a stylised version of it. The game designer may omit important aspects in order to favour a more visually appealing world. The simulation designer, on the other hand, has to focus on accuracy to allow for exacting computation recreation. The fundamental difference between the two is that games are developed to be visually appealing where simulations focus on detail and accuracy. The same could be said when comparing a painting of a building to its blueprints - both are images of the building but each exists for different purposes.

Whilst simulation and games have different goals in terms of visual fidelity, modern 3D editing and modification tools allow for games engines to produce visual results more akin to accuracy rather than style and allow for the incorporation of serious training objectives.

2.3 Memory and Recall

In recent years one of the most common areas of serious game research has focused on basic cognitive processes of the mind within the virtual world. It has generally been accepted that many of the cognitive functions such as attention, executive functions, memory, language and spatial abilities are processed in the same way when the user is within a virtual environment (Schultheis, Himelstein, and Rizzo 2002; Matheis *et al.* 2007). This link has allowed for an increased use of serious games and virtual environments to treat numerous mental disorders (Rose, Brooks, and Rizzo 2005; Brooks *et al.* 2004; Rizzo and Buckwalter 1997; Ring 1998).

To begin investigating both real and virtual search tasks it is necessary to understand the fundamentals of how the mind stores spatial information and how information is recalled from human memory. Human memory can be described as a system of storing information provided by the five senses for later recall (Baddeley 1997). Plato thought that memories were based on images that had been carved into the mind like pictures carved into wax. In more recent years, research has continued into the importance of mental imagery and memory.

In 1974, Baddeley & Hitch proposed a model for working memory which helped to describe the cognitive processing in short term memory (Baddeley and Hitch 1974). Their model was composed of four components:

- Central executive - responsible for the control and regulation of cognitive processes.
- Phonological loop - deals with sound or phonological information.
- Episodic buffer - dedicated to linking information across domains to form integrated units of visual, spatial, and verbal information with time sequencing.
- Visuospatial sketchpad - It is used in the temporary storage and manipulation of spatial and visual information, such as remembering shapes and colours.

It is this mental sketchpad which would be utilised during search tasks. It was proposed by Baddeley & Hitch that this visuospatial sketchpad would temporarily hold information about what we see. This information may contain shapes and colours of objects as well as one's movement through space. It is also involved in tasks which involve *planning* of spatial movements, such as planning one's way through a complex building.

Imagine being asked to count the number of lights in one's house. In this situation, the human mind would typically build a mental image of the house and all of its rooms. One would travel from room to room within the mental image of the house, scanning for each light and adding them up. Mental imagery is used when there is an absence of an appropriate sensory input. It is used to store information about objects such as, colour, size, shape, texture and spatial location within a scene. Many studies have concluded that using mental images as a mnemonic device can greatly increase performance on traditional tests of memory (Farah 1986). Research has also shown that if asked about the likeness of one sound to another, one would use their "mind's ear" to mentally hear the sounds (Conrad and Hull 1964).

It was suggested by Logie that the visual sketchpad could be further divided into "the visual cache", focusing on form and colour, and "the inner scribe" that deals with movement and spatial information (Logie and Marchetti 1991).

Baddeley provided further evidence to this distinction within the sketchpad showing that spatial and form recall involved the use of different hemispheres (Baddeley 2000). Further results from brain-imaging show that working memory tasks with visual objects activate mostly areas in the left hemisphere, whereas tasks with spatial information activate more areas in the right hemisphere (Cohen *et al.* 1997). Klauer shows there is less interference between visual and spatial tasks than between two visual tasks or two spatial tasks (Klauer and Zhao 2004).

By assuming that the method of form and location of objects are dealt with different by areas of the brain, it might be suggested that some technical aids in real-world systems could favour one mental system over another. Early research (Houghton, Baber, and Knight 2009) suggests, for example, that the use of a metal detector by untrained participants can lead to a significantly diminished ability to recall location information, more so than that of participants using a simpler probing method with a trowel.

Recent human factors research into the use of metal detectors during mine clearance (Herman, McMahill, and Kantor 2001) suggests that performance in location recall would increase if an additional map display was made available to the user identifying the location of targets as they were found. Additional studies (Staszewski 2006) have shown the importance of auditory sensor input. It was shown by Staszewski (Staszewski and Davison 2000) that, when using metal detectors, the audio signal from the detector would be used in building up spatial patterns in the participant's mental map.

In real-world tasks, memory recall of location and environmental features is of significant importance. Whilst searching with technical aids will often significantly speed up the process, it has been shown that when participants are challenged on surrounding environmental features their ability to recall is often disrupted (Houghton, Baber, and Knight 2009).

2.4 Examples of Search Tests

With the growing increase of simulated environments used for training, spatial awareness and spatial memory are often incorporated into the evaluation metrics (Wilson, Foreman, and Tlauka 1997; Rose *et al.* 2000). This is because spatial awareness and memory are crucial to human performance in many tasks typically trained within simulated environments (Wilson 1997). The accuracy of the simulated environment is often tested using relatively simple memory tasks performed after exposure to the system (Parsons and Rizzo 2008; Matheis *et al.* 2007). Search-based tasks have also been used to assess the most appropriate method to display the environment to the user. Pausch *et al.* describe a process for evaluating the difference in performance when a head-mounted display was used instead of a desktop display (Pausch, Proffitt, and Williams 1997). Their method of evaluation was to place a participant at the centre of a virtual room and ask them to search for ‘camouflaged’ targets whilst wearing a head-mounted display. One condition was to use the head tracking system to move the virtual view and the other was to have the participant use a 6 degree-of-freedom (6dof) sensor placed in their hands to move their head view (see Figure 2.1, left image).

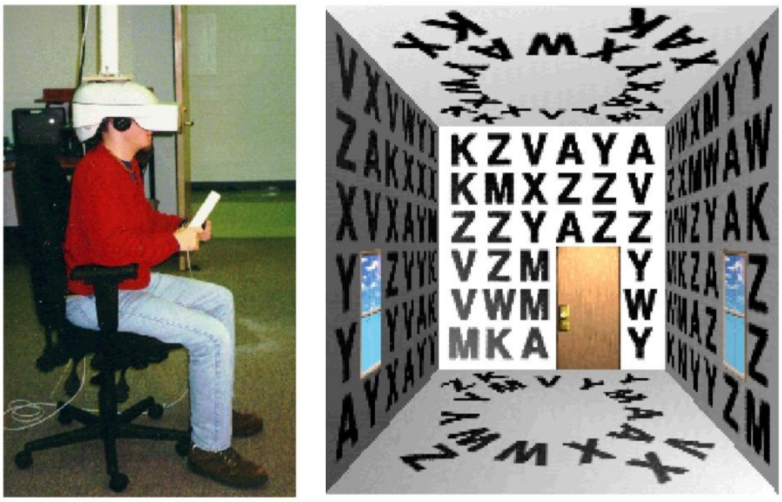


Figure 2.1: Participant wearing stereoscopic head-mounted display (left) and the virtual world with which they are presented (right). (Pausch, Proffitt, and Williams 1997)

The right image of Figure 2.1 is a rendering of the five of the six walls displayed to the participant. Each surface is textured with multiple letters serving as the ‘camouflage’ and only one specific letter is the target. Pausch *et al.*’s findings showed that there was no significant difference in the time it took for a participant to locate the target. In a second experiment they asked participants to search the room and conclude whether the target existed or not. The results from this test showed that the participants using the head-mounted display to control the head movement would typically take a shorter amount of time to conclude whether a target was present. This is most likely due to the more natural motion of the head in rotating the virtual camera view similar to real life. The participants that were using the hand-held controller were found to take longer, as they spent significantly more time re-examining areas of the room they had already searched (Robertson, Czerwinski, and Van Dantzich 1997). Whilst this work indicates the importance of the method of human computer interaction as applied to a virtual environment, it does *not* establish the differences in user performance between a virtual search task and its real world counterpart.

2.5 ROV Simulation

Unmanned and remotely operated vehicles (ROVs) have received increasing attention over the last few years, as their ability to work in extreme or dangerous conditions along with the current miniaturisation of technology have made them invaluable in areas of military, space, medicine and underwater search tasks.

With the increase of modern telecommunications hardware it has become possible to present ROV operators with more than just a single video feed. It would seem that an ever-increasing amount of research into the area of marine biology will inevitably lead to a significant increase in the amount of data that is sent back and displayed to the pilot and/or navigator or mission specialist in order to assist them with their current task (Fong and Thorpe 2001; Garcia *et al.* 2010) (e.g. see Figure 2.2). Ultimately it is the ROV developers that select what information should be sent back and how it should be presented, and this is typically done subjectively by responding to the requests of operators as they gain experience in deploying legacy systems to tackle increasingly difficult remote tasks.

Figure 2.2: Remote submersible control stations with multiple monitors providing data from camera feeds and technical aids[1].

2.6 Current ROV Simulations

During the early days of offshore drilling, engineers would construct scale model replicas to aid in spatial awareness. With the introduction of computer-aided design (CAD), oil platforms and their surrounding terrain could be visualised in 3D, while at first the limitations of the technology only allowed for static images; eventually real-time fly-throughs were possible. It did not take long to realise that additional models of ROVs could be added to enable operational planning. Eventually the use of real-time positional data from an ROV was used to further aid subsea construction and maintenance activities.

Today, whilst CAD can be used to help understand spatial awareness, it has several failings that make it unsuitable to train people in complex, remote operations. Firstly, CAD renderings tend to show an infinitely clear environment which would poorly prepare operators for the severely limiting view of distance of, for example, underwater operations. Secondly, CAD visualisations typically show an unrealistic isometric or third-person

[1] http://www.csip.co.uk/

perspective of the operation, again in contrast to the limiting viewport of the ROV. Finally, CAD software does not support complex physics, such as support and ROV vessel motion or hydrodynamic behaviours of the submersible and its umbilical.

Over the last twenty years the improvement of both computer graphics and physical modelling has meant that ROV simulators have been able to present a more suitable approximation to real ROV control.

There are, broadly speaking, two types of ROV simulator; commercial and academic. Commercial simulators are generally used in ROV training centres and typically focus on training piloting skills and manipulator usage. Modern simulation can not only recreate the data received from the ROV's cameras but also that from other onboard systems, such as depth and heading sensors, sonar, and so on. Academic ROV simulators are typically used for research focusing on developing new control methods or the development of semi- and fully-autonomous systems (Stone *et al.* 2006).

It can be said that ROV simulation has been accepted as an effective method for training piloting skills and manipulator control, as nearly all ROV piloting schools use some form of simulation (Butler and Galerne 1985). However, empirical studies that objectively assess the effectiveness of specifically ROV training are difficult to find. Due to the success of flight simulation (Miller *et al.* 1995) and general manipulator simulations it has been assumed that the same can be said, due to the similar nature of control, for ROV simulation, and this may have affected the lack of research into the area (Cooke and Connor 2006). During the late 1990s several papers did present results from evaluations from ROV simulations, one of the earliest of which is the Training for Remote Sensing and Manipulation (TRANSoM) project (Harris and Fletcher 1996).

The TRANSoM program was sponsored by the US Office of Naval Research and the research team was tasked with the creation of a virtual environment for training ROV pilots. The simulation was based on the Talon ROV, which is an inspection class vessel from Imetrix. The ROV Simulation was validated through research conducted by Barbara Fletcher (Fletcher *et al.* 1996).

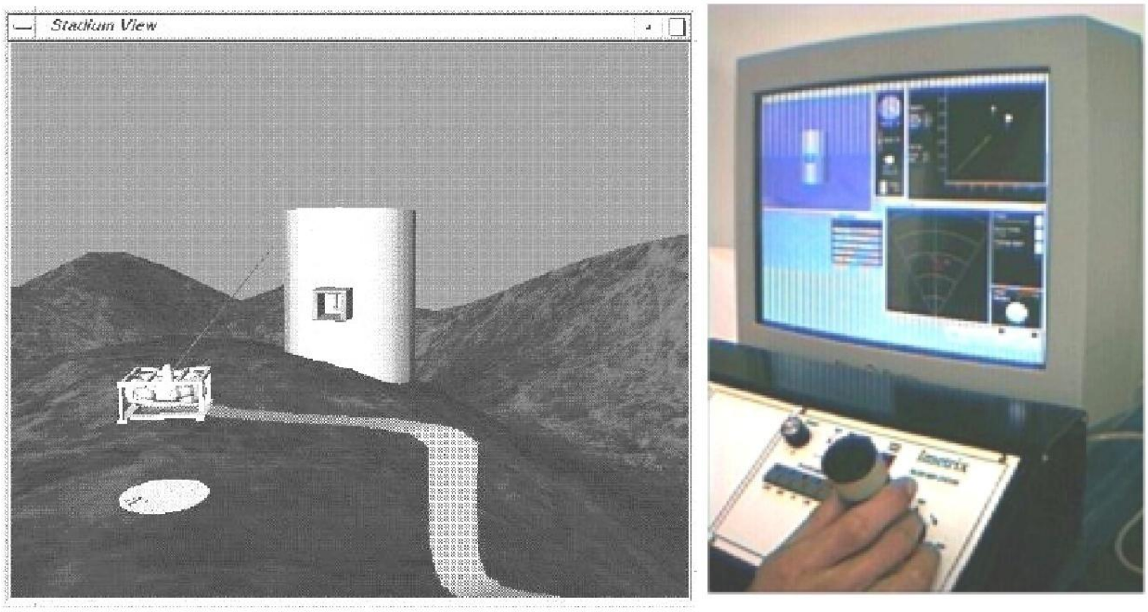


Figure 2.3: Early computer generated virtual environment for the TRANSoM system (left) and its control system (right) (Harris and Fletcher 1996).

The TRANSoM simulation uses a Silicon Graphics workstation that is connected to the real ROV control panel (see Figure 2.3). The screen itself displays the images from the virtual camera along with additional instruments including a sonar image, map views and traditional heading and depth information.

The first evaluation of the TRANSoM system comprised two components. Firstly, the system would be examined by six novice ROV pilots that had experience with the Talon ROV (Fletcher and Roberts 1998). They were then asked to fill in a survey which rated the visibility to target, video display and compass information. Five of the six participants rated the systems displays as 'effective' or 'highly effective'. As the control interface was taken directly from the Talon itself, validating the 'look and feel' of the interaction was not an issue. However, Fletcher specifically wanted to obtain quantitative measures of participant's responses of the virtual ROV to the real ROV (Fletcher and Roberts 1998).

Secondly, Fletcher compared the virtual and the real response in four key control areas; forward and lateral velocity response, heading rate and lateral effect on heading rate. It was found that the real response times typically lagged behind the analytical model within the simulation. This discrepancy was generally put down to additional transient and inertial effects. Fletcher concluded that the system provided a good representation of the system to existing ROVs. It was also concluded that, while general transient effects cause discrepancies in the behaviour of the real and its virtual counterpart, the overall correlation was high.

On reflection, the assessment of the 'look and feel' of the TRANSoM system was very subjective. The novice pilots were not given any specific tasks to perform and no objective measures were used. The assessment of the behaviours of the real and virtual simulations did provide detailed results as to how well the two systems correlated. Interestingly, even though these results pointed to the real ROV lagging in response to the virtual, the surveyed perception of the novice users was the opposite. They felt that the virtual ROV was less responsive. Fletcher states this may be due to the scale or frame rate on the simulation, although it could be suggested that the feeling of movement would be enhanced by the experience of particulate matter in the water (Harris *et al.* 2002).

The work on TRANSoM continued with N. J. Picoh and others (Pioch, Roberts, and Zeltzer 1997). They began to use the system in conjunction with head-mounted displays and head tracking systems with the goal of establishing the effectiveness of presenting the user with additional artificial viewpoints (for example a 'stadium[2]' or 'wingman[3]' view). A more task-based approach to assessment was used after an in-depth investigation at various training centres was undertaken which established three critical skill areas of ROV piloting:

- Manoeuvring – the ability to pilot the ROV while understanding the effects of drag, dynamics and momentum.

- Situational awareness – a general understanding of the vehicle's spatial position and its relation to its surrounding environment.

- Sensor information – the ability to gather information from several displays such as sonar and video. The information is typically partial or degraded.

The experiment had participants perform a four-stage task that included transit, hover, orbit and docking. Some participants were exposed to the traditional camera view while

[2] Viewed from a distant position.
[3] Viewed from a vehicle manoeuvring close by.

others were given additional ‘stadium’ and ‘wingman’ views. The group that trained with the ‘stadium’ view had significantly better performance when controlling depth and following paths.

Picoh *et al*.’s research concluded that manoeuvring skills would typically be acquired faster with subjects that had good hand-eye coordination. It also states that expert pilots would “mentally project themselves” into the ROV to assist in navigational control. Finally, the research also found that good pilots would not become dependent on any particular information display, but would distribute their attention across all displays.

Their research was focused on the use and the effectiveness of additional camera views but both of the camera views suggested were rather fictitious in nature, as a ‘stadium’ view would be impossible to obtain with a real ROV system. Again there was no direct comparison between the four stage tasks (described above) and similar procedures in the real world. What was interesting, and generally left as an additional thought, was that in their initial research it was discovered at the various training schools that a good pilot would not become fixed with one information display. While this was a measured subjectively, they did report that the untrained novice pilots would exhibit ‘tunnel vision’ and often neglect other indicators during particular manoeuvres such as orbiting.

Current ROV research has generally focused on developing new ways to simulate the dynamics of ROV flight (Jordán, Bustamante, and Cortiñas 2005) rather than the training transfer benefits. For example, new methods for simulating tethering cables (Zalesky 1998; Buckham, Driscoll, and Nahon 2004) and torsional mechanics (Buckham *et al.* 2004) have been suggested as well as their use for prototyping new ROVs (Kim *et al.* 2002).

Despite the limited research into the transfer of training benefits of simulated ROV flight, nearly all training centres have adopted the use of at least one system for certain aspects of training (Ormiston 2009). ROVsim, from Marine Simulation, is one of the more advanced simulators, offering high accuracy of physics, the ability to configure missions and the use of additional displays such as sonar (Figures 2.4 and 2.5).

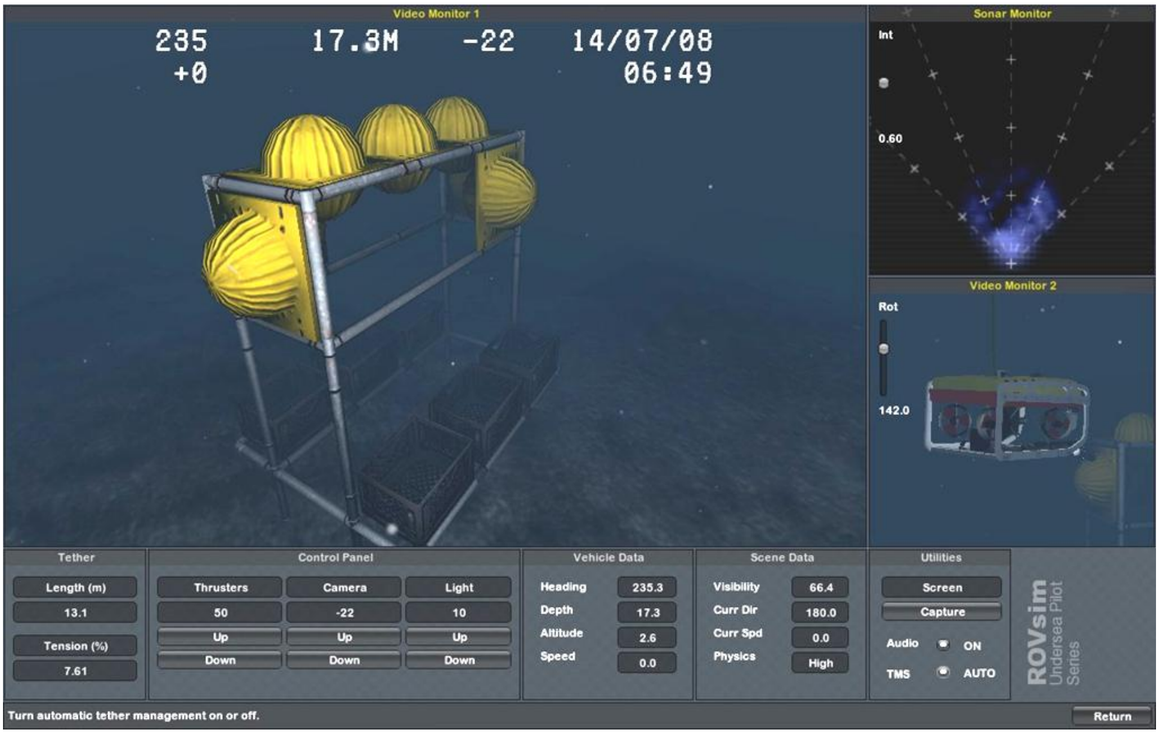


Figure 2.4: ROVsim's user interface for simulated video feeds and technical aids[4].

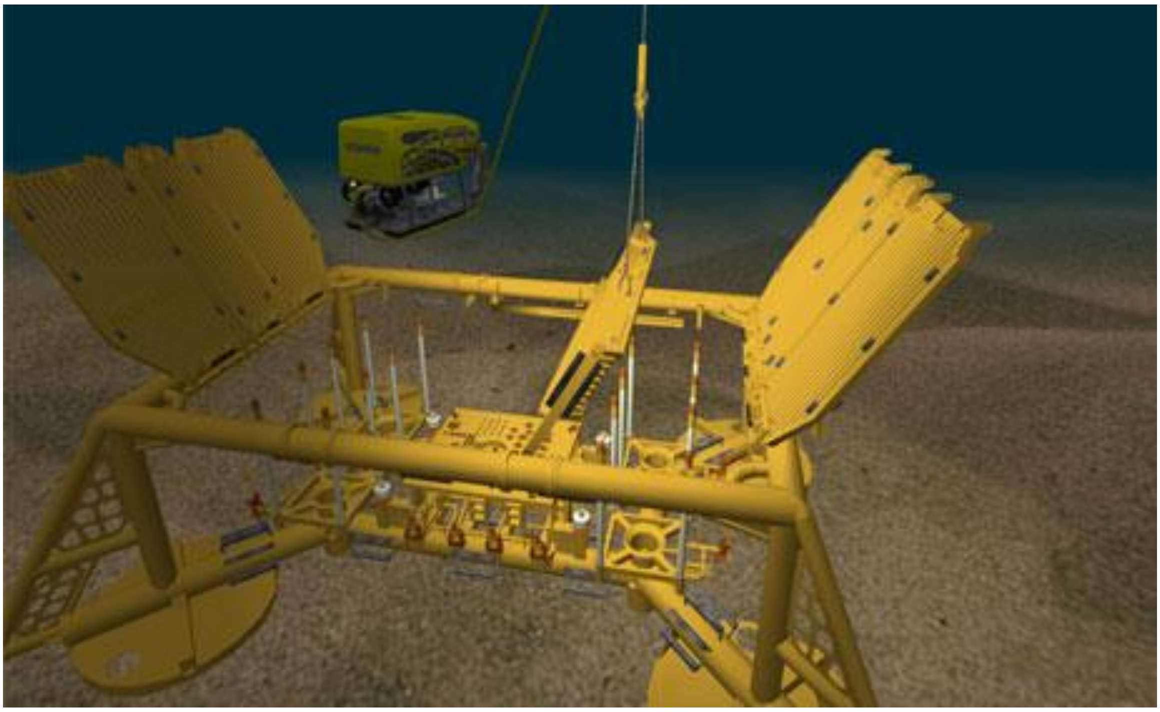


Figure 2.5: ROVsim's close up virtual camera feed during manipulation task[4].

[4] www.marinesimulation.com

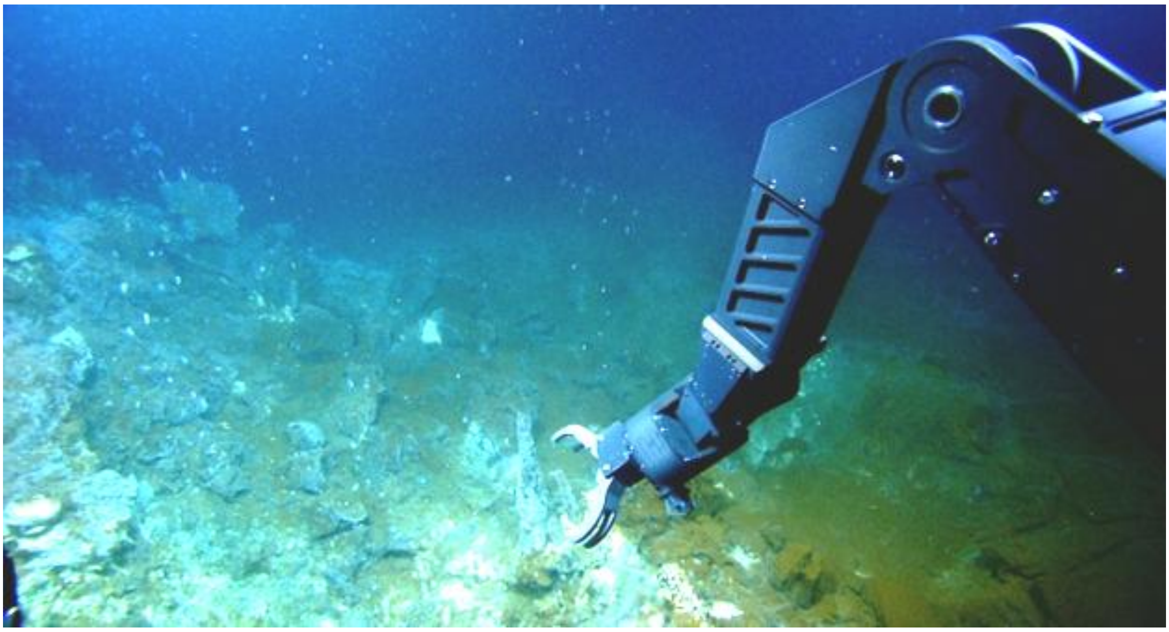


Figure 2.6: Real world close up camera feed during manipulation task, showing visual distortion effect.[5].

While ROVsim is physically accurate, its visual fidelity is relatively low especially when compared with real world underwater environments (figure 2.6). While piloting controls and manipulation may not be substantially improved through better graphics, there is an increasing demand for ROVs to be involved in search and surveillance tasks. Objects may have been deliberately hidden and subtle variations – for example disturbed earth in the case of unexploded ordnance or improvised explosive devices (IEDs) – may be a clue to their location.

2.7 ROV Technical Aids

Current simulators are able to provide realistic representations of technical search aids such as sonar and mapping information. However, no objective method has been established to quantitatively rate their use during specific tasks. The little research that does exist depends on subjective methods of data collection such as questionnaires. It should also be noted that ROV display design has been based on the same principles as those from unmanned air vehicles (Cooke and Connor 2006). ROVs are piloted in very different ways to those of UAVs due to the variations between the two operating

[5] http://www.csip.co.uk/

environments. It is therefore not sufficient to employ the same design principles without first conducting the necessary research to determine the use of additional displays and technical aids during search tasks specifically aimed at ROV systems.

2.8 Discussion

The use of serious games to create simulations for training offers clear advantages over the traditional bespoke virtual reality solutions of years gone by. Modern, user-friendly toolkits and reduced cost allow for the development of complex virtual environments that can cater to almost all areas of training, not least ROV simulation.

We have seen the detailed work that has been undertaken to better understand the human cognitive process involved when performing search and recall tasks. Simple search tasks have been used as a baseline metric for establishing the required visual fidelity within simulations, but performance alone may not be the most suitable method during complex tasks. There is a lack of research into the differences between participants performing complex search tasks in the real world and a virtual one.

We have also seen that the development of ROV simulations are based on assumptions from traditional flight simulation systems, which fail to account for the detailed environmental and degradation factors present in an underwater environment. Current ROV simulations typically have a relatively low visual fidelity which is generally regarded as appropriate for piloting and control tasks. However, ROVs are increasingly being used for inspection tasks (Raine and Lugg 1996; Baldwin 1984) which suggests a growing requirement for search-based tasks to be trained. As search is largely a visual assessment of the images received, it is hypothesised that an increased level of visual fidelity will affect not only performance but also the use of additional technical aids. There is a need to investigate the effects of increasing visual fidelity during tasks and establish if this affects the method employed during the task as well as overall performance. Search tasks have become increasingly important particularly in the detection of explosive devices both on land and underwater. New technologies such as advanced metal detectors, radar systems and an ever-increasing use of unmanned vehicles are being used to aid in search and disposal. This has led to an increased requirement for the user to monitor multiple sources of information during the task. One of the intentions of the presented research is to establish methods for assessing how additional information is used by the operator during a search task.

Chapter Three

This chapter outlines the structure and core components of modern games engines and describes the importance of selecting the most appropriate engine and supporting toolkit(s) for the task. It will also outline the differences between the large ranges of games engines available today and evaluate them for suitability for their use in developing the simulations required for this area of research.

3.1 Introduction

Over the last few years the graphics market has become saturated with games engines, from the low budget and Open Source, right up to the "AAA" engines used by the most powerful games consoles. Historically, there has always been a cost-to-features trade-off that a developer must undertake in order to decide which engine to use. However, with modern games engines, the choice is not so much based on the technical features it can supply, but rather the development tools that are available for that engine.

3.2 Games Engine Design

Modern games engines can be split into three main categories: traditional programming-based, visual programming and "sandbox". The traditional programming engines are largely coded by hand in C++ and simply use a suite of 3D graphics, sound and physics libraries to compile the engine. Whilst this type of engine gives the most freedom it does require specialist programming skills and is typically not very friendly to artists or other non-computing disciplines. Visual programming is a progression from this. Instead of manually writing sections of code to perform specific functions, a visual programming tool provides pre-written component blocks that can be combined to perform the same action without the need to edit raw code (much like Visual Basic). Although this style is more focused towards non-programmers, it still gives the users the option of creating their own components from coding. Finally, the sandbox class of editors allow for direct access to the game environment in a completely "drag-and-drop" style. Users can load a real-time 3D editor and directly modify the environment, often displayed in adjacent on-screen windows. Sandbox editors are favoured by non-programmers and artists who are, using the tools, able to prototype games rapidly. The main downside to the sandbox style of editor is that there are limitations on how much of the game engine can be altered. Modern games engines typically comprise seven core components. Each of these is briefly outlined below, indicating their use and importance.

3.3 3D Rendering

3D Rendering is the core component of any modern game engine. The renderer converts the 3D mesh objects into 2D images for rasterisation (the process of converting 3D vectors to screen pixels) to output to the computer monitor. For complicated 3D scenes, a huge amount of processing calls (a list of tasks for the processor to perform) is required for the conversion process. This led to the development of specially designed 3D processing hardware in the 1990s. This is typically performed using a set of component libraries such as DirectX or OpenGL which can interact directly with the available 3D hardware. Specialised hardware, such as the graphics processing unit developed by Nvidia (Buck 2007), is almost always used to speed up the process.

3.4 2D Rendering

In addition to a game engine's 3D capabilities, traditional 2D elements are also usually required. Head up displays, menus and GUI (graphical user interface) systems are handled by a 2D engine, although in recent years this has typically been incorporated into the 3D rendering engine.

3.5 Audio Processing

Audio in one form or another has always been incorporated into games, from simple tonal effects and cues to "surround sound". Modern games now demand high quality sound effects and music. It is often required for the sound system to process 'mp3' file formats for music and be able to incorporate spatial 3D sound effects based on the position of objects in the virtual scene, as provided from the 3D rendering engine. With the increasing use of games engines to produce "live" cut scenes between levels or chapters of the game, issues such as lip synchronisation are now becoming highly important in enhancing the believability of the game content. This requires the animation system to be closely linked to the audio system. Just like 3D rendering, modern games engines require specific hardware to produce the more complicated surround sound effects. Examples include EAX or Dolby digital.

3.6 Physics

Game engines are largely dependent on physics-based effects in one form or another, because physics is the science that governs the nature and behaviour of elements within the perceived world around us. When we speak of game engine physics we are not referring to issues such as quantum mechanics or relativity, but more to classical mechanics. Game physics deal only with the laws of motion that govern the behaviour of objects under the influence of forces such as gravity. These simple laws of motion enable objects to behave as real-world entities displaying, for example, inertia, bounce and buoyancy. Game physics has been a necessity from the very early days of computer entertainment, typically simulating the effects of explosions, smoke and estimating the projectile course of munitions. More detailed physics are necessary to simulate the motion of vehicles such as aircraft and cars. The most recent games require players to solve physics-based problems, such as delicately balancing objects to create bridges or constructing and operating a make-shift catapult. While these new and more detailed physics routines provide an interesting new game play mechanism, they are not in any way sophisticated enough, as yet, to allow for training in the real world. Modern game engine physics usually fall into two categories, rigid and soft body.

3.7 Rigid Body Physics

Rigid bodies are non-deformable geometric shapes that occupy positions in a 3D space. They are described as having six degrees of freedom (i.e. they can move in three directions (x, y, z) and can be rotated around these three axes ("roll", "pitch", "yaw"). They are subject to outside forces (e.g. gravity, wind) and have properties including mass, friction and bounce. Rigid body simulation is one of the most common physical elements to simulate and has existed in some form since the earliest computer games. In recent years, rigid body systems have been used to simulate ever-increasing complex systems such as hydraulic cranes and vehicle mechanics.

3.8 Soft Body Physics

Unlike rigid body physics, soft body dynamics focuses on accurate simulations of flexible or deformable objects. This essentially means that every vertex point that defines the geometric mesh of a virtual object can have individual dynamic forces applied to it, as opposed to a single set for the entire object. With each vertex point capable of independent movement, the overall effect allows the object to be deformed. This type of simulation is particularly CPU (central processing unit) intensive. For every vertex point within the object, the effect of dynamic forces needs to be calculated. This could easily require tens of thousands of calculations per second, even for the simplest of objects.

While full, soft body dynamics is rare within modem games engines, a simplified form, often referred to as “cloth simulation”, is used to apply dynamics to simple planar objects such as flags, clothing or hair. The technique that is the most common is based on the notion of a mass-spring system shown in figure 3.1. Simply put, a continuous cloth surface is discretised (Volino and Magnenat-Thalmann 2001) into a finite number of particles, much like a sphere is divided into a group of vertices and triangles for drawing with 3D hardware. The particles are then connected in an orderly fashion with springs. Each particle is connected with springs to its neighbours along both the horizontal and vertical axes. These springs are called “stretch’” springs because they prevent the cloth from stretching too much. Additional springs are added from each particle to its four neighbours along the diagonal directions. These “shear” springs resist any shearing movement of the cloth. Finally, each spring is connected to the four neighbours along both the horizontal and vertical axes but skipping over the closest particles. These springs are called “bend” springs and prevent the cloth from folding in on itself too easily.

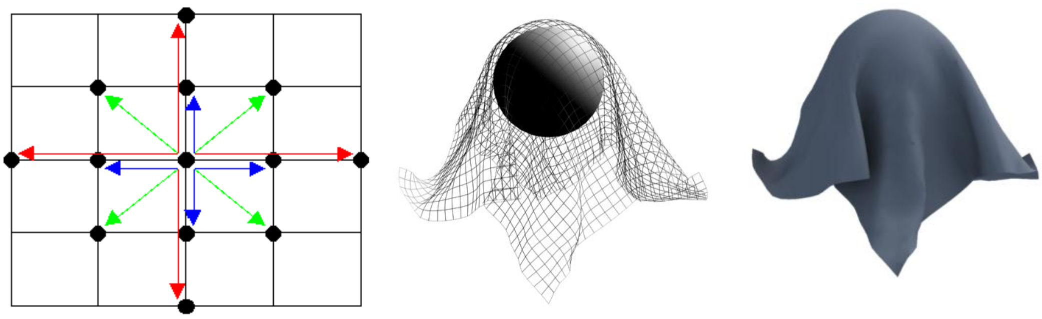


Figure 3.1: Illustrating the mass spring local vertex connections (left) and the wire frame and shaded cloth model draping over a ball (right).

Another method of simplifying the calculations from soft body dynamics is to use an underlying bone structure for each object that can be used for the calculation instead. Each bone affects a weighted group of vertex points within the virtual object. This technique is heavily used in character animation allowing for the deformation of skin and clothing as the character moves. With this method, only a fraction of the required physical calculations are required, as the character model's vertex points no longer require direct calculation. Instead, the character's movement is governed by the weighted influence of a handful of bone elements

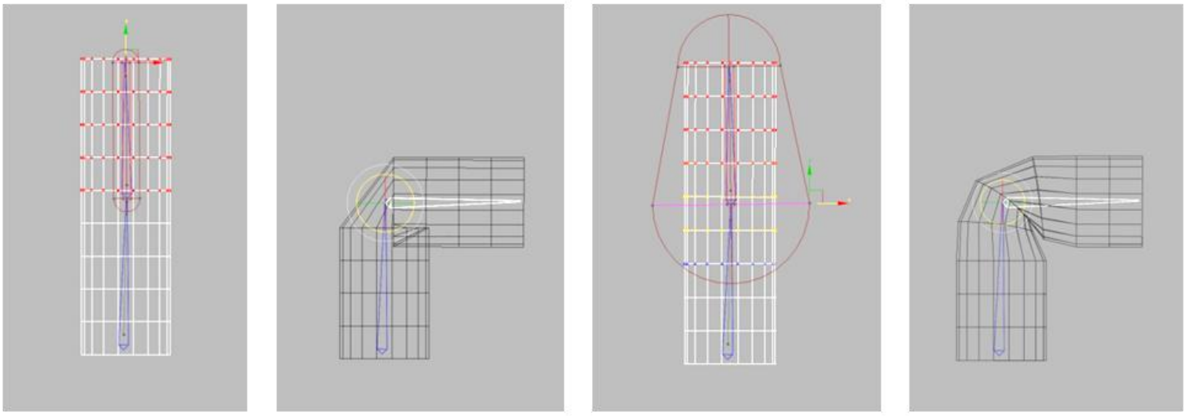


Figure 3.2: Illustrating bone and vertex interaction with fixed vertex weighting (left) and with a blended vertex weighting (right).

Figure 3.2 illustrates the bone placement and the vertex weighting. Vertices close to the selected bone are displayed in red, indicating a high level of influence, while distant vertices that are affected less are displayed as blue. Each vertex can be affected by multiple bones with varying degrees of influence allowing for soft blending between bone joints.

A form of soft body dynamics is also used to simulate the movement of virtual characters. The virtual character's skin or clothing must appear to bend and deform while moving, but as stated earlier, the processing power required for soft body simulation of objects can be immense. To reduce the processing cost a simplified virtual bone structure can be used for calculation purposes and be applied to the virtual character's mesh. While this technique is mainly used for the character models it can also be applied to objects such as rope, or even internal organs. (Figure 3.3)

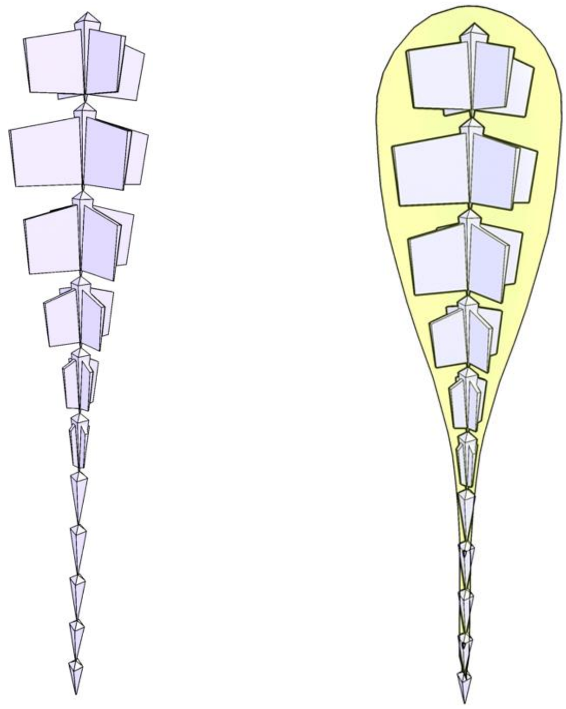


Figure 3.3: Illustrating how underlying bone structure can be adapted to fit non uniform shapes.

3.9 Particle Effects

Particle systems attempt to simulate realistic movement of objects, much like a physics system, but here the techniques focus on the fuzzy phenomenon of partial movement. Effects such as smoke, water, clouds and snow are just some of the typical examples that require specialised rendering techniques to simulate.

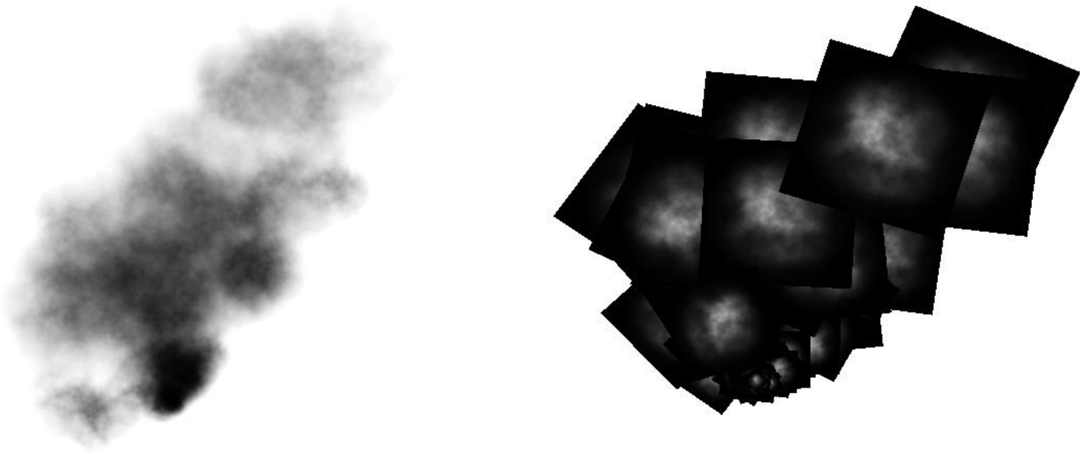


Figure 3.4: Illustrating virtual particles with blended transparencies (left) and the underlying virtual billboards with non transparency (right).

In the real world, dust and smoke comprises millions of particles that are affected by gravity, wind and inertia. It is far too computationally complex to compute the position of so many particles in current-generation games engines.

The most common way to simulate particles within a games engine is to use multiple billboard textures to represent a group of particles. Now only the position of each billboard group is calculated rather that each particle. This can be seen in Figure 3.4, where a cloud of smoke of thousands of particles is simulated just using 20 billboard textures. A particle system's position and motion in 3D space are controlled by what is referred to as an “emitter”. The emitter acts as the source of the particles, and its location in 3D space determines where they are generated and the paths along which they progress.

3.10 Animation

Modern games typically involve numerous virtual characters, many of which not only have to look realistic but move accurately as well. 3D art packages, such as 3D Studio Max and Maya can be used to create a series of animations for integration with the final game. However, modern games have the ability to not only play back animations but blend them together. For example it is possible to have two separate animation sets for running and shooting, but the game engine can combine them.

3.11 Artificial Intelligence

Broadly speaking, Artificial Intelligence (AI) attempts to "give a mind" to the virtual characters within a game. Referring to AI within a game engine is a little misleading as it should not be confused with modem academic research in the area. Academic institutes are developing AI systems that can truly learn and make decisions based on arbitrary data (Michalski, Carbonell, and Mitchell 1986), (Wenger 2004). This is not the case for games-based AI, which is, to all intents and purposes, a severely 'dumbed down' version of this. Game AI is usually based on a long series of rules which govern the actions of the AI entity. When certain conditions are met, a specific rule can be invoked (e.g. 'if the player is within range and ammunition resources are full, fire weapon'). A game engine's AI system is often closely linked to its physics systems to allow for character path-finding, line of sight calculations and collision detection.

3.12 Controller Interfacing

Controller interfacing is the term used to describe methods for interaction with the user. Game controllers, such as joysticks, gamepads, flight yokes or steering wheels are often used. The software needs to be able to interface with various forms of input device and translate the user's movements into the virtual movements of the on-screen entity, be that a character, one's own viewpoint, a vehicle, and so on.

3.13 Force/Torque ("haptic") Feedback

Haptic user interfaces have been used with virtual simulations for many years. These can range from large scale six-degree-of-freedom motion platforms for flight simulation (Reid and Nahon 1988), down to simple vibrations from an Xbox game pad. Many studies have shown the importance of a physical response from simulations and their necessity for training certain tasks (Wagner, Stylopoulos, and Howe 2002).

Modern computer games engines support their own version of haptic controllers, such as steering wheels, joysticks and even mice. For console-based games, the haptic system output is relatively basic, only capable of simulating "rumbling" and vibration sensations through the use of off-centre weighted motors within the controller housing. These systems are generally used as an additional form of multimodal information given to the user ("haptic cueing") rather than truly simulating any physical experience. Examples of this may include simple vibration when a special game object is near such, as a key to an important door or chest, or a more vigorous rumble when the player's "health" is running out. More advanced systems are present in modern steering wheels and joysticks. Through the use of multiple servos, they are capable of directly applying forces to the input device to simulate real world physical effects such as friction, G-forces and skidding. These types of controller are generally referred to as force feedback systems.

3.14 Graphics Libraries

A graphics library is a set of standardised routines that can be called upon to perform 3D rendering. Nearly all modern rendering engines are based on one of two libraries, DirectX or OpenGL.

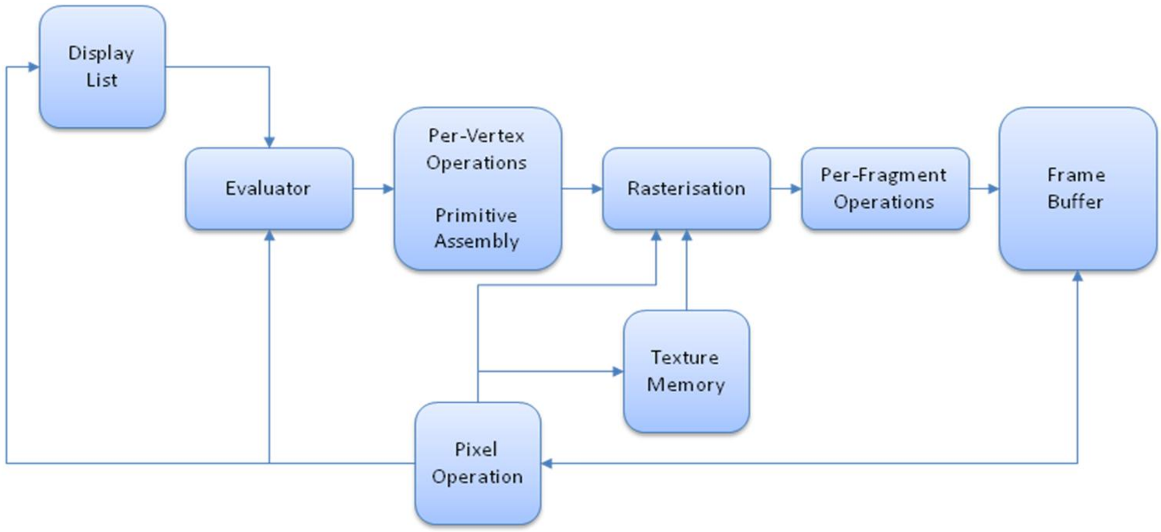


Figure 3.5: A visual representation of the OpenGl (OpenGl 2.1).

OpenGl, originally developed in 1992 by Silicon Graphics (Neider, Davis, and Woo 1997), is an environment for developing platform-independent, interactive 2D and 3D graphics applications. Figure 3.5, above, shows the standard rendering pipeline for OpenGl. In addition to the rendering of simple points and polygons ("per-vertex" operations), OpenGl also supports the use of "per-pixel" operations, such as lighting and anti-aliasing (the process of re-sampling the 3D world to provide a smoother image on sharp lines). OpenGl is more of a traditional state machine structure, using a traditional stack structure where stacks, such as transformations can be pushed and popped. In the mid-1990s this structure was preferred as it was easy to implement in C and the number of calls could be optimised through scene graphs.

Whilst OpenGL has the ability to port to almost any operating system, it does not provide any libraries for user input, networking or sound. Additional, specific libraries would have to be individually called for each of these.

In 1995 Microsoft released their own competing libraries, called DirectX (Bargen and Donnelly 1998), which were exclusive to the Windows operating system. Early versions were not favoured by programmers due to unnecessarily complex calls to perform simple actions. However, subsequent versions became more popular.

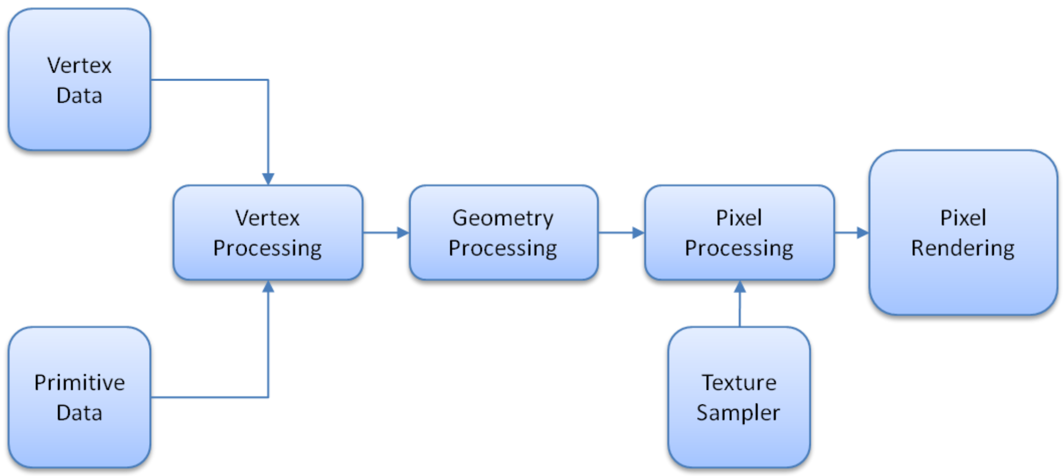


Figure 3.6: A visual representation of the Direct X Pipeline (DX9).

The main rendering pipeline is shown in Figure 3.6. It is worth noting that this structure is based on DirectX 9 (DX9). DirectX 10's (DX10) structure includes a more extensive instruction set and a geometry shader, allowing for geometry to be created directly. At the time of writing (2010) both DX9 and DX10 were still both in use, as DX10 requires Microsoft's Windows Vista to be running as the operating system. This led to games developers producing software that could utilise either DX9 or 10, depending on the operating system in use. As stated, unlike OpenGL, DirectX is a set of libraries not just for graphics rendering. The direct libraries also include DirectPlay, for communicating with interface devices, DirectSound and DirectDraw. Utilising just one set of libraries for interaction, rendering and sound has made DirectX the dominant player in the PC games industry. While companies such as "id", the developers of *Doom 3*, prefer the use of OpenGL for rendering, they still make use of DirectX for sound and controller interfacing. Although DirectX is restricted to only Microsoft Windows-based operating systems, it is generally not seen as a significant disadvantage for the gaming market, as the majority of users actually run Microsoft Windows (Gilbert and Katz 2001). The object orientated approach as well as a more encompassing set of libraries had led to the majority of games engine being DirectX based.

3.15 Games Engine Selection

There is a huge number of games engines available today, from basic coding to high level sandbox style design (Andreoli *et al.* 2005). At the beginning of any project a decision must be made early on to determine the most appropriate for the application under development. The type of games engine can be broadly classified into one of three groups, distributed along a continuum that broadly trades ease-of-use with diversity of functionality.

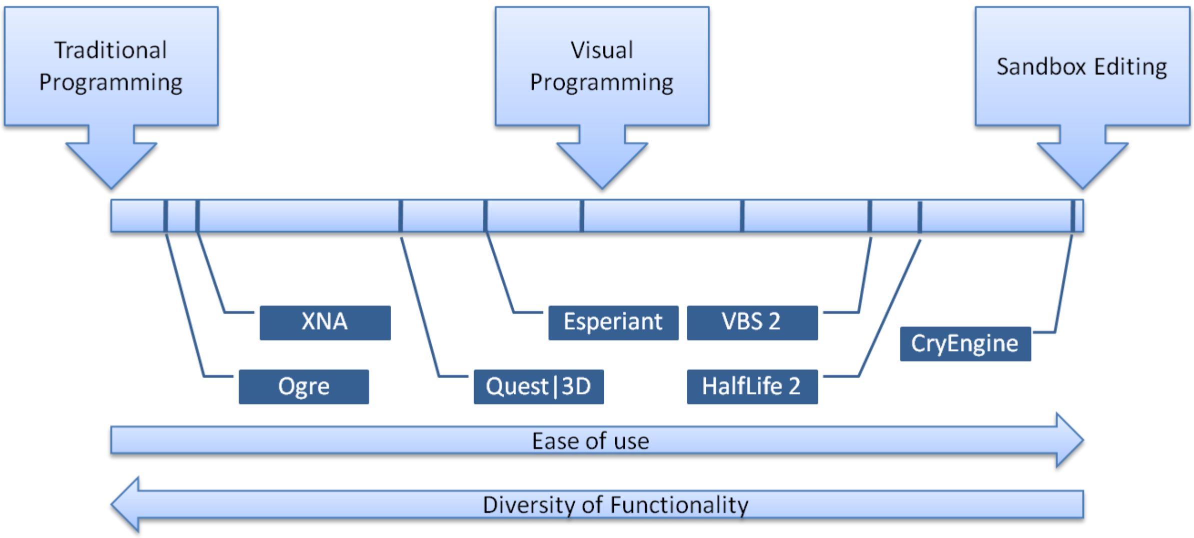


Figure 3.7: The game engine usability/functionality trade off scale.

Figure 3.7 shows some of the more popular development engines (at the time of writing) and their position on this continuum. For the present research programme, numerous engines were evaluated for their suitability to the simulations to be developed. Table 3.1 shows an overview of the engines evaluated for the research, indicating the availability of key features such as lighting, physics and cost.

Features	Ogre3D	Quest3D	Unity3D	Id Tech (Doom)	Source (HalfLife)	CryEngine (Crysis)
Development Language	C/C++	C++ /Lua/Visual Node programming	C++/Java/Lua	C++/Java/Lua	C++/Java/Lua	Lua
Lighting and Shadows	Examples of Stencil shadowing Available	Stencil shadowing and Shadow Maps	Stencil shadowing and Shadow Maps	Stencil shadowing	Stencil shadowing and Shadow Maps	Stencil shadowing, Shadow Maps and SSAO
Physics Engine	Open Dynamics/ Newtonian	Open Dynamics/ Newtonian	Inbuilt	Inbuilt	Havok	Inbuilt
GUI Editor	Non	Yes plus render to texture ability	Yes	Yes plus render to texture ability	Yes plus render to texture ability	Yes
Level Editor	Non	Non	Drag and drop sandbox	Drag and drop sandbox	Drag and drop sandbox	Real-time Drag and drop sandbox
Library Base	DirectX/Open GL	DirectX	OpenGL	OpenGL DirectSound	DirectX	DirectX
Availability of Source Code	All	Most	Partial	Non unless fully licensed	Non Unless Fully licensed	Non unless fully Licensed
Cost	Free	£100 Educational £1500 full licence	£100 Educational £1500 full licence	Limited Free educational use +100k for Licence	Limited Free educational use +100k for Licence	Limited Free educational use +10k for Licence
Operating System	Windows/ Linux/OSX	Windows	OSX for development	Windows	Windows	Windows

Table 3.1: Detailing the features and cost of six evaluated engines.

The engines evaluated fall into one of three development styles: Traditional programming (Ogre), Visual programming (Quest3D) and sandbox (IdTech, Sorce and Crytek). Before a final choice can be made on the engine used the most appropriate development style must be chosen. Each development style is described below.

3.16 Traditional Programming

Traditional programming engines, such as Ogre3D, require the developer to write their application directly in C++. The Ogre3D engine is essentially a group of class libraries that can be used instead of interfacing directly with the DirectX or OpenGL libraries. While developing applications with pre-designed Oge3D class libraries can greatly speed up development time for DirectX calls, Ogre3D provides no tools for additional game components, such as audio and physics. While only a basic set of libraries are provided, as it integrates directly with any C++ code, there are almost no limitations to what the tool can be programmed to do. Ogre3D is also an Open Source engine with a very large community allowing for greater support. It is worth noting that, while the core class libraries are free from licensing, some additional libraries do have commercial restrictions. While Ogre3D is really set up for those competent in programming, more user-friendly types of programming styles have allowed a broader base of programmers to become involved in games development. C#, which hides some of the complex details of the programming language, has now become more integrated with 3D graphics development with Microsoft's XNA, for example. The typical traditional programming interface can be seen in figure 3.8, its largest window is used to type in the appropriate instructions and library calls. The tool box holds additional code modules mainly used in the design of the user interface. After every change to the software the entire code has to be recompiled to show the difference in the 3D engine, any compiling errors are shown in the error list.

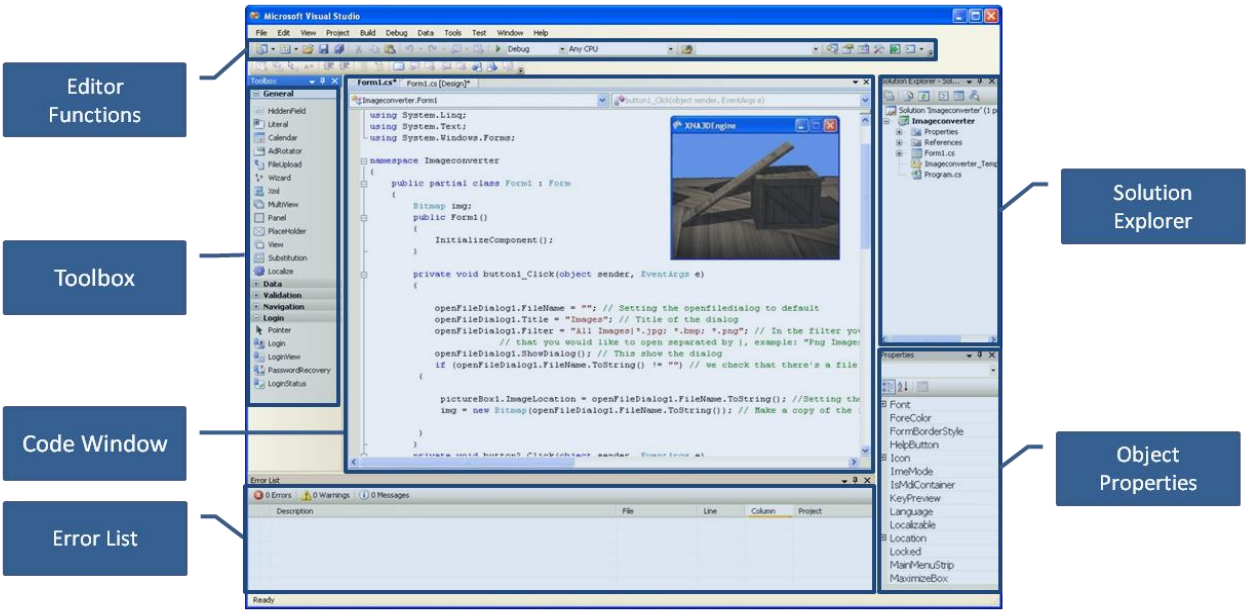


Figure 3.8: XNA Typical Programming interface.

3.17 Visual Programming

Quest3D is a 3D development tool that uses a visual style of programming to radically speed up the development process and reduce the need for heavy levels of programming. Quest3D uses a hierarchical structure of program creation allowing users to add code modules, or so-called channels, to create real-time interactive 3D applications. The channels can be created using C++, LUA scripting, mathematical expressions or they can take the form one of the pre-built channels for handling basic 3D functions.

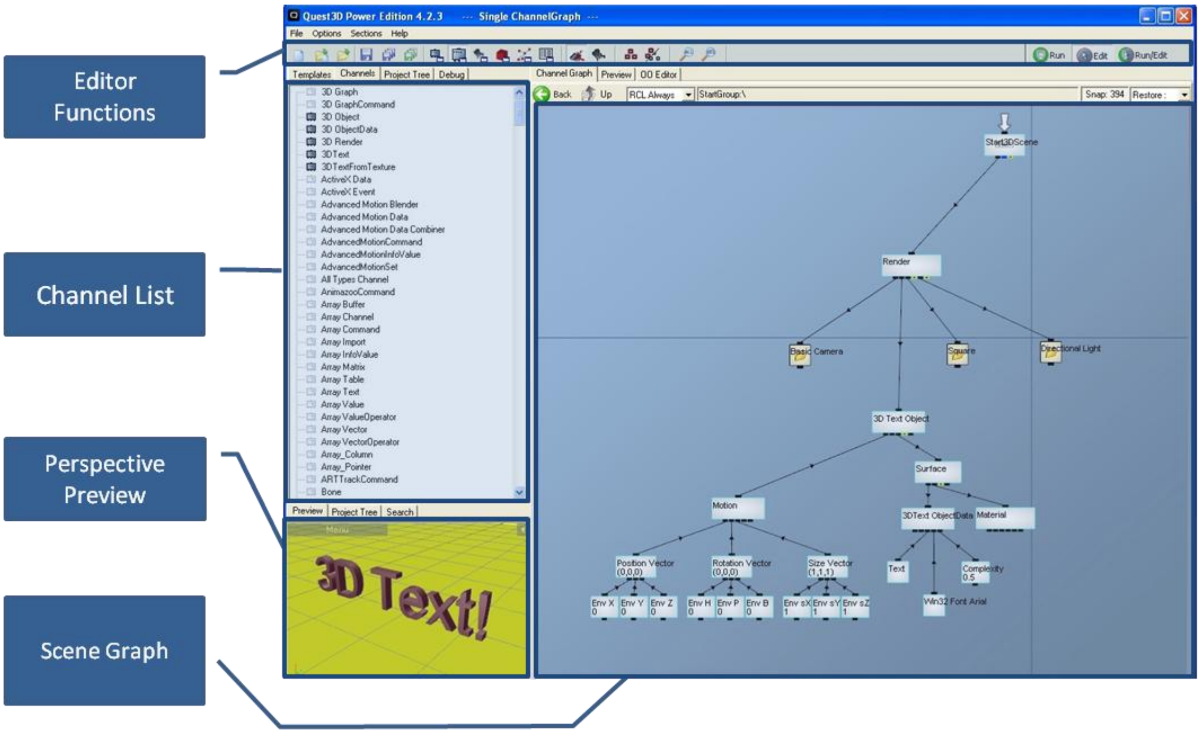


Figure 3.9: Quest3D channel based design interface.

The Quest3D interface, as shown in Figure 3.9, contains four main elements. Firstly, the editor bar allows the user to use traditional load/save functions but, more importantly, provides the means for switching between the three main editing mode views – channel graph view, animation view, object editing and run mode. On the left of Figure 3.9 is the channel list. In essence, this is where all channels, both user created and pre-built, are listed. There are around 300 standard channels, ranging from matrix operations to array management. Below the channel list is a small preview window showing the current output of the application. Finally there is the scene graph. This is the main area for development into which users drag and drop channel groups in a hierarchical format to produce a real-time application.

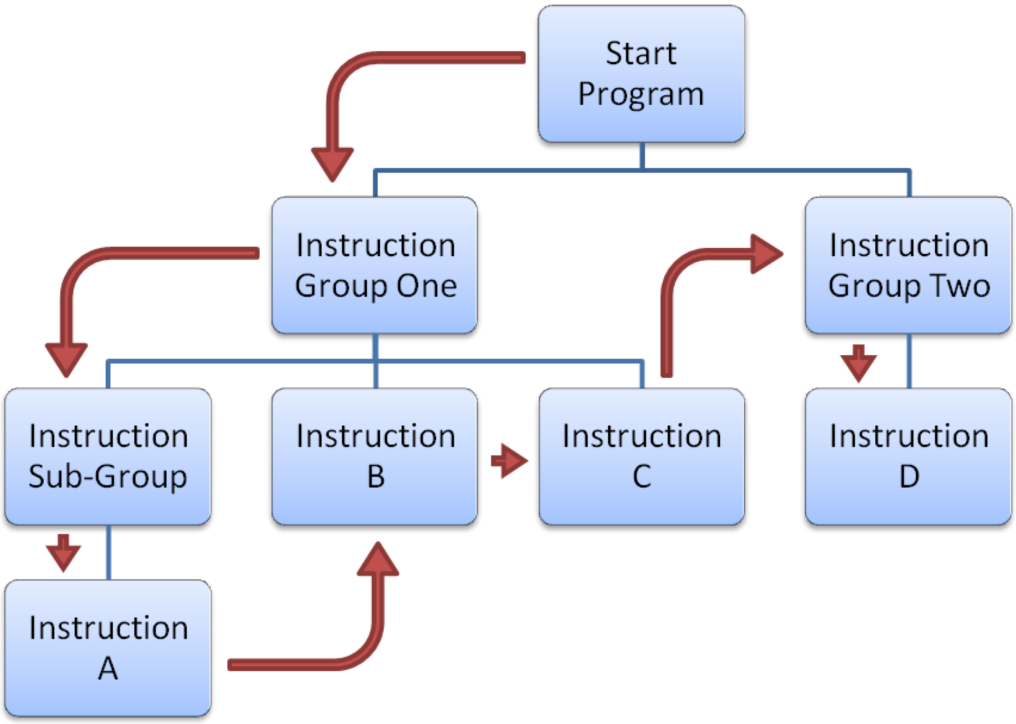


Figure 3.10: Quest3D hierarchical flow of execution through channel groups.

Figure 3.10 illustrates a basic Quest3D hierarchy with the red lines indicating the path of execution. In many ways, Quest3D's hierarchical structure represents a traditional C++ class diagram with each channel representing a class library. Quest3D, unlike Ogre3D, is able to run code dynamically without the requirement for compiling with each change. A live preview window can be opened to allow the developer to see the direct effect of changes to the channel graph. To aid in the better organisation of the application, multiple channel graphs can be used that are linked together through the use of public channel shortcuts. Any section of channel graphs can be cut, copied and pasted to speed up development.

Once the application is running the quest channel hierarchy executes instruction channels from left to right. It terms of traditional programming the run of execution can be visualised as shown in Figure 3.11.

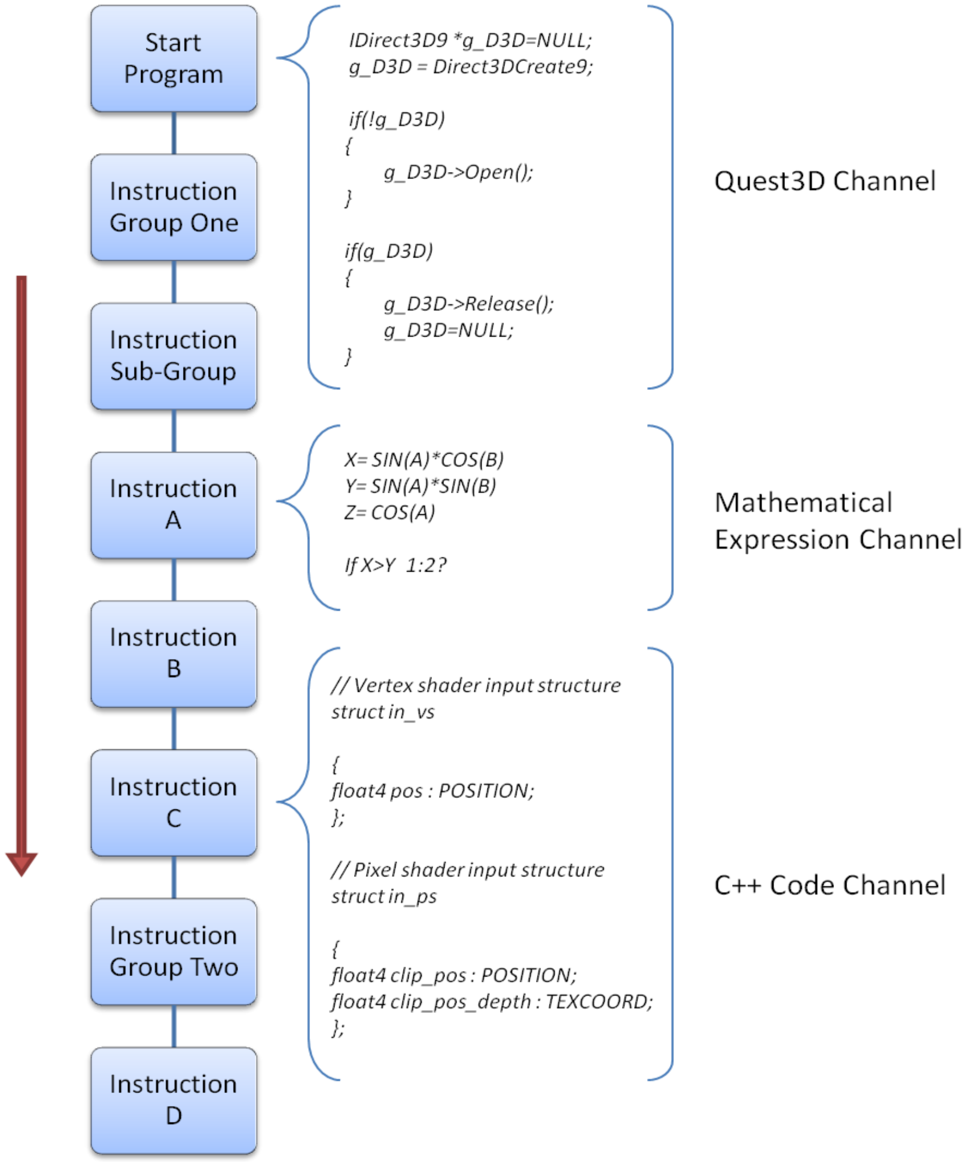


Figure 3.11: Quest3D flow of execution in traditional format.

The Quest3D inbuilt channels provide the basic building blocks of a 3D application. Operations such as hardware interfacing, DirectX library calls and sound generation can all be easily implemented without the developer spending time writing basic 3D operations. The mathematical expression channel is used to allow the user to develop their own logic and control systems. Finally, when required, developers can create their own channels using C code which is often required for more specific 3D operations, such as controller interfacing.

3.18 Sandbox Design

Crytek's *CryEngine* is the embodiment of sandbox creation. An instant drag and drop preview window, inbuilt dynamic shadows and a huge library of objects and characters make *CryEngine* one of the simplest editing packages currently available.

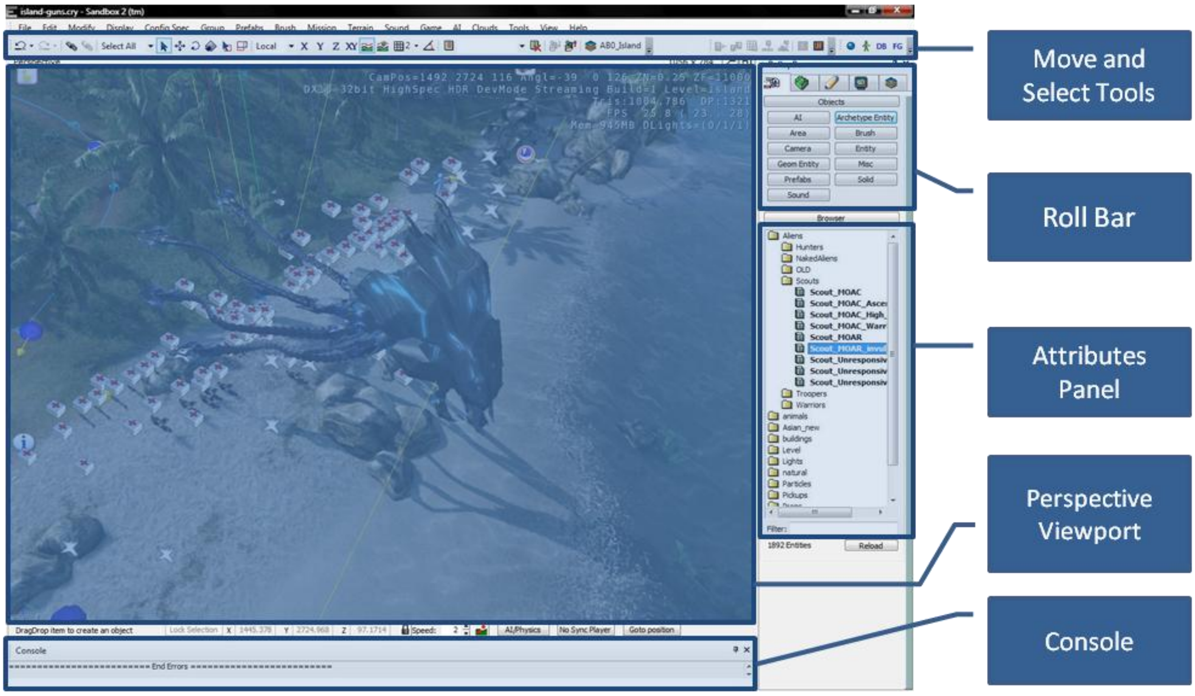


Figure 3.12: Crytek's sandbox level editing interface.

The main menu at the top of the screen in Figure 3.12 contains the menu options for the editor. It includes a number of elements, from basic file save and load functions to terrain editing; from display options and access to the more advanced functions of the editor. The perspective viewport window is the main window used to view a specific level. This is where the large majority of level design tasks will take place, such as object placement through the use of rotation and scale tools, terrain editing and in-editor play testing. The Roll Bar is basically a quick menu for many kinds of functionality within the editor, stored in an easily accessible format. The Roll Bar is split up into several tabs, containing object creation tools, terrain editing tools, display options and layer organisation tools. The Console is a command line editor, providing access to many advanced functions within the sandbox editor, including various debug and test modes.

Figure 3.13: Illustrating the above and below water visuals of the Crysis engine.

While *CryEngine*'s land-based visuals are largely considered the "best of breed" their underwater visualisations are far more basic. The developers, Crytek, as with most games companies, aim to simulate the effects and environments typically associated with the movie industry (e.g. Figure 3.13), as opposed to those more indicative of real-life scenarios. Most games show large amounts of lens flare to simulate a 'Hollywood' camera effect. In terms of their underwater simulation they have chosen to show a rippling distortion effect. While it might look visually appealing it is not in any way a realistic effect of underwater conditions. Unfortunately significant changes to the methods of rendering are just not possible in sandbox games, at least not without access to the product's source code.

3.19 Engine Comparison

As stated at the beginning of this chapter, the requirements of any specific application for games-based simulation define the qualities and functionalities of which games engine should be adopted. As the majority of the experimentation for the present research was to be based in an underwater environment, it was clear that the engine had to be able to provide a high visual quality in order to simulate the complex effects of water, and to allow for a detailed flight model to be created as well as interfacing with bespoke hardware.

Whilst the *CryEngine*, described above, provides high-quality visuals for land-based simulation, its underwater effects were more akin to an artist's representations, as opposed to a realistic appearance. Another critical drawback for *CryEngine* was its inability to interface with bespoke user control systems - only the Xbox 360 game controller is supported. At the opposite end of the spectrum we have the traditional bespoke engines such as XNA and Ogre3D. While Ogre3D had no restrictions with bespoke interfaces, a significant amount of development time would be required to support the development of underwater simulations. One final issue to consider with all the games engines is cost. While developing simulations for research and experimental purposes allows free or low cost use of the engines some aspects of the work developed here will form commercial exhibits. For this reason careful consideration was placed of the commercial cost of the engine evaluated, effectively ruling out the use of IdTech, Source and CryEngine.

It appeared, then, that Quest3D offered the best of both worlds. Whilst there are still some restrictions on development, such as being limited to DirectX 9 and only partial source code availability, the tool provides enough functionality to integrate bespoke control systems and develop complex flight models. In terms of visual quality, it would take a significant amount of time to approach Crytek's quality in above-water scenarios but below-water it is possible to produce a more realistic effect in an acceptable amount of time.

3.20 Asset Creation

In addition to the fundamental games engine component, a wide range of software tools are required to create a detailed, rich and realistic 3D environment. Assets such as 3D geometry, textures and sounds must be developed and imported into the engine. The 3D tools allow modelling with polygons, NURBS, and Subdivision modelling. They are also able to simulate particle effects, fluid dynamics, cloth, fur, hair, physical effects and inverse kinematics for character animation. Two popular and widely-used 3D modelling, texturing, animation, and effects applications are Autodesk's 3D Studio Max, currently version 10, and Autodesk's Maya (version 7).

3D Studio Max is arguably the most widely used 3D package available. It became a successor to 3D Studio versions 1.0-4.0 for DOS. It contains most tools available in Maya and is used by companies in the media and entertainment industry, including Industrial Light and Magic (ILM), Blizzard Entertainment and Ubisoft. The most powerful tool for texture creation is Adobe's Photoshop CS, currently version 2.0. Photoshop is an industry-standard tool and the market leader for commercial bitmap graphics generation.

3.21 Discussion

With an ever increasing range of games engines available to developers it is important to understand the basic components of game engines and which are required for a particular task. We have seen in this chapter the key components of modern games engines and described their basic underlying structure. While all modern games engines possess these features the methods in which they are implemented and, more importantly, how the user can use them is of critical importance. We have seen how traditional programming can give the developer complete control of all of these components but at the cost of usability and development time. Another key disadvantage to traditional programming is that any alterations to the simulation software cannot be seen until the entire code is recompiled and run. Both the visual programming and sandbox development tools allow for instant feedback on the changes made, their underlying code is compiled without disruption to the user. This allows for a much quicker workflow when positioning objects within the 3D world or altering subtle differences in the user interface.

While the sandbox development packages provide the quickest development time the lack freedom and integrating bespoke human interfaces devices can be difficult. There is also a potential problem with licensing certain high end sandbox development engines such as CryEngine. Usually the terms and conditions of the software allows for the uses of the software for non commercial applications such as academic research. However, there is no guarantee that this will not change as developers often reserve the right to alter the licensing agreement without notice. Quest3D was ultimately selected as the engine on which to base the practical elements of this thesis, due to its simple hierarchical structure and rapid development abilities.

The following chapters outline how the Quest3D engine was used to develop three simulations designed to investigate the human factors implications of performing search tasks with technical aids in a virtual environment.

Chapter Four

This chapter describes the development of a simulated virtual environment designed to investigate the use and effect of technical aids in real and virtual search tasks. The first experiment focuses on two simple technical aids, a metal detector and a trowel, and compares the effect on recall and location in both the real world and simulated tasks.

4.1 Introduction

Comparisons between a person's ability to search, locate and memorise information about a target are relatively common within in the area of virtual and augmented reality (Cutmore *et al.* 2000; Ruddle, Payne, and Jones 1999). Recall of the target information is often (Brooks 1999) used as a measure of a person's ability to search; although it may be better to describe recall as a measure of the amount attention which is applied to the task (Duncan and Humphreys 1989).

4.2 Motivation

As stated in chapter two section four, typical virtual recall tasks involve a participant being placed in a fixed position and asked to visually search an area for targets and attempt to memorise their location and/or visual features (Parsons and Rizzo 2008). Variations of methods of interaction and display, such as head-mounted systems, mono or stereoscopic views, are evaluated (Matheis *et al.* 2007) against the user's ability to recall information about the targets.

One notable example is the work performed by Katerina Mania where participants were asked to recall objects in a virtual world when using HMDs whilst the interaction method was varied (Mania *et al.* 2003). The anticipated result was that when performing the task using a more natural method of interaction (stereo head tracking) would result in the participant remembering more of the objects. However this was not the case as the participants using the simpler, lower interaction fidelity based on mouse control performed significantly better. This led to the conclusion that an interface of high simulation fidelity such as head tracking does not always correspond to visually induced memory awareness states.

While these experiments are suitable for investigating the cognitive processes of search in relation to varying environmental conditions, they lack direct applications to the real world. In most real-world situations, the act of searching is not necessarily a purely passive visual one i.e. requiring no interaction with the environment. Real-world search tasks can require more than a simple visual inspection of an area as target objects are not usually hidden in plain sight; the environment has to have some form of interaction to locate the targets. Take, for example, human performance in bomb location/detection procedures. In such tasks, information about the target's location and surrounding environmental features is still needed as this information is critical to assess the threat of the device. Similarly, targets are usually buried or hidden, requiring the use of simple search aids such as probes and metal detectors to discover them.

This style of active search requires a greater understanding of the use of different types of technical search aids, their effect on recall and whether virtual counterparts can be created to accurately portray these effects. Therefore the main motivation for this initial experiment is to investigate the effect of recall of location and surrounding features using virtual recreations of these technical search aids and compare results to those of a similar task performed in the real world. By undergoing a direct comparison between both real and virtual tasks, a greater understanding of the effects of memory while using virtual technical aids can be understood.

Due to the findings of chapter three, Quest3D has been chosen as the most appropriate engine to investigate this as its user interface allows for rapid prototyping but still allows significant freedom to develop new methods of interaction and visual effects.

4.3 Previous Work

Previous work undertaken by the University of Birmingham's School of Electronic and Electrical Engineering has already begun to look at the effects of search and recall using technical aids (Houghton, Baber, and Knight 2009). This work was limited to a real-world experiment but the method and design of the experiment made it suitable for further investigation by including the development of a virtually simulated alternative and performing a direct comparative evaluation.

The real-world experiment involved fourteen undergraduate students and was concerned with a participant's general ability to search with different technical aids. None of the participants had any prior experience of the task or in the field of bomb detection.

There were 16 trays in all laid out in a four by four grid pattern (see figure 4.1). The trays were filled with sand, eight of which also had a piece of metal buried within it. In addition, each tray also had a marker sized three by three centimetres. Each marker had a unique coloured shape printed on it. The goal of the task was to search a number of trays, as many times as the participants wished, until they were satisfied that they had found all of the metal objects and could report both the location of the objects they had found and the identity of the markers for each object's location (in terms of colour and shape).

Participants were informed that they would be asked to recall one set of information per trial (i.e., either recall the location of the object or recall the identity of the markers). Participants only completed two searches (one for each form of recall). The reason for this was to minimise the chances that they would simply learn the layout of the trays and rely on recall rather than search.

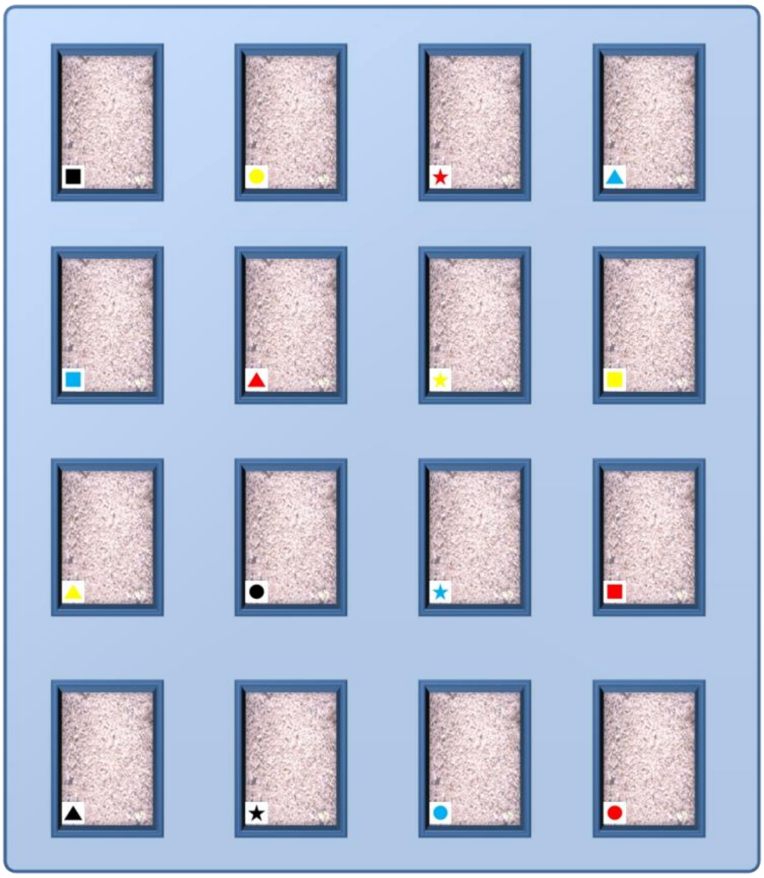


Figure 4.1: Search task experimental layout of 16 trays and their symbols.

There were two possible technical search aids which could be used: a metal detector or trowel. The participants were allocated one of the two conditions randomly. Participants were first shown two test trays in which they could grasp the basic understanding of how each technical aid should be used. Participants with the metal detector had to position the head of the device on the sand and would hear an audible tone when metal was present. Participants with the trowel probed and dug in the sand until they were satisfied that an object was, or was not, present. If a piece of metal was found, the participant was asked to bury it with sand again before moving on to the next tray. These practise trays provided both groups with an opportunity to familiarise themselves with the task prior to the study.

Figure 4.2: Participants performing search tasks in the real world using the two conditions, trowel (top) and metal detector (bottom) (Houghton, Baber et al. 2009).

Following this, the participants were led to the four by four grid of trays and asked to search each of the trays as many times as they felt necessary in order to be able to recall both the location of the metal and the associated symbols (figure 4.2). Once they were satisfied their task had been completed, they were led away from the grid and asked to recall either the metal position or the symbols that were in the trays containing metal. Recall tasks were counterbalanced across participants.

This real-world experiment concluded that recall of target location was the same across both conditions but participants performing the task using the trowel showed a statistically significant increase in the amount of correct shapes remembered. It was concluded that this may suggest that “identification is not well supported by the metal detector condition.” The goal of the virtual equivalent was to faithfully recreate the same real-world search task in order to investigate whether these conclusions still hold true and if there is any correlation between the uses of real or virtual technical aids.

4.4 Simulation Design

The implementation of the simulation, developed using Quest3D, is best described using a hierarchical description of the software. The key modules in figure 4.3, and their main functions, are now described in greater detail along with justifications for their inclusion.

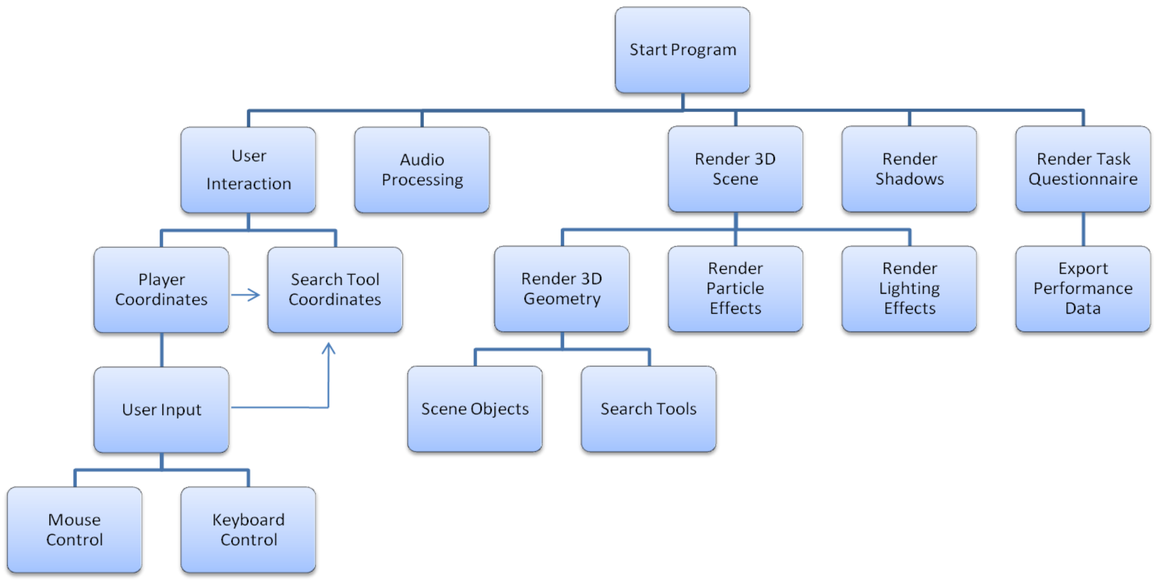


Figure 4.3: Shows a highly simplified version of the Quest3D nodal hierarchy for the search task simulation.

4.5 User Interaction

The program begins each cycle inspecting new changes from the mouse and keyboard to determine new coordinates for the player and the search tools. Unlike typical first person games, the tools that are used in this case are not locked to the player view (i.e. typically, where the player is looking is where their gun will shoot).

The control and movement of the trowel and metal detector need to work in a way that more closely represents the real-world task. When metal detectors are used in the real world participants typically move towards the area to search then remain stationary while they pass or scan the detector over the ground. The metal detector must be able to be controlled independently of the user's movements whilst the search processes are under way.

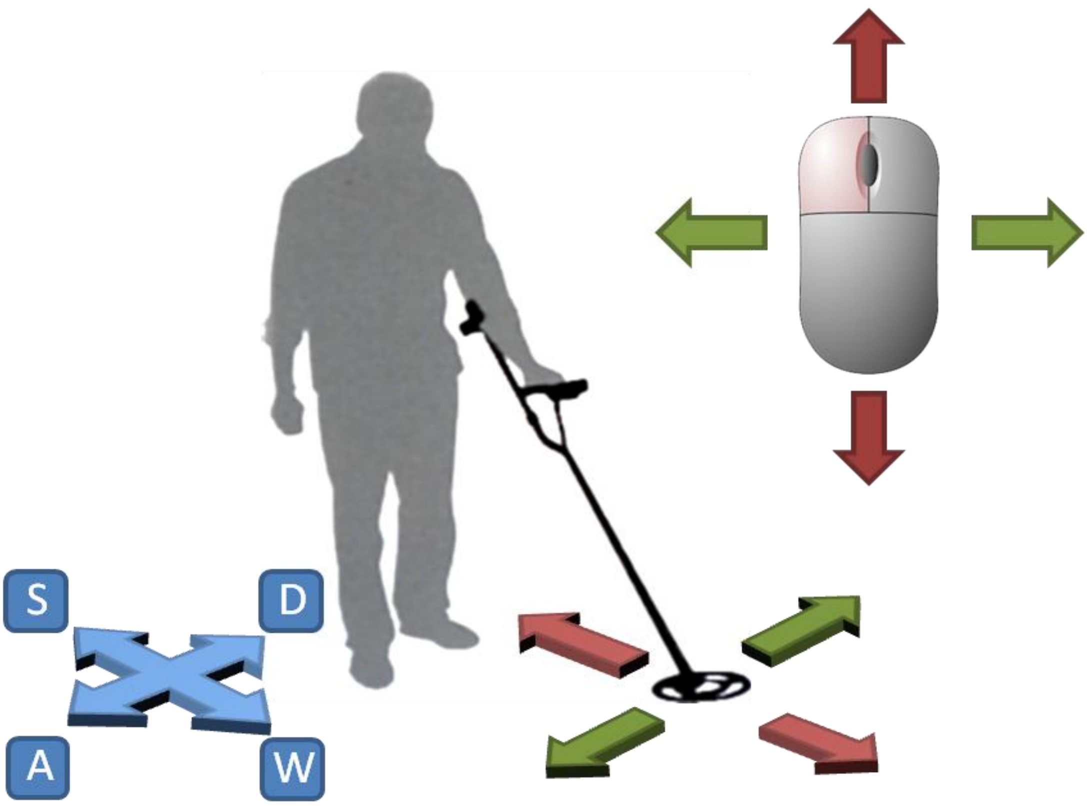


Figure 4.4:Illustrates the method of user interaction selected to control the metal detector technical aid.

Figure 4.4 shows the control method for the metal detector simulation. The standard "WASD" key configuration is used to control the player's movement and the Mouse X and Y delta change controls the metal detector. This allows the user to walk and scan at the same time. To allow for this function, the metal detector position is calculated by considering the player position and the offset of the metal detector, which in itself is based on the X and Y delta changes of the mouse.

The use of the trowel is slightly different. When people use the trowel in the real world they have to crouch down and physically interact with the soil or sand. Consequently, an additional crouch function was added to the simulation, changing the camera by positioning the user's viewpoint closer to the ground whilst searching.

4.6 Audio Processing

The metal detector sound signals must also be simulated. As the detector passes over a piece of inductive material it will begin to emit a signal, but the tone of that signal is not constant. Depending on the strength of the induction, the pitch of the tone increases.

The virtual metal detector constantly checks the distance between all of the eight targets. When the distance values drop below a certain level the audio signal can be heard. As the distance to target becomes smaller, the sound increases in volume and the pitch is increased. It reaches its highest volume and pitch when the target distance is zero.

4.7 3D Scene Render

This module in figure 4.3 renders the 3D geometry to the 2D screen based on the perspective of the user which has been determined based on the user input module. In addition to the static 3D geometry, an animated particle effect has been used to give an appropriate form of visual feedback to the user while searching using the trowel. While searching the trays with a trowel in the real world, sand and dust granules would be displaced, based on the trowel's interaction. In a virtual world simulation, rendering 10,000 particles is very intensive, as described in chapter three section nine. A simpler visual effect was implemented, as described in chapter three section nine, where multiple billboard textures are used to represent a group of particles. The particle effect was only active under three conditions - when the player was crouching, when the player was pressing the left mouse to perform the "inspect action", and when the trowel was positioned over a tray.

4.8 Render Shadows

While shadows are often used just to enhance the visual aesthetics of a virtual scene, some object shadows are often used as monocular distance cues (Schrater and Kersten 2000; Hanna, Cresswell, and Cuschieri 2002) in virtual environments. As it is important for the user to be able to perceive where the metal detector and trowel are in relation to the ground, it was believed that producing real-time shadows would be an important factor in establishing what tray the technical was over (Wanger, Ferwerda, and Greenberg 2002; Schrater and Kersten 2000). The two main methods for creating real time shadows are by creating a shadow map or by creating stencil shadows. Stencil shadows are geometry based, shadow maps are image based. (Figure 4.5).

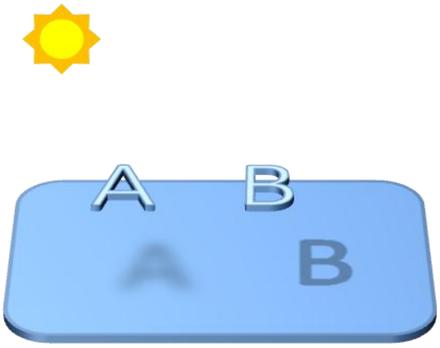


Figure 4.5: Illustration of the visual differences between shadow maps (A) and stencil shadowing (B).

While shadow maps can create a softer effect and are not dependent on geometry, they are not as accurate as a stencil shadows which was finally chosen for the simulation.

4.9 Data Export

The final module of the software (as shown in figure 4.3) records all relevant data, including completion times, player position and tray check count. It is then responsible for exporting these data as a series of arrays for later analysis within statistical packages. Each participant is given a unique identification number by the module, together with the time and date of the experiment, as well as the experimental condition undertaken. By continuously recording all of the positional data it is possible for the simulation to display a live image of the path taken by the participant (figure 4.6).

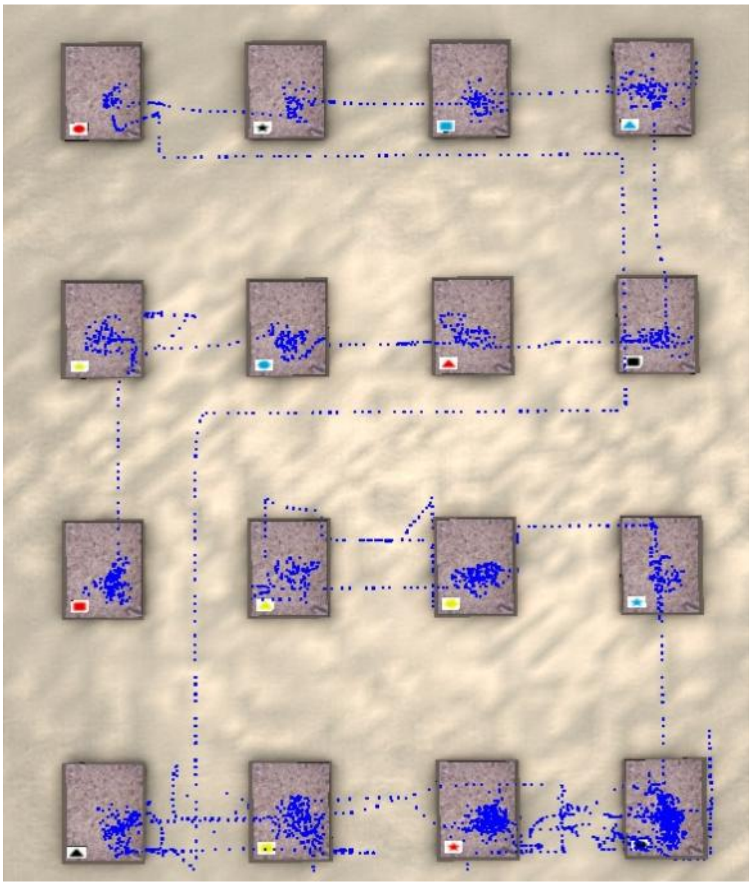


Figure 4.6: An overhead view produced by the simulation of the path taken by the participant during the search task.

4.10 Final Render

Figure 4.7 shows the final render of the simulation that the participant saw during the task. The surround walls and floors have been textured in such a way to best resemble the original real world room conditions. In addition to the 3D environment there is a semi transparent interface layer containing six buttons, the first set of buttons allow the assessor to start and stop the recording of data and return to the main menu. The second set of buttons allows for the recorded paths to be displayed onscreen for review proposes which can be displayed either as a static overlay or traced through in real-time. This interface can be set to invisible by pressing the 'Tab' key, allowing for a distraction free screen as the participant performs the task.

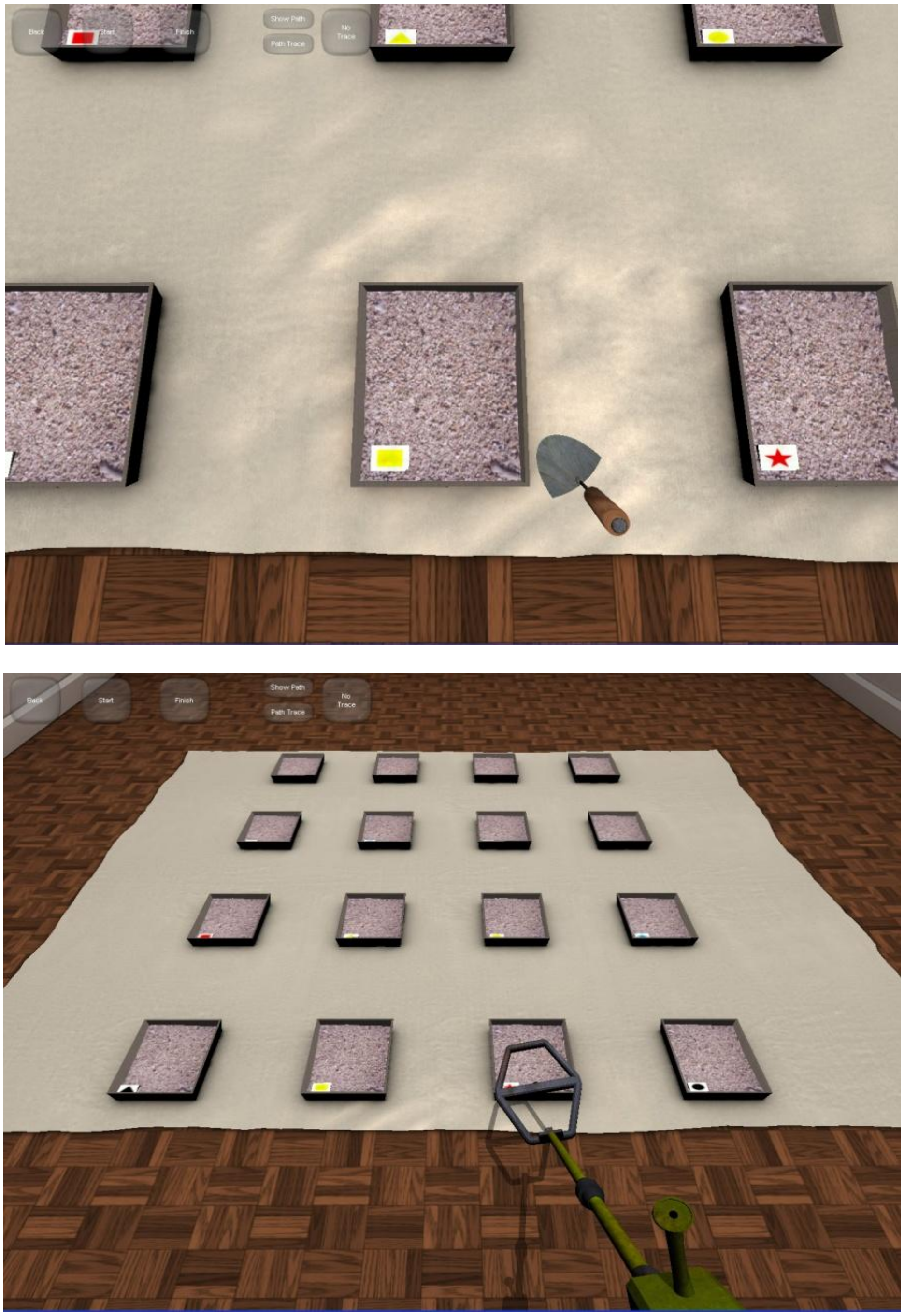


Figure 4.7: Images presented to the participants during the virtual search task using the two conditions, trowel (top) and metal detector (bottom).

4.11 Discussion

The chapter has outlined the development process of creating a virtual reconstruction of this real-world search task. Important consideration has been give to the key factor that could affect the results of the experiment. Firstly, the time in which it takes to manoeuvre around the tray and perform bending and crouching manoeuvres has been included in the simulation as it has been seen that recall from memory is affected by time (Page and Norris 1998; Baddeley and Hitch 1974; Cowan, Nugent, and Elliott 2000; Baddeley 1992). Secondly, real-time shadows were added to the simulation as their importance to judging distance and depth has been well documented (Hendrix and Barfield 1995; Hubona *et al.* 1999; Hu *et al.* 2002). Thirdly, the control system was developed to allow users to move the trowel and metal detector independently from the users' position. This was done as the small distance between the trays allowed participants in the real would to search up to three trays at a time by just stretching their arm rather than moving to each tray individually. Finally, great attention was spent in simulating the audio cues emitted from the metal detector as it has been well documented that audio cues effect memory (Dinh *et al.* 2002).

In this chapter we have seen how a previous search-based experiment undertaken in real-world conditions has yielded results that show there is a difference between recall performances based on the technical aids used. By comparing the real-world results with those from the virtual task experiments, we can test to see if the same cognitive processes of search using technical aids are evident during participants' performances in the virtual world.

Chapter Five

This chapter describes the method of experimentation used to study the relationship between virtual and real world search tasks, utilising the simulator developed in chapter Four. It will also present the results and draw conclusions from their findings.

5.1 Experimental Design

Obviously, to ensure the most accurate comparison between real and virtual search tasks the experiment must be conducted in the same way as the study described in chapter four, section three (Houghton, Baber, and Knight 2009).

5.2 Participants

The original data from the previous study (Houghton, Baber, and Knight 2009) was obtained, indicating 16 undergraduate students (14 male and 2 female) aged between 18 - 22 with an average age of 19.

For the virtual experiment, again, 16 subjects took part; they were all university engineering students aged between 18 – 22. The average age of the students was 19. Out of the twenty participants 12 were male and 4 female.

5.3 Method

To begin with, each participant was given an explanation of the task (see appendix B) and the aim of the research (as explained in chapter four, section three). They were then taken to the laptop and shown each of the training levels, one for the metal detector and one for the trowel. The control system was explained to them and they were asked to approach the two test trays in the virtual environment. They were then asked to search the two trays using each technical aid until they were happy with the mechanics of the control system and the goal of the task.

Once the training was complete each participant now began the main task. They were randomly allocated either the trowel or the metal detector. The simulation now showed the four by four grid of 16 trays and the participants were asked to search the trays until they were able to identify where all the metal objects had been hidden as well as the symbols present in each tray that contained metal. Following this, the screen was set to black and they were asked to recall either the location of objects or to identify the symbols. Next the search task was repeated and the other recall task performed. The order of technical aids used during the task and the type of recall was counterbalanced across participants. There were four conditions for the trial and the order in which they performed the different conditions was changed for each participant:

- Metal detector location search (ML)
- Metal detector symbol search (MS)
- Trowel location search (TL)
- Trowel symbol search (TS)

5.4 Subjective rating

In the original real task a subject rating system was used to assess the work load of the two conditions. This same rating system, the NASA –Task Load Index, was also used after the participants had completed each condition in the virtual equivalent in order to investigate whether the workload ratings still held true for virtual technical aids.

The NASA – TLX (Task Load Index) is a subjective workload assessment tool. It was presented by Hart & Staveland in 1988 (Hart and Staveland 1988) and is one of the most well known and used measures for self-reported workload. It comprises of six subscales that are used to form a multidimensional rating procedure. The scales used are mental demands, physical demands, temporal demands, own performance, effort and frustration. An overall work load score is determined from a weighted average of the six ratings. Figure 5.1 shows an example of the 'paper and pencil scales' for two of the subscale ratings (see appendix A).

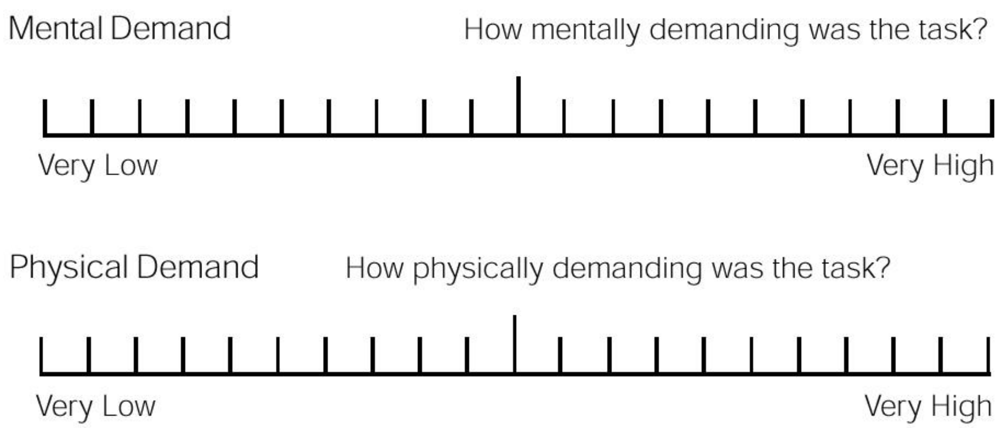


Figure 5.1: An illustration of two of the NASA TLX rating scales.

The participant was simply asked to indicate anywhere on the scale their assessment of the workload for the task. Each of the six subscales was split into 21 gradations rated ‘very high’ to ‘very low’. NASA TLX rating scales are typically used to assess the workload of human-machine systems such as aircraft cockpits, communications workstations and command and control systems. It has also been used, more recently, to assess the workload of 3D interfaces and virtual simulations (Ulinski *et al.* 2007; Ramloll and Mowat 2002).

In recent trials (Hart 2006) the NASA TLX has been used to form part of the assessment metrics to compare the workloads of real and simulated tasks. Positive correlation between workloads have been found in simulated surgical operations (Stefanidis *et al.* 2007), aircraft design (Selcon, Taylor, and Koritsas 1991) and interface control (Brickman *et al.* 2002).

5.5 Results

The results can be divided into four sections: recall performance, search activity, search paths and workload. The virtually simulated tasks results are directly compared to the results of the previous real world experiment. Two common statistical methods were used to analyse the recorded data: Pre-planned t-test comparisons were used for comparing the differences between individual recall and technical aid conditions and the Analysis Of Variance (ANOVA) procedure was used for all conditions.

The t-test is essentially a ratio of the difference between the group means and the variability of the groups. It can be thought of as a signal to noise ratio with the resultant difference between the means being affected by the variability or noise in the signal. For a significant difference to be found the difference in the means must rise above the overall group noise.

The ANalysis of VAriance (ANOVA) procedure is a set of statistical models dating back to the work of R.A.Fisher in 1925 (Fisher 1925). Essentially an ANOVA is a method of testing significance where more than two conditions are used. It is mainly used to avoid Type I errors (false positive) that are usually associated with performing multiple t-tests. An ANOVA compares the variance of the sample means with the within groups variance. If the resultant probability is less that the 0.05 the null hypothesis is rejected.

5.6 Recall performance

Recall performance was measured on two dimensions: location and shape. A Mixed analysis of variance was performed on the data. It examines device type (trowel or metal detector), recall type (location or shape) and condition (real or virtual).

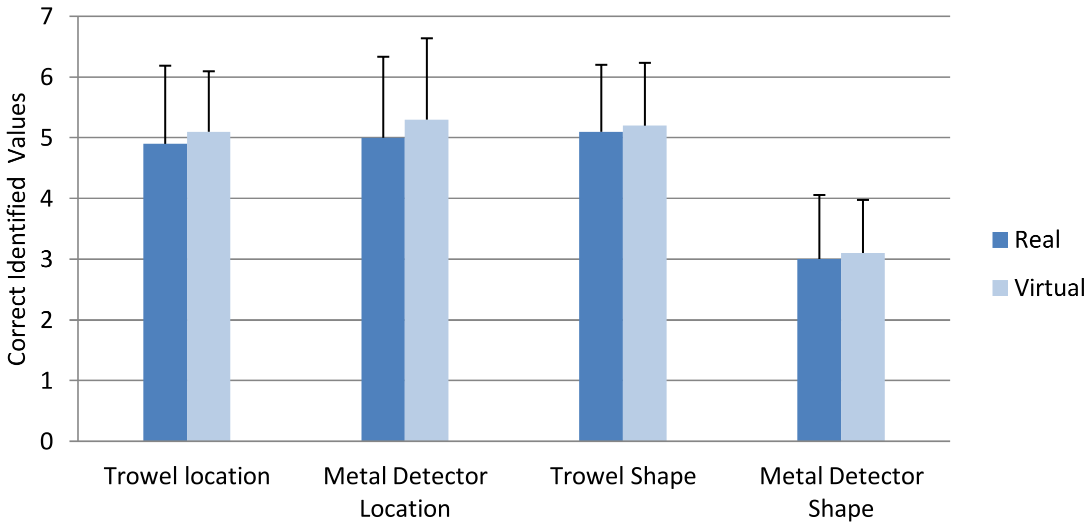


Figure 5.2: Recall performance for both conditions (trowel and metal detector) in the real and virtual task. Error bars indicate the standard deviation.

Figure 5.2 shows a difference between the recall of the shape and the device used for both real and virtual conditions. The Mixed ANOVA reveals that there was no significant effect between the real and virtual conditions [F (1,14) = 0.955, p=0.345] (between subjects effect) but there is a significant main effect between device type [F (1,14) = 13.092, p=0.003] (within subjects effect) and recall type [F (1,14) = 8.214, p=0.012]. There was only one significant interaction between recall and device [F (1,14) = 9.3, p=0.009]. We can examine this further by comparing the means of the location and shape recall conditions. The trowel had a mean **location** recall of 4.9(±1.4) items per trial for the real condition and 5.3(±1.1) for the virtual, while the metal detector condition had a mean **location** recall of 5 (± 1.5) real and 5.1(±1) virtual. In both cases this was not a significant difference, [t(15) = -0.882, p=0.407] for trowel and [t(15) = 0.228, p=0.826] for metal detector. The trowel had a mean **shape** recall of 5.1(±1.2) items per trial for the real condition and 5.2(±1) for the virtual, while the metal detector condition had a mean **shape** recall of 3(± 1) real and 3.1(±1) virtual. In both cases this was not a significant difference, [t(15) = -0.196, p=0.850] for trowel and [t(15) = 0.243, p=0.815] for metal detector. Indicating that recall of shape was worse when using the metal detector for both real and virtual conditions.

The key findings of the original study illustrated that recall of environmental features was worse when using the metal detector. This analysis has shown that the same effect occurs when performing the task virtually and that there is no significant difference in recall performance by participants and between real world and virtual conditions.

5.7 Search Activity

Data was also collected for overall time to complete the task and the number of checks per tray (figure 5.3).

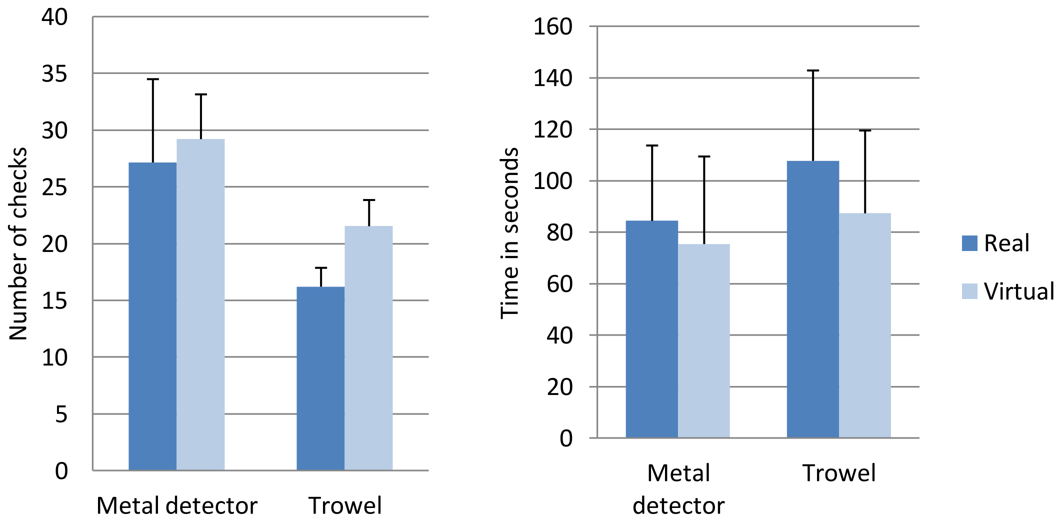


Figure 5.3: A Graph to show the number of checks during the task (left) and time taken to complete the task (right) for real and virtual conditions. Error bars indicate the standard deviation.

	Metal detector	Trowel
Real	27.17	16.22
ST DEV	7.34	1.66
Virtual	29.22	21.56
ST DEV	3.95	2.31

	Metal detector	Trowel
Real	82.2	108.5
ST DEV	33.96	31.5
Virtual	76.33	87.77
ST DEV	34.13	32.44

Table 5.1: Number of checks during the task (left) and time taken to complete the task (right) for real and virtual conditions.

By examining table 5.1 it can be seen that, participants in the trowel condition spent 108.5 (±31)s on the real task and 87.7 (±32.4)s. In the metal detector condition participants spent 82.2 (±33.9)s on the real task and 76.88 (±34)s virtual. A Mixed ANOVA was performed indicating a non significant main effect [F (1,30) = 2.188, p=0.150] for the real and virtual (between subjects) condition and a significant main effect [F (1,30) = 6.627, p=0.015] for device (within subjects). There were no significant interactions (device x real/virtual [F (1,30) = 1.041, p=0.316]). For the number of checks made, on average, participants in the trowel condition made 16.22 (± 1.6) real checks and 21.56 (±2.31)s virtual, whereas participants in the metal detector condition made 27.17 (± 7.34)s real checks and 29.22 (± 3.95)s virtual. The Mixed ANOVA indicates a significant main effect for both conditions: [F (1,30) = 7.28, p=0.011] for real and virtual (between subjects) as well as [F (1,30) = 8.768, p=0.006] for the device conditions (within subjects). Again there was no significant interaction (device x real/virtual [F (1,30) = 72.12, $p<0.0001$]).

Mean checks per tray **Metal detector** — Real world

1.4	**1.4**	**1.4**	**1.5**
1.4	**1.5**	**1.6**	**1.5**
1.4	**1.5**	**1.5**	**1.6**
1.4	**1.6**	**1.6**	**1.5**

Mean checks per tray **Trowel** — Real world

1	**0.9**	**1**	**1**
1	**1.1**	**1.1**	**0.9**
1.1	**1.1**	**0.9**	**1**
1.1	**1.1**	**1**	**0.9**

Mean checks per tray **Metal detector** — Virtual world

1.5	**1.5**	**1.4**	**1.5**
1.5	**1.5**	**1.6**	**1.5**
1.4	**1.5**	**1.5**	**1.6**
1.4	**1.6**	**1.5**	**1.5**

Mean checks per tray **Trowel** — Virtual world

0.9	**0.9**	**1.1**	**1**
1	**1.1**	**1**	**0.9**
1.1	**1**	**0.9**	**1**
1.1	**1.1**	**1**	**0.9**

Figure 5.4: Showing the Number of average checks per tray for both real and virtual conditions during the task. The blue shaded areas indicate the metal locations.

If we crudely approximate a search time per location as Mean time / Checks made, then we can see that the trowel condition resulted in around twice as much time spent per check, i.e., 6.7 (±2)s per check for trowel and 3(±1)s for the metal detector. For the virtual condition a difference is again around twice as much, 4.15 (±1.7)s per check for the trowel and 2.7 (±1.27)s for the metal detector.

The Mixed ANOVA indicates this difference between virtual and real condition check times was statistically significant [F (1,30) = 11.868, p=0.002] (between subjects) as well as finding the difference between the metal and trowel check times (within subjects) condition significant [F (1,30) = 62.17, p<0.001]. A significant interaction occurred between the condition x device [[F (1,30) = 11.675, p=0.002]. This indicates that the metal detector condition led to a higher number of checks than the trowel in both real and virtual conditions. However, as indicated by the figure 5.4, there did not seem to be any relation between the number of checks and the likelihood of finding metal in a tray.

The analysis aims to establish whether the amount of time to perform the task, or the number of required checks per tray changed when the task was performed in either the real world or virtually. The original study concluded that when using the metal detector the task was performed in a shorter time, this was reflected in the virtual counterpart. The same can be said for the number of checks per tray. Under the metal detector condition both real and virtual tasks showed an increased number of checks.

5.8 Search Paths

We can examine the search path each participant performed for both real and virtual conditions. Figure 5.5 shows the recorded output from the virtual search task simulator. This can be compared to the search paths recorded from the real world task.

Figure 5.5: Examples of typical search paths used during the virtual search task.

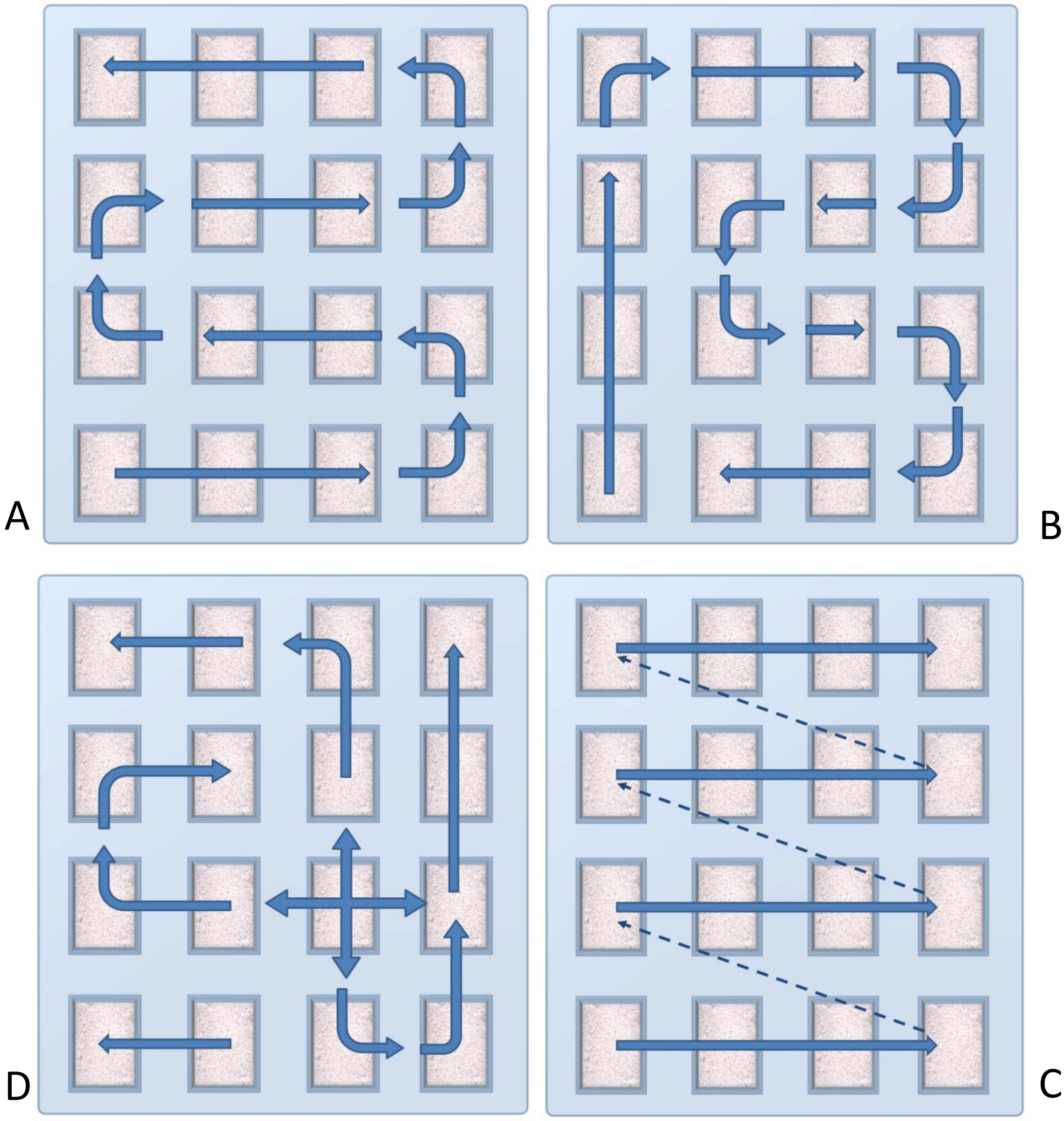


Figure 5.6: Illustrations of typical search paths used during the search tasks (with 'D' indicating a central start point).

Each search path can be roughly categorised into one of four identified styles seen in figure 5.6 (each search category may be translated rotated and inverted dependant on user). 68.8% of the virtual participants followed the 'C' style of search initially, 21.9% choose type 'B', 6.2% 'A' and 3.1% 'D'. There was no difference in the method of search path used with relation to trowel or metal detector. In the original real world experiment only types A and C were used, 15.6% and 84.4% respectively.

5.9 Subjective Rating of Workload

The results from the NASA TLX questionnaire were examined and compared to the results taken from the real world experiment. A three way mixed analysis of variance (workload dimension x device type x condition) was performed on the data and revealed a significant main effect of workload dimension [F (5,150) = 18.323, p<0.001]. A significant main effect in the device used was also found [F (1,30) = 32.583, p<0.001]. However, there was no significant main effect found between the real and virtual condition [F(1,30) = 1.530, p=0.226)]. No significant interactions were found for the data.

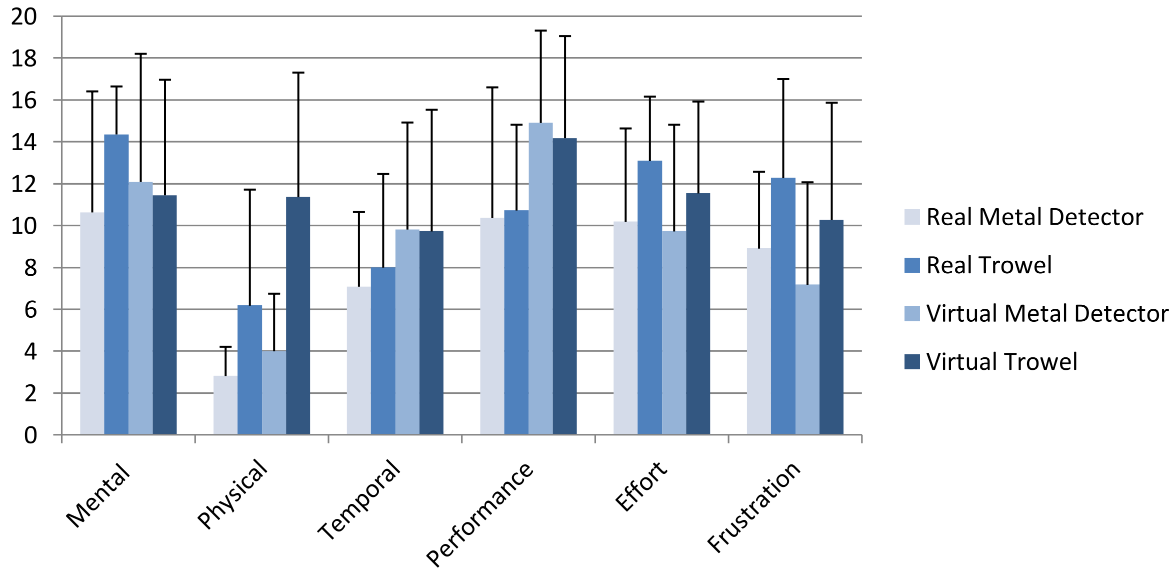


Figure 5.7: Mean NASA-TLX ratings across the two conditions (real world and virtual). Error bars indicate the standard deviation.

Inspection of figure 5.7 showed small differences on the Physical and Effort scales (which could be attributed to the need to bend and crouch when using the trowel, rather than simply walk around as with the metal detector), and also on the Mental and Frustration scales (which could imply that the search task was perceived as a little easier with the metal detector).

This direct comparison between the real world and virtual TLX rating scales was performed to determine whether there was any difference in how participants perceived the difficulty of each task dependent on condition (metal detector or trowel). The virtual simulation showed similar trends to that of the real world. In particular, even though the experiment was virtually performed, participants still believed that there was greater physical exertion needed when using the virtual trowel to the virtual metal detector.

5.10 Discussion

Previous studies have indicated that recall of surrounding environmental features is affected by the choice of technical search aids used during real-world search tasks (Houghton, Baber, and Knight 2009). It was suggested that this was most likely due to the additional time invested by operators during search than when using simpler aids such as a trowel. The goal of this experiment was to establish whether or not the same cognitive processes of memory and recall during search tasks using technical aids were present in a virtual environment.

It is worth noting at this point that there have been numerous studies (Eals and Silverman 1994; James and Kimura 1997; Herlitz, Nilsson, and Bäckman 1997) into the effect of recall performance and gender. It has been shown that female performance is significantly better than men in recall tasks. It is for this reason that male and female performance results are often presented individually, rather than grouped to together so any effect on the data can be factored out. Due to the small number of female participants (only four) there is insufficient data to reliably examine how they differ from their male counterparts in this particular test. The only method to minimise any gender difference within the test was to ensure that the four female participants were evenly distributed amongst the experimental conditions.

In both the real and virtual tasks, results from the "trowel condition" suggest better recall of shape (or location details). It was also seen that the participants' overall recall performance was not significantly different for either the virtual or real search conditions. This would indicate that the finding of the original study (increased recall for the "trowel condition" is due to the additional time invested in each search) still holds true for the virtual task.

The virtual condition led to a quicker search time. This is almost certainly attributable to the lack of physical exertion required to move between trays. Whilst there was notable effort in developing the virtual search task to ensure that – on average – the time to 'virtually' move between trays was programmed to be as realistic as possible, it is difficult to predict the speed of movement for all participants. It would be reasonable to assume that, during the real-world condition, participants' efforts may have slowed down towards the end of the task, due to exertion.

Also, whilst there was almost no physical exertion required during the virtual task (other than the activity of moving a mouse), the perceived demands surveyed in the NASA TLX form exhibited no significant difference. It would seem that the actions involved in using a virtual trowel were still perceived as requiring more physical exertion than that of the metal detector. The trowel did require more intensive movement of the mouse as the participants performed a digging motion. It could be concluded from this that participants were still acknowledging this increased demand even though it was performed with a few flicks of a wrist, rather than requiring them to crouch down physically and interact with the sand.

The success of the simulator to reproduce the same recall characteristics of the real world search process indicates that the simulator possessed the appropriate level of psychological fidelity as described by Miller (Miller and Research 1954). The fact that there was no significant difference between the way participants rated the workload demand scales further suggests that the design of the control interface for the virtual condition reflected the additional time and exertion required to use the trowel.

It should be noted that this experiment and the original published study (Houghton, Baber, and Knight 2009) had a relatively low number of participants. Both experiments had only fourteen participants. Whilst this made direct statistical comparison easier it may have affected the strength of any statistical conclusions made. This may be of particular concern when the result from a statistical test is very close to the value of significance (p=0.05) or when performing subjective tests that show large standard deviation such as the Rating of Workload (figure 5.7).

Apart from the rating of workload the remaining data was recorded through objective means and the participants repeated the task for all conditions. In addition, none of the results presented here were close to a border-line conclusion regarding significance. It is also most likely that there was an increased level of accuracy in data due to the nature in which the information collected. The original study required an observer to record timing data, paths used and the number of tray checks by hand, this was all recorded directly by the simulation software in the work presented here. It would be reasonable to state that the statistical results and conclusions presented here are at least as strong as those presented in the original study.

Chapter Six

This chapter describes the development of an ROV simulator based on a real world environment at the National Maine Aquarium in Plymouth. While initially it was used for a transfer training study for ROV piloting skills it was also used as a basis for further research into technical aid usage during search and navigation activates.

6.1 Introduction

While many ROV simulators exist (Agba 2002; Fabekovic, Eskinja, and Vukic 2008) there is a tendency for them to lack the visual realism of ROV flight. It is currently unclear what the effect of visual fidelity has on the use of technical aids. It has already been shown from the results of chapter five that the mental processes of search and recall when using technical aids is still relevant in the virtual domain. Before the use of technical aids can investigated in underwater search tasks an ROV simulator with suitable psychological fidelity must be created and evaluated. It is hypothesised that a serious games approach could very well produce a cost-effective simulation test bed with which to assess the effects on a pilot's ability to control the ROV and his or her ability to conduct underwater searches.

Only in recent years has the technology, in terms of hardware and software, made it possible to create a visually accurate model of underwater environments and the visual distortion seen by the ROV operator.

Due to the increasing ability of graphics hardware, several methods for rendering underwater scenes have been proposed (Iwasaki, Dobashi, and Nishita 2002; Cerezo and Serón 2001). However, each of these methods only paint half the picture when an underwater world is viewed through the lens of a camera and not a virtual fictitious perfect lens.

Typical methods of rendering underwater scenes in current commercial simulators include the following four important visual effects:

- Shafts of light ("God rays")
- Caustics (the envelope of light rays reflected or refracted by a curved surface)
- Shadows
- Colour

It is proposed that this is still insufficient to recreate an accurate underwater visual simulation. Four important additional affects have often been cited as important visual factors to consider during underwater simulation (Reinhard 2006; Auster, Stewart, and Sprunk 2002):

- High dynamic range of the camera (the effect of adjusting to new lighting levels)
- Distortions due to lens quality and features
- Camera Focus
- Ambient shadows

There are few transfer of training studies focusing on piloting ROVs, particularly "Micro ROVs" (underwater vehicles less that ~60cm), as access to appropriate hardware is generally extremely limited. The aim of the present experiment is to evaluate a games engine approach to ROV simulation and to assess the effects of increasing visual fidelity with the final goal of developing a simulator suitable for evaluating technical search aid usage.

6.2 Simulated ROV Development

The remainder of this chapter describes the development of a virtual ROV simulator using the COTS development tool Quest3D. The simulator not only aims to recreate an accurate model of control but also a very specific environment, namely the National Marine Aquarium's (NMA) "ExplorOcean" exhibit. ExplorOcean is one of the NMA's primary visitor attractions. It is an interactive exhibit that allows members of the public to experience piloting a real Micro ROV. Currently this exhibit is one of only a small handful of facilities across the globe where the general public has such regular open access to ROVs and it is believed that the NMA's ROVs have had more usage within this facility than any other system in operation. The exhibit not only allows visitors to use the ROVs freely (within the constraints of the NMA's daily schedule), but also requires them to pilot the ROVs though a purpose-built underwater "obstacle course". This allows for a unique opportunity to develop a virtual recreation of the ROV and to perform a skills transfer study in a highly controllable way.

6.3 ROV Environment

The real ROV ExplorOcean exhibit features a short obstacle course which members of the public can attempt to complete. (Figure 6.1) Usually, an expert can finish the course in around one minute; however, an untrained member of the public typically takes three minutes.

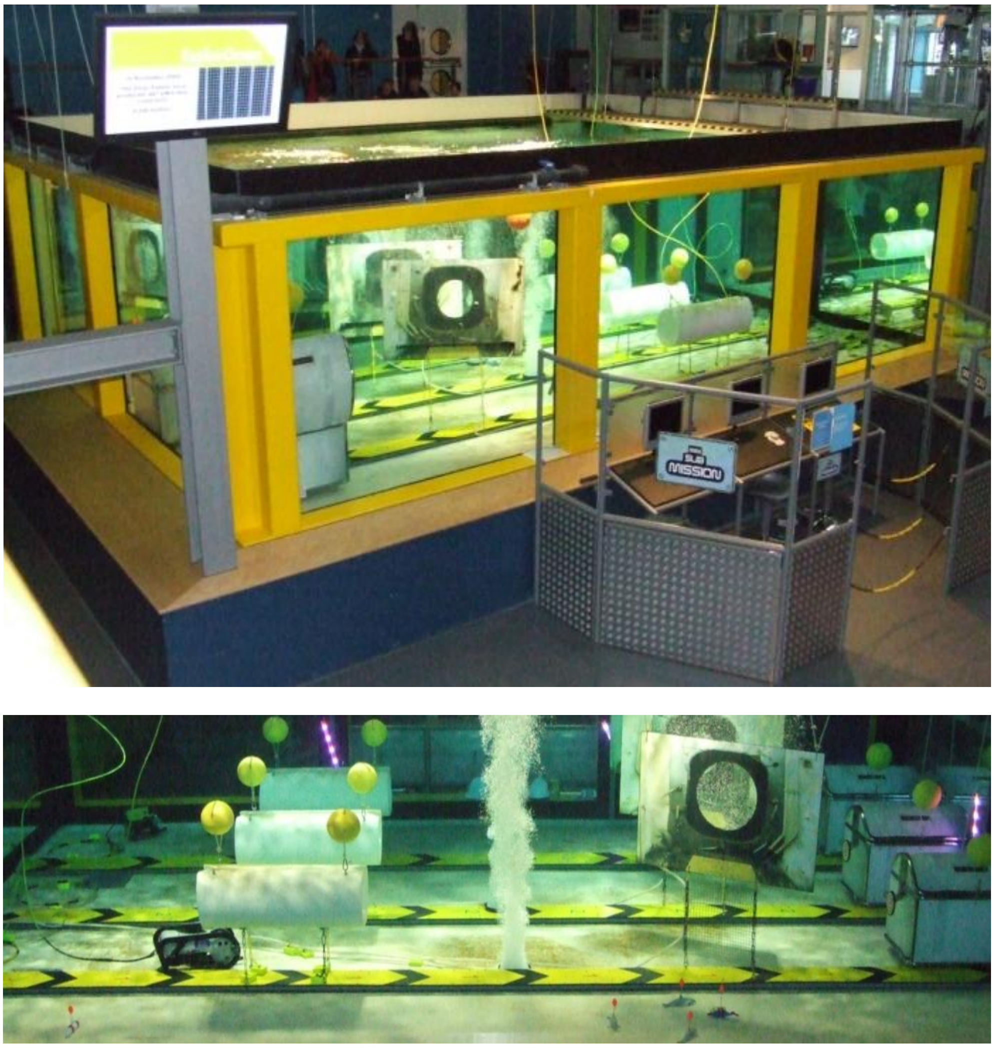


Figure 6.1: An image of the National Marine Aquarium's ExplorOcean exhibit.

The obstacle course involves the participant manoeuvring an ROV though a large tube, a porthole, a bubble stream and then they are required to hit a small target that opens a treasure chest. After landing inside the chest they then return to the start. From a simulation perspective, the entire course can be developed using varying degrees of visual fidelity, therefore allowing a further investigation into the appropriate level of detail that may be suitable for this specific type of skills training.

6.4 ROV Selection

As the aquarium possesses three different types of 'Micro ROVs' a decision had to be made as to which would be the most suitable for the investigation. The ROVs used in the aquarium are; the LBV150[6], AC-ROV[7] and VideoRay[8] (Figure 6.2). All three can be used for commercial and scientific applications and are well known to industry.

Figure 6.2: Images of the micro ROVs present at the National Marine Aquarium. From left to right LBV15[6], AC-Rov[7], and VideoRay[8].

One of these ROVs had to be chosen for simulation and the key criteria for selection were ease of use and reliability. After a discussion with the aquarium staff it was believed that the VideoRay was typically found to be easier to control as well as having a control interface which was relatively easy to reproduce. AC-ROV can move laterally. which can make it difficult at first to grasp its movement and the LBV is the largest of all three, making it harder to manoeuvre around the obstacle course. It was also stated that not only was the VideoRay generally more reliable than the other two but the aquarium had a spare. It was clear, therefore, that the VideoRay was the most appropriate ROV to simulate.

[6] www.soundocean.com
[7] www.ac-cess.com
[8] www.videoray.com

6.5 Developing the Flight Dynamics

An underwater vehicle can typically be thought of as a vehicle possessing six degrees of freedom (DOF) in motion, much like that of an aircraft. In other words, the ROV can move in six possible ways as illustrated in Figure 6.3. The ROV is essentially a free body in space with mass and inertia that can be acted upon by a series of forces. Sway, Heave, Surge related to the ROV's positional movement in each axis while Pitch, Yaw and Roll relate to the ROV orientation in each axis.

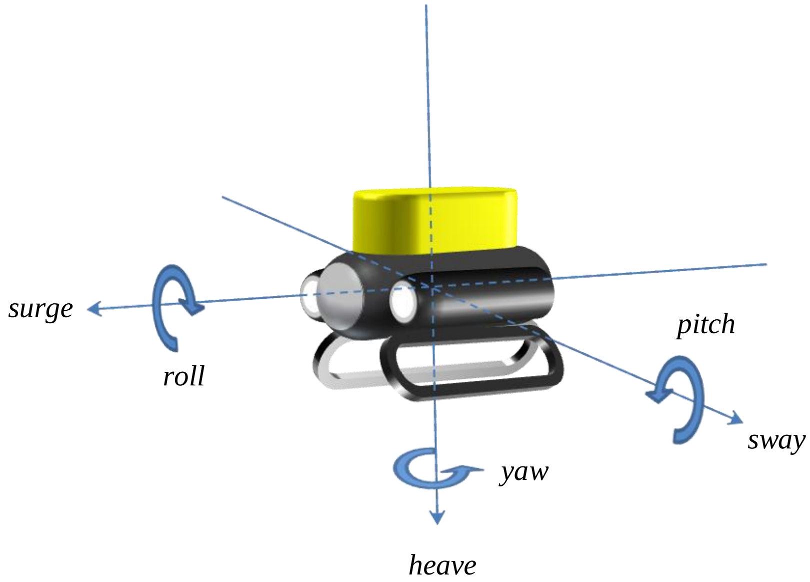


Figure 6.3: VideoRay's six degrees of freedom: roll, pitch, yaw, heave surge and sway.

It has been proposed, and generally accepted, that a non-linear 6-dof model can be used to simulate the movement of an underwater vehicle (Herman 2009). Through a series of integrations, it is possible to calculate the ROV's linear and angular velocities. In theory, by inputting the exact engine specification (see appendix F), density of the medium (water) and fictional forces, an accurate model of the ROV dynamics could be constructed. However, while it is possible to acquire the exact engine specifications of the VideoRay, it is not possible to determine the deterioration of the engine components over time. This could lead to different turning and acceleration speeds. The NMA ROV is probably one of the most heavily used in the world. Since 2002 it has been in use at the ExplorOcean exhibit running for several hours a day, every day of the year. Just like a car that has travelled over a hundred thousand miles, the engine performance is no longer going to be the same as it was when it left the factory floor. In addition, the NMA actually installed third-party propellers to the port and starboard thrusters, each of which only has two blades as opposed to the normal three fitted by the factory. This modification was undertaken to slow the ROV down to ensure that it would not be damaged by running at full speed into the ExplorOcean tank wall. Due to these factors the manufacturer's specifications for thruster performance would bear little resemblance to the NMA ROV's acceleration and drag coefficients. Previous work has been undertaken to record the specific characteristics of the VideoRay ROV. However, the ROV under investigation in that study was brand new and tested under well-controlled laboratory conditions (Wang and Clark 2006).

The best way to determine the true ROV's technical performance was by direct experimentation and to compare it to the previously recorded laboratory tests to get a better understanding of how the vehicle performs in its current state. A simple set of manoeuvres were repeatedly performed that were designed to show the current state of the ROV's acceleration and drag coefficients.

Forward thrust (surge), rotation (sway) and depth (heave) velocity can be simply examined by measuring the time it takes for the ROV to cover a certain distance at maximum input power (joystick press fully forward – figure 6.3). However the response from the electric motors is not linear. To build up a better picture of the ROV acceleration response curve, four input values from the joystick were tested, 10%, 50%, 75% and 100% of thrust. These results can be used to generate a simple response curve for the ROV.

In order to capture the ROV speed and acceleration characteristics, a video-based object tracking setup was used. A camera was placed facing the ExplorOcean exhibit, the large glass panels of the tank allowed for the ROV movements to be recorded which could then be analysed by video tracking software (Adobe After Effects). The resultant output of a series of position points over time were converted to acceleration curves by examining the distance travelled between each point. Two of the test input level response curves are shown in figure 6.4; they show a non-linear acceleration which levels out once it reaches the maximum ROV velocity given the thrust input level. The same can be done for lateral movement of the ROV which can be used to evaluate the virtual ROV model.

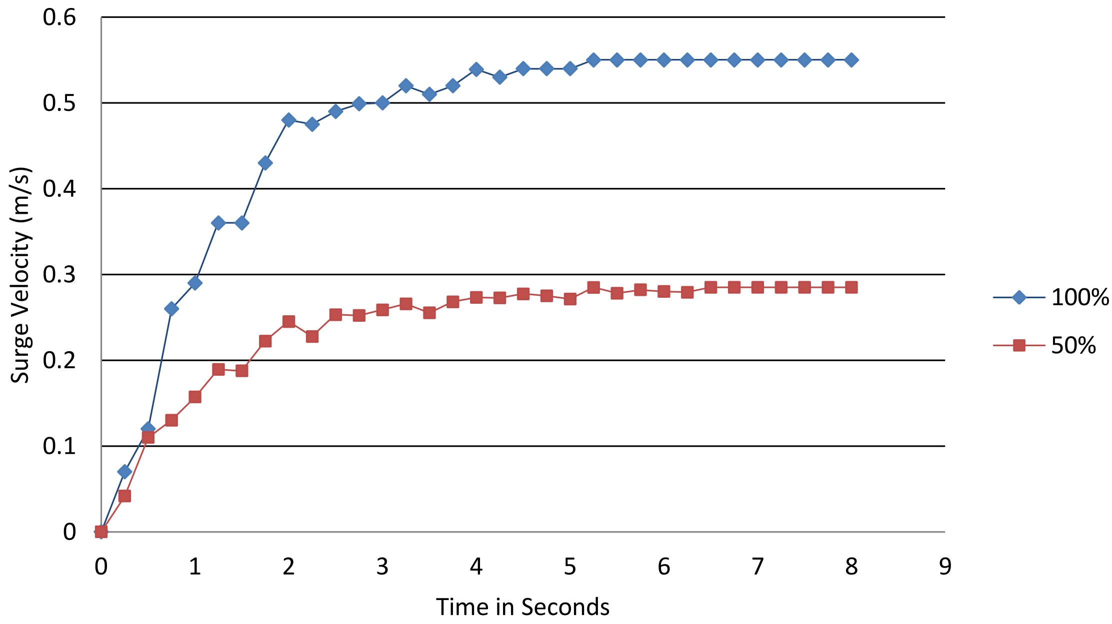


Figure 6.4: A graph showing the recorded velocity over time when the ROV is piloted in a forward/surge direction for two thruster input levels.

It is worth noting that the recorded signal is not a smooth curve, this was most likely due to two main factors: - the error associated with image-based capturing techniques and the additional variability of the effect of the umbilical cable.

A physics engine was used within Quest 3D to enable the virtual ROV to match the behaviour of the real world ROV. The Open Dynamics Engine (ODE) was used; it is a rigid body dynamics simulation engine with collision detection. Broadly speaking the ODE can compute interactions between bodies in free space dependent on forces applied to those bodies (figure 6.5).

The ROV is represented by a dynamic ridged body in free space. Within the simulation there are only two forces that can affect that body, the user motor control and the additional upward thrust of the bubble stream. The simulation must also take account of any collisions that occur. This is performed by assigning a collision shape to the ROV ridged body, a simple capsule (cylinder capped with hemispheres) is used to enclose and describe the physical limits of the ROV. The simulation must also know the shape of the surrounding ExplorOcean exhibit to produce suitable collision responses when interactions occur with the body of the tank and the obstacles.

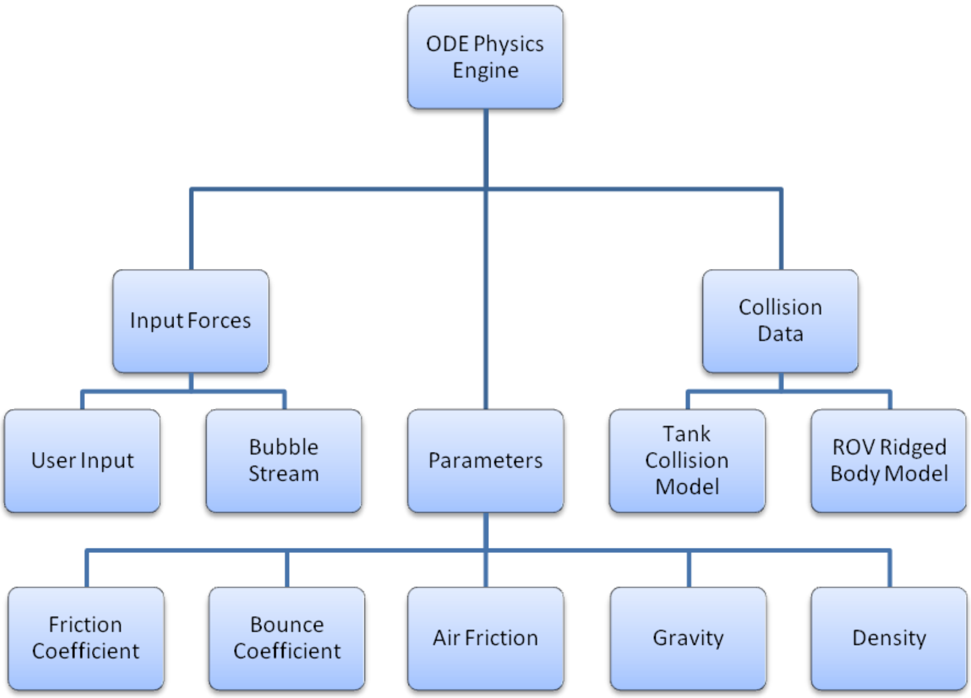


Figure 6.5: A simplified hierarchical structure of the ROV flight dynamics module for the virtual ExplorOcean simulation.

In addition to knowledge of the effecting forces and collisions models of the environment, there are a number of user-definable variables that can be set to shape the nature of the interactions. The gravity is simply a constant force that can be applied to any dynamic objects in the scene; usually this is set to 9.81 m/s^2 pointing down, just as with the real world gravity. However, when underwater, an object is also affected greatly by the upward thrust of the water. In most cases, ROVs are set to be neutrally buoyant, in other words with no forces applied from the engines the ROV will stay at a static depth. For this reason the gravity was simply set to zero to simulate a neutrally buoyant ROV.

The air friction defines how a moving free body will reduce is speed if it remain unaffected by another force. In the case of underwater flight this value can be increased to produce a more realistic response due to the resistance of the water.

The sliding friction is the force resisting the relative motion of solid surfaces sliding against each other. As the ROV is naturally buoyant there is no force bringing it in constant contact with the ground. For this reason the default value of 0.5 was used. The bounce value, which defines the amount of rebound that will occur when two objects meet, as with the level of friction, was left at its default value (0.1).

Density is the most important setting when attempting to match the acceleration response curves of the ROV. When a force is applied to a high density object its acceleration is far slower than that of a less dense object. The effect of density is plotted in figure 6.6 showing three different acceleration curves for three different densities (40, 80, 120). The value of density described by the ODE system is a dimensionless value that does not directly relate to the ROV's real world density but can be adjusted to ensure that the virtual ROV matches the response curve of its real world counterpart. The recorded response curve of the virtual ROV has a perfectly smooth curve as it is not affect by the errors in image capture and there is no umbilical cord simulated here.

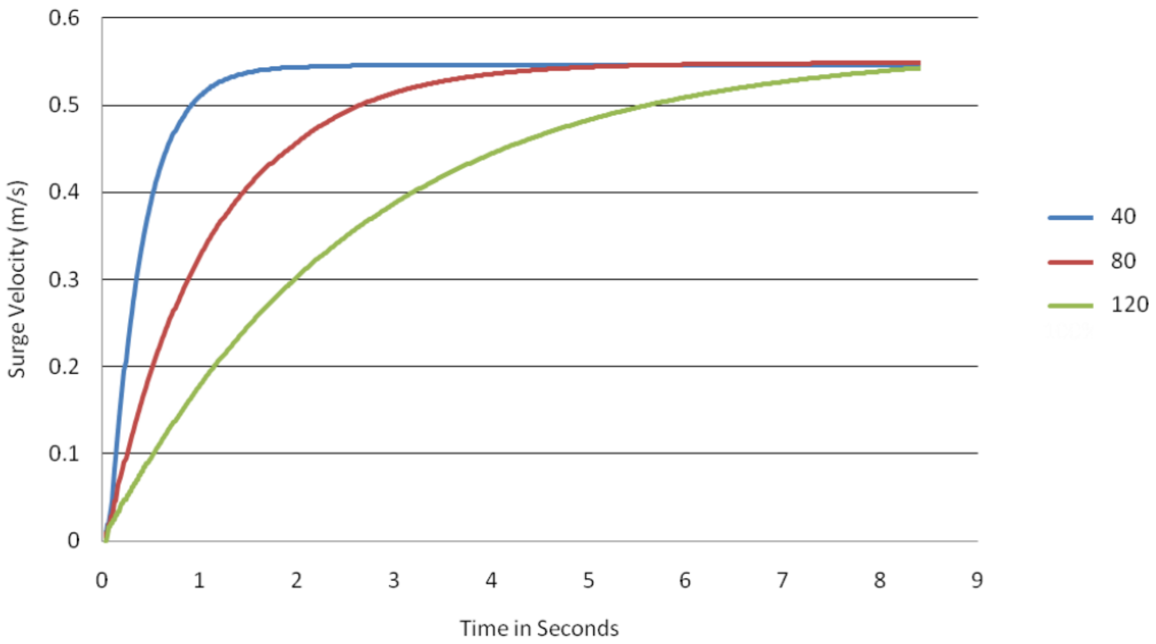


Figure 6.6: A graph showing the recorded velocity over time when the Virtual ROV is piloted in a forward/surge direction for three different ODE density levels.

By comparing the record real world response curve to that of the recorded virtual ROV, the most suitable value for density can be selected. In figure 6.7 we can see the real world curve in blue (at full forward thrust) and the red line indicates the virtual ROV response curve with the density of 88, with this setting there is a close fit to the real world ROV. The same response curves tuning can also be applied to the lateral movement of the ROV seen in figure 6.8.

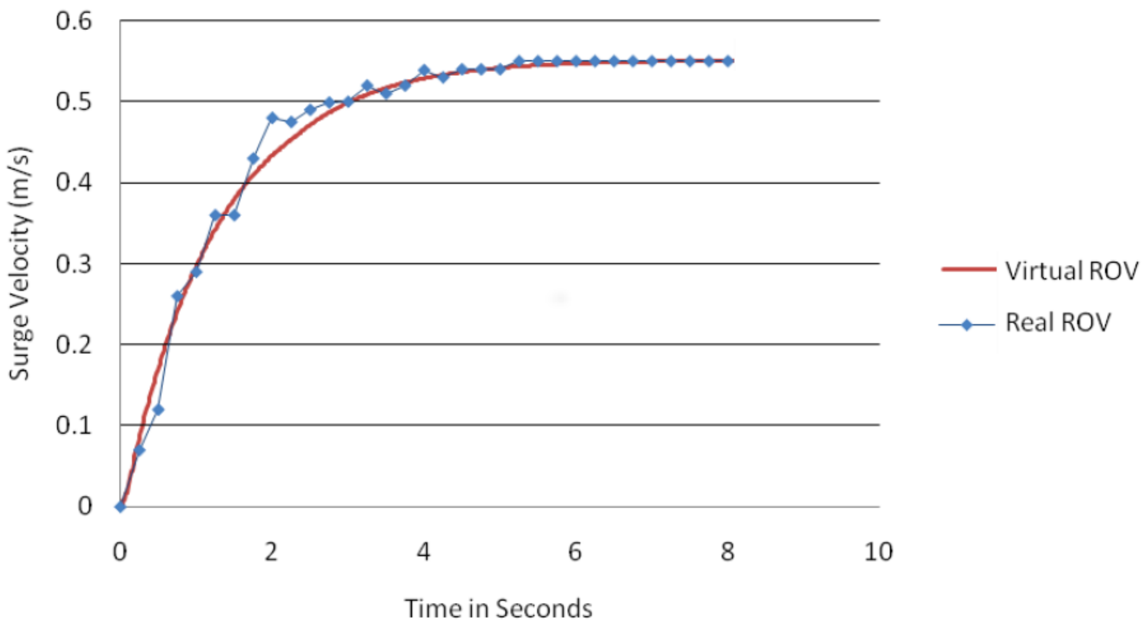


Figure 6.7: A graph showing the recorded surge velocity of the real world ROV and the virtual ROV when the density has been scaled to match.

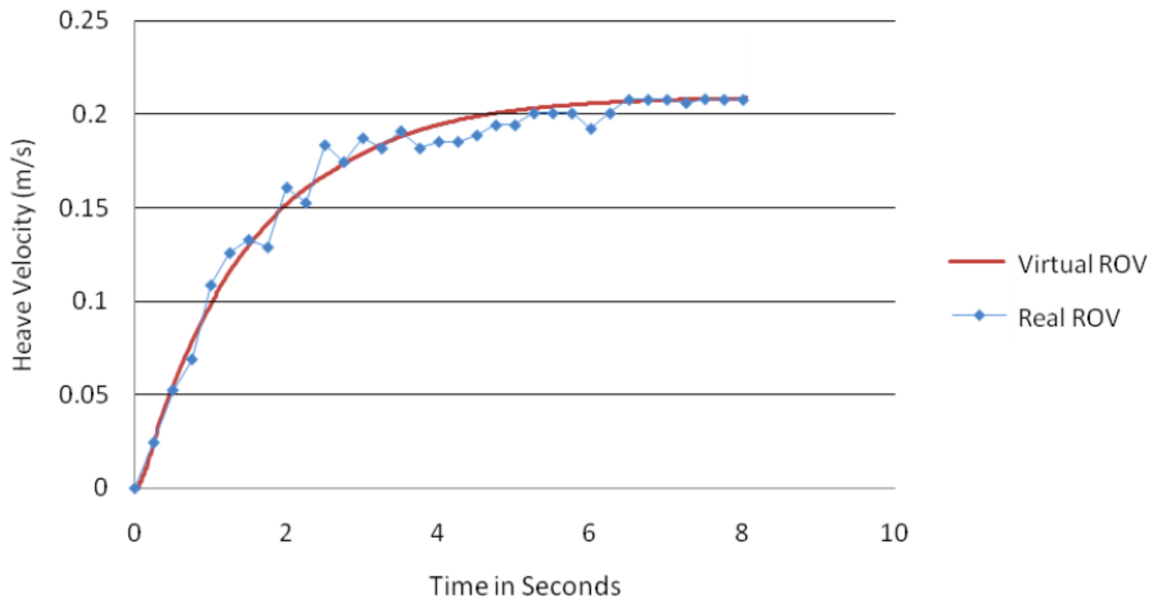


Figure 6.8: A graph showing the recorded lateral/heave velocity of the real world ROV and the virtual ROV when the ODE density has been scaled to match.

In addition to the ROV's acceleration characteristics, the rate at which the ROV decelerates due to the resistance of the water must also be investigated. Once again, computer modelling may be able to provide a very accurate model of the effects of the water resistance from a new ROV, but the NMA's model has been modified over the years and the tank water has a certain concentration of chlorine (instead of being salt-water). The best method to determine the current ROV deceleration characteristics is to once again to perform a direct experiment.

To establish the forward, or surge deceleration, the ROV was flown at maximum thrust for two metres, after which the thrust input was dropped to zero and the deceleration was recorded. This value could then be compared with the virtual ROV. By altering the ODE "air friction" setting the deceleration could be shaped to appropriately fit the record curve of the real world ROV. A setting of 0.755 was used in the final simulation (figure 6.9).

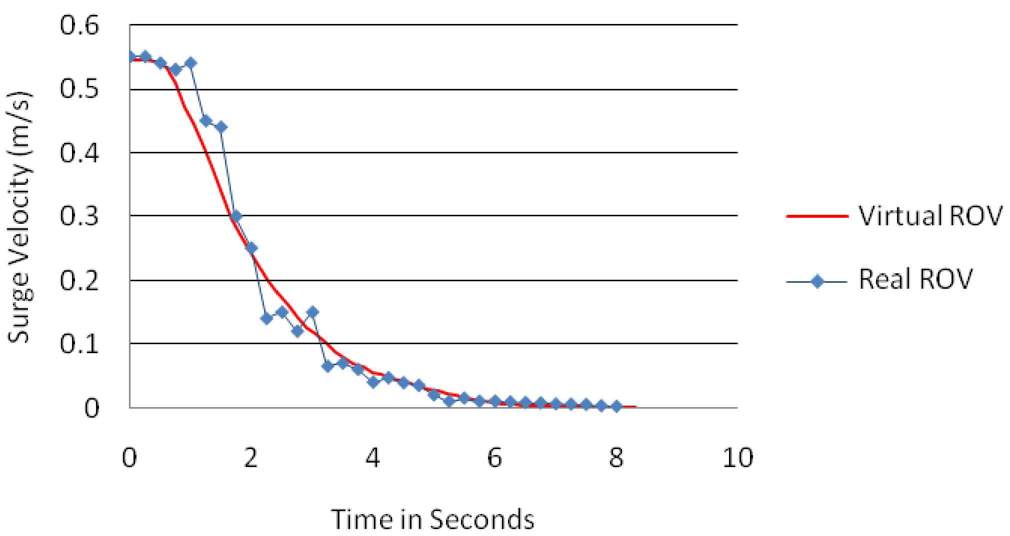


Figure 6.9: A graph showing the recorded velocity of the real world ROV and the virtual ROV during deceleration when the ODE 'air friction' has been set at 0.755.

With the ODE air friction, density and thruster power set to closely approximate the real ROV characteristics, the final test was to see how real world ROV pilots performed using the virtual ROV. Three of the regular aquarium ROV pilots performed the course four times for both real and virtual conditions (after briefly being allowed to familiarise themselves with the virtual ROV).

Participant	Trial Run	1	2	3	4	Mean
1	Real ROV	54.2	55.3	48	49.5	51.75
	Virtual ROV	68.8	61	52.7	48	57.63
2	Real ROV	74.9	65.3	50.2	51.2	60.4
	Virtual ROV	58.2	63.8	52.1	50.8	56.22
3	Real ROV	58.7	54.7	58.2	49.6	55.3
	Virtual ROV	88.2	57.4	52	47.1	58.75
4	Real ROV	54.2	55.5	51.5	52.6	53.45
	Virtual ROV	75.4	59	52.1	49.7	59.05

Table 6.1: A table showing the recorded course times in seconds from the experienced ROV pilots in both real and virtual conditions.

On inspection of their course times (Table 6.1) we see that the virtual ROV times are very close to that of the real world. Initially the aquarium pilots take longer to complete the virtual course but by the end they are in fact performing slightly better on the virtual condition. It is possible that this may be due the fact that there is no additional umbilical cord slowing down the ROV.

While the developed COTS based game engine simulation of the ROV may not be as accurate as one derived from first principles, the resulting response curves and course completion times serve to validate the developed simulation sufficiently for the use in further experimentation.

6.6 Developing the Environment

As previously stated, the environment is a recreation of the ExplorOcean exhibit at the National Marine Aquarium. Even though the tank itself is in reality a small part of the exhibit, the entire surrounding area had to be reconstructed as the pilot can clearly see the external aspects of the exhibit which, studies have shown (Vinson 1999), may be used to help navigate the vehicle by providing strong external features and cues. (Figure 6.10) The exhibit tanks span over two floors of the aquarium and the facility has a seated theatre-style viewing gallery on one side. The pilot’s control booth is opposite this gallery.

Figure 6.10: Image of the real ExplorOcean exhibit.

The environment was created using 3D Studio Max 9 and additional textures were created using Adobe Photoshop 8. Figure 6.11 broadly shows the stages that are undertaken when recreating a real-world environment.

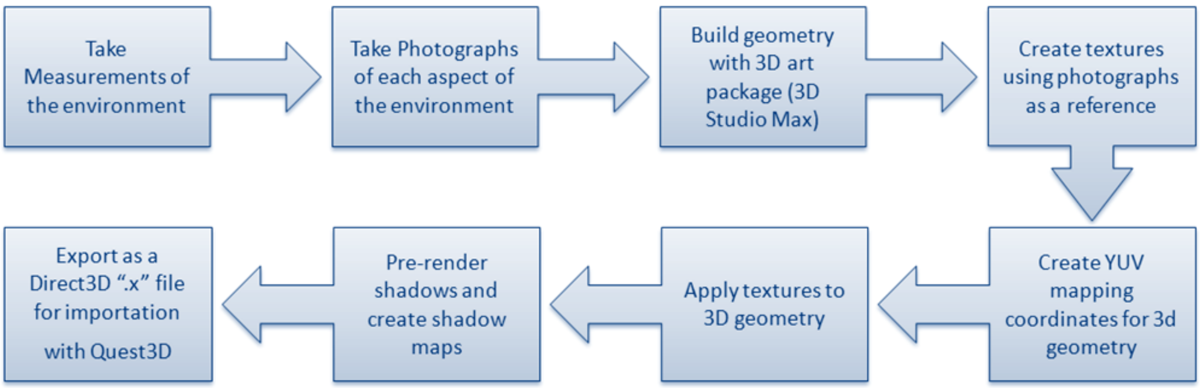


Figure 6.11: A flow diagram of the process of creating a 3D environment suitable for import into Quest 3D.

6.7 Software Structure

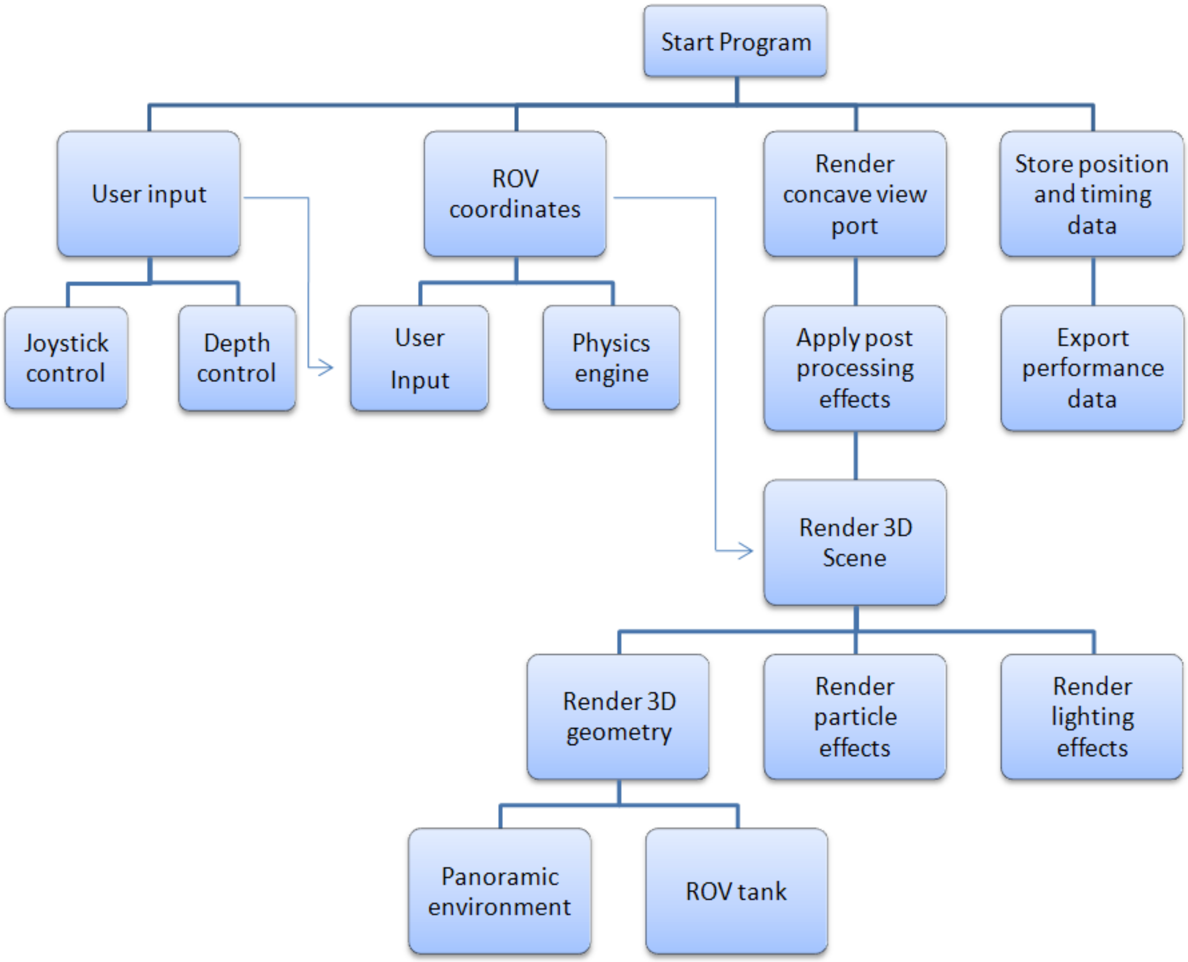


Figure 6.12: A simplified hierarchical structure virtual ExplorOcean simulation.

The system shown in figure 6.12 briefly comprises four core components. Firstly, the user input system, which receives input signals from the replica control panel's joystick and depth controller. Secondly, the physics engine, which governs the movement of the ROV. The control signals received from the user do not directly affect the ROV coordinates or movement. They are first applied to the physics module to take into account the current flight dynamics, inertial forces and local collision objects. Thirdly, the 3D scene rendering components that examine the user's current viewpoint, position and render elements from the 3D Geometry database according to the current position and orientation of the ROV. Finally, the post-processing component which re-colourises and adds a soft focusing filter to the virtual environment was added. Each one of these modules is described in detail below.

6.8 User Input

Figure 6.13 shows input signals from the depth dial and movement joystick are sampled by the user input module regularly, via the USB interface. Two further operations must be applied to these raw signals within the input module. Firstly, suitable "dead zones" for the depth and movement controls must be set. As the joystick is based on an analogue system, receiving an absolute zero value from the inputs can be rare. Usually there are small input signals (around 5% of the total possible range) being sent, even when at rest. Dead zones simply create a small area around the zero or at rest point (typically +/-10%) where small input values are ignored. Secondly, an overall calibration of the sensitivity of the control inputs is needed. An additional on-screen GUI can be displayed with adjustment sliders that affect the sensitivity of both control inputs and then the calibrated values are sent to the calibration control. Typically calibration only needs to be performed once when the software and replica control system is installed for the first time.

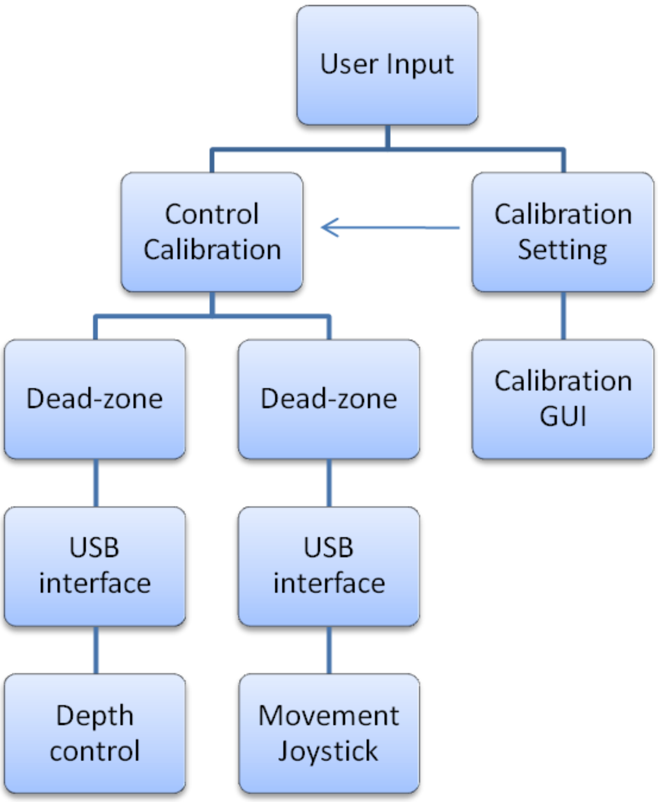


Figure 6.13: A simplified hierarchical structure of the user input module for the virtual ExplorOcean simulation.

6.9 ROV Flight Dynamics

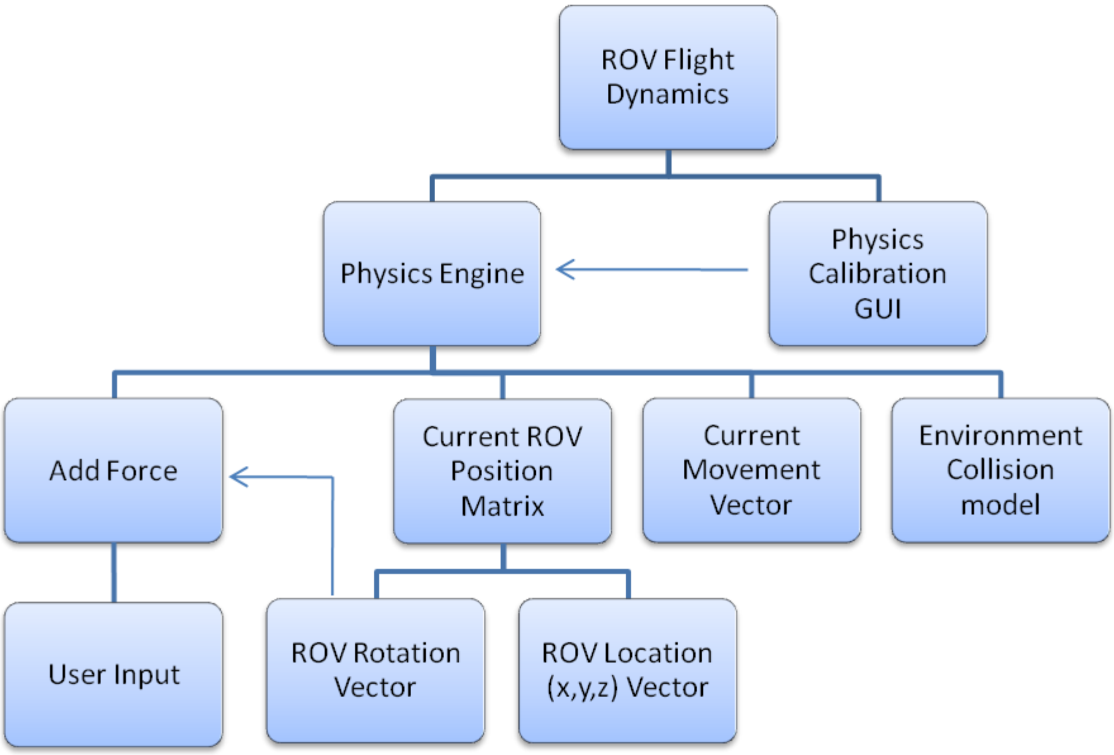


Figure 6.14: A simplified hierarchical structure of the ROV flight dynamics module for the virtual ExplorOcean simulation.

To determine the position and orientation of the ROV, the physics engine looks at four key inputs: the current position of the ROV, the current movement vector, the environmental collision model and the new values from the user's input module. The current position of the ROV is a matrix comprising two vectors. The world location vector, which stores the current x,y,z coordinates in 3D space, and the rotation vector, which stores the current orientation. (Figure 6.14)

When the user pushes forward on the joystick, a new force is applied to the ROV physics object. The strength of the force is based on the amount the user pushes forward on the joystick multiplied by the calculated engine thrust response curve of the ROV. The result is then multiplied by the ROV's current movement vector and inertial response curve. The direction depends on the current orientation of the ROV. As with the user input module there is a calibration GUI. The speed, acceleration, water friction and inertial values can be manually fine tuned if necessary.

6.10 3D Scene Render

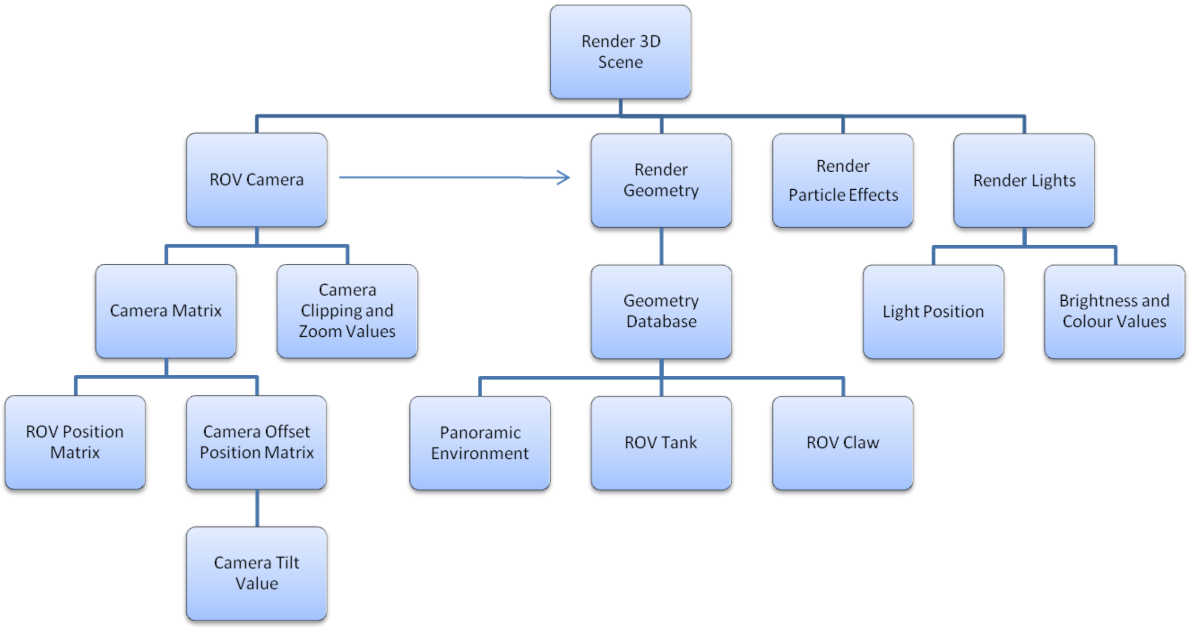


Figure 6.15: A simplified hierarchical structure of the 3D rendering module for the virtual ExplorOcean simulation.

The first step in rendering the 3D geometry is to determine the viewpoint of the user (figure 6.15). The ROV's camera is not centred to the middle of the model, it should be offset to be correctly place at the front of the ROV. This is calculated by the multiplication the current ROV position matrix and the camera offset.

The camera offset is required as the camera is not located directly at the centre of the ROV (it is offset a small amount forward). There is also a requirement to set some basic camera properties before the geometry can be rendered, such as the far and near clipping planes (clipping planes are used in 3D graphics in order to prevent the renderer from calculating surfaces at far distance from the viewer). Once the camera properties have been set, the module can now render the 3D geometry based on that perspective. The main features that are rendered are the environment and the obstacle course. Note that the ROV is not rendered itself from the user's perspective as it is not possible to see it. The only part which can be viewed is the small manipulator or claw. This is an important visual feature to render as the claw gives a clear indication as to the location of the base of the ROV and was shown to be used heavily for judging depth when entering the obstacle course tubes.

In addition to the static geometry, a number of particle effects are rendered. They attempt to simulate the bubble streams present in the real ExplorOcean tank. Rather than attempting to render hundreds of thousands of bubbles, the particle effects use a few hundred semi transparent textures of bubbles to simulate the effect. The simulated bubble particles are emitted from the bottom of the tank and given a negative gravity force to push them towards the surface. The bubble planes are also given a slow rotation as they ascend to give a more varied motion. Around 1000 bubble planes are rendered. Each of the planes have a semi transparent texture with around 10 bubbles represented on it. This gives the approximate visual look of 10,000 bubble particles overall which is sufficient to give the illusion of a dense bubble field.

The final object to be rendered is the water plane; this sits atop the water and simulates the water surface layer. The surface has to take into account three key properties of water, reflection, refraction and distortion. The reflection effect is created by taking an image from the previous frame and inverting it and adding a distortion to it (Figure 6.16).

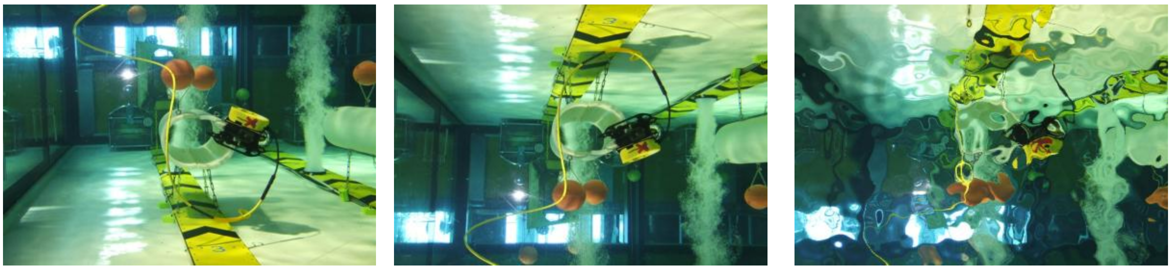


Figure 6.16: Images showing the process of creating a water surface effect. The original image (left) is inverted (middle) then distorted to generate as rippling water effect (right).

6.11 View Port Render and Post Processing

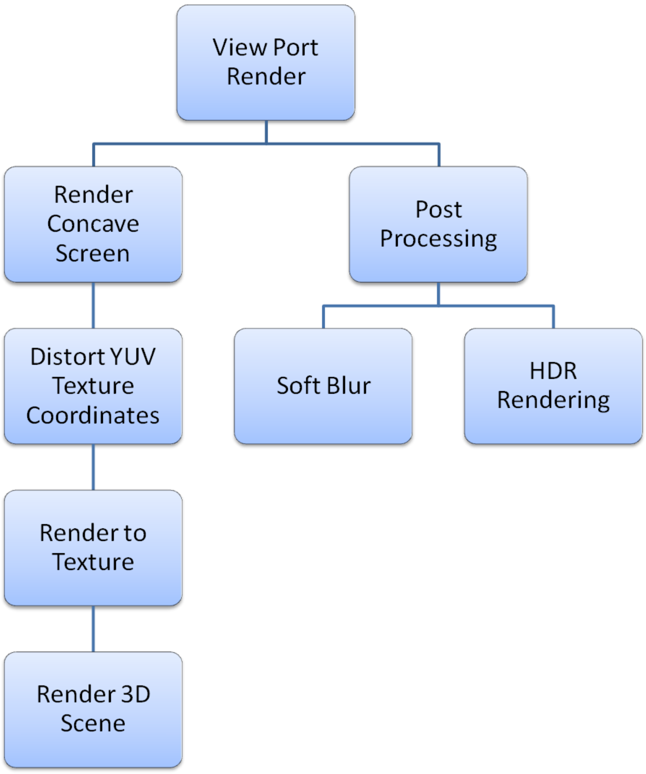


Figure 6.17: A simplified hierarchical structure of the View Port rendering module for the virtual ExplorOcean simulation.

Once the 3D elements are rendered, additional 2D rendering techniques are used to produce additional video effects such as blurring and distortion of the camera. (Figure 6.17) To allow for 2D image processing the 3D scene render module output is captured as a 2D image, or "rendered to texture". In this form the 3D scene can effectively have 2D image manipulations performed on it. There are three main operations performed. Firstly, the lens effect of the real ROV camera is simulated. As the ROV camera is behind a glass dome, light is distorted causing a fish eye effect. This effect can be relatively easily approximated by altering the texture coordinates (the values of how textures are mapped to a 3D object) to distort in the same way. (Figure 6.18)

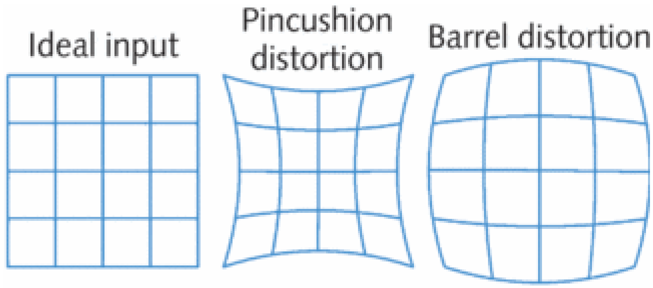


Figure 6.18: An illustration of the texture warping effect used to simulate the distortions of the ROV camera dome.

The second 2D operation is a soft blur filter; this is achieved by a simple blur operation on each of the image's pixels (Equation 6.1). Essentially each pixel's colour/brightness is affected by its surrounding neighbours blurring the differences between them. The following matrix was used as the convolution mask.

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 2 & 1 \\ 1 & 3 & 9 & 3 & 1 \\ 1 & 2 & 3 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} \tag{6.1}$$

The final image operation is the high dynamic range processing. The exposure sensitivity of ROV cameras is very high, they auto adjust the exposure depending on the current lighting levels. This often leads to sudden increase and over saturation of light when going from a dark to light environment. This same effect can be replicated in the virtual world by using high dynamic range lighting. This is the rendering of computer graphics scenes by using lighting calculations done in a larger dynamic range. This allows preservation of details that may be lost due to limiting contrast ratios.

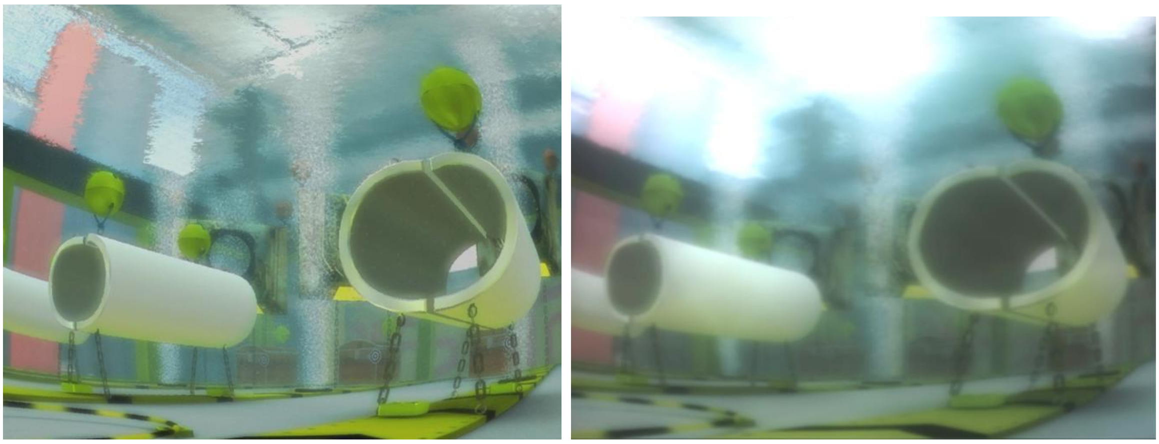


Figure 6.19: Images of the virtual ExplorOcean tank with (left) and without high dynamic range rendering (right).

Figure 6.19 show the visual difference when high dynamic range lighting is used. The ExplorOcean's stage lighting causes an over saturation of light that spreads across the surface of the water. This is a common inbuilt visual effect in Quest3D.

6.12 Data Collection

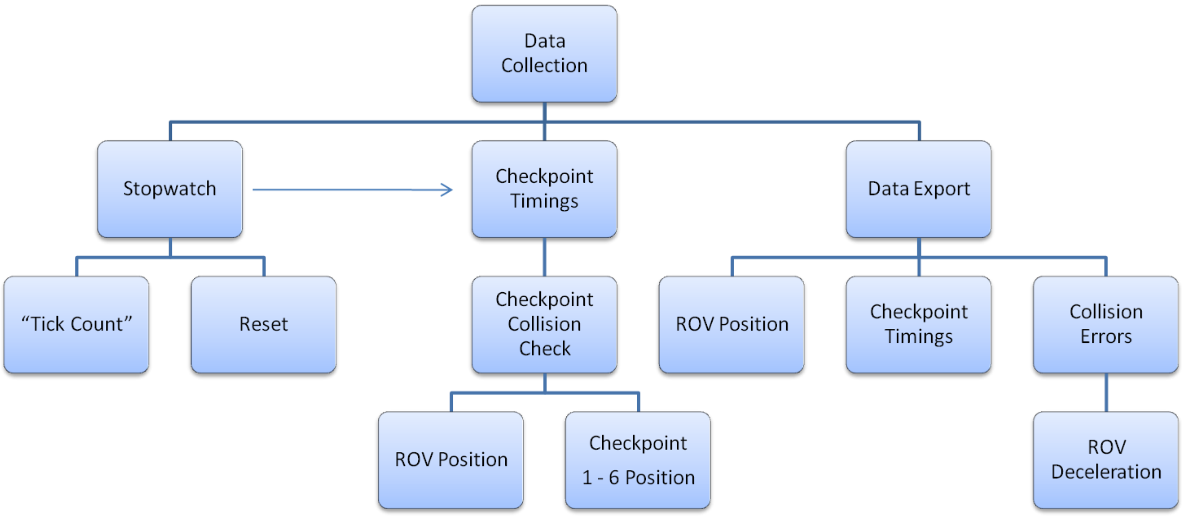


Figure 6.20: A simplified hierarchical structure of the data collection module for the virtual ExplorOcean simulation.

The data collection module, seen in figure 6.20, handles the statistical aspects of the simulation, such as timing, check points, storing the ROV's position and data export. The primary goal of this module is to record the user's performance.

The stopwatch is slightly more complicated than it appears. The program's "tick count" has to be used. The tick count is simply a value that represents the amount of time in between each call. In other words it is the time required to produce one frame. Without this the program stopwatch would record time differently depending on the processing speed of the PC it is running on.

Throughout the simulated underwater course, six checkpoints are placed and each time the user passes through one the time is recorded. While the timing information is useful for analysis, the primary reason for the checkpoints is to make sure the user cannot cheat by avoiding the obstacles. Each checkpoint has an invisible marker collision box that can be compared to the ROV's position. Once a collision is detected, the checkpoint is marked as complete. Each checkpoint will only become active if the previous one has been successfully reached.

Once the course is complete the data must be exported for later analysis. The system outputs two text files which are saved to the computer's hard disk. The first text file is the checkpoint timing data and the collision count. The collision is logged if the ROV's velocity changes suddenly; this indicates that the ROV has had a sudden change in movement that could only occur if it had collided with a solid object. The only collision which is not recorded is when the user hits the target to open the chest; this is not seen as an error collision as it is required to complete the course.

The seconded exported file is the ROV positional data. Every $1/10^{th}$ of a second, the position data of the ROV is added to an array. Once the course is complete the array is exported as a comma separated variable (CSV) text file. These data can be used to recreate the movements of the ROV and directly compare said movements to previous course runs. Often this type of position logging is referred to as an "after action review".

6.13 Control Panel Reconstruction

To portray the real ROV flight experience accurately, the physical control system must also be modelled. The real ROV control system consists of two key elements, the depth dial and the movement joystick (Figure 6.21).

.

Figure 6.21: Image of the VideoRay control system highlighting the depth and movement controls.

The other controls on the real control panel, such as lighting, camera tilt and manipulator control are ignored. All of these additional functions are irrelevant to the task of negotiating the obstacle course.

The first stage of development was to investigate current off-the-shelf game controllers to determine if they could be used for the basis of an ROV controller. A standard PC joystick, typically used for flight simulation, is the closest control system, but ROV controls are still fundamentally different from flight simulation.

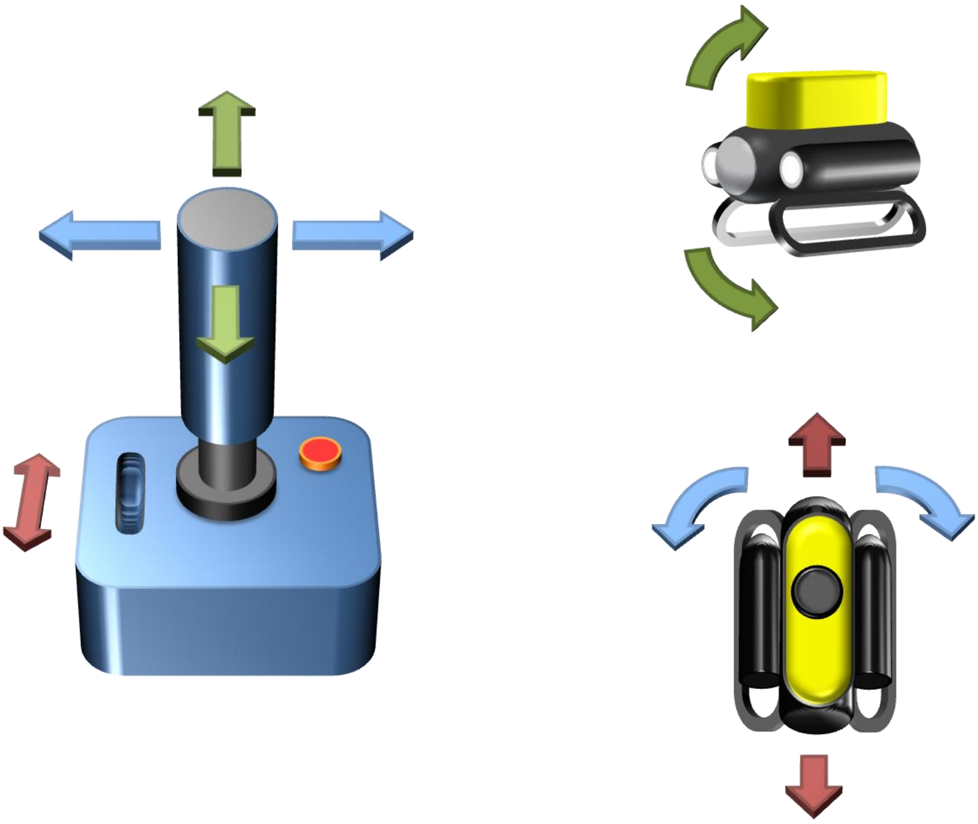


Figure 6.22: An illustration of the typical joystick axis to movement relationship. Green indicates ROV pitch, red indicates forward thrust and blue indicates yaw.

Figure 6.22 shows the relationship between the joystick and ROV if using typical aircraft flight control. As the user pushes left and right the aircraft turns left or right (yaw), which is the same as ROV flight. However, if the user pushes forward and backward the aircraft tilts up and down (pitch). This is very different from typical ROV flight. The throttle control (which is the scroll wheel or small lever on the side) is used to control the aircraft's thrust. Again this is not the same for ROV flight.

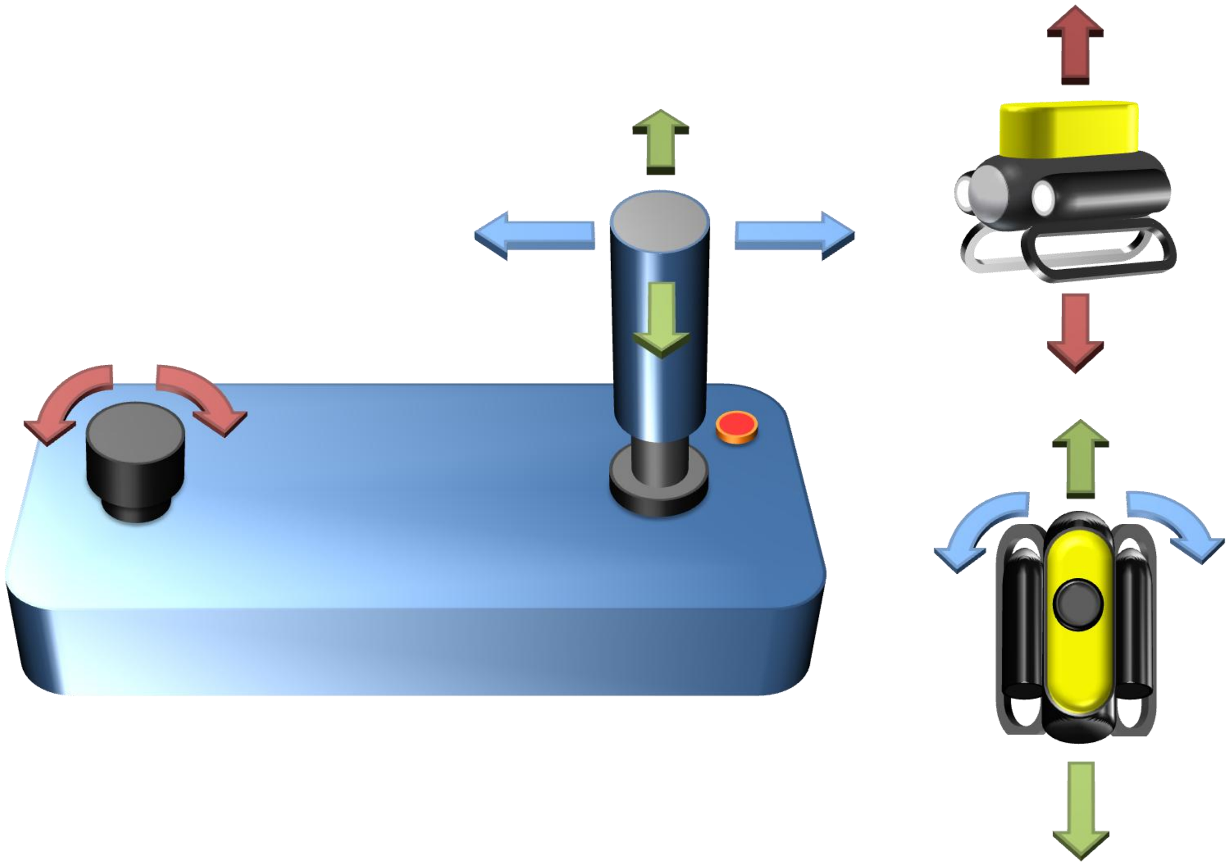


Figure 6.23: An illustration of the corrected joystick axis to movement relationship. Green indicates forward thrust, red indicates depth and blue indicates yaw.

Figure 6.23 shows the correct relationship between control and movement for a typical ROV. The main difference is that the thrust lever is replaced with a dial that controls the ROV depth by twisting it left or right. Pushing the joystick forward and backward now works the thruster control, while the ROV yaw control stays the same.

It is clear that significant changes must be made to a standard off-the-shelf joystick to create a suitably realistic approximation of the real control system. The most reliable way of allowing a control device to interface with the software was to use a standard USB interface. A relatively low-cost joystick was acquired. The joystick possessed simple analogue control, five buttons and a trust lever. The only components required for the ROV implementation were the USB interface and the joystick housing. A joystick basically consists of two variable resistors which sit in a plastic housing, with two shafts connecting it to the central stick. The two shafts also have small springs attached to ensure that the stick returns to the central position when it is released. As the central stick is moved, the resistance value is changed allowing for the joystick controller to calculate an x and y position. In addition, another variable resistor is used to set the thrust control. The thruster control resistor was changed in order to give a wider range of movement and was re-housed 26 cm away (the same distance between the joystick and depth control of the real controller). The joystick resistors were replaced with higher quality potentiometers and the main joystick return spring was also changed as the ROV control has a much lighter feel to it. The final replica control unit can be seen in figure 6.24.

Figure 6.24: An image of the final replica control system.

6.14 Final Simulator Rendering

Figures 6.25 and 6.26 show the final output from the ExplorOcean simulator in both low and high fidelity. To create the low fidelity environment several of the visual effects were omitted from the high fidelity environment. Those effects were real-time shadows, camera lens distortions, image blurring and high dynamic range lighting.

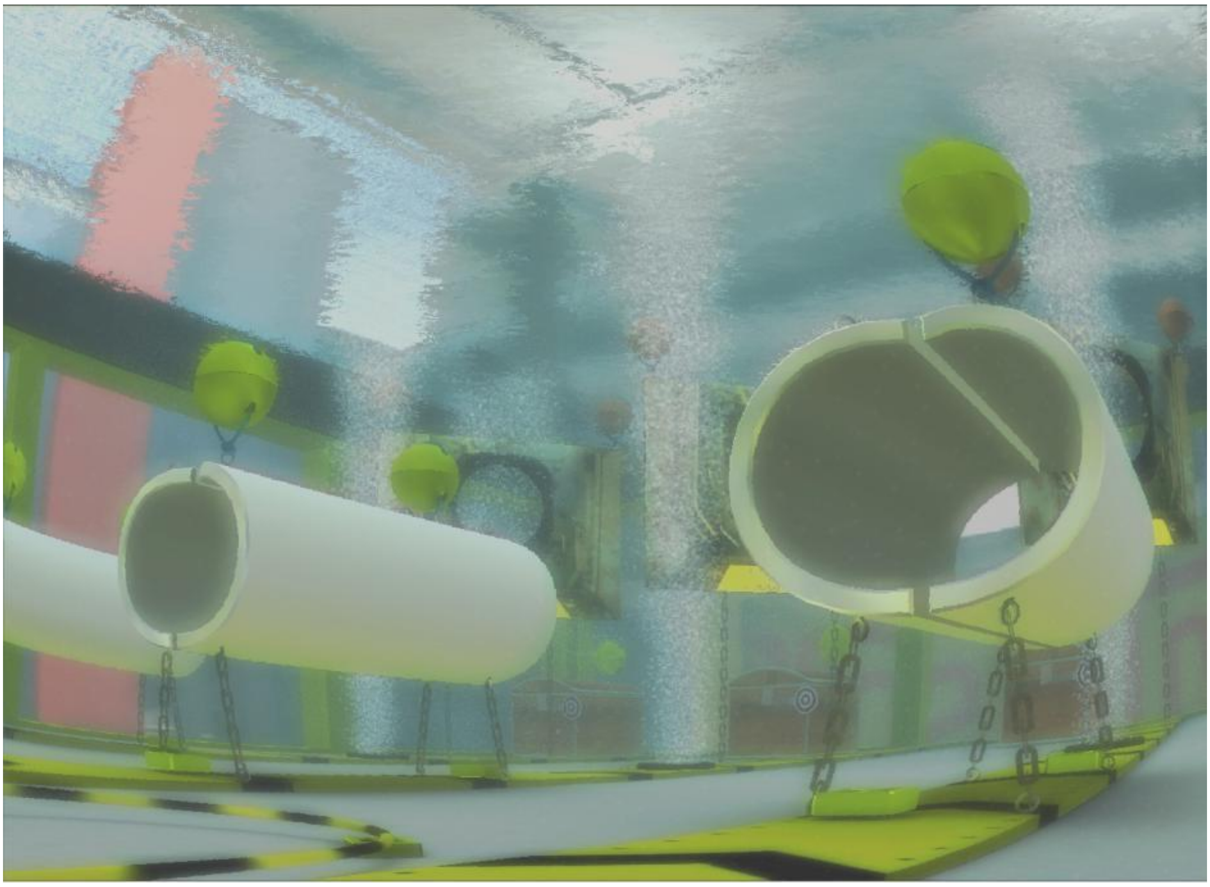


Figure 6.25: The low fidelity training environment.

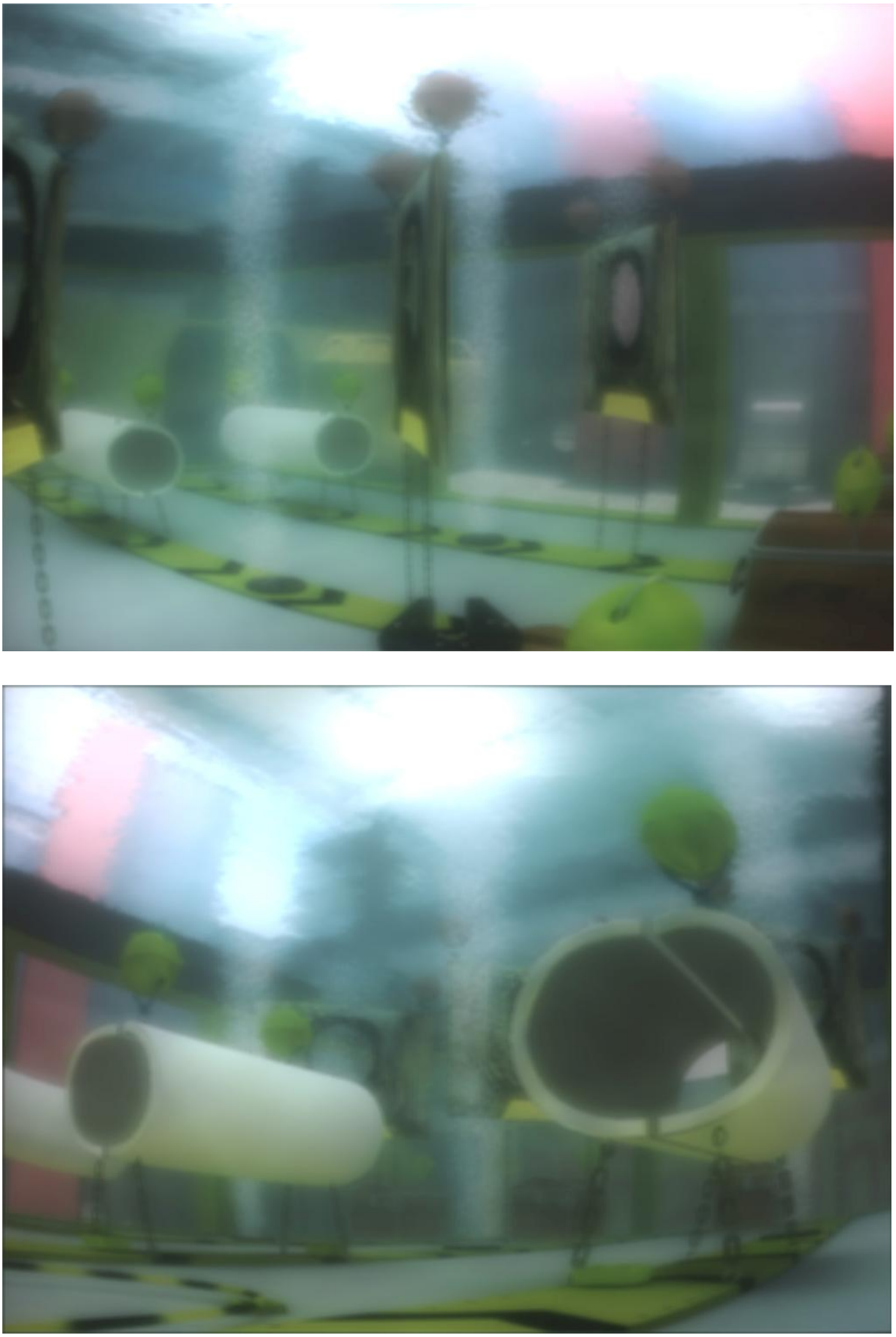


Figure 6.26: High fidelity training environment.

6.15 Discussion

This chapter has described in detail the development of an ROV simulator using a commercial off-the-shelf graphics engine selected due to the findings of chapter three. The Quest 3D engine has proven itself able to handle both the dynamics of ROV flight as well as some of the key visual aspects associated with underwater piloting (Schechner and Karpel 2004). While the key visual aspects have been included there are still a number of omissions in the virtual environment from the real world counterpart. Firstly, there is no simulation of the umbilical cable linking the power and control signals back to the control panel. In most ROV tasks this can be a critical factor in piloting an ROV, ensuring it does not get caught on the surrounding environment and being aware of the extra drag effect on the cable and deeper depths. There has been a significant amount of research conducted on simulating the effects of tethering issues as the dynamics of the cable can become increasingly complex. However, on the ExplorOcean exhibit these effects are less relevant. The NMA designed the exhibit specifically to reduce the chance of any cable snagging. It can be seen in figures 6.6 and 6.7 that slits in the obstacles have been specifically designed to allow the umbilical cable to slip though without hindering the ROV. In addition to this a complex cable runner has been installed above the ExplorOcean tank to ensure that the umbilical is always hanging directly above the ROV's position reducing any drag. For these reasons it was assumed that the effects of the umbilical during the task would be negligible and does not require simulation.

The second omission was the additional controls of the ROV, only the depth gauge and movement joystick are present on the replica control panel. After interviewing the staff that use the ExplorOcean exhibit on a day-to-day basis it was made apparent that these omitted controls would rarely be needed within the confine of the ExplorOcean tank. These controls included the ROV lights and the gripper controls. The exhibit is sufficiently lit as to simply not require any further lighting and the purpose of the transfer training study was on piloting skills not gripper control. For these reasons the replica controls were simplified to focus only on ROV piloting.

Chapter Seven

This chapter details the methods used to collect course deviation data from the ROV to provide an additional metric for assessing the affect of transfer training from the virtual to the real world. It discusses methods for data capture and how the collected data should be analysed.

7.1 Introduction

While recording positional data from the simulated ROV, calculating the position of the real ROV is a non trivial problem (Yilmaz, Javed, and Shah 2006). What is required is a method of tracking the ROV in 6 degrees of freedom without inhibiting the pilot or require any significant changes to the aquarium equipment. There are various methods for object tracking, such as signal triangulation, magnetic resonance and image based processing.

Signal triangulation is the most commonly used with mobile phone networks to establish which is the best "Cell" or tower to communicate with. It is also used for satellite tracking systems such as GPS (global positioning system). The system works by having at least three signals being transmitted from three different locations surrounding the target object. A signal receiver is placed on the target which measures the time delay from each transmitter to establish a location in space. To enable this to work for the ROV experiment at least six transmitters would be required and located around the tank. In addition, the traditional signal generators could not be used as the signals would be attenuated by the water. (Figure 7.1)

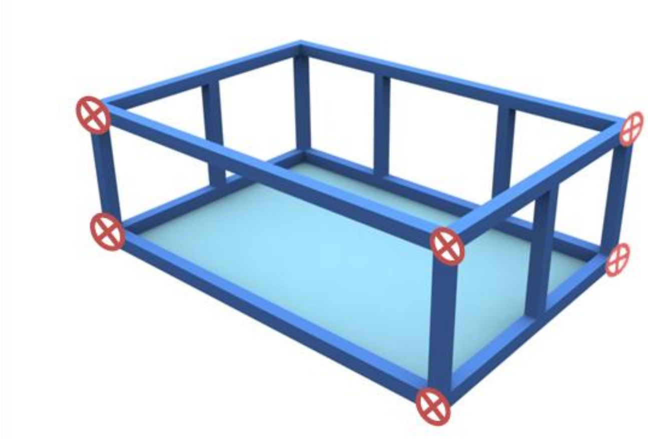


Figure 7.1: An illustration of possible locations for the six required transmitters for ROV tracking.

Sonar signals would have to be used instead as well as the design of a waterproof receiver that was small enough not to hinder performance. An alternative is to use magnetic resonance tracking systems such as the Polhemus PATRIOT[9] motion traker. Essentially a magnetic field is created and a detector can establish its locations from the field strength.

[9] www.polhemus.com

On experimentation with the system it was found that it was only accurate with a relatively small distance (around two meters) and was affected by other close electromagnetic fields. This was very problematic for ROV use as the ROV had three large motors on board that when activated would distort the local electromagnetic field giving false location readings. As the distortion was dependant on the individual use of the motors it was not possible to compensate for the distortion as it always changed dependant on the power going to each motor during manoeuvring.

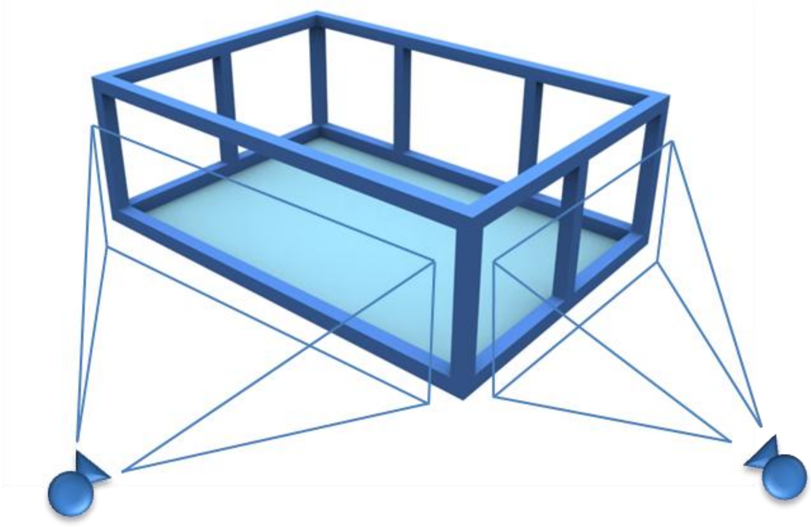


Figure 7.2: An illustration of the possible camera locations for video based tracking.

The final method investigated, and ultimately chosen, is a camera based tracking system. Figure 7.2 shows two cameras pointing at the tank. The object that is to be tracked, the VideoRay ROV, requires a visual marker to allow the cameras to pin point its location in 3D space. Camera one can be thought of as recording the X,Y position of the ROV and the other the Y,Z. To determine the position both cameras must be synchronized perfectly in time and there resultant data combined. Due to the nature of the obstacle course the secondary camera relating to determining the Y,Z axis is considerably less relevant. The course itself can, broadly speaking, be thought of as 2D. This is because all the obstacles are in a straight line, the most participants should deviate left or right is relatively negligible. It was decided that the additional software development and experimental requirements (and a second person to monitor the second camera) would provide very little scientific relevance and only the X,Y coordinate data was to be recorded.

7.2 Image processing

Before the recorded video signal from the camera can be used to calculate the course of the ROV during a trial it must be processed to compensate for the camera angle and distortions.

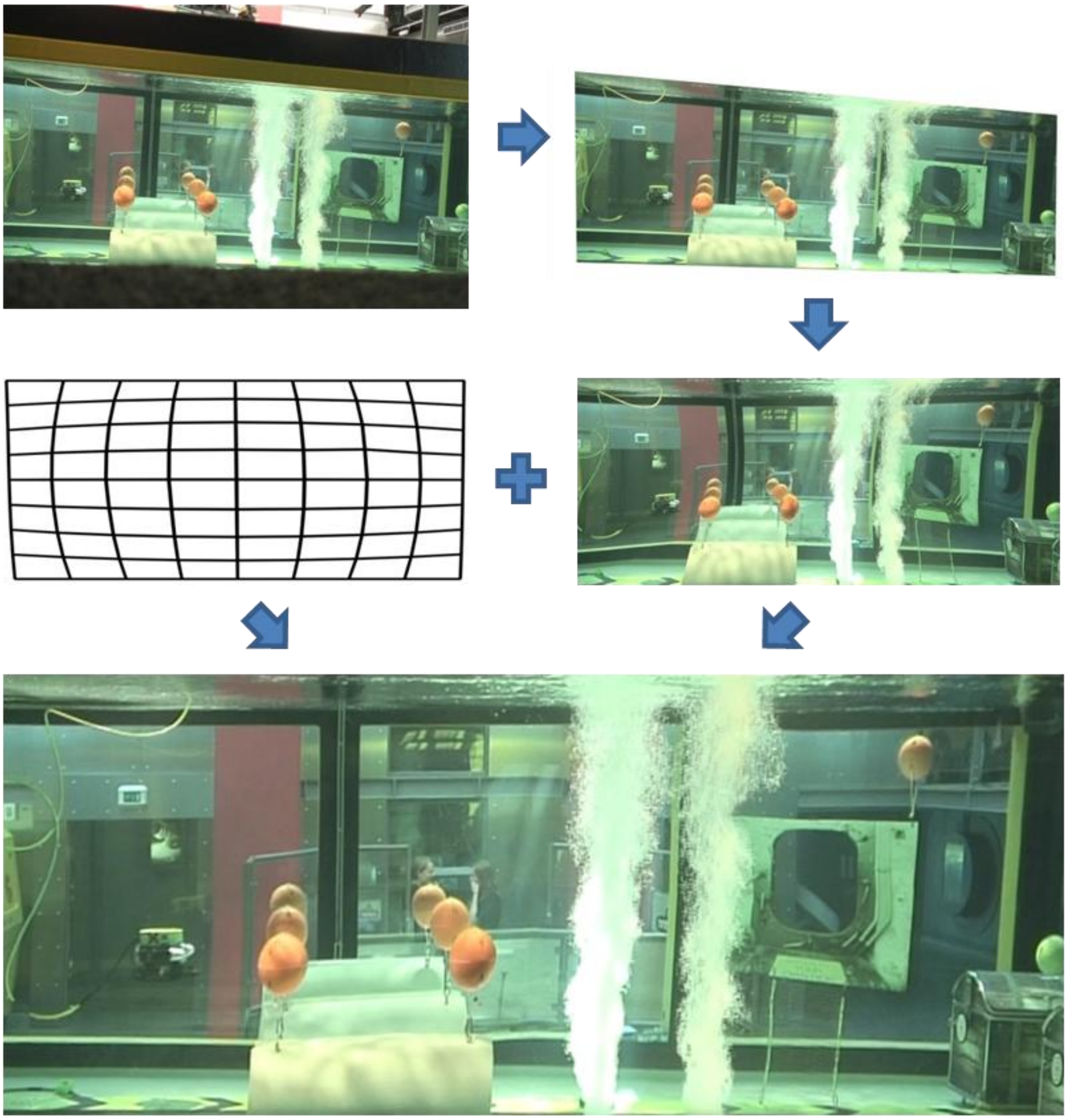


Figure 7.3: An illustration of the stages required to convert the video feed into a usable data source.

Figure 7.3 illustrates the steps used to turn the raw camera footage into a processable video sequence. The first stage is to cut out the irrelevant edges to the video; we are only concerned with the area in which the ROV can travel. The second stage is to compensate for the camera angle. Due to the physical limitations of the exhibit it was not possible to angle the camera at a perfect 90 degrees to the tank. The angle had to be offset by around five degrees which created a slight slant and perspective effect. This was compensated for by stretching the image to a perfect rectangle. The final stage is to compensate for the water and glass distortions of the tank. Both water and glass refracted the light from the tank creating a “pinching” effect in the image. A distortion map with an equal and opposite “pinch” effect was used to compensate for this.

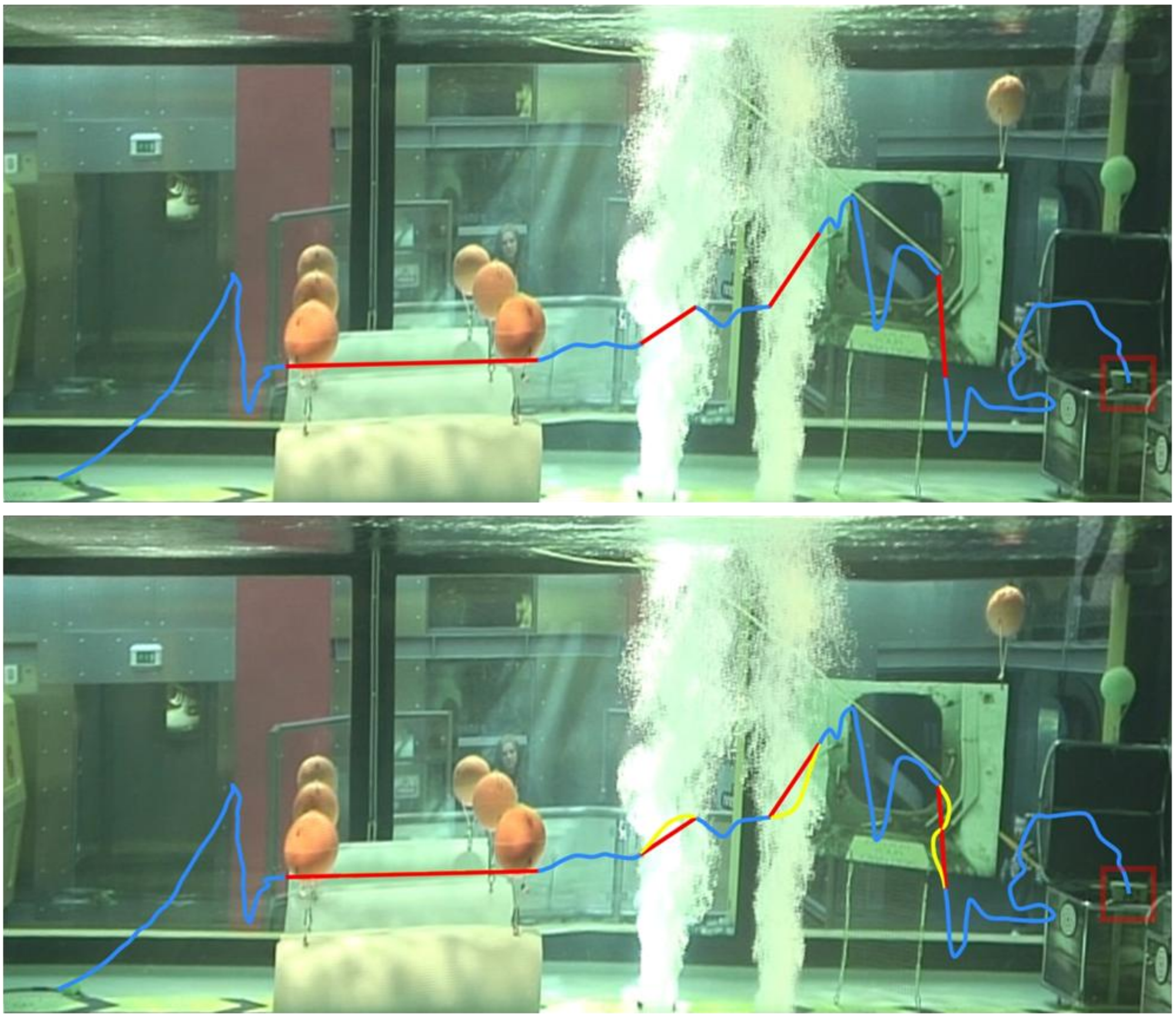


Figure 7.4: Shows the tracked path of the ROV (blue) and the areas where occlusion interrupted the signal (red).

Adobe After Effects, a common video editing and special effects tool, was used to perform the ROV tracking. In figure 7.4 the blue line indicates the extracted course from the video processing software overlaid over the original end frame (split time ROV just entering the chest). The red line indicates areas where the tracking failed to identify the ROVs position. The software assumes that the ROV progress follows a straight line between these points. This assumption is relatively accurate for when the ROV travels through the tube sections as movement is confined within that space, however this is not the case for the bubble streams and the occlusion caused by the port hole. For the three occluded sections of the video the footage was analysed manually, correcting for the deviation from the straight line that the ROV had taken. The yellow line in figure 7.5 indicates the small areas which had to be corrected by hand within the Adobe After Effects package.

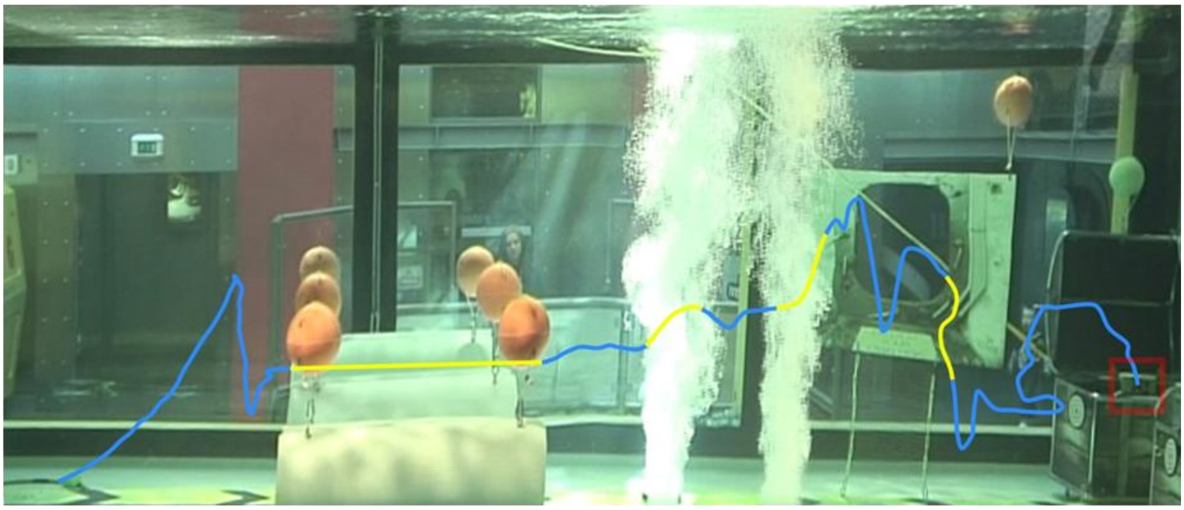


Figure 7.5: Shows the tracked path (blue) with manually correct areas (yellow).

Figure 7.5 now shows only the manually corrected parts of the ROV course. Finally an additional line (in red) is added to illustrate the path of an optimum course run performed by an expert pilot in figure 7.6.

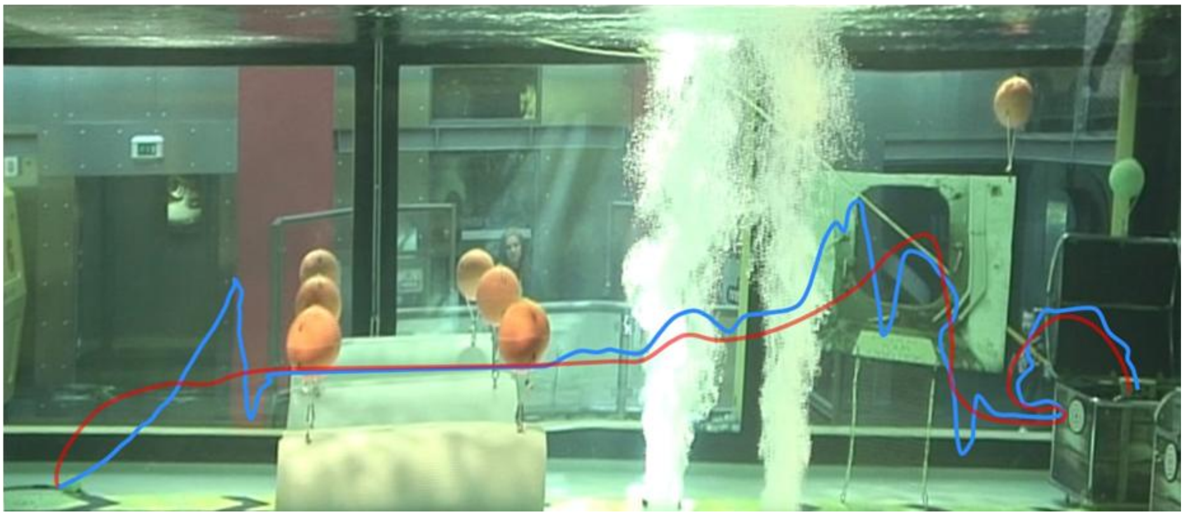


Figure 7.6: Shows the tracked path of the novice pilot (blue) and the expert track (red).

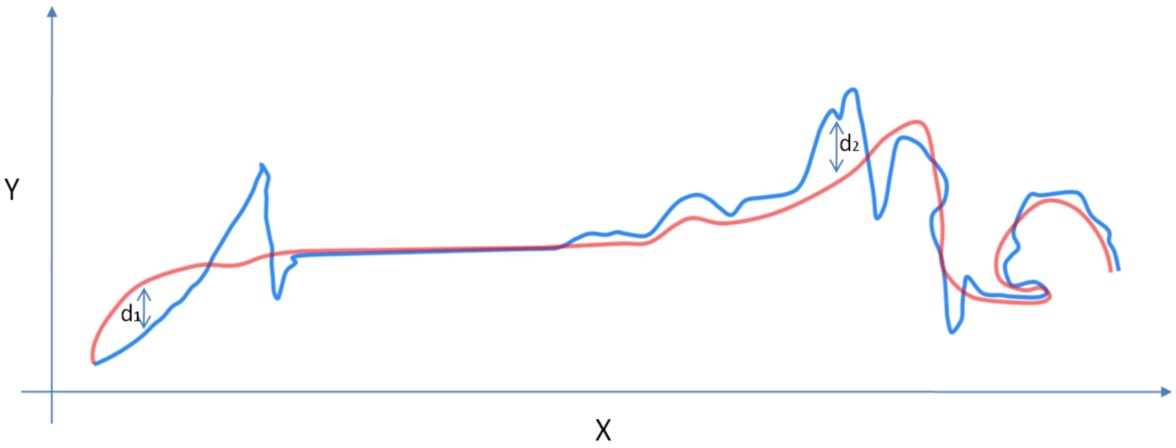


Figure 7.7: Shows the tracked path of the novice pilot (blue) and the expert track (red).

The expert ROV pilot was also recorded, on four attempts, in order to create an ideal course that the novice could be compared to figure 7.7. By examining the difference between the expert and novice courses we can create an additional metric for assessing performance rather than time alone. Clearly the expert would complete the courses considerably quicker than the novice pilots. This time difference must be compensated for, in order to directly evaluate the course deviations.

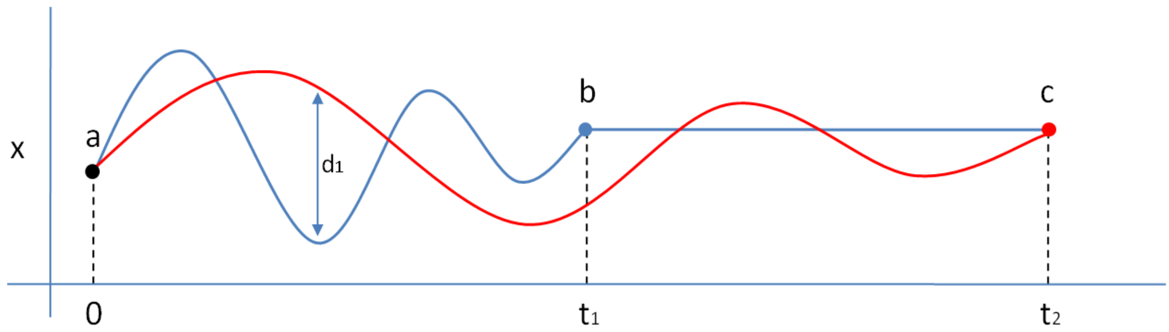


Figure 7.8: An illustration of the time distortion between two signals. Red line indicates amateur attempt and the blue line indicates the expert path.

In figure 7.8 the blue line indicates a hypothetical perfect path of the course which has been completed at time 't1'. The red line indicates the path of a novice participant, who completes the course at time t2. Between the periods of t1 and t2 the expert's position remains stationary as they have already completed the course. By inspection, the value d1 appears to show a great difference between the two paths, the novice position is far from the experts so you may conclude that the novice performed poorly. However, this may not be the case, while the novice has taken longer to complete the course they could have followed the exact same course. This means that time must not be taken account when comparing course deviations.

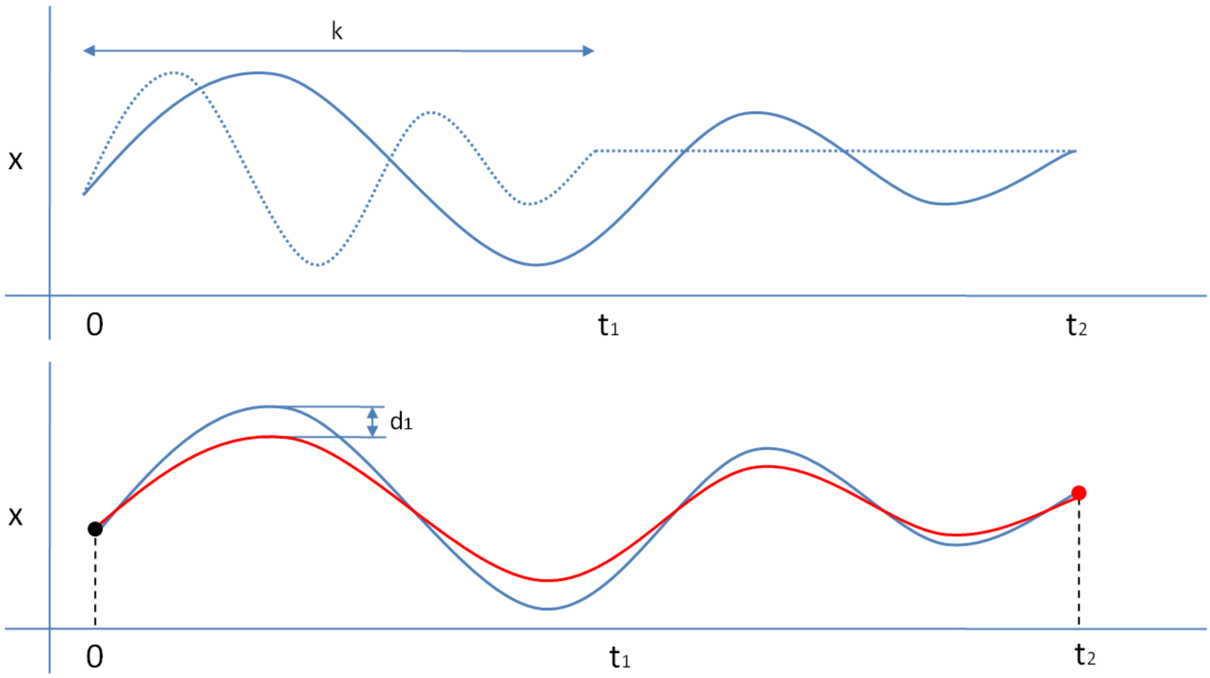


Figure 7.9: An illustration of how a signal can be re-sampled to match another signal in the time domain.

The best method of removing the effect of time is to rescale the recorded expert path so it finishes at the same time as the novices (figure 7.9). Typically the software records the current position every 10^{th} of a second. So if the course is completed in 1 minute then there will be 600 samples (10x60). If however the novice completes the course in 2 minutes then there will be 1200 samples. To remove the effect of time both of these course runs should have the same amount of samples. This is achieved by re-sampling the expert signal to have the same number of points as the novice signal. Once both the experts and novices course samples match a simple sum of square difference is used to calculate a course deviation. The pilot who performed in the quickest time in section 6.5 was used as the ideal course path for the novice pilots to be compared to.

7.3 Questionnaire

As stated in the first chapter it has been one of the aims of this thesis to establish the effects of pervious game playing experience on performance during ROV flight control. In addition to recorded performance of the participants on both the virtual and real ROV exhibit, each participant was asked to fill in a short questionnaire (see appendix C). The questionnaire mainly focused on asking questions on not only the amount of time they played computer games but what type. It has generally be established that people who play computer games a lot generally perform better in hand and eye coordination tasks (Griffiths 2002). This usually refers to people playing 'action' games rather than online card games. For this reason it was important to not only establish the time they spent playing computer games but the type of game as well.

Game type	Not at all	Less than 1 hour per week	1-3 hours per week	4-8 hours per week	9-15 hours per week	16 or more hours per week
First person shooter (Halo, Halflife, Call of duty)						
Third Person (God of war, Zelda, Ratchet and Crank)						
Real time strategy (Command and Conquer, The Sims)						
Simulation (Flight sim, Burnout, GTA)						
Casual games (Card or puzzle games)						

Table 7.1: An example of one of the questions filled out by participants.

7.4 Discussion

In this chapter we have reviewed some of the technical difficulties in accurately tracking an object in 3D space, particularly when in a water based environment. There has been a great amount of research conducted in this area (Stephens 1990; Schmidt *et al.* 2007), however it is not a goal of this thesis to present developments in the field of object tracking but merely to use it as an additional metric to further support the findings of the ROV transfer training trial. As a result a simple image based video tracking system was chosen as a method for sampling the path taken for each course run.

The propriety software, Adobe After Effects was used to track the course taken using an in built tracking function (more typically used to add digital effects to specific objects in the video). Using a video effects package to perform the tracking had several additional advantages. Firstly, the software allowed for the trimming (cutting away) of unwanted areas of the video frame before performing the tracking operation. Secondly, it is also possible to overlay subsequent frames on top of each other and correct any errors in tracking calculated by the software (as only the ROV was moving it was possible to see the course it had taken). While there are still obvious limitations to the method of image based tracking used, most notably the lack of a third dimension, the nature of the obstacle course tends to limit the movement of the ROV to a relatively linear path. In addition, a video based method for object tracking does not require any physical alterations to the ExplorOcean exhibit or ROV.

We have also seen that there is a requirement to ensure that the course data obtained though tracking correctly aligns to the expert course before a comparison is made. This was achieved through a simple re-sampling technique. Effectively the re-sampling allows for a direct side by side comparison between the expert and novice path as the time element has been removed before calculating course deviation.

Finally, the questionnaire developed will help to give a better understanding of the participants involved in the experiment and investigate whether there are any other factors that might have an effect on the piloting of the ROV.

Chapter Eight

This chapter describes the method of experimentation used to evaluate the effectiveness of the developed ROV simulator based on the National Marine Aquarium's ExplorOcean exhibit. It will also present the results and draw conclusions from their findings.

8.1 Experimental Design

This experiment was based on classic transfer training studies involving a control group. In addition to the training cohort, three aquarium-experienced ROV pilots also performed four course runs on both real and virtual conditions. This data was used to establish the optimum course route and timing data which could be used to evaluate the novices' performance.

8.2 Participants

40 participants took part in the experiment. They were all students[10] with ages between 14 and 16 with an average age of 15. Out of the 40 participants 32 were male and 8 female.

8.3 Method

The experiment split the 40 participants into four groups of ten. The first group received the high fidelity training, the second received the low fidelity training and the third group used the real world ExplorOcean exhibit for training. The final group (which formed the control group) received no training at all. To ensure a balanced test and limit any gender bias, the female participants were spread evenly amongst the four groups. Before each participant began the task, in either the real or virtual conditions, they were shown how the controls worked and asked to perform a few basic manoeuvres of the ROV until they were happy they understood the control system. This also included the control group. This was done to ensure that there was no effect on course time from simply not understanding the controls or how they related to the movements of the ROV. While the ExporOcean exhibit has open views on all sides, it should be noted that, it is not possible to see the ROV externally while in the piloting position. Participants had to rely solely on the video feed from the onboard camera just as a real world pilot would have to. They were asked to complete the virtual course four times before then proceeding to the ExplorOcean exhibit and performing four course runs on the real world ROV (see appendix H). Once they had completed the experiment they were asked to fill in a short questionnaire relating to their 'gaming' experience.

A repeated measures ANOVA is used to analyse the data as a whole, where significant differences were found an additional post hoc test was performed. Post hoc tests are used

[10] Additional parental consent was obtained.

when a significant result ($p<0.05$) has been obtained by the ANOVA and additional exploration of the differences among means is needed to provide specific information on which means are significantly different from each other. In this case, the post hoc test will help to understand how the real and virtual participants performed in greater detail. There are a wide variety of post hoc tests available such as Fisher's Least Significant Difference (LSD), Tukey's Honestly Significantly Different (HSD) and Scheffe's test, to name but a few. The question of which is the most appropriate really comes down to the type of data and how liberal (more likely to find a significant difference) or conservative (less likely to find a significant difference) the analysis should be. Fisher's LSD post hoc test is generally considered to be more liberal in finding a significant difference in the data than Tukeys's HSD or Scheffe's test (Thomas 1973). For many experiments this could lead to false positives such as a drugs performance over another. However, in the case of ROV training it is a false negative that could conclude that the virtual training is performing better than it is as it may show that there is no significant difference between the real and virtual training. Fisher's LSD test is basically a set of individual t-tests. The only difference is that, rather than compute the pooled standard deviation from only the two groups being compared, it computes the pooled standard deviation from all the groups. Using all the data to compute the pooled standard deviation gives a more accurate value for the standard deviation. The final method of data analysis is to perform a correlation between the amount of time playing computer games a week and the time in which participants are able to complete the task. The Pearson's correlation is used, a common method for detecting any correlation between variables and any significance. The value for a Pearson's (r) can fall between 0.00 (no correlation) and 1.00 (perfect correlation), a significance level is also calculated requiring a value of less than 0.05 to be considered significant.

8.4 Results

The results are divided into four sections: timing data, collision data, deviation from the ideal course and participant gaming experience. An Analysis Of Variance (ANOVA) is used for all conditions.

8.5 Course Completion Times

The first data to be considered is a direct comparison of time taken to complete the real world course for the three trained conditions and no training. As each participant performs the assault course four times, their progression was recorded for each trial run.

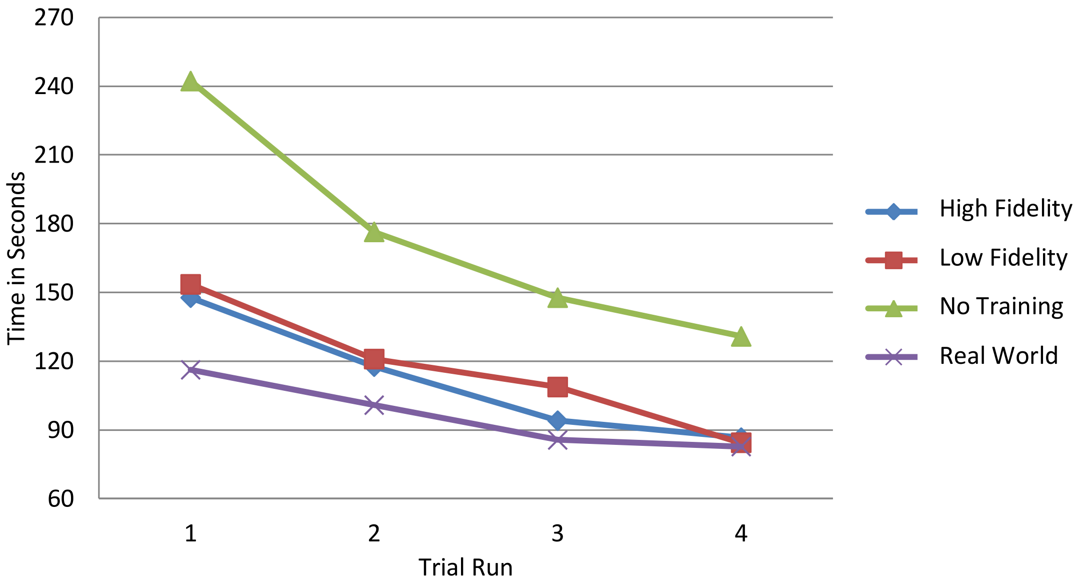


Figure 8.1: A graph showing the mean time taken to complete each of the four course runs for all conditions.

Trial Run	1	2	3	4
High Fidelity - Mean	147.70	117.60	94.00	86.60
High Fidelity - ST DEV	41.13	18.92	30.04	24.58
Low Fidelity - Mean	153.50	120.90	108.70	84.40
Low Fidelity - ST DEV	28.66	25.59	27.44	11.34
Real World - Mean	116.20	100.80	85.60	82.70
Real World - ST DEV	12.66	17.03	29.57	26.79
No Training - Mean	242.20	176.40	147.70	130.80
No Training - ST DEV	39.86	20.72	6.62	9.50

Table 8.1: The mean average and standard deviation (ST DEV) for the four course run completion times (seconds).

Clearly there is a large difference between the no training condition and the other three trained conditions (Figure 8.1). The high, low and real training conditions show means of 147.7 (±41.1)s, 153.5 (±28.6)s and 116.20 (±12.66)s respectively for the first course run where as the no training condition shows a mean of 242.20 (±39.86) (Table 8.1). A Mixed ANOVA was performed on all four of the conditions (high fidelity, low fidelity, no training and real) to establish whether this was significant. The within subjects effect of trial run had a significant main effect [$F(3,108) = 118.8$, $p<0.001$]. This is not surprising as the participants are expected to improve in subsequent course runs. The between subjects measure of training type also showed a strong significant difference [$F(3,36) =$

27, p<0.001]. There was also a significant interaction for trial run and training condition [F (9,108) = 6.949, p<0.001]. This interaction can be further investigated with the use of a Fisher's LSD post hoc test (Table 8.2).

(I) Condition	(J) Condition	Significance (p)
High Fidelity	Low Fidelity	0.568
	Real World	0.047
	No Training	<0.0001
Low Fidelity	High Fidelity	0.568
	Real World	0.034
	No Training	<0.0001
Real World	High Fidelity	0.047
	Low Fidelity	0.034
	No Training	<0.0001
No Training	High Fidelity	<0.0001
	Low Fidelity	<0.0001
	Real World	<0.0001

Table 8.2: Shows the Fisher's LSD post hoc test results for the course completion time (significant at 0.05).

All training conditions show a significant difference from the no training condition (p<0.0001). By comparing the high and low fidelity we see that there is no significant difference (p=0.568). We can also see from figure 8.1 that all three of the trained conditions converge by the fourth trial run; it would seem that by four trial runs the training condition is no longer distinguishable. It is this convergence that will inevitably lead to a significantly high interaction between the trial run and training condition. Clearly being trained on a simulator has a positive effect on the course completion times when transferred to the real world ROV.

This test was performed to evaluate how the time to complete the obstacle course is affected by the method of training received by participants. It is also used to investigate if the level of visual fidelity present in the training simulations is a factor for any of the observed differences in course completion times. The results show a positive effect for the virtual training. However, we see that fidelity has had no significant effect on the participant performance times.

8.6 Course Collision Data

So far we have used the participants' course time as a measure of the effects of simulated training. Another important metric is the amount of collisions that had taken place during each trial run.

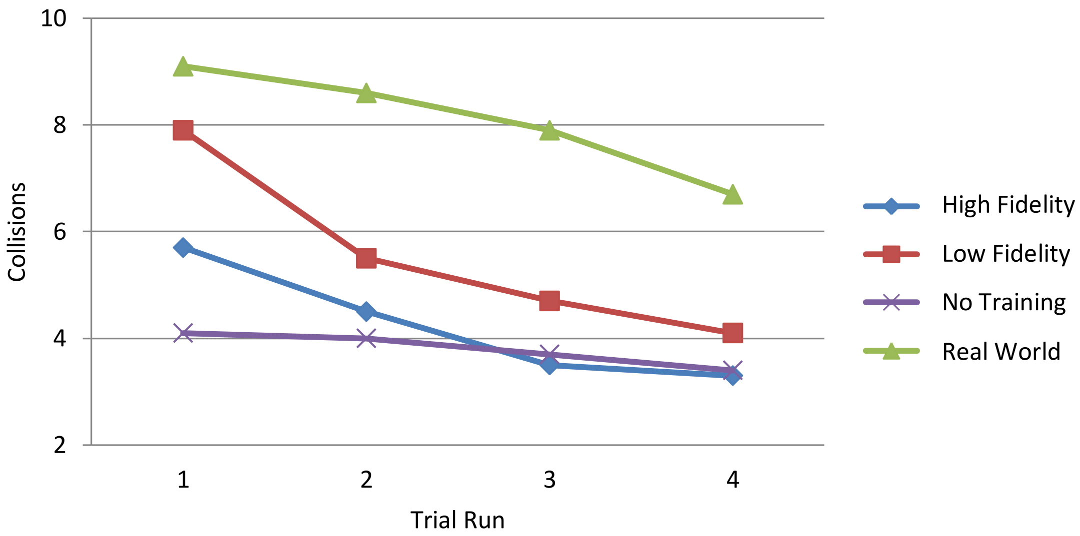


Figure 8.2: A graph showing the mean collisions recorded during the four course runs for each condition.

Trial Run	1	2	3	4
High Fidelity - Mean	5.70	4.50	3.50	3.30
High Fidelity - ST DEV	1.89	2.12	1.78	2.21
Low Fidelity - Mean	7.90	5.50	4.70	4.10
Low Fidelity - ST DEV	1.45	1.59	2.45	2.23
Real World - Mean	4.10	4.00	3.70	3.40
Real World - ST DEV	2.42	2.05	1.83	1.71
No Training - Mean	9.10	8.60	7.90	6.70
No Training - ST DEV	2.81	3.95	3.81	3.71

Table 8.3: Shows the mean average and standard deviation (ST DEV) for the four course runs collision data.

Again there is a large difference between the no training condition and the other three trained conditions (Figure 8.2). The high, low and real conditions show means for the first trial run of 5.7 (±1.89), 7.90 (±1.45) and 4.1 (±2.42) respectively where as the no training condition shows a mean of 9.10 (±2.81) (Table 8.3). A Mixed ANOVA was performed on all four of the conditions (high fidelity, low fidelity, no training and real) to establish whether this was significant. For the within subjects effects of trial run a significant main effect was found [F (3,108) = 11.61, p<0.001]. Again this is not surprising as the participants are expected to improve in subsequent course runs. The between subjects measure of training type also showed a strong significant main effect [F (3,36) = 10.07, p<0.001]. There was no significant interaction for trial run and condition [F (9,108) = 1.131, p=0.347]. This is further investigated with the use of a Fisher's LSD post hoc test (Table 8.4).

(I) Condition	(J) Condition	Significance (p)
High Fidelity	Low Fidelity	0.137
	Real World	0.602
	No Training	<0.0001
Low Fidelity	High Fidelity	0.137
	Real World	0.048
	No Training	<0.0001
Real World	High Fidelity	0.602
	Low Fidelity	0.048
	No Training	<0.0001
No Training	High Fidelity	<0.0001
	Low Fidelity	0006
	Real World	<0.0001

Table 8.4: Shows the Fisher's LSD post hoc test results for the recorded course collisions (significant at 0.05).

There is no significant difference between high fidelity and real world training conditions (p=0.602). However, there is a significant difference between low fidelity and real world training (p=0.048). This would suggest that, in terms of collisions, the high fidelity training is more suitable as a replacement to real world training than the low fidelity condition.

This analysis was used to investigate whether the type of training on a virtual ROV has an effect on the amount of collisions during the trial. The results show that there is a considerable reduction in the amount of collisions when some form of training has occurred and that the high visual fidelity condition performs closer to real world training than that the low fidelity condition. This could mean that a more cautious approach is trained when using the high fidelity condition or that participants have been trained to become accustom to the more typically real world visual experience.

8.7 Course deviation

We can now examine the course deviation data. As previously stated four expert ROV pilots performed the same task several times (section 6.5) and the best one became the optimum course which was used to compare against the novice participants.

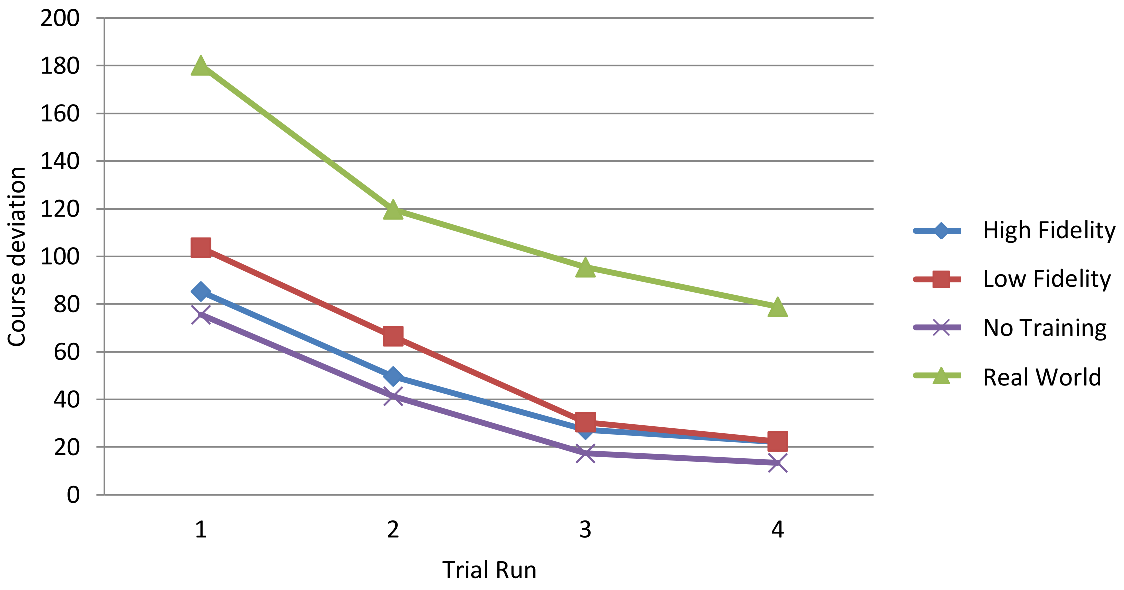


Figure 8.3: A graph showing the root mean course deviation of the four conditions for each of the course runs.

Trial Run	1	2	3	4
High Fidelity - Mean	85.26	49.61	27.31	22.21
High Fidelity - ST DEV	8.71	6.97	5.96	4.94
Low Fidelity - Mean	103.62	66.50	30.48	22.39
Low Fidelity - ST DEV	13.53	13.14	4.33	6.62
Real World - Mean	75.51	41.33	17.33	13.39
Real World - ST DEV	5.35	8.02	5.41	2.05
No Training - Mean	180.15	119.74	95.47	78.85
No Training - ST DEV	7.36	6.27	3.86	7.39

Table 8.5: Shows the mean average and standard deviation (ST DEV) for the course deviation data over four course runs.

Again there is a large difference between the no training condition and the other three trained conditions (figure 8.3). The high, low and real conditions show means of 85.26

(±8.71), 103.62 (±13.5) and 75.5 (±5.35) respectively whereas the no training condition shows a mean of 180.15 (±7.36) (Table 8.5). Again a mixed ANOVA was performed on all four of the conditions (high fidelity, low fidelity, no training and real) to establish whether this was significant. For the within subjects effects of trial run a significant main effect was found [$F(3,108) = 932.4$, $p<0.001$]. Again this is not surprising as the participants are expected to improve in subsequent course runs. The between subjects measure of training type also showed a strong significant main effect [$F(3,36) = 794.5$, $p<0.001$]. There was also a significant interaction for trial run and condition [$F(9,108) = 13.155$, $p<0.001$]. This interaction can be further investigated with the use of a Fisher's LSD post hoc test (Table 8.6).

(I) Condition	(J) Condition	Significance (p)
High Fidelity	Low Fidelity	<0.0001
	Real World	<0.0001
	No Training	<0.0001
Low Fidelity	High Fidelity	<0.0001
	Real World	<0.0001
	No Training	<0.0001
Real World	High Fidelity	<0.0001
	Low Fidelity	<0.0001
	No Training	<0.0001
No Training	High Fidelity	<0.0001
	Low Fidelity	<0.0001
	Real World	<0.0001

Table 8.6 Shows the Fisher's LSD post hoc test results for the course deviation (significant at 0.05).

All conditions show a significant difference in the variation from ideal path. This would indicate that, in terms of course deviation times the level of fidelity of training simulator has no significant effect.

Although time to complete a task is an important metric in evaluating a participant's performance, conservation of movement and correct manoeuvres should also be a factor. The results show a significant benefit to virtual training before the task, but in terms of appropriate course taken it does not matter whether it is of high or low fidelity.

8.8 Questionnaire results

As earlier stated, each participant answered a short questionnaire on their gaming habits (see appendix C for the full questionnaire). It has long been thought that computer games increase hand eye coordination (Griffiths 2002). As such, it was decided to investigate whether the amount of time spent using computer games is correlated to the participants' overall performance within the ROV.

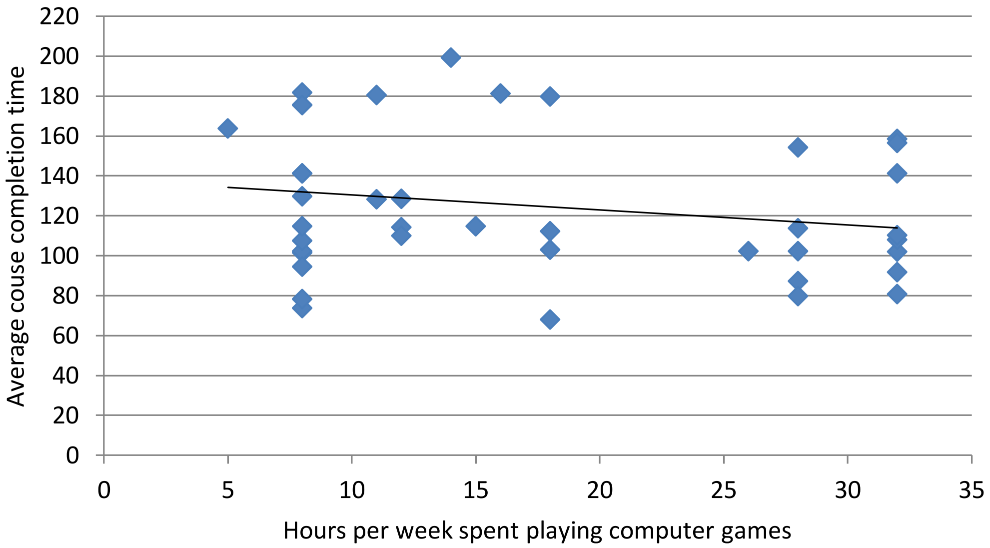


Figure 8.4: A graph showing the estimated average time spent playing computer games against the participants mean course time over four runs.

As expected there is a small, but non significant, negative correlation between the amount of time spent playing computer games per week and the average time required to complete the ROV course [$r = -0.210$, $n = 5$, $p = 0.194$] (figure 7.13). Participants that spend more than 30 hours a week on using computer games have an average course score which is 20 seconds faster than those that spend less than 10 hours a week using computer games. This clearly reflects the current findings in the area, but such a small difference reflects the fact that it is only a small factor in a participant's ability to use ROVs. This questionnaire was performed to evaluate whether there could be any other effects that may have affected the statistical analysis and potentially should be factored out. The results show only a very small effect which is not significant suggesting that it was not a major contributory factor in course times and unnecessary to factor out.

8.9 Discussion

Training transfer experiments are relatively common in the field of flight simulation (Hays *et al.* 1992), the same cannot be said for the piloting of ROVs. While numerous commercial simulators exist (Fletcher 2002) little evidence is available (at least in the public domain) as to their effectiveness in training basic piloting skills. Studies have also shown that increased visual fidelity does not necessarily lead to better training (Andre and Wickens 1995; Scerbo and Dawson 2007).

The present experiment was deigned to investigate both training transfer and effects of variable visual fidelity, as well as demonstrating the use of a serious gaming approach to the development of a viable ROV simulation.

The results suggest quite strongly that there is indeed a training benefit to using the ROV simulation before performing the same task in the real world. This conclusion further supports findings of previous work (Valverde 1973; Hays *et al.* 1992) that virtual pilot training can replace real world training for a greater cost saving and still yield effective results.

It has also been shown that there is little difference in performance between participants training using either the high or low fidelity system, in terms of the time it takes to traverse the underwater obstacle course. However, the same cannot be said for the amount of collisions recorded. Results indicate that the high fidelity simulation and the real world training had no significant difference in collisions recorded. The low fidelity simulator performed significantly worse than real world training. Whilst the lower visual fidelity simulator had numerous visual factors removed, possibly the most important omission was that of shadows. It has long been established that shadows provide an important depth cue to the user when navigating through a virtual world (Hu *et al.* 2000; Sugano, Kato, and Tachibana 2003). It would therefore seem likely that one major reason for the increase in collisions is due the participants' lack of ability to judge the distance between objects accurately without this important monocular depth cue. It would seem that the lack of this factor during training may have affected their initial ability to judge distances in the real world. However, another possibility - one important factor not present in the low fidelity training simulator - is the "fish eye" lens distortion. A distorted image has the effect of visually altering the apparent distance to and between surrounding objects. Studies have shown that images distorted by lens effects have been known to affect depth perception

(Corke 1993) particularly in surgical simulation (Tendick, Bhoyrul, and Way 1997). Participants trained on the high fidelity simulator have most likely learned to adapt to this distortion before they performed the task in the real world.

Numerous studies show that there is a positive effect on hand-and-eye coordination with an increase in gaming use (Griffiths 2002; Green and Bavelier 2003; Dye, Green, and Bavelier 2009). However, the questionnaire employed in the present study failed to find a significant result with course collision or course deviation and only a small effect when considering the course completion time when correlated to hours of game play per week. This effect may be due to all participants acknowledging a relatively high use of computer games, all indicating over five hours a week.

Whilst every attempt has been made to provide both a physically and physiologically accurate reconstruction of the task of ROV piloting, it is an impossible to achieve 100% accuracy, even with today's computational power. The dynamics of real-world underwater flight are extremely complex. Taking this into account, it is understandable why virtual training would not perform quite as well as real world training, but what has been demonstrated here is that the results suggest there is a considerable advantage over no training at all.

With such a positive transfer of training from the developed virtual simulation it would be reasonable to say that it has sufficient physiological fidelity to form the basis of a test bed that can investigate the use of technical aids during ROV search tasks.

Chapter Nine

This chapter describes the further development of the ROV simulator presented in chapter six into a test bed that can be used to evaluate the use of technical aids during search. It also describes how the use of an eye tracking system can be used to objectively measure the effect that visual fidelity has on technical aid usage.

9.1 Introduction

The final experiment brings together the technical aid-based search findings from chapter four and combines them with the ROV simulation work of chapters six, seven and eight. The aim of this chapter is to describe the development of a COTS-based simulation which can be used to investigate technical aid usage while piloting ROVs.

Search tasks using modern ROVs are significantly different from using a simple metal detector. ROV Pilots are presented with significant amounts of data from technical aids such as sonar data, bathymetric data, ROV speed, orientation and depth. It is hoped that through the evaluation of a ROV simulation that it will be possible to better understand the use of this data and whether the presence or lack of visual effects alters participant's method of search.

The research in chapter eight has demonstrated that a COTS simulation has a positive training benefit for ROV control. However, the ExplorOcean tank is not adequate to accurately represent real world search tasks due to its small size and its unnatural lighting. Therefore, a more fitting environment must be created.

In addition to the ExplorOcean exhibit, in 2004 the National Marine Aquarium purchased the HMS Scylla from the navy and sank it as an artificial reef (Stone *et al.* 2009).

9.2 The Scylla Reef Environment

In 2004 the National Marine Aquarium purchased the HMS *Scylla*, a decommissioned Leander Class Frigate. The project involved scuttling the vessel in Whitsand Bay, West of Plymouth, to form Europe's first artificial reef, with the additional aim of increasing the local economy income by around £1 million pounds annually, through increased tourist diving and interest from the marine research community. Previous projects have demonstrated the success of artificial reefs in supporting the evolution of diverse ecosystems, from plant life to new colonies of fish (Bohnsack and Sutherland 1985). From the beginning of the project, the artificial reef has supported the study of colonisation by marine biologists. Although the Scylla, after a year or two of resting on the sea bed, had become a popular destination for divers, access was difficult for the general public, for obvious reasons. After discussions with the NMA a concept of a virtual recreation of the Scylla Reef was discussed, this would not only serve the research needs of this thesis but also be advantageous to the NMA. By creating a virtual version of the Scylla Reef as a test bed for experimentation the NMA would also be able to reuse it as an ongoing exhibit. It would allow the general public to be able to experience exploring the artificial reef from the safety of dry land. The Virtual Scylla Reef was, therefore, ideal for experimentation and was developed to support the work reported here regarding the use of additional technical aids in remotely operated systems. The Scylla Reef provides the ideal environment to perform an ROV search task as its structure includes a wide external environment. In addition, the wreck’s internal structure boasts tightly packed corridors which provide an area to test the pilot’s ability to navigate a close-quarters environment.

The Virtual Scylla model was constructed using 3D Studio Max, a common content creation tool for the games industry (Figure 9.1). Detailed deck plans are often difficult to find for many naval vessels, particularly a vessel of this age. The Scylla had also been subjected to several significant refits which meant building an accurate model was difficult. Sadly, very few images were taken during the preparation stages for her sinking, but what few images existed were correlated and combined together in Adobe Photoshop to produce the basic textures.

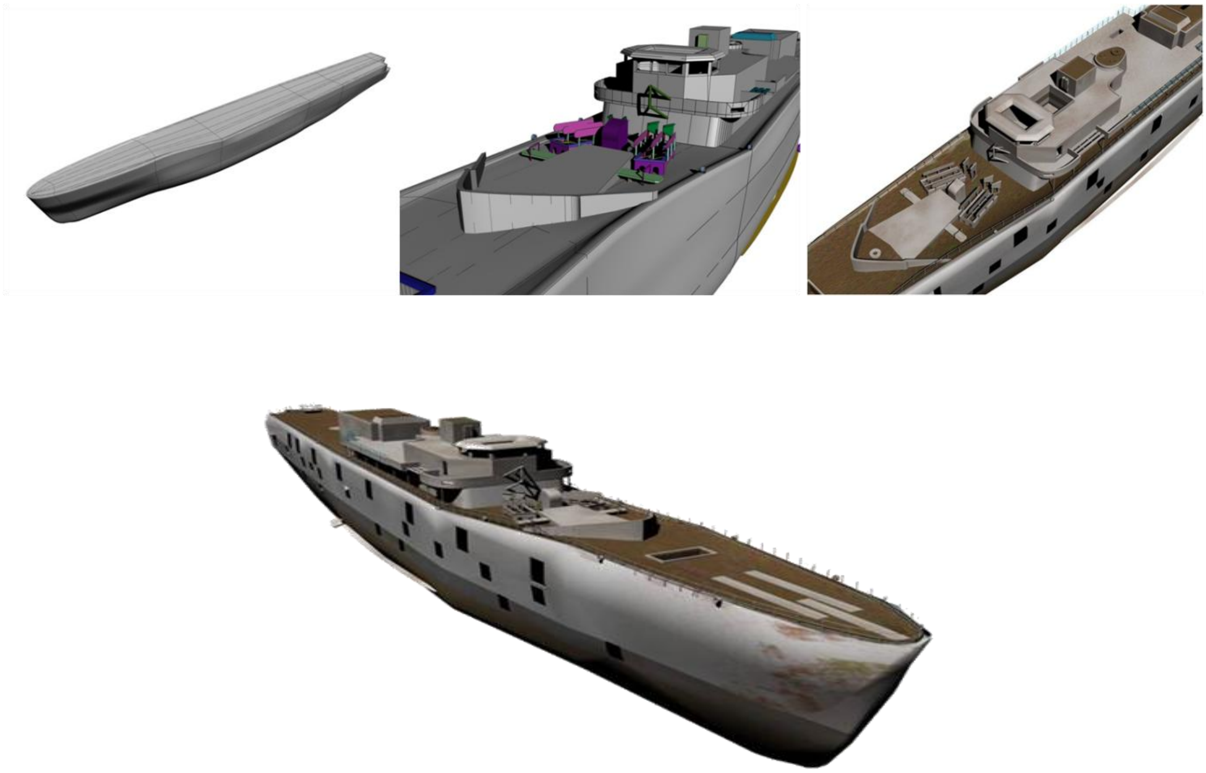


Figure 9.1: Images depicting the development process of the Scylla model in 3D Studio Max.

Initially, the Virtual Scylla underwent a prototype development stage which was based on the Crytek CryEngine rather than Quest3D. This was due to the requirement to create a prototype rapidly for demonstration to the National Marine Aquarium, thereby helping them to understand the capabilities and limitations of modern game engine visuals.

Once the Scylla model was completed, it was imported into the CryEngine test environment to be scaled and textured. Although the Scylla model itself looked relatively impressive, it still did not have a credible underwater visual “feel”, nor did it support the accurate control of a virtual ROV. CryEngine’s underwater fogging capability is, in fact, less accurate than its above-water fogging. Ironically, to achieve a reasonable subsea effect, all of the virtual water was removed from the environment, thus allowing the exploitation of the more accurate above-water fogging throughout.

In addition, particle effects were used to add to the illusion of a murky underwater environment. The particle effects are simple, in essence semi- transparent textures that fill the environment giving the look of particulate matter within the water (Figure 9.2). The final addition to the Virtual Scylla scenario was a simulation of an ROV. This presented the biggest hurdle whilst developing with the CryEngine, as the game itself did not support any kind of flying vehicle, let alone an underwater vessel. It was hoped that modifications could be made to the engine to support this, but it transpired that such adjustments would have had to be made deep within the engine’s source code. Due to this, the first CryEngine-based Virtual Scylla implementation boasted a very limited control system, with the forward thrust component permanently turned on.

Figure 9.2: An image taken from the 'CryEngine' based virtual Scylla.

Although the CryEngine-based prototype presented many developmental problems, it did serve as an effective early visual demonstration and was presented to the NMA, who approved the virtual interpretation of the reef. From this, a more accurate simulation could now be undertaken using the Quest3D engine. This time the greater development freedom provided Quest3D allowed for improved visuals and a more realistic flight model for the ROV.

9.3 Virtual Scylla Development

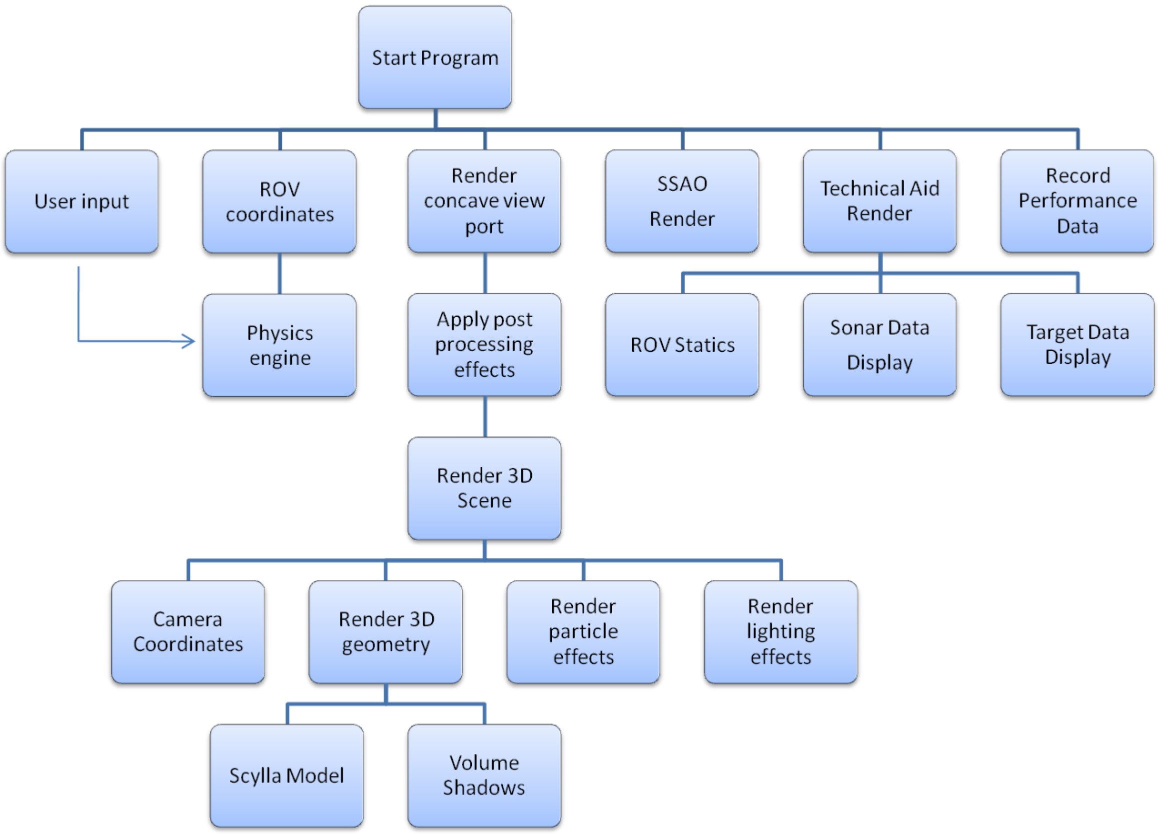


Figure 9.3: Hierarchical structure of the Virtual Scylla experiment.

The Scylla search simulation is based heavily on the ROV ExplorOcean experiment. In both cases the control and physics engines are exactly the same and their development can be reviewed in Chapter six. There are three major additions to the system, the underwater particulate matter, the screen space ambient occlusion shadow rendering system (SSAO) and finally the additional technical aids (Figure 9.3). Each of these additions is expanded below.

9.4 Underwater ambient light

Deep underwater lighting is very different to typical surface lighting. In most real-time 3D applications only the direct lights, such as the sun, are considered when producing shadows. In underwater environments the sun's light is scattered and diffused by the surface and no longer produces well defined shadows. Instead, diffused indirect lighting is present and appears to come from all directions. Ambient Occlusion (AO) is the term generally used to describe shadows that have been created from indirect lighting where light reduces in a particular area due to occlusion from surrounding objects. As seen in figure 9.4, point P is occluded as it is below the surface of another object.

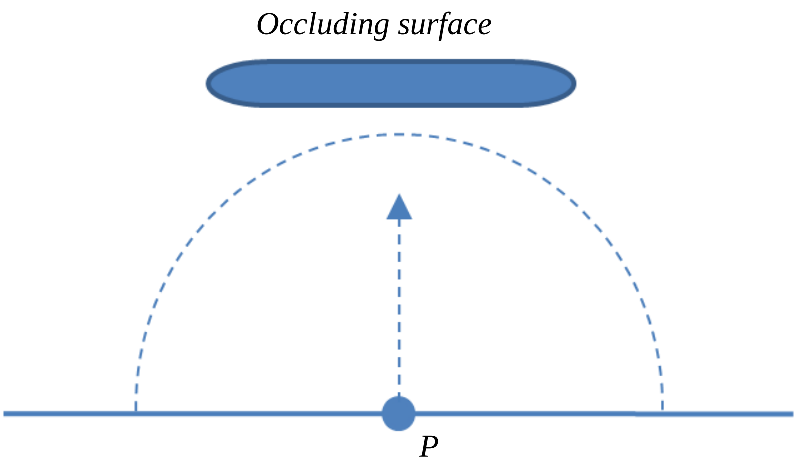


Figure 9.4: An illustration of an occluding surface covering point p.

Essentially AO can be used to approximate the level of light affecting a surface by estimating the attenuation of light due to occlusion of surrounding objects. Typically AO has only been used in non real-time 3D graphics as it involves costly ray tracing calculations.

Figure 9.5 shows the traditional, non real-time, method for calculating AO. For each point P on a surface rays are cast out to determine the brightness of a given pixel. A ray which successfully reaches the background (or pre-determined distance) adds to the pixel's brightness. If a ray intersects with another surface it does not add to the brightness. Pixels surrounded by a large amount of geometry appear darker than those that are not.

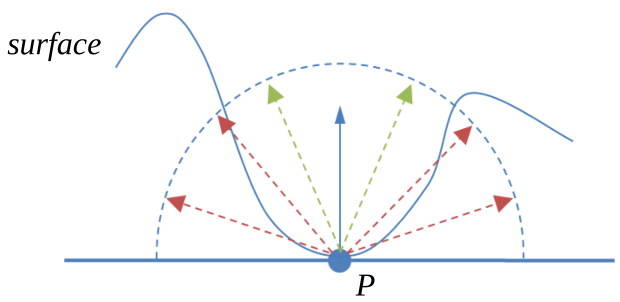


Figure 9.5: An illustration of point p on a surface tracing out local geometry collisions to calculate occlusion.

While simple to implement, this method of determining occlusion is heavily CPU intensive. The number of points sampled usually depends on the resolution of the final output image (i.e. 1024 x 768). Each one of the points may have 50 rays cast out to calculate the occlusion. The resultant rendering may take several minutes per frame, even on a high end machine.

Only recently have new methods been proposed to simulate ambient occlusion in real-time, the simplest of which is to pre-calculate the ambient occlusion data, store it as a texture map and apply it to the objects within the scene at run-time as an additional texture. This process is often referred to as "texture baking". There are two main disadvantages of using pre baked textures. Firstly, unless the size of the texture map is sufficiently large (4096 x 4096) fine details can be missed. Secondly, a baked texture map cannot take account of any movement or animation of the object within the scene as the shadow will remain static until recalculated. Because of this several techniques have been presented to produce ambient occlusion in real-time.

In 2005 the concept of ambient occlusion fields was proposed (Kontkanen and Laine 2005) where objects would no longer have to remain static to produce ambient shadows. Essentially, each object within the scene is subjected to a pre-processing stage to determine the size and average direction of occlusion that will be produced by it dependant on the light source position. This information is stored as a cube-map (a texture that can be placed on six sides of a cube that surround the object) for every animated object in the scene. When two or more animated objects are in close proximity their resulting cube-maps can be combined to give an overall average for the local occlusion.

Figure 9.6: Showing multiple animated self shadowing cubes at 17 frames per second, reproduced from (Kontkanen and Laine 2005).

Whilst the resultant images produce realistic occluded shadows that could be animated, the system did require pre-processing (figure 9.6). For complex scenes, pre-processing time was recorded at 66mins when using the high quality 64 x 64 cube maps, however a reasonable effect can be seen using the lowering quality setting at around 5 minutes. It was also noted that for multiple simple (20 cubes) animated objects, the frame rate would only be around 17 fps. This would make it unsuitable for most "gaming" applications as it fails to reach the generally accepted minimum of 25 fps for real-time rendering.

To improve the overall performance and remove the need for any pre-processing it is possible to take advantage of the Graphics Processing Unit (GPU) such as the work presented by Sarletu and Klein. Their work involved creating ambient occlusion without the use of pre-computation by using hardware acceleration *(Sarletu and Klein2004).* To achieve the effect of ambient occlusion they render the scene from the point of view of a light source to identify what vertices of the geometry can be ‘seen’ from that light. This process is the repeated for a number of light sources surrounding the object and they are added together as can be seen in figure (9.7). Essentially, vertices that are seen by fewer light sources will be the vertices that are occluded from more of the light and appear darker.

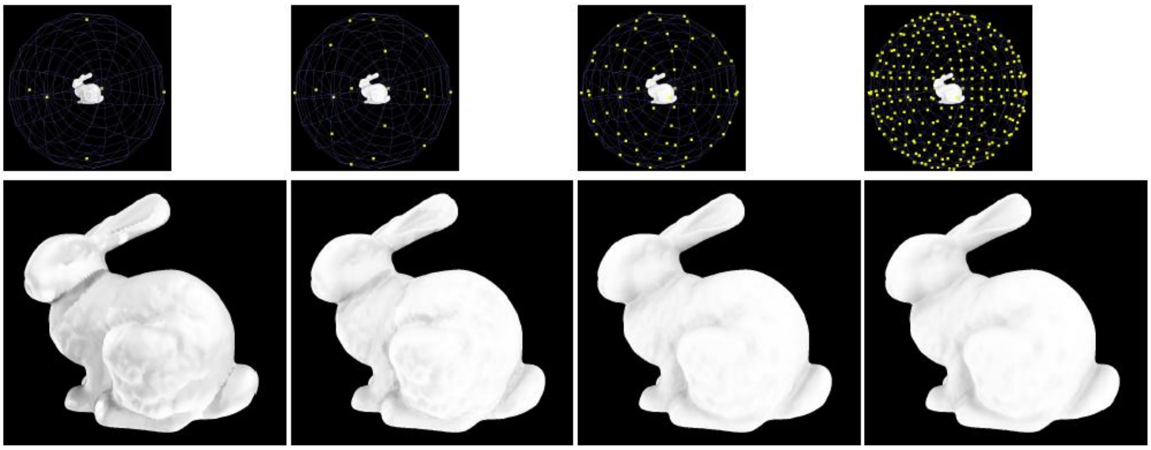


Figure 9.7: Object rendered with different number of light sources (6, 18, 66, 258). The upper row shows the light sphere configurations, where the yellow dots represent the light directions. Reproduced from (Sarletu and Klein 2004).

As this repeated calculation takes advantage of the processing power of the GPU, it requires no pre-processing time and also supports animation without significant performance drop. The recorded frames per second, when rendering 3200 vertices, are around 20, which is almost real-time. However this still would not really suit a complex game environment as with the increased demand from the other effects present such as particles, physics and shaders the overall frame rate would not be playable.

In more recent years, an alternative approach to producing ambient occlusion has been presented and successfully integrated into a main-stream game. Screen Space Ambient Occlusion (SSAO) is a method for approximating ambient occlusion in real-time without the need for intensive CPU or GPU usage. This technique was popularised by the 2007 game *Crysis*[11] and the developers have published papers documenting its development (Mittring 2007). Since then several alterations have been presented from developers such as Nvidia (Ritschel, Grosch, and Seidel 2009).

The SSAO calculation is performed by making use of a 3D scene depth map (Figure 9.4). A depth buffer can simply be thought of as an image rendered from the users current position with objects closest to the user being white and most distant being black.

[11] www.crytek.com

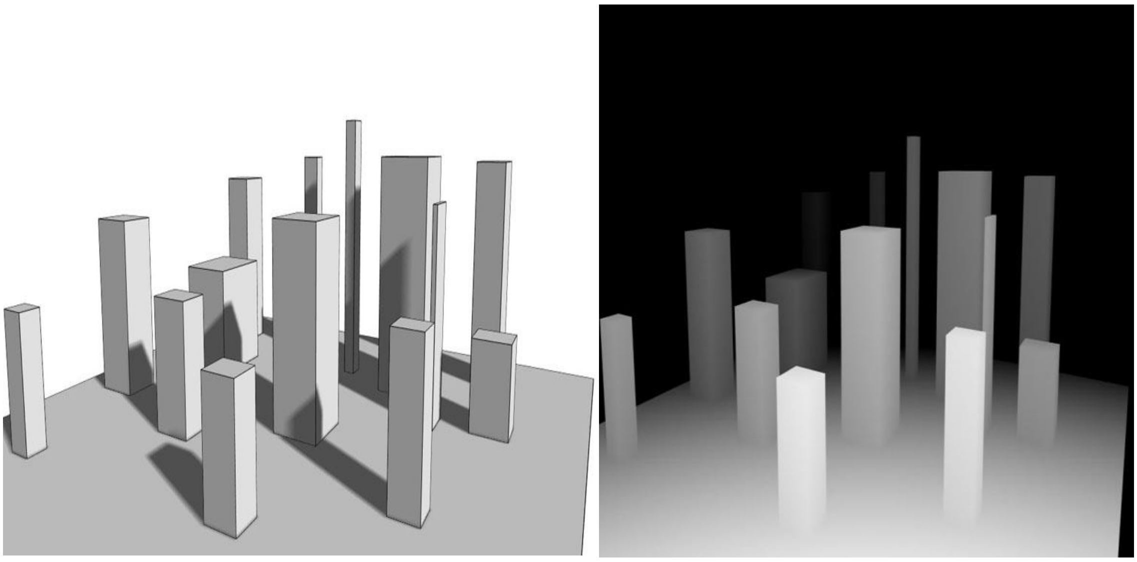


Figure 9.8: A scanline rendering of a simple 3D scene (left) and the same scene rendered as a depth map with distant objects appearing darker (right).

Figure 9.8 shows that when a depth map is created the blocks further away are much darker than the ones closer. Depth maps are typically created for many reasons in modern computer games for use in image masking, depth blur and shaders. To calculate the AO the depth map is examined on a per pixel level. For every screen pixel the surrounding depth map local pixels are sampled and used to estimate the amount of occlusion.

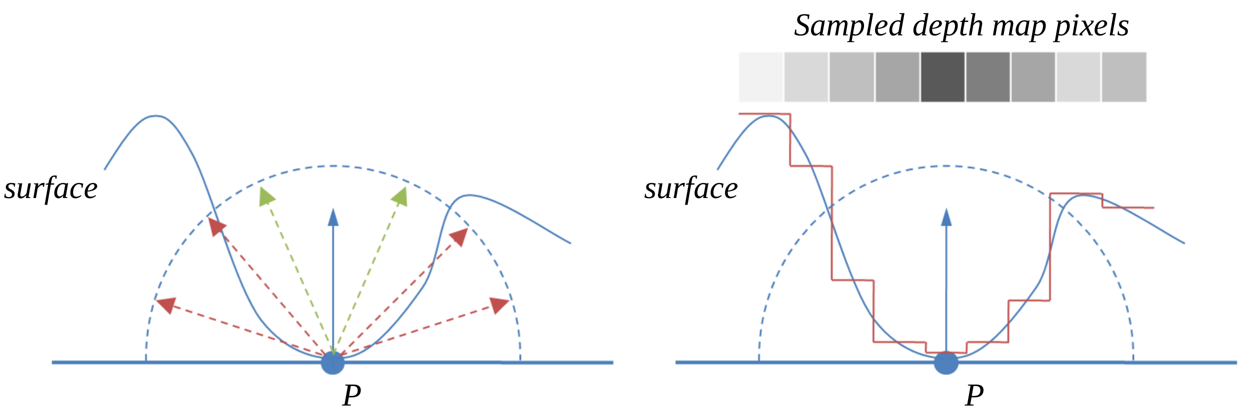


Figure 9.9: An illustration of per pixel occlusion ray casting (left) and per pixel occlusion approximation using the depth map.

Figure 9.9 shows how the depth buffer can be used to recover a basic representation of the surrounding geometry. We can see that as the surface peaks and troughs the depth value changes from white to black. Instead of tracing ray from point P, surround depth values can be inspected to determine an approximation to the resultant brightness. If Point P is surrounded by high peaks in the surface, or white values in the depth map, then its brightness should be reduced.

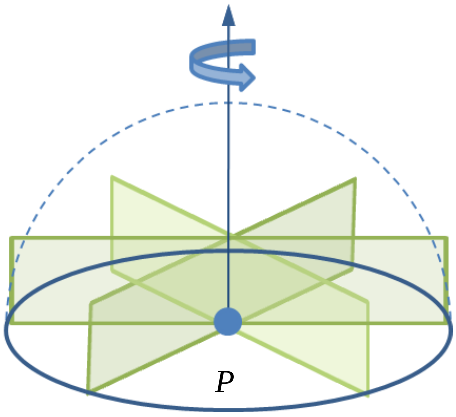


Figure 9.10: An illustration of the depth map being sampled six times through different rotational angles.

For each pixel the depth information must be sampled from multiple angles, we evaluate a fan of slices around each pixel. Typically 8 to 16 slices are used with their exact position randomised to produce a less uniform result (figure 9.10).

While this method produces a very good approximation it is still quite a demanding effect and requires a more powerful graphics card. It is proposed that the depth buffer can produce a far simpler approximation to AO with very little effect on the program performance. Essentially a difference map can be constructed by blurring the depth buffer and multiplying it with and un-blurred one (figure 9.11). The resultant image shows areas of sudden depth change which can usually be associated with an area that is occluded.

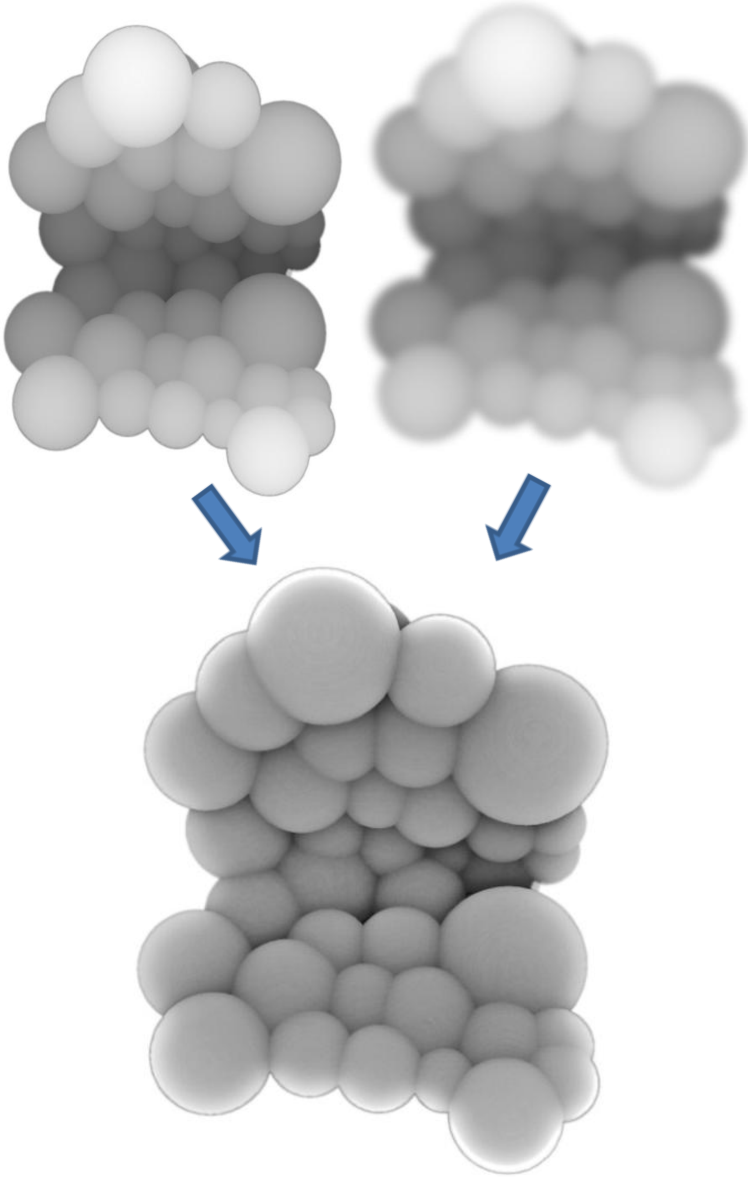


Figure 9.11: An illustration of the depth map being subtracted with a blurred depth map to create an approximate ambient occlusion texture.

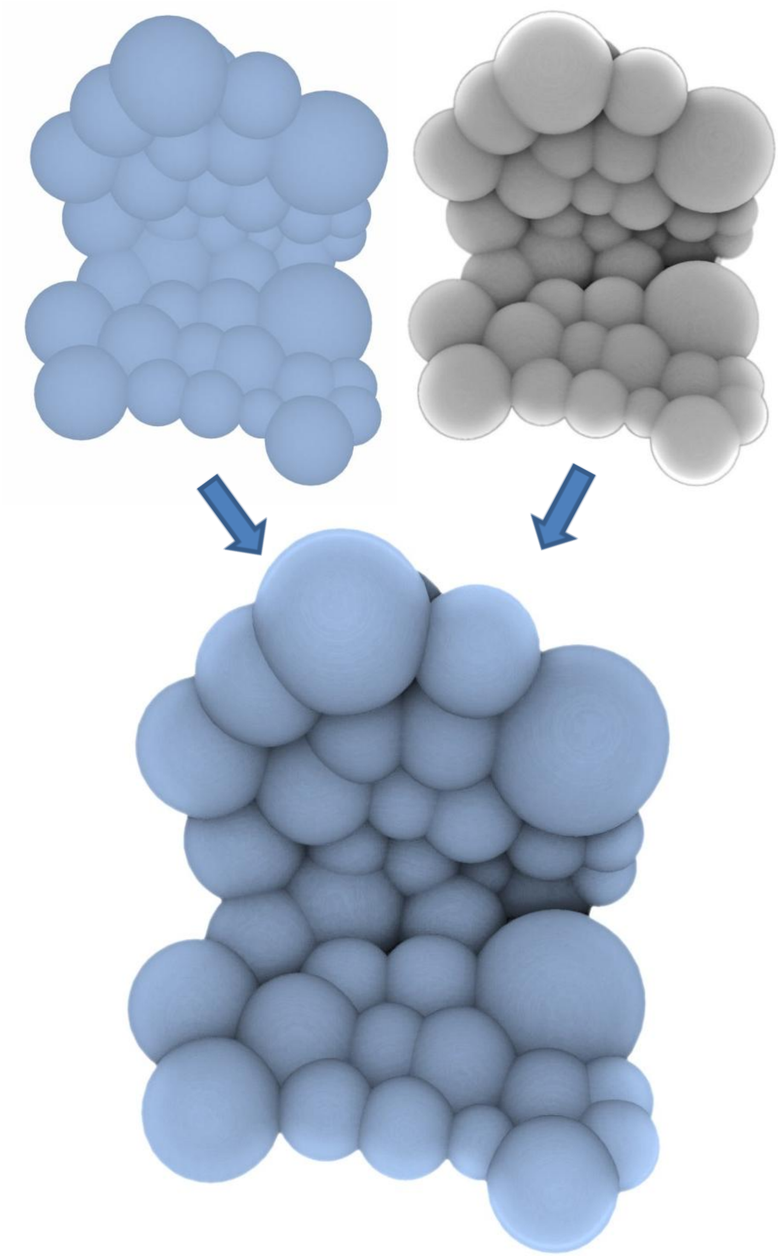


Figure 9.12: An example of combining the approximate ambient occlusion map with the shaded rendering of the 3D screen to create the approximate shadows.

Figure 9.12 shows how the flat-shaded model can be merged with the difference map to give the impression that certain areas are occluded. This process is performed within the Quest toolkit by having a semi-transparent overlay rendered on top of the basic scene rendering. The overlay multiplies the rendered 3D scene with the depth map such that darker areas of the overlay produce lower multiplier values and hence make that area of the 3D scene also appear dark.

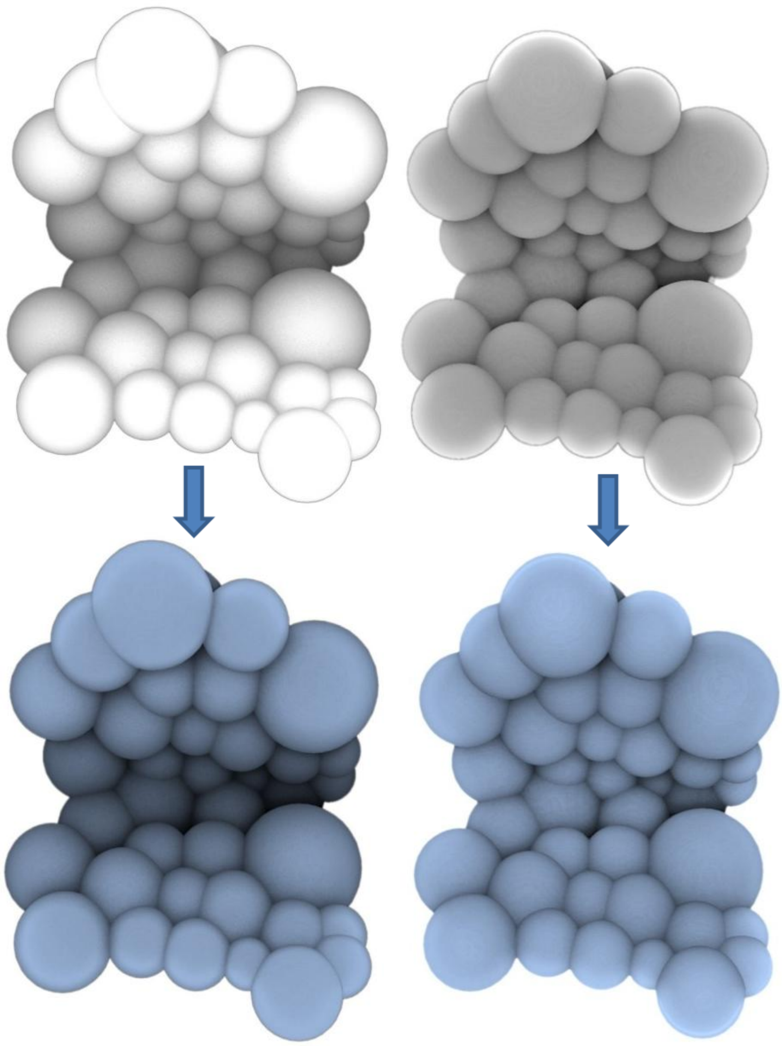


Figure 9.13: A comparison of SSAO (left) and the proposed ASSAO (right).

By looking at the resultant images (Figure 9.13), we see that the proposed Approximate Screen Space Ambient Occlusion (ASSAO) is unable to determine that the centre of the object should be darker, as it is occluded by the larger shape. It is, in effect, only able to show very local differences in depth rather than examining the image as a whole. However, the performance of the ASSAO is increased, requiring less time to render. A secondary advantage to this technique is that it does not require a high-end graphics card. In most typical games this inability to examine distant pixels would be more evident as objects such as tables with large areas of occlusion underneath would produce poor results. In the Scylla simulation, these drawbacks are less evident, as the general visual conditions make fine details difficult to see and its low processing load makes it a suitable alternative.

9.5 Screen Space Ambient Occlusion implementation

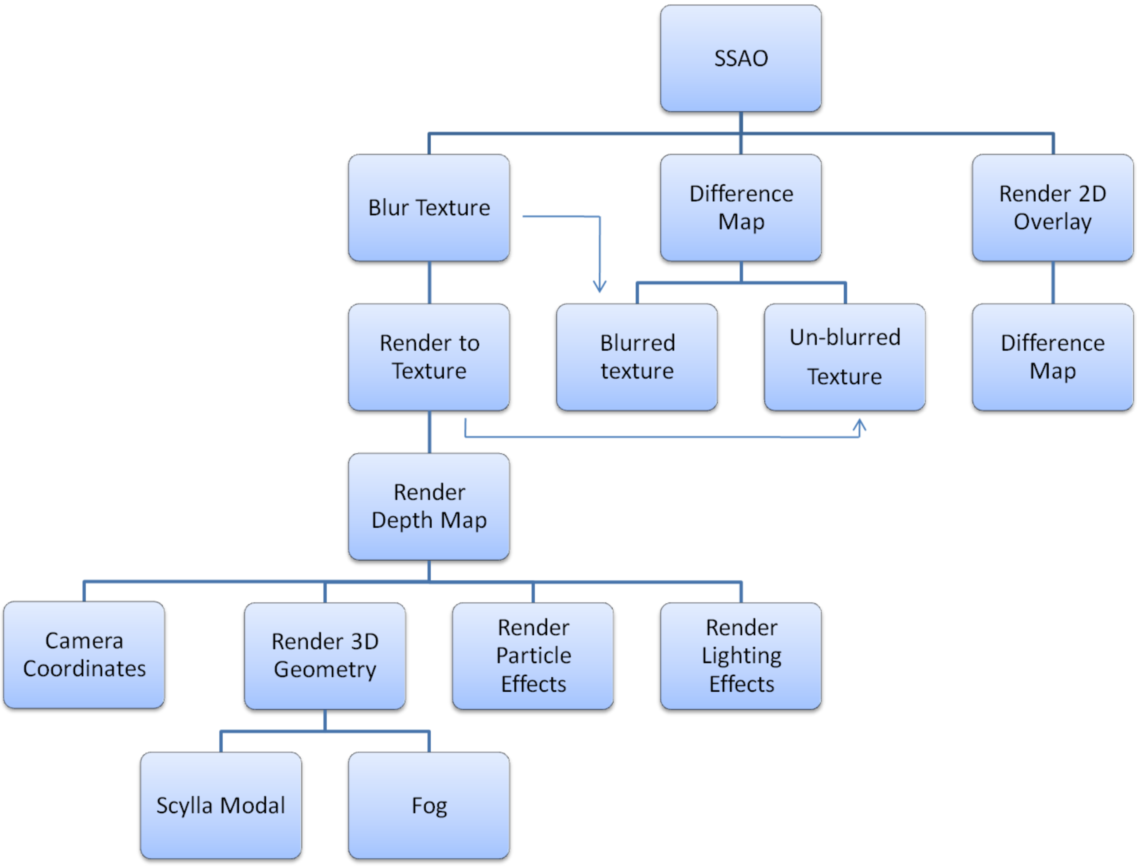


Figure 9.14: Hierarchical structure of the Ambient Lighting system (SSAO).

The final hierarchical Quest3D implementation of the proposed ASSAO is shown in figure 9.14, the depth map is initially created by rendering the scene with a dark fog layer. The resultant image will show objects at a greater distance to be darker. The depth map is then subtracted from a blurred version of itself to produce the difference map. Finally the deference map is rendered as a 2D overlay that is applied to the normal render of the scene.

9.6 Particle Effects

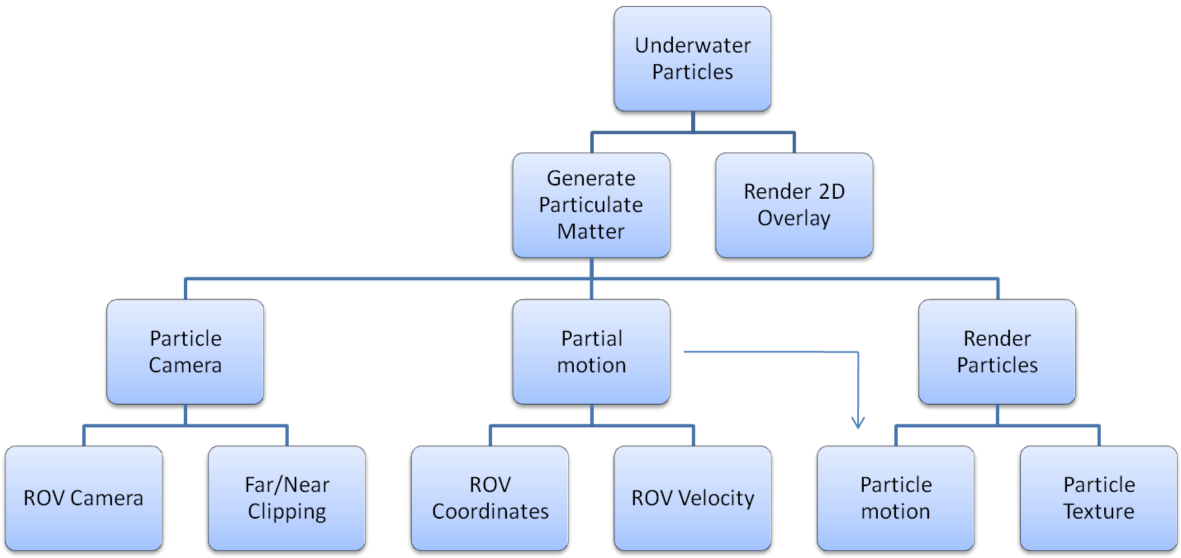
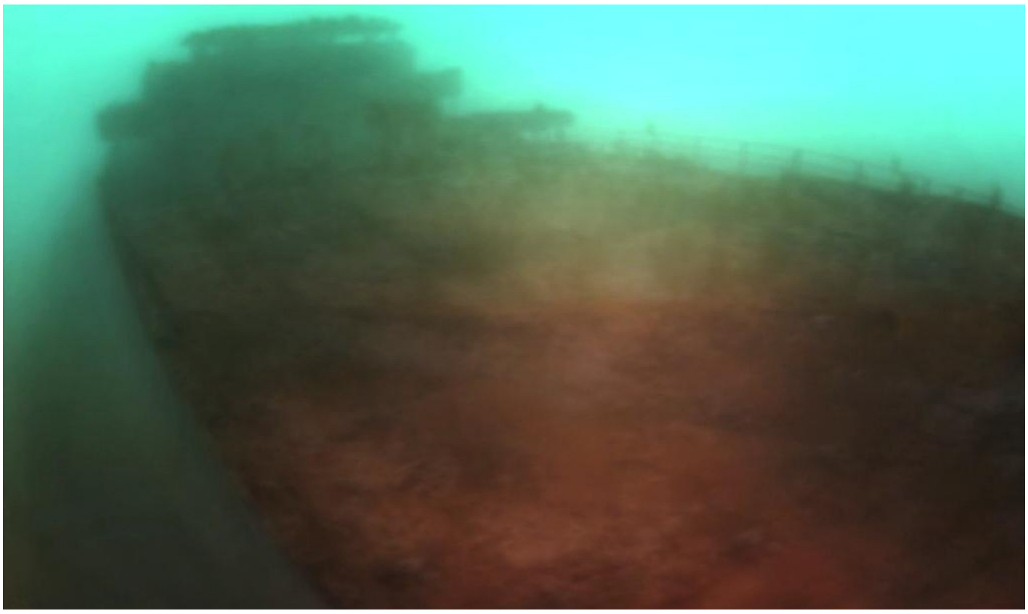


Figure 9.15: A simplified hierarchical structure of the underwater particle system developed for the Scylla experiment.

Unlike the clean water of the NMA's ExplorOcean tank, dense particulate matter is ever present in the open ocean. A detailed particle rendering system had to be created to support the simulations reported herein, in order to recreate this effect as accurately as possible, as this is one of the major reasons vision is restricted when deploying ROVs (figure 9.15). The simplest, and in many ways the most realistic way of displaying these particles is to render hundreds of thousands of them throughout the virtual environment. Certainly the result would be visually accurate, but the processing requirement would be huge. Another, more effective option is to create particles only at and around the location of the ROV as, due to the low visibility, only particles close to the user's perspective would be seen. However, this too has its problems. The particle emitter can just be set to the same position as the ROV but this alone would create the wrong visual effect – the particle volume would move with the position of the ROV. The ROV must appear as if it is travelling *through* the particle cloud and not as if it is moving with them. The solution adopted was to examine the ROV's current velocity and vector direction of travel and to create a force on the particles in exact opposition to it. This actually has the visual effect that the local particles are stationary as the ROV moves through them.

Even though 1000 particles are being rendered, this is still just a fraction of the amount of particulate matter found in the open ocean. To give the appearance of a denser cloud each of the computed particles actually represents a group of particles. Each part is made of a user-facing billboard with a semi-transparent texture. Instead of just rendering one particle, the texture is constructed of around a hundred textured particles. The overall effect gives the visual appearance of 100,000 particles. This was done in much the same way as the bubble streams in the ExplorOcean exhibit, but to a larger scale (figure 9.16).



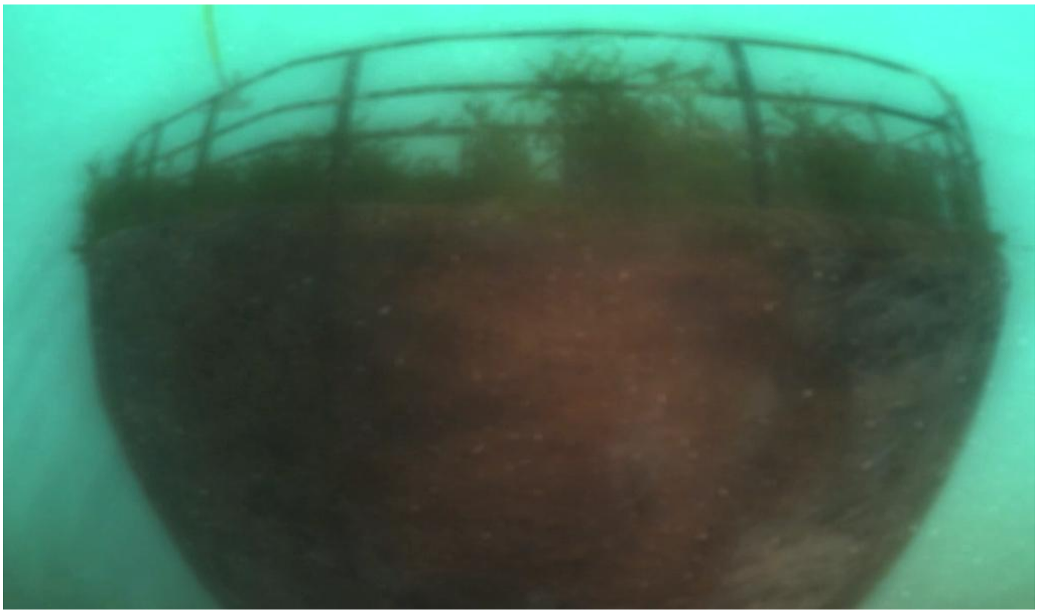


Figure 9.16: Images showing the dense particulate matter of the virtual Scylla simulator.

9.7 Display Design

There are many critical human factors issues that must be considered when designing display systems for vehicle use. When developing for remotely operated vehicles, the method used for designing information displays is even more critical due to the intrinsic disassociation of pilot and vehicle, as well as the decreased field of view available (usually one or two monitor screens).

Most of today's ROV information display systems have changed very little over the years, they typically show critical information such as altitude, heading and roll in a pictorial method much like traditional aircraft (Bell, Bayliss, and Warburton 1995). When designing displays, the pilot's frame of reference becomes an important factor to consider in terms of their spatial awareness (Howard 1989). For example, does the display represent the vehicle moving within an external world (exocentric) or does the world move around the vehicle (egocentric)? These two frames of reference, egocentric and exocentric, are also known more commonly in aviation terms as inside–out and outside-in respectively. The inside-out display method is very commonly used in aviation to illustrate the pitch and roll of a vehicle. This can be seen in Figure 9.17.

Figure 9.17: Shows two images of modern inside out displays showing the tilt or roll and banking of the aircraft in relation to the ground.

The red lines in Figure 9.17 represent the wings of the aircraft. It can be seen that they are fixed in position and it is the virtual horizon that banks from left to right. This type of illustrative display of data is relatively irrelevant for ROV flight as very few ROVs bank or roll. It is more important for ROV flight to accurately display heading and altitude (depth).

Figure 9.18: Shows examples of both analogue (left) and digital (right) method of showing altitude (depth).

Most traditional aircraft cockpits display altitude on a dial. However, modern aircraft with computer-based digital displays use an inside-out method. The arrow or needle remains in a static position with the scale moving up or down. This can be seen in figure 9.18.

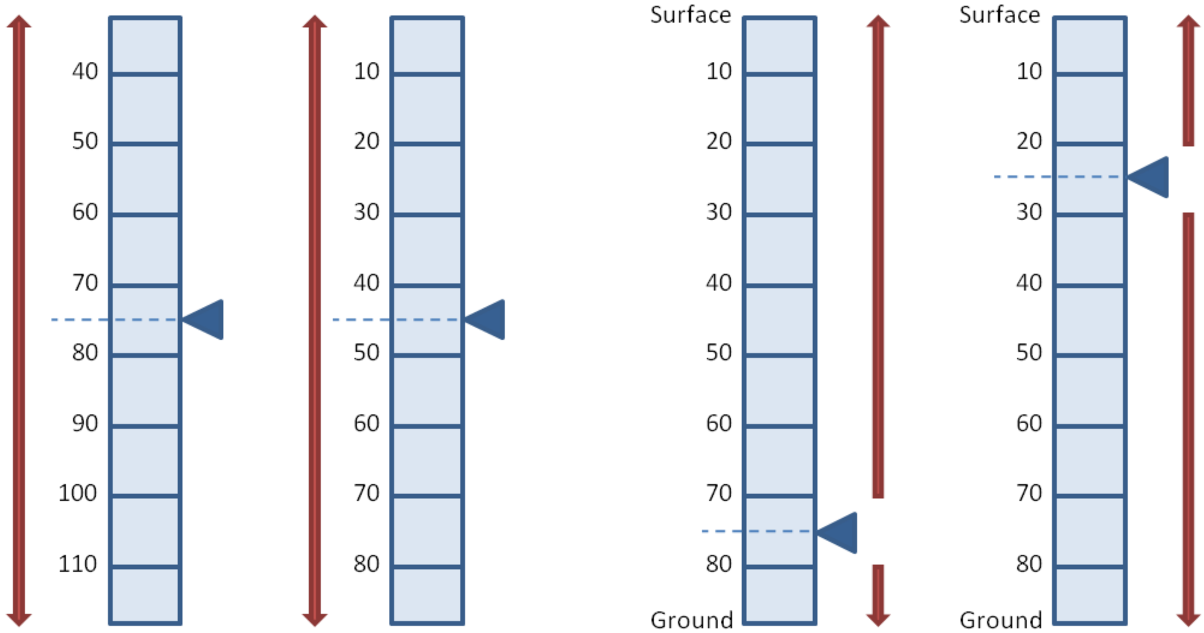


Figure 9.19: An illustration of a inside out altitude display (left) and its outside in equivalent (right).

Figure 9.19 shows how an outside-in interpretation of the data would look. Now, as the aircraft changes height, it is the needle that moves and the scale remains static and constant. The benefits of this outside-in view are that it is immediately clear where the vehicle is in relation to the surface.

A disadvantage to this is that the overall scale becomes very inaccurate when there is a large difference between the ground and highest altitude (i.e. between 0-30,000 feet). On this single scale, determining the difference between 20,000 feet and 21,000 feet would be very difficult. This is less of an issue for ROV displays as the typical depths are of the order of a few hundred meters. Again representations of both outside-in and inside-out systems for displaying heading are shown in Figure 9.20, below.

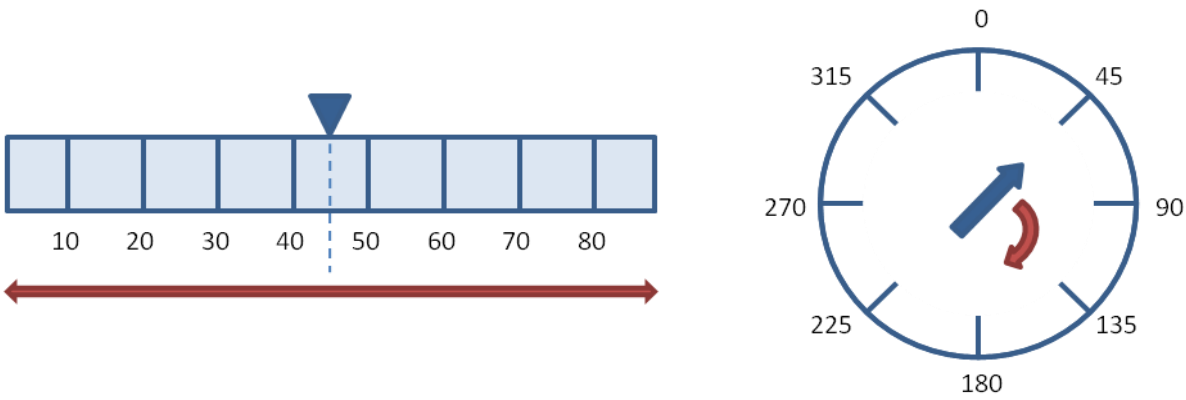


Figure 9.20: An illustration of a inside out bearing display (left) and an outside in equivalent (right).

For the Virtual Scylla experiment, displaying depth and heading information alone was not a requirement. The experiment set out to investigate how participants might use additional information about the underwater target objects to find them. Therefore the depth and heading systems must indicate the depth and heading of the ROV by the relationship between the ROV position and the target objects.

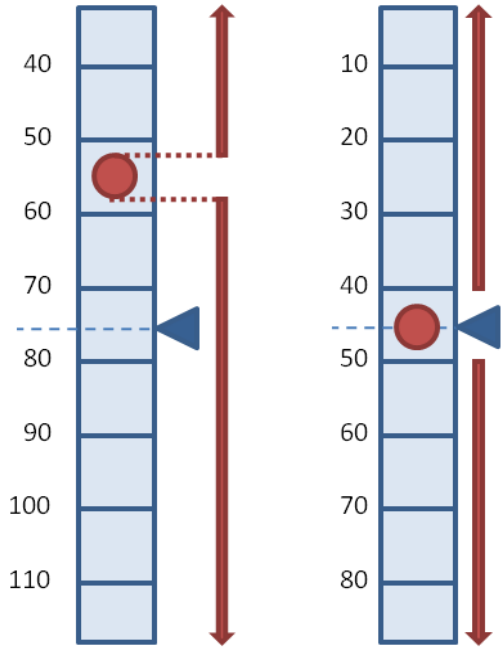


Figure 9.21: An illustration of the target depth location display in both inside out (left) and outside in (right) arrangements.

The depth gauge shown in Figure 9.21 shows the depth of the target in relation to the ROV's position. The red indicator represents the depth of the target. If the red indicator is above the fixed blue indicator arrow, then the pilot needs to raise the ROV. When the red and blue indicators align, then the ROV is at the same depth as the target (i.e. the ROV and target now share the same Z value in three-dimensional space). This style of depth indication was ultimately selected as this representation is similar to that of a spirit level (much like the air bubble in such a device, the red target should remain in the centre of the scale) and would be better suited to non ROV pilots

Most aircraft UAV and ROV heading display indicators are based on the inside-out method (Lintern, Roscoe, and Sivier 1990). With this in mind, the heading technical aid followed this design principle i.e. the ROV position marker is fixed and the target and world move around it. The target information was represented with a red marker that would indicate its location, direction and distance in relation to the ROV position (figure 9.22).

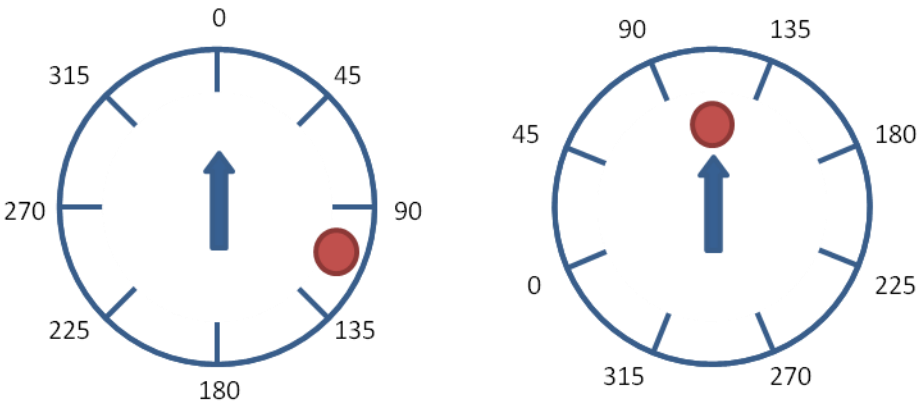


Figure 9.22: An illustration of the final design for the X,Y target bearing indicator.

As the distance between the target and the ROV is reduced, the red marker approaches the centre of the compass. Again, this is similar to a circular spirit level. Once the red and blue markers align, the ROV and target are at the same position. This can also be thought of as the ROV and target sharing the same X,Y value. There is one issue to consider with this set-up; the display is relatively small and it is difficult to represent very large and very small distances between the target and ROV. If the possible search area is 500m square, it would be difficult to see differences in a few meters. To better represent distances, the target position is represented on a logarithmic scale. In essence, as the ROV approaches the target, the red marker becomes ever more accurate. This is achieved by automatically zooming in to the local area if the target is close. In a real-world ROV search task, the exact target position would not be known to such a degree of accuracy, but for the purpose of determining the use of technical aids it is not important.

The second technical aid is a sonar scanner that is based on real-world commercial products. There are many types of sonar scanners available on the market today, each suited to a particular task. Sonar, in its simplest terms, consists of a signal generator and an electro-acoustic transducer (or receiver). The signal generator creates a pulse of sound that is usually referred to as a "ping". The transducer listens to any reflections or echoes and by a simple calculation using the speed (time) of sound in water, a distance-to-target can be calculated. Modern sonar uses multiple beam systems and arrays of transducers to create a surprisingly detailed image of the scanned area.

For modern ROV systems three types of sonar are typically used, 1D, 2D and 3D. 1D systems can only calculate the distance to target and are typically only used to calculate current distance to the ocean bed. This is only of importance to larger ships that run the risk of grounding. This simple 1D sonar can be expanded to 2D by giving it the ability to sweep an area, building up a 2D representation of the distance from all of the surrounding objects. 2D sonar systems are generally used for navigation; they can typically scan a distance of 5-75 meters at a 45 degree coverage angle up to a full 360 degree sweep (figure 9.23).

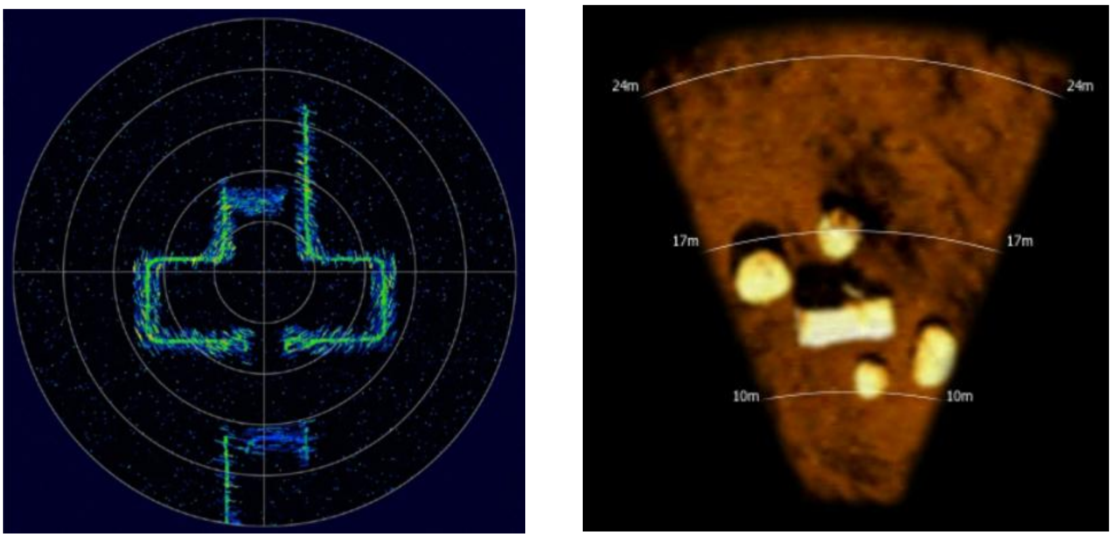


Figure 9.23: Two images showing 360 degree radar display[12] (left) and a focused 30 degree pass[12] (right).

[12] www.marinesimulation.com

Modern 3D sonar system can produce an almost video-quality image which is used for specific localised tasks, such as manipulator movement in low visibility or object inspection. While the image fidelity is high it can often be difficult to interpret for the novice user due to sonar shadows and the unusual view angles. 3D scans are usually confined to a very limited angle of sweep and have a limited scan distance.

As the task under investigation is more focused on the participants use of technical aid during search tasks and not of the ability to perform complex manipulator movements a 3D sonar scan is less relevant. Therefore a 2D sonar was selected as it is the best type for giving the pilot spatial awareness and its image output is relatively easy to interpret even to the novice pilot. As stated earlier, most 2D sonar scanners can have an adjustable scanning angle and distance. There are advantages and disadvantages to consider when selecting these values. While a longer scan range can detect objects at a greater distance, the resolution of the scan is compromised, resulting in an object being distorted or blurred.

The scanning angle can also effect the update time of the sonar and as stated earlier a 2D sonar is effectively just a 1D sonar that quickly sweeps across an area to build up an image. If the scan angle is increased then the sweep can take longer; this results in an image which lags behind the movements of the ROV. As the area to be scanned increases, the sonar sweep takes longer, leading to slow update time on the display.

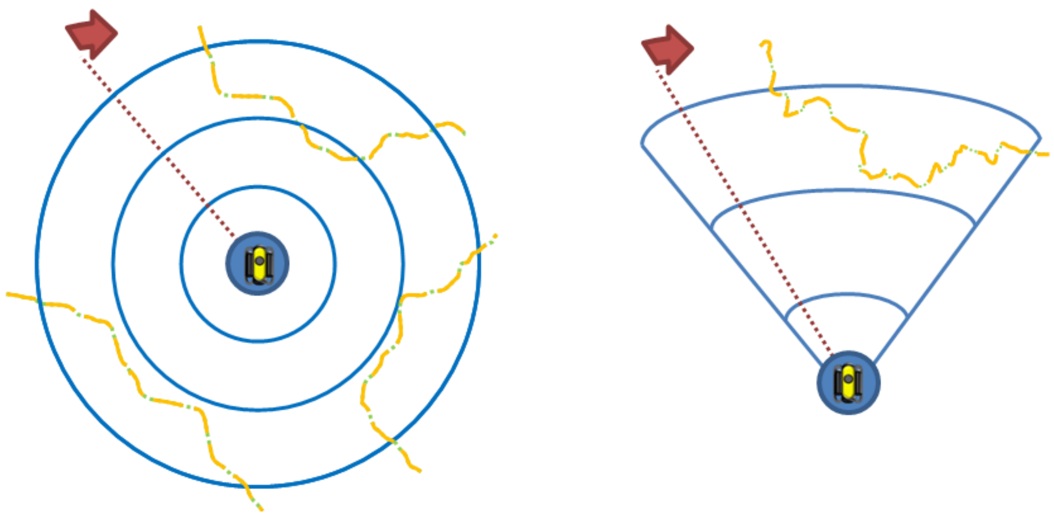


Figure 9.24: An illustration of the workings of a sonar system in 360 degree (left) and 30 degree (right) sweep mode.

A smaller scanning arc is able to produce a more detailed image which is updated more frequently however this does mean that only a partial image of the area can be seen so the pilot is unable to determine the environmental structures behind him (figure 9.24). A 360-degree sonar scan was selected for the simulation as a broader understanding of the surround structure is more important to navigation than focused details. As participants were going to be assessed (this was, probably, the first time the participants would have seen a sonar image), the decision was taken to make the sonar scan easier to interpret. As stated earlier, the sonar images can be slow to update, particularly when performing full 360-degree sweeps. This update can be very disorientating as the movement significantly lags behind the pilot's movement. Skilled pilots can adapt to this and are trained for it but with the limited trial time per user in the present study, there was not enough time to train out adverse responses to such a delay. For this reason, the sweep time has been reduced to zero and effectively the sonar display updates instantly.

The second alteration to typical 2D sonar operation is its scan penetration. Usually, once the "ping" has interacted with a solid object, the signal is reflected and can go no further. This effectively means that any object behind the solid object cannot receive the 'ping' signal and therefore cannot be seen (in sonar shadow). While within the decks of the Scylla a real-world sonar system would only display an image of the immediately surrounding walls. This would lead to a very limited view which would be difficult for the untrained eye to interpret. Once again, to shorten the time required for training the virtual sonar was given the ability to penetrate further into the structure giving the novice pilot a much better understanding of the surrounding environment. While nearly all modern sonar systems allow the user to adjust settings, such as range, angle and sweep speed 'on the fly', the virtual sonar has fixed values. This, once again, was done to limit the amount of pre-training and allow the novice pilot to proceed with the task.

The overall goal of the simulated sonar system, with its extended 'ping' penetration, effectively provides the participant with a localised map of the surrounding structures. As stated earlier, providing the user with some form of map significantly improves spatial awareness. This may be particularly important when navigating around the tight decks of the Scylla wreck. There are a number of ways in which a sonar system can be simulated. It is possible to create a physically accurate model of the sonar pulses and their respective returned echo waves. This method of simulation is typically used in sonar design as it can produce accurate measures to the effectiveness of a conceptual sonar system (Riordan, Omerdic, and Toal 2005). The drawback with a fully simulated physical system is its required processing power. This makes it a non-viable solution to real-time applications.

A games-based alternative is to continually check the distance from the centre of the ROV to the surrounding objects through a 360 degree arc and create an image of all the local collisions. This would be able to determine the distance of surrounding objects but the resolution of the image would be dependent on the number of calls made throughout the arc. For a reasonable quality of image, around 100 calls would be required. While this would be far quicker than a physically accurate model it would still have a significant impact on the simulation processing speed. An alternative method is presented here that makes use of the virtual cameras clipping planes and results in an almost instant high resolution image similar to that of sonar.

Every 3D Scene is viewed though a virtual camera, and just like its real counterpart it has variables for zoom and field of view. However, virtual cameras have an additional feature i.e. their clipping ranges. The clipping range determines the view distance of the camera, beyond this point nothing is rendered. This point is known as the far clipping plane. There is also a near clipping plane which determines the distance from the camera that close objects are not rendered. These clipping planes are usually used to prevent rendering far into the distance where the image resolution will fail to draw an object or to save processing time. Also unusual artefacts can appear if the objects are too close to the camera.

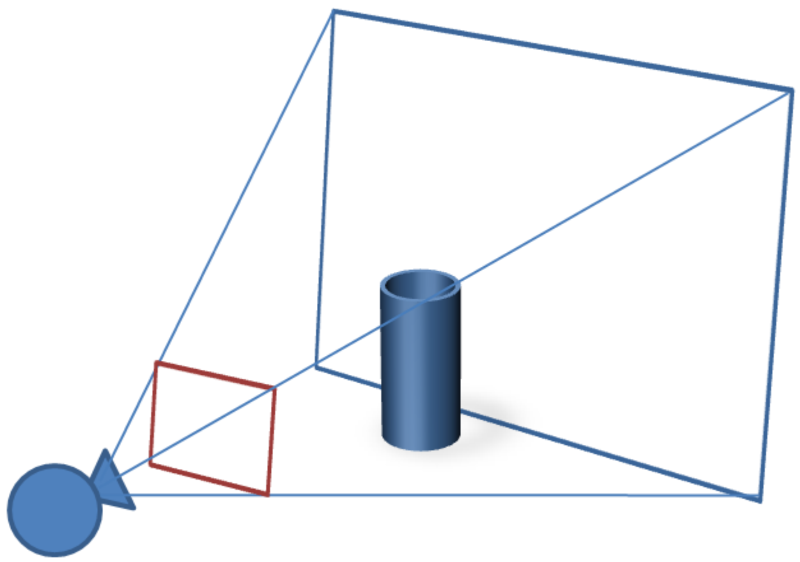


Figure 9.25: An illustration of the far (blue) and near (red) clipping planes for a virtual camera.

Figure 9.25 indicates the two clipping planes which run along the view angle of the virtual camera, as the cylinder resides between the near and far clipping plane it is rendered fully formed. Any object fully outside of the two planes will not be displayed. These clipping planes can now be used to advantage

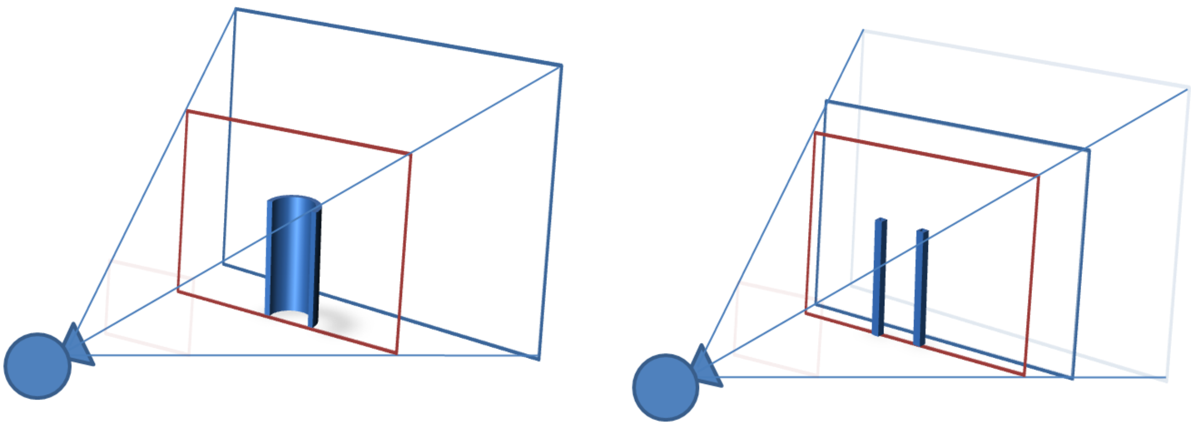


Figure 9.26: An illustration of how clipping planes can be set to cut through a virtual object.

In figure 9.26 the near clipping plane is moved further away from the camera so it bisects the cylinder. Any part of the cylinder before the near clipping plane is no longer rendered which results in the appearance that the cylinder has been cut in half. In figure 9.25 the far clipping plane is also moved but this time very close to the near clipping plane. Only a small amount of the cylinder resides between the near and far planes. The resulting rendered image shows a cross sectional slice of the cylinder geometry. This same method can be applied to the Virtual Scylla model to produce live cross sectional images of the decks. The only alteration that is required is for the camera to rotate so it focuses directly down onto the Scylla wreck.

By moving the clipping plane close together a cross section of the object is displayed. By simply rotating the camera to face directly down viewing the Scylla the clipping planes can be altered to give a live cross section of all of the decks

9.8 Sonar Display

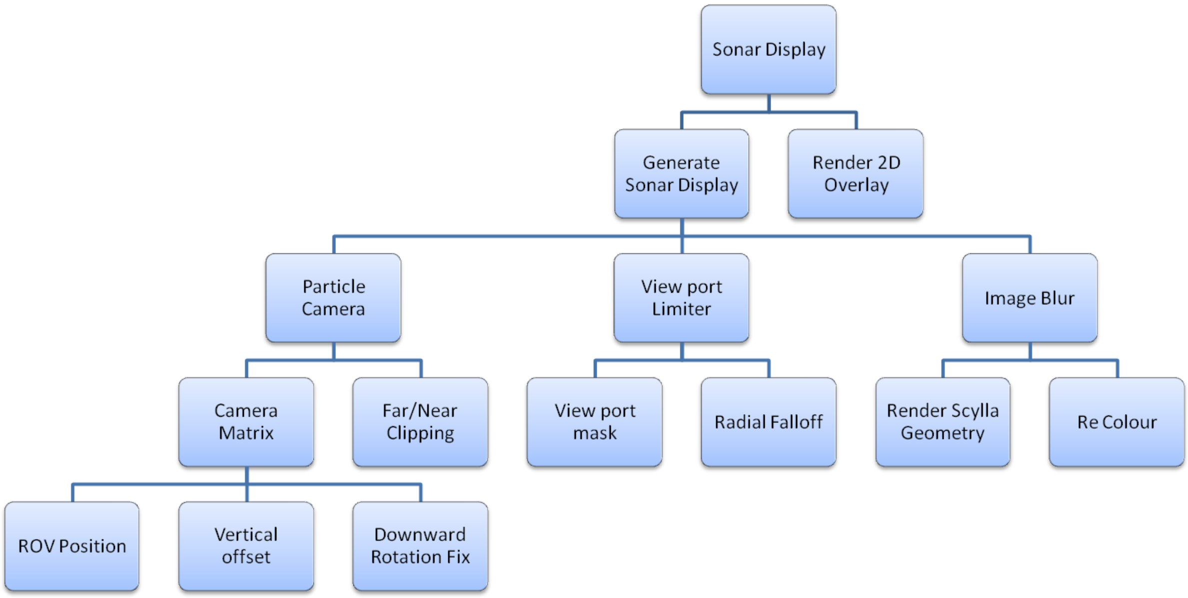


Figure 9.27: A simplified hierarchical structure of the Sonar display module for the Scylla experiment.

Figure 9.27 illustrates the program structure concerned with producing the virtual ROV sonar. The cross sectional clipping camera must also move along with the ROV to simulate a continually updating sonar scan. The ROV position can be used with an additional vertical offset and rotation to ensure the camera always faces down. As stated earlier the virtual sonar system has been given a far greater penetration depth than that of normal sonar. The top down clipped camera image, however, would give the sonar an infinite clipping depth. To reduce the extent of the sonar penetration an additional fall off mask is to be overlaid on top of the clipping image. The visual effect is closer to that of real sonar with the penetration depth being degraded from the centre of the scan. The final stage of the sonar simulation is to reduce the sharpness of the clipping image. Even the most expensive sonar systems do not provide a perfect image of the surrounding area. When a sonar ping is reflected imperfections in the medium in which it travels and localised interferences result in a blurred reflected image. To replicate the effect the received clipping image goes through a final stage of Gaussian blurring and re-colourisation resulting in an appropriate visual look of a real ROV sonar system.

9.9 Target Compass Display (X,Y Plot)

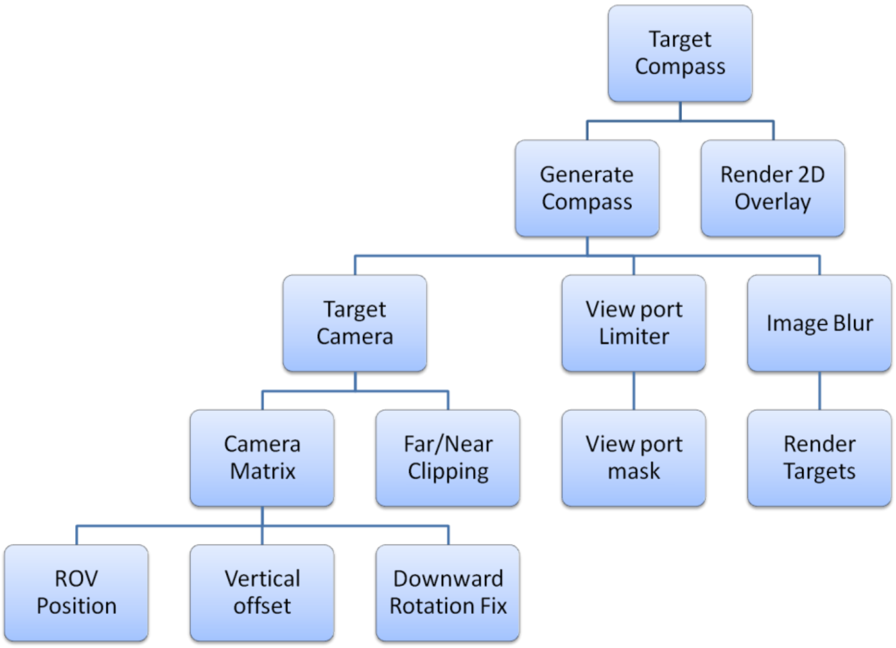


Figure 9.28: A simplified hierarchical structure of the Target compass system developed for the Scylla experiment.

The target compass (figure 9.28) is created by rendering the scene from an overhead view point looking down and drawing only the target markers. To create the overhead camera the ROV current position is used but with a vertical offset raising it overhead. The camera's orientation is changed and is rotated downwards. The camera now shows a top down view with the ROV at the centre. A mask is created to produce a circle, rather than square, overlay as well as the addition of distance rings for orientation. Only the target position markers are rendered not the entire scene.

9.10 Text Display

The final technical aids display was a simple text based information relating to the ROV movement. ROV depth, speed and bearing were displayed in simple white text typical of real world submersible displays

9.11 Final Rendering

Figures 9.29 and 9.30 show the final output from the virtual Scylla ROV simulator in both low and high fidelity. To create the low fidelity environment several of the visual effects were omitted from the high fidelity environment. Those effects were real-time ambient shadows, camera lens distortions, image blurring and high dynamic range lighting. The additional technical aids (text, depth, sonar and X,Y plot target compass) can also be seen in each corner of the display.

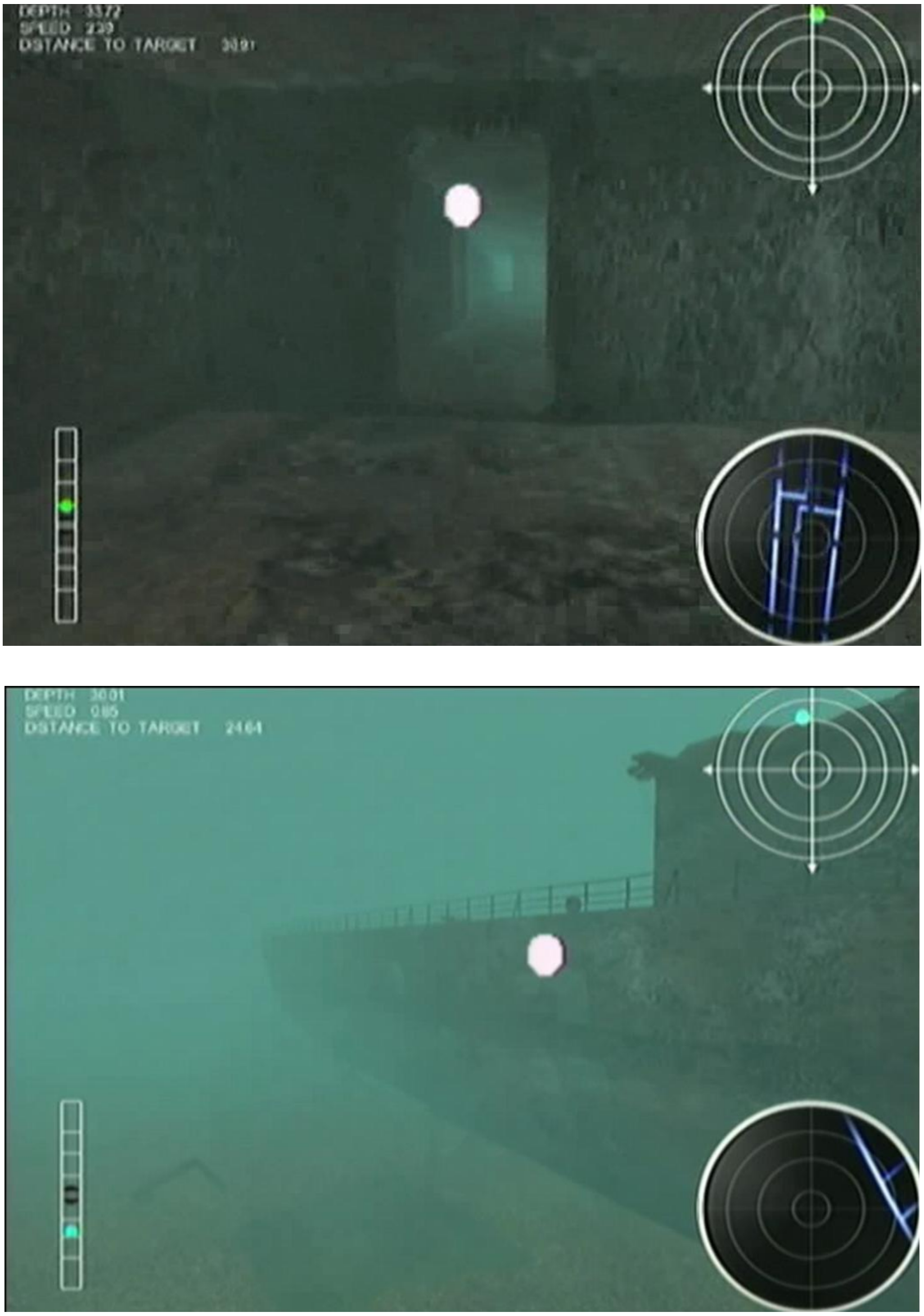


Figure 9.29: Images of the final low fidelity simulator with additional technical aids present.

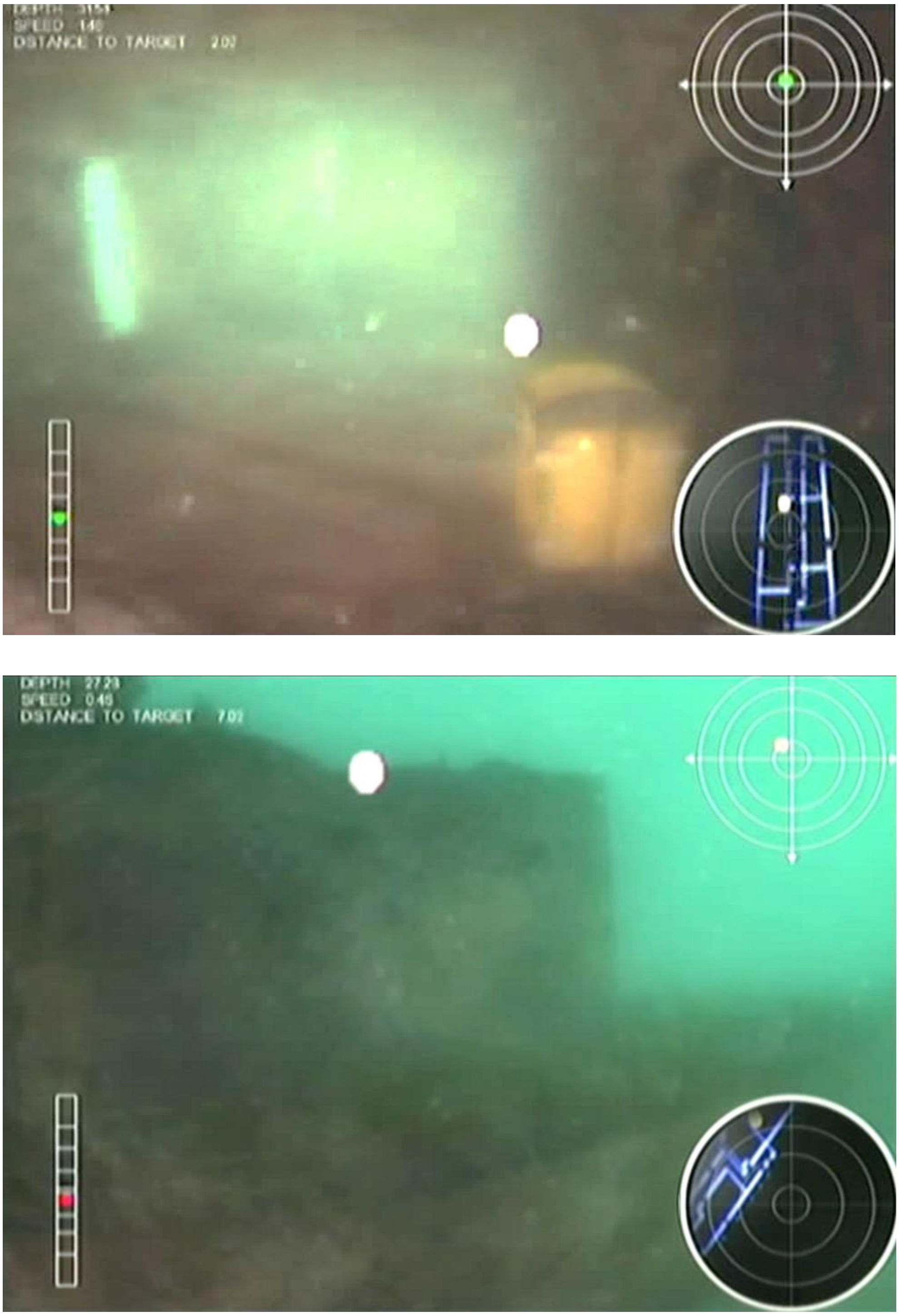


Figure 9.30: Images of the final high fidelity simulator with additional technical aids present.

9.12 Eye Tracking History and Uses

Eye tracking is the process of measuring the gaze direction of the eye as well as its movement. Eye tracking systems establish the movement of the eye by measuring the reflection of an infrared light emitting diode (LED), which generates, though illumination, a reflection off the surface of the eye. The infrared light causes the pupil to appear much brighter in contrast to the surrounding iris and this bright glint can easily be picked up by an infrared sensitive camera. It is this glint that eye tracking systems use for calibration and determining gaze. Eye tracking systems are used as a research method into psychology and cognitive understanding. The methodology of eye tracking is based on Just and Carpenter's "eye-mind" hypothesis: the location of a person's gaze corresponds to the most current thought in a person's mind (Just and Carpenter 1976). Eye tracking has been used heavily in usability testing, it is useful because it can be used to investigate behaviour that would otherwise be problematic to obtain through other more subjective measures (Karn, Ellis, and Juliano 1999; Jacob and Karn 2003).

Data collected from eye tracking systems is broken down into fixations and saccades which can be visually represented onscreen. Fixations occur when the eye is focused on a particular point on a screen and typically last between 250 and 500 milliseconds (Goldberg and Wichansky 2003). Saccades are fast movements of the eye which link together the fixations. When saccades and fixations are sequentially organized, they produce scanpaths. Analysing the scanpaths of participants enables the researcher to investigate the most basic thought processes being undertaken and can identify potential usability problems (Redline and Lankford 2001).

Although it could be argued that human thought processes might be better investigated through using surveys or verbalisation, eye tracking data provides more information than verbalisation alone and is more objective. Head-mounted eye tracking systems have successfully been used to study pilots' eye movements as they used cockpit controls and instruments (Fitts, Jones, and Milton 2005). These findings have led to cockpit redesigns and potentially reduced the likelihood of incidents caused by human error.

Eye tracking has also been extensively studied for its use in training, particularly in tasks where the user must utilise multiple information sources and awareness is critical. One such example is how eye tracking has been shown to be useful in laparoscopic surgery. A study conducted in 2004 showed there was a significant difference in the eye gaze patterns of novice surgeons to that of experts (Law *et al.* 2004). According to the research findings novice surgeons tended to follow the end of the tool, where-as experts focused their gaze more on the target object and achieved better performance results. It was suggested that eye tracking could be introduced into training to assess the skills of surgeons.

Eye tracking has further been used to evaluate training in visual inspection tasks such as the work performed by Duchowski where an aircraft cargo bay was virtually recreated which could be viewed using a head mounted display and basic inspection tasks could be performed (Duchowski *et al.* 2000). By utilising an eye tracking setup it was possible to get a better insight into why participant performance was increasing during the training. It was found that, while the mean duration of fixation times did not change overall, the number of fixations did reduce post-training. The authors conclude that an improved visual search strategy was employed, reducing the time required to visually rest on particular features.

While little research has been performed on the specifics of using eye tracking to assess ROV piloting skills, it has been used extensively in the area of aviation as studies have shown that awareness errors could be attributed to the operator's failure to perceive the situation correctly (Jones and Endsley 1996). Therefore, understating how experienced and novice pilots focus gaze and attention during a task is critical during training.

One of the most complex tasks performed by pilots is approach and landing procedures and this was the focus of a 2001 study into the display and instrument panel scanning behaviour of pilots (Anders 2001). The work involved an A330 full-flight simulator and combine head and eye tracking system to provide a point of gaze. Nine areas of interest were used to examine the pilots scanning paths including the primary navigation, warning and primary flight display. The eye tracking allowed monitoring of the pilot's panel scanning behaviour transparent to external observers. The additional recorded data could be used in pilot training to demonstrate good or bad examples of instrument scanning as well as understanding how certain equipment and technical aids are used during complex flight manoeuvres. The work concluded that performance indicators could be derived, such as a minimum viewing requirement for the primary flight display and be used as a good indicator for evaluating performance. Eye tracking has therefore been proven to be useful tool in allowing trainers to retrospectively analyse a trainee's performances in awareness of complex aircraft instrument panels.

9.13 Eye Tracking Systems Evaluation

Modern eye tracking systems usually work by having a fixed camera focused closely on infrared light reflecting off the eye, from this the pupils movements can be determined using image processing techniques. For the Scylla experiment two potential eye trackers were investigated in order to determine the most suitable.

The first was the Applied Science Laboratories Mobile Eye[9], the system consists of head mounted eye tracking optics (modified glasses with a disk reflector), colour camera, a portable digital recorder and a PC for later data analysis (figure 9.31).

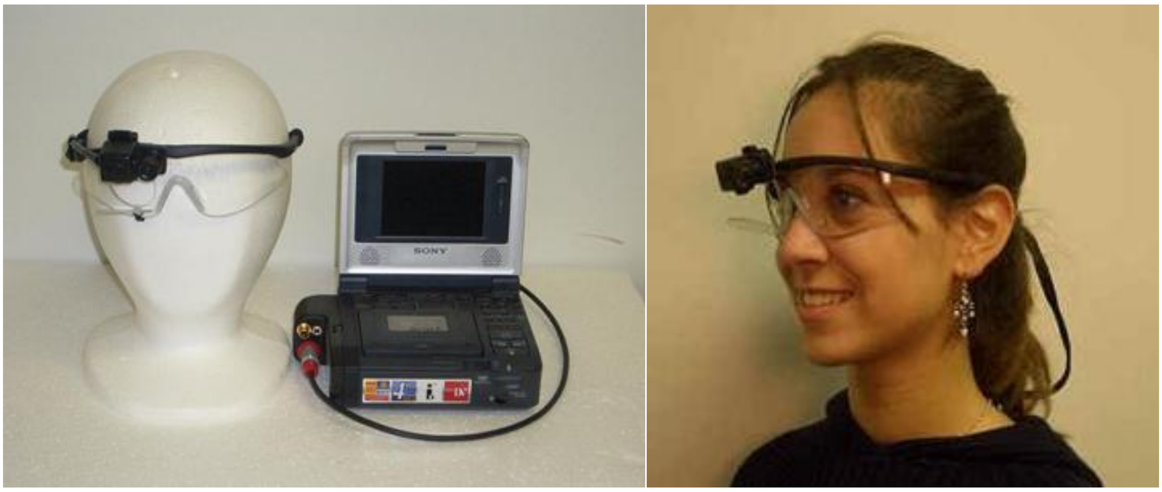


Figure 9.31: Images of the Applied Science Laboratories Mobile Eye tracking system[13].

The system is relatively lightweight and comfortable to wear and is also extremely portable. The disadvantage of the system is that it only tracks one eye and does not record head movement. It was also found that calibration could easily slip out of alignment if the reflecting optics were moved in anyway. The mobile eye is really a design for field experiments and lacks detailed eye movement capture because of this (figure 9.32).

[13] www.asleyetracking.com

Figure 9.32: An image of the Applied Science Laboratories Mobile Eye tracking system[14].

The second system to be investigated was the Eyelink II (developed by SR Research[16]), a fixed screen based system capable of tracking both eye moments as well as head movements (figure 9.33).

Figure 9.33: Images of the Eyelink II head mounted tracking system[15].

[14] www.asleyetracking.com
[15] www.sr-research.com

The Eyelink II requires considerably more equipment; the basic setup involves two PCs, one to track the eye movements and one for the experiments to be performed. In addition to the PC's, two video conversion boxes are required to merge and capture the visual display of the experiment computer along with the host pc eye tracking cursor (figure 9.34).

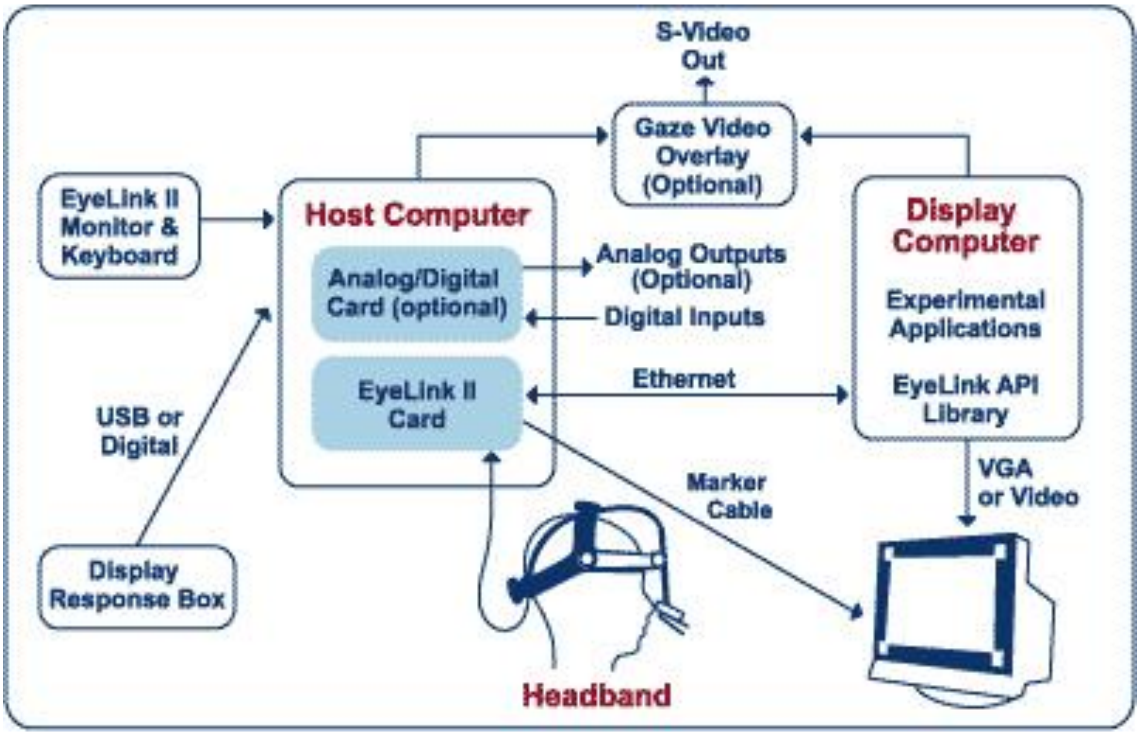


Figure 9.34: An illustration of the Eyelink II head mounted tracking system connection diagram[16].

It consists of three small video cameras mounted on a padded headband. Although the head-mounted apparatus may be considered cumbersome by current technology standards, the optical head-tracking camera integrated into the headband allows for accurate tracking of the subject's point of gaze without the need of head immobilization. The Eyelink II has several advantages over the Mobile Eye system, its capable of tracking both eyes; it can create a more accurate result by incorporating head tracking and is designed specifically to work with computer displays. For these reasons it was selected for the experiments.

[16] www.sr-research.com

9.14 Calibration

Calibration is used to collect fixations on target points, in order to map raw eye data to gaze position. Targets are presented for the participant to fixate on the Display PC while feedback graphics are presented to the experimenter on this display. The calibration is automatically checked when finished, and diagnostics given (figure 9.35).

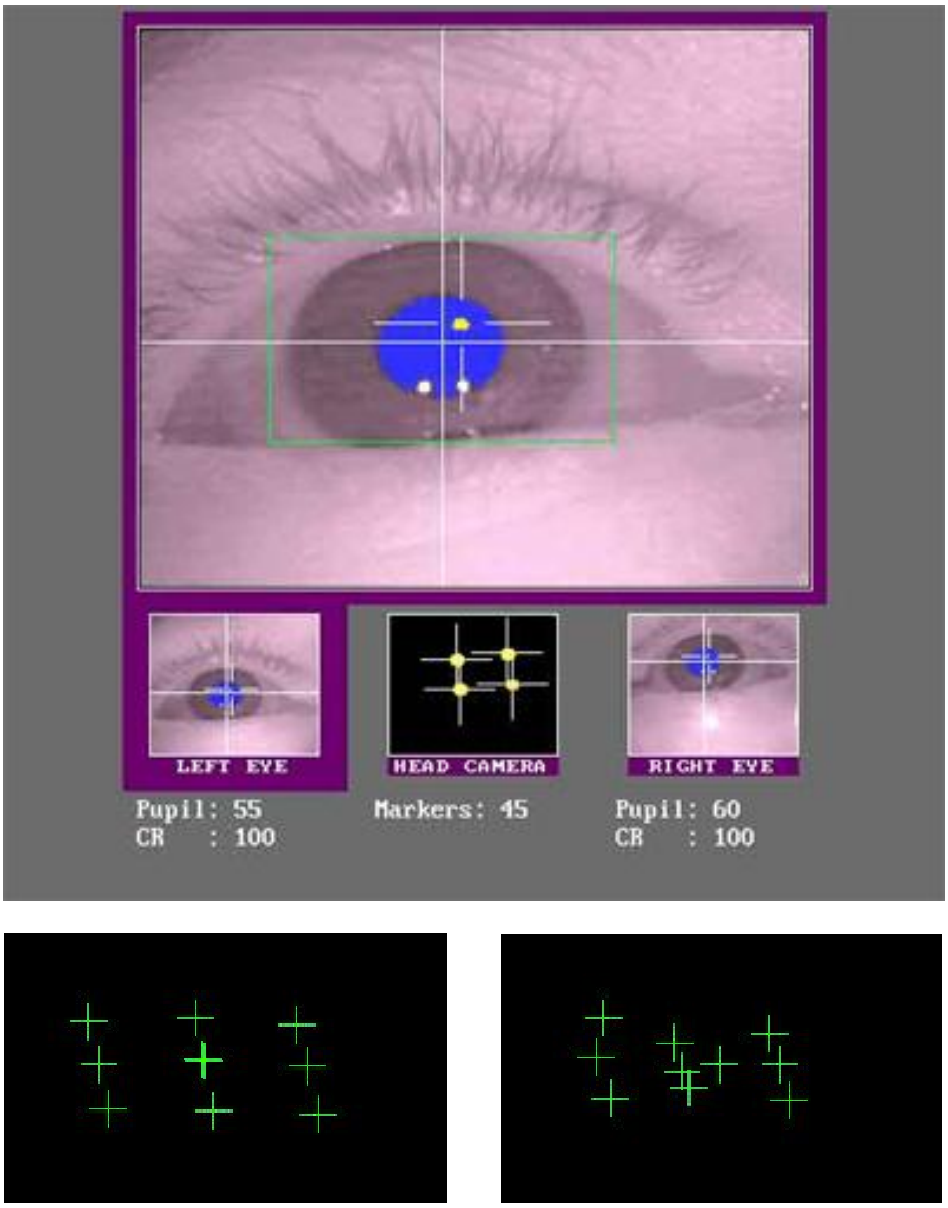


Figure 9.35: Screen images of the Eyelink II calibration screens.

9.15 Discussion

Several new techniques have been developed to push current gaming technologies to a sufficient enough level that can reflect the visual effects present in real world ROV flight. Ambient shadows, high dynamic lighting and optical distortions such as 'fish eye' lens effects have yet to be included in commercial ROV simulators. Suitable technical aids have also been developed to approximate the information presented to the user in the real world. While information displays are often included in many commercial ROV simulators it is not currently clear how these additional sources of information are used depending on the visual fidelity of the simulator. The subsequent experimentation will be able to evaluate with the exclusion of these visual features have any effect on the performance of the task or the use of the available technical aids.

To get the most accurate measurement of technical aid usage eye tracking will be used. Eye tracking is a common method to establish a person's most immediate thought. It has been used extensively for many 2D application such as evaluating web pages (Granka, Joachims, and Gay 2004) and evaluating software interfaces (Goldberg and Kotval 1999). It has been used in 3D virtual environment but primarily as a control device with limited success (Ellis *et al.* 2002). Only recently has it been used in 3D games to assess the visual attention of the user (Sennersten 2008) and it has been noted that more research in the this area is needed to formulate a model of visual attention in 3D games to enhance game level design and rendering (El-Nasr and Yan 2006).

With the incorporation of an eye tracking system with the developed simulation it is possible to accurately study affect of fidelity of task performance and the usage of technical search aids.

Chapter Ten

This chapter describes the method of experimentation and results of the final ROV study. Its primary goal is to assess the usage of technical aids during ROV search with varying levels of visual fidelity.

10.1 Experimental design

The experiment was designed to investigate whether visual fidelity of a ROV simulation has any affect on the user's dependency on technical search aids. Also we considered whether different search strategies are employed, depending on the technical aids used.

The experiment requires participants to locate three target objects hidden throughout the simulated Scylla wreck. The participants will be split into two groups, one experiencing the high fidelity and the second group using the low fidelity simulation. In order to establish the usage of each technical aid during the task eye tracking will be used. The software also records course taken, time to each target and number of collisions.

10.2 Participants

Twenty subjects took part in the experiment, they were all university engineering students aged between 18 and 22 years with an average age of 19. Out of the twenty participants 16 were male and 4 female.

10.3 Method

Before the eye tracking system was fitted and calibrated each participant was first shown the use of the replica control system and the four head up display search aids. A training level had been created in order for the participants to become accustomed to using the controls and how the ROV and target position were displayed on the technical aids. Once the participant was satisfied with the aims of the study and with the workings of the system, the eye tracking equipment was placed on their head and the calibration procedure was performed.

Upon successful calibration (error rating less than 4%) the experiment could begin. The task involved finding three targets of a classic treasure chest design hidden within the virtual Scylla wreck under two different conditions. The two conditions where high fidelity and low fidelity, as explained earlier. The order in which a participant would undergo each condition was alternated to ensure a balanced test.

The targets were placed in such a way that their locations would require the significant use of additional search aids. The locations were also chosen so as to ensure the need to navigate through multiple decks at close quarters. Once a target had been reached, an audio cue would be triggered stating "Target found, proceed to next target" as well as visual on screen text of the same message. At this point the technical aids would change the target colour and begin showing the new position. Upon reaching the final target, the audio cue would state that the final target had been reached and the mission was complete. Before the second condition is performed the eye tracking system is recalibrated to ensure the most accurate result. Once both conditions are performed, the eye tracking equipment is removed and the participant is asked to fill out two simple questionnaire sheets relating to their impression of the use of the technical aids during each condition.

The questionnaire was based on a simple rating scale with the participant simply marking on the scale the amount of time they believed they had used each technical aid and the main view screen.

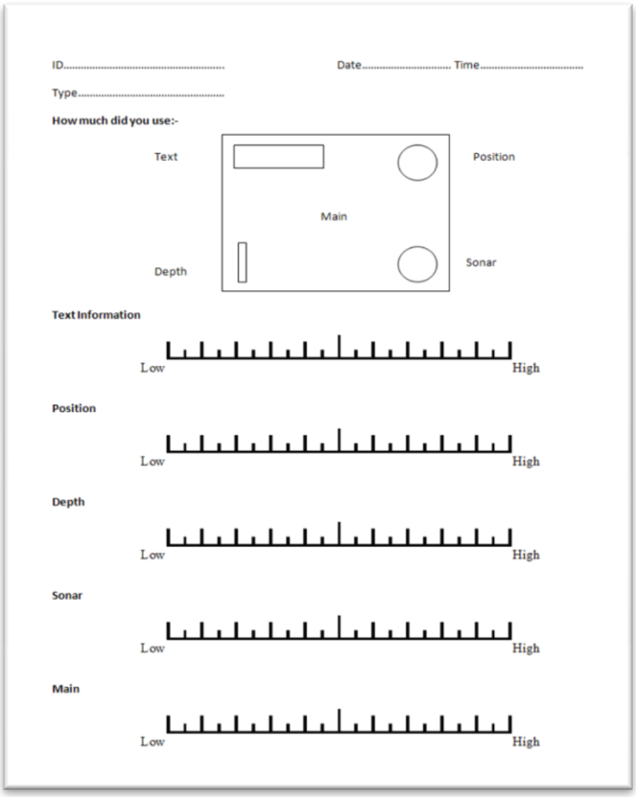


Figure 10.1: An illustration of the questionnaire used to sample the participants perceived use of the technical aids

The rating scale for the questionnaire (figure 10.1) was based on the NASA TLX format and comprised of five scales for each technical aid (depth gauge position, sonar and text information) and the main display (see appendix E). Each of the five scales is split into 21 gradations rated 'Low' to 'High'. The participant is simply asked to indicate, anywhere on the scale, their assessment of the use of each display during the high and low conditions.

10.4 Results

The Scylla search experiments results are split into four sections; timing data, collision data, eye tracking and search path data. An Analysis Of Variance (ANOVA) is used for all conditions as well as pre-planned paired comparisons for differences between individual target location times and technical aid usage between conditions.

10.5 Timing Data

We will begin with presenting the results from the participants' course times. The course time has been split into three sections, one for each target.

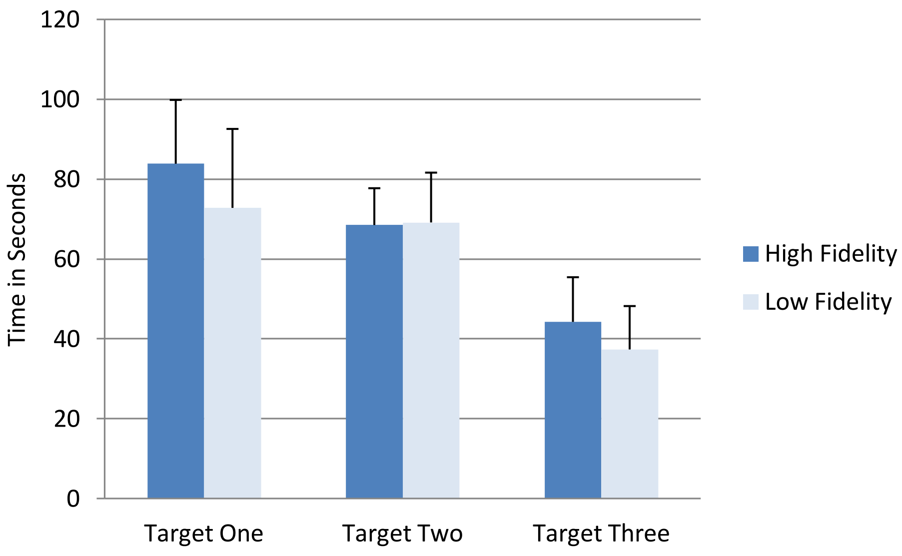


Figure 10.2: A graph showing the mean time taken to reach each target for both high and low fidelity conditions. Error bars indicate the standard deviation.

	Target One	Target Two	Target Three
High fidelity	86.1	68.15	42.3
ST Dev	19.97	11.43	10.10
Low Fidelity	61.85	66.8	35.4
ST Dev	13.35	10.12	8.80
T-Value	3.526	0.368	1.94
P-Value	0.002	0.717	0.066

Table 10:1 Shows the mean time, standard deviation (ST DEV), t and p values for each t-test of the three targets for both high and low fidelity conditions.

By examining the recorded time to target data from the experiment (figure 10.2) we can see there is a small but significant difference between the high and low fidelity conditions for the first target [$t(19) = 3.526$, $p = 0.002$] (table 10.1). Participants presented with the high fidelity simulation had, on average, taken 88 seconds while participants using the low fidelity simulation found the first target in just over a minute. For the second target the times for both conditions are almost the same [$t(19) = -0.368$, $p = 0.717$]. Finally we see that for the third target the low fidelity condition is slightly quicker and again, significant [$t(19) = 1.94$, $p = 0.066$].

While the Virtual Scylla experiment was primarily designed to evaluate technical aid usage it is also of interest how the visual fidelity would affect a participant's performance. As with the ExploreOcean simulation, the time to complete the task is a common metric used to assess performance. The results show that, when participants perform the task on the low fidelity condition, overall they do perform better. However this difference is very small and less than expected. With the effectively clearer display of the low fidelity condition, it would be reasonable to assume that the performance should be much greater. This is particularly evident when considering the second target which, on average, was found no quicker that the high fidelity condition. This may suggest that participants using the low fidelity condition had taken a less efficient path to the target.

Both the speed and number of collisions were also recorded by the simulator for each participant (figure 10.3). We can see for both speed and collisions there is a significant difference between the two conditions [$t(19) = -0.760$, $p<0.0001$] [$t(19) = -1.77$, $p = 0.033$] respectively (table 10.2). The high fidelity condition, on average, produces 22 collisions for the whole task, while the low fidelity condition tended to lead to more collisions i.e. 26. This may be due to the overall increase in average speed. Increased speed would lead to the need for increased reaction time for the participants to avoid obstacles. It would seem that while the simplistic visuals seem to encourage faster speeds it comes at the cost of increased collisions.

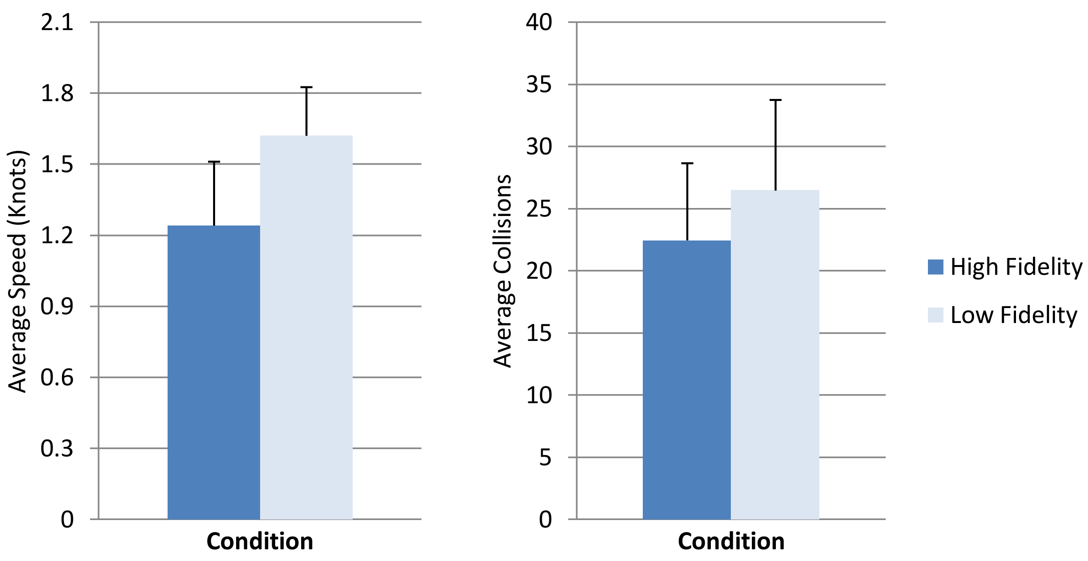


Figure 10.3: A graph showing the mean speed (left) and a graph showing the mean Collisions (right) for both high and low fidelity conditions. Error bars indicate the standard deviation.

	Average Speed
High Fidelity	1.2405
ST Dev	0.27
Low Fidelity	1.62
ST Dev	0.21
T-Value	-3.865
P-Value	<0.001

	Average Collisions
High Fidelity	22.45
ST Dev	6.20
Low Fidelity	26.45
ST Dev	7.29
T-Value	-1.595
P-Value	0.127

Table 10.2: Shows the mean, standard deviation (ST DEV), t and p values for each t-test comparing speed and collision data for both high and low fidelity conditions.

This analysis was necessary to evaluate if there is any effect on the average speed and whether a particular condition changed the amount of collisions performed. The results show an increase in overall speed for the low fidelity condition, which is most likely due to the ability to see further with the main display;- it also shows that that the increased speed came at the cost of increased collisions.

10.6 Eye tracking

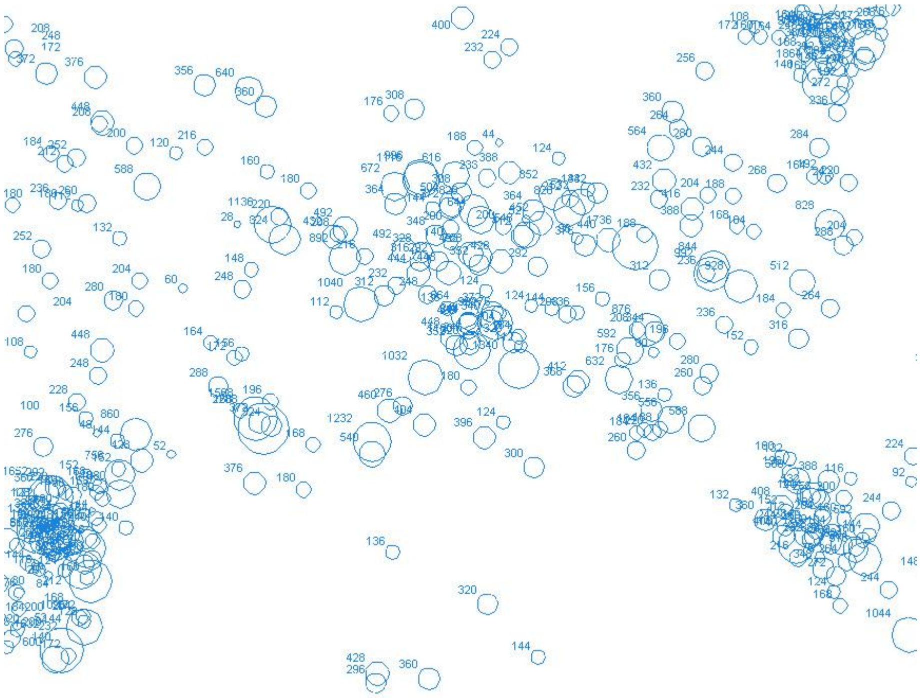
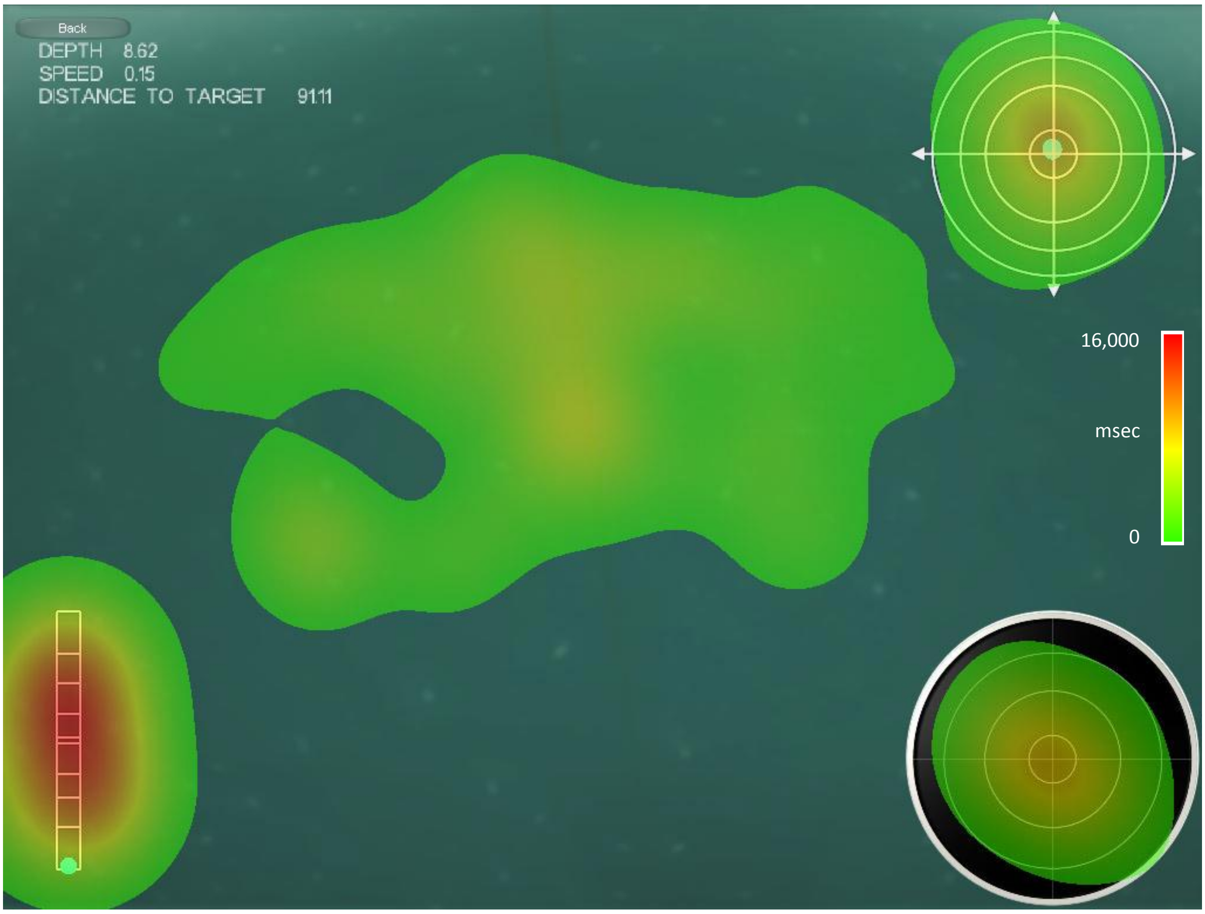


Figure 10.4: An example of the data obtained from the Eyelink II system for the high fidelity condition.

Figure 10.4 illustrates the raw gaze map received from the Eyelink 2 software. Each circle indicates a gaze (determined buy a 0.2 ms threshold) while the size of the circle represents the amount of time that gaze was held for. Figure 10.5 is a representation of the average screen use of all participants using the high fidelity ROV simulation. It shows, through a "heat map" style image, the amount of attention each display had during the task.

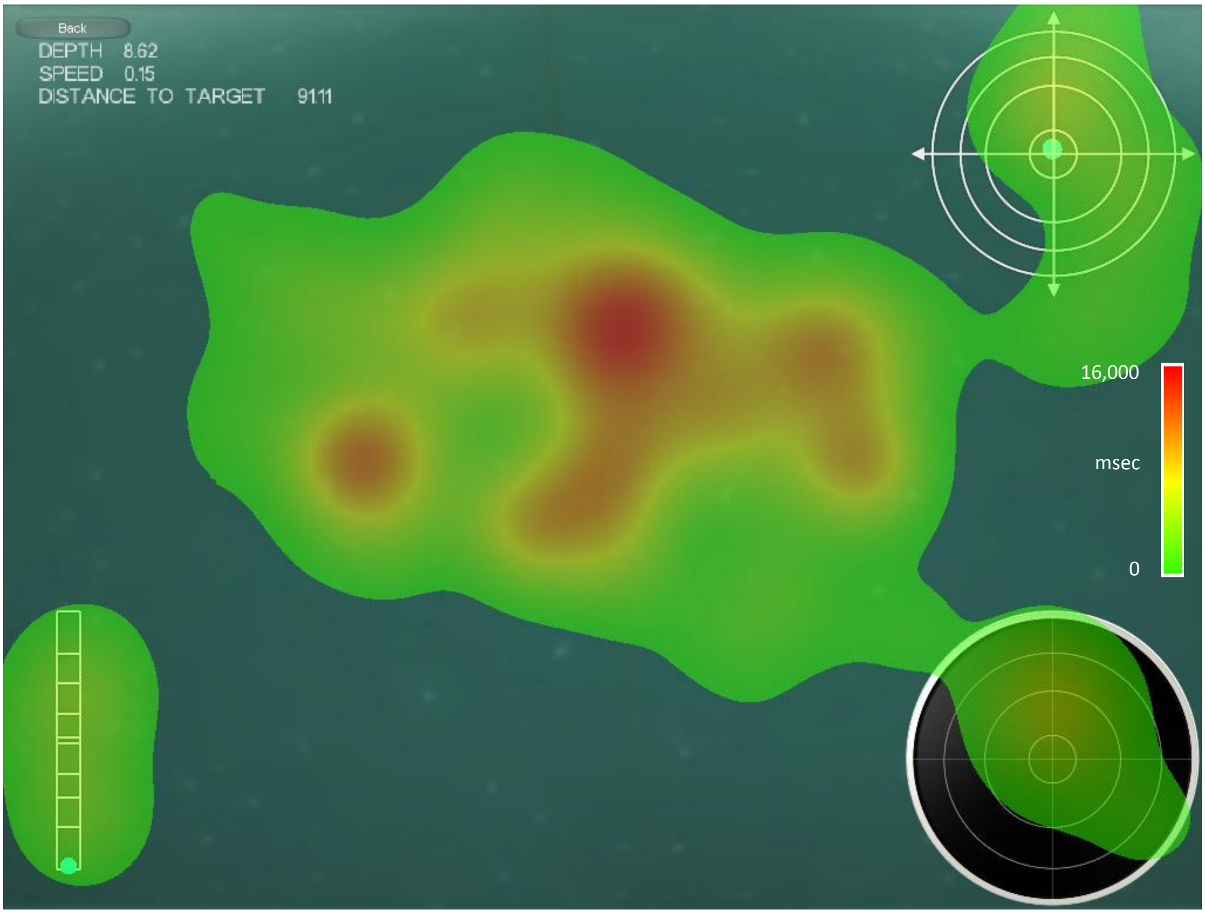
The heat map is created by the DataView software from SR research. To calculate the heat map each fixation recorded during the trial has a 2D Gaussian applied to it with its centre at the fixation location. The height of the Gaussian is weighted by the duration of the individual fixation and the width of the Gaussian is defined by an adjustable sigma value (usually set at 0.5). This process is repeated for all fixations and the resultant map is normalised. Once normalised, any fixation under 10% of the total time is not drawn.



Text Gaze (%)	Depth Gaze (%)	X,Y Gaze (%)	Sonar Gaze (%)	Video Gaze (%)
2.63	24.37	13.55	10.0	49.4

Figure 10.5: An example heat map representation of the high fidelity simulation results. Also below mean averages for technical aids used for all participants when using the high fidelity condition.

Red indicates high levels of visual attention, while green indicates low levels of attention. The depth gauge receives the most amount of attention during the task other than the main screen. Typically users would remain focused directly on the depth gauge while lining up the correct depth. The users would visit the X,Y plot just as frequently as the depth gauge, but would take short stares at it.



Text Gaze (%)	Depth Gaze (%)	X,Y Gaze (%)	Sonar Gaze (%)	Video Gaze (%)
1.72	15.669	6.04	5.95	70.60

Figure 10.6: An example heat map representation of the low fidelity simulation results. Also below mean averages for technical aids used for all participants when using the low fidelity condition. .

Figure 10.6 shows the heat map representation of average display use, but now for the low fidelity condition. It can be clearly seen that there is an increase in the use of the main screen through the trial. The visual heat map shows a very clear difference between the low and high fidelity simulations. This was further analysed statistically to determine whether there was any significant difference. An ANOVA was performed on the two conditions to establish whether this was significant. The results indicates that there is a significant main effect in both the fidelity [$F(1,19) = 4.925$, $p=0.039$] used and the technical aid usage [$F(4,76) = 72$, $p<0.001$]. A condition and device interaction was also significant [$F(4,76) = 20.8$, $p<0.001$], this occurs as all of the technical aids usages drops when using low fidelity except for the video display which increases causing the interaction seen in figure 10.7.

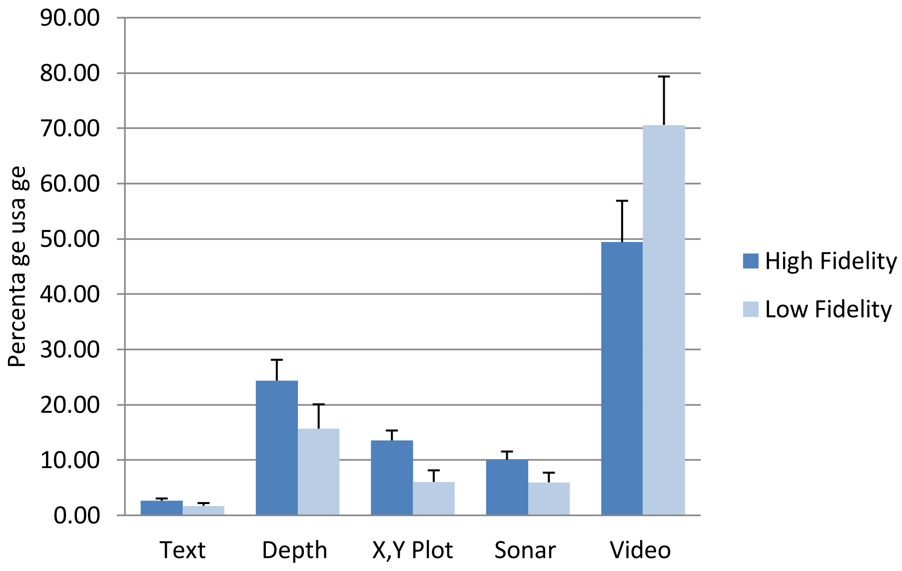


Figure 10.7: A graph showing the mean gaze percentage for both high and low fidelity conditions. Error bars indicate the standard deviation.

	Text (%)	Depth (%)	X,Y Plot (%)	Sonar (%)	Video (%)
High Fidelity	2.63	24.37	13.55	10.05	49.40
ST DEV	0.49	3.307	3.05	2.54	8.16
Low Fidelity	1.72	15.67	6.04	5.96	70.61
ST DEV	0.49	4.525	2.25	1.61	8.44
t-value	5.139	7.969	7.083	6.258	-8.06
P - value	0.055	<0.0001	<0.0001	<0.0001	<0.0001

Table 10.3: Shows the mean, standard deviation (ST DEV), t and p values for each t-test comparing percentage gaze using the high and low fidelity simulator.

Additional pre-planned paired t-tests, shown in table 10.3 clearly reflect this difference as well. All of the displays showed a significant difference ($p<0.0001$) in terms of usage except for the text display which was hardly used at all during the task for both conditions [$t(19) = 7.60$, $p = 0.055$].

This analysis was designed to evaluate one of the key questions of the research, whether simulation fidelity affects the use of technical aids. Clearly from the results a significant change is present leading to an increased usage of the radar, sonar and depth gauge.

We can also look at the eye tracking data in terms of the fixation duration rather than just the percentage duration of gaze. Figure 10.8 and table 10.4 show the average length of fixation for each display, calculated by dividing the time spent using the technical aid by the number of fixations. An ANOVA was performed on the two conditions, the results indicates that there is no significant main effect in the fidelity of simulation [F (1,19) = 0.049, p=0.872] used but a high significance for the technical aid usage [F (4,76) = 33, p<0.001]. A condition and device interaction was also significant [F (4,76) = 3.736, p=0.008], this can be investigated further by looking at the means of the technical aid fixation length.

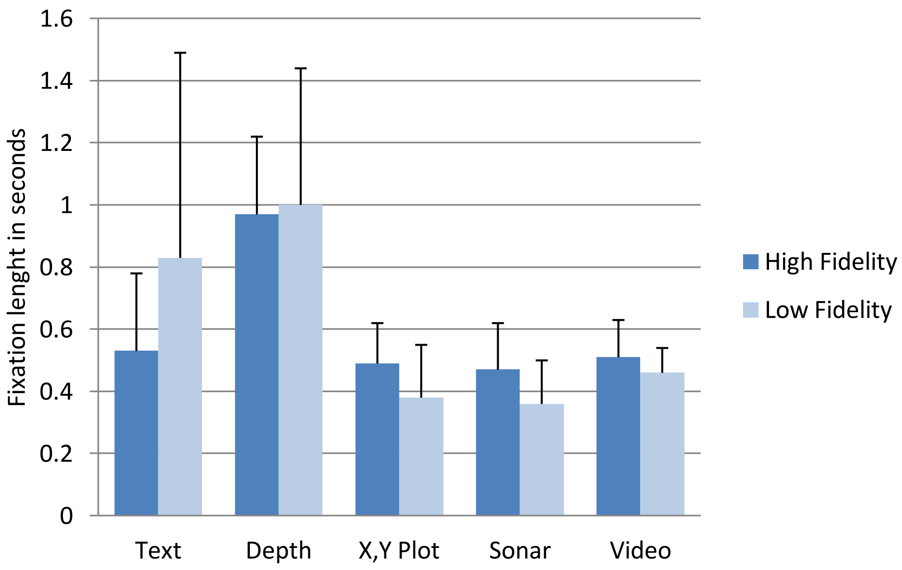


Figure 10.8: A graph showing the mean fixation length for high a low fidelity conditions. Error bars indicate the standard deviation.

	Text	Depth	X,Y Plot	Sonar	Video
High Fidelity	0.53	0.97	0.49	0.47	0.51
ST DEV	0.25	0.25	0.13	0.15	0.12
Low Fidelity	0.83	1.00	0.38	0.36	0.46
ST DEV	0.66	0.44	0.17	0.14	0.08
t-value	-2.00	-0.299	2.27	2.357	1.384
P - value	0.06	0.768	0.035	0.028	0.182

Table 10.4: Shows the mean, standard deviation (ST DEV), t and p values for each t-test comparing the fixation length for both high and low fidelity conditions.

We can see from the pre-planned t-tests in table 10.4 that the X,Y Plot and Sonar displays show a small but significant difference [$t(19) = 3.57$, $p = 0.04$], [$t(19) = 4.33$, $p = 0.03$] respectively. However, the text, depth and video usage is not significant between high and low fidelity conditions. For both conditions the average fixation duration is considerably more for the depth display (~1 second per fixation) when compared to the others. This would indicate that participants continue to hold their gaze on the depth gauge while aligning the correct depth, but the sonar and X,Y plots receive a shorter 0.5 second gaze.

This analysis was performed to assess whether the length of fixations changes dependant on the high or low fidelity condition. The results show that there is no real change in duration of the fixation between conditions. This would suggest that the way in which technical aids are used does not change but merely their amount of usage.

10.7 Eye movement patterns

As well as the overall gaze percentage we can investigate the overall movement of the eye to identify any patterns and the order in which the displays are used. Figure 10.9 shows the raw data from the eye tracker for one participant after performing the task on the low fidelity condition.

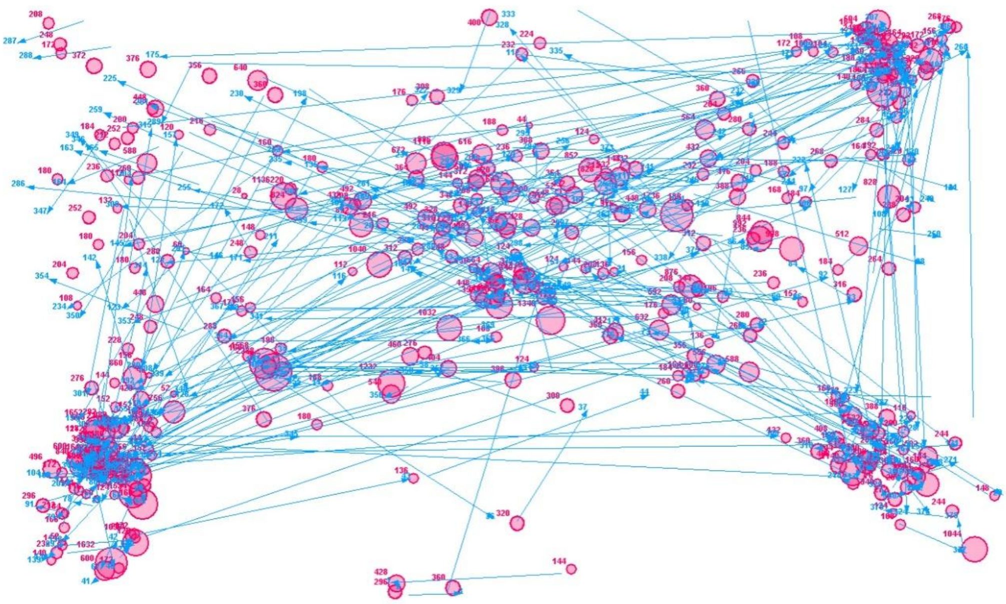


Figure 10.9: An example of the data obtained from the Eyelink II system for the high fidelity condition.

Interpreting the scan paths of the eyes by examining the saccades as a whole is a difficult task due to the sheer amount of data present. Ideally we would like to see how the scan path changed during the experiment and thereby be able to determine the use of the displays during different stages of the task. By examining the saccades from the eye link data and comparing it to the captured video of each trial, three distinct stages became apparent (figure 10.10).

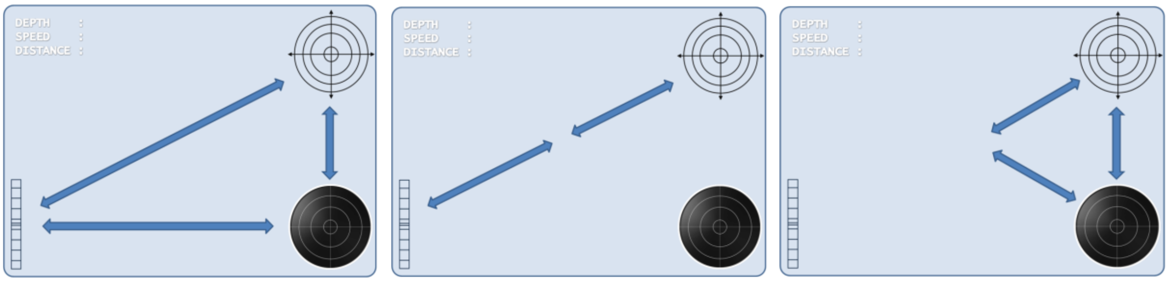


Figure 10.10: An illustration of the typical eye movements during the three observed stages: Orientation (left), Movement (middle) and Search (right).

Orientation- Within the first 10 to 20 seconds the participant quickly shifts fixation between the depth, X,Y plot and sonar. This is most likely done to orientate themselves within the virtual world and understand the target position in relation to the ROV. Examples of this behaviour for both high and low fidelity conditions can be seen in figure 10.11.

Movement- After around 20 seconds the participant has begun to pilot the ROV towards the target and fixates mostly on the depth gauge, X,Y plot and the centre of the main screen. Examples of this behaviour can be seen in figure 10.12.

Search- After around 50 seconds the distance between the ROV and target is small (less than 10 meters). At this stage the participant fixates on the main screen, X,Y plot and sonar. The correct depth is usually obtained by now and the depth gauge is no longer used. Further examples can be seen in figure 10.13.

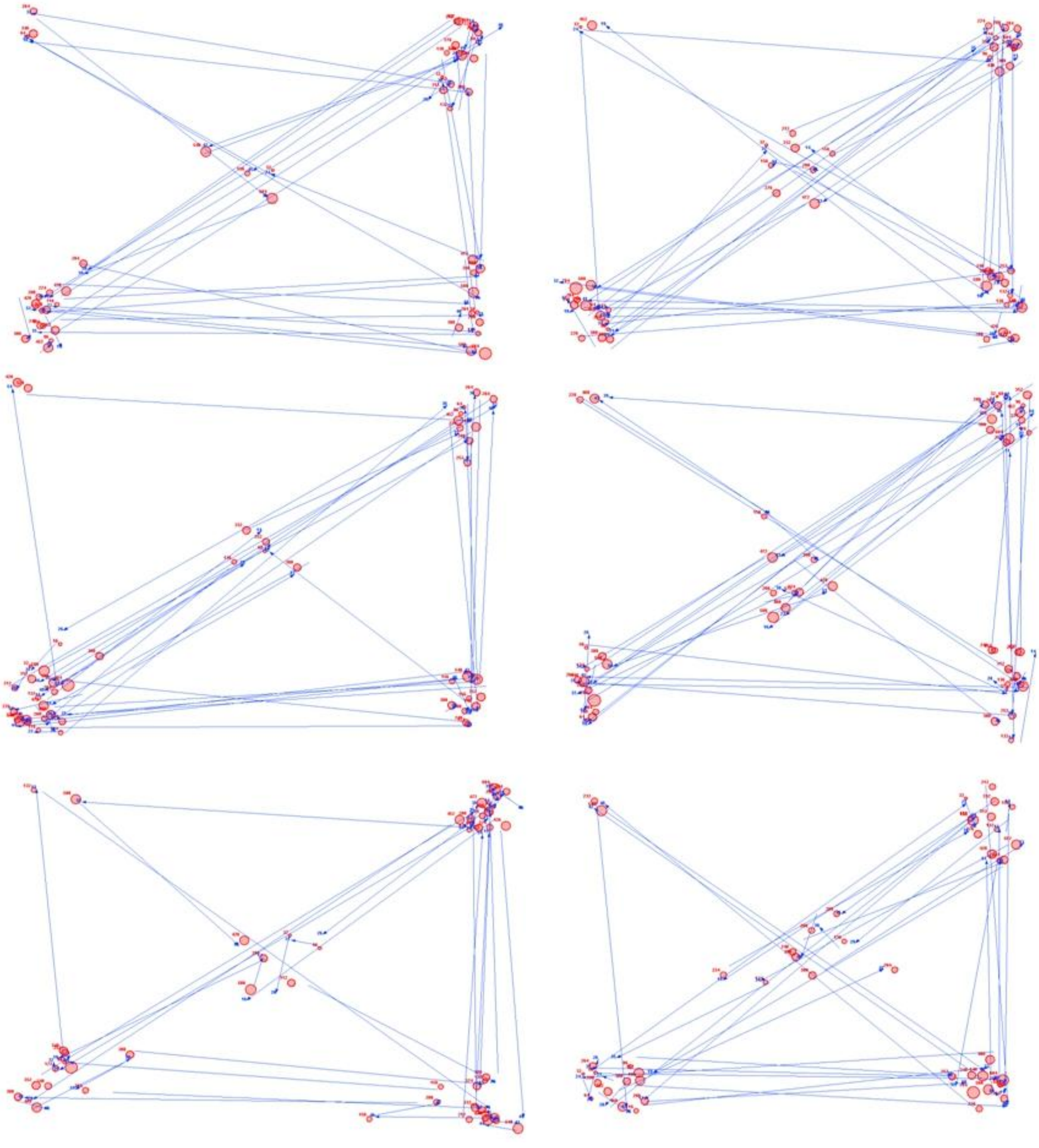


Figure 10.11: Six examples of participants' eye tracking data during the 'orientation' stage. High fidelity condition shown on the left and low *fidelity condition shown on the right.*

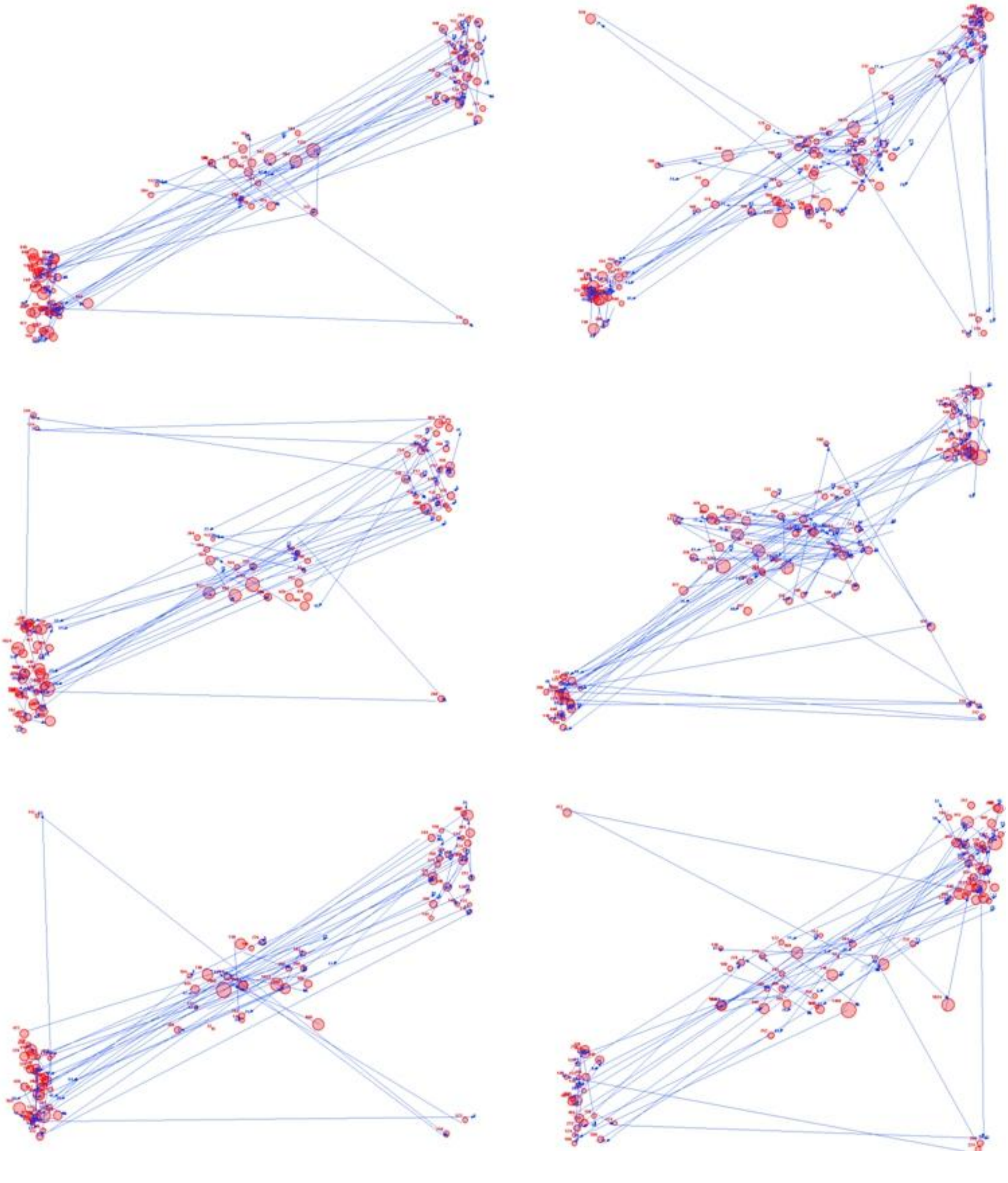


Figure 10.12: Six examples of participants' eye tracking data during the 'movement' stage. High fidelity condition shown on the left and low fidelity condition shown on the right.

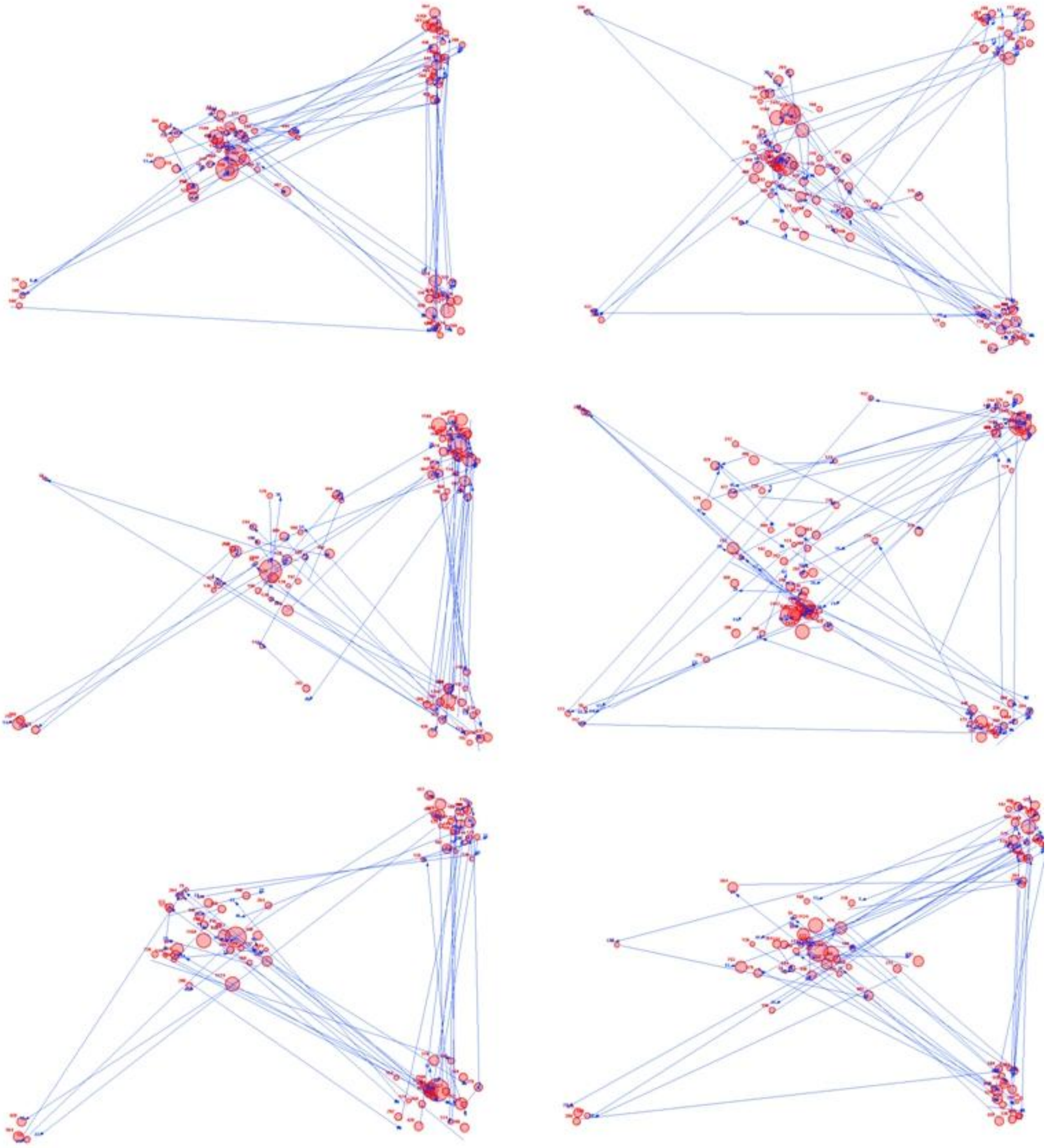


Figure 10.13: Six examples of participants' eye tracking data during the 'Search' stage. High fidelity condition shown on the left and low fidelity condition shown on the right.

10.8 Flight path

The simulation accurately logged the position of the ROV every tenth of a second allowing a flight path to be created. Figure 10.14 shows the overhead and cut away views of the Scylla showing the targets as coloured spheres (target one is blue, target two is green and target three is red).

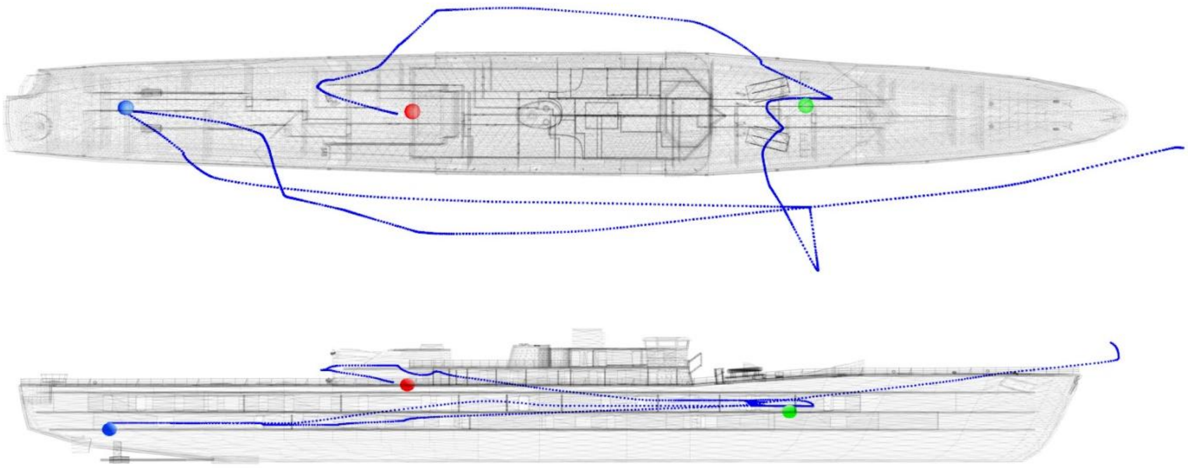


Figure 10.14: An illustration of a typical path taken by a participant during the high fidelity simulation.

During ROV Flight, participants typically had taken one of two paths. Figure 10.15 illustrates the two typical flight paths taken from an overhead perspective during the trial. The flight path towards the first target was generally the same in all cases where pilots would fly the ROV down the starboard side of the vessel and enter through one of the portals near the stern to reach the first target. After the first target had been reached the participants must now move towards the bow of the ship to reach the second target. At this point participants would either continue on the inside of the vessel down the central corridor or exit through a porthole and make their way forward on the outside of the vessel.

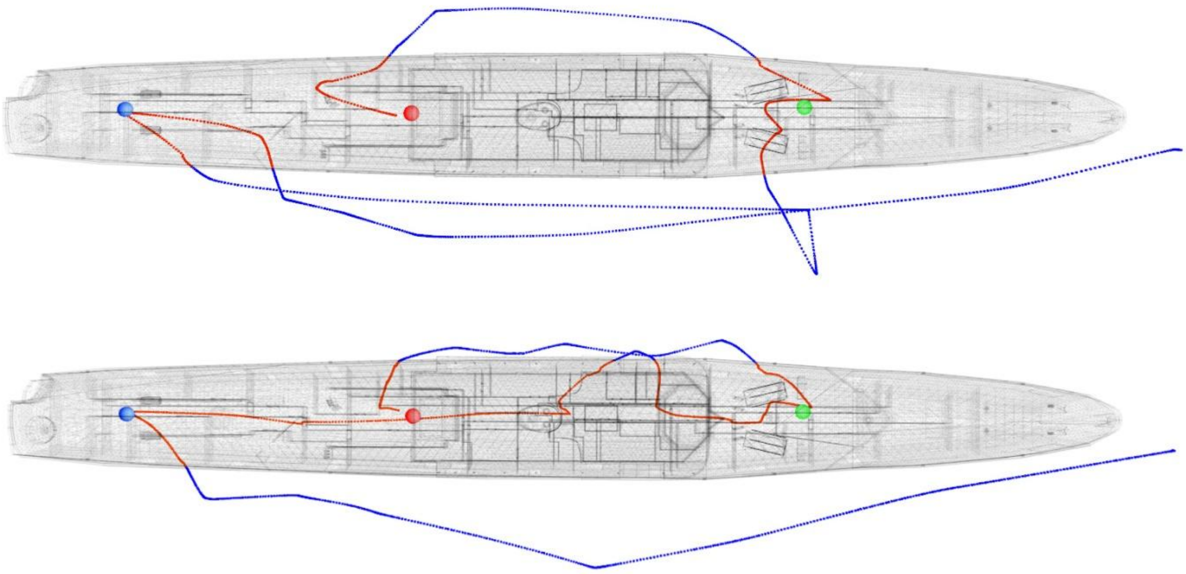


Figure 10.15: An illustration of the typical paths taken by participants during the high fidelity (top) and low fidelity (bottom) simulations. The red line indicates when the ROV was within the ship.

Participants that had travelled down the central corridor would eventually realise that they were at the incorrect depth for the next target (and on the wrong deck) and had to exit the vessel for a short period to reach the correct depth. Finally, after reaching the second target they would leave the vessel though a port hole and manoeuvre the ROV on the outside to reach the third and final target. The collected course data, when compared to the fidelity conditions, indicated that the second path type was favoured by participants using the lower fidelity condition. Further examples can be seen in figures 10.16 and 10.17.

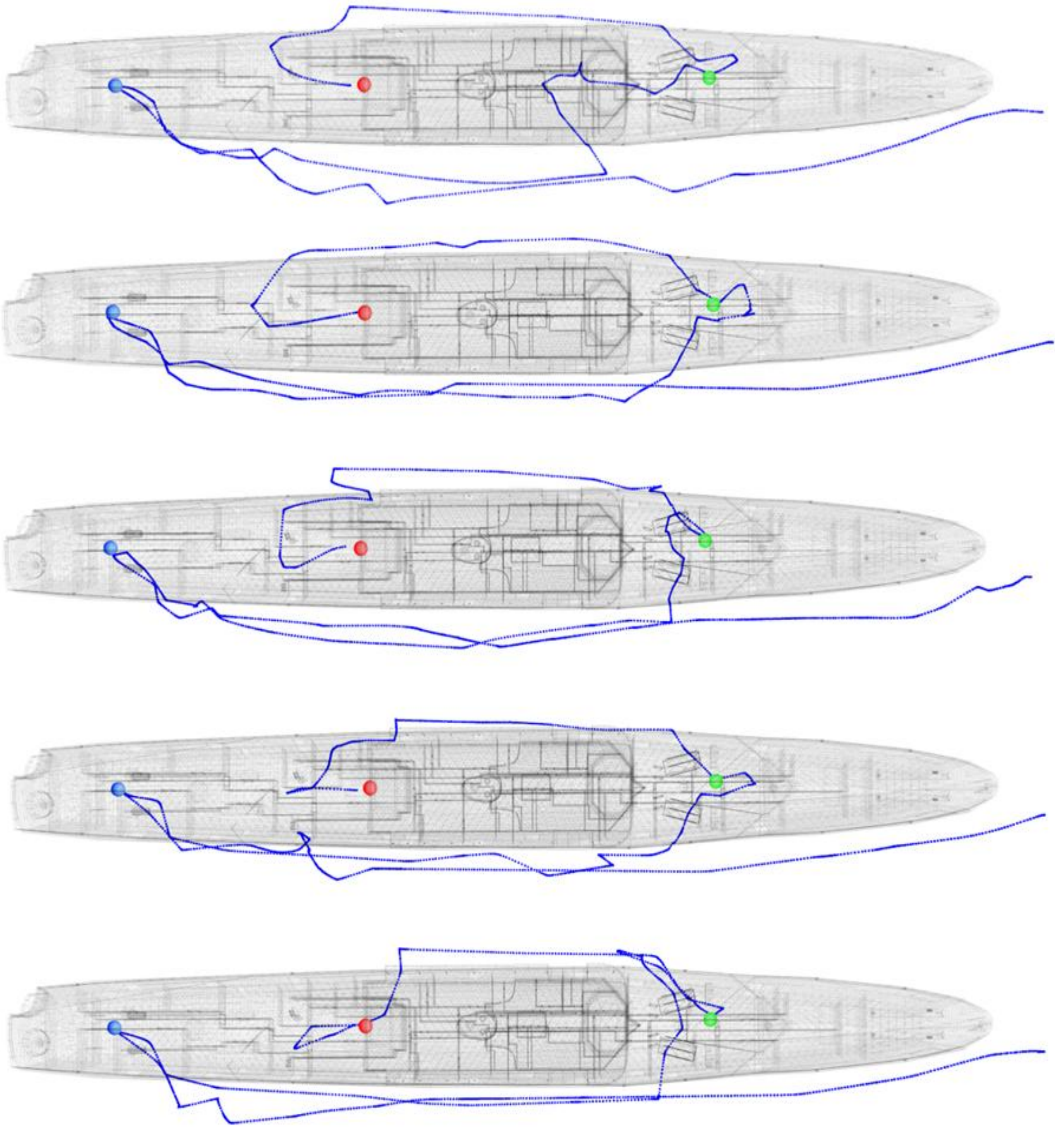


Figure 10.16: An illustration of five participants' paths taken during the high fidelity simulation.

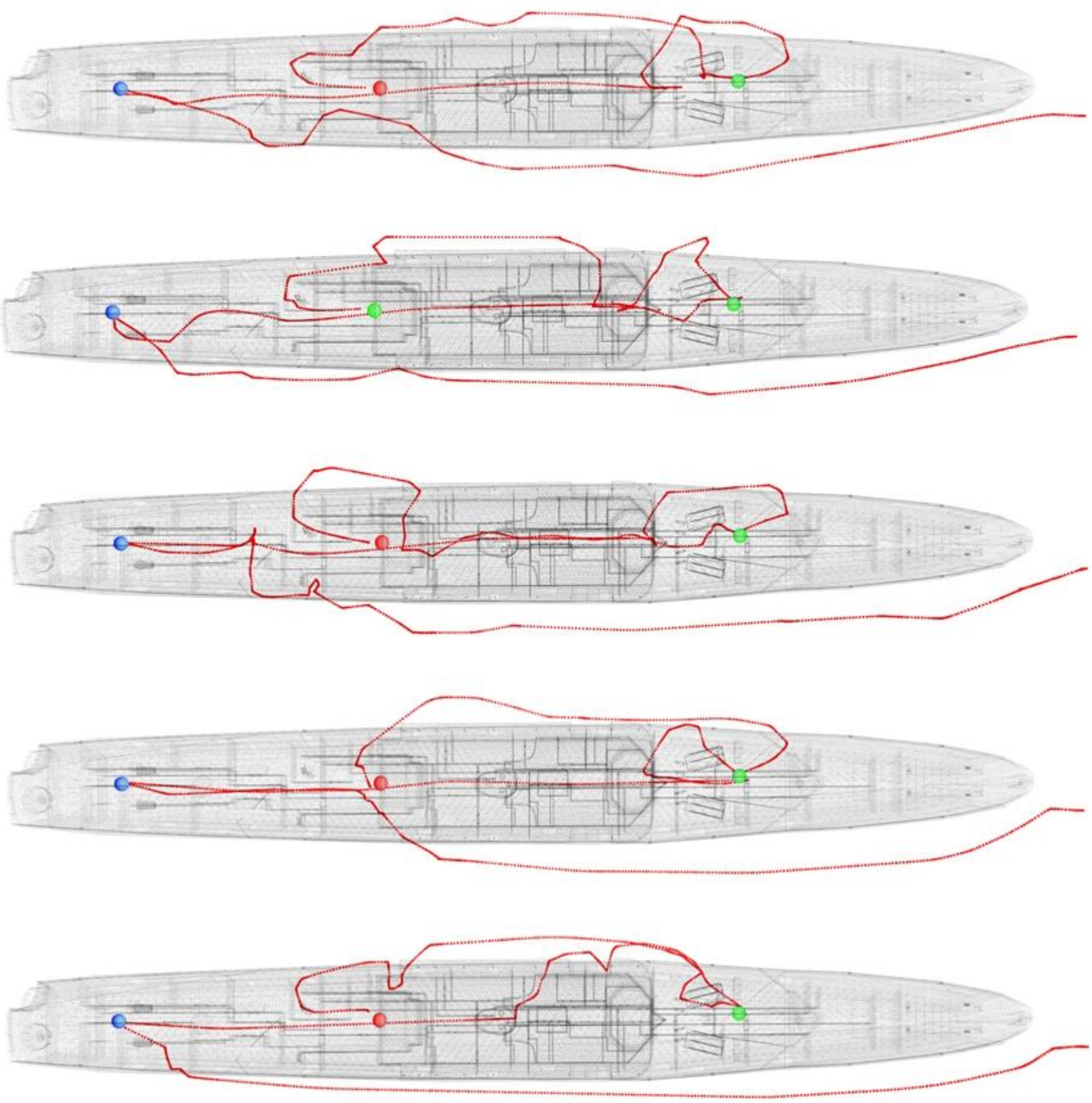


Figure 10.17: An illustration of five participants' paths taken during the low fidelity simulation.

By examining the course type taken and the condition we can see that there is a significant difference [$t(19) = -4.402$, $p<0.0001$] between the amounts of time spent within the vessel (figure 10.18). Participants using the lower fidelity condition typically spent 24.7 seconds more time within the vessel due to their preference for the second path type (table 10.5).

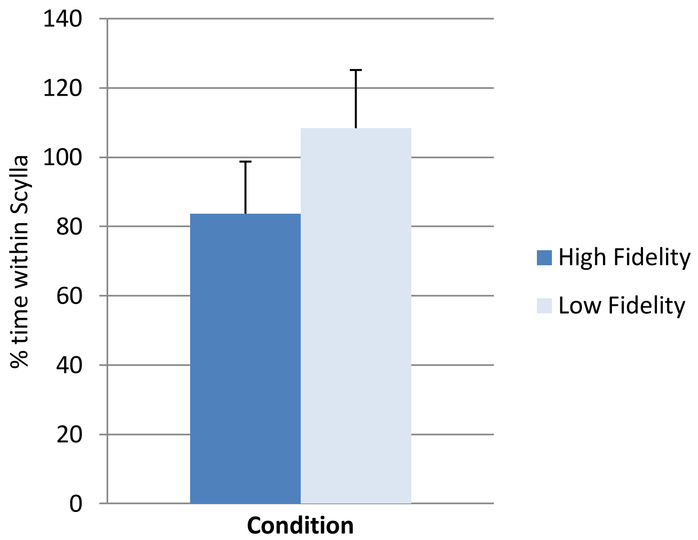


Figure 10.18: A graph of the mean percentage time spent within the Scylla for both high and low fidelity conditions. Error bars indicate the standard deviation.

	Time
High Fidelity	83.70
ST Dev	15.03
Low Fidelity	108.36
ST Dev	16.80
T-Value	-4.402
P-Value	<0.0001

Table 10.5: Shows the mean, standard deviation (ST DEV), t and p values for a t-test comparing the percentage time spent within the Scylla for both high and low fidelity conditions.

This analysis was used to show statistically the difference in course routes taken. The results show that, when participants performed the task on the low fidelity condition, they tended to manoeuvre within the ship more. However, manoeuvring internally causes a delay in reaching the second target due to the layout of the decks. This may help to explain why the low fidelity participants are not quicker in finding the second target seen in section 10.5.

10.9 Perceived Use

Finally we can examine the results from the questionnaire completed by participants about their own perceived usage of the information displays (figure 10.19). An ANOVA was again performed on the data indicating a significant main effect in condition [F (1,19) = 21.67, p<0.001] and technical aid [F (4,76) = 68.73, p<0.001]. There was also a significant intersection between condition and technical aid[F (4,76) = 11.31, p<0.001]. This can be looked at more closely by comparing the means (table 10.6).

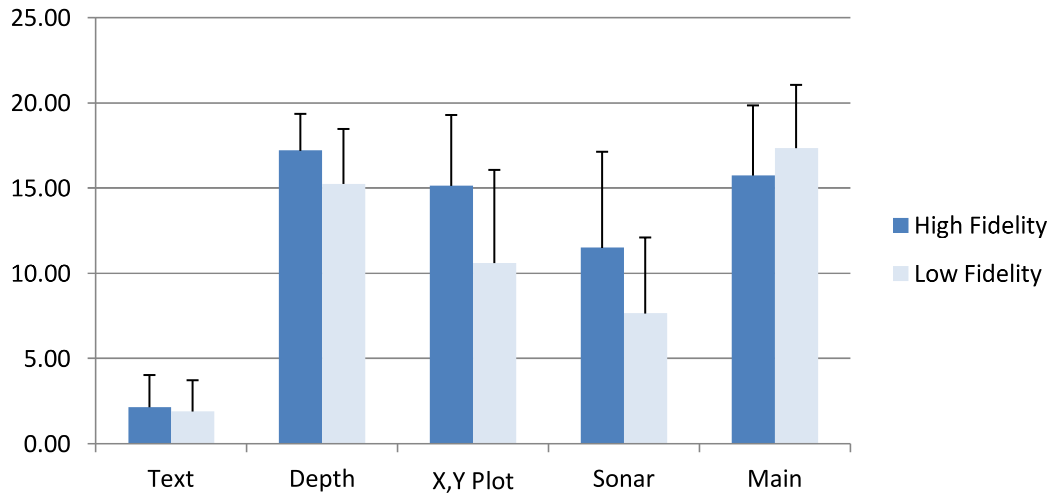


Figure 10.19: A graph of the TLX questionnaire data for both high and low fidelity conditions. Error bars indicate the standard deviation.

	Text	Depth	X,Y Plot	Sonar	Video
High Fidelity	2.15	17.20	15.15	11.50	15.75
ST Dev	1.899	2.167	4.146	5.652	4.115
Low Fidelity	1.917	15.25	10.60	7.65	17.35
ST Dev	1.832	3.228	5.481	4.464	3.717
T-Value	1.314	3.347	4.323	4.403	-1.817
P-Value	0.20441	0.003	<0.0001	<0.0001	0.085

Table 10.6: Shows the mean, standard deviation (ST DEV), t and p values for each t-test comparing the perceived usage of the technical aids.

These more subjective results correspond well with the objective results obtained with the eye tracker in terms of a significant increase in usage of the depth, X,Y plot and sonar when using the high fidelity condition [t(19) = 3.347, p = 0.003], [t(19) = 4.323, p<0.0001] [t(19) = 4.403, p<0.0001] respectively. There is no perceived difference in the main display or, the largely unused, text display.

This analysis was used to investigate whether participants could correctly perceive the usage of technical aids during the task. The results show they were able to perceive the difference on the use of each individual technical aid between high and low fidelity conditions. This result also supports the findings of the objectively captured eye tracking data that there is a significant difference in the usages of technical aids dependent on the fidelity of the simulation.

10.10 Discussion

There has been a great deal of research conducted into the use of technical aids, such as radar and their use within piloting tasks (Bell and Waag 1998), but the majority of this research has been conducted on aircraft flight systems and not ROV piloting. ROV flight is fundamentally different in control characteristics and operations are performed in a completely different environment (Barrie and Mellows 1988).

The aim of this experiment was to investigate the use of simple technical aids during a search task and establish whether the level of visual fidelity has any effect on piloting performance. In this chapter we have seen from the results that there is indeed a significant difference, not only in the end user’s performance, but in their search strategies as well. We have seen that, if the participants are exposed to a high visual fidelity simulation, then sonar, X,Y target position plots and depth gauges are used more, drawing their attention away from the main simulated remote video feed. The increased visual fidelity had the effect of diminishing and distorting the main video screen, similar to that experienced during a real world ROV flight at similar depths. This limited the amount of information experimental participants could obtain from the main video feed and encouraged them to use the other technical aids. Not only did the level of fidelity affect the use of the technical aids, it also changed the manner in which participants performed the task. During the low-fidelity condition, participants tended to stay within the Virtual Scylla wreck more. With the high fidelity condition the lack of a clear image resulted in some reluctance to travel inside. Nearly all participants preferred to manoeuvre outside of the Scylla rather than face the murky close-quarters environment of the lower decks of the simulated reef. This style of ROV flight is more indicative of that of real ROV pilots (Bell, Bayliss, and Warburton 1995). Travelling within the wreck can present many hazards such as snagged umbilical, greater chance of collision and spatial disorientation. In the low-fidelity condition the long corridors of the simulated wreck are more inviting to the participant, as they are able to view the entire length of the deck, and seem to be demonstrating a greater motivation to move internally. This tendency for participants to travel within the Scylla structure actually affected the time it took to reach the second experimental target. At first it may appear that travelling along the internal corridors would offer the quickest route to the designated target. However, eventually the participants typically reach a point when they discover that they are in fact on the wrong deck. This leads to the requirement to exit that deck and move externally to the higher deck. There is a time punishment with this move, compared to exiting the deck earlier, shortly after finding the first target. This might explain why the second target for both conditions takes the same amount of time to reach.

Finally, it should also be noted that virtual reality has often be used as a tool to develop and evaluate the use of technical aids, including head-up and head-down displays (Liu and Wen 2004; McCauley and Sharkey 1991). The results from the present experiment indicate that there is a definite need to establish the appropriate level of simulated visual fidelity, as this can drastically effect how additional information is used. Also, the perceived use of technical aids in the virtual reality test bed may not match that evident in the real world.

Generally, eye tracking data is often difficult to reliably capture, due to the nature and limitations of the equipment used, and studies have shown that in certain circumstances less than 50% of participants recorded will produce suitable results (Schnipke and Todd 2000). A larger group was sampled (30) during the eye tracking study presented within this thesis but only twenty participants maintained constant calibration throughout the experiment. While a larger sample may help to further support the conclusions presented here, the overall experimental design was developed to get the best quality data from the remaining 20 participants. Each participant performed both conditions (20 participants in both high and low fidelity rather than 10 in each) in a balanced way and within each trial there were three targets to find, rather than one allowing for the manoeuvre and search process to be repeated and observed three times per trial. In addition to the objective data obtained from the eye tracking, each participant's result from the subjective rating scale form also reflected the fact that the technical aids were used more during the higher fidelity condition. While ideally a larger sample is always better for statistical analyses, the presented results from the use of individual technical aids for each condition showed a strong significant difference ($p<0.001$). It would be reasonable to suppose that if the experiment was conducted again a similar result would be found, at least for the demographic tested here.

Chapter Eleven

This chapter summarises and concludes the work undertaken and addresses whether the aims and objectives have been met, as well as outlining possible limitations to the work and indicating areas for future investigation.

11.1 Discussion

The aim of this thesis was to develop suitable virtual environments to evaluate differences in human performance of search and navigation tasks when using various technical aids. The research was also carried out in order to investigate the effects of visual fidelity on these tasks. This was achieved through the modification of commercial off-the-shelf software packages and using innovative modelling and rendering techniques. The work presented covered two main subtasks: the recall of location and shape using hand-held tools during search procedures and the use of technical navigational aids during ROV flight. The work focused primarily on whether or not a participant's behaviour, performance or strategy differed depending on whether they were performing the task within a real-world or a simulated environment with varying levels of fidelity. The aim of the research was further split into a number of specific objectives.

The first objective was to evaluate current real-time 3D development tools to assess their suitability to support the development of training simulators for unmanned vehicle systems and technical aid-based search. Each development tool was evaluated on set criteria, such as the tool's ability to import 3D assets, ease of use, visual quality and user control.

The second objective was to examine whether results from previous real-world studies addressing cognitive search processes when using technical aids, such as metal detectors, still hold true in a virtual simulation of the task. The results of the presented research show that there was no significant difference in a person's ability to recall location and environmental features when comparing real and virtual versions of the task. In the original real-world experiment, it was found that, when participants used the metal detector it resulted in, on average, inferior performance when attempting to recall the surrounding symbols. It was concluded that this was due to a possible limitation in the attention given to the scene when using the metal detector. This limitation was further supported by the data on "average checks per tray"; it indicated a significantly lower number of checks when using the metal detector compared with the trowel. This same result was also found in the virtual equivalent, with the same significant drop in symbol recall performance when using the metal detector. One interpretation of this is that feature identification is not well supported by the metal detector condition, which might suggest a possible problem in terms of conducting a search of the environment and the location.

With the same drop in performance present in both real and virtual tasks, it could be argued that a suitable virtual training environment could be used to train out this lack of attention to detail.

The third objective was to demonstrate that a suitably accurate virtual simulation of a remotely operated vehicle could be developed using low-cost off-the-shelf development software. Its ability to simulate accurately, the visual and control characteristics of its real-world counterpart was evaluated through a detailed skills transfer study. A component objective of this was to also determine the level of the fidelity of the simulation required. Chapter six outlined the development process of an ROV simulator using off-the-shelf real-time development tools. Its physiological and physical fidelity was evaluated using a transfer training study the results of which were presented in chapter seven. The results indicated a positive training effect. There was a clear significant difference in participants who had received the virtual training before performing the task of piloting an ROV through a real world underwater obstacle. This validated the simulator as possessing the required physiological fidelity needed to develop a test-bed for further human factors experimentation in the area of search based tasks. The results also showed that, whilst the time taken to complete the obstacle course did not differ significantly between high and low fidelity virtual representations, the overall amount of collisions recorded in the low fidelity condition was significantly higher than the real world training. This could be attributable to one of two factors. Either the higher fidelity simulation provided an opportunity for participants to train with important depth cues such as shadows, or the lack of "fish eye" lens distortions evident with the low-fidelity training conditions prevented the participant from adapting to the image distortions before performing the real task.

The fourth and final objective was to develop a virtual simulation of a real world ROV search task to act as a test-bed supporting the study into the use of technical search and navigational aids. It was hypothesised that increasing the visual fidelity of the underwater simulation (i.e. effectively degrading the underwater visual representation using realistic fogging and particle rendering techniques), would increase the user's dependency on technical aids. It was also considered whether or not different search strategies were employed, depending on the technical aids used. The results showed a very significant difference, not only due to the use of the technical aids but also the search strategy taken. The clear non-degraded video view, typical of that used with many current ROV simulations, showed that participants tended to travel within the wreck. Participants using the degraded, but more realistic video feed tended to manoeuvre around the wreck from the outside (which is actually more indicative of real-world ROV search). It was also shown that the use of the simulated depth gauge tended to require constant fixed gaze while piloting the ROV to the correct depth, but the X,Y plot only required relatively short gazes when piloting the ROV to the correct X,Y position. This would indicate that participants need increased, focused, attention when performing movement in the Z (depth plane). It could be concluded from this that participants are not as used to judging location and distance in the Z plane (depth plane) as they are in the X,Y plane.

11.2 Limitations and Further Work

The first experiment investigated the mental processes of memory and recall of technical aid based real world search tasks compared to a virtual environment. The experiment performed had participants search 16 trays in fixed locations. Although this is not a realistic scenario when considering typical real world search tasks, it enabled a controlled investigation that could be replicated relatively easily within a virtual environment. A possible extension to this work would see a search task performed in a more visually complex environment where the surrounding environmental features to be recalled do not necessarily conform to any pre-set colour or shape design.

Subsequent research investigated how fidelity affects human performance when piloting a simulated ROV. Whilst great attention was paid to developing a suitably realistic ROV flight model, there were certain key omissions. Firstly, the virtual environment did not include any type of tethering cable or umbilical. The real-world facility – the National Marine Aquarium’s ExplorOcean exhibit – was specifically designed to minimise the effect of the tethering cable on the ROV (a series of cable runners are available). Nevertheless, the presence of an umbilical, even with the cable management, undoubtedly results in some handling limitations, especially given the small size of the ROVs used in the exhibit. Secondly, due to the inability to alter or make any physical changes to the ExplorOcean exhibit, it was not possible to measure any hydrodynamic forces that might be affecting the ROV’s underwater behaviour whilst participants were performing the task. This meant that timing and collision data had to be recorded by hand, which inevitably leads to a certain amount of subjectivity. As course deviation was performed using standard video techniques, the deviation could only be measured in two dimensions.

However, due to the nature of the course, essentially having all obstacles in direct alignment, this factor may have not had a great effect on the results.

In the final two experiments, only two levels of fidelity were investigated, with numerous visual omissions made to the high-fidelity simulation, such as shadows, lens distortion and particulate matter in order to create the low fidelity condition. Due to the number of visual factors omitted, it is not clear which visual effects had the most significant impact on navigational performance. In terms of collisions with objects in the simulation, the increase when participants were exposed to the low fidelity simulation was attributed to the lack of shadows (a strong monocular depth cue) or the effect of the "Fish Eye" distortion, further investigation could be conducted to investigate this.

The final experiment saw participants perform a complex search task in low and high fidelity conditions on a virtual ROV simulator. One obvious improvement would be to perform this same study in the real world in much the same way as the first and second experiments were evaluated against real world counterparts. While the created simulator was a good approximation to the real world system as evaluated in chapter six it cannot prove conclusively that the results would definitely hold true for the real world.

The use of the eye tracking system was limited to identifying the usage of the technical aids during the ROV search task. After experimentation it was clear that it could be used significantly more within the virtual world, establishing which 3D objects gain the most attention and are used as visual guides.

The simulations created during this thesis have been designed with research in mind rather than training. However, some work has been undertaken into how they can be further developed into more engaging “serious games” for educational purposes. Both the Virtual Scylla and ExplorOcean simulations were given to the National Marine Aquarium to form interactive exhibits. To increase user engagement there were several modifications made.

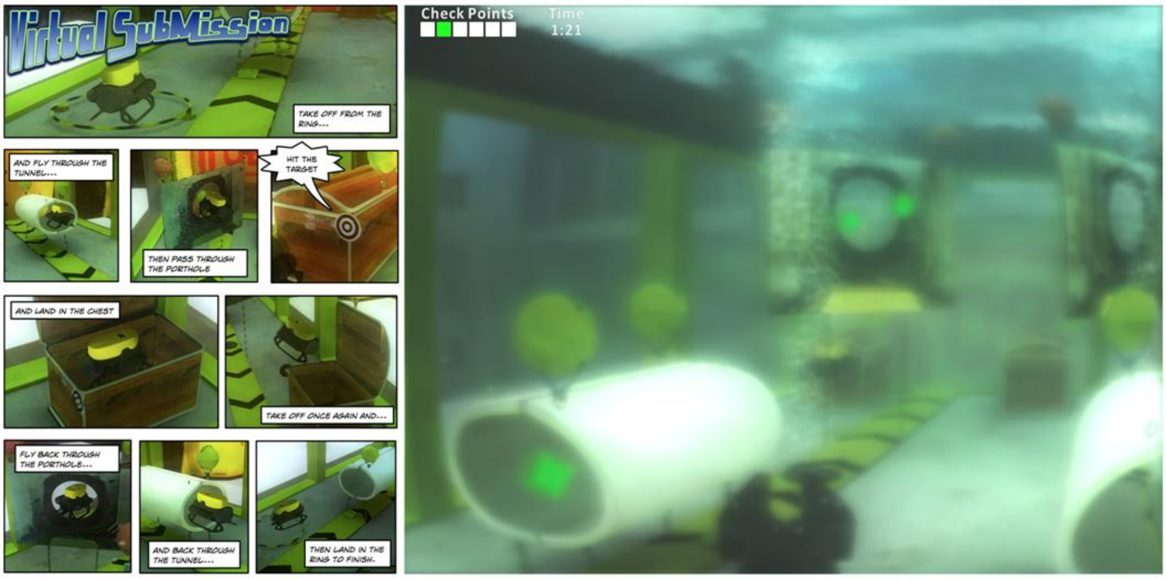


Figure 11.1: Image showing the accompanying 'comic' style instruction sheet for the Virtual Submission exhibit (left) and an in-game image of the software with additional green checkpoint crystals (right).

The ExplorOcean simulation was given an improved menu system to allow the users to quickly enter into the simulation without the assistance of aquarium staff. An additional comic book style of art-work was added as a tutorial to explain the task (figure 11.1). To begin to engage and reward the player, additional green tokens were placed along the optimal root to collect and as the player collected each one an audible tone was played. Once the course was successfully completed the player's time was displayed with an opportunity to replay the course and attempt to improve on it. The simulation allowed the player to have three attempts in total and then reset to the main menu, this would help to control how long someone could stay on the exhibition unit.

The Virtual Scylla task also had numerous modifications to make it more engaging for the user. The ROV was equipped with an underwater camera and the user was asked to take photos of certain places of the ship (the bridge, the octopus, the starfish, the treasure and the anchor).

Figure 11.2: Image showing the Virtual Scylla title screen (left) and the end of game photo score sheet (right).

The simulation could detect when a target object was in range and whether it was in view of the ROV camera, only under those conditions would the photographed image be shown in the virtual photo library (figure 11.2). Once all locations had been visited, or more than three minutes had elapsed, the game would end showing what images they had taken with the camera and their score (4/5 etc).

The metal detector experiment did not form any part of an exhibit and would take the most time to convert into a true training scenario as this was a very abstract method of searching in trays. One obvious further development would be to change the environment to a more realistic setting such as an Afghanistan street and replace the coloured marker symbols with other environmental features such as coke cans or cigarette butts. Another way in which this experiment could be expanded on is to address whether there is any significant difference in recall of environment features or target locations based on gender. The current research suggests that the current difference in recall abilities between the genders is based on our 'hunter/gather' evolutionary past (Eals and Silverman 1994). It may be interesting to see if modern technical search aids, that were unavailable to our ancestors, would affect the gender bias in recall tasks.

11.3 Conclusions

In recent years, due in part to the evolution of robust communication technologies, search and navigational tasks are increasingly being performed remotely using land, air and underwater robots. The provision of additional technical aids to assist the human operator in a variety of tasks, from piloting and navigation to the deployment of sensors and the operation of multi-function manipulators, today represents an essential development in the pursuit of effective, successful and safe remote systems performance, not to mention a critical success factor in training programs.

This thesis has shown that the execution of real-world technical aid-based search tasks could potentially be enhanced through training using virtual systems as the cognitive process of memory, recall and spatial awareness do not appear to be significantly different when directly comparing performance between a real world and simulated task.

However, it is also a finding of this thesis that the visual fidelity of the simulation has an important impact, not only on the actual exploitation of technical aids but on the manner in which humans perform remote tasks. The level of both physical and psychological fidelity must be appropriate to support the transfer of required training, as well as endeavors to ensure that the additional technical aids, such as sonar, radar position or depth, are used in the same way as their real-world equivalents. We have seen how low-cost off-the-shelf simulation development systems, or so-called "serious games", are able to provide the appropriate level of fidelity required to help achieve these aims at the fraction of the cost of bespoke simulation systems.

The use of the National Marine Aquarium's ExplorOcean facility allowed for a unique opportunity to perform a large-scale training transfer study on ROV piloting skills. Rarely is it possible to conduct such large-scale studies without large amounts of funding. It should be noted that this would have simply not been possible without a substantial amount of help and support from the Aquarium. It should also be noted that both the virtual ExplorOcean exhibit and Scylla reef simulations developed within this thesis are still in active use at the Aquarium, allowing the general public to experience ROV flight and deep-sea exploration for themselves without risk or cost implications.

References

Agba, E. I. (2002), "SeaMaster: an ROV-manipulator system simulator", *Computer Graphics and Applications, IEEE* 15 (1):24-31.

Alexander, A. L., T. Brunyé, J. Sidman, S. A. Weil, and M. A. Woburn (2005), "From gaming to training: A review of studies on fidelity, immersion, presence, and buy-in and their effects on transfer in pc-based simulations and games", Paper read at *The Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC), NTSA,* Orlando, Florida.

Anders, G. (2001), "Pilot's attention allocation during approach and landing: Eye-and headtracking research in an A330 full flight simulator". In Proceedings of the 11th International Symposium on Aviation Psychology (1):27-36.

Andre, A. D., and C. D. Wickens (1995), "When users want what's not best for them", *Ergonomics in Design: The Quarterly of Human Factors Applications* 3 (4):10-14.

Andreoli, R., R. De Chiara, U. Erra, and V. Scarano (2005), "Interactive 3d environments by using videogame engines", Proceedings of Information Visualisation Ninth International Conference IEEE, 515-520.

Auster, P. J., L. L. Stewart, and H. Sprunk (2002), "Scientific imaging: problems and solutions for ROVs". Paper read at OCEANS '88 Conference, Baltimore, MD.

Bacon, W. F., and H..E. Egeth (1991), "Local processes in preattentive feature detection", *Journal of Experimental Psychology: Human Perception and Performance* 17 (1):77-90.

Baddeley, A. (1992), "Working memory", *Science* 255 (5044):556.

Baddeley, A. (2000), "The episodic buffer: a new component of working memory?", *Trends in Cognitive Sciences* 4 (11):417-423.

Baddeley, A. D. (1997), *Human memory: Theory and practice:* Psychology Press Ltd:9-26.

Baddeley, A. D., and G. Hitch (1974), "Working memory", *Psychology of learning and motivation:*47-89.

Baldwin, R. A. (1984), "An underwater photogrammetric measurement system for structural inspection", *International Archives of Photogrammetry* 25 (A5):9-18.

Bargen, B, and P. Donnelly (1998), *Inside DirectX, M*icrosoft Programming Series:1-20.

Barrie, D. A., and S. Mellows (1988), "The future for ROVsimulators". Paper read at Advances in Underwater Technology and Offshore Engineering Conference, London.

Beck, M. R., M. S. Peterson, and M. Vomela (2006), "Memory for where, but not what, is used during visual search", *Journal of Experimental Psychology Human Perception and Performance* 32 (2):235.

Bell, C., M. Bayliss, and R. Warburton (1995), *Handbook for ROV pilot/technicians:* Oilfield Pubication Ltd.

Bell, H. H., and W. L. Waag (1998), "Evaluating the effectiveness of flight simulators for training combat skills: A review", *The International Journal of Aviation Psychology* 8 (3):223-242.

Blackman, S. (2005), "Serious games and less", A*CM Siggraph Computer Graphics* 39 (1):12-16.

Bohnsack, J. A., and D. L. Sutherland (1985), "Artificial reef research: a review with recommendations for future priorities", *Bulletin of Marine Science* 37 (1):11-39.

Brickman, B. J., L. J. Hettinger, M. M. Roe, L. Lu, D. W. Repperger, and M. W. Haas (1996), "Haptic specification of environmental events: Implications for the design of adaptive, virtual interfaces". Paper read at Virtual Reality Annual International Symposium, Santa Clara, CA.

Brooks, B. M. (1999), "The specificity of memory enhancement during interaction with a virtual environment", *Memory* 7 (1):65-78.

Brooks, B. M., F. D. Rose, J. Potter, S. Jayawardena, and A. Morling (2004), "Assessing stroke patients' prospective memory using virtual reality", *Brain Injury* 18 (4):391-401.

Buck, I. (2007), "GPU computing with NVIDIA CUDA", Paper read at SIGGRAPH '07 ACM, 6.

Buckham, B., F. R. Driscoll, and M. Nahon (2004(i)), "Development of a finite element cable model for use in low-tension dynamics simulation", *Journal of Applied Mechanics* 71:476.

Buckham, B. J., F. R. Driscoll, M. Nahon, and B. Radanovic (2004(ii)), "Torsional mechanics in dynamics simulation of low-tension marine tethers", *International Journal of Offshore and Polar Engineering* 14:218-226.

Butler, G. J., and E. Galerne (1985), "Training ROV pilots/technicians", *Sea Technology* 26 (12):30-33.

Calhoun, G. L., M. H. Draper, M. F. Abernathy, F. Delgado, M. Patzek, and Rapid Imaging Software Inc Albuquerque NM (2005), "Synthetic vision system for improving unmanned aerial vehicle operator situation awareness", Proceedings of SPIE Vol. 5802, in Enhanced and Synthetic Vision 2005.

*C*erezo, E., and F. J. Serón (2001), "Synthetic images of underwater scenes: a first approximation", Proceedings of 9th International Conference in Central Europe on Computer Graphics and Visualization WSCG'01: 395–402.

Cohen, J. D., W. M. Perlstein, T. S. Braver, L. E. Nystrom, D. C. Noll, J. Jonides, and E. E. Smith (1997), "Temporal dynamics of brain activation during a working memory task", N*ature* 386 (6625):604-608.

Conrad, R., and A. J. Hull (1964), "Information, acoustic confusion and memory span", *British Journal of Psychology* 55 (4):429-432.

Cooke, N. J., and O. Connor (2006), *Human factors of remotely operated vehicles:* Emerald Group Publishing Ltd:166-169.

Corke, P. I. (1993), "Visual control of robot manipulators a review", V*isual Serving: Real Time Control of Robot Manipulators Based on Visual Sensory Feedback:* World Scientific Publishing:1-31.

Cowan, N., L. D. Nugent, and E. M. Elliott (2000), "Memory-search and rehearsal processes and the word length effect in immediate recall: A synthesis in reply to Service", *The Quarterly Journal of Experimental Psychology Section A* 53 (3):666-670.

Crundall, D. E., G. Underwood, and P. R. Chapman (1998), "How much do novice drivers see? The effects of demand on visual search strategies in novice and experienced drivers", *Eye guidance in reading and scene perception:*395-418.

Cutmore, T. R. H., T. J. Hine, K. J. Maberly, N. M. Langford, and G. Hawgood (2000), "Cognitive and gender factors influencing navigation in a virtual environment", *International Journal of Human-Computer Studies* 53 (2):223-249.

Dinadis, N., and K. J. Vicente (1999), "Designing functional visualizations for aircraft systems status displays," *The international journal of aviation psychology* 9 (3):241-269.

Dinh, H.Q., N. Walker, C. Song, A. Kobayashi, and L. F. Hodges (1999), "Evaluating the importance of multi-sensory input on memory and the sense of presence in virtual environments", Paper read at IEEE Virtual Reality Conference 1999, Houston, Texas.

Duchowski, A. T., V. Shivashankaraiah, T. Rawls, A .K. Gramopadhye, B. J. Melloy, and B. Kanki (2000), "Binocular eye tracking in virtual reality for inspection training". Proceedings of *Symposium on Eye Tracking Research Applications*: 9 (2):89-96.

Duncan, J, and G. W. Humphreys (1989), "Visual search and stimulus similarity", *Psychological Review* 96 (3):433.

Dye, M. W. G., C. S. Green, and D. Bavelier (2009), "Increasing speed of processing with action video games", *Current Directions in Psychological Science* 18 (6):321.

Eals, M., and I. Silverman (1994), "The hunter-gatherer theory of spatial sex differences: Proximate factors mediating the female advantage in recall of object arrays", *Ethology and Sociobiology* 15 (2):95-105.

El-Nasr, M. S, and S. Yan (2006), "Visual attention in 3D video games", Proceedings of SIGCHI international conference on Advances in computer entertainment technology:22.

Ellis, S. R., B. D. Adelstein, S. Baumeler, G. J. Jense, and R. H. Jacoby (2002), "Sensor spatial distortion, visual latency, and update rate effects on 3D tracking in virtual environments" , Paper read at VRAIS 99 Conference, at Houston, Texas.

Fabekovic, Z., Z. Eskinja, and Z. Vukic (2008), "Micro ROV simulator", *ELMAR* :97-101.

Farah, M. J (1986), "The laterality of mental image generation: A test with normal subjects", *Neuropsychologia* 24 (4):541-551.

Ferwerda, J. A. (2003), "Three varieties of realism in computer graphics", *Human Vision and Electronic Imaging*: (5007):290-297.

Fisher, R. A. (1925), *Statistical methods for research workers*, Oliver & Boyd.

Fitts, P. M., R. E. Jones, and J. L. Milton (2005), "Eye movements of aircraft pilots during instrument-landing approaches", *Ergonomics: Psychological mechanisms and models in ergonomics:*56.

Fletcher, B. (2002), "ROV simulation validation and verification", Paper read at MTS/IEEE Conference OCEANS '97.

Fletcher, B., S. Harris, I. Inc, and M. A. Cataumet (1996), "Development of a virtual environment based .training system for ROV pilots", Paper read at *MTS/IEEE OCEANS '96*.

Fletcher, B., and B. Roberts (1998), "Training ROV pilots with a VE-based intelligent tutoring system", Paper read at *International/Industry Training, Simulation and Education Conference*, Orlando, FL.

Fong, T., and C. Thorpe (2001), "Vehicle teleoperation interfaces", *Autonomous robots,* 11 (1):9-18.

Garcia, J. C., M. Prats, P. J. Sanz, R. Marin, and O. Belmonte (2010), "Exploring Multimodal Interfaces For Underwater Intervention Systems", Paper read at IEEE ICRA 2010, Anchorage, AL.

Garris, R., R. Ahlers, and J. E. Driskell (2002), "Games, motivation, and learning: A research and practice model", *Simulation & Gaming* 33 (4):441.

Gilbert, R. J., and M. L. Katz (2001), "An Economist's Guide to US v. Microsoft", *Journal of Economic Perspectives* 15 (2):25-44.

Goldberg, J. H., and X. P. Kotval (1999), "Computer interface evaluation using eye movements: Methods and constructs", *International Journal of Industrial Ergonomics* 24 (6):631-645.

Goldberg, J. H., and A. M. Wichansky (2003), "Eye tracking in usability evaluation: A practitioner's guide.", in J. Hyönä, R. Radach and H. Deubel (eds.), I*n The Mind's Eye: Cognitive and applied aspects of Eye Movement Research,* Amsterdam: Elsevier, 493-516.

Granka, L. A., T. Joachims, and G. Gay (2004), "Eye-tracking analysis of user behavior in WWW search", Paper read at 27th annual international ACM SIGIR conference, Sheffield, UK.

Green, C. S., and D. Bavelier (2003), "Action video game modifies visual selective attention", N*ature* 423 (6939):534-537.

Griffiths, M. D. (2002), "The educational benefits of videogames", *Education and Health* 20 (3):47–51.

Hanna, G. B., A. B. Cresswell, and A. Cuschieri (2002), "Shadow depth cues and endoscopic task performance", *Archives of Surgery* 137 (10):1166.

Harris, L. R., M. R. Jenkin, D. Zikovitz, F. Redlick, P. Jaekl, U. T. Jasiobedzka, H. L. Jenkin, and R. S. Allison (2002), "Simulating self-motion I: Cues for the perception of motion", V*irtual Reality* 6 (2):75-85.

Harris, S. E., and B. E. Fletcher (1997), "Year one report: Training for remote sensing and manipulation", Paper read at MTS/IEEE Conference Proceedings Oceans '97 at Halifax, Nova Scotia, Canada.

Hart, S. G. (2006), "NASA-task load index (NASA-TLX); 20 years later", Paper read at Human Factors and Ergonomics Society Annual Meeting Proceedings at San Francisco, California, USA.

Hart, S. G, and L. E. Staveland (1988), "Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research", *Human mental workload* 1:139–183.

Hays, R. T., J. W. Jacobs, C. Prince, and E. Salas (1992), "Flight simulator training effectiveness: A meta-analysis", *Military Psychology* 4 (2):63-74.

Hendrix, C., and W. Barfield (1995), "Presence in virtual environments as a function of visual and auditory cues", Paper read at Virtual Reality Annual International Symposium, Research Triangle Park, NC.

Hendrix, C. (1995), "Relationship between monocular and binocular depth cues for judgements of spatial information and spatial instrument design", D*isplays* 16 (3):103-113.

Herlitz, A., L .G. Nilsson, and L. Bäckman (1997), "Gender differences in episodic memory", *Memory & Cognition* 25 (6):801-811.

Herman, H., J. D. McMahill, and G. Kantor (2001), "Enhanced operator interface for hand-held landmine detector", Paper read at *SPIE 4394.*

Herman, P. (2009), "Transformed equations of motion for underwater vehicles", *Ocean Engineering, Elsevier Ltd* 36 (3-4):306-312.

Houghton, R. J., C. Baber, and J. F. Knight (2009), "Reporting Location and Environmental Features when Using Metal Detector or Probe in a Simplified Mine Detection Task", Paper read at International Conference on Contemporary Ergonomics.

Howard, I. P. (1989), "Spatial Vision within Egocentric and Exocentric Frames of Reference", Paper read at the NASA Conference on Spatial Displays and Spatial Instruments, Asilomar, CA.

Hu, H. H., A. A. Gooch, S. H. Creem-Regehr, and W. B. Thompson (2002), "Visual cues for perceiving distances from objects to surfaces", *Presence: Teleoperators & Virtual Environments* 11 (6):652-664.

Hu, H. H., A. A. Gooch, W. B. Thompson, B. E. Smits, J. J. Rieser, and P. Shirley (2000), "Visual cues for imminent object contact in realistic virtual environment", Paper read at 11th IEEE Visualization at Salt Lake City, UT.

Hubona, G. S., P. N. Wheeler, G. W. Shirah, and M. Brandt (1999), "The role of object shadows in promoting 3D visualization", *ACM Transactions on Computer-Human Interaction* 6 (3):214-242.

Iwasaki, K., Y. Dobashi, and T. Nishita (2002), "An efficient method for rendering underwater optical effects using graphics hardware", in, *Computer Graphics Forum*. Wiley Online Library, 701-711.

Jacob, R. J. K, and K. S. Karn (2003), "Eye tracking in human-computer interaction and usability research: Ready to deliver the promises", M*ind* 2 (3):4.

James, Thomas W., and Doreen Kimura (1997), "Sex differences in remembering the locations of objects in an array: Location-shifts versus location-exchanges", *Evolution and Human Behavior* 18 (3):155-163.

Jones, D. G., and M. R. Endsley (1996), "Sources of situation awareness errors in aviation", *Aviation, Space, and Environmental Medicine*:67(6):507-512.

Jordán, M. A , J. L. Bustamante, and J. M. P. Cortiñas (2005), "Design of Adaptive Control Systems for ROVs Using Inverse Dynamics and State/Disturbance Observation", Paper read at 6th Argentine Symposium on Computing Technology, at Rosario, Set.

Just, MA, and PA Carpenter (1976), "The role of eye-fixation research in cognitive psychology", *Behavior Research Methods & Instrumentation* 8 (2):139-143.

Karn, K. S., S. Ellis, and C. Juliano (1999), "The hunt for usability: tracking eye movements", Paper read at CHI EA '99, Pittsburgh, Pennsylvania.

Kim, H., J. K. Lee, J. H. Park, B. J. Park, and D. S. Jang (2002), "Applying digital manufacturing technology to ship production and the maritime environment", *Integrated manufacturing systems* 13 (5):295-305.

Klauer, K C., and Z. Zhao (2004), "Double Dissociations in Visual and Spatial Short-Term Memory", *Journal of Experimental Psychology: General* 133 (3):355-381.

Kontkanen, J., and S. Laine (2005), "Ambient occlusion fields". Proceeding of the Symposium on Interactive 3D Graphics and Games, New York, New York.

Kozlowski, S. W. J, R. P. DeShon, S. G. Schifflett, L. R. Elliott, E. Salas, and M. D. Coovert (2004), "A psychological fidelity approach to simulation-based training: Theory, research, and principles", *Scaled worlds: Development, validation, and applications:*75–99.

Lanier, J. (1992), "Virtual Reality: The Promise of the Future", *Interactive Learning International* 8 (4):275-279.

Lanier, J., and A. Heilbrun (1988), "A Vintage Virtual Reality Interview", *Whole Earth Review magazine*. http://www.jaronlanier.com/vrint.html (accessed April 2007).

Law, B., M. S. Atkins, A. E. Kirkpatrick, and A .J. Lomax (2004), "Eye gaze patterns differentiate novice and experts in a virtual laparoscopic surgery training environment", Proceedings of the Symposium on Eye Tracking Research & Applications, New York, NY.

Li, R., C. Tao, T. A. Curran, and R. G. Smith (1997), "Digital underwater photogrammetric system for large scale underwater spatial information acquisition", *Marine Geodesy* 20 (2):163-173.

Lintern, G., S. N. Roscoe, and J. E. Sivier (1990), "Display principles, control dynamics, and environmental factors in pilot training and transfer", *Human Factors: The Journal of the Human Factors and Ergonomics Society* 32 (3):299-317.

Liu, Y. C., and M. H. Wen (2004), "Comparison of head-up display (HUD) vs. head-down display (HDD): driving performance of commercial vehicle operators in Taiwan", *International Journal of Human-Computer Studies* 61 (5):679-697.

Logie, R. H., and C. Marchetti (1991), "Visuo-spatial working memory: Visual, spatial or central executive", *Mental images in human cognition* 80.

Lok, B., S. Naik, M. Whitton, and F. P. Brooks (2003), "Effects of handling real objects and self-avatar fidelity on cognitive task performance and sense of presence in virtual environments", *Presence: Teleoperators & Virtual Environments* 12 (6):615-628.

Mania, K., T. Troscianko, R. Hawkes, and A. Chalmers (2003), "Fidelity metrics for virtual environment simulations based on spatial memory awareness states", *Presence: Teleoperators & Virtual Environments* 12 (3):296-310.

Matheis, R. J., M. T. Schultheis, L. A. Tiersky, J. DeLuca, S. R. Millis, and A. Rizzo (2007), "Is learning and memory different in a virtual environment?", *The Clinical Neuropsychologist* 21 (1):146-161.

McCauley, M. E., and T. J. Sharkey (1991), "Spatial orientation and dynamics in virtual reality systems: lessons from flight simulation", Paper read at Human Factors Society 35th Annual Meeting at San Francisco, CA.

Michalski, R. S., J. G. Carbonell, and T. M. Mitchell (1986), *Machine learning: An artificial intelligence approach (Vol. 1).* Los Altos, California,Morgan Kaufmann Publishing.

Miller, R., M. Hobday, T. Leroux-Demers, and X. Olleros (1995), "Innovation in complex systems industries: the case of flight simulation", *Industrial and Corporate Change* 4 (2):363.

Miller, R. B., and American Institute for Research (1954), "Psychological considerations in the design of training equipment", in, Pittsburgh, PA: Defense Technical Information Center.

Mittring, M. (2007), "Finding next gen: Cryengine 2", Paper read at SIGGRAPH '07 at San Diego, CA.

Moray, N., M. Fitter, D. Ostry, D. Favreau, and V. Nagy (1976), "Attention to pure tones", *The Quarterly Journal of Experimental Psychology* 28 (2):271-283.

Neider, J., T. Davis, and M. Woo (1997), O*penGL. Programming guide:* Addison-Wesley Reading, MA.

Ormiston, D. (2009), "In at the deep end", *International Water Power and Dam Construction* 61 (4):35-36.

Page, M. P.A., and D. Norris (1998), "The primacy model: A new model of immediate serial recall", *Psychological Review-New York*, 105:761-781.

Parsons, T. D., and A. A. Rizzo (2008), "Initial validation of a virtual environment for assessment of memory functioning: virtual reality cognitive performance assessment test", *CyberPsychology & Behavior* 11 (1):17-25.

Pausch, R. D. Proffitt, and G. Williams (1997), "Quantifying immersion in virtual reality", Paper read at SIGGRAPH '97 24th annual conference on Computer graphics and interactive techniques, Los Angeles, CA.

Piaget, J. (1973), "*To understand is to invent: The future of education",* New York: Grossman Publishers:99-101.

Pioch, N. J., B. Roberts, and D. Zeltzer (1997), "A virtual environment for learning to pilot remotely operated vehicles", Paper read at International Conference on Virtual Systems and MultiMedia (VSMM '97) at Geneva , Switzerland.

Raine, G. A., and M. C. Lugg (1996), "ROV inspection of welds: a reality", I*nsight* 38 (5):346-350.

Ramloll, R., and D. Mowat (2002), "Wayfinding in virtual environments using an interactive spatial cognitive map", Paper read at Fifth International Conference on Information Visualisation at London, UK.

Redline, C. D., and C. P. Lankford (2001), "Eye-movement analysis: a new tool for evaluating the design of visually administered instruments (paper and web)", Paper read at American Association of Public Opinion Research, at Montreal, Canada.

Reid, L., and M. Nahon (1988), "Response of airline pilots to variations in flight simulator motion algorithms", *Journal of Aircraft* 25 (7):639-646.

Reinhard, E. (2006), *High dynamic range imaging: acquisition, display, and image-based lighting:* Morgan Kaufmann.

Ring, H. (1998), "Is neurological rehabilitation ready for immersion in the world of virtual reality?", *Disability & Rehabilitation* 20 (3):98-101.

Riordan, J., E. Omerdic, and D. Toal (2005), "Implementation and application of a real-time sidescan sonar simulator", Paper read at Oceans '05, Europe.

Ritschel, T., T. Grosch, and H. P. Seidel (2009), "Approximating dynamic global illumination in image space", Paper read at 2009 symposium on Interactive 3D graphics and games, Boston, MA.

Rizzo, A. A., and J. G. Buckwalter (1997), "Virtual reality and cognitive assessment and rehabilitation: the state of the art", *Virtual reality in neuro-psycho-physiology: Cognitive, clinical, and methodological issues in assessment and rehabilitation:*123-146.

Robertson, G., M. Czerwinski, and M. Van Dantzich (1997), "Immersion in desktop virtual reality". Paper read at 10th annual ACM symposium on User interface software and technology, Santa Barbara, CA.

Rose, F. D., E. A. Attree, B. M. Brooks, D. M. Parslow, and P. R. Penn (2000), "Training in virtual environments: transfer to real world tasks and equivalence to real task training", *Ergonomics* 43 (4):494-511.

Rose, F. D., B. M. Brooks, and A. A. Rizzo (2005), "Virtual reality in brain damage rehabilitation: review", *CyberPsychology & Behavior* 8 (3):241-262.

Ruddle, R. A., S. J. Payne, and D. M. Jones (1997), "Navigating buildings in" desk-top" virtual environments: Experimental investigations using extended navigational experience", *Journal of Experimental Psychology Applied* 3:143-159.

Ruddle, R. A., S. J. Payne, and D. M. Jones (1999), "The effects of maps on navigation and search strategies in very-large-scale virtual environments", *Journal of Experimental Psychology Applied* 5:54-75.

Ryan, M. L. (1999), "Immersion vs. interactivity: Virtual reality and literary theory", *SubStance* 28 (2):110-137.

Sarletu, M., and G. Klein (2004), "Hardware-accelerated ambient occlusion computation", In proceedings of Vision, Modeling, and Visualization: 331-338.

Sawyer, B., and P. Smith (2008), "Serious games taxonomy", Paper read at The Serious Games Summit at San Francisco, CA.

Scerbo, M. W., and S. Dawson (2007), "High Fidelity, High Performance?", S*imulation in Healthcare* 2 (4):224.

Schechner, Y. Y., and N. Karpel (2004), "Clear underwater vision", Paper read at CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition at Washington, DC.

Schmidt, J., C. Wöhler, L. Krüger, T. Gövert, and C. Hermes (2007), "3D scene segmentation and object tracking in multiocular image sequences", Paper read at 5th International Conference on Computer Vision Systems at Bielefeld, Germany.

Schnipke, S. K., and M .W. Todd (2000), "Trials and tribulations of using an eye-tracking system", Proceedings of CHI EA '00: 273-274.

Schrater, P. R., and D. Kersten (2000), "How optimal depth cue integration depends on the task", *International Journal of Computer Vision* 40 (1):71-89.

Schultheis, M. T., J. Himelstein, and A. A. Rizzo (2002), "Virtual reality and neuropsychology: upgrading the current tools", *The Journal of head trauma rehabilitation* 17 (5):378.

Selcon, S. J., R. M. Taylor, and E. Koritsas (1991), "Workload or Situational awareness: TLX vs. SART for Aerospace Systems Design Evaluation". Proceedings of Human Factors and Ergonomics Society Annual Meeting:62-66.

Sennersten, C. (2008), "Gameplay (3D Game Engine + Ray Tracing = Visual Attention through Eye Tracking)". Blekinge Institute of Technology Licentiate Dissertion Series 10:201.

Shaffer, D. W., K. R. Squire, R. Halverson, and J. P. Gee (2005), "Video games and the future of learning", *Phi delta kappan* 87 (2):104-111.

Staszewski, J. (2006), "Spatial thinking and the design of landmine detection training", *Applied spatial cognition: From research to cognitive technology:*231–265.

Staszewski, J., and A. Davison (2000), "Mine detection training based on expert skill", *Detection and Remediation Technologies for Mines and Minelike Targets V:*90-101.

Stefanidis, D., J. R. Korndorffer Jr, S. Markley, R. Sierra, B. T. Heniford, and D. J. Scott (2007), "Closing the gap in operative performance between novices and experts: Does harder mean better for laparoscopic simulator training?", *Journal of the American College of Surgeons* 205 (2):307-313.

Stephens, R. S. (1990), "Real-time 3D object tracking", *Image and Vision Computing* 8 (1):91-96.

Stone, R., R. Guest, E. Ch'ng, C. McCririe, C. Collis, R. Mannur, I. Rehmi, and Birmingham University Department of Electronic & Electrical Engineering (2006), "Serious Gaming Technologies Support Human Factors Investigations of Advanced Interfaces for Semi-Autonomous Vehicles", Paper read at Virtual Media for Military Applications; NATO RTA HFM-136 Workshop at US Military Academy; West Point, NY.

Stone, R., D. White, R. Guest, and B. Francis (2009), "The Virtual Scylla: an exploration of “serious games”, artificial life and simulation complexity", *Virtual Reality* 13 (1):13-25.

Stone, R. S. (2003), "Whatever happened to virtual reality", *Equipment, Training and Support News (Special ITEC 2003 Edition). April:*43-47.

Sugano, N., H. Kato, and K. Tachibana (2003), "The effects of shadow representation of virtual objects in augmented reality", Paper read at The Second IEEE and ACM International Symposium on Mixed and Augmented Reality at Madison, Wisconsin.

Sutherland, I. E. (1965), "The ultimate display", Paper read at International Federation for Information Processing.

Teichner, W. H, and M. J. Krebs (1972), "Laws of the simple visual reaction time", *Psychological Review* 79 (4):344-358.

Tendick, F, S Bhoyrul, and LW Way (1997), "Comparison of laparoscopic imaging systems and conditions using a knot-tying task", *Computer Aided Surgery* 2 (1):24-33.

Thomas, D. A. H. (1973), "Multiple comparisons among means-a review", *The Statistician:*16-42.

Tsang, P. S, and M. A. Vidulich (2002), *Principles and practice of aviation psychology:* Psychology Press:357-397.

Ulinski, A., C. Zanbaka, Z. Wartell, P. Goolkasian, and L. F. Hodges (2007), "Two handed selection techniques for volumetric data", Paper read at 2007 IEEE Symposium on 3D User Interfaces at Charlotte, NC.

Valverde, H. H. (1973), "A review of flight simulator transfer of training studies", *Human Factors: The Journal of the Human Factors and Ergonomics Society* 15 (6):510-522.

Van Eck, R. (2006), "Digital game-based learning: It's not just the digital natives who are restless", *Educause Review* 41 (2):16.

Vinson, N. G. (1999), "Design guidelines for landmarks to support navigation in virtual environments". Paper read at SIGCHI conference on Human factors in computing systems at Pittsburgh, Pennsylvania.

Volino, P., and N. Magnenat-Thalmann (2001), "Comparing efficiency of integration methods for cloth simulation", Paper read at Computer Graphics International, at Hong Kong.

Wagner, C. R., N. Stylopoulos, and R. D. Howe (2002), "The role of force feedback in surgery: Analysis of blunt dissection", Paper read at Haptics Symposium at Washington, DC.

Waller, D., E. Hunt, and D. Knapp (1998), "The transfer of spatial knowledge in virtual environment training", P*resence* 7 (2):129-143.

Walton, J, M. Cooke, R. Uhrich, Naval Command Control, Ocean Surveillance Centre RDT, and E DIV San Diego CA (1993), "Advanced unmanned search system", Naval Command Control and Ocean Surveillance Center. *http://handle.dtic.mil/100.2/ADA279598* (accessed June 2009).

Wang, W., and C. M. Clark (2006), "Modeling and simulation of the VideoRay Pro III underwater vehicle", Proceedings of IEEE Oceans 06, Asia Pacific.

Wang, Z., and A. C. Bovik (2002), "A universal image quality index", *Signal Processing Letters, IEEE* 9 (3):81-84.

Wanger, L. R., J. A. Ferwerda, and D. P. Greenberg (2002), "Perceiving spatial relationships in computer-generated images", *Computer Graphics and Applications, IEEE* 12 (3):44-58.

Wenger, E. (2004), "Artificial intelligence and tutoring systems", I*nternational Journal of Artificial Intelligence in Education* 14:39-65.

Wilson, J. R. (2002), "UAVs and the human factor", *Aerospace America* 40 (7):53-57.

Wilson, P. N. (1997), "Use of virtual reality computing in spatial learning research", *Handbook of Spatial Research Paradigms and Methodologies: Spatial cognition in the child and adult:*181.

Wilson, P. N., N. Foreman, and M. Tlauka (1997), "Transfer of Spatial Information from a Virtual to Real Environment", *Human Factors* 39 (4).

Winkler, S. (2001), "Visual fidelity and perceived quality: Towards comprehensive metrics", Paper read at SPIE Human Vision and Electronic Imaging Conference.

Yilmaz, A., O. Javed, and M. Shah (2006), "Object tracking: A survey", *Acm Computing Surveys (CSUR)* 38 (4):13.

Zalesky, J. L. (1998), “A part-task trainer for underwater tether awareness”. Dept. of Electrical Engineering and Computer Science Massachusetts Institute of Technology. *http://hdl.handle.net/1721.1/50437* (accessed May, 2009).

Zhang, J., K. A. Johnson, J. T. Malin, and J. W. Smith (2002), "Human-centered information visualization", Paper read at International Workshop on dynamic Visualizations and Learning, Digital Enterprise Research Institute, University of Innsbruck.

Appendix A

NASA TLX Load Index Form and Rating Scale Definitions

NASA Task Load Index

Hart and Staveland's NASA Task Load Index (TLX) method assesses work load on five 7-point scales. Increments of high, medium and low estimates for each point result in 21 gradations on the scales.

Name	Task	Date

Mental Demand — How mentally demanding was the task?

Very Low — Very High

Physical Demand — How physically demanding was the task?

Very Low — Very High

Temporal Demand — How hurried or rushed was the pace of the task?

Very Low — Very High

Performance — How successful were you in accomplishing what you were asked to do?

Perfect — Failure

Effort — How hard did you have to work to accomplish your level of performance?

Very Low — Very High

Frustration — How insecure, discouraged, irritated, stressed, and annoyed wereyou?

Very Low — Very High

RATING SCALE DEFINITIONS

Title	Endpoints	Descriptions
MENTAL DEMAND	*Low/High*	How much mental and perceptual activity was required (e.g., thinking, deciding, calculating, remembering, looking, searching, etc.)? Was the task easy or demanding, simple or complex, exacting or forgiving?
PHYSICAL DEMAND	*Low/High*	How much physical activity was required (e.g., pushing, pulling, turning, controlling, activating, etc.)? Was the task easy or demanding, slow or brisk, slack or strenuous, restful or laborious?
TEMPORAL DEMAND	*Low/High*	How much time pressure did you feel due to the rate or pace at which the tasks or task elements occurred? Was the pace slow and leisurely or rapid and frantic?
EFFORT	*Low/High*	How hard did you have to work (mentally and physically) to accomplish your level of performance?
PERFORMANCE	*Good/Poor*	How successful do you think you were in accomplishing the goals of the task set by the experimenter (or yourself)? How satisfied were you with your performance in accomplishing these goals?
FRUSTRATION LEVEL	*Low/High*	How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?

Appendix B

User Consent Forms for Experimental Studies

University of Birmingham

Phd Research Trial

Virtual Reality Experiment to determine the effect of recall of location and environmental features using technical aids.

Consent Form

	Please tick to confirm
I have read and understood the instruction sheet.	
I have been given the opportunity to ask questions about the study.	
I am satisfied with the answers given to my queries for this study.	
I understand that I have the right to withdraw from the experiment at any time without explanation.	
I agree to report any discomfort that might result from using the screen.	
I agree to the data being collected from me to be used for research purposes.	
I agree to take part in the aforementioned research study.	

Researcher Name ………………………………………………… Date ……………..

Participant Name………………………………………………… Date……………...

University of Birmingham

Phd Research Trial

Virtual Reality Experiment of Visual Fidelity on piloting and navigation search tasks

Consent Form

	Please tick to confirm
I have read and understood the instruction sheet.	
I have been given the opportunity to ask questions about the study.	
I am satisfied with the answers given to my queries for this study.	
I understand that I have the right to withdraw from the experiment at any time without explanation.	
I agree to report any discomfort that might result from using the eye tracking.	
I agree to the data being collected from me to be used for research purposes.	
I agree to take part in the aforementioned research study.	

Researcher Name ………………………………………………… Date ……………..

Participant Name………………………………………………… Date……………...

Appendix C

ROV ExplorOcean Questionnaire

Questionnaire

As part of a joint venture between the University of Birmingham and the National Marine Aquarium we would like to ask you a few questions to help research into the effects of playing computer games on remote submersible training. (all data collected is anonymously)

Age: ☐ 5-7 ☐ 8- 10 ☐ 11- 14 ☐ 15- 18 ☐ 19- 25 ☐ 26- 35 ☐ 36- 45 ☐ 46-55 ☐ 56+

Gender: ☐ Male ☐ Female

Hobbies:
………………………………………………………………………………………......................

Which of the following do you play games on the most?

☐ Wii ☐ Playstation ☐ Xbox ☐ Hand held consoles ☐ PC ☐ None

Which type of game controller do you use the most

☐ Mouse and keyboard ☐ Joystick ☐ Xbox or Playstation game pad ☐ Wiimote

How many hours a week do you spend on the computer <u>not</u> playing games? (PC or Mac)

☐ 0 ☐ 1-5 ☐ 6-10 ☐ 11-15 ☐ 16- 20 ☐ 21- 25 ☐ 26- 30 ☐ 31+

What type of games do you play?

Game type	Not at all	Less than 1 hour per week	1-3 hours per week	4-8 hours per week	9-15 hours per week	16 or more hours per week
First person shooter (Halo, Halflife, Call of duty)						
Third Person (God of war, Zelda, Ratchet and Crank)						
Real time strategy (Command and Conquer, The Sims)						
Simulation (Flight sim, Burnout, GTA)						
Casual games (Card or puzzle games)						

Appendix D

ROV ExplorOcean Data Recording Sheet

ROV Project

Day:	Date:	Type of trial:

Student no.	Start time	No. of Collisions	Finish Time of First Half of Course	No. of Collisions	Finish Time

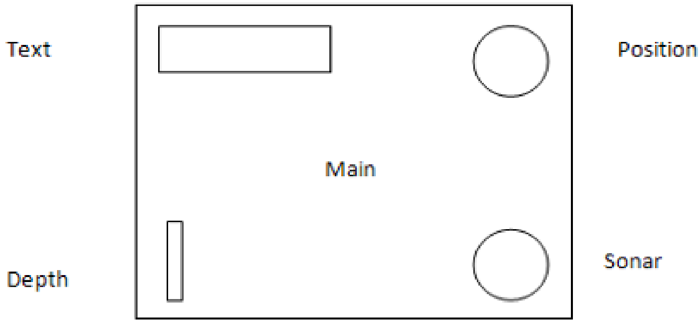
Appendix E

ROV Virtual Scylla Technical Aid Usage Ratings Sheet

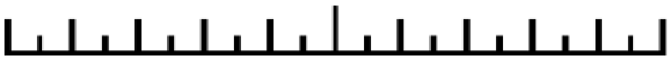
ID.. Date................................ Time....................................

Type...

How much did you use:-

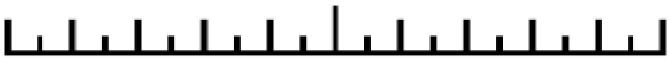


Text Information



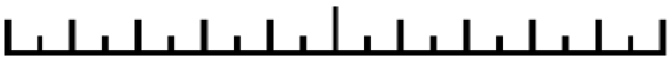
Low High

Position



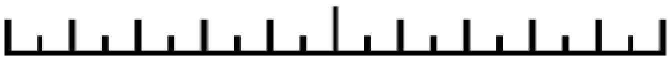
Low High

Depth



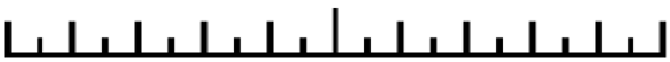
Low High

Sonar



Low High

Main



Low High

Appendix F

ROV Technical Specificaction Sheet

www.videoray.com

PRO 3 E

VideoRay
The Global Leader in Micro-ROV Technology

System Specifications

Depth Rating	152 m (500 ft)
Total System Weight	40 kgs (90 lbs) in 2 watertight Pelican cases
Submersible Size and Weight	3.8 kgs (8.4 lbs) / 12, 9, 8.5 inches (30.5, 23, 21 cm)
Speed	2.6 knots
Power Supply	100-240 VAC

Submersible Robot

Main Camera

Location:	Front - inside the pressure hull
Features:	Forward facing wide angle color camera.
Zoom:	N/A
Resolution:	570 lines of resolution
Sensitivity:	0.3 lux
Focus:	Wide Focus Range
Type:	Specify NTSC or PAL
View Angle:	90° horizontal 140° diagonal *underwater optimized wide angle*
Tilt:	Variable tilt with 180° vertical field of view

Second Camera

Rear facing high resolution black & white 430 liones of resolution / 0.1 lux

160° Vertical Tilt, Manual Focus Hi-Res Camera

Lighting

Forward

Type: 2 X 20w Halogen Lamps

Control: Variable Intensity

Second Camera - Rear

Type: Ultra high intensity LED array

External Auxiliary - Optional

Variable IntensityHalogen Lighting

Propulsion

Horizontal

Drive:	Geared
Motor:	Brushed
Propellers:	60 mm
Thrusters:	Two

Vertical

Drive:	Geared
Motor:	Brushed
Propeller:	45 mm
Thruster:	One

On-Board Sensors

Navigation: Compass (heading read out on control panel LCD)-option to overlay heading on monitor

Depth: Reading in feet or meters on control panel LCD - option to overlay depth on video monitor

Accessories and Sensors

9 pin accessory connector allows easy field integration of various instruments, sonar, and sensors

Rear Camera, LED Array, Accessory Port

Control Console

Comms Protocol:	RS-485 (if using with PC)
Case:	Watertight Pelican1550
Computer:	Optional
Display:	Requires User Supplied Monitor
Video out:	Analog composite video out
Recording:	Optional Digital Recording Package (Not Included)
Video Overlay:	Option to superimpose date, time, depth, and heading information on monitor
Audio Annotation:	Optional

Controls:
- Integrated Joystick for horizontal movement
- 3rd Axis joystick control (selectable for depth, camera tilt, lights, and manipulator)
- Vertical depth control with Auto DepthFeature
- Lighting, Camera Tilt, Focus Control, and manipulator control (manipulator not included)
- PC Pilot enabled (requires PC and XE package, not included)
- Front and Rear Camera toggle

Simple Controls - Requires User Monitor

Tether

Type:	Neutrally Buoyant Performance Tether
Length:	40 m (130 ft)
Extension:	Available - Not Included
Management:	Available - Not Included
Voltage:	48 volts DC - Maximum tether voltage
Expansion:	Modular
Breaking Strength:	1,000 lbs

Other Components

- Owners Manual
- Brass Ballast Weight Set
- Basic Tool Kit
- Hand Compass
- Tether Strain Relief
- System fits in two water-tight Pelican cases - airline checkable

XE Package Optional

152m (500 ft) Depth Rated Pro 3 E

Warranty

Two year limited warranty covering manufacturing and component defects. Upon or after warranty expiration this warranty can be renewed annually.