Honours Project Report

Teleoperation of Rescue Robots in Urban Search and Rescue Tasks

An Investigation of Factors which effect Operator Performance and Accuracy

Jason Brownbridge

Supervised By:
Dr James Gain

Department of Computer Science
University of Cape Town
2008
Abstract

We develop a novel system which allows the Nintendo Wiimote and Nunchuk to be used as control devices for teleoperating rescue robots, and find that these devices provide a good mapping for teleoperation tasks. However, they cannot be used for accurate head tracking due to the limited precision of the infrared camera used to measure lateral motion. We incorporate these devices as controllers in an existing Urban Search and Rescue simulator, with proven fidelity, and use this simulator to investigate the impact of several factors on operator performance and accuracy. These factors include different lighting conditions, camera control techniques, partial chassis visibility and the presence of a head-up display (HUD). We do this through two separate rounds of user experimentation, and find that different lighting conditions and camera control techniques impact significantly on operator performance, whereas the presence of a head-up display impacts significantly on operator accuracy.

For the lighting condition we find that performance is better when operators have greater visibility, which is not surprising. For the different camera control techniques we find that the best performance occurs with no camera control. This is surprising as it conflicts with previous research and we believe this is mainly due to time pressure on subjects, as well as, the low specificity required for the search and inspection task. We support this argument by examining subjects’ drive and camera usage patterns. We find that the presence of the HUD increases subjects’ accuracy and we attribute this to the greater situational awareness that the laser scanner display provides (which allows subjects to measure the distance between the robot and objects in its environment).

Keywords: H.5.2 [User Interfaces]: Evaluation/methodology, Input devices and strategies; I.2.9 [Robotics]: Operator interfaces; I.2.10 [Vision and Scene Understanding]: Motion, Perceptual reasoning; H.5.1 [Multimedia Information Systems]: Artificial, augmented, and virtual realities

Acknowledgements

I would like to thank my supervisor, Dr James Gain, for his ideas, insight, motivation and support throughout the course of this project. I would also like to thank Mr Stephen Marais, from the Robotics lab, for teaching me all about rescue robots and providing insights into possible areas of research. I would also like to thank my project partner, Graeme Smith, for all his support, as well as, for always being someone whom I could bounce ideas off of. Last but not least, I would like to take this opportunity to thank all the individuals who sacrificed an hour of their time to take part in our experiments.
# Table of Contents

List of Illustrations ........................................................................................................ iv

Figures .......................................................................................................................... iv

Tables ............................................................................................................................ v

1 Introduction .................................................................................................................... 1

1.1 Rescue Robots ............................................................................................................ 1

1.2 Research Problem ...................................................................................................... 2

1.3 Implementation ......................................................................................................... 2

1.4 Outline ....................................................................................................................... 3

2 Background .................................................................................................................... 4

2.1 Introduction ............................................................................................................... 4

2.2 Robot Exploration ...................................................................................................... 4

2.3 Situational Awareness ............................................................................................... 5

2.4 Virtual Environments ................................................................................................. 5

2.5 Augmented Reality ..................................................................................................... 6

2.6 Head-up Display (HUD) ........................................................................................... 7

2.7 Camera Control ......................................................................................................... 8

2.8 Real World Environments ......................................................................................... 9

2.8.1 Latency and Bandwidth ......................................................................................... 9

2.8.2 Mitigation Strategy ............................................................................................... 10

2.9 Performance Metrics ............................................................................................... 10

2.10 Evaluation .................................................................................................................. 11

3 Design and Implementation ........................................................................................ 12

3.1 Introduction ............................................................................................................... 12

3.2 System Overview ...................................................................................................... 12

3.3 Controllers ............................................................................................................... 13

3.3.1 Wii Overview ....................................................................................................... 13

3.3.2 Wii Limitations ................................................................................................... 14

3.3.3 Wii Head Tracking .............................................................................................. 14

3.3.4 Wii Software ....................................................................................................... 14
3.4 Middleware .................................................................................................................. 15
  3.4.1 Communication ................................................................................................. 15
  3.4.2 User Interface ..................................................................................................... 16
  3.4.3 Control Logic ..................................................................................................... 17
3.5 Simulator .................................................................................................................... 18
3.6 Evaluation .................................................................................................................. 19
4 Experiment Design ........................................................................................................ 20
  4.1 Introduction ........................................................................................................... 20
  4.2 Tasks ....................................................................................................................... 22
  4.3 Dependent Variables ............................................................................................. 22
    4.3.1 Definition and Operationalisation .................................................................... 22
    4.3.2 Measurement ................................................................................................... 22
  4.4 Questionnaires ....................................................................................................... 23
  4.5 Venue and Equipment ............................................................................................ 23
  4.6 Participants ............................................................................................................. 24
  4.7 Procedure ............................................................................................................... 24
  4.8 Completion .............................................................................................................. 25
  4.9 Round 1 .................................................................................................................. 25
    4.9.1 Hypothesis ......................................................................................................... 25
    4.9.2 Independent Variables .................................................................................... 25
    4.9.3 Design ............................................................................................................... 25
  4.10 Round 2 ................................................................................................................ 26
    4.10.1 Hypothesis ....................................................................................................... 26
    4.10.2 Independent Variables .................................................................................... 26
    4.10.3 Design ............................................................................................................... 27
  4.11 Evaluation ............................................................................................................. 27
5 Results ............................................................................................................................ 28
  5.1 Analysis Methodology ............................................................................................ 28
    5.1.1 Outliers ............................................................................................................. 28
    5.1.2 ANOVA ............................................................................................................. 28
    5.1.3 Procedure ......................................................................................................... 29
List of Illustrations

Figures

Figure 1 - Talon Robot and Control Unit used for Rescue Operations in the aftermath of the WTC attack..........................................................1
Figure 2 - Left Wiimote, Right Wii console ...........................................3
Figure 3 - Wii Nunchuk ........................................................................3
Figure 4 - The figures above (from [13]) show a combination of colour video, thermal imaging, direction and form a type of Augmented Reality .................................................6
Figure 5 - System Architecture Overview .............................................12
Figure 6 - Left Wiimote, Right Wii console ...........................................13
Figure 7 - Diagram illustrating how the Wiimote position and orientation can be determined. The two blue plus symbols indicate the infrared light as seen by the Wiimote camera as generated from the two infrared LEDs located in the sensor bar..............................13
Figure 8 - Wii Nunchuk ........................................................................14
Figure 9 - Example of WiiuseJ Test GUI showing Acceleration Data ..................15
Figure 10 - Left: Camera Tilt Display, Right: Laser Scanner and Camera Pan Display .................17
Figure 11 - Screenshot of Interface showing HUD .......................................21
Figure 12 - Screenshot of Interface showing HUD and Chassis ......................21
Figure 13 - Screenshot of Interface showing no HUD or Chassis ......................21
Figure 14 - Pool Balls ...........................................................................21
Figure 15 - Scatter plot showing Seek Time data........................................30
Figure 16 - Box and Whisker plot showing problematic outliers in Seek Time data......30
Figure 17 - Scatter plot showing Collision Interval data ..................................30
Figure 18 - Box and Whisker plot showing problematic outliers in Collision Interval data ......30
Figure 19 - Graph showing the impact of different Lighting Conditions on Seek Time ..........31
Figure 20 - Graph showing the impact of different Camera Control Techniques on Seek Time .........................................................................................32
Figure 21 - Scatter plot of Seek Time data .................................................33
Figure 22 - Box and Whisker plot showing problematic outliers in Seek Time data.......33
Figure 23 - Scatter plot of Collision Interval data .......................................33
Figure 24 - Box and Whisker plot showing problematic outliers in Collision Interval data ......33
Figure 25 - Graph showing the impact of HUD Visibility on Collision Interval .............35
Tables
Table 1 - ANOVA Analysis of Seek Time ................................................................. 31
Table 2 - Comparison of Different Lighting Conditions on Seek Time .................. 31
Table 3 - Comparison of different Camera Control Techniques on Seek Time ........ 32
Table 4 - ANOVA Analysis of Collision Interval .................................................... 32
Table 5 - ANOVA Analysis of Seek Time .............................................................. 34
Table 6 - ANOVA Analysis of Collision Interval .................................................... 34
Table 7 - Comparison of HUD Visibility on Collision Interval ............................. 34
1 Introduction

1.1 Rescue Robots

Human-robot interaction (HRI) currently takes many different forms. Robots can aid humans in performing dangerous tasks such as Urban Search and Rescue (USAR) [11, 12] as well as in the disposal of hazardous materials [8].

Robots can also provide assistance in more collaborative roles such as in high precision surgery or vehicle assembly and an increasing number of them are now operating in close proximity to humans, such as those used to assist the elderly [21] or handicapped [37].

Some robots are even used to provide entertainment or companionship for their human owners such as Sony’s Aibo.

Our research will focus on the use of robots for Urban Search and Rescue (USAR) tasks, as this is believed to be a near-ideal set of tasks for studying Human Robot Interaction [38].

USAR tasks involve the deployment of rescue workers (policeman, fire fighters and paramedics) as well as trained dogs to locate survivors and find victims’ bodies after catastrophes such as natural and man-made disasters. After the World Trade Center (WTC) attacks we saw the first actual deployment of rescue robots for these USAR tasks.

![Figure 1 - Talon Robot and Control Unit used for Rescue Operations in the aftermath of the WTC attack](image-url)

Rescue robots are ideally suited as they can enter voids too small or deep for a person, be deployed in hostile environments with extreme temperatures, high levels of toxicity and unstable structures. Furthermore, they can be equipped with a variety of sensors such as cameras, microphones, thermal imagers, CO2 detectors and laser scanners which allow them to successfully operated in areas with little to no visibility. Lastly, they can also be used to carry
medical payloads or other essentials to survivors, often allowing rescue workers enough time to dig up survivors.

This has particular relevance to the South African context where research is being done into the feasibility of using rescue robots to aid in mine rescues.

For these tasks, operators must have an awareness of the surroundings in situations where it often difficult to obtain accurate situational awareness [14, 46].

### 1.2 Research Problem

For our research we will explore, through user experimentation, the impact of several factors on operator performance and accuracy. We define operator performance, in terms of how quickly the operator can locate objects of interest within the environment, and accuracy in terms of how few collisions occur between the robot and objects in its environment.

The key factors we will be exploring include the impact of different lighting conditions, camera control techniques, presence or absence of a head-up display or HUD, as well the impact of making the front of the robot chassis visible to the operator.

We hope to establish, through user experimentation, whether the hypotheses are true:

1. The choice of lighting conditions impacts significantly on task performance or accuracy when teleoperating rescue robots.
2. The choice of camera control impacts significantly on task performance or accuracy when teleoperating rescue robots.
3. The partial visibility of the robot chassis impacts significantly on task performance or accuracy when teleoperating rescue robots.
4. The presence of a HUD (head-up display) impacts significantly on task performance or accuracy when teleoperating rescue robots.

We believe these hypotheses have great significance to USAR tasks as any techniques or tools for increasing operator performance or extension of the robots’ life span (through reducing negative interactions with the environment, such as collisions) may result in many lives being saved when these robots are deployed.

### 1.3 Implementation

In order test these hypotheses we modified a USAR simulator to allow for the creation of new interface elements, customized controller logic as well as the use of several new control devices, namely the Nintendo Wiimote and Nunchuk.
The Wii console is the latest generation of gaming console on offer from the Japanese gaming company Nintendo and is the current console market leader. It represents a significant departure from previous console development methodology, which focussed mainly on improving hardware performance. Instead Nintendo focussed on player interaction and developed a novel controller, namely the Wiimote.

We chose these devices largely because they were inexpensive and have an abundance of functionality, such as infrared camera, accelerometers and haptic feedback.

1.4 Outline

In Chapter 2 we discuss the background to our research and begin to frame the challenges that USAR tasks pose. This is followed by a discussion of our design and implementation in Chapter 3. Chapter 4 covers our experimental design and hypotheses and discusses the two rounds of experiments we held. In Chapter 5 we describe our analysis methodology as well as report our results which are then discussed in greater detail in Chapter 6. We conclude with Chapter 7 which contains the implications of our research as well as our main conclusions and areas of possible future work.
2 Background

2.1 Introduction
The field of teleoperation is a broad one, encompassing many diverse aspects. It entails both exploration of remote environments as well as manipulations of those environments. It may involve direct operator control or even the autonomous actions of robotic agents.

Of particular interest to our research is the teleoperation of reconnaissance and rescue robots. Thus this background chapter will focus on perception and navigation of robots within remote environments, as these two tasks are the most relevant to our area of research.

There has been some research drawing parallels between robotic exploration and navigation within the field of Virtual Environments [34] which we will examine in greater detail. Furthermore, there is strong empirical evidence to suggest greater situational awareness allows for more effective robotic control [26, 28] and thus this will also be investigated.

We believe this is a relevant area of research as the control of a viewpoint within a virtual environment has been shown to impact on all other operator tasks [22] and thus any novel interface which can improve upon existing designs may have a profound impact on this field.

2.2 Robot Exploration
Robotic exploration can be simply defined as the real-time control of a remotely located robot, either by a human operator or through the autonomous action of the robot itself. This process usually involves the use of a video feed, which is supplied by one or more cameras mounted on the robot [17].

According to the Endsley and Kaber’s taxonomy [16] and based upon work conducted by Luck et al. [33], we can distinguish between four representative modes of teleoperation, ranging from manual to semi-automated robotic control:

1. Teleoperation: Direct control of the robot by the operator through the use of input devices such as a joystick.
2. Guarded Teleoperation: This mode retains the direct control of the robot by the operator, but simple collision avoidance is performed. This allows the robot to remove any part of the motion which would result in a collision, thus preventing the operator from accidentally damaging the robot.
3. Obstacle Avoidance Waypoint Navigation: In this mode the operator plays a supervisorial role by selecting waypoints, and the robot attempts to traverse the waypoints as directly as possible using simple collision detection and no a priori knowledge of its environment.
4. Autonomous Waypoint Navigation: This differs from the previous mode, as the robot makes use of a priori knowledge of its environment to calculate the best possible route for traversing the supplied waypoints.

Direct Teleoperation provides the operator with complete control over the robot. However, this is not always desirable, especially if the operator has low situational awareness, as this can
result in the operator issuing commands which may cause the robot to interact dangerously with its environment or with itself, as in the case of articulating one of its limbs beyond its tolerance level. Guarded Teleoperation protects against this risk by using sensors and pre-programmed limitations to prevent operators issuing such commands. This is provided at the cost of less freedom of control over the robot, which in turn, may cause problems especially if the sensors are malfunctioning. Perhaps a better approach is to combine the two modes of operation, with Guarded Teleoperation as the default but allowing operators to override the pre-programmed limitations and directly teleoperate the robot. Obstacle Avoidance Waypoint Navigation is used for easily navigating between points of interest in the environment. This is not suited to USAR (Urban Search and Rescue) tasks, where operators need to inspect the environment closely, in a slow methodical manner in order to locate survivors and victims of the catastrophes. Similar reasoning can be applied to Autonomous Waypoint Navigation mode of operation.

However, regardless of the level of automation or mode of exploration utilized, human observation, judgment and supervision will still remain integral parts of the exploration process, especially with regards to rescue robots [33]. This is because USAR tasks are time sensitive, often operate in vastly different environments and require rescuers to make difficult decisions in order to save as many lives as possible, something which at present cannot be fully automated.

Milgram drew the link between robotic exploration and navigation within the field of Virtual Environments [34], which was subsequently strengthened by research done by Hughes and Lewis et al. into the implementation of a Virtual Environment for Teleoperation [23, 25, 44].

2.3 Situational Awareness
In order for situational awareness to be achieved, operators have to be provided with enough information to build comprehensive mental models of both the robot's external state, including such information as the robot's surroundings and orientation, as well as its internal state, which encompasses various aspects of its system status [28].

This is one of the key challenges faced in Human Robotic Interaction, as empirical evidence suggests that greater situational awareness allows for more effective robotic control [26, 28].

Thus, the design of user interfaces in order to enhance robot awareness and control is critical to the success of deployment of mobile robots in the field.

2.4 Virtual Environments
A virtual environment can be defined as a simulation of a real world environment [15]. Virtual Environment research is aimed at increasing the realism and sense of presence users feel, whereby the user has the impression that he or she is situated in the environment.

Telepresence in the field of Teleoperation is equivalent to presence in the field of Virtual Environments. It aims to provide the operators with enough environmental cues to enable
them to complete the task at hand successfully, without requiring them to feel the same degree of situatedness as is required in the definition of [24].

Navigation of a Virtual Environment involves manipulating one or more viewpoints such that the required objectives can be completed. This process of navigation can be broken up into two main tasks, namely wayfinding and travel [7].

Wayfinding can be thought of as the tasks that need to be performed in order to navigate a virtual environment and can be broken up into three main categories [23]:

1. Exploration: The aim of exploration is to increase spatial knowledge of the environment. This is of particular importance to rescue robots which are often deployed after a disaster and, as such, little or no correct data can be provided about their environment.
2. Search: This task aims to locate specific objects or entities within the environment.
3. Inspection: Inspection occurs once an object of interest has been located and involves manipulating the current viewpoint to examine the object more closely.

In the case of robotics the travel task can be thought of as the execution of wayfinding objectives [23] by manipulating the camera mounted on the robot through some camera control interface to achieve the desired viewpoint.

Many such interfaces exists, however, the most common ones involve a mapping of the six degrees of freedom which exist in three dimensional space onto a control device with fewer degrees of freedom. These are termed Mapped Controls [36].

2.5 Augmented Reality
Augmented Reality is a variation on typical Virtual Environments. It differs in its focus on enhancing the perception of reality and not on simulating it. This is done by superimposing or compositing virtual objects into the user's view of the real world [3]. One such example of combine real world and virtual objects is provided by Figure 4.

Figure 4 - The figures above (from [13]) show a combination of colour video, thermal imaging, direction and form a type of Augmented Reality
Augmented Reality aims to enhance a user’s perception and interaction with the real world. In terms of the field of Human Robot Interaction; it aims to increase the operator’s situational awareness.

Milgram et al. [35] note that head-up displays (HUD’s) which have mainly existed within military aviation environments, but which are gaining ground in other areas of application, fall into the realm of see-through Augmented Reality.

2.6 Head-up Display (HUD)
Baker et al. [4] discuss the success and failures of different interfaces developed for use in USAR applications and create a list of guidelines for effective interface design (of which three directly apply to our research):

1. Enhance awareness: Add elements to the interface which provide more spatial information about the robot and its environment to increase operators’ situational awareness.
2. Lower cognitive load: Provide fused sensor information, rather than requiring operators to mentally combine data from multiple sources.
3. Increase Efficiency: Minimize the use of multiple windows.

They describe and motivate three interface elements which are relevant to our research and can be used to aid in implementing the above guidelines.

Firstly, they observe that multiple studies have shown that operators often fail to re-centre the camera after panning and tilting. This can be corrected by automatically adjusting the camera when the robot starts driving, however, this can cause loss of situational awareness. They propose that a better solution is to make use of a visible indicator of the cameras orientation and thus retain situational awareness.

Secondly, they observe that instead of displaying information gathered from sonar and laser ranging in a separate window, it is better to place these readings around the video window and thus allow the operator to immediately see when sensors are detecting obstacles. They further suggest the use of bright colours to emphasize when obstacles are close to the robot and to fade the colours out when no obstacles are near.

Lastly, they note that an operator’s situational awareness can be enhanced by adding a map which emphasizes the location and orientation of the robot within its environment. However, they do not discuss how to generate such a map. This is beyond the scope of this work, suffice as to say, it is not a trivial task as there is often little or no a priori knowledge of the environment.
2.7 Camera Control
Camera control and configuration in the field of Teleoperation is equivalent to viewpoint control within the field of Virtual Environments.

In three dimensional space we have 6 degrees of freedom for manipulating the robot’s viewpoint of its environment: 3 for position (X, Y and Z) and 3 for orientation (Roll, Pitch and Yaw). Any teleoperation device must attempt to manipulate these degrees of freedom with fewer control options. Four main techniques were identified to accomplish this [23]:

1. Overloading: Extra degrees of freedom are achieved using different modes for the controller, thus the same physical buttons can perform different actions based on the state of the device.
2. Constraining: Impractical degrees of freedom are simply discarded. This would be case for ground robots, which would not be able to move vertically.
3. Coupling: This technique couples degrees of freedom as is the case for gaze-directed steering, where a robot would only be able to move in the direction it is facing.
4. Offloading: This method allows for certain elements to be controlled by an external source such as the collision avoidance algorithm or through the use of a pre-computed route.

Although overloading can be used to increase the level of control operators have over the robot, multiple states are generally frowned upon in Human Computer Interaction [39], and by the same reasoning, in Human Robot Interaction. Although more experienced operators would be able to leverage this additional control, it most likely would cause problems for novices. This is of particular concern as operators often do not have much training with these robots before they are deployed, and thus, interfaces should be kept as simple as possible.

Constraining simply refers to the limitations placed upon the operator’s viewpoint, which is provided by an actual camera with physical limitations. Thus, this mode of operation provides an intuitive mapping to the robot’s hardware and should be used in conjunction with other control techniques.

Coupling can be used to further reduce the cognitive load on operators by limiting the level control they have over the robot. Although this advantageous, as it reduces cognitive load, it should be used carefully as it also limits the functionality of the robot and by extension its usefulness.

Offloading allows for certain teleoperation tasks to be delegated in order to reduce the cognitive load on the operator, however, one must insure that the operator still retains good situational awareness, as otherwise, this may lead to cognitive dissonance.

In the field of Virtual Environments, Bowman et al. [6] examine two viewpoint configurations, namely that of gaze-directed steering and pointing with respect to relative and absolute motion.
With gaze directed steering, movement of the viewpoint is coupled to the direction of the view, whereas with pointing the user may look in one direction while moving in another. They found that the pointing technique was faster than the gaze directed technique for relative motion (absolute motion was equivalent). However, gaze-directed steering was found to be more intuitive due to the lower associated cognitive load, which results from clearly mapping between view and motion [6].

In the field of Teleoperation, Hughes et al. [24] examine three camera configurations, namely that of a fixed camera (which is equivalent to gaze-directed steering), an independent camera (which is equivalent to pointing) as well as a combination of the two.

They found there was no difference in performance between single camera and multi-camera configurations, although their results showed fundamentally different strategies were employed by operators using the two different systems. In terms of performance for search tasks (which equates to absolute motion in Virtual Environments) within the environment, they found no real difference between the fixed and independent camera configurations in terms of operator performance. However, for inspection task (which equates to relative motion in Virtual Environments) they found that the independent camera solution produced better operator performance than the fixed camera solution. They attributed this to the fact that the operators did not have to constantly reposition the robot, without a frame of reference, in order to inspect the object [23].

These results are in line with those produced by Bowman et al. [6] and indicate that the ability to independently control the camera is desirable when inspection of objects is required. However, neither party investigated allowing the operator to switch between views, which might provide an intuitive interface for the novice operator (due to the decreased cognitive load [6]), while not sacrificing the power provided by an independent view.

Although many more view configurations exist within the field of Virtual Environments, most of them are not practical for teleoperation, because of the physical limits imposed on the view by using an actual camera as the only source of visual information. Also, in the case of rescue robots, there is little or no a priori knowledge of the environment.

2.8 Real World Environments

So far we have investigated teleoperation and viewport control through the use of Virtual Environments; however when operating under real world conditions, factors such as signal strength, latency and bandwidth all contribute to the cognitive load placed on teleoperators.

2.8.1 Latency and Bandwidth

Latency is the delay between the sending of a packet and the receipt of it at its destination. In robotics latency can occur in two directions, either from the operator to the robot or from the robot to the operator [33]. In the first case there is a delay between sending a command and its execution, in the second case there is a delay between the robot executing a command and the operator perceiving the result(s).
Latency may be constant such as in the case of round trip latency in satellite communication or be variable as in the case of a disaster site where signal strength may vary.

Bandwidth describes the amount of data that can be sent over a communication channel within a given amount of time. Varying environmental conditions will result in variable bandwidth. This is especially relevant when low bandwidth radio channels are used to increase the range of transmission.

The practical effect of this on the operator is that he or she receives a reduced number of frames per second, or an equivalent reduction in quality of the visuals received [33].

2.8.2 Mitigation Strategy
Luck et al. [33] proposed a mitigation strategy whereby they increase the level of automation according to the Endsley and Kaber’s taxonomy [16]. In the experiments they conducted they tested what impact these different levels of automation had on operator performance under varying latency conditions [33].

Their results consistently showed that the higher levels of automation resulted in fewer driving errors, where driving errors included hitting the wall, over- or undershooting a turn, stopping along a straightway, and backing up, and increased driving speed. They also showed that latency with varying durations resulted in higher driving errors than latency with constant duration and that longer latency durations at low levels of automation resulted in increased driving errors.

Their qualitative results were in line with their quantitative ones, as their experiment participants reported greater difficulty in controlling robots under longer latency durations and even more so under varying latency durations.

2.9 Performance Metrics
Lewis et al. describe a framework for measuring human-robot interaction. The framework attempts to measure the navigation, perception, management, manipulation and social aspects of human-robot interaction.

They do this by establishing a list of task based metrics related to each of the aspects mention above, as well as a list of common metrics to test system, operator and robot performance [42].

Tasks are measured in terms of effectiveness, efficiency and effort. Effectiveness is a measure of how well the task was completed and for example in the case of navigation, can be measured in terms of: percentage of navigation successfully completed, coverage of area, deviation from the planned route and obstacles that were successfully avoided.

Efficiency is measure of how quickly the task was completed and can be measured in terms of time to complete task, amount of operator time for the task and amount of unplanned intervention.
Effort is a measure of operator workload and can be defined as the number of operator interventions per unit time, where interventions can be planned or unplanned. An alternative definition is the ratio of operator to robot time.

These metrics should prove useful in attempting to standardize human-robotic experimentation, especially in comparing camera configurations and the effects of various latency mitigation strategies on operator performance and cognitive workload.

2.10 Evaluation

From the literature we were able to identify USAR tasks as an ideal testing ground for Human Robot Interaction. We also saw that some modes of teleoperation were more suited for rescue robots than others.

We were able to find several parallels between navigation of a virtual environment and teleoperation. This will allow us to make use much of the research in virtual environments regarding viewpoint control.

The research indicates that different viewpoint control methods are appropriate, depending on whether one is travelling between absolute positions within an environment or inspecting objects.

We also examined how Augmented Reality and heads up displays can be used to increase situational awareness and thus increase operator performance and accuracy.

We briefly examined the use of different levels of automation to mitigate against latency. This is important as rescue robots are often deployed in less than ideal environments. The research showed that with increasing levels of automation one is able to reduce the cognitive load on the operator and decrease drive errors, while also increasing drive speed.

Furthermore we briefly examined a framework of performance metrics which could be useful when evaluating simulation results.
3  Design and Implementation

3.1  Introduction
Our system was initially designed to test the use of the Nintendo Wiimote and Nunchuk (for an introduction to these devices see section 1.3 and 3.3.1) as controllers for the teleoperation of rescue robots. This was largely due to the inexpensive nature of these devices, as well as the abundance of functionality they provide such as infrared camera, accelerometers and haptic feedback.

However, due to the shortage of robots as well as the limitations and costs involved in testing the system on actual robotic hardware and sensors, it was decided instead to make use of simulation as a viable alternative.

The simulator chosen provides a high fidelity simulation of Urban Search and Rescue (USAR) tasks, and has been validated against the standardized test arenas provided by the United States National Institute of Standards and Technology or NIST (see section 3.5 for more detail).

At this stage the focus of our design shifted from creating an entirely new system, including a simulator, to augmenting an already existing system by adding the Nintendo Wiimote and Nunchuk as a new type of controller.

We also realized that this provided an ideal opportunity to test the impact of different control techniques and controllers on teleoperation and thus much of the focus of the system implementation was to enable hypothesis testing in the field of Human Robot Interaction (HRI).

3.2  System Overview
The system was designed to follow a three tier approach as shown in Figure 5.

At the top layer are the physical control devices, which are used to send input into the system but can also serve as an alternate feedback mechanism for output such as haptic feedback.

Below that we have the middleware layer which has two main responsibilities. Firstly, it interprets messages from the top layer and converts them into commands for the simulator which is situated in the layer below. Secondly, it interprets messages from simulator and provides feedback on the robot and its

Figure 5 - System Architecture Overview
environment by updating the user interface in response to these messages. This layer is also responsible for displaying video feedback from the simulator, making use of the simulator’s client to accomplish this. Most of the focus of our implementation is on this layer, as we were able to leverage existing open source and proprietary technologies for the implementation of the other two layers.

The final layer is the simulator and is responsible for simulating the robot as well as its environments. The main requirements our system places on the simulator are the ability to simulate a large variety of robots and environments, in order for the system to be useful for hypothesis testing.

For our implementation language, we make use of the Java programming language, mainly due to the ease of development, rapid prototyping and network communication, as well as the availability of various open source software libraries which could be leveraged for implementation of our layered architecture.

The various components and implementation decisions are now presented in greater detail.

### 3.3 Controllers

#### 3.3.1 Wii Overview

The Wii console is the latest generation of gaming console (see Figure 6) on offer from the Japanese gaming company Nintendo and is the current console market leader.

![Wii Remote and Wii Console](image)

*Figure 6 - Left Wiimote, Right Wii console*

![Diagram illustrating how the Wiimote position and orientation can be determined](image)

*Figure 7 - Diagram illustrating how the Wiimote position and orientation can be determined. The two blue plus symbols indicate the infrared light as seen by the Wiimote camera as generated from the two infrared LEDs located in the sensor bar.*

The Wii Remote or Wiimote is the primary controller for the console and uses a combination of a built-in accelerometers and an infrared camera to determine the remote’s position in 3D space, relative to a sensor bar which emits infrared light (see Figure 7).

The Wiimote has an expansion port located at the bottom of the controller that allows other devices such as the Wii Nunchuk to be attached.
The Wii Nunchuk is shown in Figure 8 and possesses a joystick control which provides a natural association to the typical joystick controllers used in traditional teleoperation.

Finally, the Wiimote uses Bluetooth to communicate with the main console unit and it also includes a built in speaker to provide audio feedback, as well as rumble (Wiimote vibrating) capability for providing haptic feedback.

3.3.2 Wii Limitations
The one major limitation of this system is the fact that the Wiimote is only equipped with an accelerometer. In order to detect motion, the accelerometer measures the effect of gravity on a mass, and as such, it cannot be used to detect lateral motion, in other words motion perpendicular to the force of gravity.

This means that although the accelerometer can detect motion in the “up-down” plane, it is unable to detect motion in the “left-right” plane. In order to compensate for this, Nintendo equipped each Wiimote with an infrared camera which when used in conjunction with the sensor bar allows the Wiimote to detect lateral motion of about 30 degrees in either direction [18]. This is mainly due to the limited field of view of the infrared camera.

In order to correct this limitation, Nintendo plans to release a new accessory for the Wiimote known as the Wii Motion-Plus. This accessory will include, among other things, a gyroscope which will allow for lateral motion to be detected without the use of the sensor bar. This will mean that the Wiimote will be able to be used for full 6 degrees of freedom tracking in Virtual Environments [18].

3.3.3 Wii Head Tracking
Head tracking using the Wiimote was first popularized by Johnny Chung Lee [30], however, his technique was only suitable for tracking the translation of a body not the translation and rotation, and could more accurately be termed positional head tracking.

This poses a problem for our system and as such we adapt the design described in his paper [30] and instead mount the Wiimote on the subject’s head. This differs from his design where the Wiimote is placed in a fixed location and the infrared beacons are mounted on the subject’s head. This allows us to track the vertical (up/down) motion of the subject using the accelerometer as well the lateral (left/right) motion using the infrared camera.

3.3.4 Wii Software
In order to communicate with the Wii controllers, we make use of the open source WiiuseJ Library developed by Guilhem [31]. This library utilises the JNI framework to provide a Java wrapper for the open source Wiiuse C API developed by Laforest [45].
It provides full fidelity interaction, with the exception of audio output, for all the various Wii Controllers, including the Wiimote and Wii Nunchuk.

The WiiuseJ library is structured to allow listeners to register interest for particular events and then notifies the listeners when the event they are interested in occurs. This differs from the architecture provided by the Wiiuse C Library which requires developers to manually handle event polling.

The latest version of the library available at the time of writing is version 0.12b and this version includes several bug fixes contributed to the project by our development team.

The library is also bundled with several sample applications, as well as a GUI interface to aid in the analysis of input provided by the Wiimote and Nunchuk. A sample of the interface is provided in Figure 9 below.

![Figure 9 - Example of WiiuseJ Test GUI showing Acceleration Data](image)

### 3.4 Middleware

The middleware is where most of our development took place. This component is responsible for communicating with the simulator as well the Wii controller devices, displaying the user interface and implementing control logic.

#### 3.4.1 Communication

The Gamebots interface makes use of a text based protocol for communicating with clients. Each client is responsible for one particular robot and receives status update messages for that robot at every script engine tick. The tick rate is roughly 3 to 5 ticks per second, but this should not be confused with the rendering or physics engine tick rate which is much faster. These messages include updates on the robot’s status, including such things as position, orientation and battery life, as well as data from the various sensors such as touch, sonar and laser scanner.
The messages that are received must then be parsed by our communication’s module in a timely fashion in order to update the internal data structures so that they represent a consistent and up-to-date view of the state of the robot and its environment.

A callback interface is utilized, whereby components can register interest in particular messages. This allows for a modular design with each component handling only certain messages. This proved useful especially when constructing the various HUD elements for the user interface, as each element could register interest in specific sensor data.

Our communication’s module is also responsible for marshalling instructions from the control logic module and sending them as protocol messages, which are understood by the Gamebots interface. These messages include both navigation and camera control commands.

3.4.2 User Interface
The user interface consists of two main components: firstly, the video feedback from the robot’s camera and, secondly, the various heads up display (HUD) elements displaying the robot status and sensor data.

Two possibilities existed for capturing video feedback from the Unreal Engine. Firstly, one can capture the back buffer used to render the scene in the Unreal Client and perform some basic image processing on the data to create a sequence of frames, which can then be sent over the network. Although this technique provides the most flexibility in terms of frame rate control and post processing of the images it is limited by poor frame rates due to the overhead of locking the back buffer in order to create a copy of the data. Secondly, one can embed the Unreal Client within another application through the use of certain Win32 API calls. However, this poses a slight problem as these calls are not available within the Java programming language, which was our implementation language. This limitation can be overcome by making use of the JNI framework to make the required native calls to the Win32 API.

After careful consideration, we choose to use the second technique as it was the most easily implemented and fulfilled all our requirements for the hypothesis testing we wished to perform.

The second main component of the interface, namely the HUD, consists of two main elements:
1. A Tilt display element.

The cameras tilt is shown as an elevation arc with respect to the horizontal, and the left half of Figure 10 depicts a negative elevation of about 5 degrees. Our motivation for including this element is that operators often find it difficult to orient the camera direction and current pose of the robot. By including this element we hope to reduce the cognitive load this normally causes.

The Laser Scanner and Pan element has two roles: firstly, it indicates the current pan of the camera by overlaying the horizontal direction of the camera on top of the laser scanner.
Secondly it illustrates the distance between the robot and objects in its environment. The laser scanner works by rotating a laser by 90 degrees in either direction and emitting a beam at intervals of about one degree (this resulted in 180 samples being taken). The distance the beam travels before impacting with objects in the environment is recorded and the measurements form a semicircle of readings around the front of the robot, as the right half of Figure 10 depicts.

Thus, when viewing the Laser Scanner and Pan element (as shown in the right half of Figure 10), one can measure the distance between the robot and objects in its environment, by observing the length of the red lines, originating at the centre of the element. Short lines indicate objects are close by, whereas long lines indicate objects are further away. Thus Figure 10 indicates that there is an object roughly to the left of the robot’s current position.

Our motivation for including this element was the hope that it would allow users to accurately measure the distance between the robot and objects in the environment, thus increasing their situational awareness, and reducing the number of collisions.

All HUD elements implement a common interface which allows them to receive notification of simulator messages. This allows each HUD widget to be self contained, possessing only the logic needed to process a subset of messages.

3.4.3 Control Logic
The control logic module is responsible for interpreting input from the Wii Controllers and converting it into instructions for the simulator based on which mechanism of control is selected.

Our system was designed to allow for two basic types of robotic control (although this is easily extensible due to the modular design of the system):

1. Navigation Control: Navigating the robot through a remote environment.
2. Camera Control: Controlling the various cameras situated on the robot’s chassis.

The mechanisms for implementing these two types of control are as follows:

1. Nunchuk Joystick Events: Navigation control can be accomplished simply by interpreting the angle and magnitude supplied by joystick events as navigation commands. Camera control can be performed in a similar fashion, merely requiring an additional button event to differentiate camera from navigation commands. This forms the basis of direct control in the system.
2. Wiimote Motion Events: Camera control can also be accomplished by using Wiimote Motion events to facilitate head tracking, where each event supplies the current pan and tilt angle, which can then be used to alter the camera’s orientation. If the Wiimote is mounted on the subject’s head then these angles correlate to the orientation of the head, which allows the camera to be rotated to point in the direction the subject is looking. This forms the basis of indirect control in the system.

Every controller must implement the following two methods:
1. A ‘look at’ method, which takes a camera, as well as an orientation pose and results in the given camera being rotated to match the supplied orientation pose within some threshold.
2. A ‘drive at’ method, which takes a direction as well as a normalized speed value and results in the robot driving in the given direction at a speed that is calculated by multiplying the supplied normalized speed value by the robot’s maximum speed.

This interface allows for the implementation of several different types of camera and steering mechanisms such as Skid steering (where the robot has two tracks that can work independently of each other) or Ackerman steering (where the orientation of the front wheels of the robot can be controlled). This is because each robot can have different camera and drive modules which implement the required logic to convert the parameters of the two methods into the relevant command requests for the communication module.

Thus a controller is implemented for each of the two mechanisms but these are able to share common camera and driving modules and only differ in how the Wiimote and Nunchuk events are interpreted.

3.5 Simulator

Although several simulators were available to choose from, we decided to use de facto standard of the RoboCup Rescue Simulation League [5], namely the Urban Search and Rescue Simulator known as USARSim [44, 45].

This open source simulator is built on top of the, Unreal Engine 2.0, a proprietary game engine created by Epic Games [27]. While the internal structure of the Unreal Engine is proprietary, one can work around this constraint by utilizing the Gamebots [1, 32] interface developed by the University of Southern California’s Information Institute. This allows an external application to exchange bi-directional information with the engine

USARSim then sits above the Gamebots interface and provides a standardized way to simulate robot actuators and sensors. Extensive utilization of and research into USARSim has shown that it behaves in a predictable manner with a high correspondence to reality [10, 29, 47].

In order for simulation results to be valid for real hardware, the accuracy of the simulation model must be verified. To ensure the validity of simulations, the United States National Institute of Standards and Technology (NIST) proposes standardized test methods that can easily be replicated in both computer simulation and physical form. The actual robot can then
be tested against the computer model and thus the simulation can be calibrated to replicate similar performance for equivalent tests [40].

USARSim was developed with this in mind, as a high fidelity simulation to be used in Urban Search and Rescue (USAR) tasks, and as such, supports accurately rendering user interface elements (including camera video), modelling robot behaviour and representing the remote environment [29].

It has been validated for use with the NIST test arenas [44, 45], and further work done by Carpin et al. [9] has shown that it accurately models the physics, environment and the robot itself.

The ability to model these standard NIST test arenas, as well as, a large variety of robots intended for Urban Search and Rescue (USAR) tasks was one of the key reasons why it was chosen.

Since the Unreal Engine was initially designed for development and deployment of networked multi-player 3D games, it provides a solid foundation for the simulator and solves many of the problems that a traditional simulator would face such as the modelling, animation and rendering of virtual environments.

It also provides a comprehensive set of tools for developing objects and the environment (Unreal Editor) and it is possible to define the behaviour of in-game assets through the use of an ad-hoc scripting language known as Unreal Script.

Physics simulation is handled by the Karma Physics Engine [2], which handles the dynamics of rigid bodies transparently.

The simulator is built around the Client/Server architecture of the game engine and as such the control logic for robots may be programmed in any language that supports network communication. This has several advantages, such as the ability to offload complex computations from the simulator and thus decouple simulation from intelligence processing [47].

### 3.6 Evaluation

In this chapter we describe our design for using the Nintendo Wiimote and Nunchuk to teleoperate a robot within a simulated environment.

We see that although these controllers provide a good mapping for the tasks we require, they do have some limitations, especially in terms of their ability to provide accurate head tracking.

Although we use simulation as testing ground for our system, we show that USARSim is a high fidelity simulator and has been validated for use in USAR tasks.
4 Experiment Design

4.1 Introduction

Two rounds of experiments were conducted in order to determine the impact of several variables on both operator performance and accuracy when remotely controlling a rescue robot.

The first round of experiments investigated the impact of different lighting conditions, normal and dim, as well as different camera control techniques, including no camera control, manual camera control implemented using the joystick situated on the Nintendo Nunchuk, and gaze directed camera control implemented using the Nintendo Wiimote for head tracking.

The second round of experiments was designed based on observations made during the first round, feedback received from subjects, as well as advice from several expert users in the Robotics Lab. These various sources of information correlated around the hypothesis that subjects were having trouble judging the size of the robot due to the limited field of view (which did not allow the subjects to see any part of the robot chassis) and thus resulted in difficulty in navigating around obstacles and through narrow gaps, such as doorways.

Thus, the second round of experiments was designed to test the impact of the partial visibility of the robot’s chassis, as well as the impact of HUD (head-up display) elements on subject performance and accuracy. This was compared with the results from the first round of experiments.

The three figures below illustrate the different conditions that will be tested in the second round of experiments. Figure 11 shows the default interface which includes the HUD but does not show the front of the robot chassis. Figure 12 shows the default interface with addition of the robot chassis, which is visible above the main section of HUD elements at the bottom of the screen. Lastly, Figure 13 shows the interface with the absence of all HUD elements, care was taken that the video feedback window remained the same size for all conditions, so as to not introduce any additional bias.
Both rounds of experiments shared the same tasks and thus the same dependent variables as well as the same subject selection criteria in order to allow for comparison of results.

For tasks involving simulated victims, pool balls (similar in appearance to those depicted above) were used instead of a more accurate representation for the following two reasons:

1. The use of pool balls eliminated possible bias due to the negative psychological impact of using a realistic victim simulation. This was of great concern as subjects had no prior
training or counselling in order to deal with the post traumatic stress of experiencing the aftermath of a disaster.

2. The wide range of colours simulated a realistic field environment where victim identification might vary depending on how noticeable the victim is within the environment. This correlates to certain pool balls blending in with the surrounding environment, whereas others stand out.

4.2 Tasks
Subjects were given two tasks for each of the experiments:
1. Locate as many of the first eight pool balls as possible in the allocated time.
2. Avoid collisions while locating pool balls.

These tasks were based on real applications and limitations of rescue robots. The first task was based on the requirement for teleoperators to be able to locate victims or identify areas of interest within an environment.

The second task was based on the requirement for teleoperators to protect the robot from damage as well to ensure minimal interaction with the environment due to the possible negative affects this would have on an unstable environment (rubble after earthquake or a partially collapsed mine shaft).

4.3 Dependent Variables

4.3.1 Definition and Operationalisation
There were two independent variables with each one being related to one of the tasks:
1. Seek Time: the mean interval between locating pool balls, where seek time is a measure of performance and measured as $Time = \frac{\text{Total Time Taken}}{\text{Number of Pool Balls Found}}$, with best performance occurring when this value is minimized.
2. Collision Interval: the mean interval between collisions, where collision interval is a measure of accuracy and measured as $\text{Collision Interval} = -\frac{\text{Total Time Taken}}{\text{Number of Collisions}}$, with greatest accuracy occurring when this value is maximized.

4.3.2 Measurement
1. $\text{Total Time Taken}$ was recorded manually on the second laptop (see section 4.5) with each camera control technique given a maximum amount of time and the facilitator stopping the countdown if the participants located all eight pool balls before the time was up.
2. $\text{Number of Pool Balls Found}$ was recorded manually with subjects identifying pool balls by number or colour to the facilitator verbally. The facilitator would then confirm if the subject was correct by examining a second display showing the subject’s field of view. If the subject correctly identified the ball then the find was recorded as well as the time at which it occurred. Although this measurement was recorded manually, the mean time taken to locate balls (36.9 seconds) was at least an order of magnitude larger than any error introduced by the recording process. Furthermore, the effect of any such error would be greatly reduced due to the nature of the repeated measure
experiment design and, as such, we can safely ignore any impact it may have on the our results.

3. **Number of Collisions** was recorded automatically using the touch sensors supplied by USARSim, each time the sensor was triggered a log entry was created allowing for the total number of collisions to be determined.

### 4.4 Questionnaires

The standard Slater-Usoh-Steed (SUS) questionnaire (see section 9.2.2) was administered to subjects after each step of both rounds of experiments. This questionnaire measures subjects’ perception of physical presence within the virtual environment [41]. The initial motivation for administering this questionnaire was to determine if a positive response, in terms of presence, correlated to increased performance or accuracy as measured in terms of decreased Seek Time and increased Collision Interval respectively.

However, when analyzing the SUS Count (the mean number of 6 or 7 scores) and SUS Mean (the mean score) in relation to the different experimental conditions, no significant correlation could be found between presence and performance or accuracy, and thus, no results have been reported.

### 4.5 Venue and Equipment

Experiments were conducted within a closed room, with one participant taking part at a time.

Two laptops, a head mounted display, as well as two Nintendo Wiimotes and one Nintendo Nunchuk were used.

The i-Glasses PC/SVGA Pro 3D head mounted display (HMD) was used to display all visuals to subjects. This HMD has two independent LCD (Liquid Crystal Display) screens, one for each eye, which enable stereoscopic 3D viewing. However, the stereoscopic functionality was not used during experimentation.

Each of the LCD screens was capable of outputting visuals at a resolution of 800x600, and thus all interface elements had to be designed to take this into account.

The first laptop was used to display the simulator and record raw data, the second laptop was used to randomize the experiment and map order, as well as to time the subjects while they located the pool balls.

In terms of equipment that directly impacts upon the experiments, only the first laptop played a role, as it ran the simulator and thus determined the limit on the frame rate used for the simulation. For both rounds of the experiment the same laptop was used:

- **Processor**: Intel T2400 1.83GHz,
- **Memory**: 3GB DDR2 667MHz,
- **GPU**: Mobility Radeon X1600 256MB.
4.6 Participants
Participants were chosen from the student body at the University of Cape Town and largely consisted of Computer Science students, although there were also several Engineering, Commerce and Humanities students. All participants were required to have previous computing experience but no robotic teleoperation experience, in order to remove any possible bias this might introduce. No participants who took part in the first round of experiments were allowed to take part in the second round.

Participants previous virtual environment experience was largely due to their involvement in computer gaming, which varied from little to excessive, however, no participants had experienced a full on virtual environment experience (a head mounted display with head tracking enabled).

In total 14 participants over 2 days took part in the first round of experiments; however two participants had to be disqualified. The first one due to incomplete data collection on the facilitator’s part and second due to the introduction of an extraneous variable midway through the experiment.

In total 17 participants over 3 days took part in the second round of experiments; however, one participant had to be disqualified due to mild simulator sickness interrupting the experiment.

We did not take gender into account as a possible experimental bias, due to ethical and practical concerns.

4.7 Procedure
1. Subjects were assigned a randomized experiment and map order.
2. Subjects had to complete and sign a waiver (see section 9.1) acknowledging that they understand all risks posed by the experiment including the possible discomfort that simulator sickness might pose.
3. Subjects were then required to complete a questionnaire (see section 9.2.1) listing previous computer, virtual environment as well as teleoperation experience and to indicate any visual impairment which might bias the experiment.
4. Subjects were given a list of instructions that (see section 9.3):
   a. Described the Nintendo Wiimote and Nunchuk, as well as the various elements of both controllers.
   b. Explained how to use the two devices to control both the robot and camera.
   c. Described the various elements of the HUD and explained how to use them.
   d. Informed the subject of the tasks they would have to complete during the experiment, including giving a detailed description and image of the eight pool balls as well as the procedure to use when reporting a ball’s location.
5. Subjects were then allowed 5 minutes within a test environment in order familiarize themselves with the controls and robot as well as the tasks they would have to perform.
6. The simulator was initialized with the correct experiment and map for the next condition to be tested.

7. Subjects then were given a set of experiment instructions for the specific condition being tested (see sections 9.4 and 9.5), including a refresher on the tasks that they had to perform as well as the controls available.

8. Subjects then were given a maximum of 5 minutes to locate all pool balls. If subjects found all balls before the time was up the experiment was terminated early.

9. Subjects were required to complete a Slater-Usoh-Steed questionnaire (see section 4.4 for description and motivation and section 9.2.2 for a sample of the questionnaire).

10. Steps 6, 7, 8 and 9 where repeated until all conditions were tested.

4.8 Completion
Participants were compensated with R20 upon completion of the experiment; the one participant who fell out due to mild simulator sickness was also compensated with the full amount.

4.9 Round 1
This round of experiments investigates the impact of different lighting conditions, normal and dim, as well as different camera control techniques, including no camera control, manual camera control implemented using the joystick situated on the Nintendo Nunchuk, and gaze directed camera control implemented using the Nintendo Wiimote for head tracking.

4.9.1 Hypothesis
Two hypotheses were tested:

5. The choice of lighting conditions impacts significantly on task performance or accuracy when teleoperating rescue robots.

6. The choice of camera control impacts significantly on task performance or accuracy when teleoperating rescue robots.

Thus the two null hypotheses are:

1. The choice of lighting conditions does not impact significantly on task performance and accuracy when teleoperating rescue robots.

2. The choice of camera control does not impact significantly on task performance and accuracy when teleoperating rescue robots.

4.9.2 Independent Variables
There were two independent variables which where varied during the course of the experiment.

1. Lighting Conditions: the lighting conditions for the experiments which had two states - Normal and Dim.

2. Camera Control Technique: the technique use to control the camera which varied between No Control, Manual Control and Head Tracked Control.

4.9.3 Design
The experiment was a one-way mixed design consisting of both a between-subjects as well as within-subjects independent variable.
Firstly, each subject was given one of two possible lighting conditions, either dim or normal lighting. This was done by assigning roughly the first half of the subjects to the normal lighting level and the second half to the dim lighting level. This created two groups of subjects and constituted the between-subjects part of the experiment.

Secondly, each subject was exposed to all three different camera control techniques assigned in a randomized order on a per subject basis in order to reduce the bias introduced by the learning effect. Each control technique was tested on one of six randomized maps to further reduce any bias. This constituted the within subjects part of the experiment, as all subjects were exposed to all conditions.

Maps were designed so that they contained the same obstacles; however, the arrangement of obstacles and placement of pool balls was varied to reduce the bias introduced by the learning effect.

Subjects were allowed 5 minutes to familiarize themselves with the system and tasks, and then were given 5 minutes to complete each experiment condition. Total experiment time averaged roughly 50 minutes.

4.10 Round 2
This round of experiments tests the impact of making the front of the robot chassis visible to the operator, as well as the impact of the presence of a HUD (head-up display) on performance and accuracy.

4.10.1 Hypothesis
Two hypotheses were tested:
1. The partial visibility of the robot chassis impacts significantly on task performance or accuracy when teleoperating rescue robots.
2. The presence of a HUD (head-up display) impacts significantly on task performance or accuracy when teleoperating rescue robots.

Thus the two null hypotheses are:
1. The partial visibility of the robot chassis does not impact significantly on task performance and accuracy when teleoperating rescue robots.
2. The presence of a HUD (head-up display) does not impact significantly on task performance and accuracy when teleoperating rescue robots.

4.10.2 Independent Variables
There were three independent variables which where varied during the course the experiment.
1. Partial Robot Chassis Visibility: whether the chassis was visible or not, this varied between True and False.
2. HUD Visibility: whether the HUD was visible or not, this varied between True and False.
3. Camera Control Technique: the technique use to control the camera which varied between No Control and Manual Control.
4.10.3 Design
The design of this experiment was a three-way within-subjects design, with each subject being exposed to a combination of three independent variables, namely Partial Robot Chassis Visibility, HUD Visibility and Camera Control Technique. This allowed for comparisons between this round of experiments and subjects from the normal lighting level of the previous round of experiments.

Maps were reused from the first round of experiments and subjects were allowed 5 minutes to familiarize themselves with the system and tasks, as well as, 5 minutes to complete each experiment condition. Total experiment time again averaged roughly 50 minutes per subject.

4.11 Evaluation
The experimental design and methodology described in this chapter is geared towards improving operator performance in two areas. Firstly, we explain how we will test the impact of different techniques for decreasing the time it takes to find objects of interest within the environment. Secondly, we discuss how we will test different techniques and factors for decreasing the amount of negative interaction with the environment, in terms of reducing the number of collisions which occur between the robot and objects within its environment.
5 Results

5.1 Analysis Methodology

5.1.1 Outliers
Before performing any analysis on the data it was first analysed to see if any extreme outliers existed.

An extreme outlier was defined as:

\[ x(n) \geq m + 6(u - m) \quad \text{or} \quad x(n) \leq m - 6(m - l), \]

where \( x(n) \) is the data point under consideration, \( m \) is the mean, \( u \) is the upper quartile and \( l \) is the lower quartile. [43]

Any value satisfying either of the two conditions listed above is regarded as an extreme outlier, and the subject to whom the data point belongs is removed from consideration. Extreme outliers result in much higher variance than normal, which may result in the ANOVA test (described below) returning an incorrect result. This is due to the nature of the test which measures the variance between the means of the different experimental conditions. High variance in a condition indicates that the probability that it has no effect on the dependent variable is low, and as such, can cause the test to incorrectly return a positive result for significance.

5.1.2 ANOVA
In order to test our hypotheses, we made use of the analysis of variance technique, also known as an ANOVA. An ANOVA is used to test the difference between the means of one or more dependent variables across several samples [20]. We chose this test for two reasons: firstly, we sometimes had more than two groups for our independent variables, and as such, would have needed to conduct multiple t tests. This would have inflated the error rate. Secondly, it is not known whether the responses (dependent variables) are normally distributed. The ANOVA technique is known to be robust under these conditions as will be explained below.

In order to use this analysis technique we must satisfy two main conditions. Firstly, for each sample we must show that there is a normal distribution for the dependent variable being tested. Secondly, we must demonstrate that the dependent variable shows the same variance in all samples being compared [20].

However, it has been established that the ANOVA is robust under these conditions and as such they may be violated as long as the dependent variable’s distribution is not significantly skewed, peaked or flat [19, 20].

A measure of the normality of a distribution is the Shapiro-Wilks statistic (Becker, 1999). A significant p-value on this test implies that the sample is from a non-normally distributed population and as such we rejected any p-values outside of the 95% confidence interval.

If the distribution failed to meet our criteria for the Shapiro-Wilks statistic, we further tested the Skewness and Kurtosis of the distribution to determine if it was significantly skewed, peaked or flat.
If the distribution failed to satisfy these requirements, we transformed it by applying the logarithmic function and then re-evaluated all tests on the transformed distribution.

Some distributions did indeed need to be transformed, however, all distributions which initially failed these tests passed when transformed in this manner.

5.1.3 Procedure
Thus our procedure for analyzing results is as follows:
1. Remove any outliers as per the criteria above.
2. Test the normality of the distribution using the Shapiro-Wilks test. If the distribution passes this test continue to step 5.
3. If the distribution fails this test (p-value outside the 95% confidence interval), then attempt to determine if the distribution is significantly skewed, peaked or flat by using the Skewness and Kurtosis tests.
4. If the distribution passes these tests continue to step 5, otherwise if no transformation has been previously made transform the distribution by applying the logarithmic function and continue to step 2. If a transformation was previously made abort the analysis and determine if a different transformation function can be used to create a more normal distribution from the given distribution.
5. The distribution now satisfies the criteria of the ANOVA technique, so perform the technique and determine if any of the independent variables significantly impacts upon the value of the dependent variable (p-value within the 90% confidence interval).

5.2 Round 1

5.2.1 Removal of Outliers
Following the first step of our data analysis methodology described above, any problematic outliers were removed from both data sets (Seek Time and Collision Interval).

In Figure 16 it is clearly evident that only one such outlier exists. It was determined that this outlier was indeed problematic as per the definition and as such only the subject to whom this data point belonged was excluded from consideration.

From Figure 18, one can see that there are four outliers, however only two of them were determined to be problematic, as per definition, and as such only the subjects to whom these data points belonged were excluded from consideration.
Thus in total three subjects were excluded in order to maintain experimental validity and as such all further analysis in this section was conducted on the original data excluding the outliers found above.

The subject who was excluded due to poor Seek Time performance took longer than average to adapt to the interface, as only one of the results recorded for the subject was an extreme outlier. This result corresponded to the condition testing Seek Time when using head tracking to control the camera. Since this condition was not part of the sandbox exercise and was also the first condition this subject was exposed to, it may have resulted in confusion, especially due to the lack of any previous Virtual Environment experience.

One of the subjects excluded for performing significantly better than the other subjects in terms of Collision Interval had previous Robotic Experience as recorded in the pre-experiment questionnaire (see section 9.2.1 for a sample of the questionnaire). The other subject, who was also excluded for performing significantly better, had no discerning features to support the exclusion.

5.2.2 Seek Time

Although the Seek Time data failed the Shapiro-Wilkes test for normality, as the Null hypothesis could not be rejected, the values for Skewness and Kurtosis were not extreme enough to preclude the use of the ANOVA technique.
As can be seen from Table 1 below, both the Camera Control and Lighting conditions were found to significantly impact upon Seek Time, within the 99% and 100% confidence intervals respectively.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Degrees of Freedom</th>
<th>Sum of Squares</th>
<th>Mean Square</th>
<th>F Value</th>
<th>Pr(&gt;F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camera Control</td>
<td>2</td>
<td>1132.41</td>
<td>566.20</td>
<td>6.61390</td>
<td>0.00950</td>
</tr>
<tr>
<td>Lighting</td>
<td>1</td>
<td>1647.82</td>
<td>1647.82</td>
<td>19.24830</td>
<td>0.00062</td>
</tr>
<tr>
<td>Subject</td>
<td>7</td>
<td>1979.80</td>
<td>282.83</td>
<td>3.30380</td>
<td>0.02716</td>
</tr>
</tbody>
</table>

Since both Camera Control and Lighting were significant, further analysis is required to determine how each of these factors impacted on Seek Time.

In order to determine the impact of different levels of the lighting condition we plotted the means for both levels against each other, as shown in see Figure 19 below.

As one can clearly see from Figure 19 as well as Table 2 below, subjects performed significantly better at locating balls under the Normal Lighting condition, as Seek Time under this condition was minimized.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Mean Seek Time</th>
<th>Standard Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal Lighting (n=15)</td>
<td>29.89</td>
<td>2.86</td>
</tr>
<tr>
<td>Dim Lighting (n = 12)</td>
<td>45.61</td>
<td>5.39</td>
</tr>
</tbody>
</table>

In order to investigate the impact of different Camera Control Techniques on Seek Time, the Mean Seek Times for each technique were plotted against each other as shown in Figure 20.
As one can see from both Figure 20 and Table 3, subjects performed best with no camera control and worst with joystick control. Although it is interesting that head tracking control did indeed perform better than joystick control, this was not completely unexpected. However, the fact that no camera control resulted in the best performance is somewhat surprising and merits further discussion (see section 6.1).

### Table 3 - Comparison of different Camera Control Techniques on Seek Time

<table>
<thead>
<tr>
<th>Condition</th>
<th>Mean Seek Time</th>
<th>Standard Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Control (n=9)</td>
<td>28.82</td>
<td>4.36</td>
</tr>
<tr>
<td>Joystick Control (n=9)</td>
<td>44.67</td>
<td>7.42</td>
</tr>
<tr>
<td>Head Tracking Control (n = 9)</td>
<td>37.13</td>
<td>3.19</td>
</tr>
</tbody>
</table>

#### 5.2.3 Collision Interval

Initially all tests fail to satisfy the ANOVA assumptions, however when the data was transformed logarithmically it met the requirements.

The test was performed, however none of the conditions resulted in significant impact upon Collision Interval as shown in Table 4 below and thus no further analysis was merited.

### Table 4 - ANOVA Analysis of Collision Interval

<table>
<thead>
<tr>
<th>Condition</th>
<th>Degrees of Freedom</th>
<th>Sum of Squares</th>
<th>Mean Square</th>
<th>F Value</th>
<th>Pr(&gt;F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camera Control</td>
<td>2</td>
<td>0.07391</td>
<td>0.03696</td>
<td>0.2665</td>
<td>0.7698</td>
</tr>
<tr>
<td>Lighting</td>
<td>1</td>
<td>0.06258</td>
<td>0.06258</td>
<td>0.4513</td>
<td>0.5126</td>
</tr>
<tr>
<td>Subject</td>
<td>7</td>
<td>1.52866</td>
<td>0.21838</td>
<td>1.575</td>
<td>0.2221</td>
</tr>
</tbody>
</table>
5.3 Round 2

5.3.1 Removal of Outliers

Following the first step of our data analysis methodology described above, any problematic outliers were removed from both data sets (Seek Time and Collision Interval).

In Figure 22 it is clearly evident that only one such outlier exists. It was determined that this outlier was indeed problematic as per definition and as such only the subject to whom this data point belonged was excluded from consideration.

From Figure 24 one can see that there are four outliers, however only one of them was determined to be problematic, as per definition, and as such only the subject to whom this data points belonged was excluded from consideration.

Thus in total two subjects were excluded in order to maintain experimental validity and as such all further analysis in this section was conducted on the original data excluding the outliers found above.

The subject, who was excluded due to poor Seek Time performance, was also excluded based on the first measurement of Seek Time and may have taken longer than average to adapt to the interface.
The subject, who was excluded for significantly better Collision Interval performance, was excluded based on the last measurement of Collision Interval. This have may have been biased due to the learning effect, where the subjects rapidly adapted to the interface and tasks presented by the different conditions and showed significant improvement after each test.

5.3.2 Seek Time
Although the Seek Time data failed the Shapiro-Wilkes test for normality, the values for Skewness and Kurtosis were valid and the ANOVA technique could be performed.

However, none of the conditions impacted significantly on Seek Time as can be seen in Table 5, and as such, no further analysis is merited.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Degrees of Freedom</th>
<th>Sum of Squares</th>
<th>Mean Square</th>
<th>F Value</th>
<th>Pr(&gt;F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HUD</td>
<td>1</td>
<td>13.300</td>
<td>13.300</td>
<td>0.11730</td>
<td>0.73420</td>
</tr>
<tr>
<td>Camera</td>
<td>1</td>
<td>0.033</td>
<td>0.033</td>
<td>0.00030</td>
<td>0.98650</td>
</tr>
<tr>
<td>Robot</td>
<td>1</td>
<td>106.000</td>
<td>106.000</td>
<td>0.93310</td>
<td>0.34130</td>
</tr>
<tr>
<td>Subject</td>
<td>19</td>
<td>3712.000</td>
<td>195.400</td>
<td>1.72030</td>
<td>0.08550</td>
</tr>
</tbody>
</table>

5.3.3 Collision Interval
The Collision Interval data failed both the Shapiro-Wilkes normality test and the tests for Skewness and Kurtosis. However, when the data was transformed logarithmically it succeeded in meeting the requirements.

As one can see from the table below, only the HUD Visibility (presence) condition impacted significantly (within the 95% confidence interval) upon the Collision Interval.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Degrees of Freedom</th>
<th>Sum of Squares</th>
<th>Mean Square</th>
<th>F Value</th>
<th>Pr(&gt;F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HUD Visibility</td>
<td>1</td>
<td>1.390100</td>
<td>1.390100</td>
<td>6.97850</td>
<td>0.01266</td>
</tr>
<tr>
<td>Camera Control</td>
<td>1</td>
<td>0.000004</td>
<td>0.000004</td>
<td>0.00002</td>
<td>0.99644</td>
</tr>
<tr>
<td>Robot Visibility</td>
<td>1</td>
<td>0.033400</td>
<td>0.033400</td>
<td>0.16750</td>
<td>0.68510</td>
</tr>
<tr>
<td>Subject</td>
<td>19</td>
<td>10.362600</td>
<td>0.545400</td>
<td>2.73800</td>
<td>0.00579</td>
</tr>
</tbody>
</table>

Although this is not surprising, what was unexpected was that the Robot Visibility condition had no significant impact and this certainly merits further discussion (see section 6.2).

Since HUD visibility was significant, we conducted further analysis to determine how the two different levels of this condition affected subject performance.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Mean Collision Interval</th>
<th>Standard Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>No HUD (n=14)</td>
<td>21.82</td>
<td>3.50</td>
</tr>
<tr>
<td>HUD (n=42)</td>
<td>31.23</td>
<td>3.07</td>
</tr>
</tbody>
</table>
From Table 7 above we plotted the Mean Collision Intervals of the two conditions against each other as well as error bars to indicate standard error as shown in Figure 25 below.

As one can see from Figure 25 above, subjects performed best when the HUD was visible and worst when it was not, although this is not surprising as one would expect the laser scanner to aid subjects by allowing them to accurately measure the distance between the robot and objects in the environment.
6 Discussion

6.1 Round 1

We found lighting conditions have significant impact on subjects’ performance and thus reject the null hypothesis. This is not surprising as one expects that poor lighting will make it more difficult to locate objects in the environment.

What is interesting is that the lighting conditions do not significantly impact on collisions. This may result from various factors; however, we believe the two most likely factors are:

1. The dim lighting condition mainly impacts on subjects’ ability to see objects at a distance, thus navigating around the environment, may not result in significantly more collisions, as objects close to the robot are visible and can be avoided.
2. Subjects used the laser scanner, which is not affected by poor lighting conditions, to measure the distance between the robot and objects in its environments, and thus avoid collisions. This is the most likely factor, as in the second round of experiments we show that the presence of the HUD has significant impact on the Collision Interval. The presence of the HUD resulted in a higher Collision Interval, which implies a reduced number of collisions.

We found that the choice of camera control technique has significant impact on subjects’ performance when locating pool balls but does not have significantly impact upon the number of collisions which occur. This was expected, however, what is unexpected, is that no camera control results in the best performance, with head tracked control coming second and manual camera control performing worst.

In order to explain these strange results we re-examined the camera and drive logs. We look specifically at camera and drive patterns and found the following:

1. Subjects with camera control spent on average more than twice as much time stationary than subjects with no camera control.
2. Subjects using head tracked camera control spent more than twice as much time with the camera disjoint (differing from current robot orientation by more than 10 degrees) than subjects with manual control.

This implies that subjects with camera control spent more time stationary when examining the environment than subjects with no camera control. This negatively impacts on their ability to locate the pool balls, which were distributed throughout the environment. In a situation where individual balls were placed in less visible and accessible locations the results may have been different. Due to time pressure subjects were required to move throughout the environment rapidly in order to locate all the balls and thus time spent stationary negatively impacted on their performance.

What is interesting is the fact that subjects with head tracked control made greater use of the camera than subjects with manual control, however we believe this can be attributed to the fact that subjects with manual control could not move and control the camera simultaneously,
whereas subjects with head tracked control could. This may have slightly biased the experiments, however due to time pressure, could not be explored in greater detail.

**6.2 Round 2**

We found that the impact of making the front of the robot’s chassis visible to the operator did not significantly impact on subjects’ performance. This is not surprising as the visibility of the chassis should have no effect on seek time. What is surprising is that it also has no impact on accuracy as we initially believed that chassis visibility would aid in increasing subjects’ situational awareness.

These results were also contrary to what subjects in our first round of experiments suggested, as well as, our expect users in the Robotics Lab. We believe the most likely cause for this result is that subjects gained enough situational awareness, through the use of the laser scanner, which aided in measuring distances between the robot and objects in its environment. Thus, the addition of the chassis visibility provided no significant advantage to subjects who were able to make use of the laser scanner.

Secondly, we investigate if the presence of the HUD has significant impact on user performance and accuracy. We find that there is no significant impact on performance, as was expected, since the HUD contained no elements to highlight or aid in the search of the balls. Examples of such elements might include a map of areas visited, image processing to identify possible pool balls and other useful tools.

We did however find that the presence of the HUD results in significantly fewer collisions. This we believe can be directly attributed to the laser scanner which allows subjects to judge the distance between the robot and objects in its environment with great accuracy. However, we cannot conclusively state this, as the HUD also consisted of another element showing the orientation of the camera. We do believe that the impact of such an element can safely be discounted, as we cannot determine how it may have aided subjects to avoid collisions.
7 Conclusions

7.1 Conclusions
We successfully designed and implemented a system, which allowed the Nintendo Wii controllers to be used for teleoperation of rescue robots, in a simulated environment. These controllers were chosen due to their reasonable cost, high availability and abundance of features.

Although these controllers proved to be a good choice they have several limitations, the most problematic of these is the fact that they lack a gyroscope. This makes it difficult track motion in the horizontal plane. However, Nintendo partially solved this problem by using a combination of infrared beacons and cameras to calculate the angle of rotation. This is not an ideal solution, as motion could only be tracked accurately to about 30 degrees in either direction, due to the limited field of view of the infrared camera.

This directly impacts on our ability to perform head tracking as it artificially constrains the amount users are able to rotate their heads and may result in cognitive dissonance.

Although the system was largely successful, there are several lessons to be learned from its development process.

Firstly, the decision to use a Head Mounted Display (HMD) should not be taken lightly, as it has significant impact on the design of the user interface:

1. Text and graphics which appear legible on a monitor even at the same resolution may not be legible on the HMD.
2. Text and graphics displayed at the top, bottom, left or right appear less legible on the HMD than on a monitor at the same resolution. Thus, careful consideration should be taken as to where to place interface elements and the extremes should be avoided if possible.
3. The impact of different brightness levels on the HMD was shown, by our experimentation, to have significant impact on subject performance and an appropriate brightness setting should be chosen carefully.

Secondly, the use of an iterated user evaluation of the system is an invaluable tool, and often enables the problems mentioned above, to be found before they impacted on the system design. Unfortunately, there was not enough time between our first and second rounds of experimentation to carry out more than a simple pilot evaluation of the interface. Thus, we were caught off guard when some subjects found text near the bottom of the screen difficult to read.

Lastly, the use of open source software is invaluable when creating a complex system in a short period time; however one must be careful not to be overly optimistic. We found ourselves testing too many conditions at once, which made analysis of the results much more difficult as the effect each condition was difficult to separate.
However, we were still able to use the system for hypothesis testing, and were able to establish the following hypotheses (within, at least, a 90% confidence interval):

1. *The choice of lighting conditions impacts significantly on task performance or accuracy when teleoperating rescue robots.*
2. *The choice of camera control impacts significantly on task performance or accuracy when teleoperating rescue robots.*
3. *The presence of a HUD (heads up display) impacts significantly on task performance or accuracy when teleoperating rescue robots.*

Furthermore, we show how HUD elements have a greater impact than camera control techniques on subjects’ performance and accuracy. This is in conflict with research done by Hughes et al. [23], which showed the use of an independent, controllable camera increases overall functional presence, as witnessed by improved search performance. However, the search task used by Hughes et al. required a greater level of specificity when inspecting objects, and thus, may have had a greater impact on operator performance, as according to Bowman et al. [6, 7], the ability to independently control the camera is desirable when object inspection is required.

These results may be of great interest in the field of Human Robot Interaction (HRI) as they illustrate several possible techniques for improving operator performance and accuracy. This may result in lives be saved in the future, especially in time critically applications, such as searching for survivors and victims after a natural disaster where every second counts.

### 7.2 Future Work

There are several areas which warrant future investigation. In terms of camera control, it would be interesting to investigate if obstacles which are more difficult to locate would give an advantage to subjects with camera control.

Another interesting area to explore is what impact the restrictions placed on the camera pan and tilt had on subject’s performance. Unfortunately, these restrictions arose mainly due to limitations inherent in using the Wii System for head tracking, such as the limited field of view of the infrared camera. This could be addressed through the use of an alternate controller, with full six degrees tracking, such as the SIXAXIS Controller from Sony or the Wii Motion-Plus accessory which is due to be released next year.

Furthermore, no investigation was conducted on the impact of multiple independent cameras on subjects’ performance. As a subset of this condition, one could also investigate the impact of stereo vision, implemented using two cameras, as this should enhance the subject’s perception of depth.

In terms of the HUD interface, the only condition tested in this research was whether the HUD’s presence positively or negatively impacted on subjects’ performance. No research was conducted on which elements of the HUD were responsible for this impact.
There also exist several opportunities to test the impact of various levels of HUD integration with the virtual world. The current HUD design did not have a high level of world integration and could be improved by using techniques from Augmented Reality research. One possibility is a deeper integration of the laser scanner with the world as this would allow distances between objects to be shown overlaid on the actual objects.

Lastly, only indoor environments were used when testing subjects’ performance, and as such, this may have introduced bias.
8 References


9 Appendices

9.1 Appendix 1: Waiver

Experiment Title: The evaluation of the Nintendo Wiimote and Nunchuk for teleoperation in urban search and rescue environments.

Purpose of the research study: The purpose of this study is to determine the suitability of the Nintendo Wiimote and Nunchuk as control mechanism and means of head tracking for robotic control.

What you will be asked to do in this study: Volunteer participation in this research project will take place at the Experiment Room in the Computer Science Building. Following a brief informal briefing about the simulator, you will be given an opportunity for a 10 minute test drive of the robot within the simulator so as to become familiar with the controls and get acclimated to the virtual environment. After a short rest period, you will be asked to perform a number of tasks and after completing each task you will be given a short questionnaire.

Time Required: Approximately 50 minutes

Risks: There is a small risk of subjects developing what is ordinarily referred to as simulator sickness. It occurs infrequently to subjects who are exposed to prolonged continuous testing in simulated environments. Symptoms consist of nausea and a feeling of being light headed. The risk is minimized as a result of the short duration of each session in the simulator. Five-minute breaks will be given at intervals if needed. Potential side effects of virtual environment (VE) use include stomach discomfort, headaches, sleepiness, and mild degradation of postural stability. However, these risks are no greater than the sickness risks participants may be exposed to if they were to visit an amusement park such as Ratanga Junction or any such amusement park with attractions such as roller coasters.

Benefits/Compensation: There is no direct benefit to you from participation in this study. All volunteers will receive R20 for time and effort in completing this study.

Privacy: Your identity will be kept confidential. Your name will not be used in any report.

Voluntary participation: Your participation in this study is voluntary. You have the right to withdraw from this study at any time without consequence.
More information: For more information or if you have questions about this study, contact

Jason Brownbridge      Graeme Smith
jbrownbridge@gmail.com   graeme0811@gmail.com

☐ I have read the procedure described above
☐ I voluntarily agree to participate in the procedure
☐ I am at least 18 years of age or older

Participant ___________________________ Date __________________
9.2 Appendix 2: Questionnaires

9.2.1 Pre-experiment Questionnaire

Subject Number: ______________________________________________________

Age  ________________________________________________________________

Gender  Male  /  Female

Previous Robotic Control Experience:  Yes  /  No

Previous Computing Experience:  Yes  /  No

Corrected Vision:  Yes  /  No

Colour Blind:  Yes  /  No

For the following questions, please circle the number which best represents your experience.

1. On average how much time do you spend using a computer per week?

   1   2   3   4   5   6   7

   No time  Almost all my time

2. On average how much time do you spend gaming, per week?

   1   2   3   4   5   6   7

   No time  Almost all my time

3. What level of previous virtual reality experience have you had?

   None  2  3  4  5  6  7

   Used a head mounted display  Full virtual reality experience
9.2.2 Post-experiment Questionnaire

Experiment Number: ________________________________

Subject Number: __________________________________

For the following questions, please circle the number which best represents your experience.

1. Please rate your sense of being in the virtual environment, on a scale of 1 to 7, where 7 represents your normal experience of being in a place.

   I had a sense of “being there” in the virtual environment:

   1 2 3 4 5 6 7
   Not at all Very much

2. To what extent were there times during the experience when the virtual environment was the reality for you?

   There were times during the experience when the virtual environment was the reality for me...

   1 2 3 4 5 6 7
   At no time Almost all the time

3. When you think back to the experience, do you think of the virtual environment more as images that you saw or more as somewhere that you visited?

   The virtual environment seems to me to be more like...

   1 2 3 4 5 6 7
   Images that I saw Somewhere that I visited

4. How difficult was it to control the robot under these conditions?

   1 2 3 4 5 6 7
   Easy Impossible

5. During the time of the experience, which was the strongest on the whole, your sense of being in the virtual environment or of being elsewhere?

   I had a stronger sense of...

   1 2 3 4 5 6 7
   Being elsewhere Being in the virtual environment

6. Consider your memory of being in the virtual environment. How similar in terms of the structure of the memory is this to the structure of the memory of other places you have been today? By ‘structure of
the memory’ consider things like the extent to which you have a visual memory of the virtual environment, whether that memory is in colour, the extent to which the memory seems vivid or realistic, its size, location in your imagination, the extent to which it is panoramic in your imagination, and other such structural elements.

*I think of the virtual environment as a place in a way similar to other places that I’ve been today…*

1 2 3 4 5 6 7
Not at all Very much so

7. During the time of your experience, did you often think to yourself that you were actually in the virtual environment?

*During the experience I often thought that I was really standing in the virtual environment…*

1 2 3 4 5 6 7
Not very often Very much so

Any Comments:

_______________________________________________________________
_______________________________________________________________
_______________________________________________________________
_______________________________________________________________
_______________________________________________________________
_______________________________________________________________
_______________________________________________________________
_______________________________________________________________

49
Appendix 3: Sandbox Instructions

Intro
This is the Sandbox, where you get to learn how to drive the robot. Hold the Nunchuk in whichever hand feels more comfortable. Hold the Wiimote in the other hand. To drive the robot, press the joystick on top of the Nunchuk in the direction you wish to go. You can turn the camera independently of the robot by holding down the “Z” button on the Nunchuk and using the joystick to move the camera. Use the laser scanner/pan display and tilt display to see where the camera is pointing in relation to the robot. To re-align the camera with the robot, press the “C” button on the Nunchuk.

The robot can tip over if you drive into walls. Avoid doing this as a real robot would be severely damaged by collisions or tipping over. If you do tip over, press the “Home” button on the Wiimote to right yourself. The Wiimote will vibrate when you collide with something.

Observe the laser scanner, which allows you to judge, in more complex environments, where obstacles are and assists you in steering around them. This works by sending out 180 beams in an arc around the front of the robot and recording how far they travel before they hit something. So if you see red on the laser display, this means that there is nothing in front of you.

Overlaid on the laser-scanner display is the pan display. This shows the direction the camera is pointing in relation to the robot.

The tilt display, to the left of this, will show, when the camera is mobile, the tilt of the camera.

The GPS system displays your current coordinates as well as the coordinates of your next waypoint. Travel towards the waypoints. When you get close enough, the system will log that you have located the waypoint and a sound will play. A new waypoint will display.

Aim
Drive around the track.

You will do two types of experiments today. This sandbox aims to teach the basics of both. The first type requires you to find pool balls hidden around a map.

When you see a pool ball, say (aloud) “Located Ball <x>“ or “Located <color> ball”. This will allow us to log your finding of the balls.

The second type of experiment requires you to locate waypoints using the GPS system. Use the GPS to drive to each waypoint as it is shown. When you reach the waypoint a chime will play. Move to the next waypoint.

You have 5 minutes to familiarize yourself with the robot, have fun.
The pool balls 1 through 8 will be hidden in each map.
The blue cone shows the camera's view of the robot. The robot's current location is indicated by the center of the blue cone. When the robot's path is interrupted by an obstacle, the cone will tilt and the direction of the tilt will indicate the location of the obstacle. The cone's tilt also indicates the direction the robot is facing. The green bar at the top of the cone represents the robot's heading, and the red bar at the bottom represents the robot's orientation. The cone's tilt is relative to the robot's current direction.

The GPS system displays the robot's coordinates relative to a known point. The coordinates are shown on the screen in the form of x, y, z. The values are measured in meters, and the units are specified on the screen. The coordinates are updated in real-time as the robot moves, allowing you to track its location accurately.

The laser scanner is very useful for detecting obstacles. The scanner emits a laser beam that sweeps across the environment, and the sensor measures the time it takes for the beam to reflect back. The reflected beam is detected by the scanner, and the distance to the obstacle is calculated based on the time of flight. This information is used to generate a 3D map of the environment, which is displayed on the screen. The map shows the obstacles detected by the laser scanner, and the robot's position relative to those obstacles.

To use the laser scanner, you can select the laser scanner option from the menu. This will activate the laser scanner and display the 3D map on the screen. You can then use the map to plan your path and avoid obstacles. The laser scanner is particularly useful in environments with complex obstacles, such as narrow passages or tight corners. It can help you navigate safely and efficiently, even in challenging conditions.
9.4 Appendix 4: Round 1 Participant Instructions

9.4.1 No Camera Control

Intro
Driving the robot is the same as in the Sandbox but now you cannot turn the camera independently of the robot.

Aim
The aim of this exercise is to find as many pools as you can in 5 minutes. As before, you must speak aloud the pool balls number or colour when you locate it.

As before, avoid collisions or tipping over whenever possible.

9.4.2 Manual Camera Control

Intro
Driving the robot is the same as in the Sandbox i.e. you can turn the camera independently of the robot by holding down the “Z” button on the Nunchuk and using the joystick to move the camera.

Use the laser scanner/pan display and tilt display to see where the camera is pointing in relation to the robot.

To re-align the camera with the robot, press the “C” button on the Nunchuk.

Aim
The aim of this exercise is to find as many pool balls as you can in 5 minutes. As before, you must speak aloud the pool balls number or colour when you locate it.

As before, avoid collisions or tipping over whenever possible.

9.4.3 Head Tracked Camera Control

Intro
Driving the robot is the same as in the Sandbox but now you can turn the camera independently of the robot by turning your head in the direction you wish to look.

Use the laser scanner/pan display and tilt display to see where the camera is pointing in relation to the robot.

As the infrared sensor does not have a big range, do not turn your head too far to either side. When you near the range of the sensor, the Wiimote on your head will vibrate and a red light will show on the side of the display. Stop turning at this point.

Aim
The aim of this exercise is to find as many pools as you can in 5 minutes. As before, you must speak aloud the pool balls number or colour when you locate it.

As before, avoid collisions or tipping over whenever possible.
9.5 Appendix 5: Round 2 Participant Instructions

9.5.1 No Camera Control

Intro
Driving the robot is the same as in the Sandbox but now you cannot turn the camera independently of the robot.

Aim
The aim of this exercise is to find as many pool balls as you can in 5 minutes. As before, you must speak aloud the pool balls number or colour when you locate it.

As before, avoid collisions or tipping over whenever possible.

9.5.2 Manual Camera Control

Intro
Driving the robot is the same as in the Sandbox i.e. you can turn the camera independently of the robot by holding down the “Z” button on the Nunchuk and using the joystick to move the camera.

Use the laser scanner/pan display and tilt display to see where the camera is pointing in relation to the robot.

To re-align the camera with the robot, press the “C” button on the Nunchuk.

Aim
The aim of this exercise is to find as many pool balls as you can in 5 minutes. As before, you must speak aloud the pool balls number or colour when you locate it.

As before, avoid collisions or tipping over whenever possible.