2009

Virtual reality based multi-modal teleoperation using mixed autonomy

Muthukkumar Kadavasal Sivaraman
Iowa State University

Virtual reality based multi-modal teleoperation using mixed autonomy

by

Muthukkumar Kadavasal Sivaraman

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Human Computer Interaction

Program of Study Committee:
James H. Oliver, Major Professor
James E. Bernard
Stephen Gilbert
Greg R. Luecke
Namrata Vaswani

Iowa State University

Ames, Iowa

2009

“dedicated to

my dad”

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

This dissertation of mine would not have been possible if not for the help provided by quite a number of people, a few of which I would like to list by name here. First and foremost, I would like to express my sincere thanks to my advisor Prof. James Oliver for his excellent guidance through out my PhD research. I sincerely appreciate his patience during my research and the enthusiasm he provided through out this process. I would like to thank my dad Mr. Sivaraman R Kadavasal without whom this day of writing my acknowledgements for my PhD dissertation would not have come. At times when I questioned myself on my ability to continue higher education (MS and PhD), he was the one who reinstated the faith in me to pursue it and as well as complete it. Though, he is no more, I would still like to express my heartiest gratitude for the wealth of guidance that he provided me through out my life so far. I would like to thank my committee member Prof. James Bernard for his assistance in overall research and specifically in vehicle dynamics and in writing this thesis. I would also like to thank another committee member Dr. Stephen Gilbert for his guidance in my research work and specifically in conducting and analyzing the user studies for this research work. I thank Dr. Namrata Vaswani, my committee member for the guidance she provided in my overall research work and specifically in helping me understanding image processing and camera models that I implemented in this thesis. I would like to as well thank my committee member Prof. Greg Luecke for his guidance in my research and specifically for those thought provoking questions through out the research which helped me greatly in understanding my research problem better. I would like to thank United States Air Force Research Laboratory, Grant Number FA9550-05-1-0384 for supporting my research work. I would like to thank Dr. Bryan Walter, Dr. Jared Knutzon, and Mr. Tom Batkiewicz for their contributions and assistance in creating the initial platform for my research work. I would like to acknowledge the Virtual Reality Applications Center (VRAC) for the research infrastructure provided in completing this research work. I would like to thank Mr. Glen Galvin, system support specialist of VRAC and my colleague, Dr. Abhishek Seth for their assistance in setting up the trackers and as well other systems. I would also like to thank artist Ms. Kira Layli Scott for modeling the task environment. My sincere thanks to all VRAC staff members including Ms. Karen Koppenhaver, Ms. Lynette Sherer, Ms. Beth Hageman, Ms. Pam Shill, Mr. Kevin Teske and Ms. Jean Bessman for their assistance and support in various administrative works related to my PhD. I am thankful to my family especially my mom and sister for their motivation and as well as for the patience that they had through out my 11 year long graduate and undergraduate education. My sincere thanks to all my friends for the appreciations that they provided on my successes, for creating the fun filled environment that made my life a lot easier and as well as for their motivation and support during those time of crisis. On a lighter vein, I would also like thank the directors, producers, writers and authors of Lord of the rings – trilogy, Harry Potter series, Matrix – trilogy, Planet of the Earth series, and as well as of the sitcoms including Friends, ER and House. Also, I thank all the politicians and the media for feeding my brain with information that helped me and my friends talk, discuss, argue, and complain about in those endless coffee shop conversations without which I am sure my graduate life would have been a bit dull.

ABSTRACT

The thesis presents a multi modal teleoperation interface featuring an integrated virtual reality based simulation augmented by sensors and image processing capabilities on-board the remotely operated vehicle. The virtual reality interface fuses an existing VR model with live video feed and prediction states, thereby creating a multi modal control interface. Virtual reality addresses the typical limitations of video-based teleoperation caused by signal lag and limited field of view thereby allowing the operator to navigate in a continuous fashion. The vehicle incorporates an on-board computer and a stereo vision system to facilitate obstacle detection. A vehicle adaptation system with a priori risk maps and real state tracking system enables temporary autonomous operation of the vehicle for local navigation around obstacles and automatic re-establishment of the vehicle's teleoperated state. As both the vehicle and the operator share absolute autonomy in stages, the operation is referred to as mixed autonomous. Finally, the system provides real time update of the virtual environment based on anomalies encountered by the vehicle. The system effectively balances the autonomy between the human operator and on board vehicle intelligence. The reliability results of individual components along with overall system implementation and the results of the user study helps show that the VR based multi modal teleoperation interface is more adaptable and intuitive when compared to other interfaces.

CHAPTER 1. INTRODUCTION

Teleoperation can be broadly defined as controlling a system from a distance. One of the primary motivations behind teleoperation research is the need to perform tasks in places that are unsuitable for human presence. For example, using unmanned aerial vehicles (UAV) for reconnaissance in hostile regions [Knutzon 2004], researching the ocean floor without risking a diver's life [Lin 1999], and exploring a damaged nuclear reactor using a teleoperated ground vehicle, are a few examples in which teleoperation can play a vital role. These scenarios are comprised of complex tasks in dynamic environments that require spontaneous and critical decision making which cannot be carried out by autonomous agents, making human participation important [Knutzon 2003]. Vehicle teleoperation is a specific type of teleoperation that involves controlling remote vehicles or robots. In this dissertation, the terms vehicle teleoperation and teleoperation will be used interchangeably. Vehicle teleoperation is typically classified based on the type of vehicle used namely air, underwater, space and ground vehicles. It may also be classified based on the type of interface and control used by the operator, for example: direct / video based, mixed modal, supervisory, novel interfaces etc. The following section discusses these classifications in detail.

Teleoperation classification – Vehicle type

Air vehicles

Drones or Remotely Piloted Vehicles (RPV) are teleoperated air vehicles that were used for reconnaissance and anti aircraft training in early 20th century. However, Remotely Piloted Research Vehicles (RPRVs) developed in 1960s are full size aircraft modified for unmanned flight missions [Hallion 1984]. In recent times, the term Unmanned Aerial Vehicles (UAVs) is the most common name used for teleoperated air vehicles. UAVs are predominantly used for target identification and reconnaissance and sometimes in combat. Some of the operational tasks involved in controlling a UAV are flight control, camera orientation and sensor positioning relative to the vehicle. When performing a military reconnaissance or combat mission, a UAV typically moves back and forth between structured and unstructured environments [Spenny 1999]. This not only requires mission training prior to flight initiation but also a human in the loop to perform tasks for which he is more effective than an autonomous controller. Figure 1 shows the Predator UAV and its ground control station.

Underwater vehicles

Vehicles teleoperated under water are typically called Remotely Operated Underwater Vehicles or ROVs. ROVs are tethered underwater robots, typically operated manually on board a shipping vessel or platform. They are predominantly used for exploration, rescue and recovery operations. One of the earliest successful underwater

Figure 1. The Predator UAV and Predator Ground Station

Courtesy: http://www.airforce-technology.com/ ; http://spyflight.co.uk/Predator.htm teleoperation missions was carried out in 1966 by the US Navy to recover an atomic bomb off Spain. The vehicle also known as the Cable-controlled Undersea Recovery Vehicle (CURV) [Wernli 1982] used in this operation was developed by the Naval Ordnance Test Station (NOTS). Figure 2 shows the CURV and its operation under water. ROVs are heavily used by offshore oil industries for deep-sea inspection, maintenance and repair tasks [Qingping 1999]. The scenario involves navigation of complex unstructured sea bed along with complexity of offshore installations and murky waters necessitating the presence of a human operator to navigate and control the vehicle.

Ground vehicles

Teleoperated Unmanned Ground Vehicles also know as UGVs are primarily used in hazardous situations that range from search and rescue operations to reconnaissance and combat. Fitted with sensors such as cameras for visual feedback, infrared cameras for detecting heat, sound and motion [Nourbakhsh 2005], to laser range finders for 3D

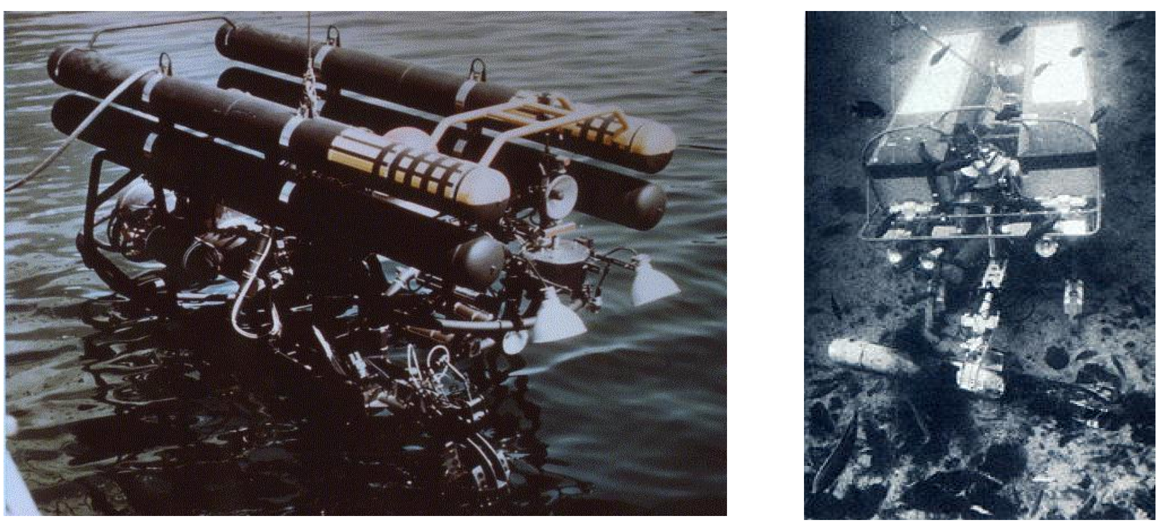

Figure 2. Underwater Vehicle CURV and Vehicle in Operation

Courtesy: http://www.spawar.navy.mil Courtesy: http://www.divingheritage.com/

reconstruction, UGVs can effectively replace a human for gathering data. However, many situations involve decision making which warrants a human in the loop. The Numbat UGV [Hainsworth 2001] shown in Figure 3, developed by CSIRO for exploration and mining is used to provide real time visual surveillance and atmospheric analysis of underground coal mines in situations that are too hazardous for manual exploration. The CRASAR robots (Center for Robot Assisted Search and Rescue) used in search and rescue operations during the World Trade Center attack is a realistic example of how teleoperated ground robots can play a vital role in saving human lives [Eagar 2001]. The Man Portable Robotic System (MPRS) developed by Spawar systems [Laird 2000], is a tunnel surveillance system that can be deployed in combat situations. The system informally named 'tunnel rat', is fitted with sensors that include light weight camera and sensors as well as heavy body armor that gives a decisive advantage for the vehicle in threat zones.

Space Vehicles

The most well known teleoperated space vehicle is the Mars rover also known as

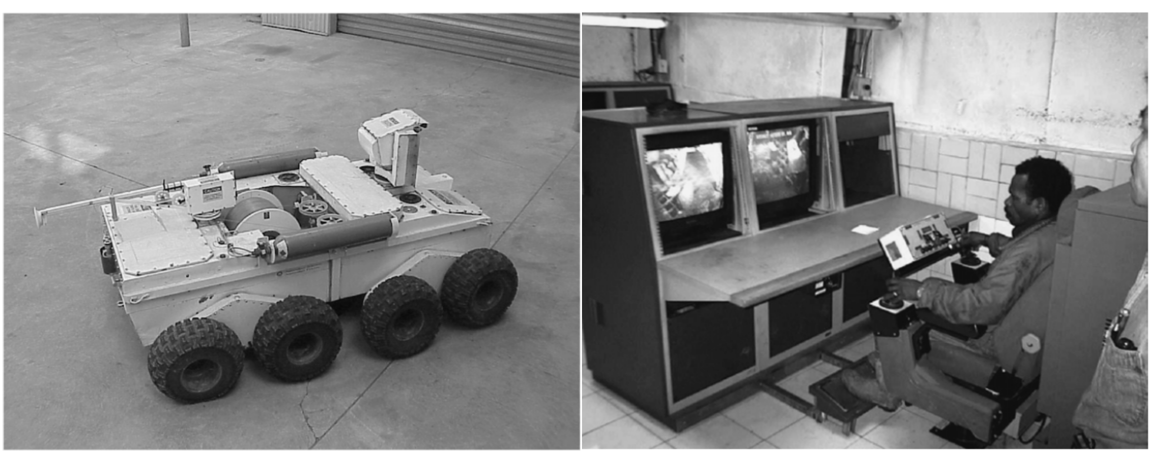

Figure 3. The Numbat UGV and Teleoperator Station [Hainsworth 2001]

Sojourner [Sojourner 2007], which was launched in space in 1996 as a part of the Mars Pathfinder mission by National Aeronautics and Space Administration (NASA). The Sojourner was fitted with devices to determine Martian terrain and atmospheric composition. As operator commands travel between the teleoperator station and the vehicle take a considerable amount of time, the vehicle incorporates an onboard intelligence mechanism that provides autonomy for low level decision making [Matijevic 1998]. However, the high level decision making and commands are provided by the human operator. In 2003, NASA sent two more rovers, namely Spirit and Opportunity to explore Mars surface and geology. Space vehicle teleoperation involves complex control mechanism mainly due to the huge lag in transmission of commands and sensor feedback. This makes it different from other teleoperation types.

Teleoperation classification – Interface and control type

All the components utilized by the teleoperator for efficiently navigating or operating the remote vehicle can be collectively termed the teleoperation interface. This includes the vehicle control systems like radio control joysticks or steering wheels, tracking system, driving simulators, haptic devices and visual feedback systems such as camera, sensor data, video/graphic display units, etc. The vehicle teleoperation interface provides tools for the human operator to effectively perceive the vehicle environment, react to situations, make decisions and issue commands. The interface should be intuitive and user adaptive, and should be able to maximize information transfer with minimum cognitive and sensory-motor work load [Fong 2001].

Teleoperation interfaces are typically classified into direct control (direct visual feedback or indirect video feedback), multi modal/multi sensor control, supervisory control and novel interfaces. Direct control is the standard architecture used in basic teleoperation and the other control interfaces mentioned above are developed either to compensate for the problems faced in direct teleoperation or for custom scenarios. The following paragraphs discuss the basic direct teleoperation architecture and its limitations.

Direct teleoperation interfaces

Direct teleoperation of a vehicle typically involves direct visual feedback (such as the hobbyists RC airplane), or indirect visual feedback from onboard video cameras or laser scanners. Figure 4 presents a schematic architecture for a video-based vehicle teleoperation system. The system is comprised of a remote vehicle, an onboard camera system, a video display unit and a vehicle control unit. The driver/teleoperator sends commands to the vehicle control unit which in turn sends the commands to remote vehicle. The camera fitted on the vehicle sends video images back to the operator station, which is then received and displayed by the video display unit. The commands and video are transmitted using wireless network communication. The operator sees the video feed and decides on the next set of

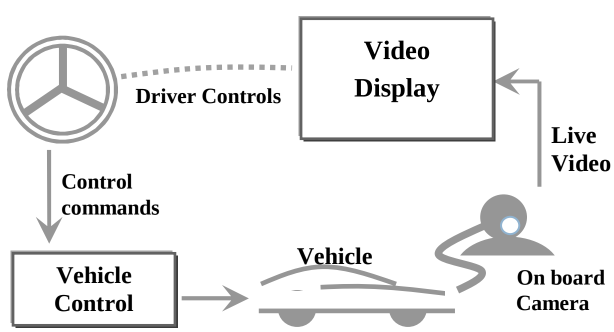

Figure 4. Architecture for Video Based Teleoperation

vehicle control commands. The operator drives as if he is inside the vehicle looking out. The phenomenon is referred to as inside/out driving [Fong 2001]. Most of the UGVs described earlier are teleoperated using direct control interfaces. Figure 3b shows the teleoperator interface for controlling the Numbat remote vehicle. The video feed obtained from the vehicle is displayed on monitors for the operator to interpret.

Limitations and Challenges

Lag

Direct control systems are subject to time lag in the transfer of the video feed as well as the vehicle commands. The commands and video transmission are affected by the amount of bandwidth available as well as the distance from the control station and the vehicle, so significant delay in communication is common. Assuming that the system has a constant lag of *t* seconds, then every command sent to the vehicle and every frame of video sent to the operator station are received *t* seconds after the actual event. Consider driving a car in busy traffic and with a delay of *t* seconds in driver's inputs. Suppose a vehicle from the nearby lane swerves into your path, your natural reflex would immediately dictate you to apply brakes and slow down to avoid collision. However, with lagged controls, you may end up colliding with the vehicle in front, resulting in a disaster [Walter 2003]. In this scenario, there was no lag on the driver's end in perceiving the surrounding world. Imagine instead that you driving the same car using a video feed from the vehicle with a *t* second lag in video transmission. In this case, your car may collide with the vehicle front even before you become aware of the vehicle's lane change. The second scenario is the realistic depiction of camera-based teleoperation. Moreover, in real life situations, the lag time is random.

Direct control interfaces coupled with such a delay can be tedious, fatiguing and error prone. The technical report by NASA on remote manipulation with transmission delay [Ferrell 1965] shows that as time delay increases the completion time of the task increases as well. The researchers found that operators cannot carry out such remote tasks in a continuous command mode but rather should switch for “move and wait” strategy. Such a strategy will work for open loop control systems, where the operator closes the loop by making adjustments to keep the system within operating parameters. However, modern teleoperation tasks are complex and time dependent and cannot be implemented as open loop systems.

Field of view

In order to effectively teleoperate a remote vehicle, the operator has to have thorough knowledge on the driving environment. However, the video feeds from onboard cameras are typically the primary source of knowledge about the driving environment. The video image is restricted by the field of view (FOV) of the camera’s optical system. A normal human eye provides 200 degrees horizontal FOV and 135 degrees vertical FOV, but even the highest quality cameras can only provide a maximum 120 degree FOV. As humans are accustomed to such a wide FOV, the teleoperator can experience difficulty in adapting to the video images which provides less information to perceive and react to. In short, the video feed obtained from the camera provides a limited or “soda straw” view of the environment due to the camera’s limited FOV [Grant 2002]. It should be noted here that the problem cannot be solved by providing a camera with wider FOV. The research conducted by the Human Research and Engineering Directorate (HRED) of U.S. Army Research Laboratory on teleoperating UGVs shows that the operator’s telepresence is significantly affected by the camera focal length [Glumm 1992]. Lens focal length determines the FOV of the camera system. The shorter the focal length the larger the FOV, but with shorter focal length the peripheral distortion of the camera is higher. The objects in the periphery may appear farther than they actually are. The converse is true for longer focal lengths. Moreover, shorter focal length results in lower camera resolution.

In teleoperation, the human operator's performance is subjected to limitations imposed by his sensory, cognitive and motor capabilities. In other words, a human being cannot efficiently perceive large chunks of information randomly presented at different intervals. Hence, teleoperation can be successful if the operator experiences a compelling illusion of what researchers' terms as *telepresence* [Riley 2004]. Telepresence means that the operator receives sufficient information about the vehicle environment, displayed in a sufficiently natural way such that the operator feels physically present at the remote site [Sheridan 1992]. The operator perceives the information provided and creates a mental model of the task environment. This internal mental model conceived by the operator at any given point of time is referred to as situational awareness of the operator [Endsley 1988]. When the system is subjected to lag and lack of FOV, and when the vehicle environment is dynamic, then the operator input and vehicle reaction are not intuitively linked in time (the operator overdrives the vehicle) resulting in the operator's potential loss of situational awareness [Kim 1992].

Need for teleoperation and motivation

The most common argument against teleoperation is that, if teleoperation has so many limitations then why can't we design an autonomous system that can carry out the same task without the human operator. As mentioned earlier, teleoperation is considered necessary for those tasks in which human participation is important. A human in the loop can provide the ability to make well informed decisions in the absence of complete and correct information [Ruff 2002]. Humans also possess excellent retention skills for problem solving in dynamically changing operating conditions [Dunkler 1988]. Hence, the human operator cannot be eliminated from the loop. Thus, teleoperation is considered one of the most reliable methods for tasks that involve high level decision making that cannot be automated.

However, the typical teleoperation interface is ridden with massive limitations and problems. These challenges motivate researchers to develop an effective teleoperation interface that can satisfy the following basic requirements.

1. Lag in signal transmission cannot be eliminated. However, the discomfort caused to the human operator can be reduced. Hence, the teleoperation interface should be able to accommodate lag.
2. The interface should be able to present sufficient information on the task environment to the operator. The information presented to the operator is considered sufficient as long the operator does not experience his cognitive or motor skills being challenged.
3. The interface must be intuitive enough to represent the operator's mental model of the task environment. Such a mental model would improve the operator's situational awareness.
4. The interface should be intelligent enough to take care of low level tasks that do not require human intelligence and decision making. This eliminates operator overload.
5. The interface should be user adaptive meaning the training time should be minimal.

CHAPTER 2. LITERATURE REVIEW

The chapter presents the literature review on various teleoperation interfaces. The interfaces are classified into the following categories, multi sensor and multi modal interfaces, augmented reality based interfaces, virtual reality based interfaces, and other novel interfaces.

Multi modal / multi sensor interfaces

Lack of peripheral vision due to the limited FOV can be compensated by adding more cameras and sensors. However, this approach requires the operator to pay attention to several different video feeds simultaneously and create a consistent mental image of the world [Walter 2003]. This increases the operator's stress and distracts her/his focus away from the task at hand. Sugimoto et al [Sugimoto 2005] proposes a teleoperation interface that merges two images obtained from two different cameras, where one camera is mounted on the front of vehicle and the other camera is mounted on a boom behind the vehicle, pointed in the same direction. The position behind the vehicle where the second camera is mounted is called the exocentric position as shown in Figure 5. In the proposed model the images from the exocentric camera can be stored and time stamped. The past images from this camera can act as the current position's peripheral data, thereby compensating for the lack of FOV of the egocentric (front) camera. Such an interface can provide more information for the operator thereby improving the situational awareness. On the other hand, receiving two different camera feeds, processing two different images from two different time stamps and merging them to create the peripheral vision is a time consuming process and can cause more lag. The final video

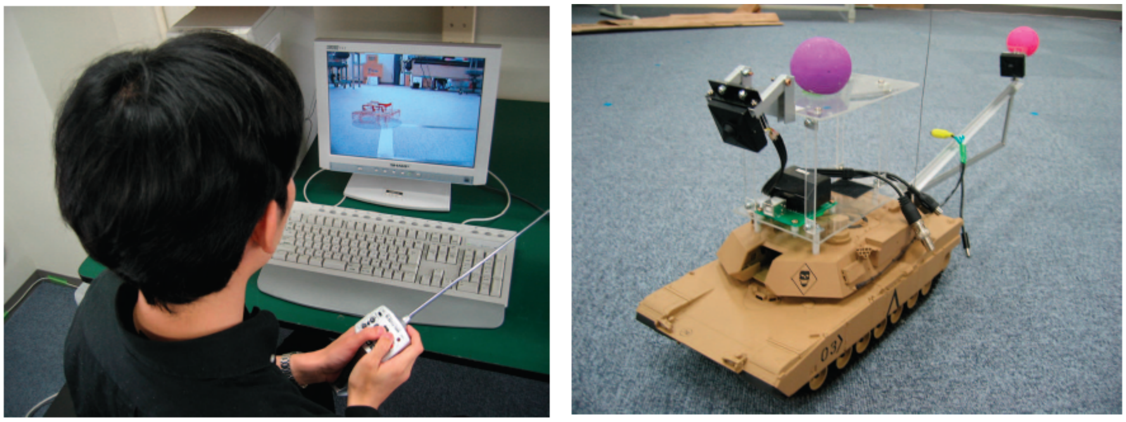

Figure 5. Operator using time follower's vision and hardware [Sugimoto 2005]

feed presents a scenario which is say *t* second delayed. So, the model can be effective for tasks that do not demand time dependent decisions.

The task driven camera based teleoperation presented by Hughes et al [Hughes 2005] proposes the idea of having multiple vehicle mounted cameras that can be controlled by the operator independent of the orientation of the vehicle. The paper shows that the proposed model is better than single fixed-camera teleoperation and can reduce the cognitive burden of the operator. Moreover, the model emphasizes the need for developing a teleoperation interface that does not encroach on the operator's control. Nevertheless, the system is affected by the lag in signal communication. Indeed the proposed system compounds the problems caused due to lag, as the operator has to issue commands to control the individual camera orientation and the commands are delayed.

Researchers have been working on providing integrated environment data by augmenting multiple sensors [Jarvis 1999]. Ricks et al [Ricks 2004] developed "ecological" displays which allow users to navigate in 3D worlds with integrated range and camera information. The video images presented in a direct video teleoperation are 2D and do not give significant information on the obstacle/scene depth. As human vision is accustomed to perceiving depth information, the camera images can cause discomfort and create mental fatigue. In their proposed model 3D range data are collected from the laser range finder and sonar fitted on the vehicle. The vehicle also possesses a camera for sending 2D images. The display interface shown in Figure 6 presents the most recent 2D image received from the camera and the most recent 3D rudimentary data from the laser range finder. Although, the system incorporates a prediction algorithm for identifying obstacle positions beforehand, the 2D and 3D image data are not synchronized. It is obvious that the operator will have difficulty following the time stamp of data shown from multiple sensors. And above all, the various data streams from multiple sensors and cameras may be subject to different and variable lag, so synchronizing them before they are presented to the user can be challenging.

Augmented reality based interfaces

Augmented reality (AR) and virtual reality (VR) technologies are enabling some of the most novel new teleoperation interfaces. VR teleoperation interfaces are discussed at the end of this chapter. Milgram et al [Milgram 1995] presented one of the earliest AR based telerobotic systems. The Augmented Reality through Graphical Overlays and Stereo video (ARGOS) took kit presented in their paper gathers quantitative spatial information from the task environment and develops a partial model of the remote 3D work site. The paper created taxonomy of level-of-autonomy in remote operations and identified the autonomy levels of AR based teleoperation.

The most common AR technique, sometimes referred to as synthetic imagery, involves overlaying and registering text and images (e.g., generated from sensor information

[Brujic-Okretic 2003]) onto a live video feed or computer generated scene. AR interfaces have been applied in assembly and maintenance processes, where instructions and reference lines can be superimposed over video or graphics representation of models. The AR interface developed by Fuchs *et al* [Fuchs 2002] uses a stereo vision based image processing algorithm to identify object position in the task environment. The object position data is then matched with the current video image and a synthetic sensor overlay is applied. The synthetic imagery is shown in Figure 6. The synthetic sensor data can assist the teleoperator in easy alignment and assembly. The experimental task considered in this paper does not demand time dependent decision making. Hence, with effective synchronizing of sensor data and the actual video image, this teleoperation interface can be effective for this particular task.

Synthetic imagery has played a major role in military research and simulations. Rapid Imaging Software Inc., in collaboration with the Human Effectiveness Directorate of U.S. Air Force Research Laboratory developed a UAV control station display that fuses synthetic vision data with a live video feed [Calhoun 2005]. The data is gathered from numerous resources and it includes terrain information, cultural features, pre mission plan, weather, etc. The system can help highlight important spatial elements over the terrain. Spatial elements

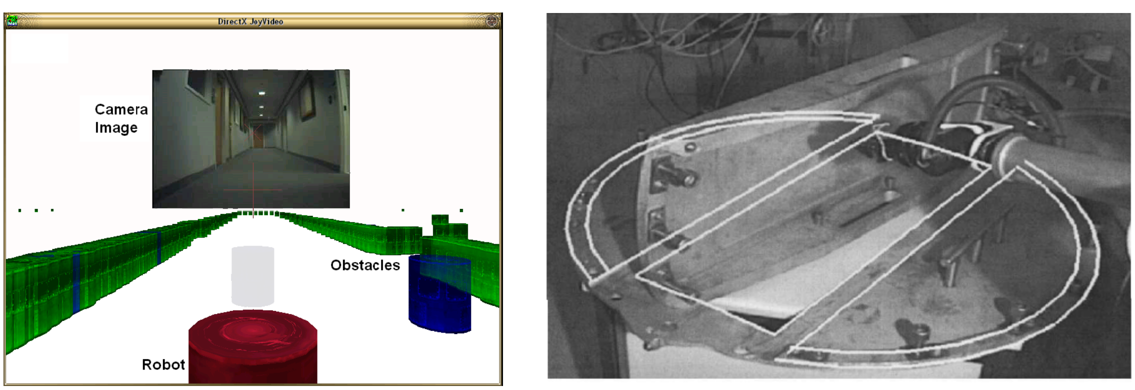

Figure 6. Ecological Display [Ricks 2004] and AR Synthetic overlay [Fuchs 2002]

include threat zones, landmarks, emergency landing sites, target locations, etc. The paper also discusses various visualization issues like data clutter, information blending and management, and data retrieval. The interface is intuitive and can improve the operator's situational awareness significantly. Figure 7 shows the synthetic imagery generated and the sensor operator work station.

AR interfaces have also been effective for teleoperation of vehicles for tunnel inspections [Lawson 2002]. This interface also uses stereoscopic sensor data, which are registered and overlaid onto video images to help operators identify cracks and holes. Unlike the maintenance assembly system explained earlier, the teleoperated sewer maintenance system will be severely affected by lag in sensor and video data. The model tries to accommodate lag by command based control approach. Here, the operator issues the next set of commands only after receiving feedback from the task environment for his previous set of commands. The time taken for completing the operation in such an approach could be enormous and is not suitable for life threatening situations.

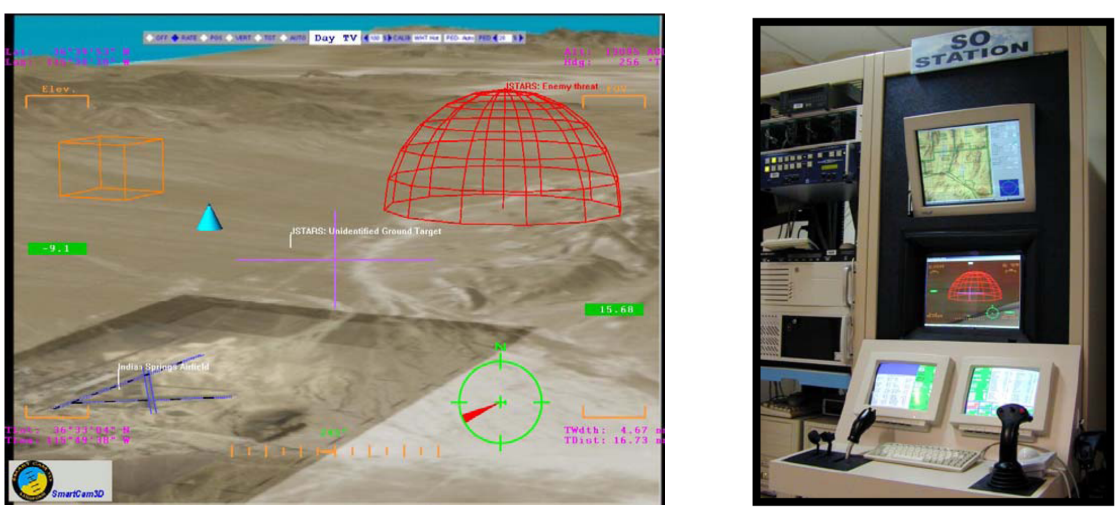

Figure 7. Synthetic Imagery for UAV control and Control Station [Calhoun 2005]

In all, AR interfaces help in providing more information to the operator, but it cannot compensate for the loss of situational awareness due to the time lag in the system and the lack of peripheral vision. Besides, the problem of merging real and virtual data for enriching the visualization is not yet completely solved. Superposition of synthetic images over real data often results in loss of real data due to occlusion [Casals 2005].

Other novel interfaces

Fong et al [Fong 2001], provides a flexible user interface that uses a Personal Digital Assistant (PDA) for teleoperating a ground robot. The PDA navigation tool displays a fusion of collected sensor data overlaid on a map in order to improve operator situational awareness and employs an event driven selective display of images to limit the effects of bandwidth consumption. PDA devices are light weight, portable and have touch sensitive displays that can make the system user adaptive and reduces cost. The interface, however, is subject to the effects of lag caused by transfer and fusion of sensor data, and lack of FOV.

Teleoperation typically requires precise control and maneuvering which is generally not possible using a direct control system. The operator experiences severe fatigue to perceive the remote environment due to lack of cognitive approach in interface design. In order to provide a realistic feel for the operator when driving the vehicle, haptic force feedback devices are added as a part of the control interface [Fong 2000]. The range sensor information is transformed into spatial forces using linear models. The forces are then experienced using the force feedback haptic device. Although, such interfaces can add comfort to the operator and improve performance, they cannot accommodate for the inherent limitations of the system, such as lag.

The network delay regulation method reported in Fraisse *et al* [Fraisse 2003] eliminates the delay jitter and makes remote estimation and prediction using a mean time delay. The system proves to be effective in controlling the jitteriness caused due to random lag. However, the model cannot accommodate lag or improve peripheral vision.

Virtual reality based interfaces

Virtual reality

A computer mediated 3D environment that a human can visualize, manipulate and interact with [Burdea 2003] and in which the user is effectively immersed, can be termed a virtual environment [Brooks 1999]. A typical VR system is comprised of a display unit, a graphics rendering system, a tracking system and a database construction and maintenance system. The system may also contain auxiliary features such as synthesized sound, force feedback (haptic) devices, tracked gloves and other interaction devices. Some of the most common display units are Head Mounted Displays (HMD), CAVE Automatic Virtual Environments, Panoramic displays and Workbenches. Projected environments are generally preferred over HMDs as they are better ergonomically and generally provide a wider FOV. In recent years virtual reality has matured from being considered research curiosity to being accepted as a research and industrial tool. Some of the common applications of VR include vehicle simulators, entertainment systems, product design, architectural design, military training and simulation, psychiatric treatment and robotic surgery in medicine. In the following section, some pertinent examples of VR based teleoperation interfaces are described in detail.

Examples

Milgram et al [Milgram 1997] reported a review on VERO (Virtual Environments for Remote Operations), a virtual reality interface for controlling and manipulating telerobotic systems. The taxonomy proposed in his model suggests that a virtual reality based interface can act effectively as a supervisory control that can provide considerable autonomy for the remote system. The VERO system consists of the virtual environment and the task environment. The virtual model in the system is updated periodically using sensor data from the task environment. The paper suggests that VR interfaces provide variable perspective viewing and are far more flexible than AR interfaces.

Virtual reality based interfaces have played a vital role in underwater teleoperation. As sub-sea scenarios are affected by murky waters and dull lighting, enhanced visualization using virtual reality and sensor data can aid teleoperators in effective maneuvering [Monferrer 2002]. The sub sea environment is very dynamic and cannot be pre modeled, but off shore structures can be modeled before hand. Lin *et al* [Lin 1999] proposed a VR based interface for controlling an underwater ROV that can help explore off shore installations (e.g., for oil companies). The VR system consists of an offshore VR model and a vehicle model. The position and orientation of the VR vehicle model is constantly updated using a sonar based underwater positioning unit. The vehicle has on board cameras and the live video feed from the cameras provides additional guidance for the operator.

The radio controlled blimp developed by Ott *et al* [Ott 2006] is a good example of how a VR interface, when coupled with multiple sensory data and reliable field data, can prove to be an effective teleoperation interface. The blimp is a teleoperated air vehicle that is primarily used for surveillance and security applications. The proposed interface is comprised of a CAVE Automatic Virtual Environment (CAVE) [Cruz-Neira 1992] that provides visual feedback for the operator, on field surveillance agents that monitor the blimp movement to provide real time updates and sensor and video data from the blimp. The teleoperator drives the vehicle from a virtual cockpit using the visuals from the CAVE. The on field agents use a PDA VoIP (Voiceover Internet Protocol) interface to provide the real time data which is integrated and displayed in the VR control room. The system enables multi modal control of the air vehicle. The architecture of the system and a sample picture of the blimp are shown in Figure 8.

Teleoperation for excavation purposes using a Virtual Environment for Remote Operations (VERO) interface was proposed by Ballantyne *et al* [Ballantyne 1998]. A Spar LaserCam is mounted on top of the system in the task environment. The range images are collected from the LaserCam in the form of a bit map with intensity values. A triangulation technique is used to reduce the amount of 3D data, and data from multiple runs of the LaserCam are then fitted together to form the virtual model of the task environment. The model is displayed using the window interface provided by VERO. The paper concludes that VR provides better situational awareness and depth perception.

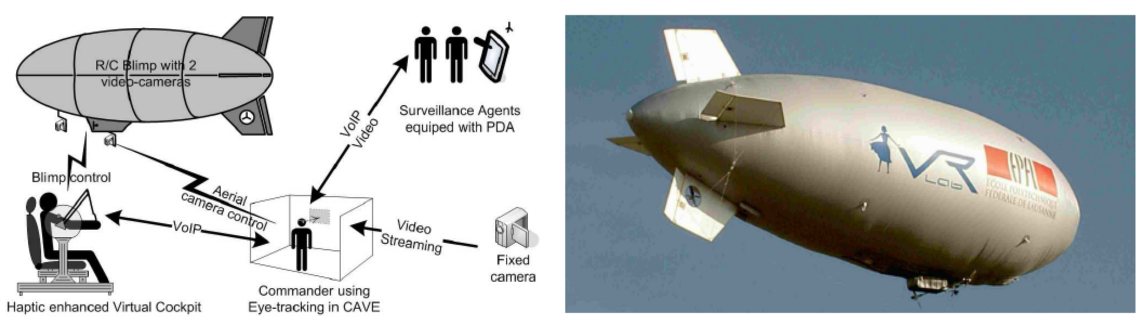

Figure 8. Architecture and Blimp – Remote Vehicle [Ott 2006]

It also indicates that virtual reality based teleoperation can help train operators beforehand, using simulated behaviors.

Why virtual reality?

The various teleoperation interfaces explained in the previous paragraphs indicate that virtual reality can play a vital role in improving the situational awareness of the teleoperator. VR can help avoid or reduce the effects of major limitations of direct control teleoperation like lag and lack of FOV. 3D representations provide users with higher comprehension of the task environment when compared to 2D viewing. The virtual model of the operating environment allows operators to custom select views around the virtual vehicle. Moreover, the views of the operator can be controlled in fly from the operator as cameras in virtual worlds are not limited to their placement and orientation, unlike the physical cameras mounted on the remote vehicle. In fact, projection display devices like CAVEs can provide a 360 degree FOV.

Review on 3D visualization technologies for teleguided robots conducted by Livatano et al [Livatino 2006] emphasizes that stereoscopic visualization assists operator in estimating egocentric and related distances. The paper presents a comparative study between various stereoscopic displays such as a 3D desktop/laptop, 1-sided CAVE, HMD and Powerwall on various characteristics including realism, immersion and user comfort. It concludes that CAVE environments can provide excellent 3D immersion and adequate level of comfort for the user. Demiralp *et al* [Demiralp 2006] conducted a qualitative and quantitative comparative study between CAVE and fish tank VR displays. The researchers took a Magnetic Resonance Imaging (MRI) visualization program as subject for the study and showed that fish tank displays are good for look-in tasks and CAVE displays are good for look-out tasks. Look-in tasks are the ones in which VR helps a user explore the details of a virtual object and the user may not need his peripheral vision, for example as in medical imaging and visualization. Look-out tasks are those like architectural walk through, vehicle simulators, urban combat, teleoperation, etc. in which the user requires peripheral vision in the environment. The study also supported the claim that stereo displays contribute significantly to user performance.

VR environments can provide a continuous display of the task environment without being affected by the lag and jitteriness of camera-based video feedback. This reduces the cognitive work load on the operator, as he/she need not keep track of a mental model of vehicle surroundings. Additionally, the virtual simulation reacts to operator's commands instantaneously. When the real vehicle follows the simulated vehicle, the effects of lag can be minimized and the operator can maintain effective control over the vehicle. The following section explains in detail the virtual reality teleoperation system developed by Walter et al [Walter 2004]. A comprehensive description of this teleoperation is necessary in order to place the thesis' research work in context.

Virtual reality teleoperation

Walter et al [Walter 2004] presents a virtual reality-based teleoperation system. Walter's approach was developed for ground vehicles and uses a large-scale immersive virtual environment as the primary visual context for the operator which is augmented with sensor-generated meta-data. This provides a broad FOV that fosters situational awareness. The system accommodates lag by essentially enabling the operator to control a simulated vehicle in the future of the actual vehicle: providing it a time series of goal states.

A schematic of Walter's virtual teleoperation architecture is depicted in Figure 9. The operator's commands are sent to a vehicle simulation that predicts the dynamic state of the virtual vehicle including its position, velocity, acceleration and heading. The vehicle dynamics simulation produces the simulated state, which is used to position the virtual vehicle and provide a desired location for the teleoperated vehicle. The idea of driving the simulated vehicle and making the teleoperated vehicle follow is based on the wagon tongue path planning method [Kroll 1970]. The teleoperated vehicle uses the simulated states as a series of goal states. A simulation run locally on the vehicle determines the inputs required to get the vehicle to approach the simulated state from its current state.

To calculate these inputs, the current state of the real vehicle (real state) is required. The "observer", an optical tracking system in Walter's implementation, provides the real state. To assist the operator in assessing the deviation between virtual and physical manifestations of the vehicle, an "informed state" is computed as the difference in vehicle

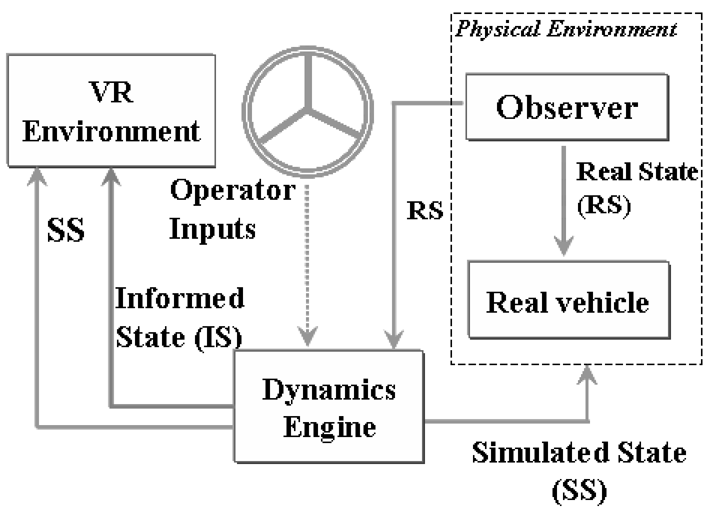

Figure 9. Architecture for VR Teleoperation

positions between the simulated state and the real state. The informed state is used to generate a virtual box surrounding the simulated vehicle that grows or shrinks depending on the magnitude of this discrepancy. This wire-frame envelope shown in Figure 10, allows operators to adjust their control to obtain higher fidelity with the remote vehicle, closing the loop between the human and the computer controlling the remote vehicle.

Experimental results

Since the operator drives the simulation instead of the real vehicle he/she will not be required to accommodate for the lag that leads to the loss of situational awareness. Walter's tests compared the camera-based teleoperation to VR teleoperation with imposed one-way signal lag times of 1, 5 and 10 seconds during a task involving navigation through a set of cones. In these tests, the teleoperated vehicle was a remote controlled model tank. The tank controller was wired to a circuit board and is computer controlled. The response of the tank to these controls was measured to create a computer simulation of the tank's dynamics and response to inputs. The computer running this simulation (the dynamics engine) was a Dell PC attached to a Microsoft Sidewinder steering wheel set. The dynamics engine used the tank simulation to generate the simulated states and then sent those states to the laptop communicating with the RC tank.

The observer system was implemented with a simple optical tracker. A red cardboard square was placed on the top of the tank towards its rear and a blue square towards its front. A webcam was situated at a fixed location above the operational environment to produce a video stream. A simple image processing algorithm was implemented to find the blue and red squares in the scene. Calibration of the camera enabled conversion of the vehicle's marker

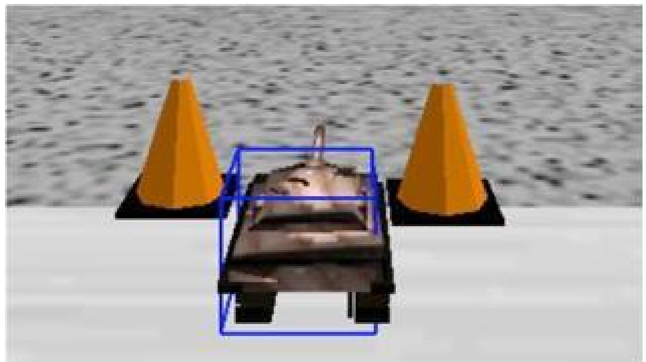

Figure 10. Graphical Representation VR Teleoperation

location into the corresponding location in the operational environment. Incorporating the fixed distance between the vehicle markers and the center of the tank, the system could determine both the tank's real-world position and heading. Further, by keeping track of the previous position and orientation of the tank, the system could provide a first order approximation for the vehicle's linear and angular velocity. This information comprised the real vehicle state required by the vehicle and the dynamics engine as shown in Figure 9. The observer subsystem was implemented on a laptop and communicated the vehicle state information via standard network protocols.

The image generator shown in Figure 9 was an SGI RealityEngine2. It received simulated and real vehicle states from the PC and laptop respectively and generated the virtual world as shown in Figure 9. The C6 CAVE device displayed the virtual world in a 10 x 10 x 10 foot room with each of six surfaces capable of displaying a rear projected stereo image. In this way, the system immersed the user with 3D graphics in every viewing direction. The dynamics PC and steering wheel were physically brought into the C6 space to position the operator within the virtual representation of the operating environment. All of the components of the test system were connected on the same low latency network.

In real UAV operating conditions, significant signal delay between the vehicle and the operator is present. To simulate this crucial effect, each command sent by the operator to the vehicle could be adjustably delayed before being transmitted. Likewise, any information returning to the simulation from the vehicle observer could also be delayed. Thus, simulated signal delay was introduced into the system. Of course constant signal delay is not sufficient to model real world behaviors. To simulate variable signal delay, random perturbations in the delay times were introduced fluctuating by ±10% around the input median value. To manage the changing signal delay times, operator commands to the vehicle were buffered at the vehicle, to ensure that they could be properly spaced in time. This type of packet buffering is a common technique in distributed systems. For example, client-based players of streaming media on the internet typically buffer a portion of the song or video before it is played so that the next frame is available in time despite unpredictable signal delay. Ensuring that the commands reach the vehicle with the correct amount of time between them is important to prevent the vehicle from following a completely different path than what the operator generated.

To test of the system, the tank was piloted through a course of cone gates within the operational environment using three methods: direct control, camera-aided teleoperation and virtual teleoperation. For each method the average time to complete the driving task was recorded as well as the number of gates successfully navigated. Direct control provided the baseline for vehicle control because it is in some sense optimal; there is no signal delay and the operator can see the vehicle directly within its operational environment. Camera-aided teleoperation provided an important benchmark because it represents the most common current interface for teleoperated vehicles. Test runs were performed by the authors for all three control methods with three levels (one, five, and ten seconds) of nominal artificial signal delay. Three runs of each type of test shown in Table 1 were run and the averages of time to completion and number of cone gates navigated are shown. These results reveal that the VR-aided teleoperation system greatly improved operator performance when compared to a lagged video-based teleoperation system. With VR-aided teleoperation, the average time to completion was not noticeably affected by signal delay, even with delays of up to 10 seconds. In contrast, the camera aided teleoperation system completion times increased rapidly with only a modest increase in signal delay. Furthermore, the situational awareness of the operator was enhanced as evidenced by the fact that fewer cones were knocked down with VR-aided teleoperation.

Problems in VR based teleoperation

Virtual reality teleoperation separates the real and simulated states, and thereby ameliorates the interface challenges caused by signal lag time. The interface provides a wider FOV and reduces the cognitive work load on the operator. The system enables far better navigation performance than video based teleoperation. Of course, state separation assumes the primacy of the virtual world created a priori, and that the operator believes what is perceived through the simulation. Research in terrain simulation and modeling has evolved sufficiently to provide three dimensional graphics model from satellite data, just short of real time [Collins 1995]. Hence, the virtual terrain can be very accurate. However, the possibility of an operating environment being different from its virtual representation is high in dynamic environments and change might occur in both time and space.

Table 1. VR Based Teleoperation Experimental Results

Test	Signal Delay (s)	Average Time (s)	Average Cones Navigated
Direct	0	26	5
Camera	1	101.1	4.67
Camera	5	357.7	4.33
Camera	10	583.5	4.33
VR	1	32.5	4.67
VR	5	34.7	5
VR	10	31	5.67

Consider a basic example of teleoperation: remote exploration of an indoor environment. The task involves driving a remote vehicle for exploring various locations of the environment. The task cannot be carried out autonomously as the interesting locations are not selected before hand. The task is relatively simple due to minimal complexity involved. The teleoperator controls the remote vehicle using the virtual model of the indoor environment. The virtual reality based teleoperation interface simulates the vehicle position based on the input commands and checks for error based on the feedback it gets from the vehicle tracker/observer. However, the virtual model of the task environment is not updated real time and real environments are dynamic. Hence not all objects present in the task environment are seen by the operator in the VR environment. Besides, the remote vehicle does not have senses on its own to react or adapt to the environment. Such a situation will invariably lead to accidents or failure of the task at hand.

Consider a nuclear facility that maintains a teleoperated vehicle to do daily maintenance. The task can be considered moderately complex and requires human in the loop for decision making. The entire nuclear facility is modeled before hand but the vehicle lacks any mechanism to perceive the environment in real time. If a situation arises in which the facility is damaged, then part of the vehicle's world model may be inaccurate leading to false assumptions from its remote operator and potentially to an accident. Both the tasks explained above are teleoperation in indoor environments. Let us take a case where a remote ground vehicle is used for surveillance or combat in an enemy land. The task is highly complex and cannot be completely automated. At the same time the environment is very dynamic and hence the vehicle cannot be continuously teleoperated.

Problem statement

Further research is required to address virtual teleoperation in a dynamic environment. The challenge lies in identifying ways to detect environmental change relative to the virtual model of the environment, use this information to enable the vehicle to adapt to the change, and provide the operator with the dynamically updated environment. The research presented here focuses on a teleoperation system that utilizes virtual reality to accommodate lag, improve FOV and enhance situational awareness and at the same time enables periodic autonomous vehicle control to adapt to surprises encountered in a dynamic environment.

CHAPTER 3. VIRTUAL REALITY BASED MULTI-MODAL TELEOPERATION USING MIXED AUTONOMY

The research presented in this dissertation builds on Walter's VR teleoperation approach by integrating on-board vehicle sensors to enable it to adapt to dynamic environments. In addition, the world model is subsequently modified to provide the operator with a dynamically updated virtual environment. The system retains all the components of Walter's VR teleoperation system, thus maintaining the advantages of accommodating lag and limited FOV. However, the real vehicle in this system is augmented with sensors and significant onboard computational power to support an obstacle detection system and limited autonomous decision making. The resulting system is essentially a fusion of VR teleoperation with autonomous obstacle avoidance.

Why sensors?

Sensor augmentation is the prerequisite for any vehicle to perceive the surrounding environment in real time. Considerable research has been reported on sensor fusion interfaces, where multiple sensor data from the real vehicle are integrated and presented to the operator. NASA Ames Research Center [Nguyen 2001], has conducted an extensive study on developing interfaces using real time sensor data. The paper presents analysis on a series of virtual reality based control interfaces that have been developed by the center over a period of time. Although, the researchers concluded that virtual reality plays a vital role in improving situational awareness they observed that a teleoperator interface can be considered comprehensive and complete only when the interface has a way to present integrated real time data from sensors which can be controlled bi-directionally. This virtual reality interface, also called Viz, was tested on a virtual prototype of Mars Lander. Images from the surface stereo imager fitted on the vehicle are processed to provide photo realistic terrain models of the interior. Moreover, the terrain model developed will facilitate future mission planning and analysis. Research by Jarvis et al [Jarvis 1999] and Ricks et al [Ricks 2004] suggests sensors varying from CCD cameras to laser range finders for acquiring information real time. However, it is noteworthy to understand that these proposed systems are modeled for teleoperating vehicles in completely unknown environments, where the teleoperator relies entirely on the lagged data and images. The multi-behavior-based mobile robot developed by Luo et al [Luo 2000] can be teleoperated based on video feed. The robot's onboard computation enables obstacle detection and path planning in circumstances where the operator cannot intervene. However, the entire teleoperation is carried out for a preplanned path with known destinations in a known environment. In addition, the path planning is computed based on an a priori environment model, with pre-stored goal states when the robot becomes autonomous. Although, such a system is essentially a path navigator with obstacle avoidance rather than a teleoperated vehicle, it demonstrates the idea of vehicle adaptation to accommodate surprises.

In the approach presented in this thesis, an overview of which is illustrated in Kadavasal *et al* [Kadavasal 2007], a sensor augmented vehicle is teleoperated based on an a priori model in a virtual environment. The immediacy of the sensory data coupled with a certain degree of vehicle autonomy not only helps the vehicle adapt to dynamic environments, but retains the edge over other teleoperation systems in overcoming time lag and limited FOV.

Providing a vehicle with sufficient autonomy is one of the solutions that researchers have come up with to overcome surprises in a dynamic environment. However, balancing the autonomy between the vehicle and the teleoperator is a challenge. Although, the question of the correct balance between the vehicle and operator depends on the context and task domain [Lyons 1990], nevertheless it is shown that considerable vehicle autonomy is necessary for successful teleoperation [Wegner 2003]. Vehicle autonomy acts as "filler" in situations where commands from teleoperators become sparse due to time delay or for other reasons. The degree of autonomy taxonomy suggested by Milgram et al places successful virtual reality based interfaces at a level in which the operator has partial a prior knowledge about the environment and is provided with partial autonomy for the vehicle. The system architecture described in the following pages support the idea of a virtual reality based teleoperation system incorporating vehicle autonomy that works efficiently within the overall bounds of instructions created by the human in loop.

There are a wide range of sensors with varying characteristics available for depth measurement and it is necessary to understand their advantages and limitations before making a selection. Meier et al [Meier 1999] and Fong et al [Fong 2001] present comparative reviews on a range of depth measurement techniques including stereo vision, laser range finders and sonar. Some of the important results presented in these papers are reproduced here for better understanding. Table 2 and 3 presents a comparison of types of sensors that are predominantly used in robotic and teleoperation interfaces. The results suggest that stereo vision provides good angular resolution with low cost and high speed. The disparity map

Table 2. Sensors comparison - 1

Situation	2D Image (Intensity)	3D Image (disparity)	Sonar (TOF)	LADAR (laser)
Smooth surfaces (no visual texture)	OK	Fails	Fails	OK
Rough surfaces (little/no texture)	Ok	Fails	OK	OK
Far obstacle (>10 m)	Fails	Fails	Fails	OK
Close obstacle (<0.5 m)	OK	Fails	OK	OK
Small Obstacle (on the ground)	Fails	OK	OK	Fails
Dark environment (no ambient light)	Fails	Fails	OK	OK

technique using coordinated stereo images is effective for detecting small objects. However, it is unfit for detecting objects that are too close or too far away from the cameras. Moreover, lack of textures in the scene and low lighting may result in extremely noisy depth resolution. Sonar, on the other hand can detect objects that are far away and are not affected by environmental lighting. However, sonar has poor angular resolution and is prone to error caused by non perpendicular and off axis targets. Further, specular reflections may result in range errors and poor depth resolution. Laser scanners are predominantly used in various teleoperation systems for obstacle avoidance [Surmann 2003, Henriksen 1997]. They have good depth resolution and are not affected by the environmental limitations. But they have low update rates when compared to other vision systems and cannot detect smaller obstacles. The prototype system developed in this research is intended for a lighted indoor environment with small static and moving obstacles. Stereo vision based sensor systems are suitable for

Table 3. Sensors Comparison - 2

Situation	2D Image	Stereo vision	Sonar
Smooth surfaces (no visual texture)	OK	OK	Fails
Rough surfaces (little/no texture)	OK	Fails	OK
Close obstacle (< 0.6 m)	OK	Fails	OK
Far obstacle (> 10 m)	OK	Fails	Fails
no external light source	Fails	Fails	OK

such situations. The system architecture explained in the following paragraphs employs a stereo vision system for obstacle avoidance.

Architecture

Figure 11 shows the high level architecture of the VR based teleoperation system. The system has three major components, namely on-board vehicle sensors, virtual reality operator interface and vehicle adaptation system. Figure 12 shows the detailed architecture for VR based multi modal teleoperation. The operator's commands are sent to the VR sim that predicts the dynamic state of the virtual vehicle including its position, velocity, acceleration and heading. The operator uses haptic-forcefeedback sidewinder wheel and pads to provide the simulated vehicle inputs. The VR dynamics engine helps generate the simulated state based on the operator inputs. This simulated state is then used to position the virtual vehicle in the VR environment and is sent as the new desired location for the teleoperated vehicle. The teleoperated vehicle follows the simulated vehicle based on the

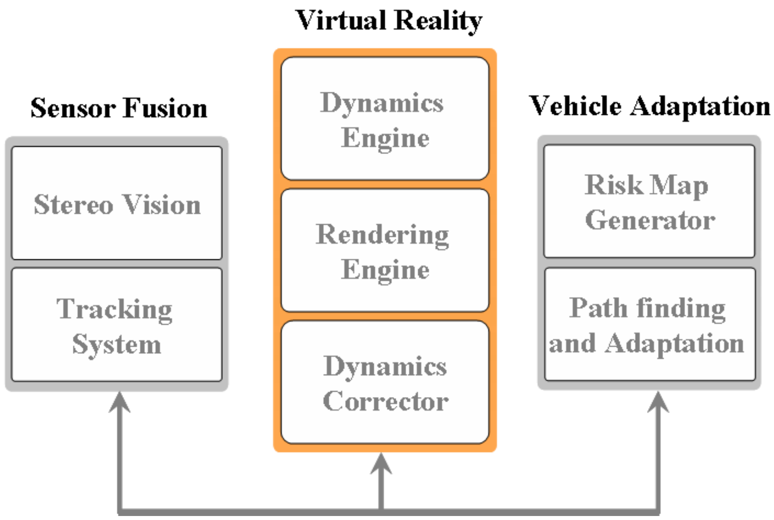

Figure 11. High Level System Architecture

wagon tongue path planning algorithm [Kroll 1970] and is similar to the model presented in Walter et al [Walter 2004].

The InterSense precision motion tracker [Wormell 2003] installed in the teleoperated vehicle environment tracks the teleoperated vehicle and provides the VR environment with the real vehicle states. The real state is accommodated for the lag associated and an 'informed state' is calculated. The error between this informed state and the current simulated state is represented in the form of a transparent blob surrounding the simulated vehicle. The transparent blob expands or contracts depending on the magnitude of the discrepancy. This virtual envelope allows operators to adjust their control to obtain higher fidelity with the remote vehicle, closing the loop between the human and the remote vehicle. The projected real state or the informed state is the current teleoperated vehicle's position as known to the VR environment. This state is represented in the form of a ghost vehicle in the VR interface. The ghost vehicle provides the operator with a clear indication of where the

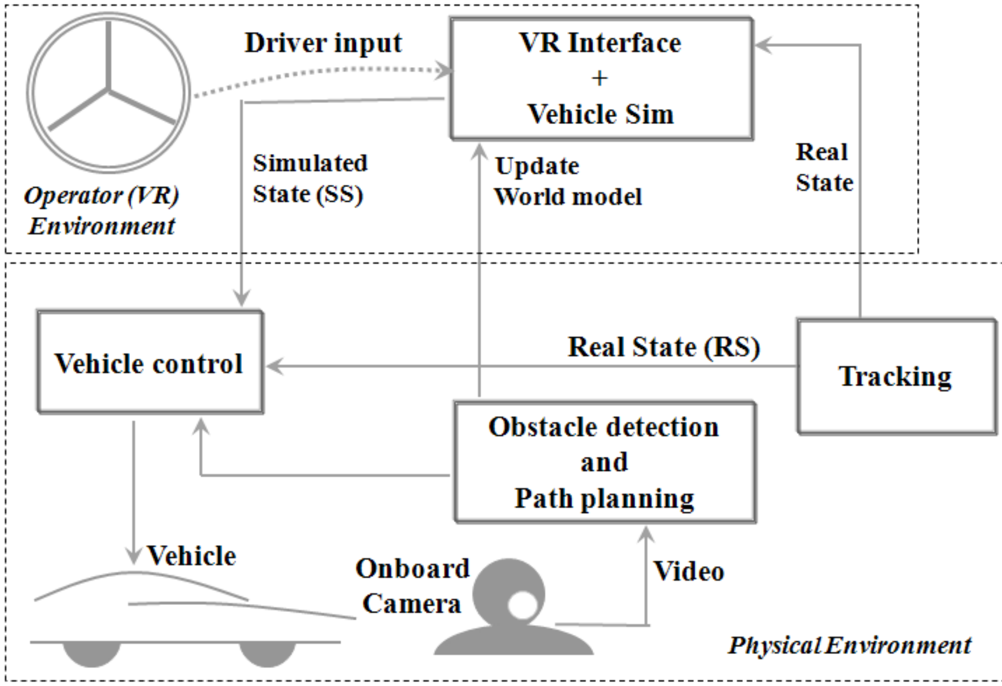

Figure 12. Architecture for Sensor Enhanced VR Teleoperation

real vehicle is thereby facilitating better user adaptation.

The vehicle is augmented with two onboard synchronized cameras, and onboard computation for image processing. These components act as the vehicle's senses. Synchronized stereo vision allows the vehicle to identify any object within a stipulated distance. If the obstacle distance is within the preset threshold value, the vehicle creates a warning. The warning informs the operator about a new object in the travel path along with the distance to the object and its dimensions and coordinate positions in state space. It also provides an estimated time to collision. The new object is computed as the difference between the real and pre-modeled environment and placed in context in the virtual environment. This update is intended to provide the operator with visual reference for the next time the vehicle is operated in the vicinity of the new object.

With the new object detected and a warning issued, the vehicle becomes autonomous. Using the latest real state from the tracking system and a risk map (described later), the autonomous vehicle identifies the nearest goal position that is along the actual path but sufficiently clear of the new object. The vehicle continues driving towards the identified goal without halting. The autonomous vehicle upon reaching the intermediate goal position, reattaches itself to the wagon tongue, i.e., the vehicle again follows the simulated vehicle's path and is no longer autonomous.

The operator is informed about the new path and the wireframe box around the simulated vehicle is updated to denote the degree of the vehicle's deviation from its simulation. However, if a path cannot be generated by the vehicle adaptation system, the vehicle stops. The operator is informed about the scenario and provided with real time video inputs. The video frames are placed in context with the vehicle position in the VR model, for better understanding of the situation. The vehicle may now be teleoperated to a safer position using the video inputs. The system facilitates VR based teleoperation with or without video thereby earning the name multi-modal.

Figure 13 shows a schematic representation of the VR based multi-modal teleoperation system. In the schematic, the simulated vehicle is shown in green and the real vehicle in black. The teleoperator drives the simulated vehicle from the CAVE [Cruz-Neira 1992]. The way points are sent to the real vehicle, denoted by red dots and the real vehicle follows the simulation. The dark gray objects are obstacles present in the a priori VR environment model and the brown object is the newfound obstacle unknown to the teleoperator. The real vehicle operation is divided into two time steps. The stereo vision system on board the vehicle detects the new obstacle in time step one. The on board computation assists the

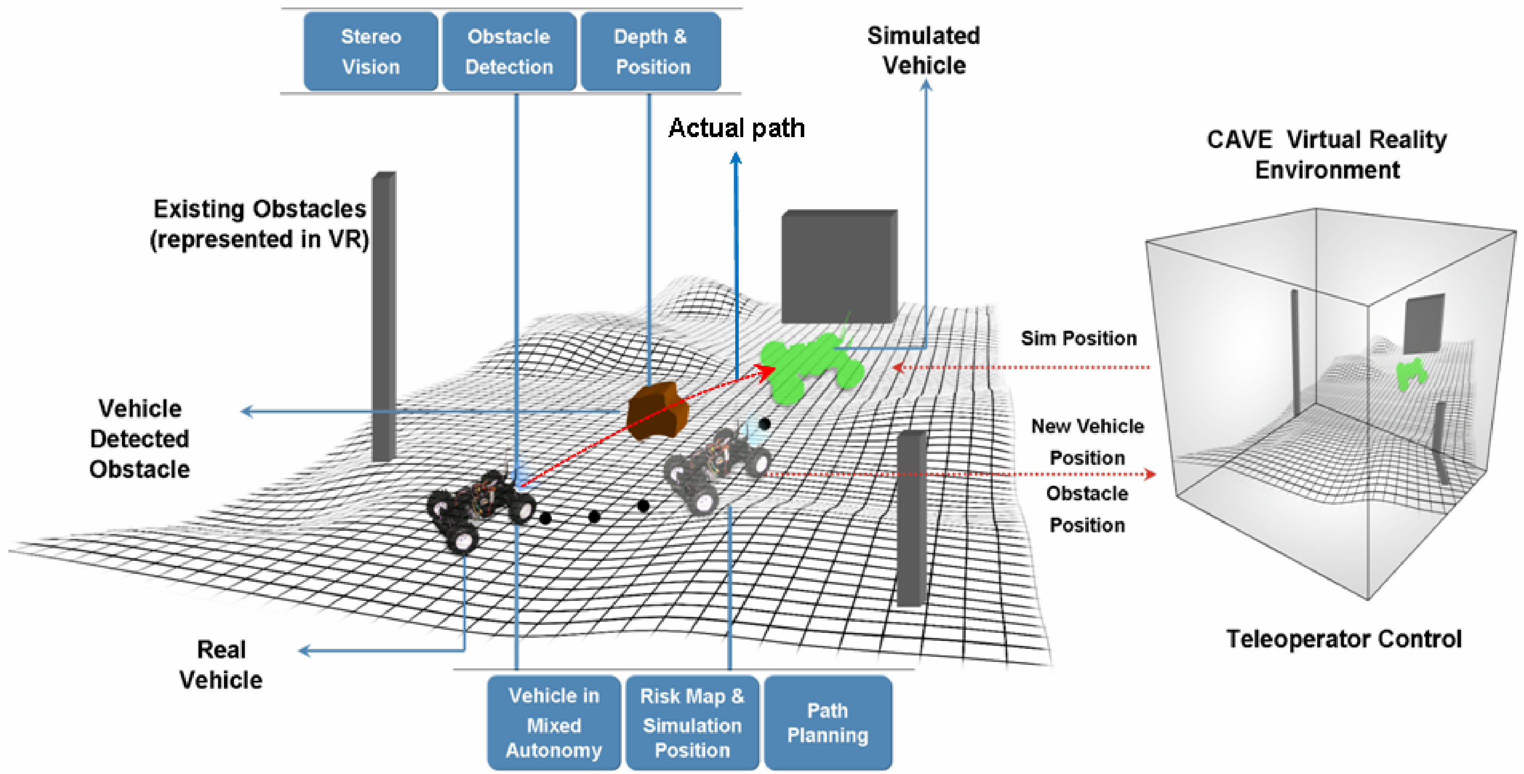

Figure 13. Schematic Representation

vehicle in identifying the corrected path in time step two. This new path is shown in the figure as black dots. The individual components of this architecture are described in the following sections.

Obstacle detection system - Stereo vision

The stereo vision system is comprised of two Unibrain firewire cameras that are connected in series and synchronized. It simulates a low level human eye, which can see and perceive the 3D world [Balakrishnan 2004]. The images from the two cameras produce different perspectives of the same scene, which helps in calculating the difference in relative displacement of the objects in the scene. This relative displacement is referred to as disparity. Simple projective geometry shows that the amount of disparity is inversely proportional to the depth of a point in the scene [Tekalp 1995]. For example, a cross section of the imaging geometry is illustrated in the Figure 14. The optical centers of the two cameras are aligned and parallel to the horizontal *X*-axis. The focal lengths *f* of both the cameras are assumed

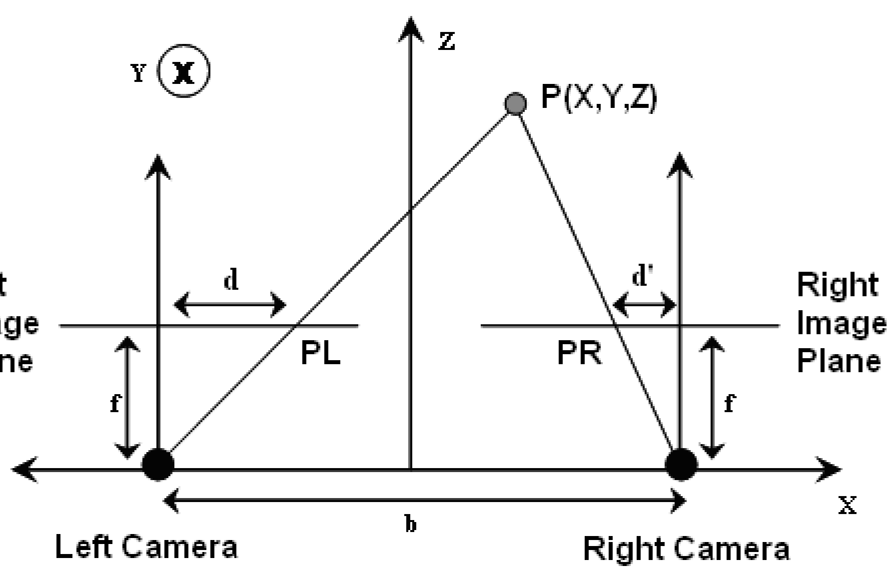

Figure 14. Stereo Image Formation

to be equal. The distance between the optical centers is *b*. Consider a point *P* (X, Y, Z) in the scene that is imaged by the left and right cameras, and PL and PR are the corresponding perspective projections of point P in left and right image planes respectively. Using similar triangles the depth of the point P can be computed as

$$Z = f * b / D \qquad \textbf{Equation 1}$$

Where D, the disparity is calculated as the difference between *d* and *d*'.

The Z value enables computation of the estimated structure and motion of the object in front of the cameras. Structure estimation involves 2D feature matching between left and right camera images. Such a stereo matching algorithm should be fast and accurate in order to calculate the disparities between the images in real time. The resultant disparity map should have object surfaces detailed and distinguished as separate regions with minimal depth discontinuities. There are numerous stereo matching algorithms in the literature. However, they are applicable only for a static camera setup and hence do not satisfy the requirements imposed by VR teleoperation.

Zitnick et al [Zitnick 1999] presents a cooperative algorithm to compute disparity using correspondence. This iterative algorithm identifies the match within the predefined 3D space and accounts for occlusion. However, the algorithm in practice takes about 8 seconds per iteration for a 256 x 256 image size. The maximum flow formulation N-Stereo algorithm by Roy et al [Roy 1998] is another stereo correspondence algorithm that computes precise depth maps albeit with relatively large computational time. Such high time costs are not suitable for a teleoperation system in which sensory data is required to perceive the environment around the vehicle in real time.

The stereo correspondence method adopted in this research is based on Birchfield et al's [Birchfield 1996] pixel-by-pixel stereo matching algorithm. The algorithm matches individual pixels in corresponding scan line pairs while allowing occluded pixels to remain unmatched. The effective pruning technique (to remove unlikely search nodes) proposed in this approach, coupled with dynamic programming reduces the computational time significantly. The algorithm estimates the depth discontinuities by matching the pixel intensities of the images. The algorithm introduces methods to identify non-textured regions and achieves a balance between computational time and depth map precision. The review of such stereo matching algorithms presented in Mark *et al* [Mark 2006] shows that Birchfield's algorithm strikes an effective balance between processing time, accuracy in matching textured and non-textured regions and reliability.

In order to provide faster stereo matching, Birchfield's algorithm assumes that the images from the left and right camera are aligned along the horizontal axis. This can be achieved by image rectification. The intrinsic and extrinsic camera calibration parameters are computed and the images are rectified. The stereo correspondence algorithm computes the disparity map from the rectified images, results of which are shown in Figure 15. The figure shows the camera image along with the computed disparity map. The process rate for the disparity map is approximately 3 Hz. The disparity results are calculated for environments that contain solid, transparent, curved shape and/or textured objects. The algorithm proves to be effective enough to provide precise object surfaces with distinguished separate regions.

The disparity map provides the offset for every pixel between the two stereo images. Further processing is required to identify the objects if any and as well their distance from

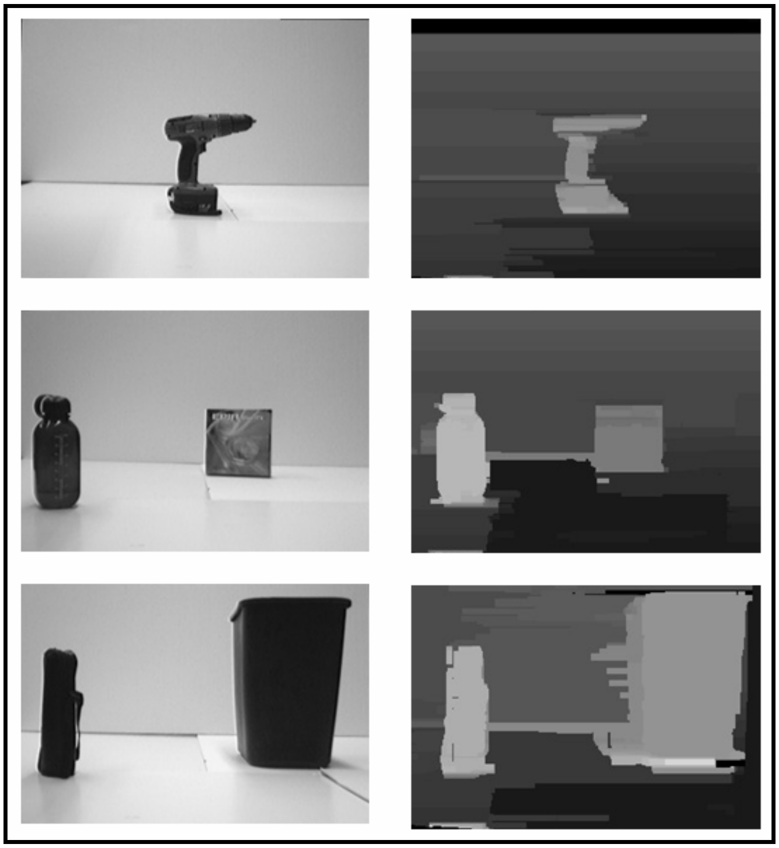

Figure 15. Disparity Maps Results for Static Camera

the camera. For this, the disparity map is first converted into a depth map using projective geometry. An optimum threshold is computed to identify the nearest objects. The objects are segmented using a region growing method and its dimensions are calculated.

Figure 16 shows the flow diagram of the designed stereo vision system. The Open CV [Open CV 2007] library is used for carrying out the image processing calculations. The Open Source Computer Vision (Open CV) library, released by Intel research, is an image processing library used predominantly in computer vision for robotics, human computer interfaces, biometry, etc. The stereo cameras are fixed to a leveled plane and calibrated using OpenCV's chessboard corner calibrators. The output parameters provide the distortion

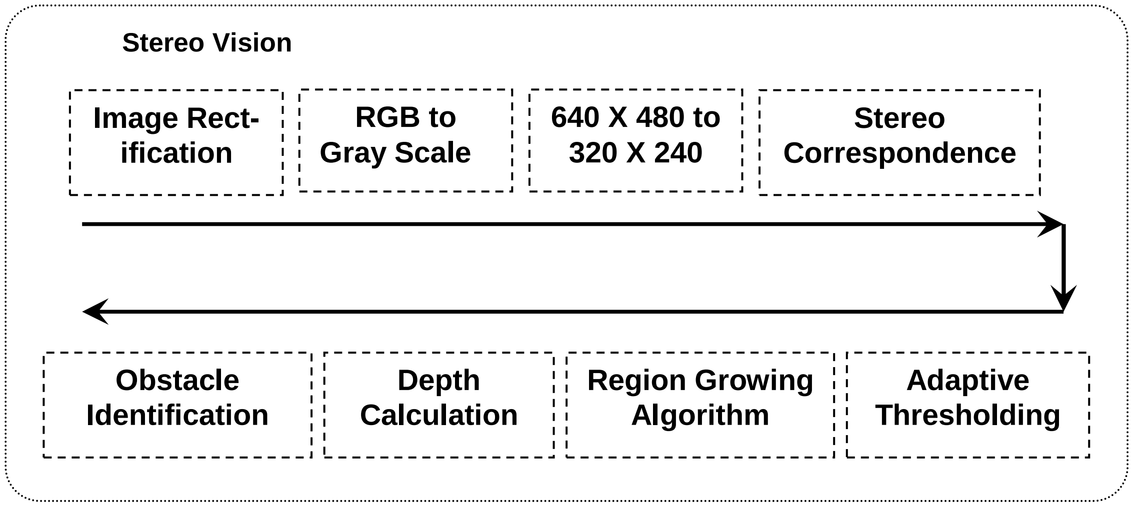

Figure 16. Stereo Vision System – Flow Diagram

and rectification matrices, along with the distance between the two camera lenses. The 640 X 480 RGB stereo images obtained from the camera are undistorted. Both the images are rectified in order to align them in the same plane. The RGB images are converted to gray scale images and its sizes reduced to 320 X 240 resolution. The stereo correspondence function then identifies the matching pixels between the images and calculates the disparity. To enhance efficiency, the maximum disparity limit set for the stereo correspondence algorithm is 100 pixels. The disparity limit bounds the search space for the correspondence search. For example, given a pixel $A(x, y)$ from the left image the correspondence algorithm will not look for matching pixel more than 100 pixels from the same (x, y) location in the right image. This obviously reduces the computational time of the search at the cost of reduced accuracy. Experiments indicate that fixing the disparity limit equal to approximately one third of the image width ensure a reasonable tradeoff between accuracy and speed for this application. When the camera is used in static conditions the disparity maps produced are relatively free of noises. However, the maps turn noisy when the camera is moved around.

To address this, the map is eroded and dilated once to remove the noises. Erosion is the process by which objects in the images are shrunk by a step size equal to that of a predefined structural element. Similarly dilation is the process where the objects in the images are increased by a step size equal to that of a predefined structural element. An equal number of erosion followed by dilation results in an image with less noise but at the same time keep the objects in the images to its original sizes. Further experimentation indicates that image erosion provides sufficient smoothing. The map is eroded once using a 3 X 3 rectangular element. The map is then dilated to restore the objects back to their original sizes using the same element size. In order to remove unnecessary pixel values a threshold limit has to be identified. The process of thresholding and object identification is time consuming and increases the cost of computation. However, these processes have to be carried out if and only if there is an obstacle/object within some prescribed vicinity. Once a depth map is calculated the system checks for number of pixels that are greater than a predefined depth threshold value. And when the number of such pixels is greater than or equal to one third of the image area, an obstacle flag is set. The images are further processed if and only if the obstacle flag is set. Otherwise, the images are discarded.

An adaptive thresholding algorithm [Otsu 1979] is used to calculate the threshold value. The image is split into four equal parts, each of size 80 X 240. A simple gray thresholding is performed on each of the four parts. The process identifies the first intensity peak in the given image and uses that as the threshold value. The threshold value for every split up image will be different. The splitting of images into four parts helps save some features which could have been removed during an ordinary thresholding. The images are then joined back to form a single threshold image. The Open CV region growing algorithm is then employed to group the common regions in the threshold image. This array of grouped regions has the list of identified obstacles. The most common disparity value is identified in each group and is then applied on the rest of the pixels in that individual group. The final output image will be blocks of regions having a uniform disparity value each. The depth of each region is then calculated using the projection formula explained earlier.

Virtual Reality based Multi-Modal Interface

VR interface architecture

The VR interface architecture allows the three components to work in tandem using an event based command loop architecture [Batkiewicz 2006]. The architecture is derived from the command-loop design pattern [Gamma 1995] and follows the object-oriented paradigm, which allows modularity and scalability in design. The flexibility of the architecture helps minimize object dependence and enables scalable interactions. The architecture presented [Kadavasal 2009] consists of three main components as shown in Figure 17. The first component is a command object, to facilitate easy inter-object communication without requiring direct knowledge about each object. The second and third components form a two-layer hierarchy, with the top layer composed of a single object (component manager) that manages the second layer (managers) and synchronizes communication between components of the second layer.

The virtual reality interface architecture has nine managers that communicate with each other using the event based command loop. When a task is assigned to a manager, the manager receives the information in the form of a command. The manager recognizes those commands which are registered under its module, and completes those tasks. If the outcome

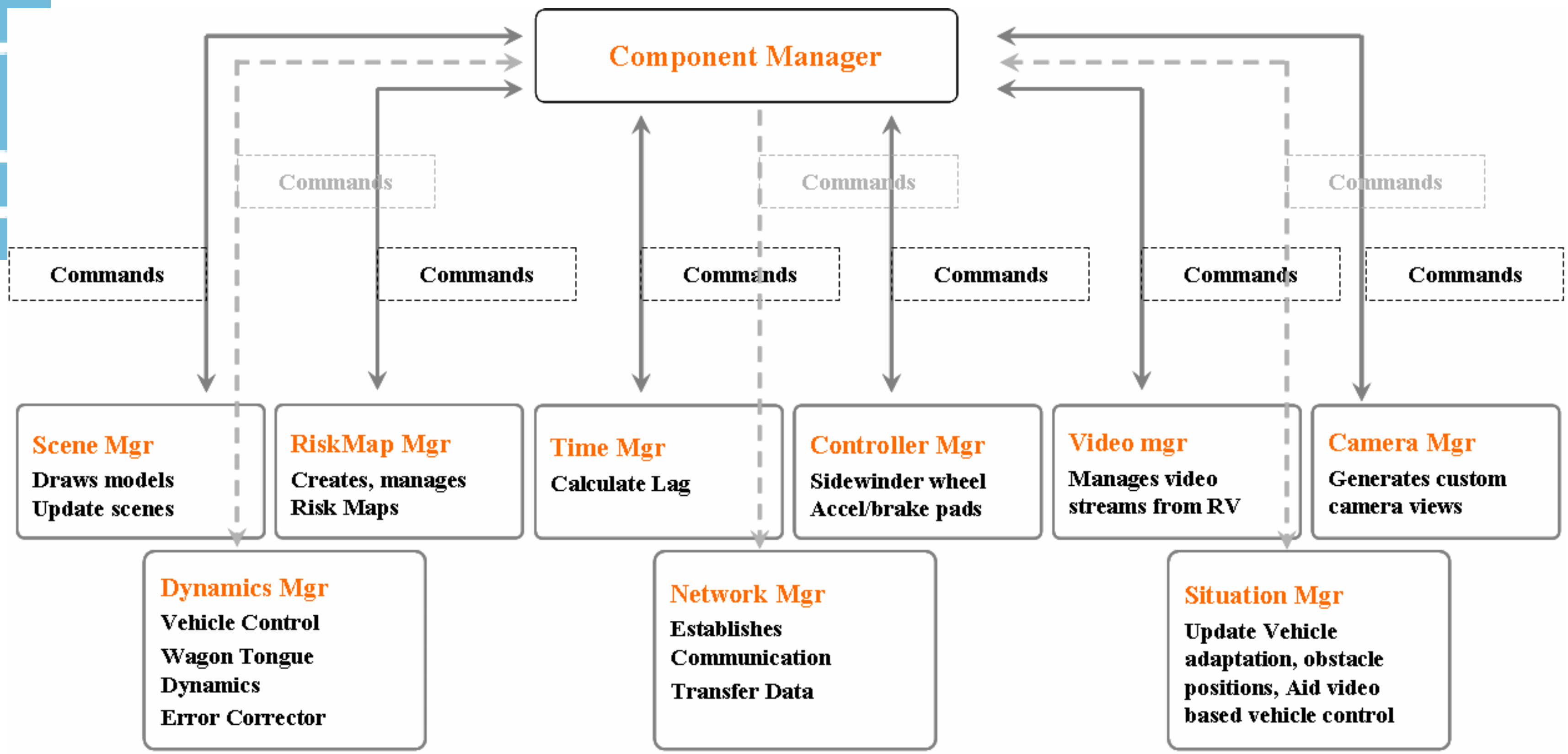

Figure 17. Virtual Reality Interface Architecture

of the completed task needs to be reflected in any other manager, the manager creates those tasks as commands and pushes them into the command list. The command list circulates among the various managers in an orderly fashion for them to be executed. The following sections describes in detail the various managers associated with the VR interface architecture.

Dynamics Manager

The dynamics manager class handles the simulated vehicle control. The manager essentially has three processes to carry out, namely, 1. Calculate the position and orientation of the simulated vehicle state based on the inputs provided by the operator, 2. Obtain the real vehicle position from the tracking system to calculate the error and establish the wagon tongue control, and 3. Correct the simulation vehicle course if the real vehicle's local path adaptation fails. The dynamic manager uses a vehicle dynamics model with zero slip angle assumption [Baack 2004]. It assumes that the direction (heading) of the vehicle and direction of the steered wheels are the same. The terrain is assumed to be flat and hence the vehicle is assumed to have negligible effect due to vertical forces on the wheels. Such a model cannot support uphill or downhill slopes. However, this is not a limitation of the methodology as a complete 3D dynamics model is realizable. A simple yaw plane model is sufficient to test the validity of the system. In addition, the model assumes that there are no violent vehicle maneuvers. A brief description of the vehicle dynamics model is provided in the following paragraphs.

The vehicle parameters are represented in Figure 18. The steer angle input for the front and rear tires are denoted as δf and δr respectively. The velocity of the vehicle along the

x-axis is denoted as u, and along the y-axis as v. L is wheel base; b is the distance between the center of gravity of the vehicle and rear tires; and β is side slip angle. The angle of rotation of the vehicle about the z-axis is termed yaw and r the yaw rate. The following are the model equations used for computing vehicle motion parameters:

$$r = \frac{u}{L}\left(\tan\delta_f - \tan\delta_r\right)$$ **Equation 2**

$$\tan\beta = \frac{b}{L}\left(\tan\delta_f - \tan\delta_r\right) - \tan\delta_r$$ **Equation 3**

$$v = u\tan^{-1}\beta$$ **Equation 4**

$$yaw(t) = yaw(t) + rdt$$ **Equation 5**

The x and y coordinates of the vehicle position are computed using the following relationships.

$$x = u\cos(yaw) + v\sin(yaw)$$ **Equation 6**

$$y = u\sin(yaw) - v\cos(yaw)$$ **Equation 7**

The stereo vision system is sensitive to vehicle vibrations and hence the real vehicle is driven at a constant speed. Similarly, the simulated vehicle is driven at constant speed with

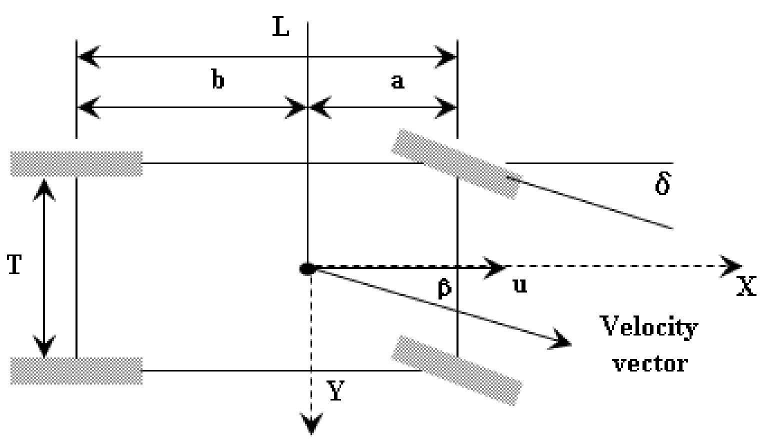

Figure 18. Vehicle Model Representation

variable steering inputs. The calculated vehicle positions are sent as a command to the network manager, which then sends it to the real vehicle. A preset command rate is incorporated into the system to control the rate at which commands are issued.

The dynamic manager implements a wagon tongue algorithm [Kroll 1970], as shown in Figure 18, to identify a projected state of the real vehicle. The algorithm takes the latest real state and projects it to a new real state based on the lag between the stations. This algorithm is used to calculate the inputs for the real vehicle as well as to calculate the error state that is represented in the VR operator station.

Wagon Tongue Algorithm

The wagon tongue method estimates the vehicle's future location taking into account its current velocity, heading, acceleration and the desired state [14]. The algorithm is implemented both in virtual reality interface and in real vehicle interface but for different purposes. The virtual reality interface uses the wagon tongue algorithm to calculate the projected real state also known as 'informed state'. This state is used to draw the transparent error blob and the ghost vehicle. The real vehicle interface utilizes the wagon tongue algorithm to identify the 'new' desired state at that time instant, and to identify the desired speed.

The algorithm projects the real state [*RS*] from the tracker to the current time based on system lag, command rate, latest teleoperated vehicle speed and the preset lag, where system lag is the existing lag between the stations in the current setup and preset lag is the user defined forced latency to simulate larger physical distance between the operator station and driving environment. In the virtual reality interface, this projected real state [*PRS*] is known as 'informed state' [*IS*]. The error [*E*] between this informed state and the current simulation state is represented in the form of a transparent error blob around the vehicle. The 'informed state' is the latest real state known to the VR environment. This is represented in the VR environment in the form of a ghost vehicle. The ghost vehicle traces the exact path taken by the teleoperated vehicle in the actual environment. Let *RS* be created at time *T* and let the current VR time be *T1*. Then,

PRS(T1) = IS(T1) = RS(T) + (T1-T)* Real Vehicle Speed **Equation 8**

PRS(T1) = IS(T1) = RS(T) + (T1-T)* Real Vehicle Speed **Equation 9**

E(T1) = IS(T1) – SS(T1) **Equation 10**

In the real vehicle, the wagon tongue algorithm projects the real state from the tracker based on system lag, command rate, vehicle speed and preset lag if any [*PRS*]. The algorithm also projects the latest simulated vehicle state [*SS*] based on system lag, command rate, latest simulated vehicle speed and preset lag [*PSS*]. However, the vehicle is going to take at least "*w*" seconds to reach this new simulated state, during which the simulated vehicle would have traveled further. Hence, the new simulated state is projected further for a time constant "*w*" also known as wagon tongue constant [*DSS*]. Once the desired future state is calculated, the distance between that state and the current state can be calculated. This distance divided by the "wagon tongue constant" (*w*) is the speed with which the vehicle will be driven by the dynamics engine. The wagon tongue constant is a value that is assumed by the software. It is the time value within which the vehicle must reach its target position. Let *RS* be created at time *T* and *SS* be created at *T1*. Let the current real vehicle time be *T2*. Then,

PRS(T2) = RS(T2) + (T2-T)* Real Vehicle Speed **Equation 11**

PSS(T2) = SS(T1) + (T1-T)* Simulated Vehicle Speed **Equation 12**

DSS(T2+w) = PSS(T2) + w*Simulated Vehicle Speed **Equation 13**

Using PRS(T2) and DSS(T2+w) the "new heading" and "desired speed" is calculated. Figure 19 shows the wagon tongue method for a wagon tongue constant value of 1 second. In general, the real vehicle never arrives at the desired position because the desired position arrives before *w* seconds passes and the method is applied anew. With required speed and position known the real vehicle interface can now calculate the heading and throttle input values for the vehicle to move.

Time Manager

The time manager employs a VPR based thread class [VR Juggler 2008], which spawns threads to communicate between the VR operator interface and the real vehicle, and between the VR operator interface and tracker station. Similar time managers are incorporated in the tracking station and real vehicle. The time manager thread uses User Datagram Protocol (UDP) to send and receive packets. When the system is initialized, the time manager starts sending packets to the VR station, the tracker station and the real vehicle. When the time managers in the tracker station and real vehicle station receive them, the managers send a response back to the VR operator station. The timers employed in each thread help calculate the time taken for a particular packet to be sent and received. The manager then calculates the average time taken based on data from multiple packets. This is the existing time lag between the systems. The system calculates this average lag for every 5 minutes in order to ensure that network variability is accommodated. The time managers calculate the lag between the VR operator station and real vehicle as well as between the VR

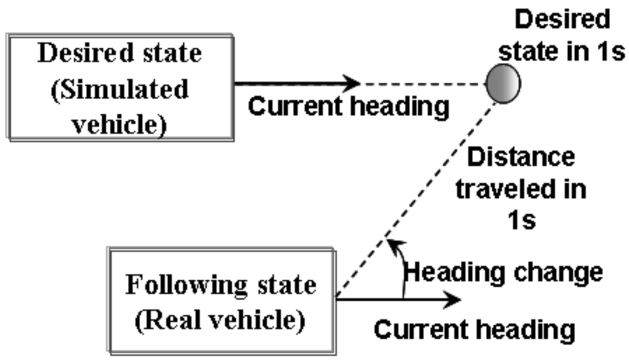

Figure 19. Wagon Tongue Method to Calculate Desired Future State

operator station and tracker station. However, the lag between real vehicle and tracker station is not calculated in order to avoid computational over load on the real vehicle. The lag value between real vehicle and tracker station is assumed to be the same as that of VR station and tracking station, as the network setup between them is similar.

The real vehicle receives data from the VR operator interface which includes system lag, command rate and an artificially added/enforced lag (if any). Command rate is the preset rate with which commands are pushed into the command list in each of the three systems. The sum of all these data is the total latency existing between the time at which the simulation data is created and the time at which it is received by the real vehicle. The real vehicle uses its own timer to identify the time at which the data has been received and projects it to the current time using the total latency value. This process ensures that the timer employed in each station, namely, the real vehicle, the VR operator interface and the tracker station are independent of each other, thereby avoiding the discrepancies caused due to global clocks.

Network Manager

The network manager is responsible for establishing communication between the tracker and VR operator interface, and between the real vehicle and VR operator interface. The class uses VPR libraries for threads and UDP as the network protocol, similar to that of the time manager. This manager receives real state data from the tracker and sends simulated state data to the real vehicle. It also receives real vehicle warnings and its video feed. The network manager sends the "updated" vehicle risk map periodically to the real vehicle.

Camera Manager

Lack of FOV as explained earlier is considered one of the major problems in teleoperation resulting in loss of situational awareness for the operator. In the system presented here, the 3D model allows the operator to set custom views of the vehicle while driving the simulation. The camera class manages the camera positions of the VR environment and allows the operator to provide keyboard inputs for changing operator views in runtime.

Video Manager

When the vehicle adaptation system fails to identify a suitable path to move beyond the obstacle, the operator is notified about the obstacle, the real and simulated vehicles stop, and the real vehicle begins sending video input directly to the VR operator interface. The video manager class receives the video and positions the frames taking into consideration both the real and simulated vehicles' position and orientation. Figure 19a shows a snapshot of the interface with the video billboard. When the operation is in video mode, the operator drives the vehicle using video feedback. Moreover, the older video frames are placed in context with the VR environment in order to provide the new wider FOV for the operator.

Scene Manager

The scene manager class manages drawing and rendering scenes and utilizes the OpenSceneGraph [OpenSceneGraph 2008] library. The manager loads the 3D models of the environment and the vehicle and carries out various functions which include drawing the informed state box, heads up display and transparent vehicle blob.

Controller Manager

The controller manager class manages the operator input devices. The VR operator interface uses a Microsoft Sidewinder force-feedback wheel with acceleration and brake pads for primary operator inputs, and keyboard and mouse for secondary inputs. The manager calibrates the sidewinder steering input and pad inputs and sends them as commands to the dynamics manager. The keyboard and mouse inputs are predominantly used for handling multiple camera views, heads up display (HUD), informed state correction box and other display features.

RiskMap Manager

The risk map manager classifies the 3D model into risk zones based on pre-existing data. The teleoperator classifies the operation's caution level as high, moderate or low at the start based on presumed risk levels. The risk map manager generates real time local risk maps taking into consideration the 3D virtual environment and preset caution levels. The risk map manager is explained in detail in section "vehicle adaptation system" in this chapter. The authors propose that this manager could act as an interface between the VR operator interface and satellite systems that can provide real time 3D data about the vehicle environment in future.

Situation Manager

This manager keeps track of the real vehicle's vehicle adaptation system and alerts the operator about new obstacles and the vehicle's autonomy status. The manager also helps in positioning new obstacles in the virtual environment.

Virtual Reality Interface

The virtual reality architecture explained in the previous paragraphs helps realize a VR interface that can control the real vehicle in mixed autonomy. Some of the major features of this interface are the environment model, ghost vehicle, camera views, transparent vehicle, heads-up-display, video inset and obstacle updates. The following section discusses these features in detail.

VR Environment and Simulated Vehicle

The virtual reality interface consists of the 3D model of the driving environment along with a representation of the simulated vehicle as shown in Figure 20a. The model is physically accurate both in terms of dimensions and objects present in the environment. The 3D model of the vehicle is a representation of the real vehicle and is scaled proportional to the actual size of the operating environment.

Multi-view Camera

The operator is provided with controls to select the camera view from which he/she can view the environment in the simulation. The user is provided with three major camera views namely, a global environment view, a local rear camera view and a local chase camera view as shown in Figures 20b, c and d. The global environment view allows the user to observe the entire driving environment from a single viewpoint. The local rear camera view and chase camera views allows the user to view the environment from the vehicle's viewpoint.

Vehicle Ghost

The VR interface provides the operator with a ghost vehicle representation. The informed state calculated using the wagon tongue algorithms provides the operator with knowledge about the latest real vehicle state. This informed state is represented in the form of a ghost vehicle in the VR environment as shown in Figure 21a. This provides the operator a visual cue of the distance between the simulated vehicle and the real vehicle.

Transparent blob

The informed state may also be represented in the form of a transparent vehicle blob around the simulated vehicle. Figure 20d shows a snapshot of the environment with transparent blob. The vehicle blob grows when the distance between the real vehicle and simulated vehicle increases, and it shrinks when the distance decreases. Although the blob represents the same informed state value as that of the vehicle ghost, it is useful when the operator is in local camera views and the real vehicle is far away from the simulated vehicle.

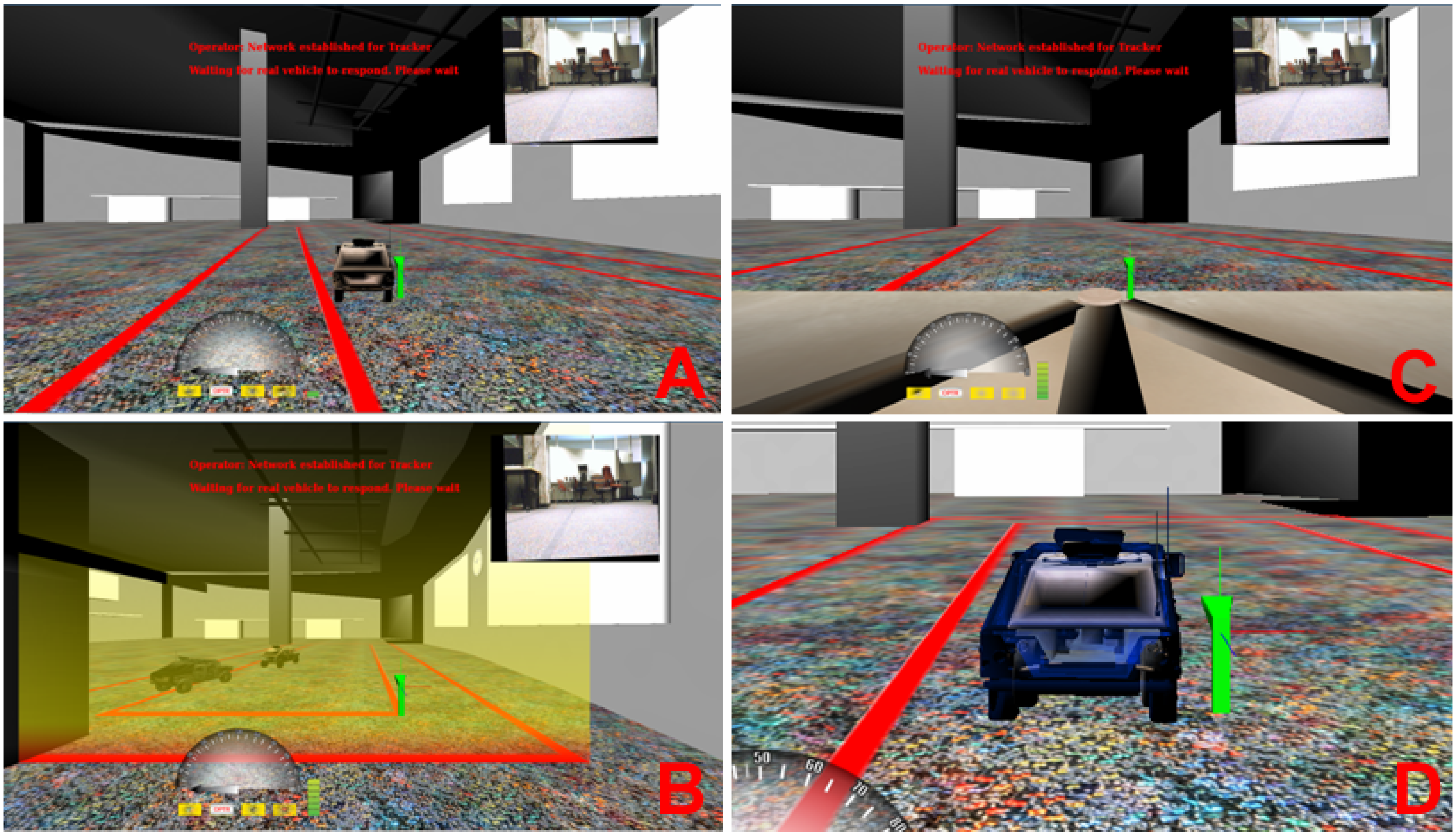

Figure 20 a. VR Environment with Sim vehicle b. Global camera view c. Local camera view – chase cam d. Local camera view – rear cam and Transparent Blob

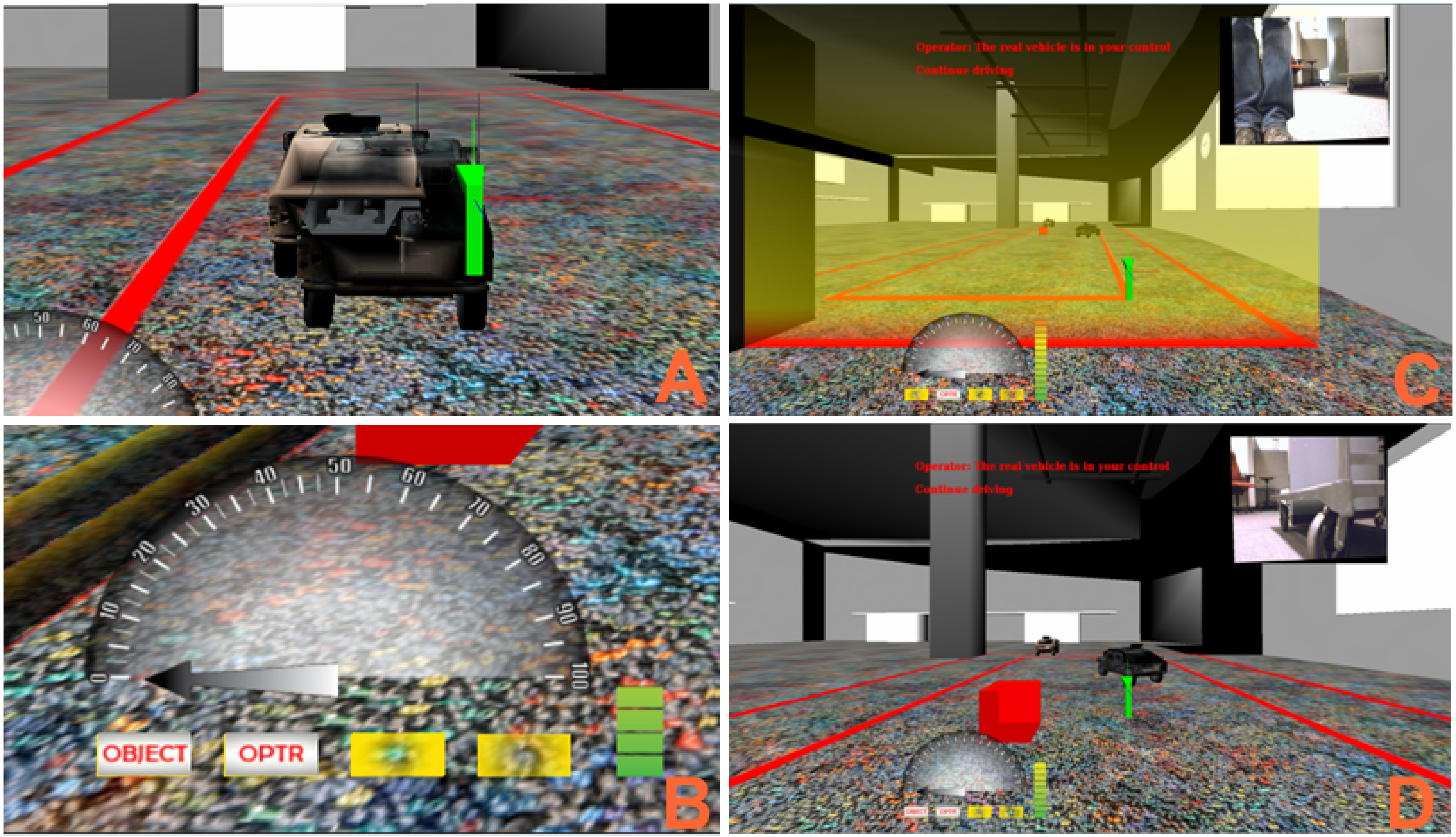

Figure 21 a. Ghost Vehicle b. Heads-Up-Display c. Updated Environment with New Obstacle d. Updated Environment – Zoomed In

Heads-up-display

The VR interface also incorporates a heads-up-display (HUD) that provides the operator with the speed of the simulated vehicle, a quantified representation of the informed state in the form of a distance bar, information regarding the control state of the real vehicle, i.e., whether it is in operator control mode or in autonomous mode, and obstacle warnings. Figure 21b shows a snapshot of the HUD.

Video Inset

The network and video managers enables the VR interface to obtain video feed directly from one of the vehicle's cameras. This video is presented to the operator in the form of a bill-board inset as shown in Figure 20 and 21. Although the video is lagged due to bandwidth limitations, nevertheless it can provide the operator with substantial information about the real environment.

Obstacle updates

The VR operator interface provides the user with obstacle warnings in the form of text messages. The operator is presented with warnings when the real vehicle identifies an obstacle and when there is a shift in autonomy. The VR interface updates the environment model with a 3D representation of the obstacle when the real vehicle identifies it. Of course, these warnings are subjected to a small amount of lag. However, providing the operator with these necessary updates improves situational awareness. Figure 21 c and 21 d shows the snapshot of the environment with the newly found obstacle added to the terrain.

Tracking system

In order to obtain the real state (position and orientation) of the vehicle, a vehicle localization method was developed. Ground vehicle teleoperation uses a wide variety of vehicle tracking methods; the common ones being Global Position Systems (GPS) and vision based tracking systems. GPS based tracking systems are predominantly used in outdoor teleoperation systems. The best of GPS has resolution levels in the range of 3-5 m. However, GPS is not suitable for indoor teleoperation systems. Vision/camera based image processing is a commonly used tracking method for testing teleoperation in indoor environments. These systems search for two or more predefined color-coded objects on the tracked vehicle and calculate its position and orientation based on the camera's position and orientation. However, the cost of computation due to image processing can lead to relatively large latency.

In the current teleoperation system, a 6-DOF InterSense IS-900 precision motion tracker [Wormell 2003] is used for vehicle localization. The hardware consists of sonistrips, tracked devices, and the processor unit. The sonistrips are long rod with ultrasonic transponders that receive signals from the base processor unit and transmit ultrasonic pulses in response. The tracked devices are sensor units that outputs *X*, *Y*, and *Z* position along with roll, pitch, and yaw information. Obviously, this tracking system is not suitable for teleoperation in general. The rationale for choosing the InterSense system is to essentially simulate a high resolution GPS in an indoor environment.

The sonistrips are mounted parallel to each other as shown in Figure 22. Each strip has 3 transponders positioned equidistant from each other. The transmission beam width for each transponder is adjusted such that the approximate cone angle is 70 degrees. The acoustic beam transmitted by the transponder is detected by the tracked device (sensor). Timers in the tracking processor record the time when the signal is transmitted. Simultaneously, timers in the tracked device record the time when the signal is received. The range measurements are made using the calculated time differences and speed of sound (calculated from the measured ambient temperature). The tracking measurements are more reliable in areas where three or more acoustic cones intersect. In the current experimental setup, the 6ft long sonistrips are placed approximately 4ft apart. The origin of the system is located between the third and fourth strip as shown in Figure 22. The strips are positioned 8ft above the ground. A snapshot of the tracking set up is shown in Figure 23.

Real vehicle station

The real vehicle station is made of real vehicle prototype along with the on board computational system which includes the stereo vision system, vehicle adaptation system and vehicle control. Figure 24 shows the architecture for the real vehicle station. The vision manager class implements the stereo vision system and the adaptation manager class

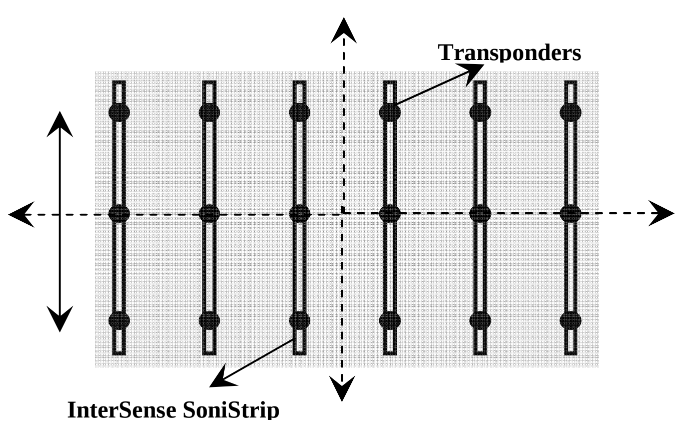

Figure 22. Tracking system - SoniStrip Arrangement

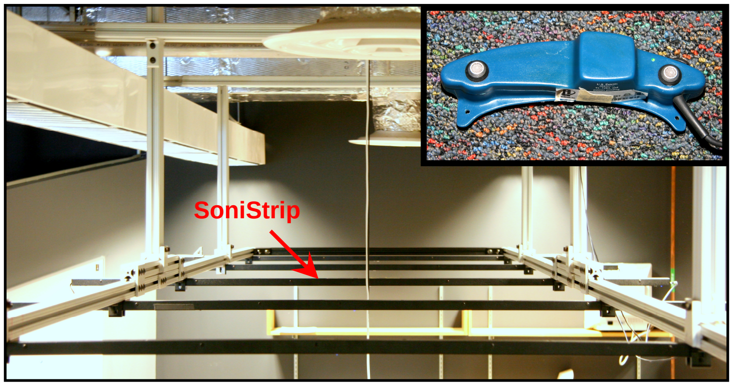

Figure 23. Tracking system - Set up – Snapshot (Inset – Tracked device)

implements the vehicle adaptation system. The time, network, video and dynamics managers work similar to the VR interface architecture.

Vehicle adaptation

The real vehicle does not have knowledge about the state space and is completely controlled by the operator. Hence, in the autonomous state, the vehicle has to either rely on its on-board sensors for determining appropriate control inputs or the information it received previously from the operator's control or some combination of both. There has been considerable research in sensor augmentation and vehicle autonomy. However, the research goal here is not to develop an autonomous vehicle that can survive in an unknown environment, but to develop a system that can be teleoperated using VR as a tool (to accommodate lag and provide FOV) and at the same time adapt to the partially unknown dynamic environment and increase the operator's degree of confidence.

System description

This research proposes an optimized path finding method that identifies paths after

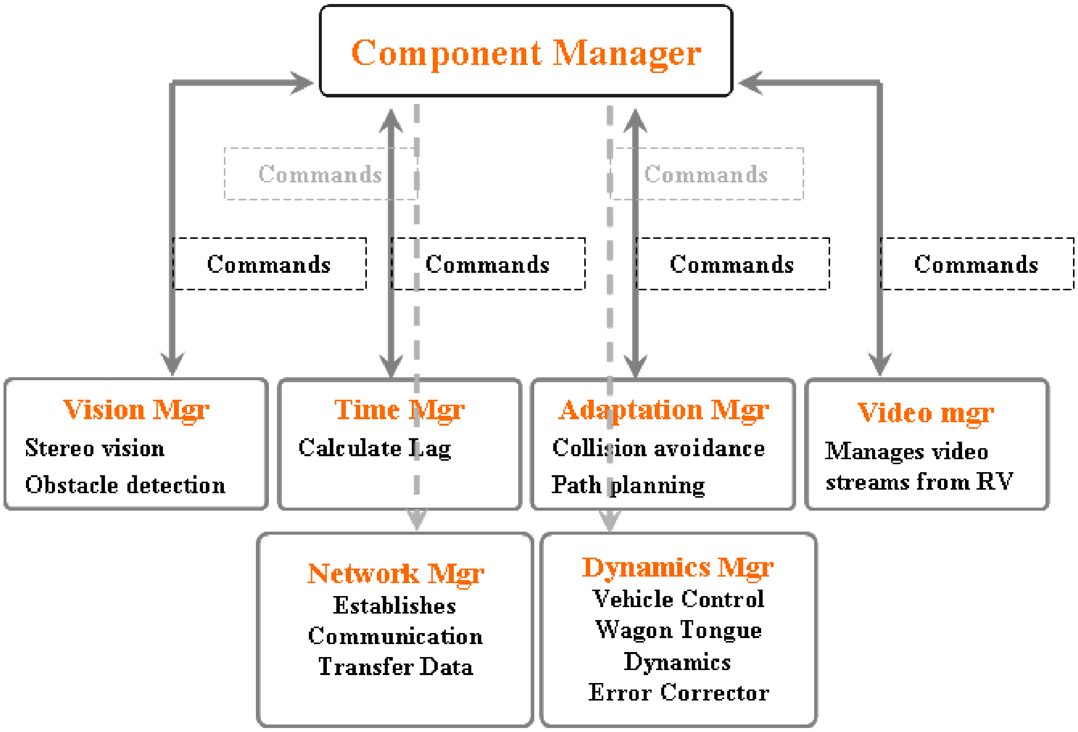

Figure 24. Real Vehicle System Architecture

correlating and synchronizing the previously available terrain knowledge and risks, with the new environment data. The architecture for the vehicle adaptation system is shown in Figure 25. The a priori model state space is classified into various zones depending on the level of risks as shown in Figure 26. It is assumed that the terrain data and risks are continuously updated within the operator's environment from various information resources (e.g., newly found enemy assets).The vehicle operation can be classified before hand with respect to the overall level of caution that is necessary. For example, driving a remote vehicle in a terrain that has suffered from an earthquake will have a different caution level when compared to driving a remote vehicle inside an enclosed space like a building. The caution level indicates

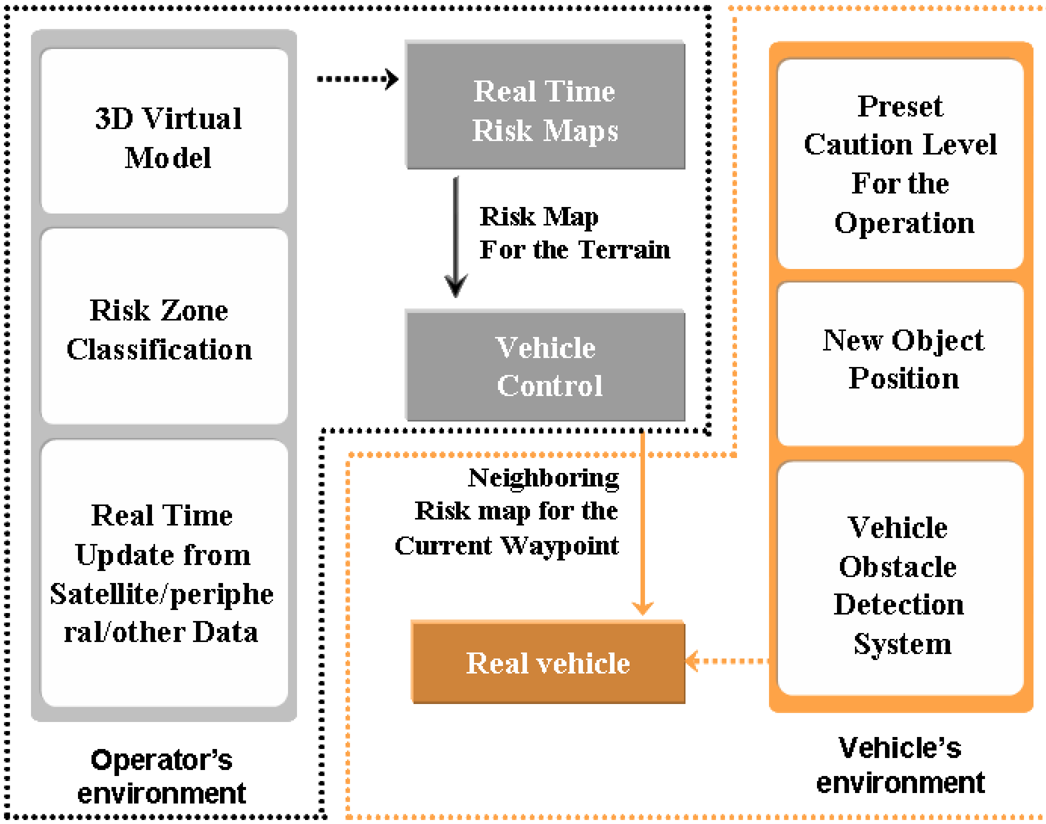

Figure 25. Vehicle Adaptation System Architecture

the degree to which the vehicle can take chances in precarious situations. This free parameter is preset by the human operator for the specific operation.

The virtual world provides the real vehicle with a risk map of neighboring regions corresponding to each goal state. The method accounts for the real state error. When an obstacle is detected, the autonomous vehicle uses its risk map for the current position and correlates the new object position to the risk map. The path planning method will then identify the new path for the vehicle based on the actual goal (simulated) state, risk levels of the neighboring zones and the preset caution level. Depending on the preset caution level, the vehicle will either consider a high, moderate or low risk neighboring zone as the alternate

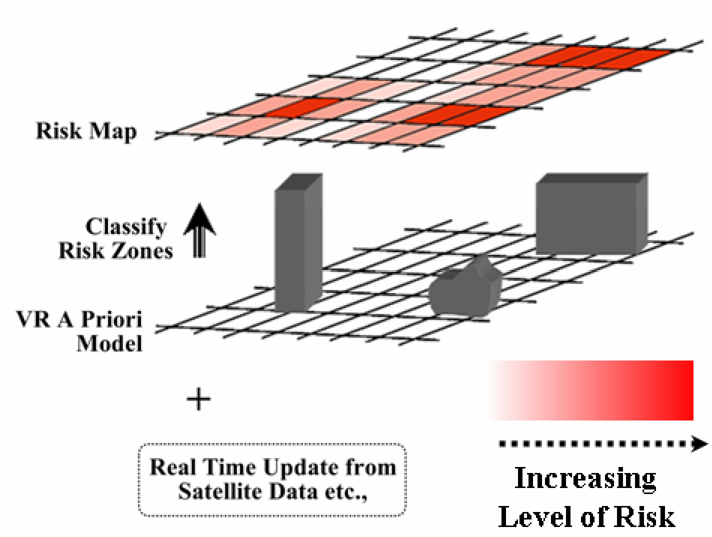

Figure 26. Risk Map Classification

path. The autonomous vehicle then reaches the intermediate goal position and reattaches itself to the wagon tongue, i.e., the vehicle follows the simulated vehicle's path and is no longer autonomous. Although, the vehicle acts autonomously while adapting to the environment, the system follows the strategic hierarchy devised by the operator. In such a system, the operator can easily predict the autonomous vehicle's actions, thereby reducing the reaction time of both the operator and the vehicle considerably. When the autonomous vehicle fails to make a decision, the vehicle stops and informs the operator about the obstacle position in state space. The operator may then take over the vehicle control and manually navigate via direct video-based teleoperation. This situation results in a context switch from simulation to real in the operator's environment.

The vehicle adaptation system is comprised of three main components, a risk map generator that provides the interface for creating risk maps based on a priori obstacles and terrain map, a real time path planner that calculates the path based on current obstacle identification, and an update system that executes the path and updates the VR operator station. The following section discusses these components in detail.

Risk map generator

Effective use of a priori representation of the remote environment is one of the prime motivating factors for using virtual reality as an interface tool for teleoperation. This a priori model can be designed based on previous knowledge about the building architecture (if operating indoors) or satellite terrain data (if operating outdoors). In the teleoperation system presented in this thesis, it is assumed that this a priori model will be updated continuously regarding terrain, weather and other information that is accessible at a “near” real time rate. Thus the virtual environment may not be truly real time, but rather a model that resembles the “real environment” as closely as possible. The terrain of the priori model is converted into a 2D map so that it can be used in the real vehicle’s path planning system. The 2D map of the environment is divided into zones depending upon the risk associated with it. For example, a pillar in the middle of a room or a rock in the middle of a desert will be considered high risk zones. Depending upon the caution level of the operation carried out, the nearby regions of such obstacles are classified with varying reduced risk values. In the research, the idea of creating and managing such risk maps is simulated manually using a risk map generator.

A standalone application was developed to create a risk map manually based on the location of a priori obstacles, and other high-risk areas identified by the operator. The output risk map is used as input for the multi-modal VR teleoperation system. In a real world implementation, this map could be output continuously using real time data. To facilitate this research, the risk map generator is a standalone 2D Windows-based application that enables risk assignment of the test environment the haptics lab in the Virtual Reality Applications Center). The zone of the haptics lab that can be tracked using Intersense tracking system [58] is approximately 22 X 11 ft in dimension. This zone is projected in the interface as a large rectangular 2D map sub divided into smaller boxes with a resolution of 20 X 20 cm. Each 20 x 20 cm box is considered to be one risk map pixel. The interface is calibrated to match the original haptics lab dimensions. The user interactively assigns risk to the map with mouse click to convert the individual 2D map pixels into “drive-able” or “un drive-able” depending upon the existence of obstacles in those areas. The scenario can be saved in the form of text file with each risk map pixel carrying a binary value depending upon its “drivability”. This text file is used as a grid by the path planner algorithm to calculate the path of the vehicle when it is acting autonomously.

The interface essentially provides a method to create risk maps for easy testing. The primary research goal of the vehicle adaptation system is to show that the remote vehicle, when provided with substantial intelligence and autonomy, can save itself in difficult scenarios. Since generating a run time risk map grid is not within the scope of this research, the risk map generator is used as a standalone interface and is not integrated with VR system.

Real-time path planner

A path planner system is developed to identify the shortest route possible between the current real vehicle position and future simulated vehicle position. The path planner, which is integrated with the real vehicle’s vision manager and wagon tongue manager, implements an *A** algorithm [Hart 1968] for finding the shortest path. The shortest path calculated is then applied via the wagon tongue algorithm on the real vehicle. The real vehicle while in autonomous mode carries out the entire execution and the operator cannot intervene during this period.

The risk map grid that is created using the risk map generator is provided to the VR operator station. The VR operator station with the help of this risk map grid provides risk map to the real vehicle continuously based on its position obtained from the tracker. When the vision manager on the real vehicle generates an obstacle warning, the real vehicle reads the risk map obtained from the VR station and maps the new obstacle position to the risk map. The path planner uses the real vehicle position to identify the starting position for the algorithm and the last received simulation value identifies the target position for the algorithm. The planner then adds the start and target position to the newly calculated risk map grid.

Robot path planning is one of the basic necessities in robot navigation systems. Most of the autonomous path planning algorithms [Ferguson 2005] use some form of probabilistic algorithm such as the *A** algorithm. Visibility graph and generalized Voronoi diagrams are other common path planning algorithms that are effective in robotic navigation. In the system presented here, the path planner is required only to show the effectiveness of the overall concept of VR teleoperation in mixed autonomy. Thus, the path planning system needs to solve a local problem in which the number of obstacles is pre-determined. Moreover, the system is required to identify reactive path trajectories and not deliberative trajectories, which in turn allows the path planner to use path planning strategies that are simple, faster and computationally inexpensive.

The path identification algorithm proposed by Calisi et al [Calisi 2005] is a derivative of probabilistic road map approach and rapid exploring random tree algorithms. The paper suggests reducing computational expense as the need for their proposed probabilistic algorithm instead of using standard deterministic or geometric based algorithms. However, the target environment used in this research is less complex and contains fewer geometric constraints and hence does not require probabilistic algorithms. Soltani et al [Soltani 2002] conducted a study to determine the potential of deterministic and probabilistic search algorithms in path planning. The study identified that the Dijkstra [Dijkstra 1952] algorithm can find optimal solutions to problems by systematically generating path nodes and testing them against a goal, but it becomes inefficient for large-scale problems. However, the standard A* algorithm can find optimal and near to optimal solutions more efficiently by directing search towards the goal using heuristic functions thereby reducing the time complexity substantially. The paper emphasized that both the Dijkstra and standard A* algorithms are greedy search methods, which arrive at a solution by making a sequence of choices, each of which looks for the best at the time without considering the potential drawbacks of making such a choice. Hence, these algorithms suffer from the curse of dimensionality effect, which limits the Dijkstra's and A*'s operation to small and medium sized problems. However, A* could produce solutions that are locally optimal. The conclusion of the Soltani et al study supports the conclusion that the A* algorithm is sufficient for a simple medium scale path planning system as the required of this application.

The A* algorithm is one of the many heuristic informed search algorithms that helps identify the shortest path between the start and goal positions. The algorithm calculates the effective cost of traversing from one node to another node by not just including the local cost but the total cost $g(n)$ for traversing from the start to that search node. The algorithm assumes a user defined heuristic function $h(n)$ which is an *estimate* of the minimum cost from any node n to the goal. Total cost $g(n)$ is the actual cost for the shortest distance between the current node and previous node. Hence the effective cost function will be $f(n) = g(n) + h(n)$. The heuristic function for the *A** can be calculated in many ways. When $h(n)$ is 0, then cost function $f(n) = g(n)$ and A* becomes Djikstra's algorithm. This algorithm is guaranteed to find the shortest path. However, the algorithm will be much slower as it removes the optimization of *A**. If $h(n)$ is less than or equal to the cost of moving from current node n to the goal, then *A** is guaranteed to find a shortest path. However, this results in increasing the number of nodes that are traversed by the algorithm, thereby making it slower. If $h(n)$ is exactly equal to the cost of moving from current node n to the goal, then *A** will only follow the best path and never expand, making it very fast. If $h(n)$ is greater than the cost of moving from n to the goal, then *A** is not guaranteed to find a shortest path, but it can run faster. The system presented in this thesis utilizes one of the most common methods used for calculating heuristic estimate known as the Manhattan method.

The local cost of traveling from one node to its neighboring node is held constant, denoted K. As the system tries to emulate a simple path planner, the cost of traveling between adjacent nodes is kept the same (K) regardless of whether the nodes share an edge or a corner. Hence $g(n)$ will be the sum of the cost of the current parent node and the value K. As mentioned earlier, the system uses the Manhattan method for calculating $h(n)$. The Manhattan method is a technique that helps calculate the shortest distance between two points in state space. Here the system calculates the total number of squares or zones when moving horizontally and vertically to reach the target zone from the current zone. However, the method ignores diagonal movement. The number of squares is then multiplied by the value *K* to identify the cost which is *h(n)*. The algorithm is initialized with the map/dimension of search area, along with a method to calculate the local/total cost and the heuristic. It is also provided with a start and goal position. The algorithm maintains a queue of nodes to be traversed known as the open set and searches the queue starting from the start position. The algorithm provides high priority to the node that has the lowest *f(x).* At each step of the algorithm, the node with the lowest *f*(*x*) value is removed from the queue, the *f* and *h* values of its neighbors are updated accordingly, and these neighbors are added to the queue. The algorithm continues until a goal node has a lower *f* value than any node in the queue (or until the queue is empty). The *f* value of the goal is then the length of the shortest path, since *h* at the goal is zero in an admissible heuristic.

The real time path planner uses the *A** algorithm to calculate the shortest path for the remote vehicle to travel. However, if no path can be generated between the real position and the target position without avoiding the obstacle or if the obstacle is too close to the vehicle, the real vehicle is brought to halt. The video manager in the real vehicle begins streaming real time video to the VR operator station. The operator is re-positioned so that the simulated state and the current real state corresponds, and then may teleoperate the real vehicle using video feed.

Update system

The path planning algorithm generates the path to be traveled to avoid the obstacle at a given time *t*. This is calculated using the real state and simulated state values again at time

t. The calculated path is stored as way points which are then executed by the vehicle manager using wagon tongue algorithm.

The vehicle stays in autonomous mode until it completes the entire path as calculated. This effectively results in the vehicle reaching an old simulated state position. This phenomenon can be better understood with the following example. Suppose a simulated state *S1* from the VR station is sent to the real vehicle at time *t*. The associated lag is assumed as *x* seconds. The wagon tongue algorithm will project this simulated state *S1* to the time $t1 = t + x$ seconds using the speed and direction to calculate a new simulated state *S2*. When the vehicle identifies an obstacle and becomes autonomous, the path planning algorithm takes the *S2* and *R1* and time *t1* (which could be a negligibly larger than $t + x$) and calculates the path. Assuming that the time taken to travel around the obstacle is *y* seconds, then the vehicle would have reached simulated state *S2* at time $t1 + y$ seconds at which time the actual simulated state could be a different position. The real vehicle informs the teleoperator about the obstacle. Apart from the advantage of having the real state being seen in the VR interface, the operator also receives a visual cue of how far the simulated vehicle is away from the real one via the transparent vehicle blob and colored distance bars. Moreover, the obstacle warning that flashes on the dash board of the operator will enable operator to understand that he/she should slow down and wait for the real vehicle to catch up. Finally the obstacle position and dimensions are displayed in the VR interface and the operator can add them to a priori model if required. The operator is also informed when the real vehicle switches back to teleoperator states.

Fail safe mode

The reliability of the vehicle adaptation system depends on the reliability of the stereo vision system. The stereo vision system computes the disparity map at 2Hz and it requires a minimum of 4 maps to identify an obstacle with confidence. Hence there is a possibility that the stereo vision system may not be able to identify obstacles that are too close to the vehicle. When the vehicle is in autonomous mode the vehicle will create a new path when it identifies the first obstacle and turns autonomous. However, when there is a second obstacle in this newly created path, the vehicle can identify it only when it is sufficiently far from the vehicle camera. Otherwise, there is a possibility that the vehicle will collide with the second obstacle. In such situations, the vehicle adaptation system enables the fail safe mode on the real vehicle. When enabled the vehicle stops and informs the operator about this obstacle allowing the operator to teleoperate the vehicle in video-enhanced mode.

CHAPTER 4. PROTOTYPE IMPLEMENTATION AND EXPERIMENTAL RESULTS

Real Vehicle Platform

The prototype vehicle is built on a toy radio controlled car platform. The vehicle inputs are controlled using phidgets. Phidgets are adapters that convert the digital signal from the onboard computer to analog values for the vehicle motor. The vehicle uses a mini-itx motherboard with a 1.4 GHz processor and 1 GB memory. The stereo vision system is enabled by two Unibrain synchronized Fire-wire cameras that are connected to the onboard computer. The cameras are connected in series so that the images obtained from them are synchronized. The vehicle's on board computer uses Windows XP as the operating system. An on board wireless network adapter enables the real vehicle to communicate with rest of the system using WIFI. The vehicle uses VR juggler's VPR thread libraries [VR Juggler 2008], an open source C++ library for spawning network threads. The stereo vision system uses Intel's OpenCV vision library [Open CV 2007] for image processing. The vehicle and the on board processor are both powered using a DC power supply. The power supply is tethered to a long wire with an AC-DC adapter and connected to the wall. In order to avoid problems due to change in rate of battery discharge over a period of time, the vehicle does not use a battery power supply for power. Lack of battery does not affect the system as the vehicle is operated in a very small indoor test area. The tracked device which is a part of the Intersense tracking system [58] is installed on top of the vehicle. The tracked device enables the tracking system identify vehicle's real time position and orientation. Figure 27a shows the snapshot of the real vehicle platform.

Virtual Reality Platform

The virtual reality station consists of a Redhat Linux machine with a 2.66 GHz processor and 4 GB memory. The VR interface use VRJuggler [VR Juggler 2008] an open source portable C++ library for developing the VR platform. The library helps drive the simulation on a range of display devices including desktop through 6-sided CAVE [43] systems. The interface uses OpenSceneGraph [OpenSceneGraph 2008], an open source high performance 3D graphics toolkit written in Standard C++ and OpenGL graphics programming language for rendering. The 3D VR model of the vehicle environment and the vehicle is built using Maya [Maya3D 2008], a 3D modeling and animation tool. The 3D vehicle simulation is provided with vehicle inputs using Microsoft sidewinder wheel, acceleration and brake pads as shown in Figure 27b. The interface also incorporates keyboard and mouse for user inputs.

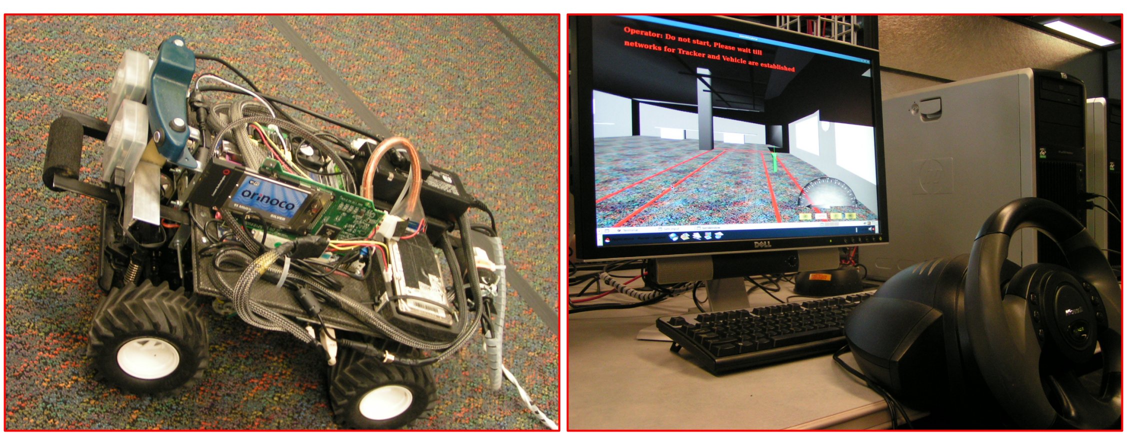

Figure 27 a. Real Vehicle Prototype b. Virtual Reality Station

Experimental results

Stereo vision system

Depth Reliability in static and moving conditions

The vision system's reliability was tested under static as well as moving camera conditions and the results of which are shown in Figure 28. Table 4 presents the depth reliability results for the stereo vision camera. The data was collected in static camera conditions for two different light settings. The results show that the stereo vision system is reliable for identifying small obstacles in indoor conditions. The depth resolution decreases

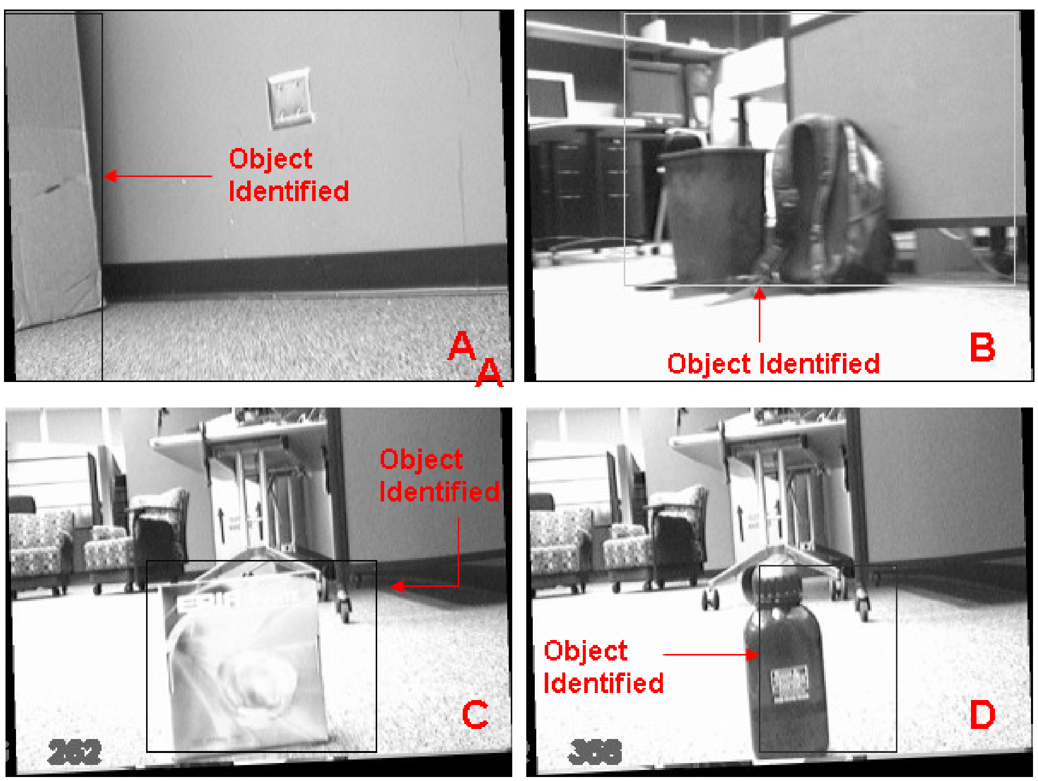

Figure 28. Stereo Vision Results A,B: Moving Camera C,D: Static Camera

with increasing distance between the camera and the object. Since the stereo vision system is affected by the environmental lighting and the camera setup, the resulting depth maps can be noisy.

Obstacle identification and avoidance test

The virtual reality teleoperation presented in the earlier chapter is essentially a fusion of sensors and virtual reality supported by onboard vehicle intelligence. This negates the disadvantages of virtual reality based teleoperation. The system relies on effectively balancing autonomy between human and machine intelligence. The research goal is to create a teleoperated vehicle controlled using a VR interface, that responds to human commands. At the same time the vehicle must adapt to situations in which the human cannot react

Table 4. Reliability Results for Stereo Vision System

Stereo Vision - Static Camera Conditions			
Lighting	**Object type**	**Measured depth (m)**	**Actual depth (m)**
Bright	Small	0.52	0.5
Dull	Small	0.54	0.5
Bright	Large	0.57	0.5
Dull	Large	0.57	0.5
Bright	Small	1.05	1
Dull	Small	1.1	1
Bright	Large	1.06	1
Dull	Large	1.13	1
Bright	Small	1.6	1.5
Dull	Small	1.68	1.5
Bright	Large	1.6	1.5
Dull	Large	1.68	1.5

effectively. A considerable amount of research has focused on balancing autonomy between the human and the agent. However, most of these falls under the category of supervisory control, in which the robot or agent is autonomous and the human operator intervenes only when the agent fails. Wegner [Wegner 2003] suggests a tele-autonomous system in which the operator supervises multiple agents at the same time but intervenes when an agent calls for help. His proposed system attempts to achieve balance in autonomy between machine and human by allowing the machines to perform only low level tasks. Such a system is not applicable for situations that involve high level tasks requiring human control and intelligence. In short the agent should exhibit efficient and useful autonomy within the overall bounds of the instructions given by the user [Lyons 1990]. The results presented in this chapter supports the idea [Kadavasal 2007] that sensor augmentation when coupled with virtual reality provides a more intuitive and "self healing" teleoperation interface [Kadavasal 2009]. In order to demonstrate the claim that balanced autonomy between vehicle intelligence and human operator is achievable; a low level vehicle adaptation experiment using video based teleoperation was conducted. In the experiment, operator and vehicle intelligence are provided with equal amounts of control over the vehicle actions. However, the vehicle intelligence follows the strategic hierarchy devised by the operator.

Significant research work has been conducted on vehicle adaptation and collision avoidance techniques using on board vehicle sensors. Tsalatsanis et al [Tsalatsanis 2007] presents a vision based tracking and collision avoidance system for mobile robots. The robots are autonomous and not under human control. The technical paper on Austin mobile robot presented by Brogdon et al [Brogden 2005] also reports stereo vision and laser based guidance and obstacle avoidance system for autonomous vehicles implemented as a part of

DARPA grand challenge. In contrast, the model presented in this thesis involves sensor augmentation and vehicle adaptation for a vehicle which is teleoperated by a human operator.

Architecture

Figure 29 shows the system architecture for sensor enhanced video based teleoperation. The system is comprised of three major components, namely, a stereo vision based obstacle detection system, vehicle adaptation system and teleoperator station. The functional details of the vision system were explained earlier. The onboard computer has capabilities to transmit video images from the camera to the teleoperator station and to activate a vehicle adaptation algorithm when required. The teleoperator station includes a display that receives the video from the vehicle and a Microsoft sidewinder force-feedback wheel for driving the vehicle.

The teleoperator drives the vehicle using the video feedback received from the vehicle camera. Assuming that the system has a constant lag of *t* seconds, then every

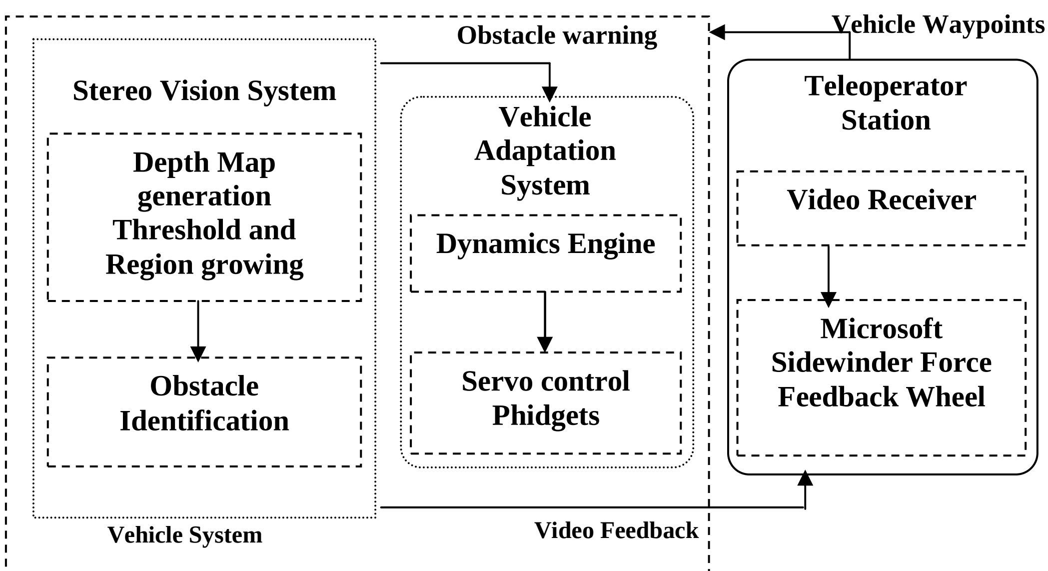

Figure 29. Architecture for Video Based Teleoperation with Vehicle Adaptation

command sent to the vehicle and every frame of video sent to the operator station are received *t* seconds after the actual event. If the vehicle environment is dynamic and there is an obstacle in front of the vehicle, which the vehicle is going to reach in less than 2t seconds, there is a possibility that the operator cannot prevent the vehicle's collision with the obstacle. However, the vehicle's onboard cameras and computation for image processing can be incorporated as the vehicle's "senses".

Vehicle Adaptation - Video Teleoperation

When the stereo vision system issues a warning based on the detected obstacle, the vehicle adaptation system is activated. The dynamics engine calculates the steering angle for the vehicle based on the object's position in the three dimensional state space with respect to the camera. The adaptation system then overrides the operator's commands and turns the vehicle autonomously. The vehicle steers away from the obstacle for a preset distance and stops. The operator is informed about the identified obstacle and continues driving the vehicle based on the video feedback. In this way, the vehicle takes control only in situations where the operator cannot react immediately and retains the attributes of a human teleoperated system.

Experimental Results

The experiment is conducted in a well lit environment, as the stereo vision system is dependent on the environment's light conditions. The video is transmitted from the vehicle camera to the teleoperator station at 620 X 480 resolution with a preset lag. The experimental setup consists of two untextured objects, namely a large cardboard box, and a small container, which act as obstacles for the vehicle. The vehicle adaptation algorithm is preset to activate if the detected object is within 1.5 meters. The vehicle moves at an average speed of 0.5 meters/second and the vision system processes disparity maps at a rate of 3 Hz. As the vehicle cameras are moving, the stereo vision system will experience loss of frames and motion blur. Although the system has a transmission rate of 15 fps in static conditions, the effective frames received by the stereo system are fewer in dynamic conditions. Figure 28 shows the stereo vision system working in both static and moving conditions. Experiments were conducted to quantify the reliability of stereo vision based vehicle adaptation system. In experimental trial runs, the teleoperator tries to drive the vehicle over the obstacle from a preset point. The trial runs are carried out on both the sample objects. The experiments are conducted to collect data according to the following criteria: 1. whether the vehicle adaptation system is activated or not, 2. whether the obstacle is avoided or not. 30 trials were conducted for each obstacle, the results of which are shown in Table 5. The size of the sample size is large (>30) and hence is normally distributed. The results show that the stereo vision based vehicle adaptation system has a 78% average success rate in avoiding an obstacle. The vehicle adaptation system activated with a success rate of 95%. These results indicate that the stereo vision system is sufficiently reliable to support the envisioned general vehicle adaptation system for balanced autonomy. In the few trials that the vehicle adaptation failed, the algorithm did not receive a warning from the stereo vision system for a detected obstacle. This can be attributed to the reduced frame rate transmission experienced by the stereo vision system when moving. A potential solution is to increase the processor capability to handle higher frame rates.

Experiments were also carried out to identify whether stereo vision based vehicle adaptation provides any improvement over video based teleoperation without sensor fusion. A vehicle path was laid out and measurements were marked at intervals of 0.5m. The initial path is curved to ensure that the obstacle placed at the end of the track comes into view of the video camera (or operator) only when the vehicle is traveling. An obstacle is placed on the vehicle track and the distance between the obstacle and the point where track curves is varied between 1m and 4 m in steps of 0.5. The stereo vision based vehicle adaptation system is disabled and the operator is allowed to drive the vehicle based on the video received from the on board camera. The lag is maintained at approximately 1 second per transmission.

The experiment was conducted for 1, 1.5, 2, 2.5, 3 and 4 m and three trials were performed for each distance step. In the next set of experiments, the lag was increased by 5 seconds and vehicle is driven again for each of the distance values. Finally, the stereo vision based vehicle adaptation system is enabled and the experiment was repeated again for a 5 second lag. The results of these experiments are shown in Figure 30.

Table 5. Vehicle Adaptation Experimental Results

Object type	Number of trials	Collision activation status	Obstacle avoidance status
Small	30	Activated 28 times Missed 2 times	Avoided 23 times Collided 7 times
Large	30	Activated 29 times Missed 1 time	Avoided 24 times Collided 6 times

The results clearly indicate that video based teleoperation with a lag of 5 or more seconds is not feasible and will result in loss of situational awareness for the operator. However with vehicle adaptation the operator can navigate the vehicle away from the obstacle with ease. This is true even in cases where the obstacle is present very close to the vehicle. The results support the argument that balancing autonomy between vehicle and human is achievable and can enhance the teleoperation process.

Discussion

The results show that autonomous vehicle adaptation can assist a teleoperator from colliding into near field obstacles [Kadavasal 2008]. The system essentially performs a switch operation from the teleoperator control to autonomous control in

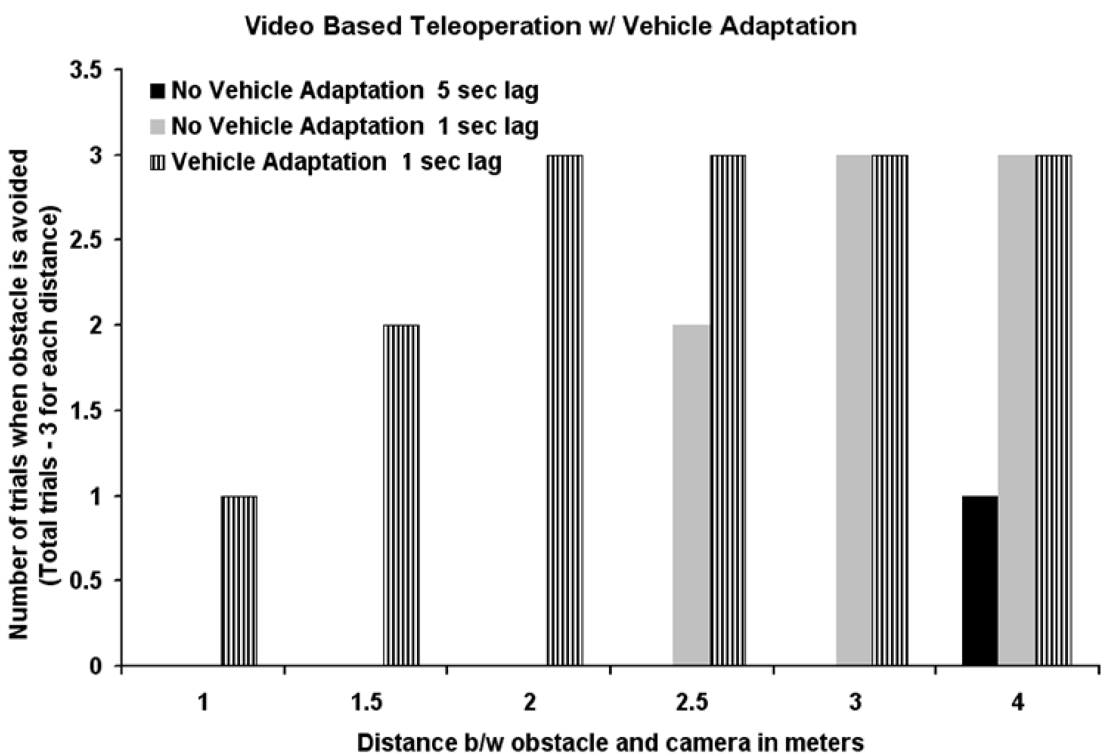

Figure 30. Experimental Comparison

situations where the operator cannot intervene. The vehicle then reestablishes its teleoperated state and follows the operator's commands. These results indicate that balancing autonomy between vehicle and video-based teleoperator station is feasible.

Reliability results for tracking system

Experiments were conducted to check the reliability of the tracking system. The effective tracking area of the system was determined using trigonometry and basic algebra. The accuracy of the system is determined by comparing the actual data from mathematical calculations to the range measurement data from the processor. For the sake of convenience, the circular range areas of the transponders were approximated to rectangles. The maximum reliable tracking area is identified and marked. In this case, the tracked area is 22 X 11 square ft (approx.). The tracking area was then divided into four zones (A, B, C and D). The actual and measured data were obtained by positioning the tracker device at predetermined places within these four zones. The experiment was repeated for 30 trials each in all four zones in order to maintain a normal distribution.

Table 6 shows the measurement error in all four zones. On an average, the tracking system has a maximum error band of ± 0.4 inches or ± 1.0 cm in both X and Y directions. The results show that the system has high resolution and is reliable enough for tracking the remote vehicle.

Overall system reliability

Based on the results presented in the previous sections it can be concluded that the individual components of the presented system are reliable and useful. Table 7 shows the

Table 6. Tracking Reliability Results

Zones	Error X (inches)	Error Y (inches)
A	0.56	0.41
B	0.28	0.48
C	0.32	0.24
D	0.5	0.28
Full tracking area	0.41	0.35

reliability test results for the complete system. The stereo vision based obstacle detection system and the vehicle adaptation system were enabled on the real vehicle station. The communication between the VR station, real vehicle and tracking station were established. The path to be taken by the vehicle was predefined and marked both in the physical and virtual environment. The stereo vision system is set with a predefined obstacle threshold of 1 meter, i.e. the vision system will mark any object it identifies as an obstacle if the object is within 1 m distance from the vehicle. It should be noted that the vehicle takes approximately 0.5 seconds to process the images and identify an obstacle and it takes approximately 0.5 seconds to react to that obstacle and chart a new path. The higher computational time is a limitation experienced mainly due to the choice of hardware. Once the communication is established, the user drives the simulated vehicle within the marked path. The user maintains a healthy distance between the real vehicle and simulated vehicle such that an obstacle can be dropped in between them. An obstacle is placed in front of the vehicle at a predefined position, namely 1 meter from the real vehicle. The vehicle's reaction is observed and tabulated. The test is repeated for distances 1, 1.5 and 2 meters, with three trials each. The

Table 7. Reliability Results - VR based Teleoperation in Mixed Autonomy

Trials	Distance of Obstacle (cms) from vehicle	Action taken at "distance" cms from obstacle	Result	Goal	Comments
1	100	10	Obstacle not avoided	Not Reached	Late activation of adaptation system
2	100	50	Obstacle avoided	Reached	All components worked correctly
3	100	50	Obstacle not avoided	Not Reached	Vision system - error in depth calculation
1	150	50	Obstacle avoided	Reached	All components worked correctly
2	150	50	Obstacle avoided	Reached	All components worked correctly
3	150	60	Obstacle avoided	Reached	All components worked correctly
1	200	100	Obstacle avoided	Reached	All components worked correctly
2	200	30	Obstacle not avoided	Not Reached	Vision system - error in depth calculation
3	200	90	Obstacle avoided	Reached	All components worked correctly

results show that the VR based mixed autonomous system works with 100% reliability when the obstacle is at 1.5 meter distance from the vehicle, although, the preset obstacle threshold for this experiment is at 1 meter. This can be attributed to the higher computational time explained earlier. The vehicle on an average travels at 0.3 meters per second and the 0.5 meter acts as a buffer distance that compensates for the high computational time.

The reliability results from the stereo vision system, the video based vehicle adaptation system, the tracking system together with the results presented for the whole system demonstrate that the system presented in this thesis is realizable. Moreover, it also shows that developed prototype is a reliable system. Although the results show that the system is feasible and reliable, they do not demonstrate actual effectiveness, i.e., is the VR based multimodal teleoperation in mixed autonomy better than other teleoperation interfaces for the defined problem? Chapter 4 presents a series of user studies that were conducted on the teleoperation interface, the results of which support the claim that VR teleoperation in mixed autonomy is a better solution for teleoperating remote vehicles in partially known environments.

CHAPTER 5. VR TELEOPERATION: USER STUDIES

Introduction

The virtual reality based teleoperation in mixed autonomy system strives toward meeting the basic requirements of teleoperation interfaces, namely accommodation of lag, reduction in cognitive work load, intuitive interaction, and easy to train and adapt to. However, it is necessary to understand, from both a quantitative and qualitative perspective, the effect that the interface has on one or more of these interface requirements. In order to evaluate this, a user study was designed and conducted on the interface with a group of users. The first part of this chapter explains user interface design principles and general requirements for their evaluation followed by usability testing and common testing/evaluation methods. The second part of the chapter presents the user study that includes system design and evaluation steps. The final part of the chapter explains the study approval process, the experiment performed, followed by results and discussion.

User Interface Design

User interface simply put is a collection of techniques and mechanisms to interact with something. It is the part of any software or hardware that a user can see, hear, touch talk to or otherwise understand or direct.

User interface design testing

The term usability defines how a product is useful, effective (easy to use), learnable and likable. A product can range from hardware - automatic sofa recliner switch, to software

- graphical command and control interface for unmanned aerial vehicles. User studies and usability testing are studies by researchers or product creators that employ participant representative of the target population to evaluate the degree to which the product meets one or more usability or interface design criteria [Rubin 1994]. Usability testing can be used as a research tool to conduct experiments that range from large sample sizes and complex test designs to informal qualitative studies with a single participant. One of the primary goals of user studies is to improve product usability thereby minimizing the risk of a product in the market place. However, user testing methods come with certain limitations, namely 1. The testing environment can never represent, depict and behave exactly like the actual environment 2. The population subset is never an accurate representation of the actual target population 3. The final results at many times are vague and subjective. Nevertheless, when conducted with care and precision user studies are a very good indicator of both potential problems and potential uses of a product.

In general usability testing involves a set of participants using the interface under a predetermined uniform setup. Based on participant's feedback and researcher's observed data a set of recommendations are then proposed. The proposed recommendations are generally nonbinding and they do not prove or disprove any hypothesis; rather they provide insights for researchers in better understanding their design. Alternatively a system or user interface can be evaluated using set of controlled experiments to help prove or disprove a set of hypothesis or theories. Hence to better understand and evaluate user interfaces it is necessary that a combination of a rigid controlled experiment with informal testing is conducted. Some of the common guidelines in usability testing are, 1. The researchers have to ensure that the participants represents closer to real users, 2. The test design makes the participant do real/actual tasks, 3. A proper observing and recording mechanism is in place for future analysis.

Evaluation criteria

Ravden et al [Ravden 1989] proposed a nine point evaluation criteria check list which can be used for conducting and evaluating user studies. The proposed evaluation check list is an assimilation of usability guidelines and checklists proposed by Clegg et al [Clegg 988], Smith et al [Smith 1986], Gardner et al [Gardner 1987] and Shneiderman et al [Shneiderman 1987]. The nine criteria proposed in the check list are as follows,

1. *Visual clarity*: This criterion helps researcher evaluate issues like data representation or misinterpretation, scene clutter, user attention and data complexity.

2. *Consistency*: The consistency of the interface helps evaluate how quick and easy it is to learn and adapt to the interface. It also helps the researcher understand response times and short term memory load.

3. *Compatibility*: This criterion determines whether the proposed product's interface adheres and conforms to existing user expectations on similar interfaces.

4. *Informative Feedback*: The interface should perform such that the users are provided with sufficient and timely feedback regarding where they are in the system, what actions they have performed, and what are the outcomes of those actions.

5. *Explicitness*: A user interface can be considered explicit if the interface is transparent and is able to help the user develop a clear understanding of his/her actions, the corresponding interface reactions, and how it relates to the application tasks. This helps reduce learning time and improves user confidence.

6. *Appropriate functionality*: This criterion denotes those forms or representations that are required to carry out a specific application task. For example, if the user task involves traveling inside a virtual world from point A to point B, and then select an object at point B, the interface should have appropriate functional requirements to carry out such tasks. The functional requirements in this example could be a selection button, or selection by click, and presence of an object, etc.

7. *Flexibility and control*: The user interface should provide flexibility in the way the information is presented to the user and how it can be handled. This in turn allows the user to have control over those tasks.

8. *Error prevention and correction*: The system should be designed to detect user errors and minimize them when they occur.

9. *User guidance and support*: The interface should provide user with relevant information, guidance and support during the entire period of time when the application task is performed.

Evaluation/Testing

This section describes some of the widely used methodologies for testing. Most testing methodologies differ on the emphasis given to quantitative versus qualitative measures or to the amount of interaction. Moreover, it should be noted that a single methodology can be applied in multiple ways depending on the point of the product cycle at which the user study is conducted. This is described further in the following sections. The later part of this section explains some of the commonly used testing techniques that are employed as a part of testing methodologies.

Rubin et al [Rubin 1994] proposes four basic types of testing methodologies namely, exploratory test, validation test, comparison test, and assessment test.

1. *Exploratory test*: As the name suggests, exploratory tests are primarily used to allow user to explore the basic features of the interface. The test does not provide the user with any specific task to perform or goal to achieve, but rather allows the user to identify, choose and use the various interactions available in the interface. Such a testing methodology can be used to evaluate progress made at various stages of a product cycle. The method can be used to train or help participants learn the test interface before asking them to participate in task-based studies. Moreover, exploratory tests allow researchers to employ informal qualitative testing techniques such as the "Think aloud" technique [Ericsson 1984], in wihch the participant is asked to think aloud during the course of the test. This technique is mainly used to understand the thought process of the participant when he/she is using the interface.

2. *Assessment test*: In this test, the user performs actual tasks rather than simply exploring the interface. The test has less emphasis to the thought process and more on actual user behavior. The tasks given are realistic tasks that can help the researcher identify actual usability deficiencies in the product. The tests include collecting quantitative data as well.

3. *Validation test*: Validation tests are performed on a product at the end of a product cycle. The results obtained from such tests are compared to standard usability benchmarks for better understanding. The tests measure both quantitative and qualitative parameters and test all possible interactions in the product.

4. *Comparison test*: When compared to all the above test methodologies the comparison test is a not a mutually exclusive test method. This test method helps compare the interface to

other contemporary interface styles and interaction methods. However, the test is implemented as a part of exploratory test, assessment test, or validation test.

User study

One of the ways the virtual reality teleoperation interface presented here could be evaluated is employing a usability study as explained in the previous sections. Many robotic and teleoperation interfaces been subject to usability studies to understand their proposed interfaces. For example, Endo et al [Endo 2004] conducted a usability study to evaluate a graphical robot task planner interface. The paper uses assessment based tests that are conducted at the tail end of the design [Beyer 1998] to understand the interface. The paper employs a statistical hypothesis to understand the measured data. The usability study conducted by Marble et al [Marble 2003] utilizes a simple task-based validation test to evaluate their mixed initiative robotic system. Their proposed mixed initiative system is a semi autonomous system and has parameters to be evaluated similar to those of the current mixed autonomous system. However, the usability study carried out in Marble et al [Marble 2003] is a binary test rather than a user study and the researchers did not make an attempt to qualitatively or quantitatively understand the system. Nevertheless usability studies [Genov 2009] conducted in teleoperation systems [Lunenburger 2007] have shown that user tests are one among the best ways to evaluate graphical interfaces. The following sections will discuss the test interface, the parameters to be evaluated, the techniques applied to evaluate them, and the usability test scenarios.

Some of the basic requirements for better teleoperation interfaces are accommodating lag, reducing short term memory load, ease of learning and, providing informative feedback.

The virtual reality based teleoperation in mixed autonomy system described in this thesis, strives towards improving existing teleoperation interfaces for the above mentioned requirements. Virtual reality, as an alternative to video as a feedback, can accommodate a considerable amount of lag and at the same time the proposed mixed autonomy can assist the user in not worrying about unknown obstacles in the path of the vehicle, thereby reducing a considerable amount of short term memory load. Moreover, since it behaves consistently, the proposed teleoperation interface can potentially provide a better user interface by giving informative feedback to the user about the actual environment.

To evaluate the proposed teleoperation interface, a combinatorial user study was devised which includes gathering qualitative data from the user regarding the interface and at the same time collecting quantitative data regarding how the user behaved and performed in various goal-based scenarios. The following section presents the system used for the user study along with the user study design, the user study questionnaire and other study materials.

System description

The system consists of three major components namely the virtual reality station, the real vehicle and tracker station. The virtual reality station presents the teleoperator interface to the user. The interface includes the 3D model of the actual driving environment and a simulated 3D vehicle which acts as the representation for the real vehicle. The 3D vehicle model in the VR environment can be driven by the user. The real vehicle receives its own position from the tracking station and marks it as initial position. It receives the simulated vehicle's position and marks it as goal position. The vehicle computes a path and generates the necessary throttle and vehicle steer input to reach the goal.

The real vehicle's projected location is shown to the user in the VR environment as a ghost vehicle. The speed of the simulated vehicle is displayed in the form of a speedometer within the heads-up-display. A distance bar is presented to quantify the wagon tongue distance between the simulated vehicle and the real vehicle. The real vehicle has a stereo vision based obstacle identification system which can be enabled or disabled depending upon the task at hand. When enabled, the vehicle automatically activates a path planning system that helps it calculate a suitable path around an obstacle. The real vehicle has the ability to stream real time video to the virtual reality station. The user can bring up the real time video feed when it is required.

The combinatorial user study devised for this research identifies and evaluates three major parameters of the proposed interface, namely, 1. Operator awareness, 2. Interface design, and, 3. Operator adaptability.

Operator awareness

The mental model created by the user based on his past and present information at any given time can be defined as the situational awareness of the user. An interface is considered to assist the user with better situational awareness when it can provide sufficient information for the user to make his/her decisions at any given time. With video based teleoperation interfaces, the clarity and usefulness of the mental model created by the user depends more on the capability of the user's short term memory rather than on the interface's effectiveness. In the VR based mixed autonomous teleoperation, the researcher expects virtual reality to reduce the short term memory load. The 3D model of the environment coupled with wider FOV can improve the operator's situational awareness considerably.

Interface design

The virtual reality based teleoperation interface proposes intuitive methods to represent real time data in the VR interface, for better understanding of the user. This includes, a ghost representation of the real vehicle, a transparent vehicle blob that grows and shrinks depending on the distance of the wagon tongue, a distance bar that quantifies the gap in the wagon tongue, an input system with haptic force-feedback that simulates a real steering wheel and acceleration pad, multiple camera views that can be controlled on the fly, obstacle warnings and operator instructions. Although, all three features, distance bar, transparent blob, and ghost vehicle are representations of real state information in different forms, they convey the real time data in different ways. The distance bar is closer to a quantitative representation of the actual distance between real vehicle and simulated vehicle. When the operator is driving the vehicle with a zoomed out view, he/she may not be able to decipher the distance between the vehicles from the actual vehicle representations. The distance bar helps the operator in such situations. The ghost vehicle is the actual representation of the real vehicle in VR. The vehicle shows the exact path traced by the real vehicle in the actual environment. It helps the user by providing a visual cue when the real vehicle detaches from the wagon tongue and becomes autonomous. The transparent blob around the simulated vehicle grows and shrinks depending on the distance between it and the real vehicle. Moreover, the blob orients itself to the same direction as that of the real vehicle. When the user is in chase camera view or has driven far away from the real vehicle, he/she will experience difficulty determining the orientation of the real vehicle. The transparent blob can assist the user in such situations. The usability study is designed in part to evaluate and understand the effect of these tools on the user when teleoperating a vehicle. The first scenario of the usability study is designed to evaluate operator awareness and interface design.

Operator adaptability

In the mixed autonomous system presented here, the vehicle shifts into autonomous mode when encountering an obstacle within a threshold distance. The scenario involves, human operator sharing/giving up autonomy of the vehicle for brief amounts of time. It is important that the system and interface design enables a seamless transition of autonomy between the human in the loop and the vehicle. This can be achieved by providing the user with real time information about the real vehicle along with driving suggestions and observations. The second and third scenarios of the usability study are designed to evaluate operator adaptability.

Design

The user study designed for this research is based on a combinatorial approach where both quantifiable and subjective data are gathered from the user and evaluated. The study consists of three scenarios namely, virtual reality teleoperation, video based virtual reality teleoperation, and virtual reality teleoperation in mixed autonomy. Each scenario requires participants take part in a task/goal based validation test followed by a survey. The validation test allows the gathering of quantifiable data, which will be further analyzed. The survey method provides user opinions and user understanding of the interface. The study also employs a "think aloud" technique, in which the participant is asked to explain his actions as he/she works through the task. The "think aloud" technique is also employed along with the survey method to better understand user ratings. For example, if the participant rates the difficulty level of the interface to a quantity in the range of 1 – 10, the they are asked to use the "think aloud" technique to describe what aspects of the interface made them select that rating. The study is recorded in two different ways. All quantitative data is automatically stored in the form of a text file when the participant carries out the required tasks. The qualitative data obtained during the "think aloud" process is written in the survey document along with participant comments and inputs.

User Scenarios

Scenario 1: Scenario 1 presents a basic virtual reality teleoperation system where the participant is allowed to control the real vehicle from the virtual reality station. The stereo vision based obstacle identification system and the path planning system are disabled on the real vehicle. The virtual reality model shown on the user's screen consists of a 3D model of the research lab and the simulated vehicle is synchronized to the real vehicle position at the start of the test. The virtual environment and real environment are identical. The Intersense tracking is available for an area of 22 X 11 ft. This tracking area is marked in the 3D model for the benefit of the user. The participant is asked to drive the simulated vehicle from one end to the other end of the tracking area lengthwise. A snapshot of the interface is shown in Figure 31. The task does not involve any time based goals or any specific path to be taken. The goals of this case scenario in the user study are to evaluate operator's situational awareness and participant's comfort with the proposed interface design. This is done through the “think aloud’ technique and the survey which the participant will complete at the end of this scenario. Although the the quantitative data obtained during this scenario are stored, it is not used for further analysis. Hence, scenario 1 can be used as a learning tool for the participant and helps him/her get accustomed to the interface and gain knowledge about interface controls.

Scenario 2: Most of the current virtual reality based teleoperation interfaces includes a live video feed to provide the teleoperator with real time data. However, the real time video feed is lagged and is not synchronized with the system position at any instant in time. The resulting system negates the advantages of virtual reality as a tool for teleoperation and turns the operation video based. In this user study, video based virtual reality teleoperation is used as bench mark data to compare and understand the effects of the proposed mixed autonomous VR system. In this scenario, the participant is provided with live video feed from the real vehicle. The video feed shows the actual environment and is lagged by one second. The

Figure 31. Snapshot of VR Interface in Scenario 1

participant is provided with a starting position and a goal position as shown in Figure 32. The task in this scenario again involves driving the real vehicle using the virtual reality system. However, due to lag, the real environment and virtual environment are not spatially registered. When the participant starts driving the vehicle, an obstacle (object) is placed at approximately 1.5 m distance from the real vehicle. The task for the participant is to use the lagged video feed and the position of the real vehicle shown as a ghost in the VR interface, to drive the real vehicle around the obstacle and reach his/her goal position. It should be noted that the object is placed in front of the vehicle only after the participant has started performing the task. The maximum speed with which the simulated vehicle can be driven is 0.3 meters/second. Hence the user has at least 3 seconds to see the obstacle and react to it. The quantitative data stored in this study are the path of the simulated vehicle along with time intervals, the path of the real vehicle along with time intervals, the obstacle position and the task validation i.e., whether the participant cleared the obstacle or not. The participant

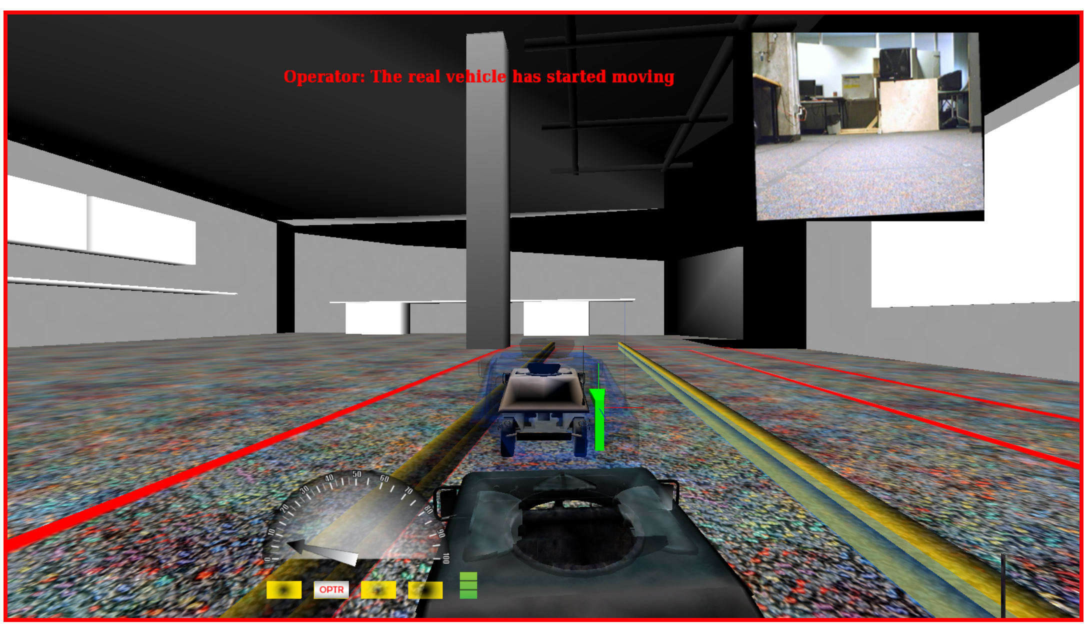

Figure 32. Snapshot of VR Interface in Scenario 2 and 3

is provided with two attempts to clear the obstacle. If the participant clears the obstacle in the first attempt he/she moves to the next scenario. Finally, the participant completes a written survey explaining his/her decisions and actions. Although the scenario does not contribute any substantial information towards the concept of mixed autonomous teleoperation on its own, nevertheless it acts as a yardstick for current teleoperation interfaces that are subject to latency problems. Moreover, it helps participants understand the proposed virtual reality teleoperation with mixed autonomy in context with other teleoperation interfaces.

Scenario 3: This scenario evaluates the proposed virtual reality teleoperation in mixed autonomy. The virtual reality interface is similar to the interface presented in scenario 2. Again, the virtual reality model and real environment are not identical, and a new obstacle is added to the real environment once the participant starts executing the task. However, in this scenario, the stereo vision based obstacle identification system and the path planning system is enabled on the real vehicle. When the real vehicle identifies the obstacle, it prompts a warning on the virtual reality interface, informing the operator about the obstacle, and the shift in autonomy. The operator can also notice the vehicle's new path with the help of the ghost representation. The operator is able to see the lagged video feed to understand the vehicle actions and obstacle location. The stereo vision system requires the obstacle to be seen at least a meter away to identify and create a path around it. In order for the vehicle to autonomously navigate around the obstacle, the goal position (simulated vehicle position) must be beyond the obstacle. To simulate this scenario, the user is asked to drive the simulated vehicle and reach an intermediate goal position and then allow the real vehicle to catch up with the simulated vehicle. This allows a gap of around 2 meters to be created between the simulated vehicle and real vehicle. When the real vehicle starts driving, the obstacle is dropped at 1.5 meter from the vehicle and the intermediate goal position is now a meter beyond the obstacle. The participant gets a regular feed of warnings, vehicle position and other information. When the participant feels that the vehicle has automatically avoided the obstacle and reached the intermediate goal position he/she continue driving the vehicle to the final goal position. Record data includes the simulated vehicle path, real vehicle path and the real vehicle's re-routed path. The participant is then asked to complete a survey at the end of the scenario.

Institutional Review Board

As the usability study involves human subjects, an approval by the ethical review board, otherwise known as the institutional review board, is necessary. The basic steps for obtaining the approval are the following,

1. Complete protecting human research participants training designed by the National Institutes of Health and obtain the certificate of training.
2. Complete the human subjects review form which provides details of the study including design, participant details, researcher details and qualifications, purpose of the study, benefits and risks associated, etc.
3. Complete the informed consent document template. An informed consent document is the statement which is provided to the user explaining the user study, the design, user rights and benefits. The document will have to be read and signed by the subjects and the researcher before the start of the study.

4. Provide the institutional review board with all materials that will be used during the study. This includes the survey questionnaire, research tables, forms and sheets, data recording mechanism, etc.

The necessary permission for this study was obtained from Iowa State University office of research assurances and institutional review board. The informed consent document template and recruitment letter that were used for this user study are presented in the appendix of this thesis.

Experiments

The user study conducted involves using haptic force-feedback devices like the sidewinder steering wheel, brake and acceleration pads and interacting with a virtual reality based game-like interface. Moreover, the entire system requires reasonable knowledge on vehicle control, network communication and 3D interaction. In order to make sure that the participants have the relevant experience, the target group was restricted to students at Iowa State University who have prior knowledge with computers and computer games. The following sections presents in detail the experiments conducted followed by results, analysis and discussion.

The experiments were conducted at the Haptics laboratory at the Virtual Reality Applications Center, Iowa State University. The test setup used here is described in chapter 4 of this thesis. The recruited participants were told about their rights and benefits. They were instructed in detail about the system and tasks to be performed. The participants were asked to carry out the tasks scenario 1, scenario 2 and scenario 3 in that order. In all 12 participants were studied and their results recorded and analyzed. The following sections present both the quantitative and qualitative data collected from the user study.

Results and discussion

The goal of this user study is to evaluate three main parameters in the proposed teleoperation interface. They are operator awareness, interface design and operator adaptability. Scenario 1 is designed to helps understand and evaluate the interface design with respect to situational awareness. Scenario 2 and 3 are designed to evaluate operator adaptability when teleoperating the remote vehicle through a partially known environment. Moreover scenario 2 and 3 uses the same interface framework as scenario 1 which will provide additional data to further evaluate operator awareness and interface design.

Data Collection

Table 8 presents the qualitative data collected from Scenario 1, 2 and 3. The data summarized in this table is based on the information provided by the participants during the survey and from the information collected by the researcher based on the "think aloud" technique. This includes user profile and user interface feedback for all three scenarios and the outcome of individual validation tests performed by the user.

The quantitative data collected from the validation tests includes the x and y position of the real and simulated vehicle within the tracking area for every time instant dt, where dt is 1 second. It should be noted that this data is stored in the virtual reality station as it is necessary that the time is synchronized between simulated and real vehicles. However, there exists a lag between the tracking station which feeds the real data and the virtual reality station which stores them. The data recorder takes into account this lag by projecting the real vehicle data to the current time. Also, the data recorder on the real vehicle keeps track of the path generated by the path planner, obstacle position and the risk map generated for all participants.

Analysis

14 participants were identified with the required user profile, who were willing to participate in this user study. However, two of the participants could not complete the study due to personal reasons, which resulted in a total of 12 data sets. The survey questions and answers along with the data from the "think aloud" can be broadly classified into three main categories, namely, general user interface analysis, user interface analysis when doing video based obstacle avoidance and user interface analysis when using mixed autonomy for obstacle avoidance. The participant's opinion on the proposed user interface can be further classified based on interface features and characteristics. The features include the use of virtual reality, presence of interface features like multiple camera views, distance bar for showing the gap between the real and simulated vehicles, transparent blob around the simulated vehicle representing the gap as well as orientation of the real vehicle, and ghost vehicle showing where the real vehicle is currently present. The characteristics of the interface include predictability of the interface, responsiveness, situational awareness and difficulty level. As the interface characteristics differ for every scenario, they are repeated for all three categories. The information presented above can help evaluate and understand operator awareness, adaptability and interface design.

Operator awareness: The mental model of the environment created by the participants based on what he/she has seen in the past and what he/she is seeing currently for every instant of time can be described as situational awareness or operator awareness. The common term that can be used to describe situational awareness is “environment knowledge”. The following questions provide data for understanding participant’s situational awareness.

Rate in the scale of 1-10, 1 being poor and 10 being excellent, how confident you were in each step that the real vehicle is driving and following your user inputs.

Rate in the scale of 1-10, 1 being poor and 10 being excellent, whether you were able to get the environment knowledge as quick as possible at every instant.

All the participants listed in Table 4.1 rated their situational awareness to be high in scenario 1. Based on conversations with them participants were confident about where they were in the environment, due to the presence of the 3D model in front of them. Moreover, the 3D model reduces the short term memory load significantly as participants were confident that if they want to know more about the environment they can always switch views and use the virtual camera to travel around. However, participant lost substantial environment knowledge when using video as a means to avoid the obstacle. The participant still has the 3D model for providing the larger picture. However, he/she had difficulties in figuring out the distance between the vehicle and obstacle based on the 2D video billboard resulting in underestimation or overestimation of depth, thereby losing situational awareness. In scenario 3, the vehicle shares autonomy with the user and avoids the obstacle on its own. This reduces the short term work load on the operator thereby allowing him/her to concentrate on the actual teleoperation task. The presence of ghost vehicle showing the rerouted path along with text warnings and live video feedback provides enough information for the operator about the actual environment and the real vehicle decisions. The operator situational awareness in scenario 3 increases substantially as shown in the tabulated results.

Interface design: The teleoperator interface is evaluated based on the design features and interface characteristics. The design features include distance bar, the transparent blob and the ghost vehicle. The interface characteristics are comprised of responsiveness, predictability and the difficulty level. Participants were told about the features and were provided with an opportunity to explore them during the practice test and during the validation tests. Table 8 lists user preferences on these features. It can be seen that all features are equally used and accepted by the participants. Here are a few excerpts from user comments on these design features,

"I primarily gauged the distance by the distance bar where it seemed anything under 4 bars worked well."

"Maneuvering the vehicle was pretty easy. The visual representation of the sim and the real vehicle was easy enough to follow"

However, participant preferences are not always in the same order. This could be attributed to the fact that, although the tasks and interface remains the same for all participants, nevertheless each participant observed and experienced the situation differently and executed their respective tasks in their own way. The observations show that the proposed design features are helpful and significantly improves operator situational awareness. Also, the participants had suggestions to improve the design further. Some excerpts are as follows,

"The real vehicle representation was by far the most useful. I wish the ghost vehicle moved more smoothly even if it where just interpolated."

"It's fairly easy to control. I wish the ghost vehicle was more transparent"

The following questions helped gauge user understanding of interface characteristics.

Rate in the scale of 1-10 1 being poor and 10 being excellent, did performing an operation lead to a predictable result, that is was the interface responsive enough for the actions you performed

In the scale 1- 10 1 being poor and 10 being excellent, circle a number that reflects your experience appropriately in using this system so far in this experiment

a. Frustrating 1 2 3 4 5 6 7 8 9 10 Satisfying

b. Difficult 1 2 3 4 5 6 7 8 9 10 Easy

Based on the results tabulated it can be concluded that the majority of users felt the interface to be responsive enough and were able to anticipate the interface reactions easily. Almost all users felt the virtual reality interfaces in scenario 1 and scenario 3 had very low difficulty level and the experience satisfying. Participants found the multiple camera views to be very useful. Having a moving camera that follows the vehicle from behind helps simulate actual driving conditions for the participants thereby improving the teleoperating experience. It helped them learn the surroundings without being affected by the lagged video feedback.

Operator Adaptability: The proposed virtual reality based teleoperation in mixed autonomy is unique due to the presence of temporary shifts in autonomy between the actual vehicle and teleoperator and vice versa. The resulting system requires a user interface that has the capability to provide seamless transition to the user during these shifts. In the absence of any intuitive real time indicators, there exists a maximum probability that the user might not know who is in control of the vehicle at that instant of time. This could result in loss of situational awareness for the operator and may eventually result in loss of the vehicle as well.

The proposed interface strives to achieve this seamless shift in autonomy by carefully integrating text messages and warnings along with visual cues like the ghost vehicle and video feedback to best convey the presence of an obstacle and shift in control. Scenario 3 is designed to evaluate this characteristic of the proposed interface. The participants were asked to evaluate the situational awareness they experienced, along with predictability, responsiveness and difficulty level of the task at hand. Some of the user comments are as follows,

> *"It gives me comfort knowing that real vehicle can act with some degree of autonomy. It makes the lag not feel so dangerous."*
>
> *"I can see the virtual vehicle and real vehicle with separation. I like the idea."*
>
> *"The visual cues in the form of video feedback was appealing"*
>
> *"The video was helpful in understanding what was going on. The vehicle performed well and it was easy to follow"*

From Table 8 it can be seen that the majority of the participants rated the interface to be highly responsive and predictable. Also, participants experienced very high situational awareness and rated their experience and comfort level to be high and satisfying.

Video versus Mixed Autonomy: The quantitative data recorded in scenario 2 and 3 includes position coordinates of the real and simulated vehicle for every second. This allows comparison of the distance maintained between the simulated and real vehicles at any instant of time as well as the total time taken to complete the task at hand. In order to compare the scenarios, both video based obstacle maneuvers and autonomous obstacle maneuvers have to be executed identically. This includes having the same starting position, goal position and obstacle position. The researcher ensured both of these tasks are executed identically for all participants. Figure 33 shows a box plot for the tests scenario 2 and scenario 3 against the time taken to complete the tasks, where scenario 2 is video based obstacle maneuver and scenario 3 is autonomous obstacle maneuver.

The box plot represents the data collected in the form of a box, where the box shows the data that falls with 2 inter-quartile ranges. The top most line of the box indicates the value of the corresponding axis at 75th percentile and bottom most line of the box indicates the value of the corresponding axis at 25th percentile. The middle line denotes the value at 50th percentile. The lower and upper most whiskers are called the outliers. Any data observation which lies more than 1.5 times the quartile range lower than the first quartile or 1.5 times the

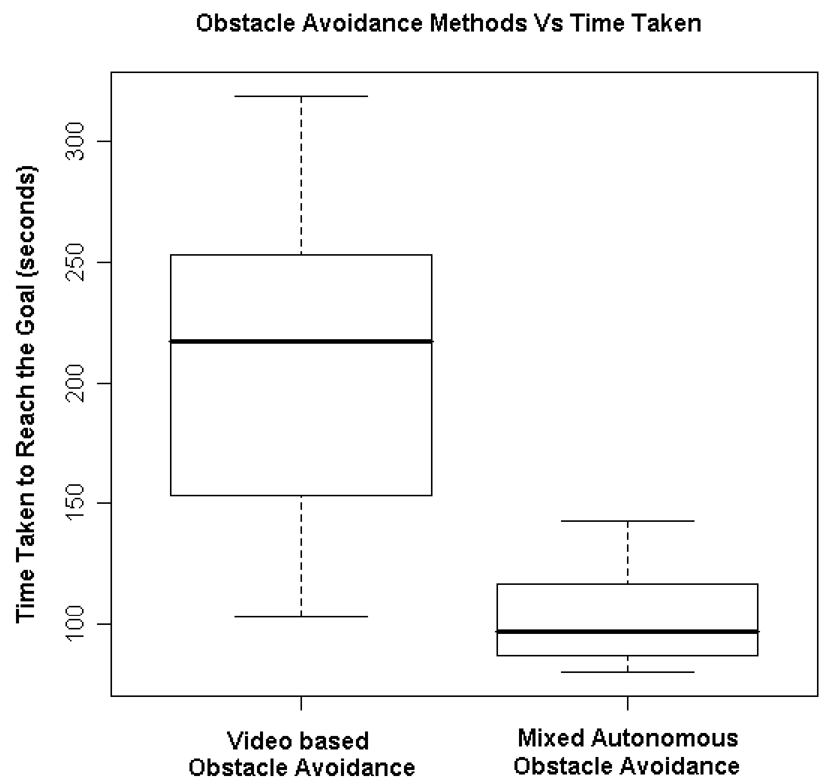

Figure 33. Box Plot – Type of Obstacle Avoidance Vs Time Taken

Table 8. User Studies - Qualitative Results from User Comments and Researcher Observations

	Profile		User Interface									Using Video						Using Mixed Autonomy					
User Number	Gamer	Engineering/Science	Virtual Reality	Multiple Camera views	Ghost vehicle	Transparent Vehicle	Distance Bar	Predicatbility	Responsiveness	Situational Awareness	Difficulty Level	Predictability	Situational Awareness	Responsiveness	Difficulty Level	Outcome - Attempt 1	Outcome - Attempt 2	Predictability	Situational Awareness	Responsiveness	Difficulty Level	Outcome	Choice
1	✗	✓	Useful	Not Useful	Less Useful	Mod. Useful	Highly Useful	High	High	High	Low	High	High	High	Mod.	✗	✓	High	High	High	Low	✓	Autonomy
2	✗	✓	Useful	Useful	Highly Useful	Less Useful	Mod. Useful	High	High	High	Low	High	High	High	Low	✓	-	High	High	High	Low	✓	Autonomy
3	✗	✓	Useful	Useful	Mod. Useful	Less Useful	Highly Useful	High	High	High	Low	High	High	High	Low	✓	-	High	High	Mod.	Low	✓	Autonomy
4	✗	✓	Useful	Useful	Highly Useful	Mod. Useful	Less Useful	High	High	High	Low	High	Mod.	High	Mod.	✓	-	High	High	High	Low	✓	Autonomy
5	✗	✓	Useful	Useful	Highly Useful	Less Useful	Mod. Useful	High	High	High	Low	Mod.	Mod.	High	Mod.	✓	-	High	High	High	Low	✓	Autonomy
6	✗	✓	Useful	Didn't Use	Highly Useful	Didn't Use	Mod. Useful	High	High	High	Low	High	Mod.	High	Low	✓	-	High	High	High	Low	✓	Autonomy
7	✓	✓	Useful	Useful	Didn't User	Didn't Use	Highly Useful	High	High	High	Low	High	Low	High	High	✓	-	Mod.	High	High	Low	✓	Autonomy
8	✗	✓	Useful	Useful	Mod. Useful	Less Useful	Highly Useful	High	High	High	Low	High	Low	High	Low	✗	✓	High	High	High	Low	✓	Both
9	✓	✓	Useful	Didn't Use	Highly Useful	Mod. Useful	Less Useful	High	High	High	Low	Low	Low	High	High	✗	✗	Low	High	High	Mod.	✓	Autonomy
10	✓	✓	Useful	Useful	Mod. Useful	Highly Useful	Mod. Useful	Mod.	Mod.	High	Low	Low	Low	High	High	✗	✓	Low	High	High	Low	✓	Autonomy
11	✓	✓	Useful	Useful	Highly Useful	Less Useful	Mod. Useful	High	High	High	Low	Mod.	Low	High	Low	✗	✓	High	High	High	Low	✓	Both
12	✓	✓	Useful	Useful	Mod. Useful	Highly Useful	Less Useful	High	High	High	Low	Low	Low	High	High	✗	✓	High	High	High	High	✓	Autonomy

Key: Mod. - Moderate

quartile range higher than the third quartile is considered an outlier. A box plot chosen to represent the data can be useful to display differences between populations without making any assumptions of the underlying statistical distribution, i.e., they are non-parametric. The spacing between the different parts of the box helps indicate the degree of dispersion (spread) and skewness in the data, and identify outliers.

The box plot shown in Figure 34 for video based obstacle avoidance shows that the 25th, 50th and 75th quartiles are 152 seconds, 220 seconds and 252 seconds respectively. The outliers are approximately 100 and 340 seconds. This indicates that the data lies closer to 220 seconds indicating that on a median scale it takes approximately 220 seconds to use video as a tool to avoid the obstacle and reach the goal in the given set up. In the box plot for mixed autonomous obstacle avoidance, the 25th, 50th and 75th quartiles are 50 seconds, 90 seconds and 120 seconds respectively, with outliers of approximately 40 and 140 seconds. This indicates that the data lies closer to 90 seconds, or on a median scale, it takes approximately 90 seconds to use autonomy as a tool to avoid the obstacle and reach the goal in the given set up. Based on the differences between the two box plots it can be concluded that it takes approximately double the time to use video as obstacle maneuver interface tool when compared to using autonomy for obstacle maneuver. This shows that the mixed autonomous teleoperation system has better performance when compared to video based virtual reality teleoperation. The data is further analyzed using a statistical T test.

A "T" test [Casella 2005] is a statistical hypothesis test in which the test statistic follows a "*T*" distribution. A distribution is considered "*T*", when the mean of the sample is assumed to be normally distributed but the sample sizes are so small that the distribution may not necessarily be normal. The calculated mean and standard deviation for this sample size may deviate from the "real" sample mean and deviation if there is a chance to collect data from a larger sample size. Hence, any conclusions of the *T* test are confined to that sample size and cannot be extrapolated. The assumptions for a *T* test are as follows,

1. The distribution of data is considered normal.
2. The samples can have equal or unequal variances
3. The samples can be independent or dependent, where independent samples come from randomly selected groups and dependent samples come from two groups matched on a single variable or when single group is tested twice.

A standard *T* test procedure involves defining a test statistic. This includes identifying whether the two samples to compare are of equal size or not, equal variance or not and whether they are independent or dependent. Based on this, the relevant test method is selected and applied. A null hypothesis is defined and depending upon the *T* test outcome the hypothesis is either rejected or not rejected. The null hypothesis, denoted by H_0, is a variable that describes the statistical behavior of a data set. The final output of any statistical test like the *T* test will provide results which may either contradict or not contradict the null hypothesis. If the result contradicts the null hypothesis, the hypothesis is rejected. However, if the result does not contradict the null hypothesis it does not prove the null hypothesis is correct. In other words a null hypothesis can always be rejected or not rejected, but can never be accepted. Failing to reject the null hypothesis gives no strong reason to change decisions predicated on its truth. Moreover, it also allows for the possibility of obtaining further data and then re-examining the same hypothesis. The study used the R [Project R 2009] software for computing the *T* tests.

When comparing the sample groups for scenario 2 and scenario 3 sample is clearly small, equally sized and independent. The null hypothesis H_0 is "*there is no significant difference in outcome when using video or autonomy for obstacle avoidance based on time taken on each attempt*". The data set is assumed to have unequal variances. The type of T test used for evaluating such data is called the Welch T test [Casella 2005]. The statistic t in the Welch t test is defined as follows,

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}}} \qquad DF = \frac{\left(\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}\right)^2}{\frac{\left(\frac{s_1^2}{N_1}\right)^2}{N_1 - 1} + \frac{\left(\frac{s_2^2}{N_2}\right)^2}{N_2 - 1}}$$

Equation 14

DF is Degrees of freedom
$\bar{X}_i$ is sample mean,
s_i^2 is sample variance,
N_i is sample size

The results of the Welch *T* test for the time taken in video based obstacle avoidance and mixed autonomous obstacle avoidance is listed in Table 9. The *T* test results produces a p-value of 0.0006368 and the sample mean for time taken using video is 202.9 seconds and time taken using autonomy is 103 seconds with a level of significance value of 0.05. As the p value is very small and falls within the level of significance the null hypothesis can be rejected. In other words, the test results show that there is a significant difference in outcome between using video and autonomy for obstacle avoidance based on time taken. And this statement can be made with a certainty of 99.93 % ((1-0.0006368)*100). The t test and the sample mean value shows that obstacle avoidance using mixed autonomy improves the teleoperation performance significantly. Moreover, the qualitative results discussed earlier shows that there is significant improvement in operator situational awareness when using mixed autonomy for teleoperating in partially known environments. From the above results, it can be concluded that the virtual reality based teleoperation in mixed autonomy presented here in this thesis is better than current teleoperation interfaces.

Driver abilities versus outcome: The major goal for this user study is to evaluate operator awareness, interface design, and operator adaptability for the proposed virtual reality based teleoperation in mixed autonomy. Based on the results presented in the previous sections all three main parameters have been studied and evaluated. The results indicate that

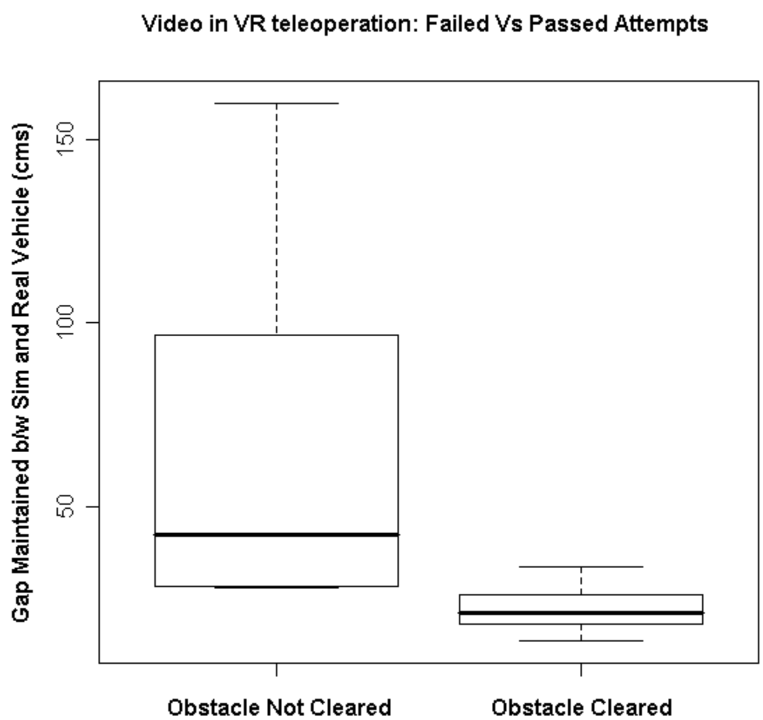

Figure 34. Box Plot - Obstacle Avoidance Result Vs Gap Maintained

Table 9. "T" Test Results

Scenario	Virtual reality teleoperation with **video** for obstacle avoidance: Fail vs Pass - All Results		
T test type	Welch 2 sample T test - Independent unequal sizes with unequal variance		
Null Hypothesis	There is no significant difference in outcome due to the difference in gap maintained by the users between their sim and real vehicles		
T Value	2.0464	**Degrees of freedom**	5.09
P Value	0.09508	**Sample mean**	Gap when failed: 66.25156 Gap when passed: 22.17827
Scenario	Virtual reality teleoperation with **video** for obstacle avoidance: Fail vs Pass within **same user sets**		
T test type	Paired T test - Dependent equal sizes with unequal variance		
Null Hypothesis	Users did not learn anything significant from their first failed attempt which made them clear the second attempt		
T Value	1.3239	**Degrees of freedom**	5.09
P Value	0.2774	**Sample mean of difference**	Gap maintained by the User: 43.097
Scenario	Virtual reality teleoperation with **autonomy** for obstacle avoidance: Video vs Autonomy - All Results		
T test type	Welch 2 sample T test - Independent unequal sizes with unequal variance		
Null Hypothesis	There is no significant difference in outcome when using Video or Autonomy for obstacle avoidance based on time taken in each attempt		
T Value	4.5738	**Degrees of freedom**	12.018
P Value	0.0006368	**Sample mean**	Video Time: 202.9091s Autonomy Time: 103.0s

loss of situational awareness is one of the primary reasons for operator failure in video based virtual reality teleoperation. And the parameters which affected situational awareness in video based virtual reality teleoperation are lag and absence of depth perception. One way to accommodate lag and avoid lack of depth perception is to employ a complete virtual reality based teleoperation. However, this is possible only for completely known environments. In the case of partially known environments, lag has to be accommodated using tools other than VR. Video or sensory inserts and mixed autonomy are some of the tools available for such scenarios. The results presented show that mixed autonomy is a better solution for this problem. However, video inserts can be helpful at times. In fact some participants were able to avoid the obstacle using video in VR. Although they took more time to do it. To understand the boundary constraints of this solution from the collected data additional analyses were conducted.

Based on the successes of some participants in video based virtual reality teleoperation, it can be concluded that there could be some underlying parameters which, when maintained within threshold, could help a user employ video in VR as an effective means to teleoperate in partially known environments without losing situational awareness, albeit at an expense of time. The following paragraphs explain the analysis conducted to study these underlying parameters. The significant data to study are, Study 1: The difference between the users who failed using video and those who cleared using video. The sample here could be overlapping. However as the task is independent the sample can be considered independent, Study 2: The difference between actions taken when users who failed in their first attempt but cleared in their second attempt. This group involves the same users and hence the sample is dependent. The main parameter which could be understood from evaluating study group 1 is “level of caution”. The participants who cleared the test could have been more responsible in driving the vehicle when compared to the other group. In study group 2 participants could have learned “something” from their first attempt which made them succeed in their second attempt. The study could help to understand if there is any “degree of learning” that helped the user to succeed in their second attempt.

Level of Caution: In the proposed virtual reality interface, the user drives the simulated vehicle ahead of the real vehicle. The real vehicle always acts as a follower and traces a path to reach the simulated vehicle at any instant in time. In order to have better command over the real vehicle, it is necessary that the user keeps the wagon tongue, i.e., the distance between the simulated and real vehicles relatively small. This wagon tongue value correlates directly with to the level of caution employed by the participant. At any given time a user is better able to maneuver the vehicle using video in VR if he/she maintains a smaller wagon tongue distance.

The data collected in validation tests for scenario 2 and 3 includes both simulated vehicle position and real position for every second. It can be assumed that the level of caution is important only until the vehicle avoids the obstacle. If the participants decided that they have avoided the obstacle based on the video inputs, then their level of caution would automatically come down. They would then drive the vehicle to reach the goal without worrying about any new obstacle. This position corresponds to the maximum deviation along the vehicle *x*-axis of the simulated vehicle data, after which the user would have performed the necessary corrective action to reach the goal. To explore this phenomenon the gap or wagon tongue distance is calculated between the simulated and real vehicle positions for every second until the simulated vehicle is in the position it was assumed by the user to have avoided the obstacle. A mean tongue distance per second value is determined by calculating the average of all tongue distance per second for every attempt. This mean tongue distance per second is denoted as “vehicle gap”. In all there were 10 successful user data points and 6 failure user data points to evaluate, which resulted in 10 vehicle gap values that could have helped clear the obstacle and 6 vehicle gap value that could have resulted failures. The results were represented using a box plot as shown in Figure 34.

From the plot, it can be inferred that the 25th, 50th and 75th quartile values of vehicle gap in failed attempts are 25, 45 and 100 cm respectively, and in the passed attempts are 15, 20, 25 cm respectively. Thus on average the participant was able to clear the obstacle when maintaining a smaller vehicle gap. Moreover, a *T* test was conducted on the two sets of data. The sample size is unequal, the data is assumed to have unequal variances and the data is independent. The Welch *T* test formula is applied for a 0.1 confidence interval. The null hypothesis here is *“there is no significant difference in outcome due to the difference in gap maintained by the user between the simulated and real vehicle”*. Table 9 presents the *T* test results for this group. The p-value calculated here is 0.095 and is less than the level of significance value 0.1. Hence the null hypothesis can be rejected. In other words, the test results show that there is a significant difference in outcome due to the difference in gap maintained by the user between the simulated and real vehicle. This can be stated with a 90.5% confidence. Thus the conclusion can be drawn that cautious users who maintained smaller distance between the simulated and real vehicle had a 90.5% more chance than other users in dodging the obstacle and reaching the goal position using video enabled VR. The mean “vehicle gap” for success is calculated to be 22cms.

Degree of Learning: These results gives rise to another question, “is level of caution the only parameter that helps the user improve his/her situational awareness in a video based task? Or are there other factors which the user learned during the first attempt that made him/her pass the second attempt”. In order to answer this question, the users who failed to avoid the obstacle in his/her first attempt using video but were able to clear the obstacle on the second attempt were considered. There were in total 4 data points in this category.

The box plot shown in Figure 35 explains the difference in vehicle gap between attempt 1 and attempt 2 data. The graph is very similar to the previous box plot and the vehicle gap value at 50^{th} quartile for successful attempts (15 cm) is far less than that of the failed attempts (35 cm). This shows that vehicle gap is one of the factors which helped the participant dodge the obstacle in his/her second attempt. A *T* test was conducted on these data points. It can be observed that, the size is small and unequal, the sample has unequal variances and the data is dependent. The users who took the first test and the second test are the same. Paired *T* test [96] is the type of *T* test suitable for such a

test statistic. The mathematical formula for a paired *T* test is as follows:

$$t = \frac{\overline{X}_D - \mu_0}{S_D / \sqrt{N}} \qquad \textbf{Equation 15}$$

Where $\overline{X}_D$ is the average of difference in value between the 2 data sets,

S_D is the standard deviation in the difference of value between the 2 data sets,

N is the number of samples,

μ_0 is the constant if any to which the average of the difference is tested against.

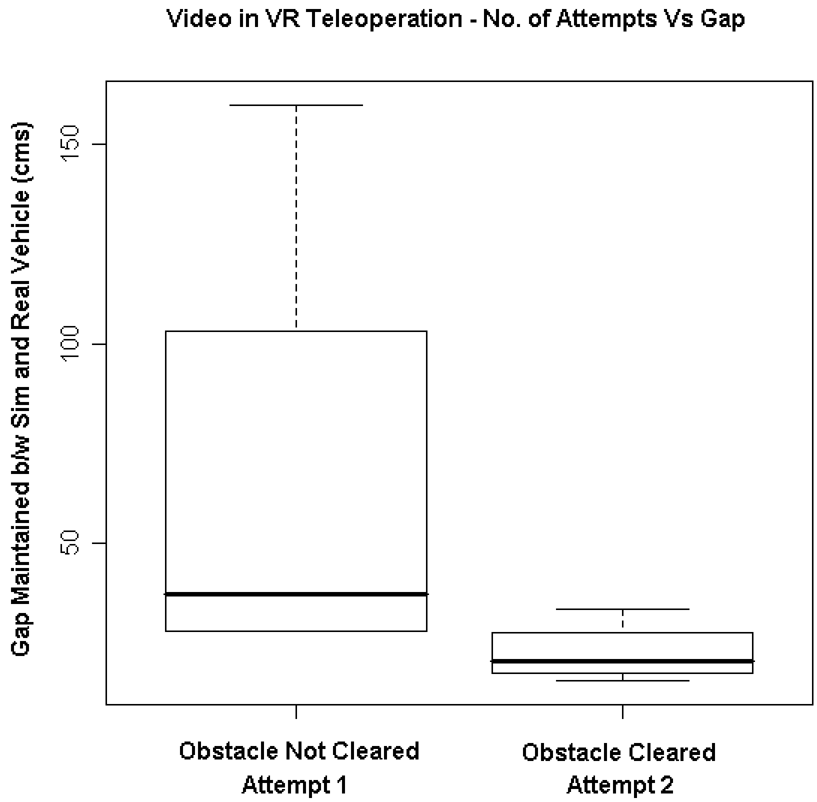

Figure 35. Box Plot - Obstacle Avoidance Result Vs Gap Maintained (Dependant users)

The results of the paired *T* test are presented in Table 9. The assumed null hypothesis here is that *"difference in vehicle gap maintained between the simulated and real vehicle by the user is the only significant parameter which determines the outcome of the test"*. The test is conducted with a level of significance of 0.1. The p-value calculated for this test is 0.277 which is greater than the level of significance value 0.1. Hence, the null hypothesis cannot be rejected. Based on the test results obtained from both these *T* tests, it can be observed that users who were cautious had a 90.5% better chance to clear the obstacle using video, however, their chances reduce to around 73% certainty if that is the only factor they count on. The study shows that users who cleared their second attempt learned something more than just being cautious and slow. The answer to this can be identified from the various observations made by the participants during the think aloud technique. The following are some of the observations made by users, who participated in the video tasks,

"There was inconvenience in estimating how far the obstacle really was"

"I tried twice and succeeded at last. I came to know the presence of the obstacle from the camera view" – researcher observations: the participant had difficulty placing the obstacle in context. The participant guesstimated the obstacle position short of the actual distance

"User verbal comments: video does not tell me how far the obstacle is from me"

The observations and comments indicate that the participants found the video to be helpful enough in identifying whether there is an obstacle or not. However, the 2D picture did not provide them with the necessary depth perception which a human eye is accustomed to for decision making. This resulted in user either overshooting or undershooting their steer turn resulting in loss of the vehicle. The test results along with the qualitative observations shows that the two main factors that helped participant dodge an obstacle, namely level of caution and ability to estimate depth through learning. These results indicate the conclusion that a cautious driver who has better sense of depth perception can maintain a better situational awareness when using video in VR for teleoperating partially known environments. Most of the participants who cleared the obstacle using the video in their first attempt were cautious and had better sense of estimating depth. Participants who failed in their first attempt but cleared in their second attempt realized that they can accommodate lag by being slow. Moreover the practical experience they gained by colliding with the obstacle in their first attempt helped them better estimate the depth in their second attempt. However, it should be noted that real world scenarios do not provide with multiple attempts.

Discussion

A user study was designed and presented to evaluate the proposed virtual reality teleoperation using mixed autonomy. The chapter discussed in detail relevance of user studies, their requirements and guidelines, methodologies and techniques used, the proposed interface, evaluating parameters for the proposed interface, user study design along with the questionnaire, results and analysis. It was understood that the major factors that affect a teleoperation user interface are namely lag, FOV and unknown obstacles/surprises. The user study was designed such that it can evaluate and help assess whether the proposed interface can assist teleoperators on these factors. The study consisted of three main scenarios designed to evaluate operator awareness, adaptability and interface design. 12 users participated in the user study. A questionnaire was used along with the think aloud technique and a data recorder to collect data from the tasks. The qualitative data collected were studied and tabulated. The quantitative data collected were processed and analyzed using statistical methods. Based on the final analysis it is concluded that the virtual reality based teleoperation using mixed autonomy provides a better interface that can improve operator awareness and performance. The interface design features along with mixed autonomy allow an operator to better adapt to surprises when teleoperating in a partially known environment. Moreover, the statistical analysis of video based virtual reality teleoperation provided some interesting insights. The results showed that higher level of caution coupled with good depth perception would help users accommodate lag and accommodate surprises when using video as a tool for obstacle avoidance. However, increased depth perception is not possible to achieve using 2D video feedback. The scenario explains the need for additional 3D sensory feedback which can provide depth perception. Nevertheless, such feedback would be subject to lag resulting in deteriorating the user experience further. The mixed autonomous VR system is useful in such a situation. The system provides the real vehicle with partial autonomy for a brief amount of time to carry out the low level task and at the same time keeps the operator in the loop. This allows the operator to carry out his teleoperation task at hand and not worry about low level tasks like obstacle avoidance. The qualitative user study results presented in Table 8 shows that majority of the users preferred the proposed teleoperator interface when compared to video in VR interface. A few users indicated that although they liked the proposed mixed autonomous interface they would still like to have the option of human in the loop overriding vehicle actions rather than the other way round. The statistical analysis and user observations helps show that balancing and sharing autonomy between a vehicle and teleoperator is feasible as long as the human in the loop is well informed. Moreover, majority of the users suggested that they are willing to share or give up autonomy for a brief period of time in situations where the human is not helpful for the larger benefit of the task at hand. The user study helps reach the conclusion that the proposed virtual reality based multi-modal interface using mixed autonomy is intuitive and adaptable when compared to other teleoperation interfaces.

CHAPTER 6. CONCLUSION AND FUTURE WORK

Conclusion

I presented a virtual reality based multimodal teleoperation interface that facilitates teleoperators to control remote vehicles in partially known environments. The interface consists of a stereo vision based real-time obstacle detection system; a virtual reality based 3D interface and a vehicle adaptation system. The system allows the operator to drive a simulated vehicle in virtual reality to control the real vehicle. The stereo vision system acts as the vehicle's senses and helps identify obstacles in the vehicle path that are not seen by the operator in VR. The on board vehicle adaptation system provides the vehicle with autonomy for a brief amount of time that is sufficient enough to change the vehicle's travel path to avoid the obstacle but at the same time controls the vehicle such that it stays within the goals set by the human operator. The multiple perspective views possible in VR combined with intuitively presented real time information allows the operator to drive the vehicle with less distraction and improves operator situational awareness. The reliability results along with user study results show that VR in mixed autonomy mode is effective when compared to standard teleoperation interfaces in partially known environments.

The challenges in partially known environments are multifold. There exists a constant possibility that what the operator sees the "old" environment is not exactly how the real environment is. Even if a method is devised to identify this "difference", the challenge lies in how the operator is presented with this difference and how he/she is facilitated by the system to navigate in this scenario without loss of situational awareness.

The most commonly used video based teleoperation system can provide the operator with new objects in the scene through video. However, the presence of lag and “soda straw” view impairs situational awareness. As the environment is partially known, using VR is a sensible solution. VR as a tool separates the simulation state from the real state and allows the teleoperator to drive the vehicle without video. As 2D video is not the primary interface for decision making, the lag that otherwise persists in a video based teleoperation system is accommodated. Moreover, VR provides better FOV for the operator when compared to the “soda straw” view available in 2D video based teleoperation systems. However, VR can be used only in environments where the operator has complete prior knowledge about the terrain which is not the case for partially known environments.

The teleoperation taxonomy presented by Milgram et al [40] demonstrated that an effective use of autonomy in VR based teleoperation is possible. A balanced autonomy is possible only if we can identify suitable sensors and provide considerable intelligence to the teleoperated vehicle. The virtual reality teleoperation system presented in this dissertation showed that providing a solution to partially known environments with new static differences will be the first step towards solving this problem. The dissertation identified that a synchronized stereo vision system is a sensor reliable enough for teleoperating remote vehicle in well lighted indoor or outdoor facilities that has small static obstacles. The system might not be useful in other environmental conditions. The research goal is not to identify an all purpose vehicle sensory system but rather to identify that sensor that is effective enough to show that VR with autonomy can be effective tool for teleoperation. The vision system presented uses pixel matching algorithms and projective geometry to identify objects present in front of the camera. The system has very negligible error in static camera conditions and a

95% success in moving camera conditions. However, when the stereo vision system is used in conjunction with a simple vehicle adaptation system with a threshold of 1.5 m the success drops to 78%. Nevertheless, the reliability is sufficient to understand the effectiveness of VR aided teleoperation with vehicle autonomy.

The vision system helped address the challenge in identifying the differences that exists between the “old” and the “real” environment. However, there exist substantial problems in using this information effectively to teleoperate the vehicle. The research shows that so far there exists no solution in informing the operator about this new difference and making him/her carry out this new task. As this brings back the very challenge of accommodating lag which was alleviated earlier using VR. Moreover, it also distracts the operator from carrying out his/her actual tasks at hand. The answer lay in providing considerable autonomy to the vehicle for a brief period of time and balancing it effectively between vehicle and human throughout the entire operation. The vehicle adaptation system implemented in this thesis uses the on board computational power provided on the vehicle to process the obstacles/objects identified. The system with the help of the latest environmental terrain data that is obtained from the operator station, maps this new object location onto the terrain map. It then charts out a new path for the vehicle to follow. This results in a shift in autonomy from human to vehicle for a brief period of time. However, this shift does not stall or distract the operator from charting out his/her future short term goals in the teleoperation mission.

The prototype developed and presented in this thesis implements a simple A* search algorithm for path identification in order to realize the proof of concept. However, there exist more sophisticated and reliable path planning algorithms that can help identify the most suitable path for the vehicle in real life scenarios. The reliability results of the proposed vehicle adaptation system show that the VR aided teleoperation with mixed autonomy is realizable and practical. The stereo vision system and vehicle adaptation system does produce substantial errors while identifying the obstacle and as well as in reaching the goal through the new path. But this can be attributed to the on board computational limitations. Moreover, it should be noted that the prototype was built on a toy car platform that has substantial mechanical play resulting in imprecise vehicle control. This can be alleviated by using a more sophisticated vehicle.

Teleoperation systems are favored over autonomous systems for situations where human in the loop is considered important. However, in this proposed model the human is asked to give up autonomy for a brief amount of time to carry out those tasks which are not possible by him/her. The challenge then lies in creating a VR interface that not only presents the new obstacle/object data to the teleoperator but at the same time informs him/her about the vehicle actions including the shift in autonomy without affecting his/her situational awareness. The multiple camera views allowed in the VR interface provided operator with capability to better understand the environment surroundings. Real time data like real vehicle location and distance between the real and sim vehicle are represented intuitively in the form of a ghost vehicle, transparent vehicle and distance bar. Moreover, the operator is informed about the shift in autonomy and the presence of obstacles using warnings. Finally, the real environment is updated with the new obstacle. The user studies results presented in this thesis show that the above described features of the VR interface increased operator's environment knowledge substantially and thereby improved operator situational awareness.

The VR based multi-modal teleoperation met its objective in retaining the human control necessary for decision making while providing a considerable level of autonomy for the vehicle to accommodate surprises encountered in its immediate vicinity. In this way, the operator did not experience the loss of situational awareness due to lag and lack of peripheral vision while navigating a dynamic environment, thereby increasing accuracy and utility. The interface effectively balances autonomy between man and machine and retains the human control necessary for decision making while providing considerable autonomy for the vehicle to accommodate surprises encountered in its immediate vicinity. By allowing the vehicle to temporarily detach from the simulated state during the warning period, the operator continues driving in the simulated state with additional knowledge about the real state in the form of the transparent vehicle. The VR based multi modal teleoperation interface is shown to be more adaptable and intuitive when compared to other interfaces.

Future work

The results from the experiments and user studies helped show that the VR based multi-modal interface was able to meet its requirements as proposed in this thesis's problem statement. The proposed interface is primarily designed to address the first step towards using VR for teleoperating in a dynamic environment. The interface is designed to teleoperate in a partially known environment with static "new" obstacles. However, a complete VR interface should be able to accommodate any new surprises in the real environment which includes both new static and new dynamic objects. The current system uses stereo vision as the primary sensor that is well suited for indoor environments (the test bed). The future work here should identify a better sensor or more than one primary sensor for the vehicle has to accommodate dynamic objects. Moreover, teleoperating in dynamic environments also involves tracking moving objects and this can be done by using tracking algorithms like Kalman filtering.

Although the stereo vision system implemented can identify more than one obstacle in front of the camera system, it reacts only to the closest obstacle. A sophisticated vision system and an improvised path planning algorithm could help the system accommodate more than one obstacle simultaneously. In the current system it is assumed that the terrain data is kept “near” up to date using satellite data and other real time systems. However, the risk map is generated beforehand and no updates are made to the VR terrain except for the obstacles identified by the vision system. This can be further improved by integrating the standalone risk map generator to the virtual reality system and generating the 3D model in real time.

The system can be further improved by using a more sophisticated dynamic model. The addition of a fully functional 3D dynamics model would not only allow creating vehicle simulations with more realistic terrains that includes hills and valleys but as well will allow driving the real vehicle prototype in outdoor environments. Moreover, such a model will help improve the overall performance of the teleoperation system.

Finally, the virtual reality interface could be greatly improved by placing the video frames in context of the 3D model. In the current system the vehicle transfers the control back to the operator for a video in VR teleoperation in situations where it cannot identify a suitable path around the obstacle. The operator situational awareness in such situations can be improved by placing the video feed in context of view point from which it is taken in the actual environment. In this way, the operator would be able to get a more realistic perspective of the real environment in VR.

APPENDIX A. USER STUDY QUESTIONNAIRE

The following pages present the survey questionnaire for all three scenarios. This study is conducted to understand the effectiveness of the virtual reality based user interface that is designed to teleoperate a remote vehicle. The system comprises of an operator station, the remote vehicle and tracking station. The operator station has a virtual reality interface which is used to drive a simulated vehicle. The real vehicle receives the data from the operator station regarding where the simulated vehicle is, and follows the same path. The tracking station provides local position of the real vehicle to the operator station and to the real vehicle.

Moreover, the real vehicle has capability to carry out low level tasks like obstacle detection and path planning. The real vehicle can identify obstacles on its path using a stereo vision based obstacle avoidance system and can design a new path such that it can avoid the obstacle using a on board path planner. Both these systems can be enabled and disabled when necessary.

This user study is designed to understand two major issues.

1. Operator awareness of the environment
2. Operator adaptability to the environmental changes

Experiment 1: Operator awareness of the environment

Constraints

1. The vision system for obstacle detection and path planning in the real vehicle will be disabled

2. There will be no difference between the operator’s virtual reality environment and real vehicle driving environment. In the sense, what you see in the 3D model is what exists in real world.
3. The simulated vehicle will be driven in the VR environment by the operator (User - you) and the real vehicle will follow the simulated vehicle and drive in the real environment

User steps

1. The user takes the position as operator of the simulated vehicle
2. The user will see a 3D model of Haptics lab with a simulated vehicle ready to be driven.
3. Microsoft sidewinder steering wheel is used to provide steering input to drive the simulated vehicle and the accel pads are used for providing throttle.
4. The user will use the sidewinder steering wheel and acceleration pads to travel around.
5. The system does not have separate braking system.
6. The vehicle stops the moment, the user takes his/her control off the acceleration pads.
7. The user will see a transparent vehicle blob around the simulated vehicle indicating how far the real and sim are away from each other.
8. There will also be a color bar indicator showing this error distance.
9. The speed of the vehicle can be identified using the speedometer on screen.
10. The user is provided with 3 major views to look around the environment, chase cam view which puts the user eye in front of the vehicle, perspective cam view, which

puts the user eye behind the vehicle and rear cam view which puts the user eye on top of the overall operator environment. The user can use the button controls to navigate among the views and choose based on their preference.

11. The user will be provided with a starting position and a goal position. The path to be traveled is marked in the 3D model.
12. The user task is to drive the simulated vehicle along the given track and reach the goal position.
13. The transparent vehicle will show how much real vehicle is away. And the real vehicle will be shown separately as a solid 3D model.
14. The user should remember that the real should be close to the sim, for the vehicle to be in control.
15. Hence the user should make effort to get the real as close to sim as possible while driving and reaching the goal position

Questionnaire

System/Past Experience (Questions will be filled prior to carrying out the experiment)

1. Have you played 3D computer games, if yes, please answer the following question.

   How long have you played 3D computer games?

   _____ Very rarely

   _____ Less than an hour per week

   _____ 1-5 hours per week

   _____ more than 10 hours per week

2. Select game systems you are familiar with or used. You can select more than one if necessary.

_____ X Box

_____ Sony Play station

_____ Nintendo Wii

3. Are you familiar with virtual reality? If so, how will you rate yourself with regard to knowledge on virtual reality?

_____ No prior experience or knowledge

_____ Have heard about it

_____ Will rate me as knowledgeable

_____ Will rate me as expert

4. Have you attended user studies for any other projects or research work? If yes, did you experience any problem during the course of study which the researcher of this study should be aware off? (if yes please explain in detail)

____

Following questions shall be filled after conducting the experiment

Overall User Reactions

1. In the scale 1- 10, 1 being poor and 10 being excellent, circle a number that reflects your experience appropriately in using this system so far in this experiment

a. Frustrating 1 2 3 4 5 6 7 8 9 10 Satisfying

b. Difficult 1 2 3 4 5 6 7 8 9 10 Easy

Explain in detail your previous answer.

____

Virtual Reality Interface Experience

1. Rate in the scale of 1-10, 1 being poor and 10 being excellent, how comfortable you were in using the input gadgets sidewinder wheel and acceleration pads.

   Poor 1 2 3 4 5 6 7 8 9 10 Excellent

2. Is there any other better input device that you could suggest in place of the sidewinder wheels for inputs?

   ____

3. Rate in the scale of 1-10, 1 being poor and 10 being excellent, how confident you were in each step that the real vehicle is driving and following your user inputs

   Poor 1 2 3 4 5 6 7 8 9 10 Excellent

4. Where the transparent blob, real vehicle representation and distance bar helpful in showing real time information? ____

   If yes, please write them in the order of most useful starting first

   ____

5. Rate in the scale of 1-10, 1 being poor and 10 being excellent, whether you were able to get the environment knowledge as quick as possible at every instant.

   Poor 1 2 3 4 5 6 7 8 9 10 Excellent

6. Rate in the scale of 1-10, 1 being poor and 10 being excellent, did performing an operation lead to a predictable result, that is was the interface responsive enough for the actions you performed

   Poor 1 2 3 4 5 6 7 8 9 10 Excellent

7. Did you have an opportunity to use the multiple camera views? __________

8. If yes, was that property useful in performing your task or did it create confusion in carrying out your task? Explain ______________________________________

Experiment 2: Operator adaptability - Scenario 1

Constraints

All constraints, assumptions and steps are the same as experiment one. Except that now there will be unknown obstacles present in the environment

1. The operator will be driving the vehicle using the VR interface.
2. The virtual reality 3D model and real vehicle environment will not be identical.
3. A live video feed will be provided to get real environment details.
4. An obstacle will be placed on the predefined path at any position.
5. And operator action/reaction will be studied.

The scenario 1 is used as the referral base for understanding scenario 2.

User Steps

1. The user takes the position as operator of the simulated vehicle.
2. The user will see a 3D model of Haptics lab with a simulated vehicle ready to be driven.
3. Microsoft sidewinder steering wheel is used to provide steering input to drive the simulated vehicle and the accel pads are used for providing throttle.
4. The user will use the sidewinder steering wheel and acceleration pads to travel around.

5. The system does not have separate braking system.
6. The sim vehicle stops the moment, the user takes his/her control off the acceleration pads.
7. The user will see a transparent vehicle blob around the simulated vehicle indicated how far the real and sim are away from each other.
8. There will also be a color bar indicator showing this error distance.
9. The speed of the vehicle can be identified using the speedometer on screen.
10. The user is provided with 3 major views to look around the environment, chase cam view which puts the user eye in front of the vehicle, perspective cam view, which puts the user eye behind the vehicle and rear cam view which puts the user eye on top of the overall operator environment. The user can use the button controls to navigate the views as they prefer.
11. The user will be provided with a starting position and a goal position. The path to be traveled is marked in the 3D model.
12. The user task is to drive the simulated vehicle along the given track and reach the goal position.
13. However, this time the user will have an option to see video feed in front of the simulated vehicle in the VR interface.
14. Once the user started driving the vehicle the researcher will place an obstacle in between the real vehicle and the goal position.
15. The user should make an attempt to drive the vehicle around the obstacle and reach the goal position using the video feed.

16. The transparent vehicle will show how much the real vehicle is away from sim. And the real vehicle will be shown separately as a solid 3D model.
17. The user should remember that the real should be close to the sim for the vehicle to be in control.
18. Hence the user should make effort to get the real as close to sim as possible while driving and reaching the goal position

Researcher's Data

User No.	Obstacle position from starting position (m)	Result of the event

Questionnaire

Overall User Reactions

1. In the scale 1- 10, 1 being poor and 10 being excellent, circle a number that reflects your experience appropriately in using this system, in this experiment.

   a. Frustrating 1 2 3 4 5 6 7 8 9 10 Satisfying

   b. Difficult 1 2 3 4 5 6 7 8 9 10 Easy

2. Explain in detail your previous answer.

____

Virtual reality interface

1. Rate in the scale of 1-10, 1 being poor and 10 being excellent, how confident you were in each step that the real vehicle is driving and following your user inputs.

   Poor 1 2 3 4 5 6 7 8 9 10 Excellent

2. Rate in the scale of 1-10, 1 being poor and 10 being excellent, whether you were able to get the environment knowledge as quick as possible at every instant.

   Poor 1 2 3 4 5 6 7 8 9 10 Excellent

3. Were you able to avoid the obstacle that was added while you were driving? And how did you come to know about the presence of obstacle?

   ____

4. Rate in the scale of 1- 10, 1 being poor and 10 being excellent how comfortable you were in using the video feed as a means to avoid the obstacle

   Poor 1 2 3 4 5 6 7 8 9 10 Excellent

Experiment 2: Operator adaptability - Scenario 2

Constraints

1. The stereo vision system for obstacle detection and path planning will be enabled.
2. The real vehicle will have capability to go autonomous, that is not to listen to user's low level commands depending on its intelligence system.
3. There will be two goal positions to reach, one intermediate and one final.

User steps

1. The user will assume the position of the teleoperator and drive the simulated vehicle.
2. He/she will be asked to reach an intermediate goal position. There will be a t second lag, in this case a 30 second lag, after which the real vehicle will start. This ensures that a gap is established between the sim state and real state.
3. The real vehicle starts after 30 seconds and works towards reaching the sim position

4. The researcher will place an obstacle in between the already identified goal position and current real vehicle position.
5. The real vehicle will go autonomous and travel around the obstacle.
6. In the mean time, the user will get a flash warning on obstacle and vehicle being autonomous, and will be asked to slow down.
7. The user will then be asked to resume driving when the vehicle reached the intermediate goal position.
8. The user will then drive like in scenario 1 to reach the final goal position

Researcher's Data

User No.	Obstacle position from starting position (m)	Result of the event

Questionnaire

Overall User Reactions

1. In the scale 1- 10, 1 being poor and 10 being excellent, circle a number that reflects your experience appropriately in using this system so far in this experiment.

   a. Frustrating 1 2 3 4 5 6 7 8 9 10 Satisfying

   b. Difficult 1 2 3 4 5 6 7 8 9 10 Easy
2. Explain in detail your previous answer.

____

Virtual reality interface

1. Rate in the scale of 1-10, 1 being poor and 10 being excellent, explain how comfortable you were in understanding what was going on.

Poor 1 2 3 4 5 6 7 8 9 10 Excellent

2. Can you describe in your words what happened in this scenario.

   ____

3. Rate in the scale of 1-10, 1 being bad and 10 being excellent, how much the warnings and information provided, helped you react to the situation.

   Poor 1 2 3 4 5 6 7 8 9 10 Excellent

4. If you have been asked to choose between the scenario with video feed and this scenario in order to avoid obstacle which one will you choose (based on your experience in this user study)

   a. Obstacle avoidance using video feed.

   b. Autonomous Obstacle avoidance

Informed consent document

Title of Study

Virtual reality based multi-modal teleoperation using mixed autonomy: A user study

Investigators

Muthukkumar S. Kadavasal,

Human Computer Interaction program,

Virtual Reality Applications Center, Ames, IA

Prof. James H. Oliver,

Director of Human Computer Interaction program,

Virtual Reality Applications Center Ames, IA

This is a research study. Please take your time in deciding if you would like to participate. Please feel free to ask questions at any time.

Introduction

The purpose of this study is to understand and evaluate a virtual reality based 3D user interface that is created to control remote ground vehicles. You are invited to participate in this study because you are affiliated with Iowa State University and have basic knowledge in computers and experienced playing computer games.

Description of Procedures

If you agree to participate in this study, your participation will last for one hour which includes 3 different scenario trials. During the study you may expect the following study procedures to be followed: The user will be asked to drive a remote toy vehicle using a graphical 3D interface displayed in a monitor. The user will drive along the required test path three times followed by which he/she would be requested to under take a 15-30 minute user survey. “The user may skip any question that he/she do not wish to answer or that makes them feel uncomfortable.

Risks

While participating in this study you may experience the following risks: NONE

Benefits

If you decide to participate in this study there will be no direct benefit to you [A benefit is defined as a “desired outcome or advantage.”] It is hoped that the information gained in this study will benefit user interface research by providing information and research data for designing better graphical 3D user interfaces.

Participant Rights

Your participation in this study is completely voluntary and you may refuse to participate or leave the study at any time. If you decide to not participate in the study or leave the study early, it will not result in any penalty or loss of benefits to which you are otherwise entitled.

Confidentiality

Records identifying participants will be kept confidential to the extent permitted by applicable laws and regulations and will not be made publicly available. However, federal government regulatory agencies [the sponsor – Air Force Research laboratory], auditing departments of Iowa State University, and the Institutional Review Board (a committee that reviews and approves human subject research studies) may inspect and/or copy your records for quality assurance and data analysis. These records may contain private information. To ensure confidentiality to the extent permitted by law, the following measures will be taken – All subjects will be assigned a unique number code and will be used on forms instead of their name. No identifiers will be kept with the data. The research does not include maintaining a registry of users for future research. The name and the contact information of the participants will not be stored for future reference. All the coded data will be stored in Virtual Reality Applications Center's memory storage and can be accessed only by the researcher or the system administrator of the center. If the results are published, your identity will remain confidential.

Questions or Problems

You are encouraged to ask questions at any time during this study.

- For further information about the study contact Muthukkumar Kadavasal at ksmkumar@iastate.edu or at 515 450 4291.
- Also Prof. James H. Oliver at oliver@iastate.edu or at 515 294 3092
- If you have any questions about the rights of research subjects or research-related injury, please contact the IRB Administrator, (515) 294-4566, IRB@iastate.edu, or Director, (515) 294-3115, Office of Research Assurances, Iowa State University, Ames, Iowa 50011.

Participant Signature

Your signature indicates that you voluntarily agree to participate in this study, that the study has been explained to you, that you have been given the time to read the document and that your questions have been satisfactorily answered. You will receive a copy of the written informed consent prior to your participation in the study.

Participant’s Name (printed)

(Participant’s Signature) (Date)

Investigator Statement

I certify that the participant has been given adequate time to read and learn about the study and all of their questions have been answered. It is my opinion that the participant understands the purpose, risks, benefits and the procedures that will be followed in this study and has voluntarily agreed to participate.

(Signature of Person Obtaining Informed Consent) (Date)

Recruitment letter

Dear Sir/Madam,

I am a PhD Candidate in Human Computer Interaction program at Virtual Reality Applications Center, Iowa State University, Ames, IA. My research topic is Virtual Reality based Interfaces for operating remote vehicles. This letter is regarding a User study that I am conducting to understand the Virtual Reality based 3D graphical interface developed by me to control a remote ground vehicle. I would like to request your participation in this study. The participant is expected to have basic computer skills and have some experience playing 3D computer games.

Study brief

The ground vehicle here is a toy car which has an on board computer for processing data. The participant will be asked to drive a real toy car remotely using the 3D simulation interface that is shown in front of him/her in a personal computer. The participant may carry out 3 trials in driving the simulation but each being a different scenario. The results of trials will be tabulated by the researcher. Finally the participant if willing can fill out a survey form to help researcher better understand and evaluate the interface. The researcher would like to inform that the participation is strictly voluntary and the participant has the right to withdraw from the study at any time. I have attached the consent form along with this email for your better understanding of this study. If you are interested in participating in this study please contact me at ksmkumar@iastate.edu to identify a suitable time.

Thank you,

Muthukkumar Kadavasal

BIBLIOGRAPHY

Autodesk Maya 3D, www.Autodesk.com/Maya, visited Mar 2008.

Baack, E. J., 2004, “A zero slip vehicle dynamics model for virtual reality applications”, Masters Thesis, Iowa State University.

Balakrishnan, G., Sainarayanan, G., Nagarajan, R., and Yaacob, S., 2004, “Stereopsis method for visually impaired to identify obstacles based on distance”, Proceedings of the Third International Conference on Image and Graphics, pp. 580 – 583.

Ballantyne, J., Wong, E., 1998, “A virtual environment display for teleoperated excavation”, IEEE/RSJ International Conference on Intelligent Robots and Systems, 3, pp.1894-1899.

Batkiewicz, T., 2006, “An integrated task manager for virtual command and control”, Masters Thesis, Iowa State University.

Beyer H., and Holtzblatt, K., 1998, “*Contextual design: defining customer-centered systems*,” San Mateo, CA, Morgan Kaufmann.

Birchfield, S., and Tomasi, C., 1996, Depth discontinuities by pixel-to-pixel stereo, Technical Report STAN-CSTR -96-1573, Stanford University, pp. 1073-1083.

Brogden, J., Dunlap, R., Dyson, P., Nicolas, J. M., Nicolas, A. M., Nicolas, J. M., McCauley, D., Miner, D.,O’Quin, J., Polkowski, S., and Roever, J., 2005, “Austin robot technology inc. DARPA grand challenge”, Technical Report, Austin Robot Technology Inc.

Brooks, F. P., 1999, “What's real about virtual reality?”, IEEE Computer and Graphics Applications, 19(6), pp.16-27.

Brujic-Okretic, V., Guillemaut, J. Y., Hitchin, L. J., Michielen, M., and Parker, G.A., 2003, "Remote vehicle maneuvering using augmented reality", International Conference on Visual Information Engineering, Ideas, Applications, Experience, pp. 186-189.

Burdea, G. C., and Coiffet, P., 2003, "*Virtual reality technology*", Wiley Interscience Publication, 2nd Edition.

Calhoun, G. L., Draper, M. H., Abernathy, M. F., Delgado, F., and Patzek M., 2005, "Synthetic vision system for improving unmanned aerial vehicle operator situation awareness", Proc. SPIE Enhanced and Synthetic Vision, 5802, pp. 219-230.

Calisi, D., Farinelli, A., Iocchi, L., Nardi, D., 2005, "Autonomous navigation and exploration in a rescue environment," Safety, Security and Rescue Robotics, Workshop, 2005 IEEE International, pp. 54-59.

Casals, A., 2005, "Robotics research: Assisted teleoperation through the merging of real and virtual images", pp. 135-144, DOI -10.1007/11008941_15.

Casella, G., and Berger, R. L., 2005, "*Statistical inference*," Duxbury Press, 2nd edition.

Clegg, C. W., Warr, P. B., Green, T. R. G., Monk, A., Kemp, N., Allison, G., and Lansdale, M., 1988, "*People and computers: how to evaluate your company's new technology*," Eliss Horwood Chichester.

Collins, R. T., 1995, "Site model acquisition and extension from aerial images," Technical Report, University of Massachusetts Department of Computer Science.

Cruz-Neira, C., Sandin, D. J., DeFanti, T. A., Kenyon, R. V., Hart, J. C., 1992, "The CAVE: audio visual experience automatic virtual environment", Communications of the ACM, 35(6), p.64-72.

Demiralp, C., Jackson, C.D., Karelitz D.B., Zhang S., Laidlaw D.H., 2006, "CAVE and fishtank virtual-reality displays: a qualitative and quantitative comparison", Transactions on Visualization and Computer Graphics , 12(3), pp. 323-330.

Dijkstra, E. W., 1959, "A note on two problems in connection with graphs," *Numeriche Mathematik*, 1, pp. 269–71.

Dunkler, O., Mitchell, C.M. Govindaraj, T., and Ammons, J.C., 1988, "The effectiveness of supervisory control strategies in scheduling flexible manufacturing systems", IEEE Transactions on Systems, Man and Cybernetics, 18(2), pp.223-237.

Eagar, T., and Musso, C., 2001, "Remote-controlled robots search world trade center rubble", Article in JOM, Minerals, Metals and Materials Society, 53, pp. 4-7.

Endo, Y, MacKenzie, D. C., Arkin, R. C, 2004, "Usability evaluation of high-level user assistance for robot mission specification," IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews , 34(2), pp.168-180.

Endsley, M.R., 1988, "Situation awareness global assessment technique (SAGAT)", Aerospace and Electronics Conference, 3, pp.789-795.

Ericsson, K. A., and Simon, H. A., "*Protocol analysis: verbal reports as data*," Cambridge, MA: MIT Press, 1984.

Ferguson, D., and Stentz, A., 2005, "The delayed D* algorithm for efficient path replanning," Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2045 – 2050.

Ferrell, W. R., 1965, “Remote manipulation with transmission delay”, NASA Technical Note, D 2665, Prepared by MIT.

Fong, T., and Thorpe, C., 2001, “Vehicle teleoperation interfaces”, *Autonomous Robots*,11, pp. 9-18.

Fong, T., Conti, F., Grange, S. and Baur, C., 2000, “Novel interfaces for remote driving: gesture, haptic and PDA”, SPIE Telemanipulator and Telepresence Technologies VII.

Fong, T. W., Thorpe, C., and Baur, C., 2001, “Advanced interfaces for vehicle teleoperation: Collaborative control, sensor fusion displays, and remote driving tools,” *Autonomous Robots*, 11, pp. 77-85.

Fraisse, P., and Leleve, A., 2003, “Teleoperation over IP network: Network delay regulation and adaptive control”, *Autonomous Robots*, 15, pp. 225-235.

Fuchs, P., Nashaashibi, F., and Maman, D., 2002, “Assistance for telepresence by stereovision based augmented reality and interactivity in 3D space,” *Presence*, 11(5).

Gamma, E., Helm, R., Johnson, R., Vlissides, J., 1995, *Design patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley.

Gardner, M. M., and Christie, B., 1987, “A*pplying cognitive psychology to user interface and design*,” John Wiley and Sons, Chichester.

Genov, A., Keavney, M., and Zazelenchuk, T., 2009, “Usability testing with real data,” *Journal of Usability Studies*, 4(2), pp. 85-92.

Glumm, M., Kilduff, P., and Masley, A., 1992, “A study on the effects of lens focal length on remote driver performance”, Army Research Laboratory, Technical Report ARL-TR-25.

Grant, R., 2002, “Reach forward”, *Journal of the Air Force Association*, 85(10).

Hainsworth, D. W., 2001, “Teleoperation user interfaces for mining robotics”, *Autonomous Robots,*11, pp. 19-28.

Hallion, R. P., 1984, “On the frontier: Flight research at Dryden 1946-1981”, NASA SP-4303.

Hart, P. E., Nilsson, N. J., and Raphael, B., 1968, “A formal basis for the heuristic determination of minimum cost paths,” IEEE Transactions on Systems Science and Cybernetics, 4(2), pp. 100–107.

Henriksen, L., and Krotkov, E., 1997, “Natural terrain hazard detection with a laser rangefinder”, IEEE International Conference on Robotics and Automation, 2, pp. 968-973.

Hughes, S.B., and Lewis, M., 2005, “Task-driven camera operations for robotic exploration”, IEEE Transactions on Systems, Man and Cybernetics, 35(4), pp. 513-522.

Jarvis, R. A., 1999, “Sensor rich teleoperation of an excavating machine”, Pmc. Field and Service Robotics, pp. 238-243.

Kadavasal, M. S., Oliver, J. H., 2007, “Sensor enhanced virtual reality teleoperation in dynamic environment”, In Proceedings IEEE Virtual Reality Conference, pp. 297-298.

Kadavasal, M. S, Oliver, J. H., 2007, “Towards sensor enhanced virtual reality teleoperation in a dynamic environment”, Proceedings of ASME Design Engineering Technical Conferences and Computer and Information in Engineering Conferences.

Kadavasal, M. S, Seth, Abhishek, Oliver, J. H., 2008, “Virtual reality based multi-modal teleoperation using mixed autonomy,” Proceedings of ASME Design Engineering Technical Conferences and Computer and Information in Engineering Conferences.

Kadavasal, M. S, Oliver, J. H., 2009, “Sensor augmented virtual reality based teleoperation using mixed autonomy,” *J. Computing Information Science Eng*., 9(1).

Kadavasal, M. S, Oliver, J. H., 2009, “Virtual reality interface design for multi-modal teleoperation”, Proceedings of ASME – AFM World Conference on Innovative Virtual Reality.

Kim, W.S., Hannaford, B., and Bejczy, A.K., 1992, “Force-reflection and shared compliant control in operating telemanipulators with time delay”, IEEE Trans. Robotics and Automation, 8(2), pp.176-185.

Knutzon, J. S., 2003, “Tracking and control design for a virtual reality aided teleoperation system,” Masters Thesis, Iowa State University.

Knutzon, J. S., Sannier, A.V., and Oliver, J.H., 2004, “An immersive approach to command and control,” *Journal of Battlefield Technology*, 7.

Kroll, C. V., Roland, R. D., Jr., 1970, “A preview-predictor model of driver behavior in emergency situations,” Technical Report, Cornell Aeronautical Laboratory, Inc., Buffalo, N.Y., Sponsor: Bureau of Public Roads, Traffic Systems Research Division, Washington, D.C., Report No. CAL VJ-2251-V-6. UMTRI-15397.

Laird, R.T., and Bruch, M.H., 2000, “Issues in vehicle teleoperation for tunnel and sewer reconnaissance”, IEEE Conference on Robotics and Automation.

Lawson, S. W., Pretlove, J. R. G., Wheeler, A. C., Parker, G. A., 2002, “Augmented reality as a tool to aid the telerobotic exploration and characterization of remote environments,” *Presence*, 11(4), pp. 352-367.

Lin, Q., and Kuo, C., 1999, “Assisting the teleoperation of an unmanned underwater vehicle using a synthetic subsea scenario,” *Presence*, 8(5), pp. 520 -530.

Livatino, S. and Privitera, F., 2006, “3D visualization technologies for teleguided robots”, In Proceedings of the ACM Symposium on Virtual Reality Software and Technology, pp. 240-243.

Lunenburger, L., Wellner, M., Banz, R., Colombo, G., Riener, R., 2007, “Combining immersive virtual environments with robot-aided gait training,” IEEE 10th International Conference on Rehabilitation Robotics , pp.421-424.

Luo, R. C., Chen, T. M., 2000, “Development of a multibehavior-based mobile robot for remote supervisory control through the internet”, IEEE/ASME Transactions on Mechatronics, 5(4).

Lyons, D.M. and Allton, J.H., 1990, “Achieving a balance between autonomy and teleoperation in specifying plans for a planetary rover”, SPIE Vol. 1387 Cooperative Intelligent Robotics in Space, pp. 124-133.

Marble, J.L.; Bruemmer, D.J.; Few, D.A., 2003, “Lessons learned from usability tests with a collaborative cognitive workspace for human-robot teams,” IEEE International Conference on Systems, Man and Cybernetics, 1, pp. 448-453.

van der Mark, W., Gavrila, D.M., 2006, “Real-time dense stereo for intelligent vehicles”, IEEE Transactions on Intelligent Transportation Systems, 7(1), pp. 38-50.

Matijevic, J., 1998 “The pathfinder mission to mars: Autonomous navigation and the sojourner microrover”, *Science*, 280 (5362), pp. 454.

Meier, R., Fong, T. W., Thorpe, C., and Baur, C., 1999, “Sensor fusion based user interface for vehicle teleoperation”, International Conference on Field and Service Robotics.

Milgram, P., Rastogi, A., and Grodski, J.J., 1995, “Telerobotic control using augmented reality”, Proceedings 4th IEEE International Workshop on Robot and Human Communication, pp.21-29.

Milgram, P., and Ballantyne, J., 1997, “Real world teleoperation via virtual environment modeling”, In International Conference on Artificial Reality & Tele-existence.

Monferrer, A., Bonyuet, D., 2002, “Cooperative robot teleoperation through virtual reality interfaces”, Proceedings Sixth International Conference on Information Visualization, pp. 243-248.

Nguyen, A., Bualat, M., Edwards, J., Flueckiger, L., Neveu, C., Schwehr, K., Wagner, M. D., and Zbinden, E., 2001, “Virtual reality interfaces for visualization and control of remote vehicles”, *Autonomous Robots*, 11, pp. 59–68.

Nourbakhsh, I., Sycara, K., Koes, M., Yong, M., Lewis, M., and Burion S., 2005, “Human-robot teaming for search and rescue”, IEEE Pervasive Computing: Mobile and Ubiquitous Systems, pp. 72-78.

Open CV, http://sourceforge.net/projects/opencvlibrary/, visited on Nov 2007.

OpenSceneGraph, http://www.openscenegraph.org/projects/osg, visited on Mar 2008.

Otsu, N., 1979, “A threshold selection method from gray-level histograms,” IEEE Transactions on Systems, Man and Cybernetics, 9(1), pp.62-66.

Ott, R., Gutiérrez, M., Thalmann, D., and Vexo, F., 2006, “Advanced virtual reality technologies for surveillance and security applications”, In Proceedings of the 2006 ACM international Conference on Virtual Reality Continuum and Its Applications, pp. 163-170.

Qingping, L., and Chengi, K., 1999, “Assisting the teleoperation of an unmanned underwater vehicle using a synthetic subsea scenario”, *Presence*, 8(5), pp. 520-530.

R Project, http://www.r-project.org/, visited on Mar 2009.

Ravden, S., and Johnson, G., 1989, "E*valuating usability of human-computer interfaces: a practical method*," Ellis Horwood Limited.

Ricks, B. Nielsen, C. W. and Goodrich, M.A., 2004, "Ecological displays for robot interaction: A new perspective," in Proc. IROS, 3, pp. 2855- 2860.

Riley, J. M., Kaber, D. B., and Draper, J. V., 2004, "Situation awareness and attention allocation measures for quantifying telepresence experiences in teleoperation", Human Factors in. Ergonomics and Manufacturing, 14(1), pp. 51-67.

Roy, S., and Cox, I., 1998, "A maximum-flow formulation of the n-camera stereo correspondence problem", In 6th International Conference on Computer Vision, pp. 492-502.

Rubin, J., 1994, "H*andbook of usability testing: How to plan, design and conduct effective tests*," John Wiley & Sons, Inc.

Ruff, H. A., Narayanan, S., and Draper, M. H., 2002, "Human interaction with levels of automation and decision-aid fidelity in the supervisory control of multiple simulated unmanned air vehicles", *Presence: Teleoperators and Virtual Environments*, 11(4), pp. 335-351.

Sheridan, T., 1992, "*Telerobotics, automation, and human supervisory control*", MIT Press: Cambridge, MA.

Shneiderman, B., 1987, "D*esigning the user interface: Strategies for effective human computer interaction*," Addison Wesley, Reading, MA.

Smith, S. L., and Mosier, J. N., 1986, “Guidelines for designing user interface software,” The Mitre Corporation, Beford, MA., USA., Report No. MTR -10090, EST-TR-86-278.

Soltani, A. R., Tawfik, H., Goulermas, J. Y., and Fernando, T., 2002, “Path planning in construction sites: performance evaluation of the Dijkstra, A*, and GA search algorithms,” *Advanced Engineering Informatics*, 16(4), pp. 291-303, ISSN 1474-0346.

Sojourner: http://mpfwww.jpl.nasa.gov/MPF/rover/sojourner.html, Visited Nov 2007.

Spenny, C. H., 1999, “Closely supervised control of a target-steered UAV”, Proc. SPIE Telemanipulator and Telepresence Technologies VI, 3840, pp. 179-190.

Sugimoto, M., Kagotani, G., Nii, H., Shiroma, N., Inami, M., and Matsuno, F., 2005, “Time follower's vision: A teleoperation interface with past images,” IEEE Computer Graphics and Applications, 25(1), pp. 54-63.

Surmann, H., Nuchter, A., and Hertzberg, J., 2003, “An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments”, Robotics and Autonomous Systems, 45, pp. 181-198.

Tekalp, A. M., “*Digital video processing*”, published by Prentice Hall PTR, 1st edition, pp. 218-244, 1995.

Tsalatsanis, A., Valavanis, K., and Yalsin, A., 2007, “Vision based target tracking and collision avoidance for mobile robots”, *Journal of Intelligent Robotic Systems*, 48, pp. 285-304.

VR Juggler, http://www.vrjuggler.org/, visited on Mar 2008.

Walter, B. E., 2003, “Virtual reality aided teleoperation”, Master’s Thesis, Iowa State University.

Walter, B. E., Knutzon, J. S., Sannier, A. S and Oliver, J. H., 2004, “VR aided control of UAVs,” 3rd AIAA Unmanned Unlimited Technical Conference, Workshop and Exhibit, Paper No. AIAA 2004-6320.

Wernli, R. L., 1982, “Robotics undersea”, *J. of Mechanical Engineering*, 104.

Wegner, R., 2003, “Balancing robotic teleoperation and autonomy in a complex and dynamic environment”, Thesis, University of Manitoba.

Wormell, D., Foxlin, E., 2003, “Advancements in 3D interactive devices for virtual environments”, Joint International Immersive Projection Technologies /Eurographics Workshop on Virtual Environments.

Zitnick, C., and Kanade, T., 1999, “A cooperative algorithm for stereo matching and occlusion detection”, Tech. report CMU-RI-TR-99-35, Robotics Institute, Carnegie Mellon University.