An Immersive Virtual Environment for Teleoperation of Remote Robotic Agents for Everyday Applications in Prohibitive Environments

Alireza Tavakkoli*
Department of Computer Science and Engineering
University of Nevada, Reno

Brandon Wilson†
Department of Computer Science
University of Houston-Victoria

Matthew Bounds‡
Department of Computer Science
University of Houston-Victoria

Abstract

There are currently important limitations in allowing for a more efficient man-machine collaboration in environments hostile to human presence. An immersive and intuitive user interface has the potential to bridge the gap between the human and the robot he/she is tasked to operate in the remote environment. In this work, we propose a novel architecture that allows for collecting large amount of sensory data to build models of the world and its inhabitants and present this information to the teleoperator of the robot. This environment maintain interfaces that are intuitive to the operator and accurately represents the robot's real-world state and environment. The proposed game client is capable of handling multiple users, much like a traditional multiplayer game, while visualizing multiple robotic agents operating within the real world. A set of user studies are conducted to validate the performance of the proposed architecture compared to traditional tele-robotic applications. The experimental results show significant improvements in both task completion time and task completion rate over traditional tools.

Index Terms: Human-centered computing—Interaction paradigms—Virtual reality—; Embedded and cyber-physical systems—Robotics—External interfaces for robotics

1 Introduction

The tele-operation of robotic platforms plays a vital role in space exploration, military reconnaissance, undersea operations, robotic surgery, training of personnel, and search and rescue operations in unsafe locations. Despite the push for more autonomous robotic agents, tele-operation will still be necessary as a ”default mode” for users, both to fix and to prevent errors caused by autonomous behavior as well as to boost the user's trust of the robot agent [3]. In addition, the task of robot tele-operation will become more commonplace as more and more viable robotic platforms become affordable and mainstream [5]. However, this can be a difficult task with traditional robot control schemes due to information overload for the end-user with complex scenes [6].

In order to provide enough information to operators to effectively and efficiently control a robot remotely, the end user must have access to real-world visual odometry, positional and environment map data, as well as any laser range finder, sonar, or other sensor data used by the robot for obstacle detection and localization. In addition, this data is further increased when multiple collaborative robots are considered. Traditional robot control schemes employ simple user interfaces to display information as a Heads-Up Display (HUD) [3]. This traditional user interface is used to both offer information about the robot's current state as well as the state of its environment.

*e-mail: tavakkol@unr.edu
†e-mail: wilsonbj@uhv.edu
‡e-mail: boundsm@uhv.edu

Games are traditionally thought of as a non-scientific medium of entertainment [7]. However, under the hood games are an immersive medium rendered through 3D virtual environments. In addition, these mediums are becoming more and more immersive through the advent of Virtual Reality (VR) Head-Mounted Displays (HMD's). Furthermore, these HMD's are becoming more and more commercially available [4]. VR will play an increasingly important role in tele-operation due to the synergy between robots and VR technology [2]. VR provides the user with increased immersion and enables the user to interact directly with their environment in an intuitive way. In addition, the robot provides a source of force feedback to any given tele-operation system. In theory, as an immersive and 3D medium capable of intuitive user interaction, a VR-enabled game is an ideal medium for the tele-operation of remote robotic platforms. However, there are many problems that must be addressed in order to utilize a game as a user interface for robot tele-operation. Below, we present the main contributions of this paper that addresses the outlining challenges of an effective immersive virtual environment useful for teleoperation of remote robotic agent.

1.1 Contributions

Fig. 1 shows the big picture behind the proposed system. In order to enable this system, we proposed four main contributions as outlined below:

Game Engine Integration: A range of new functionalities including, computer vision, robotics control frameworks, and parallel processes in support of the entire system are introduced and added to the base Unreal Engine 4 game engine in order to enable controlling of remote physical agents.

Heterogeneous Architecture: This architecture provides easy integration of robot clients and their unique robot interfaces into a GPU-accelerated High Performance Computing (HPC) server and an end-user dynamic and immersive virtual reality. The architecture allows for the offloading of computationally-intensive tasks to the HPC server, which facilitates communications between the end-user's UE4 game client and the real-world robot platform in order to place users virtually in the same environment as the robot to interact with both the robot and it's environment. User input is then translated to real-world actuators on the robot platform.

Networking Communication: A crossplatform networking API, called CrossSock has been developed as a result of this work [1], as a header-only and cross-platform API. CrossSock allows for the integration of a breadth of robotic platforms as well as the integration with all of the platforms that the proposed architecture supports for the end-user interface.

Dynamic Level Generation: The proposed VR-Telerobotic platform is designed to operate in a fully observable environment that has been scanned beforehand. We automatically generate walls and rooms for the end-user to explore. To add immersion, an In-Game Editor (IGE) has been provided in the end-user game interface for the manipulation of the world.

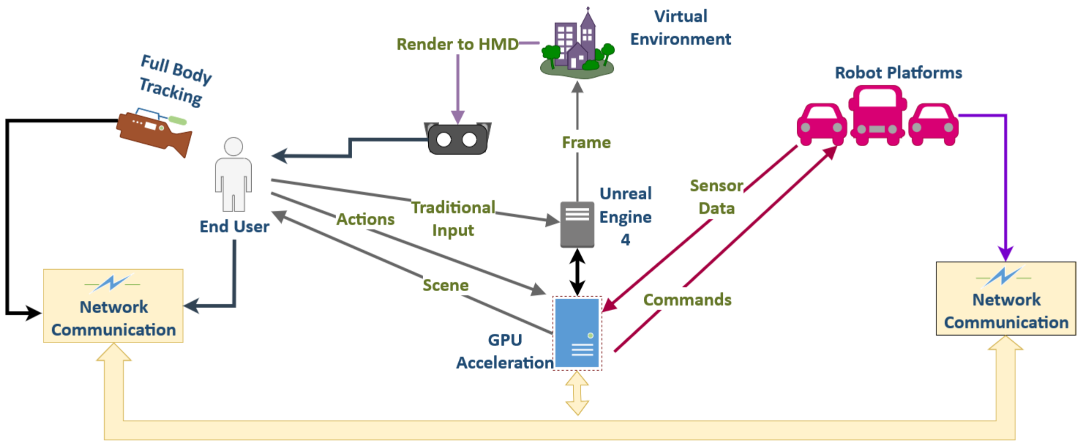


Figure 1: The big picture: The proposed architecture is comprised of two real and one virtual environment, bridging the gap between the remote robot and its teleoperator.

2 The Proposed Implementation

2.1 The CrossSock Networking API

As discussed earlier, in order to provide a framework for robot platforms a networking system needs to be utilized as a common language between the robot platforms, the centralized HPC server, and the end-user game clients.

To meet these challenges, the CrossSock networking library has been developed and made open to the public [1]. CrossSock was developed based on Berkeley sockets, which is dependent only on system calls. CrossSock supports both the Windows-based WinSock and UNIX-based POSIX libraries and wraps them under a single API, allowing for cross-platform source code. In addition, the CrossSock library is type-safe and object-oriented, which provides a more user-friendly environment for developing network-enabled applications.

CrossSock is a header-only library. As such, it allows for easy integration into any new or existing project. CrossSock also provides several system utilities and a high-level packet implementation. Finally, CrossSock provides an event-based client-server architecture for easily developing network applications from scratch. The low-level socket classes and the high-level client-server classes are separate. Therefore, developers can include only those CrossSock features they need for their projects.

The current CrossSock implementation employs simple, adjustable timers to control the rate at which various data is updated to the server and client machines/computers. It does not, however, dynamically adjust these time values based on current network bandwidth usage. CrossSock monitors all connected entities and ensures that they are alive/responsive, allowing for client re-connections within a specific time window. But when a client is connected and requests or broadcasts data, it is sent in a greedy fashion to the target as soon as possible. In a future version of CrossSock, network bandwidth and latency handling, along with encryption, is planned.

2.1.1 Client-Server Architecture

The proposed architecture is split into three major network components. a High-Performance Computing (HPC) server provides GPU-enabled services. Robotic agents are included in the architecture through the Robot Operating By ITself (ROBIT) client framework. Finnaly, an end-user game client is provided via Unreal Engine 4 (UE4), which may include multiple ROBIT actors.

2.2 Proposed System Results

In this section we present views of the proposed architecture in action, starting with normal usage in an office hallway shown in Fig. 3(a). In this figure, the PatrolBot is being driven manually down a hall. The environment is known, and is visualized around the robot. The spheres displayed in the virtual environment represent the real-world readings from the robot in real-time from its LRF and sonar sensors. In addition to the environmental data, the current 'throttle' (normalized speed) of the robot is displayed behind it as a percentage. The robots name is shown hovering above its current location.

Fig. 3(b) shows the optional distance sensor (LRF and sonar) visualizations as a raycast in the virtual environment from the robot to its final location. This visualization can be enabled on the current robot whenever the proximity warnings are not enough for manual piloting. Next, Fig. 4(a) demonstrates the Architecture's ability to automatically path the robot from its original location to any reachable location in the mapped environment. In this figure, red waypoints show the path's segments, and the final green waypoints represents the robots goal.

Obstacle visualization is shown in Fig. 4(b). The circled section is an unknown obstacle not reflected in the map data, which is visible on the stereo camera. In the current implementaiton of the proposed architecture, this type of unknown obstacle is visualized as free-floating proximity warnings. If an obstacle is of known origin (e.g. a bench or other type of encountered objects), the mesh for the object is placed within the environment.

In certain applications, the environmental conditions are outside of the operators' control. Moreover, there may be scenarios in which parts or all of a sensory system may fail whilst the robot and its operator are performing a task. One such scenario is the sudden loss of visual condition or camera input. The proposed architecture has significant benefit over traditional control mechanisms in such situations, as shown in Fig. 5. Here, the robot is navigating a maze in the real-world environment. During this operation, the lights are on, but suddenly are turned off is shown. In traditional applications, such loss of sensory data would amount to the entire operation's failure. However, since this environmental condition does not alter the structure of environment, human operator takes control of the robot's operation within the Virtual Environment to bring the task to completion.

2.3 Human Performance and Usability Study

To validate the benefits and usability of the proposed architecture, a human usability study is designed, in which human subjects are

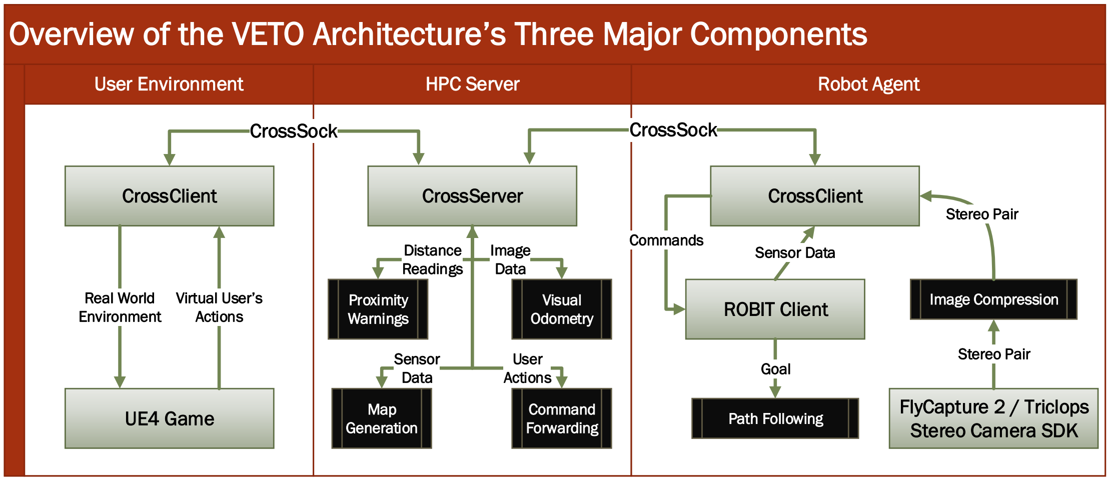
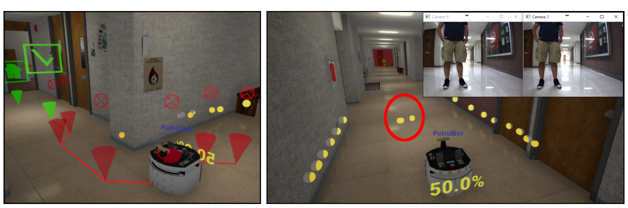


Figure 2: An overview of the three major components of the proposed architecture of Virtual Environment for Tele-Operation (VETO).

(a) (b)

Figure 3: A robot being driven automatically, and visualization of the laser range finder sensor on the robot (right).



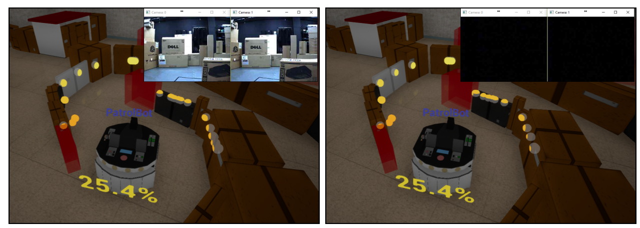
(a) (b)

Figure 4: A robot being driven automatically with the proposed architecture through path planning and following(left), and an unknown obstacle visualized as a proximity warning(right).

asked to perform two structured tasks remotely via a robotic agent, as described below.

The Navigation Task: In the first task, called *Navigation*, the subject is asked to navigate a robot through a maze. There are several physical and non-physical hazard zones populating the maze. The subject are given 20 minutes to complete the navigation. The time it takes to completely navigate the maze, as well as the number of physical and non-physical (hazard zone) hits are recorded for each subject.

The Exploration Task: In the second task, called *Exploration*, the subject is asked to explore the office corridors to find three symbols placed randomly within the environment. Again, the subject, is given 20 minutes to find all three symbols. The time to complete the task, as well as the number of symbols found by the subject are recorded.



(a) (b)

Figure 5: The virtual environment, as seen with the lights on (left) and lights off (right).

3 Methods

3.1 Study Population

Institutional research ethics approval was obtained prior to the study's implementation. There are 25 subjects in this round of the study. Participants are among students, staff, and faculty at the University of Houston-Victoria. 44% of the participants are female and 56% are male. 36% of participants are younger than 24 years, 40% are between 25 and 34, and 24% are older than 35 years. The study participants are quite diverse, including African American (12%), Asian (24%), Hispanic (32%), and White (24%) ethnic groups. About 8% of the participants did not declare their ethnicity. Participants' highest degree earned was High School (28%), Associate (16%), Bachelor's (28%), Master's (8%), and Doctoral (20%).

3.2 Experiment Setup

A prospective, randomized, repeated measures study design was used in which participants complete 2 sessions of the study: (1) Virtual Reality (VR) treatment session that utilized the proposed VR architecture to operate a remote robot, and (2) Regular Operated Vehicle (ROV) control session that utilized regular user interfaces and a monitor to control the robot. In order to counterbalance the study, participants were randomly assigned one of the two sessions first and asked to returned to participate in the other session after a period of 30 days has passed. The period between sessions is set to be more than 30 days to reduce the treatment effect on participants.

In each session participants were asked to complete the following

(a) (b)

Figure 6: (a)- The ROV session view while the subject is performing a navigation task. (b)- The VR view as seen by a subject performing an exploration task.

task:

Navigation: The subject is asked to navigate a robot through a maze populated by several physical and non-physical hazard zones within 20 minutes. Dependent variables are the time it takes to completely navigate the maze (T_{nav}), the number of physical hits (h_{phys}), and the number of non-physical hazard zone hits (h_{haz}).

3.3 Instrument

We designed a 5-question instrument to validate the performance of the proposed VR-mediated teleoperation framework and to compare its performance with traditional user interfaces. Questions **Q1**, **Q2**, and **Q5**, marked with *, were designed to validates the proposed system with respect to traditional interfaces. As such, we hypothesize that there will be no statistically significant difference in responses for these questions. Questions **Q3** and **Q4**, marked with †, were designed to show the benefit of the proposed interfaces compared to traditional teleoperation interfaces. We hypothesize that VR-mediated interface score significantly lower on **Q3** and significantly higher on **Q4**.

Q1***:** The interfaces were intuitive for me to use.
Q2***:** The interfaces were user friendly.
Q3†**:** I had trouble completing the tasks with the remote robot.
Q4†**:** The quality of the visual feedback was good.
Q5***:** I believe the robot completed all of the tasks that I had delegated or commanded using available interfaces.

3.4 Statistics

We evaluated the performance of the immersive virtual reality interfaces with traditional telerobotic interfaces using both inferential and descriptive statistics. For the inferential statistics we used a two-way analysis of variance (ANOVA). These analyses validated the reliability of the proposed immersive VR interfaces with the traditional teleoperation systems and their superior performance.

In order to validate the proposed VR-mediated teleoperation interface, we designed the proposed 5-item instrument to compare traditional teleoperation interfaces with the proposed architecture. Q1, Q2, and Q5 aim to distinguish usability and reliability of the proposed interfaces compared with traditional interfaces. We evaluate the following three hypotheses:

H1: There is no statistically significant difference on Q1 answers for the ROV and VR conditions.
H2: There is no statistically significant difference on Q2 answers for the ROV and VR conditions.
H3: There is no statistically significant difference on Q5 answers for the ROV and VR conditions.

Table 1: Validation of VR-mediated User Interface for Telerobotics ($N = 48$).

	VR		ROV		ANOVA		
	M	SD	M	SD	F	P	d
Q1	4.375	0.679	4.292	1.085	0.094	0.760	1
Q2	4.333	1.014	4.333	0.580	$1e^{-14}$	1.000	1
Q3	1.418	0.428	2.167	1.362	7.543	0.009	1
Q4	4.208	0.607	3.333	0.754	13.506	0.001	1
Q5	4.792	0.259	4.625	0.418	0.984	0.326	1

Table 2: Validation of VR-mediated User Interface for Telerobotics ($N = 48$).

	VR		ROV		ANOVA		
	M	SD	M	SD	F	P	d
Time	630	151	750	218	4.925	0.031	1
Hit$_p$	0.13	0.34	0.88	1.33	7.179	0.012	1
Hit$_h$	0.08	0.28	2.375	1.64	45.683	$1e^{-8}$	1

In order to demonstrate the effectiveness of the proposed VR-mediated teleoperation interface over traditional interfaces. We evaluate the following five hypotheses:

H4: There is a statistically significant difference on Q3 answers for the ROV and VR conditions.
H5: There is a statistically significant difference on Q4 answers for the ROV and VR conditions.
H6: There is a statistically significant difference in navigation time between the ROV and VR conditions.
H7: There is a statistically significant difference in the number of physical hits between the ROV and VR conditions.
H8: There is a statistically significant difference in the number of hazard hits between the ROV and VR conditions.

4 Discussions

Table 1 and Table 2 present the results of both the inferential and descriptive statistical analyses performed on both the VR and the ROV conditions.

4.1 Interface Validation

The proposed immersive VR interfaces for teleoperation are valid as compared with traditional interfaces in intuitiveness, usability, and friendliness. In particular, we found that there were no statistically significant difference between the VR-mediated and traditional interfaces based on Q1, Q2, and Q5, i.e., participants responded that both interfaces were interfaces were intuitive for to use and user friendly, and that robot the completed all of the tasks using both types of interfaces.

4.2 Interface Effectiveness

The proposed immersive VR interfaces were statistically superior in terms of performance and effectiveness compared to the traditional interfaces. In particular we found the five hypotheses (H4-H8) to be validated:

For H4, a repeated measures single-factor analysis of variance (ANOVA) revealed a statistically significant difference by interface design for difficulty completing tasks, $F = 7.543, p = .009$. VR interfaces scored significantly lower than traditional interfaces on the user having difficulty completing tasks.

For H5, a repeated measures single-factor analysis of variance (ANOVA) revealed a statistically significant difference by interface design on quality of visual feedback, $F = 13.506, p = .001$. VR interfaces scored significantly higher than traditional interfaces on the quality of visual feedback.

For H6, a repeated measures single-factor analysis of variance (ANOVA) revealed a statistically significant difference by interface design for completion time, $F = 4.925, p = .031$. VR interfaces scored significantly lower than traditional interfaces on the task completion time.

For H7, a repeated measures single-factor analysis of variance (ANOVA) revealed a statistically significant difference by interface design on the number of physical hits, $F = 7.179, p = .012$. VR interfaces scored significantly lower than traditional interfaces on the number of times that the operator hit a physical object while operating the robot.

For H8, a repeated measures single-factor analysis of variance (ANOVA) revealed a statistically significant difference by interface design on the number of hazard zone hits, $F = 45.683, p = 10^-8$. VR interfaces scored significantly lower than traditional interfaces on the number of times that the operator hit a hazard zone while operating the robot.

5 Conclusions and Future Work

This paper presented an integrated architecture for an interactive and immersive virtual reality environment for tele-robotics and telepresence applications. The proposed system utilizes a two-tier client-server architecture comprising of a computational server and a multitude of heterogeneous clients ranging from the virtual reality clients to robotics remote operational clients. A human usability study is designed to evaluate and validate the performance of the proposed architecture with the current robotic tele-operation tools. The study shows improvements in both the speed of task completion as well as the task completion accuracy.

The design of the proposed architecture is extensible, and as such supports a range of applications and research projects. One of the future direction of this work is to create a mechanism to utilize the sensory data from the robotics agents to build the virtual environments in real-time. In addition, we are investigating more intuitive means of interaction with the remotely operated robots by mapping natural hand and body gestures onto the operational modes of the robotic agents via a hierarchical activity-intent recognition model.

Acknowledgments

This material is based upon work supported by the U.S. Army Research Laboratory and the U. S. Department of Defense under grant numbers W911NF-15-1-0024 and W911NF-15-1-0455. This support does not necessarily imply endorsement by the DoD.

References

[1] Brandon Wilson. Crosssock. *gitHub*, 2017.

[2] G. C. Burdea. Invited review: the synergy between virtual reality and robotics. *IEEE Transactions on Robotics and Automation*, 15(3):400–410, 1999.

[3] J. Y. Chen, E. C. Haas, and M. J. Barnes. Human performance issues and user interface design for teleoperated robots. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(6):1231–1245, 2007.

[4] R. A. Earnshaw. *Virtual reality systems*. Academic press, NY, 2014.

[5] R. Nickerson. Pioneering the personal robotics industry. In *IEEE International Conference on Technologies for Practical Robot Applications, 2009*, pp. 179–185. IEEE, IEEE, NY, 2009.

[6] C. W. Nielsen, M. A. Goodrich, and R. W. Ricks. Ecological interfaces for improving mobile robot teleoperation. *IEEE Transactions on Robotics*, 23(5):927–941, 2007.

[7] D. Oblinger. Simulations, games, and learning. *ELI White Paper*, 1(1), 2006.