A VIRTUAL REALITY-BASED TELEOPERATION SYSTEM FOR THERAPISTS TO DELIVER ROBOT-MEDIATED INTERVENTIONS TO INDIVIDUALS WITH ASD USING HUMANOID ROBOTS

by

ROMAN KULIKOVSKIY

A thesis submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL AND COMPUTER ENGINEERING

2021

Oakland University
Rochester, Michigan

Thesis Advisory Committee:

Wing-Yue Geoffrey Louie, Ph.D., Chair
Shadi G. Alawneh, Ph.D.
Khalid Mahmood Malik, Ph.D.

To my mother and father

ACKNOWLEDGMENTS

First, I would like to thank Yan Chernikov for his YouTube series on C++, graphics programming, and game engine development. Without those, this project would be significantly harder.

Next, I would like to thank my supervisor, Dr. Louie, for his guidance and expertise in conducting research and help with writing. Also, I want to thank Megan Sochanski, Ala'aldin Hijaz, and Dr. Korneder for their contribution to the experiment portion of this work. As well, I want to thank committee advisors Dr. Alawneh and Dr. Malik.

Roman Kulikovskiy

ABSTRACT

A VIRTUAL REALITY-BASED TELEOPERATION SYSTEM FOR THERAPISTS TO DELIVER ROBOT-MEDIATED INTERVENTIONS TO INDIVIDUALS WITH ASD USING HUMANOID ROBOTS

by

Roman Kulikovskiy

Adviser: Wing-Yue Geoffrey Louie, Ph.D.

Socially Assistive Robots (SARs) have demonstrated success in the delivery of interventions to individuals with Autism Spectrum Disorder (ASD). To date, these robot-mediated interventions have primarily been designed and implemented by robotics researchers. It remains unclear whether therapists could independently utilize robots to deliver therapies in clinical settings. In this thesis, we conducted a study to investigate whether therapists could design and implement robot-mediated interventions for children with ASD. Furthermore, we compared therapists' performance, efficiency, and perceptions towards using a Virtual Reality (VR) and kinesthetic-based interface for delivering robot-mediated interventions. Overall, our results demonstrated therapists could independently design and implement interventions with a SAR. They were faster at designing a new intervention using VR than a kinesthetic interface. Therapists also had similar performance to delivering in-person interventions when utilizing VR to deliver interventions with the robot. Therapists reported moderate workload using the VR interface and perceived VR to be usable but with room for improvement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES—Continued

LIST OF ABBREVIATIONS

SAR	Socially Assistive Robot
ASD	Autism Spectrum Disorder
RMI	Robot-Mediated Intervention
HRI	Human-Robot Interaction
VR	Virtual Reality
HMD	Head Mounted Display
SDK	Software Development Kit
ROS	Robot Operating System
GUI	Graphical User Interface
DOF	Degrees of Freedom
ABA	Applied Behavioral Analysis
NASA	National Aeronautics and Space Administration
BCBA-D	Board-Certified Behavior Analyst-Doctoral
IRB	Institutional Review Board
SUS	System Usability Scale
NASA-TLX	NASA Task Load Index

INTRODUCTION

A major application area for Socially Assistive Robots (SARs) is providing therapeutic treatment to individuals with Autism Spectrum Disorder (ASD). According to the Center for Disease Control, 1 in 54 individuals are identified with ASD [1]. ASD is a condition that affects an individual's social, emotional, adaptive and communication skills [1]. Early intervention for children with ASD has long-term positive impacts for these individuals [2], making it a current focus for SAR research. In general, these robot-mediated interventions (RMIs) have demonstrated positive outcomes for individuals with ASD [3]-[10].

While research on SARs for the delivery of therapies to individuals with ASD has primarily focused on those with ASD, the end-users (e.g., therapists) operating the robots have not received the same level of exploration [11]. End-user perceptions are important to address because if they do not feel they have the capability and/or knowledge to operate the technology, it will not be used. A study with experienced and future professionals demonstrated positive feedback and interest in using SARs for therapies. However, the participants felt they did not have the knowledge to use the technology [12]. The study in [12] was limited because end-users were provided only a brief demonstration of the robot's capabilities but did not operate the robot. Furthermore, RMIs have primarily been designed and implemented by researchers [13]. It remains unclear whether end-users could design interventions and operate robots to deliver an intervention to an individual with ASD. It is important to enable end-users to operate SARs so they can evaluate the clinical applicability of RMIs [13], [14]. Hence, there is presently an open opportunity to identify: 1) whether end-users could be trained to operate a SAR to deliver new therapies and 2) what tools would be most effective for end-users to operate a SAR.

Herein, the focus of this work is on end-users operating SARs via teleoperation as it is a valuable and commonly utilized tool for: 1) rapidly prototyping, evaluating, and implementing interventions delivered by a SAR; 2) data collection for development of the autonomy of a robot; and 3) evaluating models for human-robot interactions (HRIs) [15]-[18]. Putting these technologies directly in the hands of end-users, via teleoperation, will ensure that RMI can be evaluated for their clinical validity, which addresses an existing gap in RMIs [13]. Teleoperation also serves as a way for end-users to understand the capabilities and limitations of SARs as they are developed for interventions.

Studies have also shown that due to the rapidly changing needs of individuals with ASD, end-users require the control of the robot to be simple, fast, flexible, and usable [19]. Furthermore, we focus on therapies requiring a humanoid SAR to communicate both verbally and nonverbally as a majority of clinically relevant therapies for individuals with ASD require human-like verbal and nonverbal communication skills (e.g., imitation therapy and emotion recognition) [20]-[22].

The aims of this thesis were:

1) The development of a VR-based teleoperation system for humanoid SARs to allow therapists to be trained to design and operate SARs to effectively, as well as efficiently deliver new therapies to individuals with ASD.
2) Investigation of therapist performance, efficiency, and perceptions with VR-based teleoperation and commercial kinesthetic teaching interfaces for

designing RMIs and teleoperating a SAR to deliver the intervention, while focusing on human-like verbal and nonverbal communication.

3) The tool for future work in collecting data on the way therapists would run the intervention as if they were a robot. Therefore, potentially providing an opportunity for development of better autonomous robot-mediated interventions.

In the first chapter we look at why robots are used to teach children with ASD, the way VR teleoperation is currently used for physical manipulation tasks, and how social tasks are different. In the second chapter we present a VR-based teleoperation system that allows for the control of humanoid SAR by a user’s body movements in VR. Chapter three then focuses on a study that evaluates the VR-based teleoperation system in comparison to conventional control of humanoid SARs. Namely, it focuses on hypotheses and variables that were compared, participant recruitment, the intervention that was used, experimental design and procedure, measures that were collected, and results.

CHAPTER ONE

LITERATURE REVIEW

In this chapter, first we will look at how robots are currently being used to teach children with ASD and the drawbacks of using pre-scripted motions for those purposes. Then we will look at an alternative to pre-scripting motions, Virtual Reality teleoperation, that has been used in other fields of robotics, namely for physical manipulation tasks. Further we will discuss the difference between the physical manipulation tasks and social tasks and how social tasks could also benefit from VR teleoperation.

1.1 Robots Teaching Children with ASD

To date, studies focusing on using SARs for the delivery of therapies to individuals with ASD have predominantly consisted of robots using pre-scripted social behaviors while the robot is teleoperated [3], [4], [15], [17], [23]. Pre-scripted robot behaviors are advantageous in the delivery of therapies for ASD because they are consistent. Kinesthetic teaching is a common method for pre-scripting motions for robot behaviors [24]. Kinesthetic teaching means that robots' motions are created by physically guiding robot through those motions first and remembering that trajectory. However, pre-scripting social behaviors via kinesthetic teaching can result in repetitive behaviors because users must demonstrate prior to an interaction all the behaviors a robot should utilize. Furthermore, kinesthetic teaching can lead to unnatural body language and can be a difficult process for creating social behaviors because interpersonal communication is an automatic process that humans find difficult to describe explicitly [25], [26]. Professionals working with this population have indicated such repetitive and unnatural robot behaviors must be used with care because the goal of therapy is not to teach individuals with ASD to interact with a robot but to enable them to generalize a skill to human-human interactions [27], [28]. This is because humans produce large variations when encoding social behaviors with the same semantic meaning [29]. Therefore, robots should model/simulate human-like variation during communication if they are to effectively teach children with ASD social, emotional, and communication skills which transfer to human-human interaction. Virtual reality (VR) motion tracking could be a potentially valuable technology for enabling therapists to immerse themselves in a human-robot interaction and naturally teleoperate a robot's motions in real-time to deliver therapies to children with ASD.

1.2 Using VR for Physical Manipulation Tasks

VR motion tracking systems have already demonstrated success for enabling users to teleoperate industrial robots [30]-[32] and humanoid robots [33]-[35] for physical manipulation tasks. In general, the VR systems developed for industrial robots enabled users to teleoperate the position and orientation of the end-effectors for the robots while the user was immersed in the robot's point of view. A study with users also demonstrated that participants were more successful as well as efficient in completing physical manipulations when using a VR-based system then a joystick and perceived the VR to be easier to use as well as intuitive [32]. VR-based systems have also been developed to enable users to provide joint-based control of a humanoid robot's upper torso while again being immersed in the robot's point of view. Namely, these systems first either utilize skeleton tracking to determine a user's joint positions [33], [34] or infer joint positions according to limited user pose information (i.e., hand, elbow, and head position as well as forearm vector) [35]. Joint positions are then utilized to compute user arm joint angles and mapped to a humanoid robot's joint angles so that a user can teleoperate the robot by moving his/her own body. A study with novice users demonstrated that joint-based VR control of a humanoid robot for pick & place and pouring tasks was preferred over a kinesthetic teaching system. Furthermore, users had a lower perceived workload and were more efficient with performing tasks requiring two arms (i.e., pouring) [35].

1.3 Social Tasks

Social tasks focus on clearly communicating a message to people through both verbal, nonverbal, and affective cues. Teleoperating a robot to communicate effectively requires the simultaneous control of a robot's voice as well as the dynamics, positions, and orientations of all the robot's joints. This is because nonverbal communication is a whole-body effort [29]. In contrast, users teleoperating a robot for a physical manipulation task will primarily focus on the position and orientation of the robot's end-effectors [30]-[35]. Prior works on evaluating users' experiences utilizing VR systems to teleoperate a robot have also only focused on physical tasks where the environment is static and can only be altered by the robot [32], [35]. However, during a social task other people participating in the social interaction have their own beliefs, affect, goals, and intentions and, therefore, requires a robot to rapidly adapt to the needs of the individual. Such adaptation is important because humans will have negative attitudes towards a robot if it fails to follow social norms or produces socially inappropriate behaviors within the context of an interaction [36]-[39].

To summarize, the current methods for delivery of RMI are predominantly consisted of robots using pre-scripted social behaviors. However, pre-scripted social behaviors have several drawbacks, such as repetitiveness and unnatural body language. Virtual reality (VR) motion tracking could be a potentially valuable technology addressing those drawbacks. Currently VR motion tracking systems has been teleoperation of robots and had a lot of success in physical manipulation tasks. Yet social tasks are significantly different from the physical tasks. Therefore, this thesis will mainly focus on using VR for social tasks.

CHAPTER TWO

VR TELEOPERATION SYSTEM

In this chapter we will look at the VR teleoperation system design and implementation. First the general system overview and setup will be discussed. Then the VR rendering application will be discussed in terms of its architecture, rendering pipeline, the way 3D models are rendered, the way UI was done, and how the input from VR was handled. After, the robot system and the interface to it will be discussed with. Lastly the inverse kinematics solver that was used to compute joint angles will be presented.

2.1 General System Overview and Setup

A VR-based teleoperation system was developed for teleoperating a humanoid robot to perform social tasks while immersed in the robot's perspective. We focused on humanoid robots because they can exhibit human-like verbal and nonverbal communication. This enables individuals with ASD that interact with robots in RMIs to better transfer learned skills to real life human-human contexts. The VR teleoperation system developed is presented in Figure 1. The system consisted of three main modules: 1) a VR rendering application; 2) humanoid robot; and 3) a kinematics solver. The VR-based teleoperation interface consisted of a head mounted display (HMD), hand-held controllers, and a microphone so that a user could teleoperate the robot. The VR rendering module generated the graphical output to be displayed on an HMD to provide visual feedback to the user and create an immersive teleoperation experience from the perspective of the robot. The kinematics solver utilized the sensed HMD and hand-held

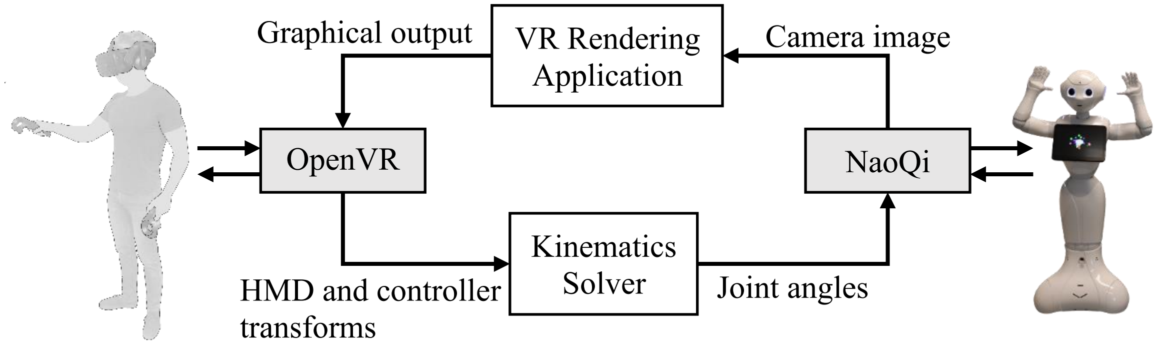


Figure 1. VR-based teleoperation system

controller positions to determine the joint angles of the user so that they could be mapped to the humanoid's joints.

The system also recorded simultaneously the robot's sensory data, primarily camera feed, and joint angles generated by the teleoperator. This allowed for generation of a dataset that could later be utilized in future works.

2.2 VR Rendering Application

2.2.1 System Overview and Usage

The VR-based teleoperation interface was developed to take user inputs for teleoperating a robot and immersing a user in the robot's perspective. The OpenVR SDK and SteamVR runtime were used for all VR software development [40], [41]. The commercially available HTC Vive HMD, hand-held controllers, and microphone were used as input devices by the user to intuitively control the robot's head, arms, and voice, respectively. Namely, users can naturally move their head while the HMD tracks the position and orientation of the user's head. Users could also move their arms while the hand-held controllers would track the user's hand position and orientation. These head, arm, and hand teleoperation inputs were provided to the kinematics solver module to map user motions to robot motions. The user could speak at the robot's location through a microphone. The HMD was used to immerse a user within a social interaction from the perspective of the robot. Audio from the robot's microphone was transmitted to the user's headphones and the egocentric view of the visual information perceived by the robot was transmitted to the HMD. This is achieved by the VR renderer module generating graphics from the RGB camera in the robot's head to be displayed to the teleoperator in the HMD.

The VR rendering application was developed in two parts. First the OpenGL-based rendering engine library was developed. Then that library was used to develop the resulting application. We chose to use the developed library to build a ROS application to allow for better utilization with different robots and robotics' tools. The developed application was a ROS node that was subscribing to an image topic and was publishing joint angles for the robot. Outside of ROS environment it also was providing the image into each eye of the headset and obtained controller information from SteamVR [41], as well as displayed information on the regular screen.

To use the application, a user starts SteamVR [41] which is a runtime used for virtual reality and then runs the developed ROS node. After that, a desktop window will appear displaying a point of view (from the screen) into the VR world. The user could use a keyboard to look around. The screen view can be seen in Figure 2. The purpose of this screen was to provide a feedback about the system to people who are not in the VR itself but for example observing it, or as in this case, were running the experiment. This feature was particularly useful for training stage as the scientist conducting tutorial could point on the virtual screen in VR world from the computer itself.

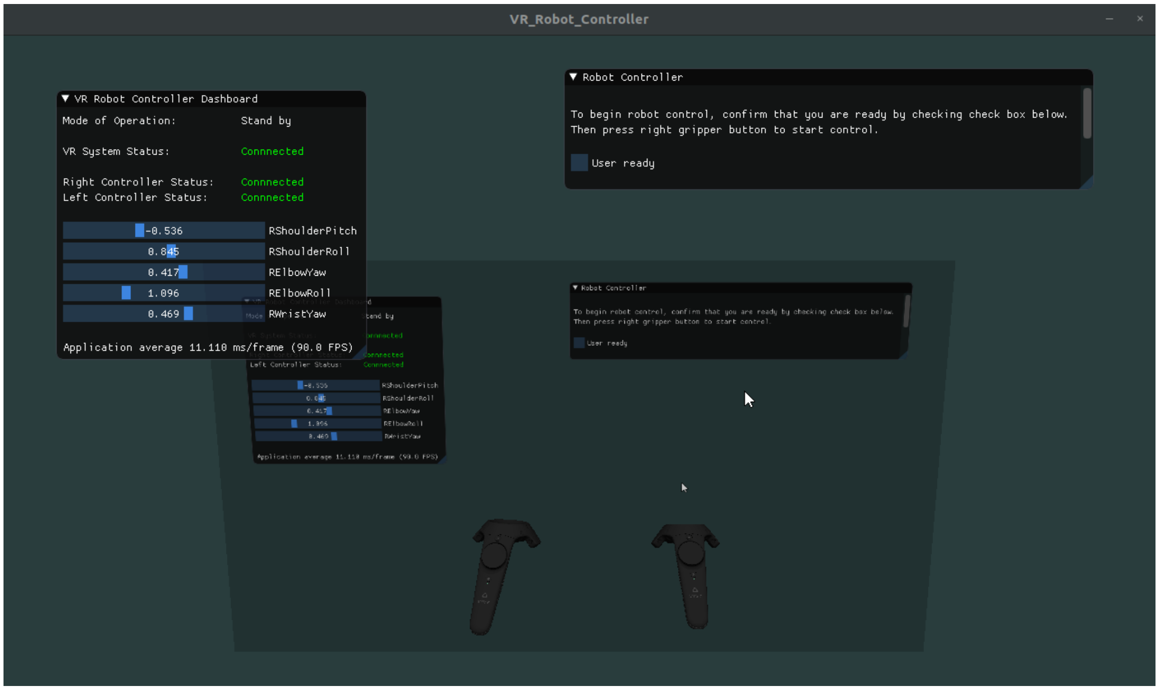


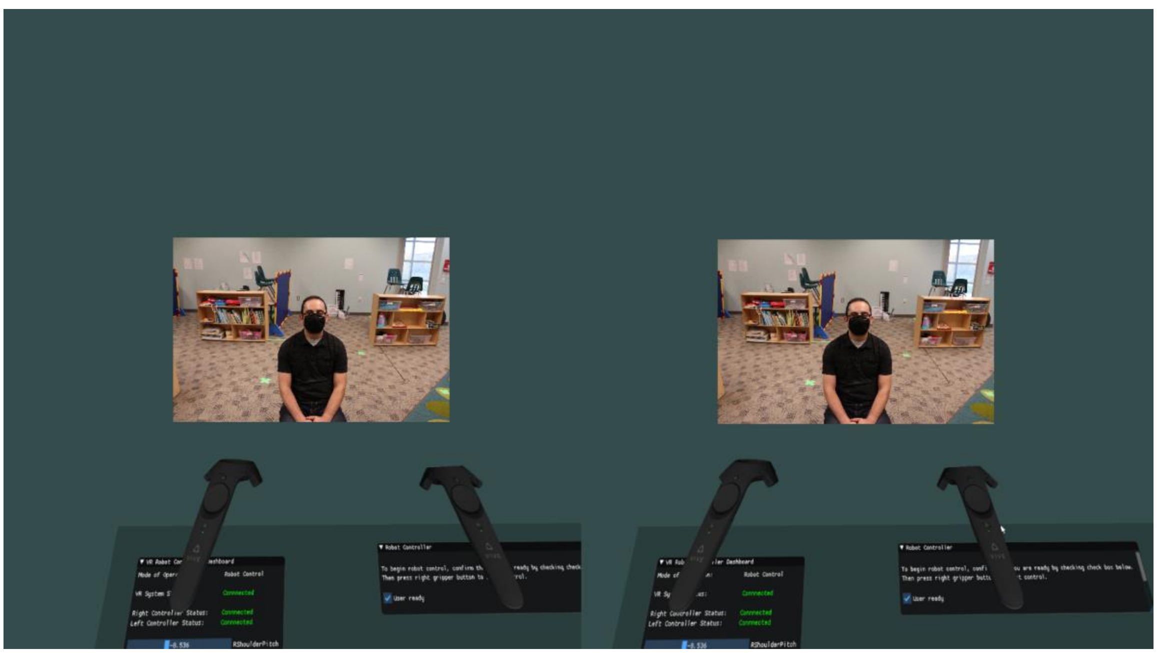
Figure 2. Screen view of the system

When the user would put the headset on, he/she appears to be in the virtual environment. Inside of this environment the user was able to observe the two controllers that were rendered based on their position and a menu-like screen, which was fixed to a specific position. Using controllers, the user choses the desired settings on the screen and could start using the application to control the robot.

Application was made to work in two different modes of operation. The first mode of operation was a “Stand by” mode. In the “Stand by” mode the communication through ROS were cut-off. This allowed users to adjust settings and get into stands as well as prepare for controlling robot. The screen in the “Stand by” mode also displayed information on controllers that were connected as well as robot’s joint angles. The view presented to user in the “Stand by” mode can be observed in Figure 3(a). The second mode of operation was “Robot Control” mode. In this mode of operation, the menu like screen would shift down to clear the view and the new screen would appear that would display the image from the topic that application is subscribed to. This can be observed in Figure 3(b). Unlike the menu-screen, this screen was moving with the head of the user, always staying at front of their eyes. This was done because the head is also being used to control the robot and therefore it was important that user could see what robot sees in all head positions. In this mode, the communication on publishers and subscribers was resumed and computed joint angles were sent to the specified ROS topic as well as monoscopic view of the RGB camera from the robot’s head was received.

A monoscopic view was utilized because unlike physical manipulation tasks depth perception is not a requirement for non-contact socially interactive tasks. The hardware required for stereoscopic views are not typically available on commercial social robot platforms and as a result require additional or specialized cameras with careful spatial arrangements. Furthermore, stereoscopic views have the potential to increase the risk of cybersickness if the rendered views in the HMD do not match a user’s physiology [42]. Cybersickness could be amplified if there is latency when the user is controlling the robot’s head motions [43]. To reduce these discrepancies, a virtual screen was created in the VR environment so that the robot’s RGB camera video stream could be displayed on it. The virtual screen was always a fixed distance from the user while he/she was immersed in the robot’s view. If the user moved his/her head the virtual screen would

(a.)



(b.)

Figure 3. VR view for left and right eye for two modes of operation. (a.) "Stand-by" with its menu screen. (b.) "Robot Control" with robot's camera feed following user’s eyes.

follow the user's head motions in the virtual environment so that the screen remained centered in his/her view. This technique of projecting a robot's view onto a screen instead of directly into each of the user's left or right eye views has been shown to reduce motion sickness and improve immersion during robot teleoperation [44].

2.2.2 System Architecture

The VR Rendering Application consisted of two main components. First was an OpenGL-based rendering engine library that provided abstraction layer over rendering of 3D models, GUI interfaces, and VR inputs and outputs. And the second was an interface to ROS that provided calculation and further communication of joint angles to the robot and camera feed from the robot. Both components were later combined in development of a ROS application which in turn was a VR Rendering Application (Figure 4).

The design of rendering engine library was done with an idea that it will be used for development of applications. For those purposes, an interface class *Application* was created that provided 4 abstract methods implementations for which were required to run the application. The methods were *Init* for initialization of application, *Run* which implemented the main running loop of the application, *Shutdown* which implements methods for proper shutting down of application, and *OnEvent* which was used to dispatch events. Inside of application users of the library were open to create instances and use many different classes that provided low- and high-level abstractions over rendering and user interface tools. The list of classes and their brief description may be observed in the Appendix B. Utilizing those, the application was put together that allowed to render models of controllers, accessing VR devices' data, rendering to screen and into the headset, and providing user interface.

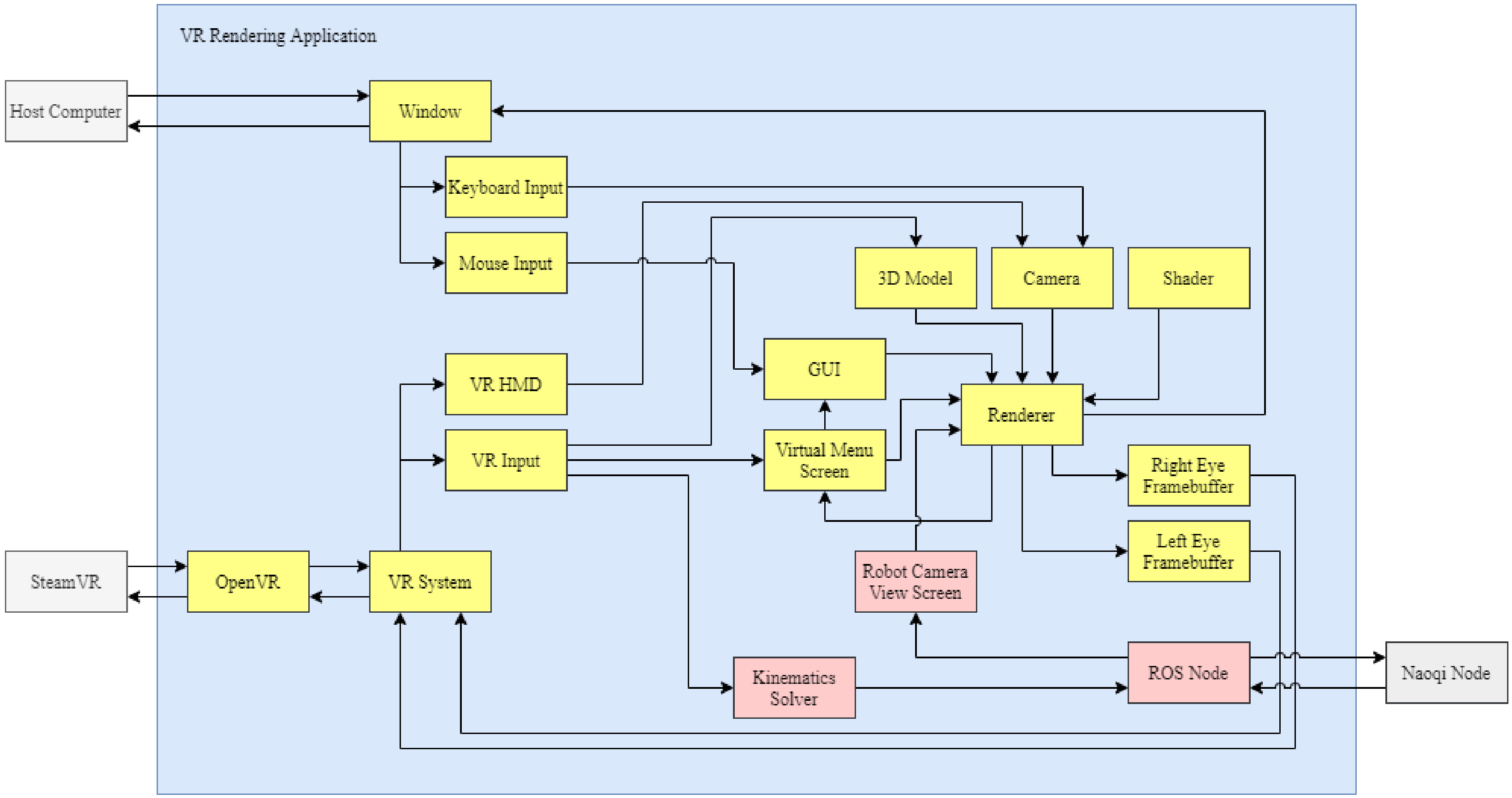


Figure 4. Approximate view of VR Rendering Application architecture

The second portion of the system was the interface to ROS. This was designed as a *RosNode* class that when initialized would create a ROS node. The *RosNode* had methods to set transforms for head, and two hands. Using those transforms joint angles were calculated with the model described further in section 2.4. Then on each update of application the joint angles were published to the topics that were used by interfaces to the robot.

2.2.3 Rendering Pipeline

For this work everything that has been displayed on the screen and inside of the VR headset had to be rendered or drawn. In the VR headset each eye gets its own screen which displays its specific perspective, as well image is also displayed on the screen. Additionally, the refresh rate in the VR headset should remain around 90Hz to keep person immersed and avoid nausea. For those reasons, a GPU is often used to accelerate graphics rendering. To access the VR hardware OpenVR SDK was used which allows for pretty much all mainstream graphics rendering APIs for GPUs. We decided to go with OpenGL since it is well known and allows for cross-platform operation. Since the system was implemented to work with robot's and ROS most often it will require to run on Linux system, therefore it would be important to have a common cross-platform API.

With OpenGL as rendering API the rendering engine library had to implement and abstract out the OpenGL rendering pipeline. In short, OpenGL rendering pipeline is a series of operations that turns data provided to OpenGL into an image that appears on screen. The pipeline itself takes as an input the raw byte data on vertices fields of which later are specified to via API calls to specify a so-called vertex array object. The vertex itself is a data point which could consist of information specified by user of the API. In the case of this system vertex consisted of *x*, *y*, and *z* coordinates; *r*, *g*, and *b* color values; *u*, *v* texture mapping coordinates; and *x*, *y*, and *z* components of a normal vector (that could be used, for example, reflections and lighting). Then the vertex array object went through the pipeline. The pipeline consisted of the following key stages:

1. *Vertex processing* where on each *vertex shader* was applied. The *vertex shader* was a program written in GLSL programing language, compiled, and linked to run on the GPU. The main role of the shader was to apply transform model, view, and perspective matrices to each of the vertices to know where each should appear on the screen. In this stage tessellation and geometry shading would take place as well.
2. *Vertex Post-Processing* in which primitive assembly (triangle) as well as clipping (depth clamping).
3. *Rasterization* where fragments are generated.
4. *Fragment Shading* where now for each fragment the *fragment shader* was applied. The role of *fragment shader* is mainly in specifying the color that fragment. This includes coloring, textures application, light calculations for each fragment, etc.
5. *Per-Sample* operations was the last step in the pipeline which included things like *scissor test* which discarded fragments that are outside of specified region; *blending* which was used to create slightly transparent screen in this system; and many other tools that OpenGL provides.

The pipeline is also shown in the Figure 5. In the developed rendering engine library, the pipeline was abstracted out. User of the library was only required to provide a

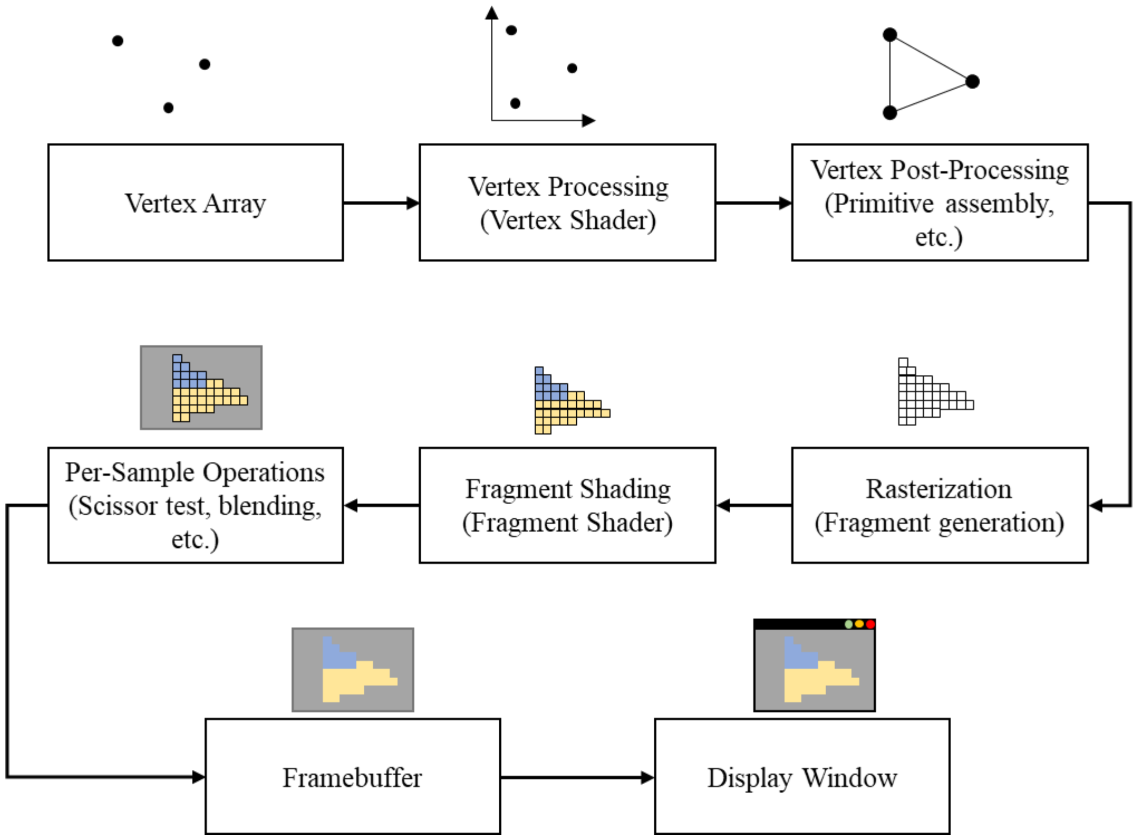


Figure 5. OpenGL rendering pipeline

3D model that they would like to draw, then use the default shader that was provided with the library or write their own and issue a draw call on each iteration of an application when the model was supposed to be drawn. The image then was drawn to a left and right eye framebuffer from which OpenVR [40] was taking it into VR hardware display.

<u>2.2.4 3D Model Rendering</u>

Since use of VR was one of the key points of this thesis, all the renderings had to be done in 3D space and with perspective of each eye. Without that, the view in the headset would be distorted and challenging to see. Therefore, everything that was visible in this system inside of the VR headset, was either a 3D model or displayed on a 3D model. 3D model itself usually consists of collection of coordinates that form certain primitives, like triangles, with respect to the model's origin, and some other auxiliary data. From there to the display of model on the screen or in the framebuffer for an eye there are usually three steps that must be taken. Since we know the coordinates of vertices in the reference frame of the model, we need to apply the transformation matrix to each of those vertices to find out where it should be in the coordinate frame of the world, we are observing it in. The most common name for that transform matrix is "model matrix". From there, all vertices of the scene should be displayed in the coordinate frame of a camera which is done by applying "view matrix" transform to the vertices. Since we have two eyes to display to, the two different view matrices are being used one for each eye. Lastly, the "projection matrix" should also be applied to the objects to account for projection view. Since both view and projection matrices are more hardware specific, they were received from OpenVR [40] API to allow for use with any of the supported by SteamVR [41] devices. In the result for each of the generic vertex *v* in the 3D model, the new coordinate v' was obtained by:

$$v' = P * V * M * v \tag{2.1}$$

Where *P* is perspective matrix, *V* is a view matrix, and *M* is a model matrix. Figure 6 shows how the series of transformations described above turn a 3D model's coordinates in its model space into the screen space.

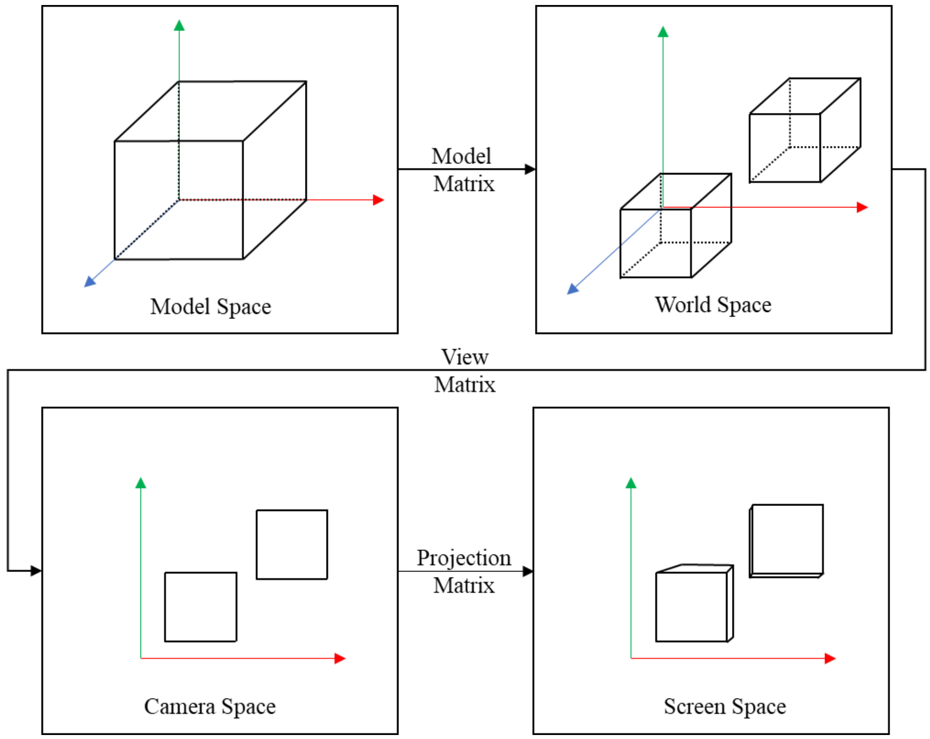


Figure 6. Model View Projection visualization for 3D model rendering.

2.2.5 UI Screen Rendering

To provide the user interface inside of VR space we created a virtual screen inside of the VR space and displayed UI on it. Providing interface in such manner allowed for more intuitive way of using the UI since people are used to clicking on things that are on the screen. As well, this allowed to use separate, already developed library for GUI drawing, called imgui [45]. Using imgui [45], the UI was created and drawn to the framebuffer. Later that framebuffer was used as a texture for the screen 3D model that in turn was displaying the screen. The same framebuffer was also overlayed over the image that was being drawn on the computer screen so observers could observe more clearly the UI. To provide the input to the UI the VR input had to be handled.

2.2.6 VR Input Handling

The main input source from the VR system came through OpenVR [40] in terms of transform matrices of devices with respect to certain origin and states of buttons. The application was constantly checking on the needed inputs. The inputs from VR system were used not only to control the robot but also to interface the system.

One of such interfaces was the viewpoint for each eye for rendering. Based on the position of the head and transform from head to eye for each eye, the view matrices were obtained for rendering. This allowed for immersive experience where each eye served as a camera in the scene.

The second way that input was used to interface the system was to create a pointer for the screen that displayed a UI. This allowed users to use controller as a laser pointer on the UI screen and use the buttons on the controller to click on the UI. This was done by computing the intersection coordinates between the virtual screen and a ray casted from the controller. Transformation matrix of controller with respect to origin ($T_{Origin}^{Controller}$) was provided by the input from VR system. And the model matrix, which is also a transformation matrix, of the screen with respect to origin (T_{Origin}^{Screen}) was defined by user during initialization. The mode transform pointed to a reference frame of the screen in the upper left corner. The transformation and the setup may be observed in the Figure 7. Using those transformations, the transform matrix of the controller in the reference frame of the screen was computed:

$$T_{Screen}^{Controller} = (T_{Origin}^{Screen})^{-1} * T_{Origin}^{Controller} \qquad (2.2)$$

From the transformation matrix of controller in the screen reference frame, position vector (p) and direction vector (v) of z axis were extracted

$$v = p - T_{Screen}^{Controller} * [0 \quad 0 \quad -1] \tag{2.3}$$

Which later were used to compute the coordinates of the intersection with the *xy*-plane of the virtual screen's coordinate frame

$$(x, y) = p - \left(\frac{p_z}{v_z}\right) * v \tag{2.4}$$

Lastly, the transform matrices for head, and both controller also served as an input to an inverse kinematic solver that is being discussed in section 2.4.

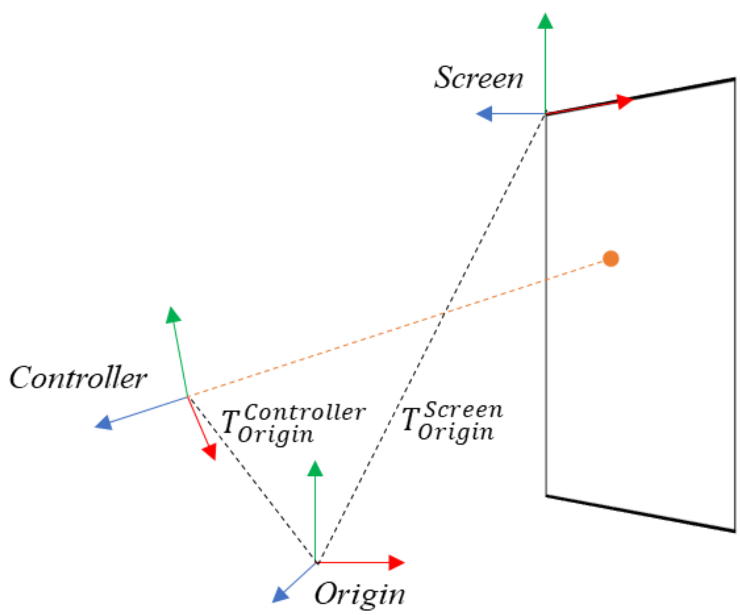


Figure 7. UI pointer.

2.3 Robot

2.3.1 Robot Overview

Our VR system was developed to teleoperate robots with a humanoid upper torso, and we utilized the Pepper robot (Figure 8) as an example. Pepper can exhibit human-like upper body movements using two degrees of freedom (DOF) in the neck, five DOF in each arm, and one DOF in each hand. The two DOF in the robot's neck provides yaw and pitch rotations of the head. The five DOF in each of the robot's arms allow shoulder pitch, shoulder roll, elbow yaw, elbow roll, and wrist yaw movements. The one DOF in each hand allows the robot to open and close its hands. The robot can monitor the individuals and environment around itself via an RGB camera and microphone sensor.

Figure 8. Pepper robot

2.3.2 Interfacing the Robot

To interface the robot a separate ROS application was developed. Application was designed to read joint angles to which robot should set its motors. To implement the application, the NaoQi SDK was used [45]. NaoQi is an SDK by Softbank Robotics for their robots Pepper, Nao, and Romeo. Therefore, the application was implemented separately to allow rendering application to be agnostic of robotic system that is being used.

2.4 Inverse Kinematics Model

2.4.1 Overview

The primary goal was to enable a user to naturally demonstrate motions and map these motions to a humanoid robot. The kinematics solver module enables a user to control a robot’s joint movements using an HMD and two hand-held controllers. The HTC Vive tracks and provides a transformation matrix for each of the devices relative to an origin frame (i.e., the ground). We used a modified version of the kinematics equations presented in [35] to map user motions to the robot’s motions based on the HMD and hand-held controller transformations. In Figure 9, the transforms that were provided as an input to the system, as well as an approximated shoulder transform and elbow point may be observed. Using the input provided, the inverse kinematics model was derived to compute joint angles for pepper. More detailed view of the Pepper joints can be seen in Figure 10.

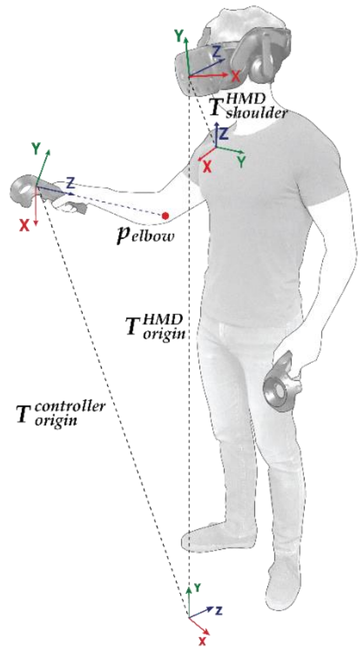


Figure 9. Transforms and coordinate frames used in kinematics calculations

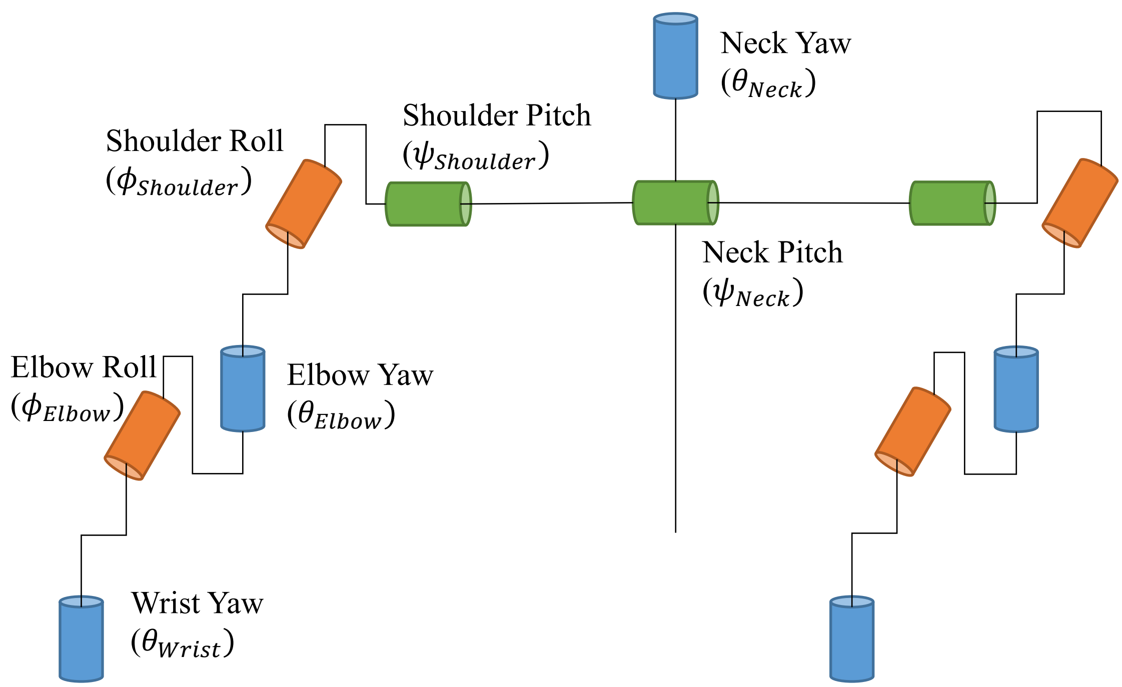


Figure 10. Pepper robot joints.

2.4.2 Equations

The user's head motions are mapped to the robot's head motions by controlling the robot's neck yaw (ψ_{Neck}) and pitch (θ_{Neck}). This is accomplished by directly mapping the yaw and pitch components from the transformation matrix of the tracked HMD (T_{Origin}^{HMD}) to the robot's neck yaw and pitch rotations:

$$\psi_{Neck} = \text{yaw}\left(T_{Origin}^{HMD}\right) \tag{2.5}$$

$$\theta_{Neck} = -\text{pitch}\left(T_{Origin}^{HMD}\right) \tag{2.6}$$

User arm motions were mapped to the robot arm motionsby first determining the elbow position of the user. Basedon our empirical evaluation, we approximated that the user'swrists remained fixed when holding a hand-held controllerdue to the ergonomics of the controller, and for adultstheir elbow position was offset by 35 cm in the positive z direction from the controller:

$$p_{Controller}^{Elbow} = \begin{bmatrix} 0 & 0 & 0.35 \end{bmatrix} \tag{2.7}$$

The elbow position was the transformed into the shoulder reference frame where the shoulder pitch and roll were then calculated . The transformation matrix from the shoulder to the head was thaken from [35], which was derived from anthropomorphic measurements on adults. Formally, the HMD to shoulder transformation matrix ($T_{Shoulder}^{HMD}$) was defined as:

$$T_{Shoulder}^{HMD} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0.21 \\ 0 & 1 & 0 & 0.225 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} & R_o^h & & 0 \\ & & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.8}$$

where R_o^h is the rotation matrix of the HMD from the origin and used to ensure the shoulder frame was not affected by the rotation of a user's head.

The position of the elbow can then be transformed into the shoulder frame (p_s^e) by:

$$p_{Shoulder}^{Elbow} = T_{Shoulder}^{HMD} * (T_{Origin}^{HMD})^{-1} * T_{Origin}^{Controller} * p_{Controller}^{Elbow} \tag{2.9}$$

Given the position of the elbow in the shoulder frame, the shoulder pitch (θ_s) can be calculated by:

$$\theta_{Shoulder} = \text{atan2}(-p_{Shouler,z}^{Elbow}, p_{Shoulder,x}^{Elbow}) \tag{2.10}$$

Prior to calculating the shoulder roll, the elbow position in the shoulder frame needs to account for the shoulder pitch rotation. The elbow position in the shoulder frame after the shoulder pitch rotation ($p_{Shoulder\prime}^{Elbow}$) can be determined by:

$$p_{Shoulder\prime}^{Elbow} = \begin{bmatrix} \cos\theta_{Shoulder} & 0 & -\sin\theta_{Shoulder} \\ 0 & 1 & 0 \\ \sin\theta_{Shoulder} & 0 & \cos\theta_{Shoulder} \end{bmatrix} p_{Shoulder}^{Elbow} \tag{2.11}$$

The shoulder roll angle ($\phi_{Shoulder}$) can then be calculated by:

$$\phi_{Shoulder} = \text{atan2}(-p_{Shoulder\prime,y}^{e}, p_{Shoulder\prime,x}^{e}) \tag{2.12}$$

To calculate the elbow angles, it was necessary to infer the direction of the forearm. We approximated the forearm to be in the $-z$ direction relative to the controller:

$$e_{Controller}^{Forearm} = [0 \quad 0 \quad -1] \tag{2.13}$$

We can then determine the direction of the forearm in the elbow frame (e_c^f) after the shoulder pitch and roll rotations by:

$$e_{Elbow}^{Forearm} = R_{Elbow}^{Shoulder} * R_{Shoulder}^{HMD} * \left(R_{Origin}^{HMD}\right)^{-1} * R_{Origin}^{Controller} * e_{Controller}^{Forearm} \tag{2.14}$$

where the rotation matrix between shoulder and elbow ($R_{Elbow}^{Shoulder}$) was defined as:

$$R_{Elbow}^{Shoulder} = \begin{bmatrix} \cos\phi_{S.} & \sin\phi_{S.} & 0 \\ -\sin\phi_{S.} & \cos\phi_{S.} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_{S.} & 0 & -\sin\theta_{S.} \\ 0 & 1 & 0 \\ \sin\theta_{S.} & 0 & \cos\theta_{S.} \end{bmatrix} \tag{2.15}$$

(where *S.* stands for *Shoulder*). Once the forearm direction in the elbow frame of reference we can then calculate the elbow yaw (ψ_{Elbow}) by:

$$\psi_{Elbow} = \text{atan2}(e_{Elbow,z}^{Forearm}, e_{Elbow,x}^{Forearm}) \tag{2.16}$$

The forearm direction in the elbow frame needs to account for the elbow yaw prior to calculating the elbow roll. The forearm in the elbow frame after the elbow yaw ($e_{Elbow'}^{Forearm}$) is:

$$e_{Elbow'}^{Forearm} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi_{Elbow} & \sin\psi_{Elbow} \\ 0 & -\sin\psi_{Elbow} & \cos\psi_{Elbow} \end{bmatrix} e_{Elbow}^{Forearm} \tag{2.17}$$

The elbow roll (ϕ_{Elbow}) then calculated by:

$$\phi_{Elbow} = \text{atan2}(e_{Elbow',y}^{Forearm}, e_{Elbow',x}^{Forearm}) \tag{2.18}$$

The wrists for the robot remained fixed ad user could control the fingers on the robot by depressing the trigger button on the hand-held controllers to have the robot make a fist.

CHAPTER THREE

VR TELEOPERATION SYSTEM EVALUATION

In this chapter the study that was designed for evaluation of VR teleoperation system will be discussed.

3.1 Hypothesis and Variable Compared

A user study with therapists from an Applied Behavior Analysis (ABA) clinic for children with ASD was conducted to evaluate whether therapists could plan, design, and implement a robot therapy. The developed VR teleoperation interface was compared with an existing kinesthetic-based interface, SoftBank Robotics Choregraphe, available for Pepper. The goal of the study was to evaluate therapist training time, efficiency, performance, and perceptions for the two methods of designing and teleoperating a robot to deliver an intervention. The hypotheses we evaluated include:

H1: Therapists require less time to plan and design an intervention utilizing VR than the kinesthetic interface.

H2: There will be a difference in the number of errors produced by a therapist during in-person, VR, and kinesthetic intervention delivery. Namely, therapists will produce the most errors with the kinesthetic interface.

H3: There will be a difference in the time it takes a therapist for in-person, VR, and kinesthetic intervention delivery. Namely, therapists will require the most time with the kinesthetic interface.

H4: Therapists will rate the VR interface to have a higher usability than the kinesthetic interface.

H5: Therapists will rate the VR interface to have a lower workload than the kinesthetic interface.

3.2 Participants

The participants in this study included therapists delivering the robot-based intervention and the children receiving those interventions. All participants were recruited from a university-based ABA autism clinic.

The inclusion criteria for therapists were: 1) working at an ABA clinic, and 2) no prior history of seizures with VR. We recruited ten therapist participants but only eight participants (one male and seven females) with ages ranging from 22-33 (μ=25.13, σ=4.05) completed the entire study. Two participants withdrew due to lack of availability. The participants were primarily female because a large proportion of individuals practicing in the field of ABA are female and the male to female ratio is representative of the population [46]. The inclusion criteria for the children were: 1) 3-8 years old; 2) a DSM-V diagnosis for ASD; and 3) has not mastered the skill of emotion recognition. In total, we had 4 child participants (3 males and 1 female) with an age range of 4-6 (μ=5,σ=0.95).

3.3 Emotion Recognition Intervention

A board-certified behavior analyst-doctoral (BCBA-D) developed an ABA emotion recognition intervention for the children to learn to recognize emotions only from an individual's body language, without facial expressions nor sound effects. We chose this intervention because one of the challenges faced by individuals with ASD is recognizing emotions [47] and recognizing emotions from body language is especially relevant during COVID-19 due to facial expressions being occluded by masks. This intervention also allows therapists to experience teleoperating the robot to socially interact with an individual using both verbal and nonverbal communication.

The interventions followed standard ABA clinical procedures and were broken down into three components. First, the therapist teleoperator would ask the child how he/she (i.e., the robot) is feeling while presenting an emotion using only the robot's movements. Initially, a vocal prompt of the correct emotion was provided until the child was able to respond correctly without the prompt. Second, the child would then be provided an opportunity to respond to the question. Third, the therapist teleoperator would then respond to the child with social praise if he/she answered correctly or follow-up with a prompt for the correct emotion if the child answered incorrectly or provided no response. This three-step sequence defines a single discrete trial and a complete intervention consisted of a total of nine trials. Each intervention aims to teach three different emotions by presenting each emotion three times (i.e. three trials) in a randomized order. Six emotions were chosen and placed into two groups. Group one included happy, scared, and tired whereas group two included sad, surprised, and angry. The grouping of emotions was chosen by the BCBA-D to balance the difficulty of the emotions between the groups.

3.4 Experimental Design and Procedure

A within-subjects experiment was designed for each therapist to be trained to utilize both interfaces to control Pepper and then independently planned, designed, and implemented a robot-based emotion recognition intervention. The experiment was reviewed by an IRB board and consent was obtained from all participants. The study was divided into three days for each participant: 1) the intervention delivered by the therapist in-person to the child; 2) the therapist designing the robot-based intervention and delivering a mock intervention with the two interfaces, and 3) the therapist delivering the intervention to a child by controlling the robot with the two interfaces. Participants were video recorded for post study analysis.

3.4.1 Commercial Teleoperation Method via Choregraphe

For the kinesthetic-based interface SoftBank Robotics Choregraphe was used. Choregraph is a desktop application that allows for development on SoftBank Robotics robots, one of which is Pepper. The application allows for more intuitive use than for example text-based programing and most likely would be the friendliest to a new user. Since therapists were not expected to have experience with robotics and/or programing this was the best commercially available user-friendly tool.

Choregraphe presents itself as a GUI interface (Figure 11) which could be used to connect to or simulate a robot; create animations, behaviors, and dialogs as well as play them on robot; monitor sensors such as a camera; and if needed augment the behaviors with python code. For the purposes of this study, we used only creation of animations and behaviors, as well as monitoring of sensors of this tool. Choreographe allows for creation of animations via kinesthetic teaching. To do so users must go into animation mode, create key frames, and then start moving robot the way they like robot to move. To turn the animation to the behavior, user would then have to record their voice in an audio file and add it to the animation. Then whenever the user would like to play the behavior on a robot, he or she would just select the behavior from the list and press the play button. The choregraph allows a user to observe the virtual robot that displays all of the motions done by actual robot as well as camera feed of the robot.

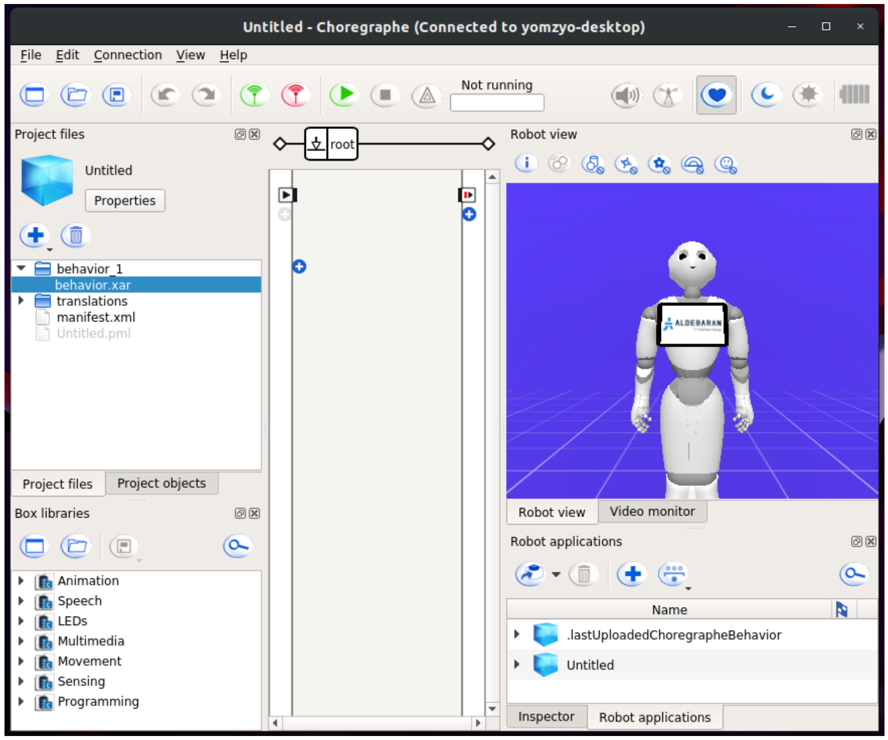


Figure 11. Choregraphe interface to design and run intervention using robot.

3.4.2 Day 1. Intervention without Robot

The goal of day 1 was to have therapists familiarize themselves with the intervention and video record the sessions so they could use it as a reference for when they control the socially assistive robot to deliver the intervention. Following typical clinic procedures for new interventions the therapists were provided instructions on the intervention design by the BCBA-D and randomly assigned a group of emotions to teach a child. There were six total emotions split among two groups of participants. The participants were split into groups at random, with group one delivering the emotions ‘happy’, ‘fear’, and ‘tired’, and the second group delivering ‘sad’, ‘surprise’, and ‘angry’. The grouping of emotions was chosen by the BCBA-D to balance the difficulty of the emotions between the two groups. The participants were then given a sample sheet of their three emotions containing pictures they could use as reference to generate their own body language for the emotion. Therapists were told that any facial expressions or sound effects accompanying the emotion were to be avoided when delivering the intervention. When the therapist was ready, he/she then delivered the emotion recognition intervention to a child with ASD.

3.4.3 Day 2. Training

The goal on day 2 was to train the therapists to utilize the two interfaces and allow them to experience the process needed to deliver a new intervention with a SAR. After being trained with each interface the participants planned and designed their own robot-based emotion recognition intervention. They then delivered a mock intervention (Figure 12) with an adult stand-in that simulated responses expected from a child. The order the interfaces were presented to each participant was counterbalanced.

3.4.3.1 VR Training

The training session for the virtual reality interface consisted of a researcher first explaining to the participant that the virtual reality equipment will allow the participant to perceive (i.e., hear or see) what the robot perceives and control the robot's arm, head movements, and speech. The researcher then used the virtual reality equipment to demonstrate him/her controlling the robot's head and arm movements. The participant then had the opportunity to practice controlling the robot's movements (Figure 13) while the robot faced a mirror. This provided the participant visual feedback while he/she was controlling the robot's movements. Once the participant indicated they were ready and

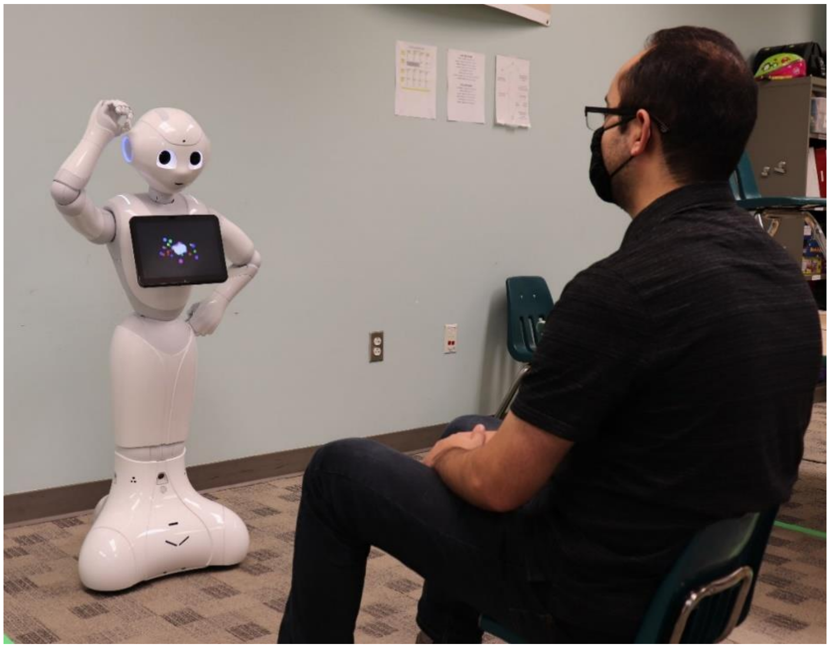


Figure 12. Mock intervention

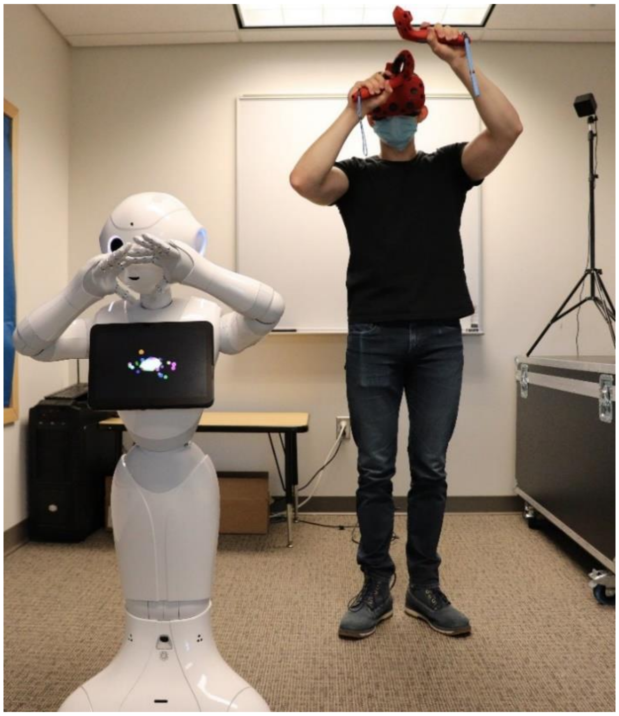


Figure 13. VR teleoperation training.

understood how to control the robot, he/she was asked to deliver the emotion recognition intervention in a mock therapy with an adult researcher.

3.4.3.2 Kinesthetic Training

The kinesthetic interface training session began with a walkthrough of Choregraphe. The participants were taught how to create a robot behavior (i.e., robot motion trajectories and/or speech) and were guided by the researcher through a creation of a sample behavior. Once confident the participant was asked to create robot behaviors for their emotion recognition intervention using Choregraphe without assistance (Figure 14). Participants then ran a practice intervention to confirm they had all behaviors to run an intervention and had the opportunity to add or change behaviors. A mock intervention was then run with a researcher. The participant could hear and see from the robot's perspective by wearing headphones and viewing a stream of the robot's camera from a monitor. The participant then conducted the intervention by selecting with a mouse the appropriate behaviors from the list of robot behaviors he/she created.

3.4.4 Day 3. Actual Intervention

The goal of day 3 was to have the participants use the two teleoperation interfaces to deliver the interventions they prepared on day 2 this time conducted with a child with ASD. The child receiving the treatment was chosen according to schedule availability. The order the interfaces were used by each participant was the same as on day 2.

3.4.3.1 VR Procedure

The intervention session for the virtual reality interface consisted of having a researcher starting the virtual reality interface and having the participant first practice their set of emotions to allow them to review the controls. Once the participant indicated

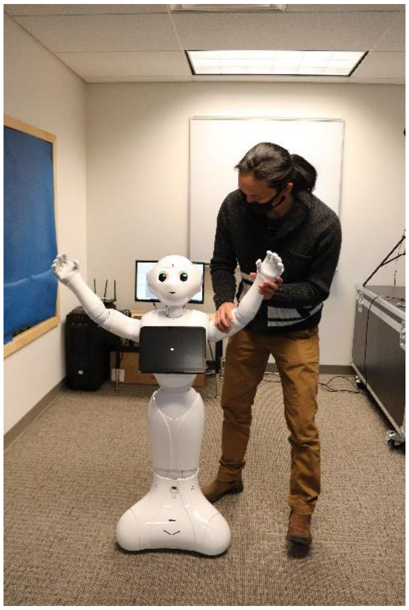


Figure 14. Kinesthetic teaching

they were ready for the intervention, the robot was moved to the room where the intervention will be held with the child, while the participant controls the robot remotely using the VR controls. The intervention consisted of nine total trials, three of each emotion with a randomized order, delivered as a typical SD-Prompt-Consequence ABA session.

3.4.3.2 Kinesthetic Procedure

The intervention session for the kinesthetic teaching consisted of having a researcher starting the Choreograph application and loading the pre-saved robot behaviors which the participant had created and saved in their own day two training session. Once Choreograph and the pre-saved robot behaviors were loaded, the robot was moved to the room where the intervention will be held with the child, while the participant controlled the robot by running the appropriate robot behavior. During the intervention, the participant could hear and see what the robot hears and sees through the Choreograph window. The intervention consisted of nine total trials, three of each emotion with a randomized order, delivered as a typical SD-Prompt-Consequence ABA session.

3.5 Measures and Data Analysis

Therapist performance and efficiency in planning, designing, and implementing an RMI was evaluated using three measures: intervention planning and design time, treatment integrity, and intervention time. These measures were used for the VR and kinesthetic interfaces and were defined as:

Intervention planning and design time refers to any time the participant put towards creating and/or practicing their intervention. For the kinesthetic interface, this consisted of the participant planning and creating their behaviors using Choregraphe. For the VR interface, it was the amount of time the participant practiced their intervention with the SAR in front of a mirror.

Treatment Integrity refers to the extent to which the intervention was implemented successfully [48]. In the emotion recognition intervention this was evaluated according to the correct application of the components of an ABA discrete trial. The components included: establishing attending, presenting an instruction, providing a required prompt, prompting at the correct time, re-presenting an instruction, providing a prompt, reinforcing correct responses, and ending the trial. Not all components are required in each trial because the requirement of a component is dependent on a child’s response. Treatment integrity was then calculated as the total number of correctly implemented components divided by the total required components and converted to a percentage. Treatment integrity was collected for interventions implemented with the child on days 1 and 3.

Intervention times refers to the time required to implement an intervention. Only behaviors that pertained to the intervention were included in measuring intervention time for both interfaces. Again, intervention time was only collected on days with the child, days 1 and 3.

Participant perceptions of usability and workload towards the interfaces were also measured using the System Usability Scale (SUS) [49] and NASA Task Load Index (NASA-TLX) [50], respectively. The System Usability Scale consists of 10 items with five possible response options that range from Strongly agree to Strongly disagree. The result of System Usability Scale was a 0-100 score. The score is not a percentage and should be considered in terms of percentile ranking. The NASA-TLX assesses workload score based on a weighted average of 6 subclasses. For each of the subclasses, participant had to estimate response on 21 graduation scale. The score produced by NASA-TLX is 0-100 where higher score means higher perceived workload. These post-task questionnaires were administered on days 2 and 3, after the participants utilized each interface to implement an intervention. In total, four questionnaires were administered to each participant. Open-ended questions were also administered to investigate participants' user experience with the interfaces. The list of open-ended questions can be seen in the Appendix B.

After collecting the data from all the therapist participants, a paired, two-tailed t-test was conducted to test hypotheses H1, H4, and H5 with the VR and kinesthetic interfaces as the conditions for each dependent variable. Prior to running the paired t-test, we confirmed the dependent variables were normally distributed using the Shapiro-Wilk test. A repeated measures ANOVA was conducted for H2 and H3 with the in-person, VR, and kinesthetic interventions as the conditions for the dependent variables. Shapiro-Wilk and Mauchly's tests were used to test for normality and sphericity, respectively. When sphericity was violated, a Greenhouse-Geiser correction was applied. An $\alpha = 0.05$ was set for all tests.

3.6 Results and Discussion

3.6.1 Efficiency

All participants were capable of designing and planning an RMI using both the VR and kinesthetic interfaces. On average, participants designed and planned the intervention faster with the VR interface (μ=523.4s, σ=235.9) than the kinesthetic interface (μ=1631.6s, σ=1068.5). There was a statistically significant difference in time required by therapists for designing and planning an intervention using the VR compared to the kinesthetic interface ($t(7)=-3.093$, p=0.017), which supports *H1*. This finding aligns with the short answer responses from participants stating that the kinesthetic design and planning were more labor intensive and required significant preparation. The comparison of average time to design and plan the intervention are shown in Table 1 and in Figure 15.

The participants required on average 83.4s (σ=22.4) to deliver the intervention in-person, 148.4s (σ=20.8) using VR, and 235.8s (σ=122.4) using the kinesthetic interface. There was a statistically significant difference between in-person, VR, and kinesthetic intervention delivery ($F(1.072,7.504)=10.491$, p=0.012). Post hoc tests with Bonferroni

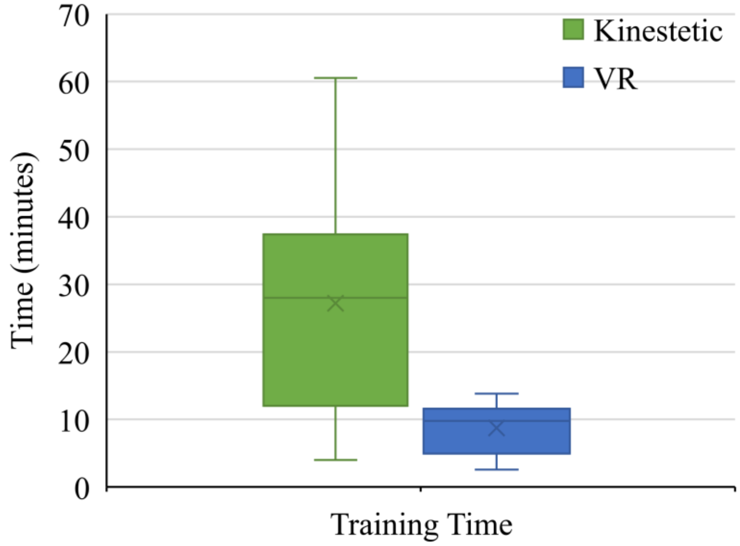


Figure 15. Training time with VR and Kinesthetic systems

corrections revealed there was a statistically significant difference in intervention time between in-person and VR (p=0.002), as well as in-person and kinesthetic (p=0.013). There was no statistically significant difference in intervention time between VR and kinesthetic delivery (p=0.247). Therefore, *H3* was not supported. The rejection of *H3* is likely explained by participants indicating they would need more practice with VR to use the robot more efficiently. Additionally, one survey indicated that it was more difficult to tell the system what to do instead of doing it themselves. The comparison of times to run the different types of interventions is shown in Table 2 and Figure 16.

3.6.2 Efficacy

The participants' treatment integrity while delivering the interventions in-person was 84.0% (σ=17.5). Therapists delivering the intervention to the child using the VR

Table 1. Average training time with VR interface and with kinesthetic teaching

VR Interface	Kinesthetic
523.4s	1631.6s

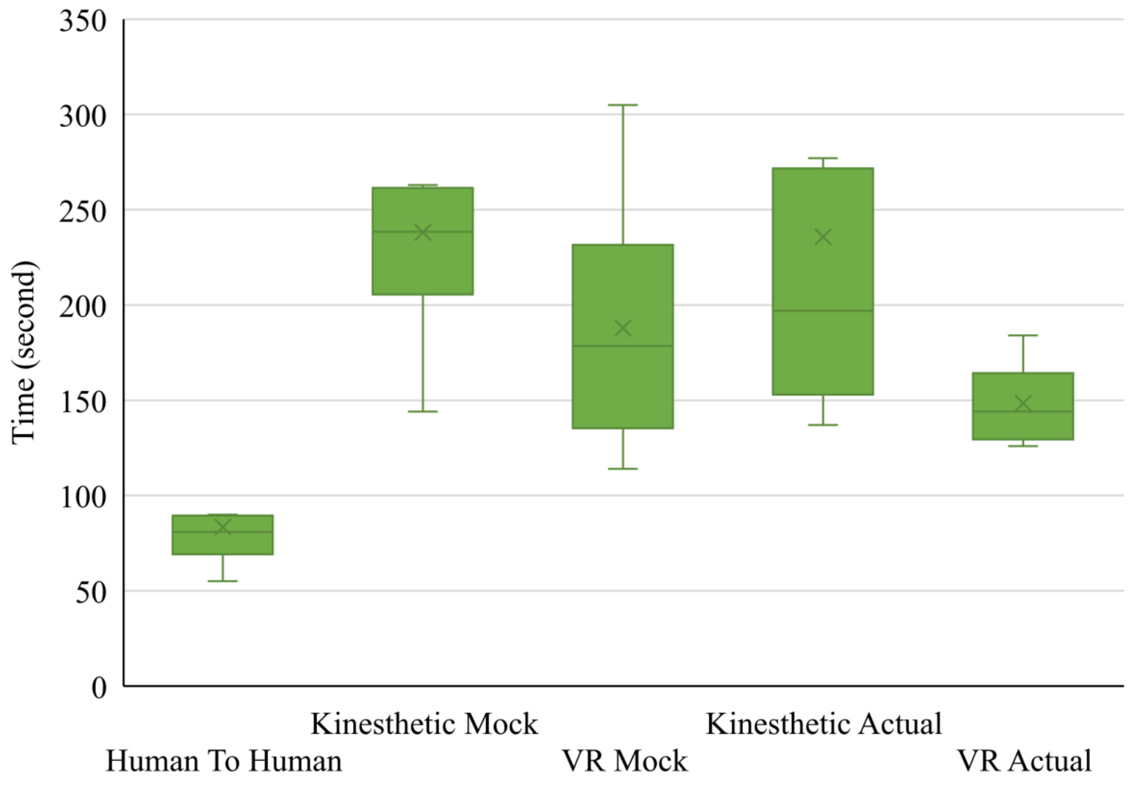


Figure 16. Time taken to run the intervention

interface to teleoperate the robot resulted in similar treatment integrity as in-person 83.7% (σ=12.7). When participants used the kinesthetic interface with the child there was a decrease in the average treatment integrity to 52.9% (σ=22.5). There was a statistically significant difference in treatment integrity between in-person, VR, and kinesthetic intervention delivery ($F(1.208,8.453)=15.374$, $p=0.003$). Post hoc tests with a Bonferroni correction revealed that the difference in treatment integrity between in-person and VR was not statistically significant (p=1.000), but between in-person and kinesthetic was statistically significant (p=0.021). The difference in treatment integrity between VR and kinesthetic intervention delivery was also statistically significant (p=0.008), which supports H2. Participants reported it was easier to implement the components of an intervention using VR and participants found it easier to adapt to a child's changing needs. The comparison of treatment integrity scores by different types of interventions is shown in Table 2 and Figure 17.

3.6.3 System Usability (SUS scale)

On average, participants rated the usability of the VR interface with a SUS score of 63.75 (σ=11.1) for the mock intervention, and 53.75 (σ=16.6) for the intervention with a child. Participants rated the usability of the kinesthetic interface with a SUS score of 52.19 (σ=13.5) for the mock intervention, and 44.38 (σ=10.5) for the intervention with the child. With a higher SUS score in the mock and real interventions with a child, the VR interface demonstrated better system usability. However, there was not a statistically significant difference between therapists' SUS scores for the interfaces during the mock ($t(7)=1.527$, $p=0.171$) or real interventions ($t(7)=1.309$, $p=0.232$). This suggests that *H4* was not supported in this study. A common challenge brought up by participants with the

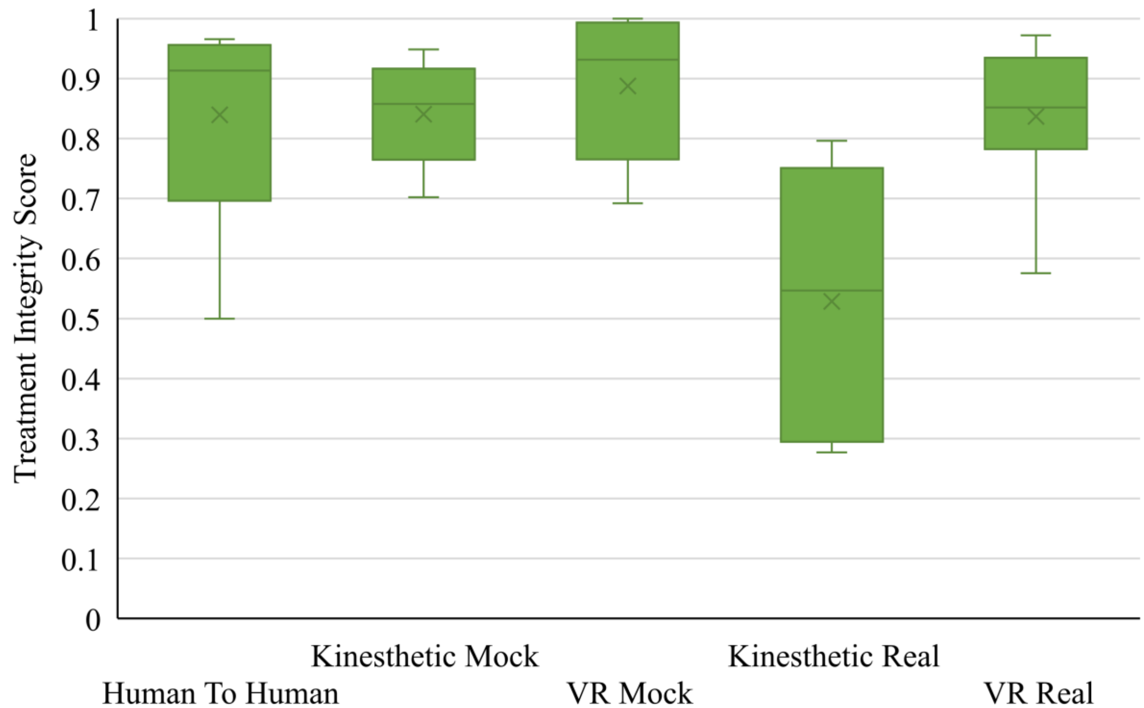


Figure 17. Treatment integrity scores

VR interface was their lack of body awareness and slight differences in embodiment. In the future, we plan to investigate the addition of a third person perspective of the robot to provide visual feedback and improve usability. The comparison of System Usability Scale scores by different types of interventions is shown in Table 3 and Figure 18.

<u>3.6.4 Perceived Workload (NASA-TLX)</u>

The workload between the VR and kinesthetic interface for the mock interventions were similar, but for the real interventions with the child, the VR interface had lower perceived workload. Participants, on average, rated the workload of using the VR interface with a NASA-TLX score of 55.67 (σ=17.9) for the mock intervention and 56.83 (σ=12.6) for the

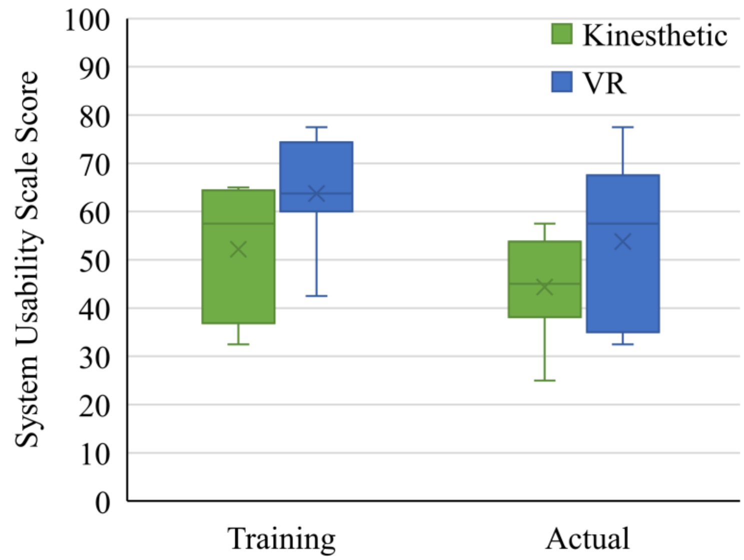


Figure 18. SUS scale score comparison

real intervention. Participants' average rating for workload using the kinesthetic interface was 59.67 (σ=12.1) for the mock intervention and 71.21 (σ=12.0) for the real intervention with the child. Participants perceptions on workload between the VR and kinesthetic interfaces during the mock intervention ($t(7)=-0.574$, p=0.584) and intervention with children with ASD ($t(7)=-1.784$, p=0.118) suggests that there was no statistically significant difference. Hence, *H5* was not supported. The workload scores for using the VR interface, for both the mock and real interventions, coincided with the median NASA-TLX global workload scores typically observed with robot operation tasks [51]. This is expected because therapists need to constantly adapt to the learning needs of a child which can result in high mental demand. The comparison of NASA-TLX scores by different types of interventions is shown in Table 3. As well workload contribution comparison for training is shown in Figure 19 and for real intervention in Figure 20.

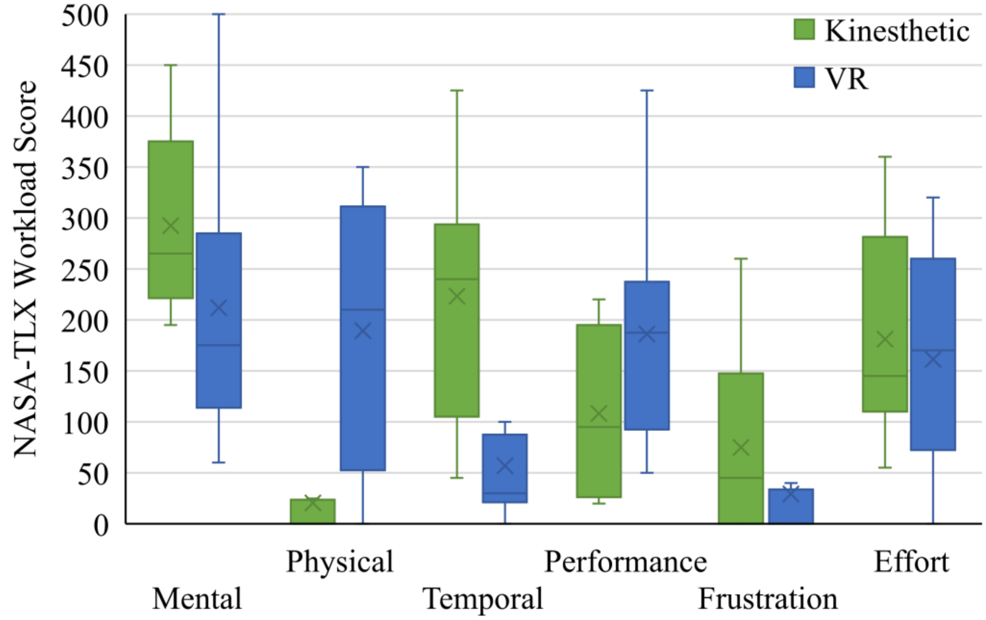


Figure 19. Training workload comparison

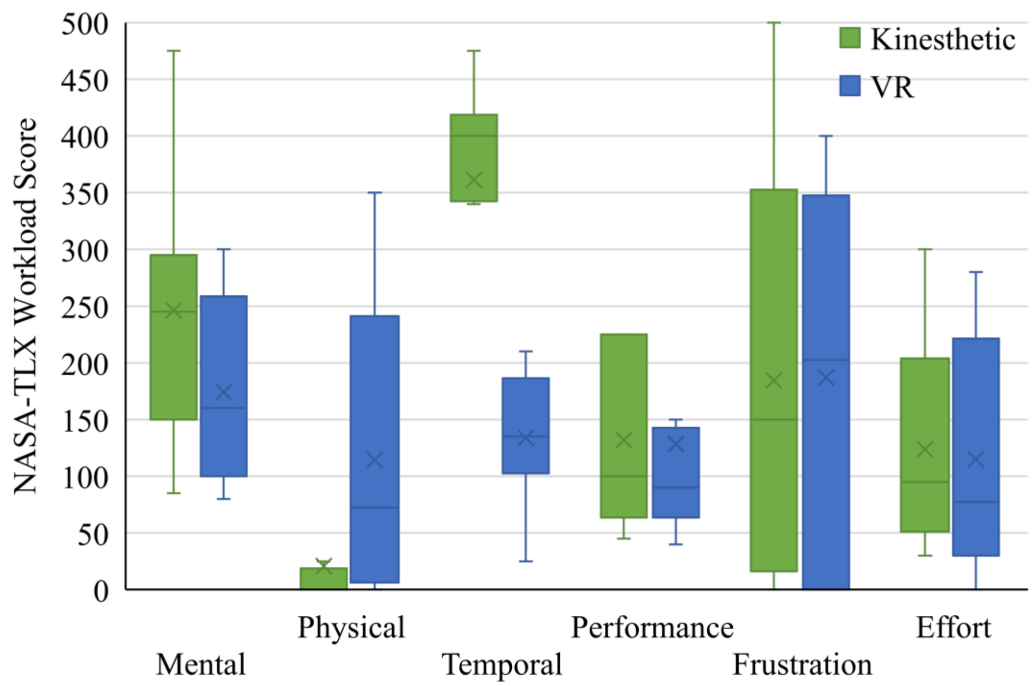


Figure 20. Actual intervention workload comparison

Table 2. Measurements for different intervention types

	Human to Human	VR Mock	Kinesthetic Mock	VR Actual	Kinesthetic Actual
Average time	83.4s	187.9s	238.1s	148.4s	235.8s
Treatment Integrity	84.0%	88.8%	84.0%	83.7%	52.9%
SUS scale score	-	63.75	52.19	53.75	44.38
NASA-TLX score	-	55.67	59.67	56.83	71.21

CONCLUSION AND FUTURE WORK

This thesis has three primary contributions:

1. The development of the VR-based teleoperation system for therapists to deliver robot mediated interventions to individuals with ASD using humanoid robots.
2. A comparison study between our developed VR-based teleoperation and a current state-of-the-art commercially available tool for teleoperating a humanoid robot for social tasks.
3. A tool for collecting data on the way therapists would run an intervention as if they were a robot.

Overall, the evaluation of VR-based teleoperation system demonstrated that participants had positive perceptions towards using the robot for intervention delivery and therapists can be trained to design and implement their own RMI. In general, participants were better at utilizing VR rather than kinesthetic to deliver interventions. Namely, their performance using VR was on par with in-person interventions in terms of treatment integrity, and participants could design and plan interventions faster using VR over the kinesthetic interface. Furthermore, all participants found VR more natural for delivering interventions and most participants found VR to be more adaptive to children's changing behavior than the kinesthetic interface.

One of the bigger challenges with using VR system, as it was identified in the questionnaire, was the lack of body awareness. This makes sense, since everything that is currently visible to the operator, is just HTC Vive controllers in the VR space camera feed. At the same time, camera that is installed on the robot, does not have large enough field of view to provide peripheral vision of robot's arms. As result, therapists sometimes may feel unsure if the robot does what they want it to do. To solve this issue, 3 recommendations can be made with possible solutions. First, the camera can be upgraded to the camera with larger field of view, that will provide peripheral vision. Second, the robot can be rendered in the VR space, such that operator will see a first-person view. Therefore, obtaining virtual peripheral vision. And lastly, showing a third person view of the robot into VR space. This will allow therapists to see the whole robot and be sure of what it does. In the future works, the last method most likely will be implemented since it will provide the most of information to the teleoperator.

For this study, the therapists were of age 22-33, which means that all of them were somewhat younger. Also, the ratio of female to male therapists in this study was 7 to 1. Even though it representative of the ABA therapists [46], there are possible biases. Since we introduce a new technology, it would be interesting to explore the performance of that technology among different age groups and genders in the future.

In the future, the goal is to have robot capable of delivering intervention to individuals with ASD autonomously. The robot would have to be able of generating natural non-verbal and verbal behaviors that would be going along with the intervention procedure. As well as working towards reaching a goal of the intervention itself, which usually is to teach a certain skill. At the same time, the robot should be able to adapt to changes in the setting and needs of the individual with ASD.

Our next steps towards that goal would be to utilize the teleoperation system to create a dataset of motion and sound produced by a teleoperated robot in response to sensory information such as what robot sees and hears. Such a dataset can then be analyzed to gain insights on the therapists' behaviors. As well it will be used for the design of a learning from demonstration algorithm to enable the robot to learn to autonomously perform the task from the therapists' demonstrations.

Currently we plan on dividing the problem, of making a robot-mediated intervention delivery autonomous, into two parts. The first part is learning the intervention structure and flow. This is needed so the dataset can be segmented based on the intervention stages instead of raw sensory data, therefore providing robot with additional knowledge on its current state. The second part in making robot autonomous is generation of verbal and non-verbal behaviors. Once again for this part, knowledge about what segment of intervention would also be beneficial. With segmented dataset several different generative algorithms could be explored to create behaviors for those segments of the intervention.

APPENDIX A

IRB APPROVAL

OAKLAND UNIVERSITY

Institutional Review Board

August 12, 2020

Protocol #: 1348929-1

Research Team:
Jessica Korneder

Wing-Yue Geoffrey Louie

The IRB has reviewed the Modification submission for the following study, "Socially Assistive Robots for Autism Care." The modified study with the changes listed below has been Approved through an expedited review per federal regulations.

The approved modifications are the following:
1) To improve the clarity of the study procedures, the following has been added/clarified:
a. The "research assistants" who also serves as research participants are defined as "therapist research assistants"
b.The potential participants (i.e. children with ASD, their parents, and therapist research assistants) would be in contact at the ABA clinic regardless of the research procedures has been added.
c. The social validity survey will be completed by the therapist research assistant who is controlling the robot during the intervention. The therapist research assistant will complete the intervention at the end of each intervention session while in the clinic. The purpose for these changes are to clarify the study.
d. A detailed description and figure for the setting of the study to describe where all researchers, participants, and robots will be located at the clinic. The primary purpose for this was to clarify the procedures and provide evidence that researchers will be able to socially distance during the study.

2) To minimize contact between researchers and participants as much as possible due to COVID-19, the consent process will be extended to also include online video conferencing, phone, or e-mail communications for reviewing study procedures, reviewing consent forms, and obtaining consent.

3) Extend the types of interventions the socially assistive robot will be delivering to the children with ASD. Namely, the socially assistive robot will be now also delivering other interventions that are typical of their treatment programs including communication, social skill, play skill, classroom readiness skill, and academic skill interventions. The purpose for increasing the range of interventions delivered by the robot to the children with ASD is to keep up with their learning rate and progress in their treatment programs. This will increase the time needed to complete the intervention from 1-2 hr. to 3-4 hr.

4) Addition of a 50-100 research compensation to the therapist research assistant controlling the socially assistive robot. The rationale for this change is to compensate the therapist research assistant for his/her time.

5) Addition of disinfection procedures for the study to minimize COVID related risks. At the ABA autism clinic, daily

cleaning and disinfecting procedures are already being conducted in compliance with CDC protocols. The same procedures will be followed to disinfect all robot, virtual reality, and laptop computer hardware. Namely, any “touch points” on any shared equipment will be wiped down between users. At a minimum, workstations and workspaces will be cleaned and disinfected at the beginning and end of each study session.

6) Addition of the following key personnel as research assistant: Megan Sochanski, undergraduate student, SECS, Alaaldin Hijaz, graduate student, SECS, and Roman Kulikovskiy, graduate student, SECS

7) Revise the Therapist Research Assistant Consent Form as follows:
a. Change the terminology for "research assistants" to "therapist research assistants" to improve the clarity of the document.
b. Remove the "wh-questions" because the researchers will extend the types of interventions the socially assistive robot will be delivering to the children with ASD. Namely, the socially assistive robot will be now also delivering other interventions that are typical of their treatment programs including communication, social skill, play skill, classroom readiness skill, and academic skill interventions. The purpose for increasing the range of interventions delivered by the robot to the children with ASD is to keep up with their learning rate and progress in their treatment program.
c. Add that the participants will be also controlling the robot with a laptop computer and the sessions could last 3-4 hours. This was to clarify details to the participants because they will be using the virtual reality headset and a laptop computer. Furthermore, the sessions maybe longer depending on their technology experience.
d. Add information indicating that participants will receive $50 at the end of the study to compensate them for their time.

8) Revise the script for recruiting therapist research assistants as follows:
a. Modify the terminology for "research assistants" to "therapist research assistants" to improve the clarity of the document.
b. Remove the wh-question answering skills because the robot may now teach communication, social skill, play skill, classroom readiness skill, and academic skill interventions.
c. Clarify that participants will be using both a laptop computer and virtual reality headset to control the robot.
d. Clarify that the study may take 3-4 hours depending on the technology experience of the participants.
e. Add information that participants will be receive $50 as a research incentive.

The IRB approved Consent Form and Recruitment document V 8/12/20 have been attached to this submission under Attachments in the Submission Details page. Please make sure to use the IRB approved version in consenting and recruiting participants.

The IRB date stamped consent document(s) has been published under Attachments in the Submission Details page. Please download the IRB date stamped consent and use it in consenting participants.

You are approved to implement the aforementioned modifications. Please retain a copy of this notification for your records.

This letter can also be found in Cayuse under the Letters tab in the Submission Details page.

If you have any questions, please contact the IRB office.

Thank you.

The Oakland University IRB

APPENDIX B

LIST OF CLASSES IMPLEMENTED BY RENDERING ENGINE LIBRARY AND THEIR BRIEF DESCRIPTION

Table 3. List of classes implemented by the rendering engine library and their brief description

Class name	Description
Application	Interface class that should be used to develop application in.
Canvas	Creates a plane with its own frame buffer which can be used to render to. In the case of this application the GUI was rendered to it. Also generates intersection even if the pointer transform matrix is passed on update.
IDrawable	Interface class for drawable objects.
Framebuffer	Object to which the image is being rendered.
GUI	Implements GUI rendering calls and event callbacks.
Input	Implements methods for obtaining input from mouse and keyboard.
Material	Abstraction over shader and texture
Mesh	Abstraction over vertex array. Implements a draw method that will draw the mesh with the bound shader and material.
Model	Abstraction for 3D model and loading them from a file.
PerspectiveCamera	Class containing a camera information that is used for rendering. Such as field of view, aspect ratio, near and far clip distance, projection matrix, view matrix, and positional vector of camera.
PerspectiveCamera Controller	Implements methods for changing the camera’s position and orientation.

Table 3—Continued

Class name	Description
ReferenceFrame	Provides a drawable reference frame that can be used to draw transformation matrices.
Renderer	Static class that provides basic rendering controls.
Shader	An abstraction over shader program. Implements loading it from a file, compiling, linking it, and binding when needed to be used.
Texture2D	Implements a texture class. Provides methods for loading texture from file, loading it to GPU and binding it when has to be used.
Timestep	Creates timestep object that can be used for timing calculations via overloaded operators.
VRCamera	Like PerspectiveCamera but with data for two perspectives of left and right eyes.
VRHmd	Abstraction over VR headset and holds hardware information that is needed for rendering, such as recommended renderer size.
VRSystem	Provides interface to the VR hardware, mainly an abstraction over OpenVR API.
Window	Allows to create and interface a desktop window.
WindowOverlay	An overlay that can be drawn over the framebuffer region. In this project it was used to draw GUI from the VR on the desktop screen.

APPENDIX C

LIST OF OPEN-ENDED QUESTIONS ADMINISTERED TO THE PARTICIPANTS

Questions that were asked after each use of interface:

1) Did you find it easy to customize an intervention?
2) Would you prefer another method for teaching the robot a new intervention (e.g., keyboard and mouse, physically moving the robot's arms, virtual reality)?
3) Do you think you could develop new interventions for the robot to deliver to the children?
4) Do you think you could develop new interventions for the robot to deliver to the children?

Question asked at the end of study:

5) Which method of controlling the robot would allow you to better adapt to a child during an intervention?
6) Which method of controlling the robot felt most natural?
7) What changes, if any, would need to be made for you to use the robot in the daily delivery of an intervention to children with ASD?
8) Do you have any additional comments or suggestions?

REFERENCES

[1] K. A. Shaw, M. J. Maenner, J. Baio, A. Washington, D. L. Christensen, L. D. Wiggins, S. Pettygrove, J. G. Andrews, T. White, C. R. Rosenberg, J. N. Constantino, R. T. Fitzgerald, W. Zahorodny, J. Shenouda, J. L. Daniels, A. Salinas, M. S. Durkin and P. M. Dietz, "Early Identification of Autism Spectrum Disorder Among Children Aged 4 Years - Early Autism and Developmental Disabilities Monitoring Network, Six Sites, United States, 2016," *Morbidity and mortality weekly report. Surveillance summaries,* vol. 69, p. 1–11, 2020.

[2] D. Granpeesheh, D. R. Dixon, J. Tarbox, A. M. Kaplan and A. E. Wilke, "The effects of age and treatment intensity on behavioral intervention outcomes for children with autism spectrum disorders," *Research in Autism Spectrum Disorders,* vol. 3, p. 1014–1022, 2009.

[3] W.-Y. G. Louie, J. Korneder, I. Abbas and C. Pawluk, "A study on an applied behavior analysis-based robot-mediated listening comprehension intervention for ASD," *Paladyn, Journal of Behavioral Robotics*, vol. 12, p. 31–46, 2021.

[4] B. Robins, K. Dautenhahn, R. Boekhorst and A. Billard, "Robotic assistants in therapy and education of children with autism: Can a small humanoid robot help encourage social interaction skills?," *Universal Access in the Information Society,* vol. 4, p. 105–120, 2005.

[5] J. Greczek, E. Kaszubski, A. Atrash and M. Matarić, "Graded cueing feedback in robot-mediated imitation practice for children with autism spectrum disorders," in *IEEE International Symposium on Robot and Human Interactive Communication*, 2014.

[6] E. Bekele, J. A. Crittendon, A. Swanson, N. Sarkar and Z. E. Warren, "Pilot clinical application of an adaptive robotic system for young children with autism," *Autism,* vol. 18, p. 598–608, 2014.

[7] C. A. Pop, R. E. Simut, S. Pintea, J. Saldien, A. S. Rusu, J. Vanderfaeillie, D. O. David, D. Lefeber and B. Vanderborght, "Social Robots vs. Computer Display: Does the Way Social Stories are Delivered Make a Difference for Their Effectiveness on ASD Children?," *Journal of Educational Computing Research,* vol. 49, p. 381–401, 2013.

[8] D. Feil-Seifer and M. Mataric', "Robot-Assisted Therapy for Children with Autism Spectrum Disorders," in *Proceedings of the International Conference on Interaction Design and Children*, 2008.

[9] K. Jordan, M. King, S. Hellersteth, A. Wirén and H. Mulligan, "Robotic Technology for Teaching Adolescents with Autism Spectrum Disorder: A Feasibility Study," in *Proceedings of the International Conference on Rehabilitation Engineering & Assistive Technology*, 2012.

[10] H. E. Andreae, P. M. Andreae, J. Low and D. Brown, "A Study of Auti: A Socially Assistive Robotic Toy," in *Proceedings of the Conference on Interaction Design and Children*, 2014.

[11] A. M. Alcorn, E. Ainger, V. Charisi, S. Mantinioti, S. Petrović, B. R. Schadenberg, T. Tavassoli and E. Pellicano, "Educators' Views on Using Humanoid Robots With Autistic Learners in Special Education Settings in England," *Frontiers in Robotics and AI,* vol. 6, p. 1–15, 2019.

[12] D. Conti, S. Di Nuovo, S. Buono and A. Di Nuovo, "Robots in Education and Care of Children with Developmental Disabilities: A Study on Acceptance by Experienced and Future Professionals," *International Journal of Social Robotics,* vol. 9, p. 51–62, 2017.

[13] M. Begum, R. W. Serna and H. A. Yanco, "Are Robots Ready to Deliver Autism Interventions? A Comprehensive Review," *International Journal of Social Robotics,* vol. 8, p. 157–181, 2016.

[14] J. J. Diehl, L. M. Schmitt, M. Villano and C. R. Crowell, "The clinical use of robots for individuals with Autism Spectrum Disorders: A critical review," *Research in Autism Spectrum Disorders,* vol. 6, p. 249–262, 2012.

[15] M. Hirokawa, A. Funahashi, Y. Itoh and K. Suzuki, "Adaptive Behavior Acquisition of a Robot Based on Affective Feedback and Improvised Teleoperation," *IEEE Transactions on Cognitive and Developmental Systems,* vol. 11, p. 405–413, 2019.

[16] K. Tsiakas, M. Kyrarini, V. Karkaletsis, F. Makedon and O. Korn, "A Taxonomy in Robot-Assisted Training: Current Trends, Needs and Challenges," *Technologies,* vol. 6, p. 1–19, 2018.

[17] B. A. English, A. Coates and A. Howard, "Recognition of gestural behaviors expressed by humanoid robotic platforms for teaching affect recognition to children with autism-a healthy subjects pilot study," in *International Conference on Social Robotics*, 2017.

[18] P. G. Esteban, P. Baxter, T. Belpaeme, E. Billing, H. Cai, H.-L. Cao, M. Coeckelbergh, C. Costescu, D. David, A. De Beir, Y. Fang, Z. Ju, J. Kennedy, H. Liu, A. Mazel, A. Pandey, K. Richardson, E. Senft, S. Thill, G. Van de Perre, B. Vanderborght, D. Vernon, H. Yu and T. Ziemke, "How to Build a Supervised Autonomous System for Robot-Enhanced Therapy for Children with Autism Spectrum Disorder," *Paladyn, Journal of Behavioral Robotics,* vol. 8, p. 18–38, 2017.

[19] T. Hughes-Roberts and D. Brown, "Implementing a Robot-Based Pedagogy in the Classroom: Initial Results from Stakeholder Interviews," in *International Conference on Interactive Technologies and Games,* 2015.

[20] L. Schreibman, G. Dawson, A. C. Stahmer, R. Landa, S. J. Rogers, G. G. McGee, C. Kasari, B. Ingersoll, A. P. Kaiser, Y. Bruinsma, E. McNerney, A. Wetherby and A. Halladay, "Naturalistic Developmental Behavioral Interventions: Empirically Validated Treatments for Autism Spectrum Disorder," *Journal of autism and developmental disorders,* vol. 45, pp. 2411-–2428, 2015.

[21] F. Eyssel, D. Kuchenbrandt, S. Bobinger, L. de Ruiter and F. Hegel, "'If You Sound like Me, You Must Be More Human': On the Interplay of Robot and User Features on Human-Robot Acceptance and Anthropomorphism," in *ACM/IEEE International Conference on Human-Robot Interaction*, 2012.

[22] R. Tamagawa, C. I. Watson, I. H. Kuo, B. A. MacDonald and E. Broadbent, "The Effects of Synthesized Voice Accents on User Perceptions of Robots," *International Journal of Social Robotics,* vol. 3, pp. 253-–262, 2011.

[23] A. Taheri, A. Meghdari, M. Alemi and H. Pouretemad, "Teaching music to children with autism: A social robotics challenge," *Scientia Iranica,* vol. 26, p. 40–58, 2019.

[24] K. Fischer, F. Kirstein, L. C. Jensen, N. Krüger, K. Kukliński, M. V. aus der Wieschen and T. R. Savarimuthu, "A comparison of types of robot control for programming by Demonstration," in *ACM/IEEE International Conference on Human-Robot Interaction*, 2016.

[25] S. Calinon and A. Billard, "Incremental Learning of Gestures by Imitation in a Humanoid Robot," in *ACM/IEEE International Conference on Human-Robot Interaction*, 2007.

[26] K. Vogeley and G. Bente, ""Artificial humans": Psychology and neuroscience perspectives on embodiment and nonverbal communication," *Neural Networks,* vol. 23, p. 1077–1090, 2010.

[27] C. A. G. J. Huijnen, M. A. S. Lexis, R. Jansens and L. P. de Witte, "Roles, Strengths and Challenges of Using Robots in Interventions for Children with Autism Spectrum Disorder (ASD)," *Journal of Autism and Developmental Disorders,* vol. 49, p. 11–21, 2019.

[28] L. McHugh, A. Bobarnac and P. Reed, "Brief report: teaching situation-based emotions to children with autistic spectrum disorder," *Journal of autism and developmental disorders,* vol. 41, p. 1423–1428, 2011.

[29] J. Burgoon, L. Guerrero and K. Floyd, Nonverbal Communication, Allyn & Bacon, 2010.

[30] J. Chen, M. Glover, C. Yang, C. Li, Z. Li and A. Cangelosi, "Development of an Immersive Interface for Robot Teleoperation," in *Towards Autonomous Robotic Systems*, Cham, 2017.

[31] D. Whitney, E. Rosen, D. Ullman, E. Phillips and S. Tellex, "ROS Reality: A Virtual Reality Framework Using Consumer-Grade Hardware for ROS-Enabled Robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018.

[32] A. Jackson, B. D. Northcutt and G. Sukthankar, "The Benefits of Immersive Demonstrations for Teaching Robots," in *ACM/IEEE International Conference on Human-Robot Interaction*, 2019.

[33] C. Stanton, A. Bogdanovych and E. Ratanasena, "Teleoperation of a humanoid robot using full-body motion capture, example movements, and machine learning," in *Proceedings of Australasian Conference on Robotics and Automation*, 2012.

[34] L. Fritsche, F. Unverzag, J. Peters and R. Calandra, "First-person tele-operation of a humanoid robot," in *IEEE Conference on Humanoid Robots*, 2015.

[35] M. Hirschmanner, C. Tsiourti, T. Patten and M. Vincze, "Virtual Reality Teleoperation of a Humanoid Robot Using Markerless Human Upper Body Pose Imitation," in *IEEE International Conference on Humanoid Robots*, 2019.

[36] J. Kennedy, P. Baxter and T. Belpaeme, "The Impact of Robot Tutor Nonverbal Social Behavior on Child Learning," *Frontiers in ICT,* vol. 4, p. 1–15, 2017.

[37] M. K. Lee and L. Takayama, "“Now, I Have a Body”: Uses and Social Norms for Mobile Remote Presence in the Workplace," in *ACM CHI Conference on Human Factors in Computing Systems*, 2011.

[38] J. Jung, T. Kanda and M. S. Kim, "Guidelines for Contextual Motion Design of a Humanoid Robot," *International Journal of Social Robotics,* vol. 5, p. 153–169, 2013.

[39] L. Wang, P.-L. P. Rau, V. Evers, B. K. Robinson and P. Hinds, "When in Rome: The Role of Culture & Context in Adherence to Robot Recommendations," in *ACM/IEEE International Conference on Human-Robot Interaction*, 2010.

[40] Valve Corporation, "OpenVR," 2020.

[41] Valve Corporation, "SteamVR," 2020.

[42] J. Häkkinen, M. Pölönen, J. Takatalo and G. Nyman, "Simulator sickness in virtual display gaming: A comparison of stereoscopic and non-stereoscopic situations," in *Proceedings of the Conference on Human-Computer Interaction with Mobile Devices and Services*, 2006.

[43] M. H. Draper, E. S. Viirre, T. A. Furness and V. J. Gawron, "Effects of image scale and system time delay on simulator sickness within head-coupled virtual environments," *Human Factors,* vol. 43, p. 129–146, 2001.

[44] E. D. Visser and C. Tossell, "Improving the Visual Comfort of Virtual Reality Telepresence for Robotics," in *International Conference on Social Robotics*, 2019.

[45] O. Cornut, "imgui," 2020.

[46] SoftBank Robotics, "NAOqi Developer Guide," 2020.

[47] M. R. Nosik and L. L. Grow, "Prominent Women in Behavior Analysis: An Introduction," *Behavior Analyst,* vol. 38, p. 225–227, 2015.

[48] S. Kuusikko, H. Haapsamo, E. Jansson-Verkasalo, T. Hurtig, M. L. Mattila, H. Ebeling, K. Jussila, S. Bölte and I. Moilanen, "Emotion recognition in children and adolescents with autism spectrum disorders," *Journal of Autism and Developmental Disorders,* vol. 39, p. 938–945, 2009.

[49] J. O. Cooper, T. E. Heron and W. L. Heward, Applied Behavior Analysis, Hoboken, NJ: Pearson Education, 2019.

[50] J. Brooke, "System usability scale (SUS): a quick-and-dirty method of system evaluation user information," in *Usability Evaluation in Industry*, P. W. Jordan, B. Thomas, B. A. Weerdmeester and I. L. McClelland, Eds., London, UK: Taylor and Francis, 1986, pp. 184-194.

[51] S. G. Hart and L. E. Staveland, "Development of NASA-TLX (Task Load Index): results of empirical and theoretical research," *Advances in Psychology,* vol. 52, pp. 139-183, 1988.

[52] R. A. Grier, "How High is High? A Meta-Analysis of NASA-TLX Global Workload Scores," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting,* vol. 59, p. 1727–1731, 2015.