Immersive Commodity Telepresence with the TRINA Robot Avatar

Joao Marcos Correia Marques[1*†], Patrick Naughton[1†], Jing-Chen Peng[1†], Yifan Zhu[1†], James Seungbum Nam[2], Qianxi Kong[2], Xuanpu Zhang[2], Aman Penmetcha[2], Ruifan Ji[3], Nairen Fu[1], Vignesh Ravibaskar[1], Ryan Yan[2], Neil Malhotra[4] and Kris Hauser[1]

[1]Department of Computer Science, Grainger School of Engineering, University of Illinois at Urbana-Champaign, 201 N Goodwin Ave, Urbana, 61801, Illinois, USA.
[2]Department of Mechanical Science and Engineering, Grainger School of Engineering, University of Illinois at Urbana-Champaign, 1206 W Green St, Urbana, 61801, Illinois, USA.
[3]Department of Electrical and Computer Engineering, Grainger School of Engineering, University of Illinois at Urbana-Champaign, 306 N Wright St, Urbana, 61801, Illinois, USA.
[4]VRotors, Inc., Pasadena, California, USA.

*Corresponding author(s). E-mail(s): jmc12@illinois.edu;
Contributing authors: pn10@illinois.edu; jcpeng2@illinois.edu; yifan16@illinois.edu; sn29@illinois.edu; qianxik2@illinois.edu; xuanpuz2@illinois.edu; amanp3@illinois.edu; ruifanj2@illinois.edu; nairenf2@illinois.edu; vignesh8@illinois.edu; ryanyan2@illinois.edu; neil@vrotors.com; kkhauser@illinois.edu;
†These authors contributed equally to this work and are listed in alphabetical order

Abstract

Immersive robotic avatars have the potential to aid and replace humans in a variety of applications such as telemedicine and search-and-rescue operations, reducing the need for travel and the risk to people working in dangerous environments. Many challenges, such as kinematic differences between people and robots, reduced perceptual feedback, and communication latency, currently limit how well robot avatars can achieve full immersion. This paper presents AVATRINA, a teleoperated robot designed to address some of these concerns and maximize the operator's capabilities while using a commodity light-weight human-machine interface. Team AVATRINA took 4th place at the recent $10 million ANA Avatar XPRIZE competition, which required contestants to design avatar systems that could be controlled by novice operators to complete various manipulation, navigation, and social interaction tasks. This paper details the components of AVATRINA and the design process that contributed to our success at the competition. We highlight a novel study on one of these components, namely the effects of baseline-interpupillary distance matching and head mobility for immersive stereo vision and hand-eye coordination.

Keywords: Telepresence, Haptics, Teleoperation, Robotics

1 Introduction

Telepresence [12], or telexistence [64], aims to enable a human operator to feel as though they are actually present in a remote robot's environment through immersive vision and rich haptic feedback. Teleoperation of mobile manipulators has the potential to aid a wide variety of applications including telemedicine [40, 50, 74], search-and-rescue operations [23, 24], and remote environment exploration [9, 32, 41], by leveraging the perception and planning capabilities of humans. However, telepresence can be hindered by multiple factors such as kinematic differences between the robot and operator [3], reduced perceptual feedback, and network latency [44]. In addition, the cost of the robot and teleoperation hardware limits the accessibility of telepresence robots. Exoskeletons, which are commonly used operator hardware for teleoperation [73] in research labs and industrial settings [68, 71], are not only costly but also require careful setup and calibration to use.

This paper presents AVATRINA (AVAtar Tele-Robotic Intelligent Nursing Assistant), shown in Fig. 1, an immersive avatar robot operated with low-cost commodity hardware. We describe the design of the system, including its optimized kinematics, sensing and perception systems, and user interfaces, which creates an immersive telepresence experience. The main contributions of this work are:

- An immersive, novice-friendly teleoperation system with human-like manipulation, communication, and sensing capabilities, controlled by lightweight commodity operator hardware.
- Human subjects studies to validate the design of our perception system. We examine the effect of: (1) matching the distance between the robot's eye cameras to the operator's interpupillary distance and (2) providing more degrees of freedom to move the robot's head, and show weak evidence that these methods improve the operator's depth perception.
- A set of task-oriented metrics for optimizing robot hardware design for immersive teleoperation, complementary to traditional workspace analysis.

Team AVATRINA, a collaboration between the University of Illinois and vRotors[1], built AVATRINA to compete in the ANA Avatar XPRIZE competition[2] finals. This competition aimed to accelerate the development of robot avatar technologies, improving the quality and variety of haptic sensing and rendering devices, and promoting fundamental research in system integration, networking, and virtual reality to create responsive, immersive, and intuitive telepresence systems. Our team achieved 4th place and was among the 4 teams that completed all 10 tasks at the competition[3]. In this paper, we also share the insights and experience gained from participating in the ANA Avatar XPRIZE competition and on how to design an immersive Avatar system.

This paper is organized as follows. In Sec. 2, we review related literature. In Sec. 3, we describe the system design goals and the overall system components. In Secs. 4–9, we describe the major components of the proposed avatar system, including the manipulators, locomotion, vision, hands, and software architecture. Finally, we present our participation and evaluation in the ANA Avatar XPRIZE in Sec. 10, and offer discussions and our lessons learned in Sec. 11.

2 Related Work

Avatar embodiments can take many forms depending on their intended applications. Many of the avatars presented during the DARPA robotics challenge [27] opted for humanoid bipedal robots with an alternative wheeled locomotion mode for faster traversal of flat ground and more stable environmental interaction. Two examples are team IHMC's entry [20] and KAIST's DRC-HUBO+ [33]. Other notable designs include RoboSimian's quadruped design [21] and Team NimbRo's wheeled centaur-like platform Momaro [57].

These robots were designed to operate in challenging locomotion scenarios and were primarily focused on efficient task completion under adversarial networking and environmental conditions. As such, their UI was focused on redundancy and robustness, with some interfaces requiring

[1] https://www.vrotors.com/
[2] https://www.xprize.org/prizes/avatar
[3] https://youtu.be/lOnV1Go6Op0?t=28364

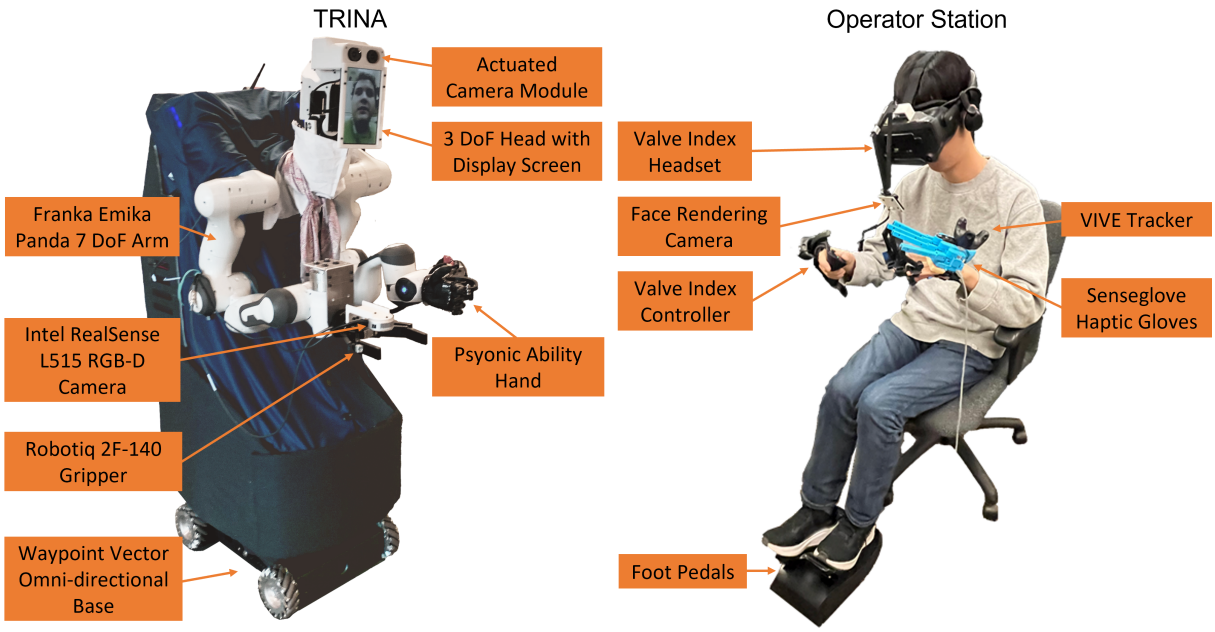


Fig. 1: An overview of the AVATRINA robot and operator station. The robot is configured at the "home" configuration. [Best viewed in color.]

up to seven operators to man a single robot [57]. Furthermore, these robots fulfilled no social function and thus lacked expressive features for communication with remote individuals.

Avatars are not restricted to enabling humans to perform remote or dangerous activities. They can also be used to help people with disabilities expand their independence in performing activities of daily living and working [43, 67]. Being designed for indoor use, however, imposes other important constraints on these robots, such as maximum size, weight and cost, and introduces different manipulation requirements [22].

Yet other platforms focus on operator immersion and on the avatar's socially expressive capabilities. The TELESAR VI platform [65], for instance, follows a legged humanoid design with human proportions, but forgoes practical arm payloads and robot mobility in favor of enabling expressive arm, torso, finger and hand motion. Its interface also prioritizes immersion, using a VR Head Mounted Display (HMD) and haptic gloves which render finger forces, vibrations, and temperatures sensed from embedded sensors in the anthropomorphic avatar hand.

The ANA Avatar XPRIZE finals was a culmination of efforts to unify these research directions into socially capable, task efficient, and immersive robot avatars. Since the competition did not require challenging environment traversal, most teams opted for wheeled robot bases [35, 38, 58]. Indeed, as noted by Luo et al. [35], avatar embodiments that relied solely on legged locomotion all irrecoverably fell during the competition. Furthermore, as social tasks were an integral part of this competition, most embodiments paid special attention to properly rendering the operator's face and voice.

Out of the 4 teams that completed all 10 tasks in the competition, 3 used VR headsets for immersive operation (NimbRo [58], Pollen Robotics and AVATRINA [38]), while Team Northeastern's [35] system used an ultra-widescreen monitor for visualization. The teams that used VR employed different strategies to reconstruct the operator's face without the headset on the robot, while Team Northeastern did not have to address this problem.

The user interfaces of the top 3 teams all used some form of exoskeleton for force haptic transmission to the arms and fingers. NimbRo's consisted of a rig with additional robot manipulators mounted to the operator's arms to render force, Team Northeastern leveraged mirrored hydraulic

mechanisms between operator and avatar for rendering arm and finger forces, and Pollen Robotics used a 1 DoF elbow mounted exoskeleton to provide arm force feedback. AVATRINA was the only team to exclusively use vibratory and visual cues for rendering arm forces to the operator and complete the full course.

Furthermore, these systems used different methods for remote texture sensing - Pollen Robotics and NimbRo used acoustic based sensors [48], while Team Northeastern leveraged both contact acoustics and sensed vibrations and forces on the hydraulic actuators for identifying surfaces [35]. Our system captured a high-resolution heightmap of surfaces and used it to render distinct vibratory and auditory cues for each surface without requiring direct contact.

3 System Overview

This section describes overall design goals for the system and an overview of the major components. Individual components will be described in following sections.

3.1 System Design Goals

AVATRINA is designed to be a socially-capable robot that can be controlled by a human *operator* to navigate and manipulate objects in remote environments designed for humans. The social capabilities should enable the operator to interact naturally with other human *recipients* in the remote environment. It should be easy to operate for novice users, accessible to a wide number of operators, and ergonomic enough for long-term usage. We emphasize that ergonomic and lightweight user interfaces are necessary to enable longer-term applications of teleoperation, such as tele-work and data gathering for imitation learning. Lightweight user interfaces improve scalability to many users and the likelihood of adoption. Moreover, AVATRINA should be a stable, maintainable platform to enable reproducible research in tele-nursing, telerobotics, and mobile manipulation for many years.

Our system design goals include:

1. The robot should have similar manipulation capabilities to humans, with human-like kinematics and load capabilities.
2. The operator station should be comfortable, ergonomic, and lightweight, requiring minimal setup.
3. The operator interface should be intuitive and immersive.
4. Recipients in the remote environment should feel the operator's presence.
5. The robot should use as many off-the-shelf parts as possible for ease of construction, maintenance, and reproduction.
6. The robot should have easily reconfigurable end-effectors and sensors to support different applications.

3.2 Overall System Components

The overall system consists of the AVATRINA robot and the *operator station*, summarized in Fig. 1.

The robot is a bimanual mobile manipulator consisting of two robot arms (Franka Emika Panda), an omnidirectional mobile base (Waypoint Vector), and a custom 3 DoF anthropomorphic head with a custom adjustable baseline stereo camera. Other system specifications are listed in Table 1. We support different end-effectors, including the 6 DoF anthropomorphic Psyonic Ability hand, the Robotiq 2F-140 and 2F-85 parallel jaw grippers, the Franka hand, and the Righthand Reflex 3-fingered hand. The default configuration has the Psyonic hand on the left arm and a 2F-140 Robotiq gripper on the right arm. The robot is similar in size to an adult, standing at 1.85 m tall and 0.7 m wide, and has a forward reach of 0.5 m. Four ultrasonic rangefinders around the perimeter of the robot provide enhanced situational awareness. In the default configuration, there is an RGB-D sensor mounted at the wrist of the right arm for close-range inspection and texture sensing. AVATRINA can operate for about 2 hours on its onboard battery. The robot is connected to the Internet via WiFi and can connect to an operator station via a handshake configured through designated servers.

The operator station is designed to consist of commodity hardware, so that the robot can potentially be connected to any operator with Internet access. It consists of an desktop PC, a VR head-mounted display (HMD) and controllers, foot pedals, and the SenseGlove haptic gloves (whose specifications are noted in Table 1). We

support various commercial VR products such as the Valve Index, HTC Vive Pro, and Meta Quest 2. The poses of the HMD, controllers, and gloves are tracked by Vive Trackers system. In the default setup (Fig. 1), the operator wears the Valve Index headset, a SenseGlove on their left hand, and holds a Valve Index controller with their right hand. All of these components could be readily purchased at the time of the competition for under USD 8000.00, with the following cost breakdown:

- Operator PC ∼ $3000.00
- SenseGlove - $3000.00
- Valve Index and Base Stations - $1000.00
- Vive Trackers ∼ $200.00
- Pedals ∼ $150

Fig. 2 shows how these devices are mapped to AVATRINA's different components to give the operator control over AVATRINA's motion.

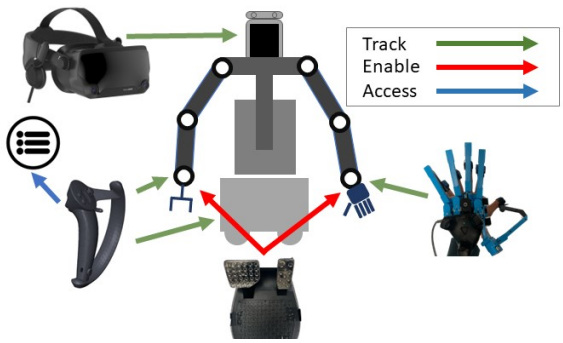


Fig. 2: Mapping between operator control devices and AVATRINA's movement. AVATRINA's head tracks the VR headset and provides stereo camera feedback. The right VR controller controls the right gripper, and the right joystick commands the base translation and rotation. The left anthropomorphic Psyonic gripper tracks the SenseGlove. The foot pedals enable the movements of the end-effectors. The right controller can also bring up a menu of different semi-autonomous functionalities. [Best viewed in color.]

4 Manipulators and Locomotion

AVATRINA both interacts with and is controlled by people, so the design criteria for its manipulators differ substantially from typical criteria for industrial robots. The operator is primarily concerned with the visibility of AVATRINA's hands and the intuitiveness of the arm control, while the recipient desires a compliant and predictable robot with which they can safely interact. This section describes how we designed the robot's hardware and control software to meet these criteria.

Table 1: AVATRINA Canonical Specifications.

Overall	
Length x Width x Height	1.0 m x 0.68 m x 1.75 m
Weight	157 kg
Manipulation and Locomotion	
Payload	3 kg (including gripper)
Forward Reach	0.5 m
Max Speed	0.2 m/s
Sensors	
RGB-D Sensors	Configurable, default: 1
Stereo Camera	1
Ultrasonic Rangefinders	3
Miscellaneous	
Estimated Battery Life	2 hrs
Battery Capacity	1534 Wh
Onboard Compute	
Main Computer	
CPU	AMD Ryzen 9 5900X
GPU	NVIDIA RTX 3060 12 GB
RAM	64 GB DDR4 3200 MT/s
OS	Ubuntu 20.04
Control Computer	
CPU	AMD Ryzen 7 5800U
RAM	16 GB DDR4 3200 MT/s
OS	Ubuntu 20.04 (PREEMPT_RT)
Operator Station Compute	
CPU	AMD Ryzen 9 5900X
GPU	NVIDIA RTX 3090
RAM	32 GB DDR4 3200 MT/s
OS	Windows 11
Operator Station Hardware	
Head-Mounted Display	Valve Index Headset
Right Hand Tracking	Valve Index Controller
Left Hand Tracking	Vive Tracker and SenseGlove
Arm Activation	Thrustmaster F430 Pedals
Facial Rendering Camera	ELP-USBFHD01M-L170

4.1 Kinematic Design Optimization

In designing AVATRINA's manipulation capabilities, we considered several criteria related to the mounting of the arms to the torso, including:

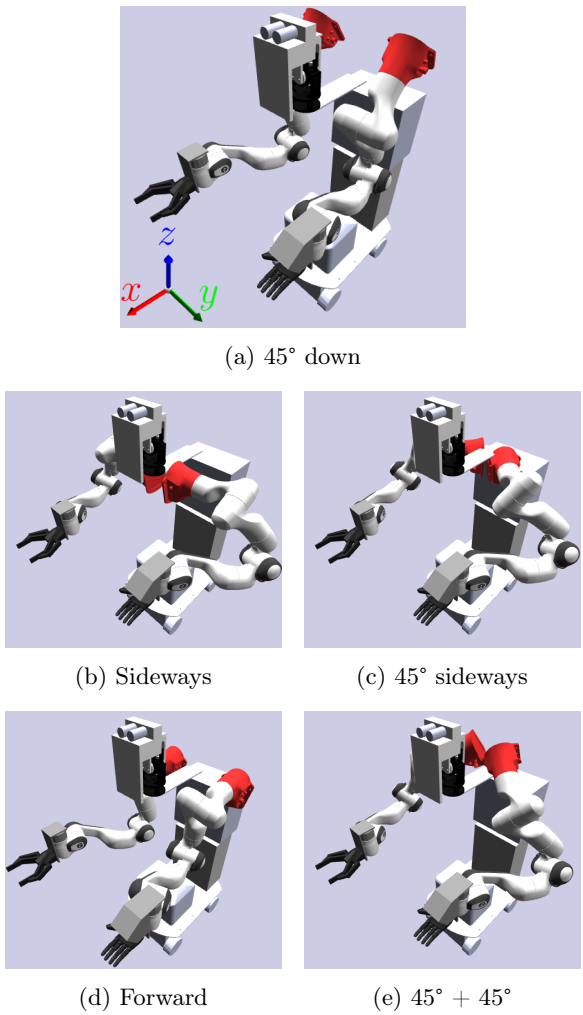
- The arms should have a large dexterous workspace, and be able to track the operator smoothly through that workspace.

- The arms should have human-like kinematics to make their behavior predictable for the operator and recipients.
- During manipulation tasks, the robot should minimize occlusion of the object being manipulated.
- The robot's width should be less than 800 mm, to enable passing through the Americans with Disabilities Act (ADA) compliant doors[4].

These criteria are highly coupled and often conflict with one another, making it difficult even for experienced engineers to reason about them through intuition alone. We therefore employed a *co-design* approach, using computational methods to optimize free design parameters (arm and hand mounting) while simultaneously considering the control software in the evaluation loop [10, 19].

The design parameters θ include the arm mounting $SE(3)$ transform and the 2 gripper mounting $SE(3)$ transforms. We assumed the arms will be mounted symmetrically, so we only optimize one shoulder mount and reflect it to obtain the other. To further simplify the problem, we first optimize the arm mounting transforms, then fine-tune the gripper transforms.

We propose four metrics to capture our design goals: 1) the robot's ability to track the operator, 2) visibility of the grippers, 3) shoulder width, and 4) human-likeness. We manually proposed a set of promising design parameters θ, shown in Fig. 3 guided primarily by manufacturability constraints, and evaluated these metrics for each candidate design.



(a) 45° down

(b) Sideways

(c) 45° sideways

(d) Forward

(e) 45° + 45°

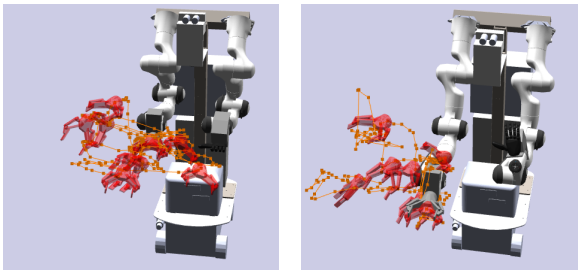
Fig. 3: Candidate arm mounting poses. Shoulder attachment points are highlighted in red. [Best viewed in color.]

4.1.1 Design Metrics

To evaluate our designs in context, we first record a person teleoperating a floating gripper in simulation to cover the robot workspace for representative manipulation tasks, such as tabletop manipulation (Fig. 4). Ideally the robot should be able to match these trajectories with its end-effector (EE). Each trajectory is given by $\mathbf{T} = (T_1, T_2, \ldots, T_N)$, a list of N $SE(3)$ transforms expressed in the robot base frame.

For a candidate set of mounting parameters θ we can then simulate tracking the trajectory using the robot's arm controller (subsection 4.2). Each subsequent robot configuration q_t is derived by repeatedly applying the controller on the simulated robot state at time step $t-1$ to bring the EE towards T_t. The controller biases each joint towards the center of its limits to aid in redundancy resolution. The resulting joint trajectory $\hat{\mathbf{Q}}(\theta) = (q_1, \ldots, q_N)$, is thus a function of θ. For ease of notation we also denote the simulated end-effector trajectory as $\hat{\mathbf{T}}(\theta) = (\hat{T}_1, \ldots, \hat{T}_N)$.

[4] https://www.access-board.gov/ada/



(a) Trajectory A (b) Trajectory B

Fig. 4: Trajectories used in kinematic optimization and inverse kinematics (IK) evaluation. Trajectory A is mostly within the robot's expected workspace; Trajectory B extends further to the right, testing the robot's ability to reach farther poses and recover from commands that exceed workspace limits. [Best viewed in color.]

The operator tracking error metric M_{track} is defined as:

$$e_t([R|p],[\hat{R}|\hat{p}]) = \mathbf{1}\Big(\|p-\hat{p}\|_2 > \epsilon_x \quad \text{or} \quad \sphericalangle(R^{-1}\hat{R}) > \epsilon_\theta\Big), \tag{1}$$

$$M_{track}(\theta) = \frac{1}{N}\sum_{t=1}^{N} e_t(T_t, \hat{T}_t(\theta)).$$

Here $\sphericalangle(\cdot)$ indicates the angular deviation of a rotation matrix from the identity, and $\mathbf{1}(\cdot)$ is the indicator function converting `true` to 1 and `false` to 0. The metric counts the number of time steps that tracking error exceeds thresholds in either position or orientation. We use this rather than the sum of pose errors, since small errors will not be noticed but tracking loss tends to confuse and frustrate the operator. We set the tolerances to $\epsilon_x = 0.05$ m and $\epsilon_\theta = 0.3$ rad.

The visibility metric is the fraction of the grippers not occluded by the arms as seen from the robot's camera pose as the robot moves along $\hat{\mathbf{Q}}$. The camera pose is defined as the midpoint between the robot's two eye cameras while the head points forward. For each simulated pose, we render the camera image to find the number of gripper pixels visible, $\hat{v}_g(\hat{q})$, and the number of total gripper pixels that would be visible if the arms were not present, $v_g(\hat{T})$. This metric is defined as:

$$M_{vis}(\theta) = \frac{1}{N}\sum_{t=1}^{N} \frac{\hat{v}_g(\hat{q}_t)}{v_g(\hat{T}_t)} \tag{2}$$

The shoulder width metric M_{width} is defined by the maximum horizontal extents of the arms in the most compact home configuration. The human-likeness metric M_{human} is a subjective measure. To determine this, we surveyed lab members by asking the following: "Cast your rating of human-likeness of the different shoulder mountings on a scale of 1 - 5, with 5 being very human-like." Recordings of the entire robot with different shoulder mountings undergoing simulated motions were presented. M_{human} is then defined as the average of the ratings from all responses.

4.1.2 Results

First, we compared five different arm mounting configurations and evaluated them based on M_{width}, M_{human}, M_{track}, and M_{vis}. The Low gripper mount (Fig. 5) was chosen arbitrarily for this evaluation. Results are shown in Table 2. We chose 45° down by prioritizing width, human likeness, and tracking error in the workspace in front of the robot (Trajectory A), while maintaining acceptable error in out-of-range reaching tasks (Trajectory B) and comparable visibility with other mountings that maintain good tracking. These trajectories are sampled at 125Hz, with a maximum commanded velocity of the end effector at $\tilde{8}$ m/s.

We then compared different gripper mounting configurations (shown in Fig. 5) using M_t and M_v. For the right hand, we observed little difference between the low and high mounted grippers, as seen in Table 3, while the end mount exhibits especially poor tracking performance. This is because when oriented forward for tabletop manipulation tasks, the gripper places two of the wrist joints in near-singularity, causing many tracking failures. Ultimately, we chose the low mount to reduce wrist interference with obstacles when manipulating objects on a table. For both hands, the tracking metrics indicate that we should choose the higher mount. However, we noticed that this mount would result in collisions with the environment during tabletop manipulation tasks, a failure mode not captured by our metrics. Therefore, we

Table 2: Arm mounting evaluation. Lower width (M_{width}) and tracking error (M_{track}) is better; higher human likeness (M_{human}) and visibility (M_{vis}) is better.

			Trajectory A		Trajectory B	
Mount	$M_{width} \downarrow$	$M_{human} \uparrow$	$M_{track} \downarrow$	$M_{vis} \uparrow$	$M_{track} \downarrow$	$M_{vis} \uparrow$
45° down	**68** cm	**4.00**	0.13%	39.19%	42.76%	31.39%
Forwards	**68** cm	2.25	14.88%	37.75%	55.45%	31.71%
Sideways	91 cm	3.25	29.60%	**49.17%**	43.33%	**38.86%**
45° sideways	79 cm	3.25	10.92%	41.26%	**29.82%**	35.02%
45° + 45°	79 cm	2.25	**0.07%**	36.30%	38.17%	32.71%

chose the low mount despite it having slightly worse metrics. For future work, this indicates that metrics should incorporate more context of the robot's tasks, such as expected environmental constraints.

(a) Low mount. (b) High mount. (c) End mount.

Fig. 5: Candidate gripper mounts.

Table 3: Gripper mounting evaluation results.

		Trajectory A		Trajectory B	
Side	**Mount**	$M_{track} \downarrow$	$M_{vis} \uparrow$	$M_{track} \downarrow$	$M_{vis} \uparrow$
Right	low	1.02%	**0.566**	28.41%	**0.570**
	high	**0.22%**	0.520	**24.59%**	0.549
	end	76.64%	0.343	82.59%	0.422
Left	low	0.14%	**0.448**	27.44%	**0.711**
	high	**0.08%**	0.365	**7.77%**	0.617
	end	44.86%	0.214	36.54%	0.322

4.2 Arm Control

AVATRINA's arms are to track target end effector poses. Our arm control algorithm consists of two levels. On the higher level, an inverse kinematics (IK) solver computes desired joint positions q_{desired} to track the target end effector pose T_{target}. The controller interpolates between the robot's current configuration and q_{desired}, checking for self collisions. Environment collisions are not handled by the controller and left to the operator. q_{desired} is then sent to the lower level controller if there is no collision. All of these steps are performed at 125Hz for each arm. On the lower level, we compute joint torques τ_j to track q_{desired} at 1 kHz while also allowing for compliant motion to ensure safety during contacts.

4.2.1 Inverse Kinematics for Teleoperation

Real-time tracking with IK is a challenging problem due to discontinuities, singularities, and local minima, which can cause abrupt movement or loss of tracking. To deal with this problem, we explored two different approaches to IK solving: In one method, IK target poses are modified to ensure dexterity according to some measure of manipulability [37] before being sent to a generic IK solver. The other method adopts a quadratic program IK solver that explicitly enforces velocity bounds without modifying target poses [28].

In the first approach, we apply the measure of manipulability (MoM) method proposed by Marani et al. [37] that modifies the target transform given by the user so that the measure of manipulability of each arm remains above a threshold. The measure of manipulability (M_m) is given by

$$M_m(q) = \sqrt{\det \boldsymbol{J}\boldsymbol{J}^T}. \tag{3}$$

where $\mathbf{J} \equiv \mathbf{J}(q)$ refers to the Jacobian of the arm relating its 7 joint velocities to the end effector angular and translational velocity. M_m is positive semi-definite, reaching zero only when the robot arm is in singularity.

MoM prevents M_m from decreasing below a small threshold by 1) computing a correction motion that zeros out any components of the commanded motion (in task space) that decrease M_m if the arm is at or below a given threshold, and at the same time, 2) introducing a small motion that pushes the robot in the direction of increasing M_m. We refer to the original paper [37] for more details. The modified IK target is tracked by a standard Newton's-method-based IK solver [26]. If the IK solution is infeasible, for example due to self collisions or joint limits, no attempt is made to track the target and the controller does not move the robot. This ensures that the robot does not enter infeasible configurations, assuming it starts in a feasible configuration.

The second approach uses a quadratic program (QP) solver based on the resolved-rate controller described in [28]. Let δr_{target} denote the 6D error vector between the current end effector transform $T_{ee}(q_{\text{desired}})$ and T_{target}. At every time step, a step in joint space δq is found as the solution to a constrained optimization problem where the objective function consists of tracking accuracy, joint velocity penalty, and a joint angle bias for redundancy resolution:

$$\begin{aligned} \arg\min_{\delta q} \quad & \|\delta r_{\text{target}} - \boldsymbol{J}\delta q\|^2_{W_r} \\ & + \|\delta q\|^2_{W_q} \\ & + \|q + \delta q - q_{\text{bias}}\|^2_{W_b} \end{aligned} \tag{4}$$

$$\text{subject to} \quad q_{\min} \leq q + \delta q \leq q_{\max},$$

where $\|r\|^2_W = r^T W r$ is the weighted squared norm of a vector, and the constraints encode the arm's joint limits. This optimization problem is solved with a generic convex optimization problem solver [1] in real-time. In our implementation, we empirically chose $W_r = I, W_q = 0.02I$, and $W_b = 0.0005I$.

After solving for the target step δq, the new target joint configuration of the arm is checked for self collisions and collisions with other components of the robot. If the IK solution is infeasible, for example due to self collisions or joint limits, no attempt is made to track the target and the controller does not move the robot.

For both approaches, we considered the following bias configurations:

- Neutral bias: Bias solvers towards the middle of the arm's joint limits.
- Shoulder angle bias: We developed a heuristic for the shoulder angle q_{shoulder} of the robot based on the target end effector transform, which guides how far outwards the elbow should swing. The IK solver biases the robot's shoulder angle towards this value. For more details, see Appendix A.

We evaluated the two different approaches for solving IK under different bias configurations by comparing the tracking metric M_t. The optimized robot design discussed in subsubsection 4.1.2 and the same two trajectories (shown in Fig. 4) were used for all the experiments in this section. The results are reported in Table 4. QP generally provides lower tracking error, while MoM achieves faster computation. Table 4 also shows that the shoulder heuristic joint biasing generally improves tracking performance without sacrificing computation speed. One reason for this is that seeking more "natural-looking" elbow poses tends to help the robot avoid inverting the shoulder and elbow joints, as seen in Fig. 6, preventing the arm from getting stuck in regions of low dexterity. This is especially true for QP, which can allow the arm to enter singular configurations to achieve better tracking in the short term, at the expense of losing tracking at later points in the trajectory. We used biased MoM during the XPRIZE finals competition to ensure fast tracking, whereas the biased QP method is used in this paper's design evaluation (subsubsection 4.1.2) and human subjects experiments (subsection 5.4).

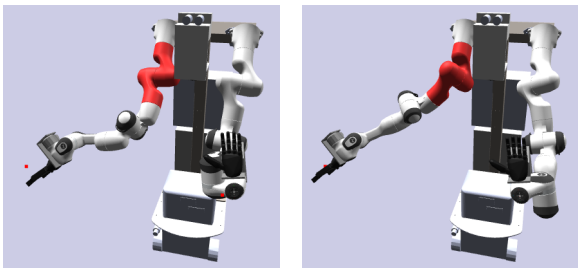
4.2.2 Low level torque control

To track desired joint positions (q_{desired}) coming from the inverse kinematics controller, the low-level controller computes joint torques τ_{target}, which are sent to the robot's low level controller at 1 kHz. The raw commands coming from the high level controller are passed through a complementary filter, and target joint velocities are computed as the velocity of the filtered target position using a finite difference method. τ_{target} is computed by

$$\begin{aligned} \tau_{pd} &= k_p \Delta q + k_d \Delta \dot{q} + \text{FF}(q, \dot{q}), \\ \bar{\tau}_{\text{target}} &= \text{plimit}(\tau_{pd}, \dot{q}) + c_q(q) + c_{\dot{q}}(\dot{q}), \\ \tau_{\text{target}} &= \text{clamp}(\bar{\tau}_{\text{target}}, \tau_{\max}, \tau_{\min}), \end{aligned} \tag{5}$$

Table 4: Tracking error rate for IK solver and bias heuristics, on two trajectories. Time is average time per loop iteration.

		Trajectory A		Trajectory B	
Solving Method	**Bias**	$M_{track} \downarrow$	Time (ms) $\downarrow$	$M_{track} \downarrow$	Time (ms) $\downarrow$
Measure of Manipulability	Neutral	11.17%	0.41	32.60%	1.00
Quadratic Programming	Neutral	**0.13%**	1.32	44.47%	2.72
Measure of Manipulability	Shoulder angle	2.12%	**0.40**	33.60%	**0.91**
Quadratic Programming	Shoulder angle	1.22%	1.37	**28.42%**	2.97



(a) IK with neutral bias "picks" the wrong direction for the shoulder.

(b) IK with heuristic bias keeps the shoulder angle on the correct side.

Fig. 6: The specific failure case in trajectory B accounting for the poor performance of IK solving without heuristic biasing. See Table 4 for full results. [Best viewed in color.]

where it uses a PD controller with the feedforward terms while respecting joint position, joint velocity, power, and torque limits. Here $\mathrm{FF}(q, \dot{q})$ is the feedforward term, computed based on the dynamics model of the robot (provided by the manufacturer) to compensate for gravity and Coriolis terms. To account for joint position and velocity limits, we use a quadratic control barrier function, which imposes a smooth increase in resistance as the joint is pushed towards its position (resp. velocity) limits:

$$c_x(x) = \begin{cases} -K\left(\frac{x_{\max}-x}{\epsilon_x}\right)^2 & \text{if } x > x_{\max} - \epsilon_x \\ K\left(\frac{x_{\min}-x}{\epsilon_x}\right)^2 & \text{if } x < x_{\min} + \epsilon_x \\ 0 & \text{otherwise,} \end{cases} \quad (6)$$

where the parameters $K, \epsilon_x, x_{\max}$, and $x_{\min}$ can be tuned based on the quantity being limited.

The Franka Emika Panda arm also has built-in joint power and torque limits for safety reasons. Violating these limits causes the arm to engage a protective stop, causing the operator to lose control of the arm temporarily. To prevent this, we add power and torque limits to our controller. The power limit is implemented as

$$\mathrm{plimit}(\tau, \dot{q}) = \begin{cases} p_{\max}/\dot{q} & \text{if } \tau\dot{q} > p_{\max} \\ \tau & \text{otherwise,} \end{cases} \quad (7)$$

and the torque limit clamps between minimum and maximum torques. Here, $\dot{q}_{\text{target}}$ is estimated by taking a finite difference derivative of q_{target}.

4.3 Arm Operator Interface

We use a relative, clutching-based, identity-scaled system for controlling AVATRINA's arms, which was chosen for a few reasons. First, relative (compared to absolute) tracking allows the operator to reach more of AVATRINA's workspace while keeping their arms in a more comfortable pose. Second, using a clutch to activate the operator's control of the arms helps prevent unintentional motion, and allows the operator to rest their arms more naturally when not moving AVATRINA to reduce fatigue [15, 34]. Finally, using a one-to-one scaling between the operator's and AVATRINA's motion helps to preserve the operator's proprioception, and allows them to better predict how AVATRINA will move based on their input commands.

To take control of AVATRINA's arms, the operator depresses a corresponding (left or right) foot pedal and moves a controller whose transform is tracked via four Valve Index Base Stations. While a pedal is depressed, the motion of the corresponding controller is mapped to the motion of the EE target of the corresponding arm. The inital

pose of the EE target is reset to AVATRINA's current EE pose whenever the pedal is depressed. The arm controller described in subsection 4.2 then attempts to track this target.

We found that operators were best able to quickly understand how their motions translated to AVATRINA when their controllers were tracked in a frame that follows their body, rather than their head. Because we do not track the operator's body orientation, we need to approximate it. We found that operators tend to face forward when they begin clutching, so each time the operator begins clutching, we compute a "forward" frame by taking just the z component of the head frame's rotation,

$$R^{\text{world}}_{\text{forward}} = \begin{bmatrix} 0 \\ 0 \\ \phi_z \end{bmatrix}. \tag{8}$$

This frame is kept constant while the clutch is pressed so that arm-body motion is consistent, even if the operator moves their head. Throughout this section, a subscript denotes the object while a superscript denotes the reference frame.

AVATRINA's arm motion in its base frame follows the operator's motion in the forward frame. We compute the change in controller positions and rotation relative to the forward frame:

$$\Delta p_c^{\text{forward}} = R^{\text{forward}}_{\text{world}} \Delta p_c^{\text{world}} \tag{9}$$

$$\Delta R_c^{\text{forward}} = R^{\text{forward}}_{\text{world}} \Delta R_c^{\text{world}} \tag{10}$$

where c denotes the operator's controller and Δ denotes the increment from the last time step. We then move AVATRINA's EE target, denoted t, by the same increment in the base frame to produce a new target pose $\tilde{T}_t^{\text{base}}$:

$$\begin{aligned} \tilde{R}_t^{\text{base}} &= R_t^{\text{base}} \Delta R_c^{\text{forward}} \\ \tilde{p}_t^{\text{base}} &= p_t^{\text{base}} + \Delta p_c^{\text{forward}} \\ \tilde{T}_t^{\text{base}} &= \left[\tilde{R}_t^{\text{base}} \mid \tilde{p}_t^{\text{base}} \right] \end{aligned} \tag{11}$$

This target pose is then tracked by the controller described in subsection 4.2 and we update $T_t^{\text{base}} \leftarrow \tilde{T}_t^{\text{base}}$.

4.4 Base Control

The omnidirectional base controller tracks a desired velocity coming from the operator at 10 Hz by computing a trapezoidal velocity profile with different acceleration and deceleration limits. This velocity profile is tracked by a low-level controller that comes with the off-the-shelf base. The translational and rotational accelerations are limited to $0.5\,\text{m/s}^2$ and $2.5\,\text{rad/s}^2$ respectively, while the decelerations are limited to $1\,\text{m/s}^2$ and $2.5\,\text{rad/s}^2$. The maximum acceleration is kept relatively low to make the base begin moving smoothly, while the deceleration is allowed to be larger so the operator can quickly stop if they need to. The translational and rotational velocities are limited to $0.2\,\text{m/s}$ and $0.5\,\text{rad/s}$ respectively.

The operator controls the base with a joystick on the right-hand controller. The operator simply pushes the joystick to set the desired translational velocity of the base. To rotate the base, the operator pushes down on the joystick and moves it either left or right to control the angular velocity. Translational velocities are set to 0 while AVATRINA is rotating to avoid accidental translation of the base, which operators found to be jarring. The operator can also switch the base into a "fast" mode using the in-headset menu shown in Fig. 7. This increases the maximum translational speed of the base to $1.0\,\text{m/s}$, and enables 2D LiDARs that automatically slow and eventually stop the base if a nearby object is detected.

5 Vision Subsystem

This section discusses our approach to the design of the vision system that maximizes visual immersion. Several methods are commonly used to provide this feedback in teleoperation systems, including displaying a stream from a static camera on the robot to a monitor [35], or showing the operator a reconstruction of the remote environment [8, 11, 45, 62]. However, using only a monitor to display feedback reduces the operator's depth perception, and 3D reconstructions exhibit artifacts from limited resolution, occlusion, and noise.

Our approach is to stream a stereoscopic view of AVATRINA's environment to the operator's HMD, which is a common strategy for teleoperated robots since it aids with depth perception [36, 39, 56]. We also match the operator's HMD movement to AVATRINA's head movements, which provides greater effective field of view and improved immersiveness. AVATRINA also uses a custom rig to adjust AVATRINA's

interpupillary distance (IPD) to match the operator's, and we present a human subjects study to verify that this modification improves the operator's hand-eye coordination.

5.1 VR UI

The VR user interface is rendered with Unity 2021. It is configured with OpenXR to be compatible with commercial VR products such as the Valve Index, HTC Vive Pro, and Meta Quest 2. The two video feeds from the robot's eyes are each displayed on a plane for the operator's corresponding eye, which has been reported to reduce motion sickness compared to restricting the operator's field of view to only the captured images [25]. Additionally, similarly to [56], the screen's pose is determined by AVATRINA's head orientation rather than being fixed to the HMD orientation, which reduces motion sickness and compensates for head motion latency. To prevent double vision and provide accurate depth perception, the planes are set apart a distance equal to the user's IPD. By pressing the menu button, users access an augmented menu that allows the operator to activate assistive operational modes as shown in Fig. 7.

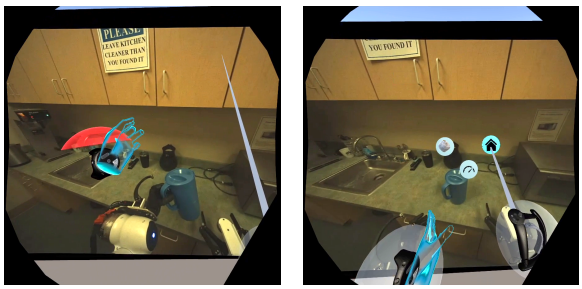


Fig. 7: (Left) View from the operator's perspective. A force-feedback sphere (described in subsection 6.2) in red shows the operator the force on AVATRINA's left hand. The blue ghost hand shows the sensed configuration of the operator's left hand. (Right) Heads-up display menu enables semi-autonomous functionalities: texture-sensing mode (left), arm "homing" (right), and base speed adjustment (center) icons. [Best viewed in color.]

5.2 3 DoF Head

The 3 DoF anthropomorphic head is driven by three Dynamixel XM430 servo motors for roll, pitch, and yaw rotations, whose rotation axes intersect at the same point to mimic the function of a human neck. The head follows the HMD orientation by computing joint velocity commands for the servo motor controller (PI velocity control) at 50 Hz. If any collisions are detected, no attempt is made to track the target and the controller does not move the head.

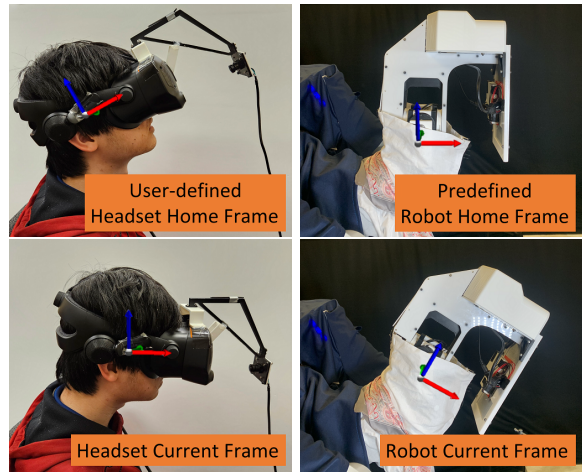


Fig. 8: A customizable home position lets the operator choose more comfortable head postures. Once the operator sets the home position, the rotation from the *robot home* to the *robot current* frame tracks the rotation from the *headset home* to the *headset current* frame. [Best viewed in color.]

A relative transform mode of control is used to reduce the neck fatigue of the operator in tasks that require holding an uncomfortable pose. By pressing the home button, AVATRINA's head returns to the predefined default ("robot home") position (top right of Fig. 8) and the HMD's current pose is registered as the "headset home." After the robot head is homed, it tracks the relative rotation from the headset home pose to the operator's current pose, as shown in Fig. 8. For instance, in tabletop tasks, the operator may look up, home the head, and then look forward. The relative motion makes the robot head face downwards towards the table, so the operator does not have to maintain an uncomfortable looking-down pose for a prolonged period of time.

5.3 Adjustable IPD

A custom stereoscopic camera was developed that mounted two Allied Vision Alvium 1800

U-500C 5 MP cameras outfitted with 1.67 mm focal-length wide-angle lenses to AVATRINA's head. This camera and lens combination offers an approximately 120° × 100° field-of-view. Both cameras are mounted to a linear rail and connected to a pair of Actuonix L12-R linear actuators that can adjust the distance between the two cameras with approximately 1 mm of resolution, as shown in Fig. 9. This mechanism can achieve baseline distances ranging from 49.44 to 69.88 mm. At initial setup, the baseline distance is matched to the user's interpupillary distance (IPD), which has been found to improve spatial perception in virtual reality [66].

The cameras are set to a fixed exposure time of 8 ms to ensure high frame rate and low latency, and use built-in gain and white balance adjustment to adapt to varying lighting conditions. Since the mounting is not perfectly parallel, the images are rectified before transmission. To further improve frame rate and latency, the video streams are down-sampled using bilinear interpolation and concatenated giving a 2072 × 778 (1036 × 778 for each eye) video stream. FFMPEG is used to encode the video stream using the `vp8` codec with a specified maximum bitrate. Using a typical network connection, with a maximum bandwidth of 13 Mbps, this stream typically achieves a framerate of 30 fps and a latency of ∼220 ms.

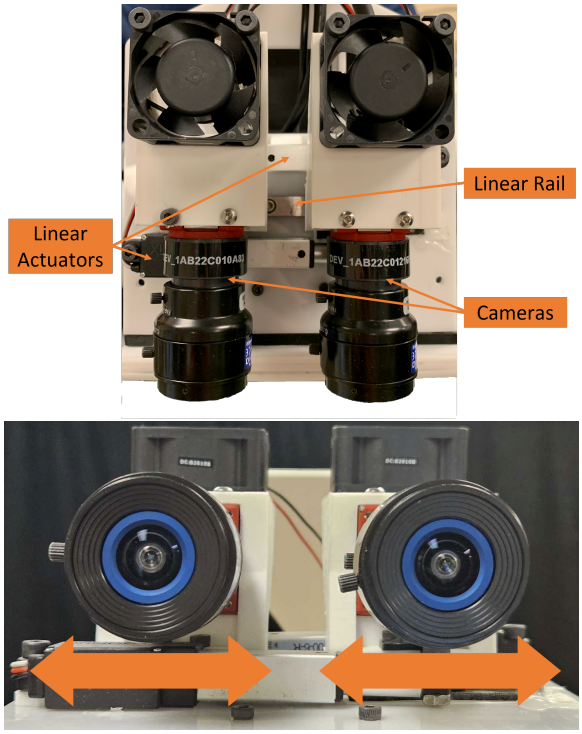


Fig. 9: Mechanism that enables live adjustment of AVATRINA's IPD to match the operator's. One linear actuator is attached to each camera and can move it left and right along the linear rail.

5.4 Human Subjects Studies

We conducted human subject studies to evaluate the possible advantages of a 3 DoF head and to evaluate the effects of mismatch between IPD and stereo baseline on hand-eye coordination. Both studies use a peg-in-hole task shown in Fig. 10.

The effect of IPD mismatch on depth perception and user comfort in VR has been investigated both geometrically and empirically [18, 52, 66]. Previous studies have found that mismatch between the user's IPD and the rendered IPD can result in inaccurate depth perception, although the effect is less than would be predicted by geometric analysis. Stereo televisualization introduces another source of potential error: the baseline between stereo cameras may not match the operator's IPD. We therefore perform a human subjects study to investigate the effect of different levels of mismatch between AVATRINA's IPD and the operator's on hand-eye coordination.

Besides stereo disparity, operators observe the remote scene from multiple viewpoints, providing depth cues from parallax and the ability to peer around occlusions. Humans use head, torso, and body movements to change viewpoint. This ability has been incorporated into some telepresence systems by including a movable neck and head assembly to move the robot's "eye" cameras. The effect of varying the number of DoFs of these assemblies has been previously investigated in [56], which found that increasing the number of head DoFs from 0 to 3 to 6 improved success rate and speed of completion of a peg-in-hole task. However, this study had a few limitations. First, the sample size was small and subjects were members of the research team. Moreover, the 6 DoF robot used in this study is likely prohibitively expensive for many telepresence robots and most entries in the XPRIZE exhibited 2 or 3 DoF heads. We

address these limitations by conducting a larger study and testing smaller changes in head DoFs (0 to 2 to 3) to provide more fine-grained data about the effect of head DoFs on manipulation.

We formulated the following hypotheses a-priori about our televisualization system:

- **H1:** The larger the mismatch between the user's IPD and AVATRINA's IPD, the more slowly subjects will complete each of the peg-in-hole tasks.
- **H2:** Subjects will complete each task more quickly as the number of head DoFs increases.
- **H3:** Subjects will examine the peg board for less time before starting the task as the number of head DoFs increases.

To test H1, H2, and H3 we designed a peg-in-hole task with 3 red holes and 3 blue holes. Fig. 10 shows the setup used to test subjects. The peg used had an outer diameter of 16.1 mm while the inner diameter of the holes was 20.7 mm. We used a peg with a large tolerance so that novice operators could learn the task quickly without extensive training. The robot was constrained to use one arm which could translate in 3 DoFs and rotate in only 1 DoF (horizontal axis) to isolate the effects of depth perception and to avoid singularities. We recruited 16 subjects (9 male, 7 female) from the university's student population. Subjects self-selected their fitness for the study after hearing the procedure. No subjects opted out of the study. Subjects were of age 19–30 (mean: 24.6) and self-reported their familiarity with AVATRINA to be an average of 2.9 out of 7 on a Likert scale [55]. Two subjects had previously been trained in how to use the robot in prior studies. Each subject measured their own IPD using a ruler and a mirror, and then fine-tuned the HMD to find the most comfortable IPD setting. A researcher then trained the subject to use the head and arm, which lasted approximately 20 minutes.

First, to test the effects of IPD mismatch, the head pose was fixed and the subjects completed tasks under four settings of AVATRINA's IPD: Matched, Average (62.72 mm [14]), Minimum (49.44 mm), and Maximum (69.88 mm). This was first done with the red holes and then repeated with the blue holes. The conditions were tested in a randomized order, which was unknown to the subjects. Then, to test the effects of robot head DoFs, the IPD was set to Matched and the subjects completed tasks under three head DoF conditions (0, 2, and 3), first with the red holes and then with the blue holes. Each subject experienced the conditions in a randomized order and was informed of the condition since they had to consciously use their neck to use the different DoFs. During the DoF trials, the pegboard was obscured until the subject said they were ready, and the time between the reveal of the pegboard and the subject's first motion was recorded as that trial's "planning time." Trials on the red holes were used as training trials to reduce the impact of learning effects; only times on the blue holes were recorded and are analyzed here.

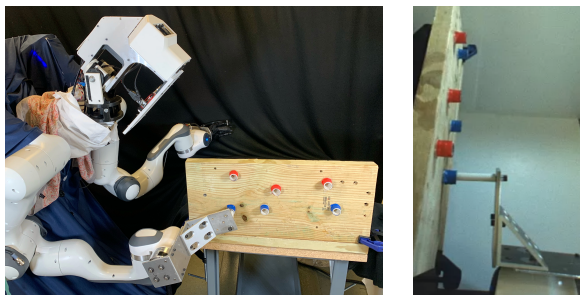


Fig. 10: (Left) A peg-in-hole experiment is used to examine how the robot's neck and stereo camera settings affect depth perception. Two sets of holes were used to reduce learning effects between trials. (Right) Cropped first-person view as the operator completes the first hole of the trial. [Best viewed in color.]

For each trial, the time the subject took to insert the peg into each hole was recorded as their task completion time, starting from the subject's first movement of AVATRINA's arm. After each trial, the subject removed their headset to fill out a modified version of the presence questionnaire [72] to gauge their subjective feelings of presence in the remote environment.

Fig. 11 shows how subjects' task completion times changed as the absolute IPD mismatch changed for each hole. To test **H1**, we ran a generalized estimating equations (GEE) regression [31] grouped by subject with an autoregressive covariance structure [54], where distance was computed as the distance between trial indices. After applying the Bonferroni correction [5], we found significant correlation between IPD mismatch and completion time for Hole 2 ($\beta = 0.434$ s/mm, $\sigma_{\mathrm{M}} = 0.173$ s/mm, $p = 0.0372$) but no significant

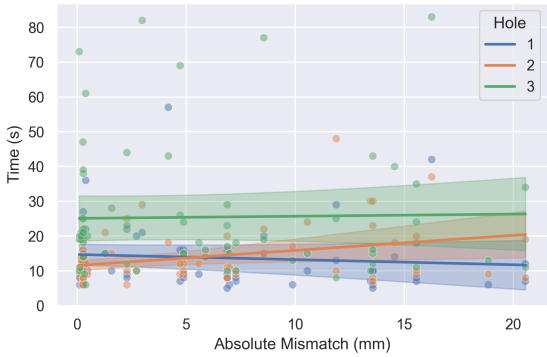


Fig. 11: Effect of IPD-baseline mismatch on peg-in-hole task completion time for each of the 3 holes. Lines show fitted regression models and shaded regions indicate 95% confidence intervals. Dots show individual measurements. [Best viewed in color.]

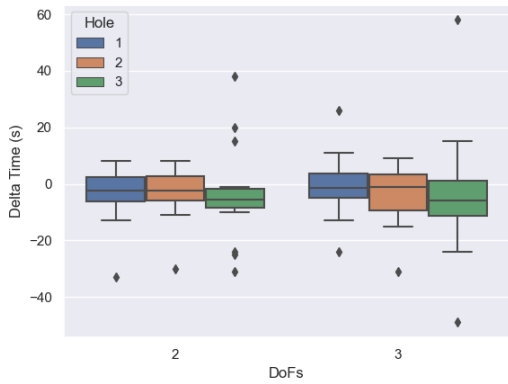


Fig. 12: Change in task completion time for each hole with respect to each subject's task completion time at 0 DoFs. [Best viewed in color.]

effect for Holes 1 and 3 ($p = 1.0$ for both), providing *weak evidence to support* **H1**. We note that there are more outlying data points for Holes 1 and 3 compared to Hole 2, and hypothesize that this may be because insertion into Holes 1 and 3 requires the robot's arm to be at less dexterous configurations than for Hole 2. This may contribute more variability to subjects' performance on these tasks, overwhelming the effect of IPD mismatch.

Fig. 12 shows the subjects' changes in task completion time from the 0 DoF condition for the 2 and 3 DoF conditions. We ran a Shapiro-Wilk test [59] on the change in completion time from 0 DoFs for each hole at the 2 and 3 DoF conditions, which indicated significant deviation from normality. To test **H2**, we ran a Friedman test [55] for each hole with Bonferroni correction [5] and found no significant effects ($p = 1.0$, $p = 1.0$, $p = 0.140$ for Holes 1, 2, and 3 respectively). Post hoc pairwise one-sided Wilcoxon-signed-rank comparisons provided weak evidence that completion time for Hole 3 is smaller when using 2 DoFs compared to 0 DoFs ($M = 3.94$ s, $SD = 16.54$ s, $p = 0.0719$) and when using 3 DoFs compared to 0 DoFs ($M = 4.00$ s, $SD = 21.35$ s, $p = 0.0877$), but not when using 3 DoFs compared to 2 DoFs ($p = 0.470$). This corresponds to a 14.7% and a 10.6% average reduction in task completion time when switching from a 0 to 2 DoF neck and from 0 to 3 DoF neck respectively. *These results weakly support* **H2** and corroborate previous research [56] showing that increasing head DoFs improves task performance on tasks that require depth perception and occlusion resolution, but with diminishing returns. Finally, we ran a Friedman test [55] on the planning times and found no significant effect ($p = 0.859$), providing no support for **H3**.

To gauge the subjects' subjective feelings of presence in the remote environment, we selected 8 of the most relevant questions from the presence questionnaire [72], and modified them slightly to better fit our experiment. Fig. 13 shows the survey questions, which were rated on a 7-point Likert scale [55], and subjects' responses for both the IPD and head DoFs experiments. For all questions, a higher score is better. For the IPD experiments, we bin the conditions into 5 mm bins. Across different IPD mismatches, we see little change in subjective scores except for questions 3 and 4 where mismatches of $\geq$15 mm tend to consistently result in lower scores than other conditions. These lower scores may indicate that at high IPD mismatch, subjects begin to notice distortions in their vision, but the relatively unchanged score of question 7 indicates that they do not feel it affects their performance. For the head DoFs experiments, we observed relatively pronounced changes in subjects' answers to questions 4 and 5 when moving from the 0 to the 2 or 3 DoF conditions, indicating that in addition to improving task performance, subjects also consciously notice that they can move more freely to see objects

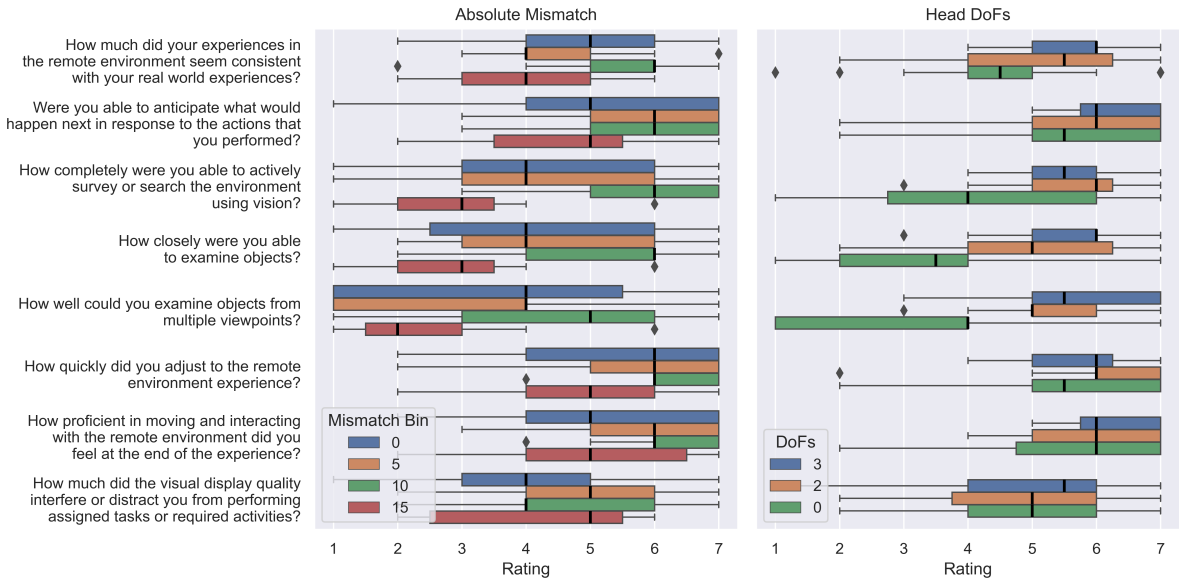


Fig. 13: Change in user ratings of different measures of presence across IPD mismatch and head DoF conditions. [Best viewed in color.]

when AVATRINA's neck has more DoFs. Consistent with our task performance findings, there seems to be no significant difference between the 2 and 3 DoF conditions.

Our experiments suggest that matching stereo baseline to the operator's IPD improves telemanipulation proficiency in certain regions of a robot's workspace. It may also improve comfort for long-term use as suggested by prior VR studies. Increasing head DoFs improves manipulation proficiency as well, but pan-tilt may be sufficient for many tasks. Fixed-baseline cameras near the average human IPD may be satisfactory for some applications but may impair performance for operators with large or small IPDs.

6 Hand Control and Sensing

In this section, we describe how the operator controls and receives force feedback from AVATRINA's end effectors. The operator controls AVATRINA's parallel-jaw gripper by pulling a trigger on their right-hand VR controller. Control of the anthropomorphic Psyonic Ability hand is more complex, requiring retargeting from the SenseGlove's sensing to the hand's actuating capabilities.

6.1 Hand Motion Retargeting

The goal of hand motion retargeting is to map the motion of a high-DoF (21) human hand to a low-DoF (6) anthropomorphic gripper. The SenseGlove tracks 4 degrees of freedom (three in flexion, and one "splay") for each finger. The full hand motion retargeting problem is defined as finding a mapping G between the 20 SenseGlove readings q_{glove} to 6 gripper motor positions, $q_{gripper}$, shown in Fig. 14. Note that the gripper uses a transmission mechanism so that a single motor command bends both joints in each finger in a synergistic manner.

6.1.1 Methods

We seek a mapping of the form

$$G : (\mathbb{R}^4 \to \mathbb{R})^4 \times (\mathbb{R}^4 \to \mathbb{R}^2), \qquad (12)$$

where each of the four fingers maps from four SenseGlove DoFs to one Psyonic DoF, while the thumb maps from four SenseGlove DoFs to two thumb DoFs.

To find this mapping, we designed a set of 15 gripper poses, shown in Fig. 15, that cover the Psyonic's workspace and important grasps such as

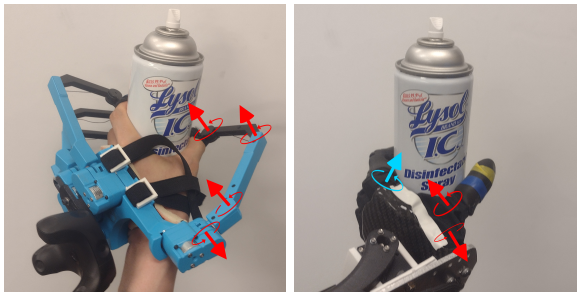


Fig. 14: Hand Motion Retargeting. Each finger is measured with 4 DoFs on the SenseGlove (left), but the Psyonic gripper (right) only has 2 DoFs for the thumb (red) and 1 DoF per every other finger (blue). [Best viewed in color.]

key and power grasps. To adapt to each operator's hand, a calibration procedure is performed. The operator first mimics the gripper poses while wearing the SenseGlove and the corresponding SenseGlove readings are recorded. Not all fingers are considered for each calibration pose, since they may be difficult to imitate or irrelevant to the current calibration step. The mapping G is then learned by supervised learning to minimize calibration error. Note that we learn a mapping for each finger and thumb individually.

The gripper poses used were designed to match the human poses visually; however this may not capture the operator's intent. For example, for a pinch grasp on a thin object, the operator may want to pinch harder on an object, but once the operator's fingers touch, the SenseGlove is not able to detect an intent to apply extra force. To compensate for this, we tweaked the target gripper joint angles to form a set of "biased" calibration poses in order to increase grasp success. Fig. 16 demonstrates how joint biasing is set up.

6.1.2 Results

We evaluate the calibration performance on a different set of 4 grasps, shown in Fig. 17. First, we hand designed Psyonic poses for the four grasps. Then, each operator is instructed to physically perform each grasp while wearing the SenseGlove, and the SenseGlove joint angles are recorded. Finally, the recorded SenseGlove angles are transferred to the Psyonic using the mapping calibrated for the operator. These angles are compared against the hand-designed poses using

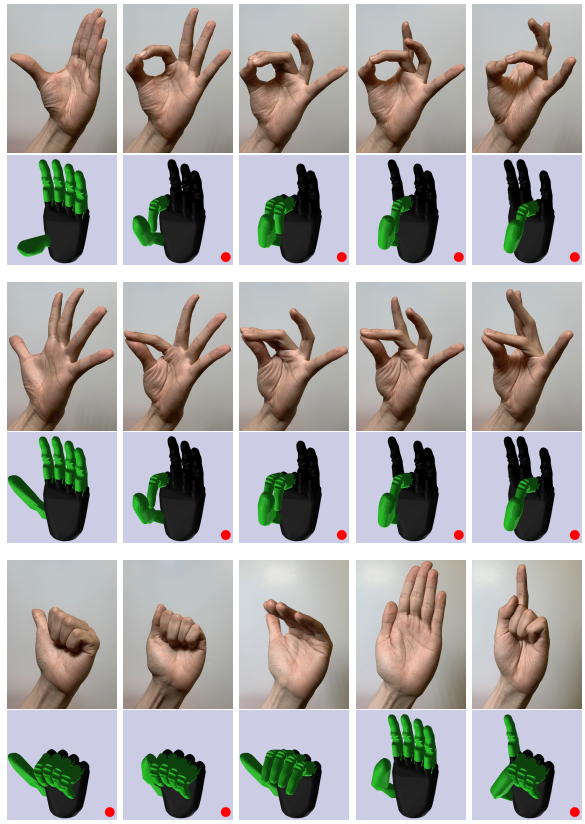


Fig. 15: The hand calibration poses used. Human hand and target Psyonic pose are shown. Green highlight shows fingers that are considered for the given calibration pose. A red dot indicates that joint angle biasing was applied to the pose. [Best viewed in color.]

Fig. 16: Example for how the joint angles were biased for hand calibration. Bias effect is exaggerated for clarity. [Best viewed in color.]

two metrics: the mean absolute error (MAE) of the joint angles, and the grasp success score. The grasp success score is defined by using the Psyonic to pick up the physical object and repeatedly shaking the object with increasing intensity, as shown in Fig. 18. The number of shaking rounds before the object is dropped, up to 5, is recorded as the grasp success score. Two shaking trials are conducted for each grasp.

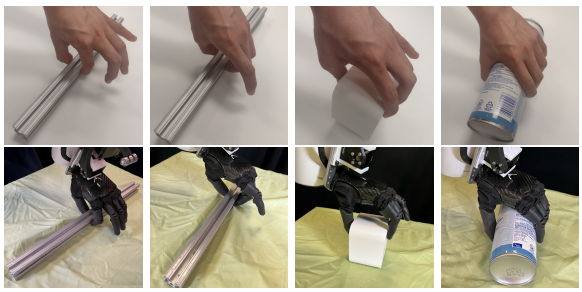


Fig. 17: Pairs of poses used to test grasping. From left to right: Pinch 1, Pinch 2, Cube, Power.

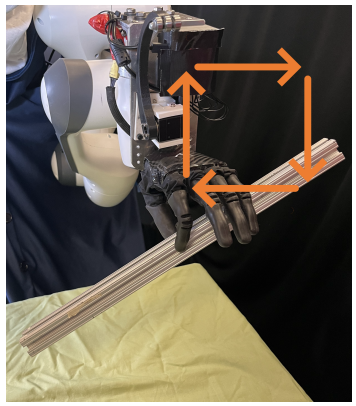


Fig. 18: Illustration of shaking procedure. Robot picks object up from a predefined pose (in Fig. 17), then shakes in a square motion in the air with increasing intensity.

We experiment with two different regressors: Gaussian process regression and kernel ridge regression [49], and test each regressor using the unbiased and biased calibration poses. The calibration results for 3 researchers on this project are shown in Table 5. Overall, the two regressors perform similarly in grasp success, and both of them are helped by biasing the calibration poses. In terms of MAE, kernel ridge regression performs better than Gaussian process regression. Using biased calibration increases grasp success, at the cost of increasing MAE. We decide to use biased calibration since the slight mismatch in angles could be compensated for by the operator.

6.2 Force Feedback

AVATRINA provides feedback to the operator about the forces on its arms and fingers using several channels, including audio cues, vibrations, and resistive forces on the operator's fingers. Other avatar systems have implemented exoskeletons that can directly apply forces to the operator's arms to mirror the forces the avatar experiences, which can be effective, but require custom hardware and careful calibration [30, 35]. We opt for operator feedback that can be rendered using off-the-shelf components.

Arm forces are estimated using the measured joint torques of the Franka Emika Panda arms using a damped pseudoinverse method:

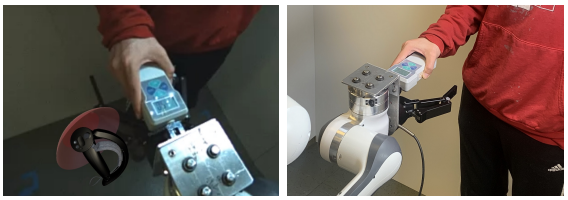
$$\begin{aligned} (J^\intercal)^\dagger &= \left(JJ^T + \epsilon I\right)^{-1} J \\ F_{est} &= -(J^\intercal)^\dagger(\tau - \tau_g) \end{aligned} \tag{13}$$

where $\tau \in \mathbb{R}^7$ are the measured joint torques, $\tau_g \in \mathbb{R}^7$ is the generalized gravity torque on each joint due to the robot's own weight, computed by the Franka API, and $F_{est} \in \mathbb{R}^6$ is the computed 6D wrench. $J \in \mathbb{R}^{6\times7}$ is the Jacobian, and $\epsilon = 0.0001$ is a small constant trading off between accuracy and stability when near singular configurations. Measured forces are rendered through two means to the operator: The magnitude of the force is rendered as a vibration on both the Valve Index controller and the SenseGlove as a linear mapping between zero and maximum vibration and minimum and maximum sensed forces (3-15 N). It is also visually rendered in the HMD using an augmented reality *force sphere*, shown in Fig. 19. The opacity of the force sphere is proportional to the sensed force magnitude and the axis points opposite the sensed force direction. The resulting effect is that of a wall around your hands in VR that "solidifies" as the sensed force increases. These two modalities combine to give the operator accurate and visceral feedback about the forces

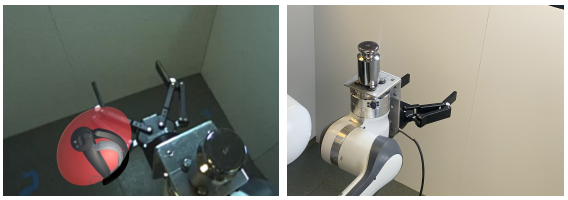
Table 5: Gripper calibration evaluation on four grasp attempts. Scoring represents the number of shaking cycles survived, averaged across six trials with three different operators. Error is grasp pose MAE averaged over all joints.

Biasing	**Model**	**Grasp Success Score**				**Error (rad)**
		Pinch 1	Pinch 2	Cube	Power	
manual	None	3.50	5.00	4.00	5.00	-
unbiased	GP	2.67	**5.00**	3.00	**5.00**	0.317
unbiased	KRidge	2.17	4.67	2.00	**5.00**	**0.217**
biased	GP	**4.17**	4.83	3.33	**5.00**	0.402
biased	KRidge	3.50	**5.00**	**3.50**	**5.00**	0.282

AVATRINA's hands are experiencing, improving the operator's situational awareness.



(a) VR view (left) and real world view (right) of a 5N force being applied to the end effector from the front.



(b) VR view (left) and real world view (right) of a 9.8N force being applied to the end effector from the top.

Fig. 19: Examples of force-rendering partial spheres on the right arm. The direction of the sphere indicates the direction from which the force is being applied. A stronger force is rendered with a more opaque and saturated color, as can be seen in the sphere in (b) compared to the sphere in (a). [Best viewed in color.]

To render the sensation of grabbing an object, the SenseGlove can provide independent finger resistances to motion. The Psyonic gripper provides 6 force measurements from each of its fingers: while any of these measurements exceeds a threshold, we fully lock the corresponding SenseGlove finger, preventing the operator from further closing that finger. This feedback transmits the sensation of grabbing objects from AVATRINA to the operator.

7 Augmented Capabilities

In addition to the immersive components of AVATRINA, we enhance the teleoperation experience of the operator through assistive functionalities. These include shared control and supervisory control modes, texture sensing and rendering, and proximity sensing and warnings.

7.1 Shared Control

AVATRINA has several assistive functions accessible via a GUI in the VR interface to supplement direct teleoperation. Fig. 7 shows icons for a subset of them that were used in the XPRIZE finals competition.

To help achieve fine manipulation, the operator can modify the translation from their movement to the robot's movement by modifying the scaling of their motions and by enabling different virtual fixtures [6] to constrain the robot's motion. To achieve a desired scaling δ, we multiply $\Delta R_c^{\text{forward}}$ and $\Delta p_c^{\text{forward}}$ in (11) by δ before using them to compute the next target.

To constrain the robot's motion, a "constraint transform", T_χ^{base}, is stored. This transform is updated to the current end-effector target, T_t^{base}, each time the operator modifies the set of active constraints. When any constraints are active, the target transform T_t^{base} of (11) found at each time step is projected onto T_χ^{base} by finding the

relative transformation T_t^{χ} and setting the constrained dimensions to 0. This modified transform is then applied to T_{χ}^{base} to find the final target end-effector transform. We allow the operator to modify T_{χ}^{base} (by moving the end-effector) so that the operator can orient the directions of the constraints relative to the environment.

Across our two official runs at the XPRIZE competition, these constraints were modified a total of 5 times, concentrated toward the end of both runs on Task 10, to achieve steady scanning of objects obscured from view.

7.2 Semi-autonomous Controls

In addition, AVATRINA supports user-triggered autonomous capabilities (i.e., supervisory control). The most commonly used is "homing," which moves both arms to a predetermined, dexterous, tucked configuration, shown in Fig. 1. This allows the operator to recover from unfavorable configurations they may encounter during manipulation and also makes the robot narrow so that it can fit through tight passages. When this action is enabled, AVATRINA uses Klampt [26] to query a sampling-based motion planner (implemented as a combination of a probabilistic roadmap of trees [2] and a single-query bidirectional roadmap with lazy collision checking [63] to connect the roots) to find a plan that takes AVATRINA's arms from their current configuration to the home configuration. This plan is then tracked using joint-space impedance control. The operator can cancel this execution at any time by pressing either clutch. Additionally, if the arm deviates from the planned trajectory too much, execution will automatically be canceled to avoid applying large forces to AVATRINA or the environment. Other autonomous capabilities include "snapping," which causes AVATRINA to automatically align the forward direction of its end effector with the normal of a plane detected in the environment [42], point-and-click grasping, and object placing.

Across our two XPRIZE finals runs, the homing action was used 7 times. Operators used this action to tuck AVATRINA's arms tightly to fit through narrow passages and to recover the arms when they were in unusual configurations. We did not train the operator to use additional autonomous functions for the sake of training brevity.

7.3 Texture Rendering

Sensing texture is an emerging component of telepresence systems that can allow operators to identify objects that are out of sight. Texture sensing and rendering on AVATRINA is achieved by mapping a heightmap of a sensed surface into vibrotactile and auditory cues. This surface heightmap is captured through an Intel RealSense L515 camera mounted to AVATRINA's right wrist. This captures a heightmap with a height resolution of 1 mm, a resolution of 1024×768 pixels, and a FoV of $70° \times 55°$. A sample heightmap taken during the XPRIZE competition is shown in Fig. 21. To initiate texture sensing, the operator uses the GUI shown in Fig. 7 to enter "texture mode," causing AVATRINA to autonomously plan and execute a path to a saved configuration that positions this camera so that it looks straight down. Texture mode also constrains the right arm to only move in the horizontal plane so the operator can easily sweep the camera across a wide area without focusing on keeping the camera level and at the proper height. As the operator scans, the sensed heightmap is rendered in VR as a 3D mesh, shown in Fig. 20, similarly to [48]. The operator can then release the clutch and hover their controller over that mesh to feel the sensed texture. Texture is rendered by finding the line between the previous and current controller positions at each time step and computing C, the sequence of n heightmap grid cells crossed by the orthogonal projection of this line into the horizontal plane, using Bresenham's line algorithm [7]. Using the values in the heightmap z_i, we compute a roughness intensity:

$$\text{intensity} = \lambda \sum_{i=2}^{n} |z_i - z_{i-1}| \tag{14}$$

where λ is a heuristically tuned scaling factor. This intensity is then mapped to the volume of a looping brown noise clip at every frame, while the accumulated intensity over 4 frames is mapped to the intensity of the vibration of the controller hovering over the surface. This creates a vibratory and auditory experience that allows the user to distinguish between different surface roughnesses

remotely without requiring the robot to touch or see the target surface.

7.4 Ultrasonic Proximity Warnings

Four MaxBotics MB1604 ultrasonic rangefinders are mounted to AVATRINA's base to measure the distances to the closest obstacle in front of, behind, to the left, and right of AVATRINA. This sensor has a range of 2 cm to 5 m and a nominal accuracy of 1%. The measured distances are streamed to the operator station computer. A beep sound is played in a loop if the operator moves towards an obstacle in close proximity, with pitch, speed, and volume that scale inversely with the robot's distance to the obstacle. The beep will come from the location of the obstacle relative to the robot: for instance, if the obstacle is to the right of the robot, the beep will come from the right side of the operator.

8 Recipient Interaction

AVATRINA supports communication between the operator and recipients in the remote environment. It supports two-way audio communication via a microphone and speaker, and displays a reconstruction of the operator's face to convey emotions through facial expressions.

8.1 Facial Rendering

While a VR interface can improve immersion for the operator, most HMDs partially occlude the operator's face. Consequently, remote interaction with others is harmed due to the absence of complete facial expressions, which fulfill a crucial role in communication [13]. To resolve this, we build a facial reconstruction pipeline which employs talking head animation pipelines to recreate a credible HMD-free view of the operator. Siarohin et al. [61] developed a first order motion model (FOMM) for performing image animation between visually similar images. Inspired by this approach, Rochow et al. [53] propose a method to render an HMD-free view of an operator by animating a set of static pictures. However, the operator's HMD produces heavy occlusion of the face, making it infeasible to apply FOMM directly, as there would be missing facial keypoints in the occluded facial image. Consequently, they heavily modify their HMD to include in-headset infrared cameras to capture gaze direction for facial animation.

Our facial rendering pipeline obviates the need for heavy HMD modifications, only requiring a small camera pointing towards the operator's face to be attached for capturing lower face keypoints. To create the top half video, we record a video of the operator neutrally blinking with the HMD off, aligning it to the HMD's viewpoint by using keypoints extracted from their chin and mouth. The resulting composited image is then input to the FOMM animation network which produces the final reconstruction of the operator's face. The FOMM network is able to handle moderate misalignments, as shown in Fig. 22. The final reconstruction is then transmitted to AVATRINA via Zoom[5] and displayed on AVATRINA's head. The facial rendering pipeline ran at about 33ms per frame on a gaming laptop equipped with a RTX 3070 mobile graphics card, while the audio and video latency of the zoom interface are reported to be around 200-300 ms depending on network conditions. Sample outputs of this pipeline, as well as a brief analysis of its capabilities and shortcomings are provided in Appendix B.

8.2 Audio Communication

Audio captured from the HMD is transmitted to AVATRINA using the same Zoom call that transmits the final facial reconstruction, and is played through a front-facing speaker. This proved to be a robust solution for audio communication, as Zoom can automatically adapt to different network conditions, and automatically reconnects if Internet connection is lost. The audio captured from the robot's stereo microphone (Shure MV88+) is played using the HMD's over-the-ear headphones.

9 Software Architecture

AVATRINA is a complex system that requires many hardware and software components across operating systems and programming languages to communicate with each other. To manage this complexity, we designed AVATRINA's software to be modular with respect to both different robot

[5] https://zoom.us/

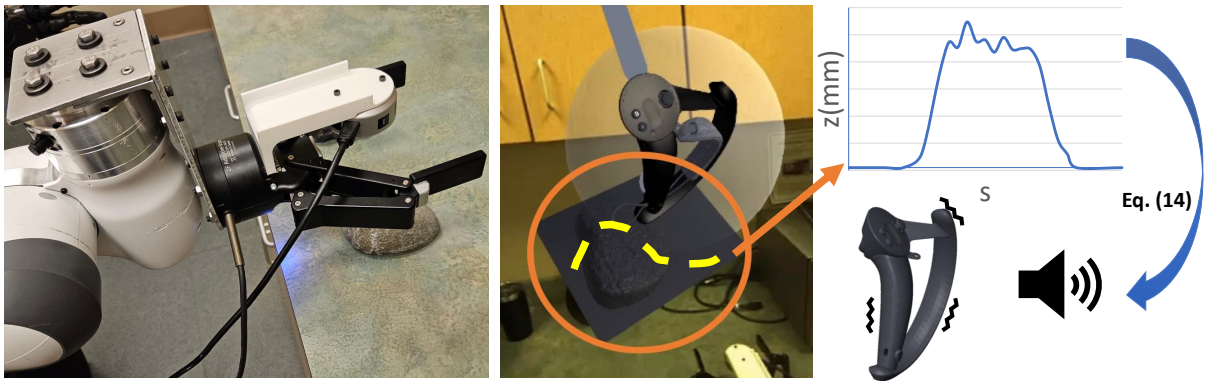


Fig. 20: A rock being scanned (left) and reconstructed as an augmented reality mesh displayed in the headset (right). Note the detail on the rock's surface despite the short-range LiDAR noise. The operator is also able to "feel" the virtual object as the controller moves along the surface of the mesh, with the texture under the scanner mapped to vibrotactile and audio feedback. An example trajectory is shown as a yellow dashed line. This trajectory creates a curve of height in time (indicated by the orange arrow), which then uses equation Eq. 14 to create the sound and vibration patterns felt in the controller. Note that these patterns are influenced by both the rock's surface and the speed with which the operator moves along its surface, like in [60] [Best viewed in color.]

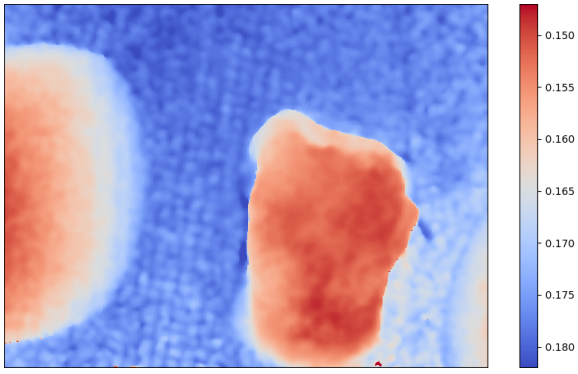


Fig. 21: A raw frame taken from the depth camera. Red represents pixels closer to the camera; blue represents pixels further from the camera (distance in meters). Two rocks are present in the depth image, and the camera is able to capture shape information as well as macro-scale texture information: The left rock is smoother and has a sloped surface, while the right rock has a bumpier top surface. [Best viewed in color.]

and interface hardware to enable rapid prototyping and testing. Fig. 23 shows a block diagram describing how the different hardware and software components of AVATRINA communicate. This section describes how AVATRINA's software architecture achieves this goal, as well as the networking infrastructure that enables reliable communication between the operator station and AVATRINA.

9.1 Network Architecture

TRINA uses three main channels to communicate between the operator station and robot: Zoom, SmartFox, and WebRTC. Each of these channels uses broker servers to establish connections, so AVATRINA can be controlled from anywhere with a stable Internet connection without any custom network setup.

To stream audio between the operator and robot, and to stream the facial reconstruction of the operator to the robot, we use standard Zoom clients. We also found it useful during debugging to have an audio channel separate from the main communication link, which allows for verbal communication between the operator and recipient even if an error occurs in the main link.

A SmartFox server[6] manages the communication of state and control commands between the operator station and AVATRINA. The operator station streams messages to the server containing the operator's inputs, such as the current pose of the operator's head and hands, and these are relayed to AVATRINA. AVATRINA streams

[6] https://www.smartfoxserver.com/

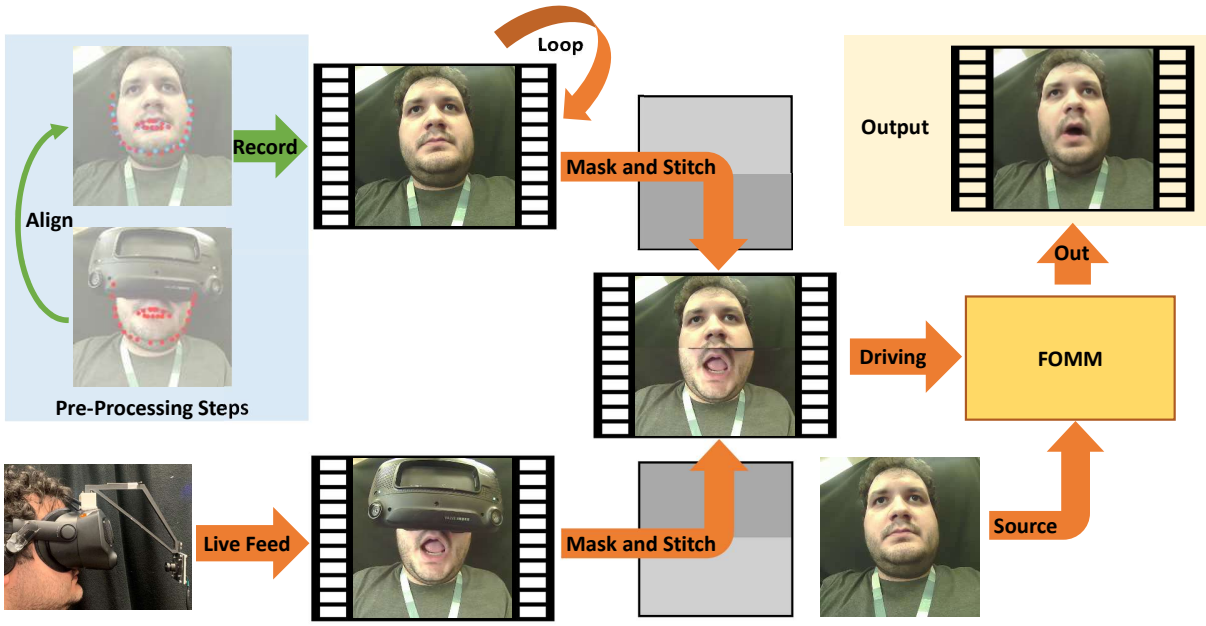


Fig. 22: The facial reconstruction pipeline. During the preprocessing stage, lower face facial landmarks of the operator (red dots) are used to manually capture an unobstructed video of the operator blinking from the HMD camera's perspective by aligning them with landmarks extracted during the recording (blue dots). The steps in green are performed manually. Online, the headset camera feed is stitched with the masked looping blinking video to generate the driving video input to the image animation pipeline, FOMM [61], which animates a neutral picture of the operator (source). Note how a slight misalignment in the stitched image (center) still produces plausible outputs. [Best viewed in color.]

a JSON representation of its current state to the operator station and also provides other control and UI information. For example, an unreachable target end-effector pose is signaled to the operator by tinting the color of the VR screen.

A WebRTC peer-to-peer channel is used to stream AVATRINA's stereo camera video to the operator station. When the operator station starts up, it sends a Session Description Protocol (SDP) offer using SmartFox every 5 s until it receives a handshake message from AVATRINA. Once this handshake has occurred, a peer-to-peer connection between AVATRINA and the operator station is established. If a disconnection is detected, the operator station resumes its loop of sending SDP offers to enable automatic reconnection.

9.2 Centralized Middleware

A key design goal is for AVATRINA to accommodate multiple grippers, multiple sensors, and multiple user interfaces, such as different VR interfaces, mouse and keyboard commands, and a web interface (in development). It is also a research platform for both teleoperation [42] and semi-autonomous operation [74]. To enable a smooth development process through design variations, we introduced a middleware called Jarvis that serves as a common and flexible interface layer between diverse computational elements.

As opposed to other packages like ROS [51], Jarvis uses a centralized key-data store to manage robot state. This database, implemented in RedisJSON, makes it easier to inspect the state of the robot at a glance and due to its non-typed nature, facilitates rapid prototyping. Jarvis also enables developers to define APIs that encapsulate related functions on the robot, which reduces cognitive complexity.

Jarvis' main functionality is to provide users a uniform layer of access to APIs. Through a common `jarvis` object, accessing robot functions is done through a `jarvis.<APINAME>.function()`

call. Fig. 23 shows how these APIs interface with the rest of the AVATRINA system. For instance, controlling the robot's physical state (i.e. moving arms, base, grippers, and neck) is done by accessing the Motion API. The Sensor API provides unified access to the raw data of all of the robot's cameras and range finders; the VR UI API manages communication between AVATRINA and the operator's headset interface; the Screen UI provides a local keyboard and mouse interface to control AVATRINA for debugging; and the Perception API processes information from the Sensor API to provide access to higher level perception data, such as object segmentation masks and possible grasp affordances.

Implementers of APIs can use different communication paradigms depending on bandwidth requirements. Set/get involves direct access to Redis-JSON elements. Procedure calls are processed through a Redis-RPC paradigm. Custom paradigms are available to bypass the Redis backend. For example, the motion-API implements an XML-RPC client for faster communication with robot hardware, and the Sensor API ferries its data (point clouds, RGB-D images) through Unix sockets.

During operation, different applications can then use these APIs to command AVATRINA. For example, in Fig. 23, the Direct Teleop app reads the abstracted user input from the Canonical Controller, and translates user motion into commanded robot motion. The app mediates between different modes of control, allowing the operator to start up actions like sending AVATRINA's arms to different configurations, or to directly control AVATRINA's arms using their own arm motion.

9.3 Modular Robot Controller

To enable rapid iteration of robot components, AVATRINA's control software is built so that arbitrary numbers of components are unified into a single robot. All components can be controlled at once, or each piece can be controlled individually.

The robot is represented as a set of *sub-robot* controllers. The sub-robots that make up AVATRINA are specified and configured in a JSON file. The software also builds a unified URDF model dynamically upon reading this file. Each controller implements the Klampt Robot Interface Layer (RIL) concept via the `RobotInterfaceBase` class, which supports common behavior like `startup, shutdown, loop`, and also procedures like `get/set_joint_config` for reading the joint values and commanding joint positions. Adding a new component simply requires implementing a new base controller class. Swapping components is also easy: to switch from the Robotiq gripper to Franka gripper, the only change needed is to bind the `right_gripper` component in JSON to the Franka gripper's controller class. The `right_gripper` part is then bound correctly to the correct controller when accessed by, for example, teleoperation code.

The majority of control code is written in Python to make prototyping easier. More computationally intensive tasks like inverse kinematics and collision checking are handled by Klampt [26] for performance. Some low-level controllers (such as the Franka torque controllers) are time-sensitive and are implemented in C++ as Python extension modules.

9.4 Modular User Interface

The interface between AVATRINA and the operator station is also encapsulated with an API to make it easy to experiment with input systems. The operator station API is defined by the `CanonicalController` interface, which specifies key control abstractions, e.g., commands to end effector transforms, head orientation, and base motions. Thus, we can switch seamlessly between control interfaces using thin wrappers. For example, we can easily switch between a VR-style input system to a simple mouse-and-keyboard UI for debugging, or disable the SenseGlove and use two VR controllers.

10 End-to-end Evaluation: ANA Avatar XPRIZE

Our team competed in the ANA Avatar XPRIZE finals competition using the AVATRINA robot in November of 2022. This competition required teams to train a new operator (a judge) on how to use their robot in 45 minutes to complete 10 different tasks in 25 minutes that ranged from navigation, to manipulation, to human interaction. Each robot was scored on its ability to complete the following tasks in a binary fashion (partial points were not awarded).

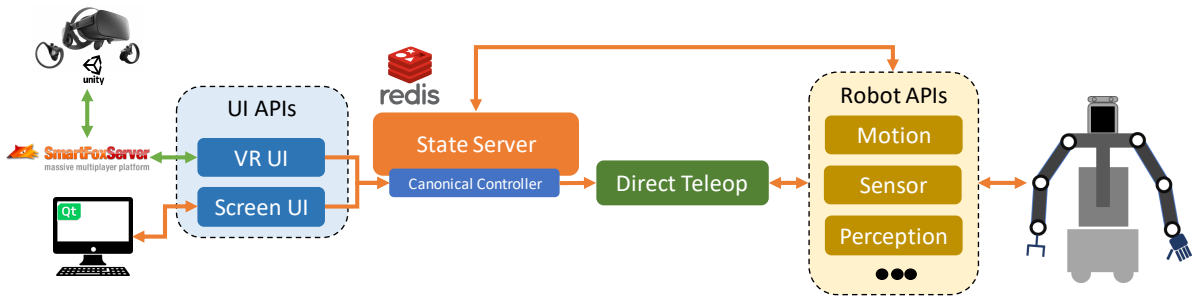


Fig. 23: A schematic of AVATRINA's middleware architecture. Orange arrows indicate communication within the AVATRINA system and green arrows indicate external network (WiFi) connections. Direct Teleoperation is the entry point into the system, allowing the operator to directly control AVATRINA's motion. The CanonicalController defines a standard interface that allows multiple devices to control AVATRINA. Using APIs to interface with AVATRINA's hardware allows higher-level software to be agnostic to specific hardware implementations, such as which robot arms are used. [Best viewed in color.]

1. Was the Avatar able to move to the designated area?
2. Did the Avatar introduce themselves to the mission commander?
3. Was the Avatar able to confirm (repeat back) the mission goals?
4. Was the Avatar able to activate the switch?
5. Was the Avatar able to move to the next designated area?
6. Was the Avatar able to identify the heavy canister?
7. Was the Avatar able to lift up and place the heavy canister into the designated slot?
8. Was the Avatar able to navigate through a narrow pathway to get to the designated area?
9. Was the Avatar able to utilize a drill within the domain area?
10. Was the Avatar able to feel the texture of the object without seeing it and retrieve the requested one?

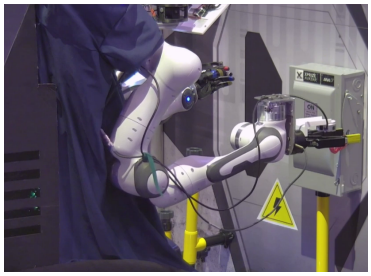
The operator and recipient judges could also award up to 3 and 2 points respectively based on subjective impressions. More details on the tasks and scoring criteria can be found in these papers that summarize the competition [4, 17]. Fig. 24 shows AVATRINA completing some of these tasks during the competition.

Table 6: Outline of the training procedure used at the XPRIZE competition.

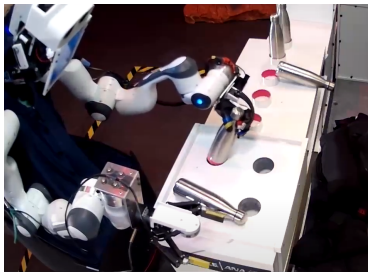
Step	Time (min)
Measure operator IPD.	1
Show operator video of AVATRINA being teleoperated.	2
Don and calibrate the SenseGlove.	7
Don and adjust the HMD.	2
Learn and practice controlling AVATRINA's head.	2
Learn and practice clutching each arm.	4
Learn and practice "homing" the arms.	1
Show warnings when AVATRINA enters a fault state.	1
Learn and practice moving AVATRINA's base.	2
Practice manipulating canisters and feeling weight.	10
Practice picking up and using drill.	8
Practice scanning for and feeling rock texture.	5

10.1 Team AVATRINA Operator Training Procedure

During the competition, teams had 30 minutes to set up their robot and operator station, and 45 minutes to train a judge recruited by XPRIZE to use the robot to complete the 10 tasks as quickly as possible. To optimize our use of this time, we developed a training protocol to get the operator comfortable with controlling AVATRINA and allow them to practice the competition tasks. We rehearsed this procedure many times before the competition so that each team member knew their assigned role. One person was responsible for walking the operator through each of the basic functionalities of AVATRINA, then a different team member walked them through strategies used to complete each task. Table 6 outlines the training procedure.

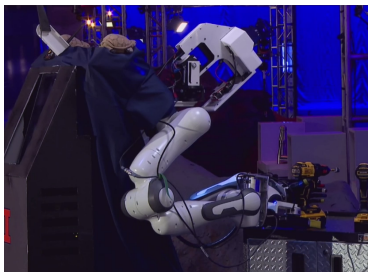


(a) Task 4: Switch



(b) Tasks 6 and 7: Canister

(c) Task 8: Narrow Passage



(d) Task 9: Drill Pickup

(e) Task 9: Drill Use

(f) Task 10: Rock Texture

Fig. 24: Images of AVATRINA executing some of the key tasks during the XPRIZE competition. [Best viewed in color.]

10.2 Team Results

With a total score of 14.5/15 points, team AVATRINA came in 4th place overall and was one of four teams to complete all 10 tasks. Fig. 24 shows AVATRINA completing some of the tasks on the course. Table 7 shows the results of all teams at the final competition in ranked order. Our system proved easy to learn in the competition, giving judges ample time during the training period (20+ minutes) to practice on the actual competition tasks.

On our first official run, the operator successfully completed all 10 tasks. The operator made use of both the Psyonic gripper and the parallel jaw gripper for completing different tasks depending on which gripper was best suited for each task, highlighting the benefit of our asymmetric design. During the final task of texture sensing, they also used the assistive texture mode to scan and feel the textures of the various rocks. Interestingly, after scanning, the operator opted to disable the assistance to perform the final grasp, suggesting that more work needs to be done to automatically detect when the operator intends to use different assistive features. The competition conditions provided an unreliable network connection during this run, so we disabled our facial rendering pipeline to minimize the system's required bandwidth. Despite this measure, the operator station still lost connection to AVATRINA partway through the run, which was resolved after a few minutes via an operator-requested manual restart of the operator station.

Table 7: Completion times and final scores from the XPRIZE finals competition as reported in [4]

Team	Score	Time (mm:ss)
NimbRo [58]	15.0	05:50
Pollen Robotics	15.0	10:50
Team Northeastern [35]	14.5	21:09
AVATRINA	**14.5**	**24:47**
i-Botics [69]	14.0	25:00
Team UNIST[47]	13.5	25:00
Inbiodroid	13.0	25:00
Team SNU [46]	12.5	25:00
AlterEgo [29]	12.5	25:00
Dragon Tree Labs	11.0	25:00
Avatar-Hubo [70]	9.5	25:00
Last Mile[16]	9.0	25:00

During our second run, our network connection was more stable, and our new operator completed tasks more aggressively, allowing them to reach

the start of Task 10 within approximately 12 minutes. While attempting this task, AVATRINA's arm collided with the environment with high force, causing it to engage mechanical brakes. Fig. 25 shows the operator's view after this happened. Since the arm could not be remotely recovered, this prevented the operator from further using AVATRINA's right arm. The operator tried to use the left arm to finish the task, but since there was no texture sensor on this arm, they were ultimately unsuccessful. Fig. 26 presents a breakdown of the completion times for each of the tasks on both days when compared to the competition winner, NimBro [58].

Further, we demonstrated that all 10 tasks could be completed without using arm force feedback mechanisms, indicating that lightweight interfaces can still be used to great effect for teleoperation. However, the emergency stop failure demonstrates the importance of designing remote resets into complex telerobotic system such as AVATRINA. Sensor-enabled collision avoidance strategies during driving and manipulation could have also avoided the failure.

Fig. 25: The operator's view after AVATRINA's right arm engaged mechanical brakes in response to excessive force. [Best viewed in color.]

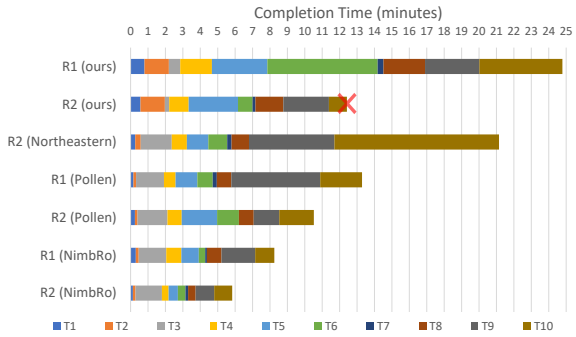


Fig. 26: Breakdown of task completion times on both days compared to the successful runs of other teams,NimbRo [58], Pollen Robotics and Team Northeastern [35], which placed first, second and third, respectively, as reported by [58]. T1-T10 denote tasks 1-10 and R1 and R2 denote the first and second runs, respectively. Our Run 2 timing for task 10 is stopped when the emergency stop is triggered by rough contact. Note how tasks 5 and 8 are considerably slower for our team due to the slow moving base, while manipulation times are competitive. [Best viewed in color.]

11 Discussion and Lessons Learned

In this section, we offer our ideas of what the important considerations for building a successful Avatar robot are, lessons learned from the competition, and future directions.

11.1 One-to-one Correspondence

Multiple other teams have argued that making the mapping from operator to robot as simple [35] and as close to an identity transform as possible [58] maximizes operability and makes the system easy to learn. AVATRINA departs from this philosophy slightly by providing a user-configurable offset between the operator's head and AVATRINA's head, and by using a clutch-based, relative pose tracking system to control the arms. Both of these decisions were mainly motivated by the embodiment gap between AVATRINA and the operator: because the capabilities of AVATRINA do not exactly match those of the operator, AVATRINA often must complete tasks somewhat differently than a person would with their own body. For example, since the cameras

and HMD used have a smaller field of view than a person does, the operator has to make more extreme movements with their head than they would in real life to gather the same amount of information or to achieve a satisfactory viewpoint to complete a task, which can be uncomfortable. The user-configurable offset between the operator's head and AVATRINA's head can alleviate some of this discomfort. A similar pattern is observed for AVATRINA's arms, especially when using the parallel jaw gripper which functions much differently from a hand.

Despite these differences, we observed that novice operators were able to quickly learn how to use the system, which suggests that one-to-one correspondence might not always strictly necessary for task proficiency. This suggests an opportunity to improve operator ergonomics without sacrificing avatar capability, which must be considered more seriously if extended operation is required, or if the robot must complete a task in a manner that would be uncomfortable for a person.

11.2 Force Feedback

Teams at the competition adopted two different philosophies for force feedback to the operator. While many teams provided force feedback through exoskeletons, our team was the most successful system that adopted commodity VR hardware that does not offer force feedback. This suggests that while the lack of force feedback reduces the telepresence of the operator, novice operators are still able to complete complex manipulation tasks quickly. Since the comparatively low cost of these user interfaces could provide broader access to avatar technology, this presents motivation to further improve operator capabilities using accessible interfaces.

11.3 Autonomous Functions

The system we deployed at the XPRIZE finals incorporated some autonomous functions, such as automatically returning the arms to the home configuration when the operator clicks a button. However, this was a small subset of functions we created during development, and we limited these functions to the most helpful subset to avoid overwhelming the operator and to improve training speeds. The tasks chosen by XPRIZE were sufficiently straightforward that autonomous assistance was not particularly helpful in many cases. For tasks that require more precision, such as handling hazardous or fragile materials, shared control may be more useful. Also, coordinated tasks like bimanual manipulation may also see more of a benefit from shared control. Moreover, the XPRIZE provided access to a relatively high bandwidth, low latency network, whereas less stable networks may render direct teleoperation difficult or even impossible. Autonomous and semi-autonomous functions may also be preferable for situations with unstable networks.

11.4 Robustness

System robustness in teleoperated systems is crucial, and although we performed significant amounts of testing and refinement of our system in the lab, human operators can trigger many unexpected conditions. Our system failure during the 10th task on Day 2 was due to an unrecoverable mechanical lock of the arm. Had the Franka Emika Panda arms provided an API to recover from such a state or had we developed more autonomous recovery protocols [58], the operator could have likely resumed the task.

11.5 Testing

In preparation for the XPRIZE competition, our team regularly tested our system under simulated competition settings. This allowed us to find and address false assumptions we had made about our system and the tasks, and gave us many opportunities to show our system to new operators, who often gave useful suggestions for new features or improvements. Frequent testing also helped to make set up of the system and training of the operator routine so that both could be done quickly and efficiently at the competition.

12 Conclusion

In this paper, we presented an immersive, novice-friendly avatar system with human-like manipulation, communication, and sensing capabilities. Our system adopts exoskeleton-free operator hardware with commodity VR equipment and was the most successful exoskeleton-free avatar at the ANA Avatar XPRIZE competition. We believe this is essential for deploying Avatar robots in the

world in the future where people can have easy access to the robot anywhere there is Internet connection. This paper also conducts human subject studies to investigate the effect of interpupillary distance and head mobility on operator hand-eye coordination. Finally, we offer our insights on how to build a successful immersive avatar system and lessons learned from the ANA Avatar XPRIZE competition.

Acknowledgements

We would like to thank all the people who provided constructive feedback on the design of AVATRINA throughout the years. We would also like to thank Siqi Lai for helping with figure editing. Finally, we thank Jack Yu, Vicky Ma, and Zoey Spengler for helping with AVATRINA's aesthetic design.

Declarations

Funding

This work was partially supported by the National Science Foundation under Grant #2025782.

Conflict of Interest

The authors have no relevant financial or non-financial interests to disclose.

Ethics Approval

All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki Declaration and its later amendments or comparable ethical standards. The study was approved by the University of Illinois at Urbana-Champaign Institutional Review Board (#24169). Informed consent was obtained from all individual participants included in the study.

Data Availablity

The televisualization dataset generated in the current study is not publicly available to protect the privacy of the subjects but is available from any corresponding author on reasonable request.

References

[1] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd, "A rewriting system for convex optimization problems," *Journal of Control and Decision*, vol. 5, no. 1, pp. 42–60, 2018.

[2] M. Akinc, K. E. Bekris, B. Y. Chen, A. M. Ladd, E. Plaku, and L. E. Kavraki, "Probabilistic Roadmaps of Trees for Parallel Computation of Multiple Query Roadmaps," in *Robotics Research. The Eleventh International Symposium*, B. Siciliano, O. Khatib, P. Dario, and R. Chatila, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, vol. 15, pp. 80–89. [Online]. Available: http://link.springer.com/10.1007/11008941_9

[3] J. Avalos and O. E. Ramos, "Real-time teleoperation with the baxter robot and the kinect sensor," in *2017 IEEE 3rd Colombian Conference on Automatic Control (CCAC)*, 2017, pp. 1–4.

[4] S. Behnke, J. A. Adams, and D. Locke, "The $10 Million ANA Avatar XPRIZE Competition Advanced Immersive Telepresence Systems," 2023.

[5] P. Bickel, P. Diggle, S. Fienberg, U. Gather, I. Olkin, and S. Zeger, "Springer series in statistics," 2009.

[6] S. A. Bowyer, B. L. Davies, and F. Rodriguez y Baena, "Active Constraints/Virtual Fixtures: A Survey," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 138–157, Feb. 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6634270/

[7] J. E. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Systems journal*, vol. 4, no. 1, pp. 25–30, 1965.

[8] G. Bruder, F. Steinicke, and A. Nüchter, "Poster: Immersive point cloud virtual environments," in *2014 IEEE Symposium on 3D User Interfaces (3DUI)*, 2014, pp. 161–162.

[9] J. O. Burns, D. A. Kring, J. B. Hopkins, S. Norris, T. J. W. Lazio, and J. Kasper, "A lunar L2-Farside exploration and science mission concept with the Orion Multi-Purpose Crew Vehicle and a teleoperated lander/rover," *Advances in Space Research*, vol. 52, no. 2, pp. 306–320, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0273117712006990

[10] L. Carlone and C. Pinciroli, "Robot Co-design: Beyond the Monotone Case," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 3024–3030.

[11] Y. Chen, B. Zhang, J. Zhou, and K. Wang, "Real-time 3D unstructured environment reconstruction utilizing VR and Kinect-based immersive teleoperation for agricultural field robots," *Computers and Electronics in Agriculture*, vol. 175, p. 105579, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169920311479

[12] J. V. Draper, D. B. Kaber, and J. M. Usher, "Telepresence," *Human Factors*, vol. 40, no. 3, pp. 354–375, 1998. [Online]. Available: https://doi.org/10.1518/001872098779591386

[13] C. Frith, "Role of facial expressions in social interactions," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 364, no. 1535, pp. 3453–3458, 2009.

[14] C. C. Gordon, C. L. Blackwell, B. Bradtmiller, J. L. Parham, P. Barrientos, S. Paquette, B. D. Corner, J. Carson, J. Venezia, B. M. Rockwell, M. Mucher, and S. Kristensen, "2012 anthropometric survey of u.s. army personnel: Methods and summary statistics," 2014.

[15] J. T. Hansberger, C. Peng, S. L. Mathis, V. Areyur Shanthakumar, S. C. Meacham, L. Cao, and V. R. Blakely, "Dispelling the Gorilla Arm Syndrome: The Viability of Prolonged Gesture Interactions," in *Virtual, Augmented and Mixed Reality*, S. Lackey and J. Chen, Eds. Cham: Springer International Publishing, 2017, vol. 10280, pp. 505–520, series Title: Lecture Notes in Computer Science. [Online]. Available: https://link.springer.com/10.1007/978-3-319-57987-0_41

[16] M. Haruna, M. Ogino, S. Tagashira, M. Kashiwa, S. Morita, T. Koike-Akino, K. Imai, T. Zuho, M. Makita, and Y. Takahashi, "Avatar Technologies of Team LAST MILE Toward Mobile Smart Device Operation Service," in *2nd Workshop Towards Robot Avatars, ICRA 2023*, London, 2023. [Online]. Available: https://www.ais.uni-bonn.de/ICRA2023AvatarWS/contributions/ICRA_2023_Avatar_WS_Haruna.pdf

[17] K. Hauser, E. Watson, J. Bae, J. Bankston, S. Behnke, B. Borgia, M. Catalano, S. Dafarra, J. van Erp, T. Ferris, J. Fishel, G. Hoffman, S. Ivaldi, F. Kanehiro, A. Kheddar, G. Lannuzel, J. Morie, P. Naughton, S. NGuyen, P. Oh, T. Padir, J. Pippine, J. Park, D. Pucci, J. Vaz, P. Whitney, P. Wu, and D. Locke, "Analysis and perspectives on the ana avatar xprize competition," *Intl. Journal of Social Robotics*, 2023 (submitted).

[18] P. B. Hibbard, L. C. van Dam, and P. Scarfe, "The Implications of Interpupillary Distance Variability for Virtual Reality," in *2020 International Conference on 3D Immersion (IC3D)*, Dec. 2020, pp. 1–7, 0.5.

[19] P. K. Jamwal, S. Xie, and K. C. Aw, "Kinematic design optimization of a parallel ankle rehabilitation robot using modified genetic algorithm," *Robotics and Autonomous Systems*, vol. 57, no. 10, pp. 1018–1027, 2009, 5th International Conference on Computational Intelligence, Robotics and Autonomous Systems (5th CIRAS). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889009001080

[20] M. Johnson, B. Shrewsbury, S. Bertrand, T. Wu, D. Duran, M. Floyd, P. Abeles, D. Stephen, N. Mertins, A. Lesman, J. Carff, W. Rifenburgh, P. Kaveti, W. Straatman, J. Smith, M. Griffioen, B. Layton, T. de Boer, T. Koolen, P. Neuhaus, and J. Pratt, "Team IHMC's Lessons Learned from the DARPA Robotics Challenge Trials," *Journal of Field Robotics*, vol. 32, no. 2, pp. 192–208, 2015. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21571

[21] S. Karumanchi, K. Edelberg, I. Baldwin, J. Nash, J. Reid, C. Bergh, J. Leichty, K. Carpenter, M. Shekels, M. Gildner, D. Newill-Smith, J. Carlton, J. Koehler, T. Dobreva, M. Frost, P. Hebert, J. Borders, J. Ma, B. Douillard, P. Backes, B. Kennedy, B. Satzinger, C. Lau, K. Byl, K. Shankar, and J. Burdick, "Team RoboSimian: Semi-autonomous Mobile Manipulation at the 2015 DARPA Robotics Challenge Finals," *Journal of Field Robotics*, vol. 34, no. 2, pp. 305–332, 3 2017. [Online]. Available: http://doi.wiley.com/10.1002/rob.21676

[22] C. C. Kemp, A. Edsinger, H. M. Clever, and B. Matulevich, "The Design of Stretch: A Compact, Lightweight Mobile Manipulator for Indoor Human Environments," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 3150–3157.

[23] T. Klamt, D. Rodriguez, M. Schwarz, C. Lenz, D. Pavlichenko, D. Droeschel, and S. Behnke, "Supervised Autonomous Locomotion and Manipulation for Disaster Response with a Centaur-Like Robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1–8.

[24] T. Klamt, M. Schwarz, C. Lenz, L. Baccelliere, D. Buongiorno, T. Cichon, A. DiGuardo, D. Droeschel, M. Gabardi, M. Kamedula, N. Kashiri, A. Laurenzi, D. Leonardis, L. Muratore, D. Pavlichenko, A. S. Periyasamy, D. Rodriguez, M. Solazzi, A. Frisoli, M. Gustmann, J. Roßmann, U. Süss, N. G. Tsagarakis, and S. Behnke, "Remote mobile manipulation with the centauro robot: Full-body telepresence and autonomous operator assistance," *Journal of Field Robotics*, vol. 37, no. 5, pp. 889–919, 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21895

[25] T. Kot and P. Novák, "Application of virtual reality in teleoperation of the military mobile robotic system TAROS," *International journal of advanced robotic systems*, vol. 15, no. 1, p. 1729881417751545, 2018.

[26] Kris Hauser, "Klampt python api," 2023, [Online; accessed 11-August-2023]. [Online]. Available: http://motion.cs.illinois.edu/software/klampt/latest/pyklampt_docs/

[27] E. Krotkov, D. Hackett, L. Jackel, M. Perschbacher, J. Pippine, J. Strauss, G. Pratt, and C. Orlowski, "The DARPA Robotics Challenge Finals: Results and Perspectives," *Journal of Field Robotics*, vol. 34, no. 2, pp. 229–240, 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21683

[28] A. Leeper, K. Hsiao, M. Ciocarlie, I. Sucan, and K. Salisbury, "Methods for collision-free arm teleoperation in clutter using constraints from 3d sensor data," in *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2013, pp. 520–527.

[29] G. Lentini, A. Settimi, D. Caporale, M. Garabini, G. Grioli, L. Pallottino, M. G. Catalano, and A. Bicchi, "Alter-ego: a mobile robot with a functionally anthropomorphic upper body designed for physical interaction," *IEEE Robotics & Automation Magazine*, vol. 26, no. 4, pp. 94–107, 2019.

[30] C. Lenz and S. Behnke, "Bimanual telemanipulation with force and haptic feedback trough an anthropomorphic avatar system," *Robotics and Autonomous Systems*, vol. 161, p. 104338, 12 2022.

[31] K.-Y. Liang and S. L. Zeger, "Longitudinal data analysis using generalized linear models," *Biometrika*, vol. 73, no. 1, pp. 13–22, 1986. [Online]. Available: https://academic.oup.com/biomet/article-lookup/doi/10.1093/biomet/73.1.13

[32] N. Y. Lii, D. Leidner, P. Birkenkampf, B. Pleintinger, R. Bayer, and T. Krueger, "Toward scalable intuitive telecommand of robots for space deployment with METERON SUPVIS Justin," in *The 14th Symposium on Advanced Space Technologies for Robotics and Automation (ASTRA)*. European Space Agency, 6 2017. [Online]. Available: https://elib.dlr.de/113125/

[33] J. Lim, I. Lee, I. Shim, H. Jung, H. M. Joe, H. Bae, O. Sim, J. Oh, T. Jung, S. Shin, K. Joo, M. Kim, K. Lee, Y. Bok, D.-G. Choi, B. Cho, S. Kim, J. Heo, I. Kim, J. Lee, I. S. Kwon, and J.-H. Oh, "Robot System of DRC-HUBO+ and Control Strategy of Team KAIST in DARPA Robotics Challenge Finals," *Journal of Field Robotics*, vol. 34, no. 4, pp. 802–829, 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21673

[34] T.-C. Lin, A. U. Krishnan, and Z. Li, "Physical Fatigue Analysis of Assistive Robot Teleoperation via Whole-body Motion Mapping," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Macau, China: IEEE, Nov. 2019, pp. 2240–2245. [Online]. Available: https://ieeexplore.ieee.org/document/8968544/

[35] R. Luo, C. Wang, C. Keil, D. Nguyen, H. Mayne, S. Alt, E. Schwarm, E. Mendoza, T. Padır, and J. P. Whitney, "Team Northeastern's Approach to ANA XPRIZE Avatar Final Testing: A Holistic Approach to Telepresence and Lessons Learned," *arXiv preprint arXiv:2303.04932*, 3 2023. [Online]. Available: http://arxiv.org/abs/2303.04932

[36] Y. Luo, J. Wang, H.-N. Liang, S. Luo, and E. G. Lim, "Monoscopic vs. Stereoscopic Views and Display Types in the Teleoperation of Unmanned Ground Vehicles for Object Avoidance," in *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*, 2021, pp. 418–425.

[37] G. Marani, J. Kim, J. Yuh, and W. K. Chung, "A real-time approach for singularity avoidance in resolved motion rate control of robotic manipulators," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1973–1978, 2002.

[38] J. M. C. Marques, P. Naughton, Y. Zhu, N. Malhotra, and K. Hauser, "Commodity Telepresence with the AvaTRINA Nursebot in the ANA Avatar XPRIZE Semifinals," in *RSS 2022 Workshop on "Towards Robot*

Avatars: Perspectives on the ANA Avatar XPRIZE Competition, 2022.

[39] H. Martins, I. Oakley, and R. Ventura, "Design and evaluation of a head-mounted display for immersive 3D teleoperation of field robots," *Robotica*, vol. 33, no. 10, p. 2166–2185, 2015.

[40] S. Mehrdad, F. Liu, M. T. Pham, A. Lelevé, and S. F. Atashzar, "Review of Advanced Medical Telerobots," *Applied Sciences*, vol. 11, no. 1, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/1/209

[41] G. G. Muscolo, S. Marcheschi, M. Fontana, and M. Bergamasco, "Dynamics Modeling of Human–Machine Control Interface for Underwater Teleoperation," *Robotica*, vol. 39, no. 4, p. 618–632, 2021.

[42] P. Naughton and K. Hauser, "Structured Action Prediction for Teleoperation in Open Worlds," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3099–3105, Apr. 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9691823/

[43] V. Nguyen, "Increasing Independence with Stretch: A Mobile Robot Enabling Functional Performance in Daily Activities," 2023.

[44] J. Orlosky, K. Theofilis, K. Kiyokawa, and Y. Nagai, "Effects of Throughput Delay on Perception of Robot Teleoperation and Head Control Precision in Remote Monitoring Tasks," *Presence: Teleoperators and Virtual Environments*, vol. 27, no. 2, pp. 226–241, 02 2018. [Online]. Available: https://doi.org/10.1162/pres_a_00328

[45] S. Orts-Escolano, C. Rhemann, S. Fanello, W. Chang, A. Kowdle, Y. Degtyarev, D. Kim, P. L. Davidson, S. Khamis, M. Dou, V. Tankovich, C. Loop, Q. Cai, P. A. Chou, S. Mennicken, J. Valentin, V. Pradeep, S. Wang, S. B. Kang, P. Kohli, Y. Lutchyn, C. Keskin, and S. Izadi, "Holoportation: Virtual 3D Teleportation in Real-Time," in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, ser. UIST '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 741–754. [Online]. Available: https://doi.org/10.1145/2984511.2984517

[46] B. Park, J. Jung, J. Sim, S. Kim, J. Ahn, D. Lim, D. Kim, M. Kim, S. Park, E. Sung *et al.*, "Team snu's avatar system for teleoperation using humanoid robot: Ana avatar xprize competition," in *RSS 2022 Workshop on "Towards Robot Avatars: Perspectives on the ANA Avatar XPRIZE Competition*, 2022.

[47] S. Park, J. Kim, H. Lee, M. Jo, D. Gong, D. Ju, S. Kim, D. Won, and J. Bae, "Team UNIST at the $10M ANA Avatar XPRIZE: Core Technologies, Integration, and Evaluation," in *2nd Workshop Towards Robot Avatars, ICRA 2023*, London, 2023. [Online]. Available: https://www.ais.uni-bonn.de/ICRA2023AvatarWS/contributions/ICRA_2023_Avatar_WS_SPark.pdf

[48] B. Pätzold, A. Rochow, M. Schreiber, R. Memmesheimer, C. Lenz, M. Schwarz, and S. Behnke, "Audio-based Roughness Sensing and Tactile Feedback for Haptic Perception in Telepresence," *arXiv preprint arXiv:2303.07186v1*, 3 2023. [Online]. Available: https://arxiv.org/abs/2303.07186v1

[49] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[50] L. Qian, A. Deguet, and P. Kazanzides, "ARssist: augmented reality on a head-mounted display for the first assistant in robotic surgery," *Healthcare Technology Letters*, vol. 5, no. 5, pp. 194–200, 2018. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/htl.2018.5065

[51] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, and others, "ROS: an open-source Robot Operating System," in *ICRA workshop on open*

source software, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[52] R. S. Renner, E. Steindecker, M. MüLler, B. M. Velichkovsky, R. Stelzer, S. Pannasch, and J. R. Helmert, "The Influence of the Stereo Base on Blind and Sighted Reaches in a Virtual Environment," *ACM Transactions on Applied Perception*, vol. 12, no. 2, pp. 1–18, Apr. 2015. [Online]. Available: https://dl.acm.org/doi/10.1145/2724716

[53] A. Rochow, M. Schwarz, M. Schreiber, and S. Behnke, "VR Facial Animation for Immersive Telepresence Avatars," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Kyoto, Japan, 2022, pp. 2167–2174.

[54] B. Rosner and A. Munoz, "Autoregressive modelling for the analysis of longitudinal data with unequally spaced examinations," *Statistics in Medicine*, vol. 7, no. 1-2, pp. 59–71, Jan. 1988. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/sim.4780070110

[55] S. Sarantakos, *Social research*. Bloomsbury Publishing, 2017.

[56] M. Schwarz and S. Behnke, "Low-Latency Immersive 6D Televisualization with Spherical Rendering," in *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, 2021, pp. 320–325. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9555797/

[57] M. Schwarz, M. Beul, D. Droeschel, T. Klamt, C. Lenz, D. Pavlichenko, T. Rodehutskors, M. Schreiber, N. Araslanov, I. Ivanov, J. Razlaw, S. Schüller, D. Schwarz, A. Topalidou-Kyniazopoulou, and S. Behnke, "DRC Team NimbRo Rescue: Perception and Control for Centaur-Like Mobile Manipulation Robot Momaro," in *The DARPA Robotics Challenge Finals: Humanoid Robots To The Rescue*, M. Spenko, S. Buerger, and K. Iagnemma, Eds. Cham: Springer International Publishing, 2018, pp. 145–190. [Online]. Available: https://doi.org/10.1007/978-3-319-74666-1_5

[58] M. Schwarz, C. Lenz, R. Memmesheimer, B. Pätzold, A. Rochow, M. Schreiber, and S. Behnke, "Robust Immersive Telepresence and Mobile Telemanipulation: NimbRo wins ANA Avatar XPRIZE Finals," *arXiv preprint arXiv:2303.03297v1*, 3 2023. [Online]. Available: http://arxiv.org/abs/2303.03297

[59] S. S. Shapiro and M. B. Wilk, "An Analysis of Variance Test for Normality (Complete Samples)," *Biometrika*, vol. 52, no. 3/4, pp. 591–611, 1965. [Online]. Available: https://www.jstor.org/stable/2333709

[60] S. Shin and S. Choi, "Geometry-based haptic texture modeling and rendering using photometric stereo," in *2018 IEEE Haptics Symposium (HAPTICS)*, 2018, pp. 262–269.

[61] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, "First Order Motion Model for Image Animation," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d\textquotesingle Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/31c0b36aef265d9221af80872ceb62f9-Paper.pdf

[62] P. Stotko, S. Krumpen, M. Schwarz, C. Lenz, S. Behnke, R. Klein, and M. Weinmann, "A VR System for Immersive Teleoperation and Live Exploration with a Mobile Robot," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 11 2019, pp. 3630–3637.

[63] G. Sánchez and J.-C. Latombe, "A Single-Query Bi-Directional Probabilistic Roadmap Planner with Lazy Collision Checking," in *Robotics Research*, B. Siciliano, O. Khatib, F. Groen, R. A. Jarvis, and A. Zelinsky, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, vol. 6, pp. 403–417. [Online]. Available: http://link.springer.com/10.1007/3-540-36460-9_27

[64] S. Tachi, K. Minamizawa, M. Furukawa, and C. L. Fernando, "Telexistence — from 1980 to

2012," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 10 2012, pp. 5440–5441. [Online]. Available: https://ieeexplore.ieee.org/document/6386296

[65] S. Tachi, Y. Inoue, and F. Kato, "TELESAR VI: Telexistence Surrogate Anthropomorphic Robot VI," *International Journal of Humanoid Robotics*, vol. 17, no. 05, p. 2050019, 2020. [Online]. Available: https://doi.org/10.1142/S021984362050019X

[66] A. Takagi, S. Yamazaki, Y. Saito, and N. Taniguchi, "Development of a stereo video see-through HMD for AR systems," in *Proceedings IEEE and ACM International Symposium on Augmented Reality (ISAR 2000)*. Munich, Germany: IEEE, 2000, pp. 68–77. [Online]. Available: http://ieeexplore.ieee.org/document/880925/

[67] K. Takeuchi, Y. Yamazaki, and K. Yoshifuji, "Avatar Work: Telework for Disabled People Unable to Go Outside by Using Avatar Robots," in *Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 53–60. [Online]. Available: https://doi.org/10.1145/3371382.3380737

[68] Toyota, "Toyota unveils third generation humanoid robot t-hr3," 2017. [Online]. Available: https://global.toyota/en/detail/19666346

[69] J. B. Van Erp, C. Sallaberry, C. Brekelmans, D. Dresscher, F. Ter Haar, G. Englebienne, J. Van Bruggen, J. De Greeff, L. F. S. Pereira, A. Toet *et al.*, "What comes after telepresence? embodiment, social presence and transporting one's functional and social self," in *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2022, pp. 2067–2072.

[70] J. C. Vaz, A. Dave, N. Kassai, N. Kosanovic, and P. Y. Oh, "Immersive auditory-visual real-time avatar system of ana avatar xprize finalist avatar-hubo," in *2022 IEEE International Conference on Advanced Robotics and Its Social Impacts (ARSO)*. IEEE, 2022, pp. 1–6.

[71] A. Wang, J. Ramos, J. Mayo, W. Ubellacker, J. Cheung, and S. Kim, "The hermes humanoid system: A platform for full-body teleoperation with balance feedback," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 2015, pp. 730–737.

[72] B. G. Witmer and M. J. Singer, "Measuring Presence in Virtual Environments: A Presence Questionnaire," *Presence: Teleoperators and Virtual Environments*, vol. 7, no. 3, pp. 225–240, Jun. 1998, 2. [Online]. Available: https://direct.mit.edu/pvar/article/7/3/225-240/92643

[73] C. Yang, J. Zhang, Y. Chen, Y. Dong, and Y. Zhang, "A review of exoskeleton-type systems and their key technologies," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 222, no. 8, pp. 1599–1612, 2008.

[74] Y. Zhu, A. Smith, and K. Hauser, "Automated Heart and Lung Auscultation in Robotic Physical Examinations," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4204–4211, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9707618

A Elbow Heuristic Details

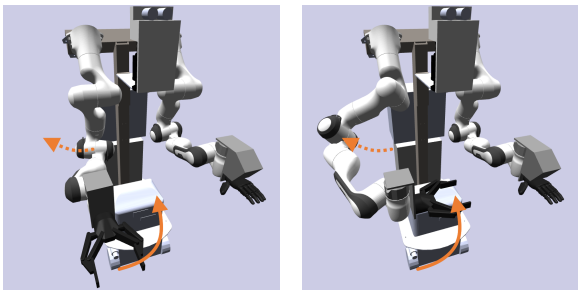


Fig. 27: Concept of the bias configuration heuristic. Based on the commanded motion of the end effector, the elbow pose is controlled to appear more human-like and reduce the rate of tracking failures due to joint limits and self-collision. [Best viewed in color.]

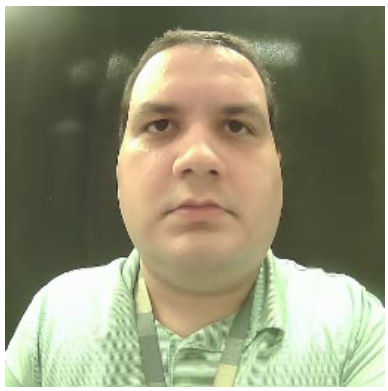
Let the 3D rotation and translation vectors $r = \begin{bmatrix} r_x & r_y & r_z \end{bmatrix}^T$ and $t = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^T$ represent the commanded $SE(3)$ transform T_{ee} of a hand. The shoulder angle heuristic for the left and right arms are given by:

$$\begin{aligned} q_{\text{shoulder, right}} &= -(\max(k_{\text{angle}}(r_y + r_z), 0) \\ &\quad + k_z(t_z - z_0)) \\ q_{\text{shoulder, left}} &= +(\max(k_{\text{angle}}(r_y - r_z), 0) \\ &\quad + k_z(t_z - z_0)) \end{aligned} \tag{15}$$

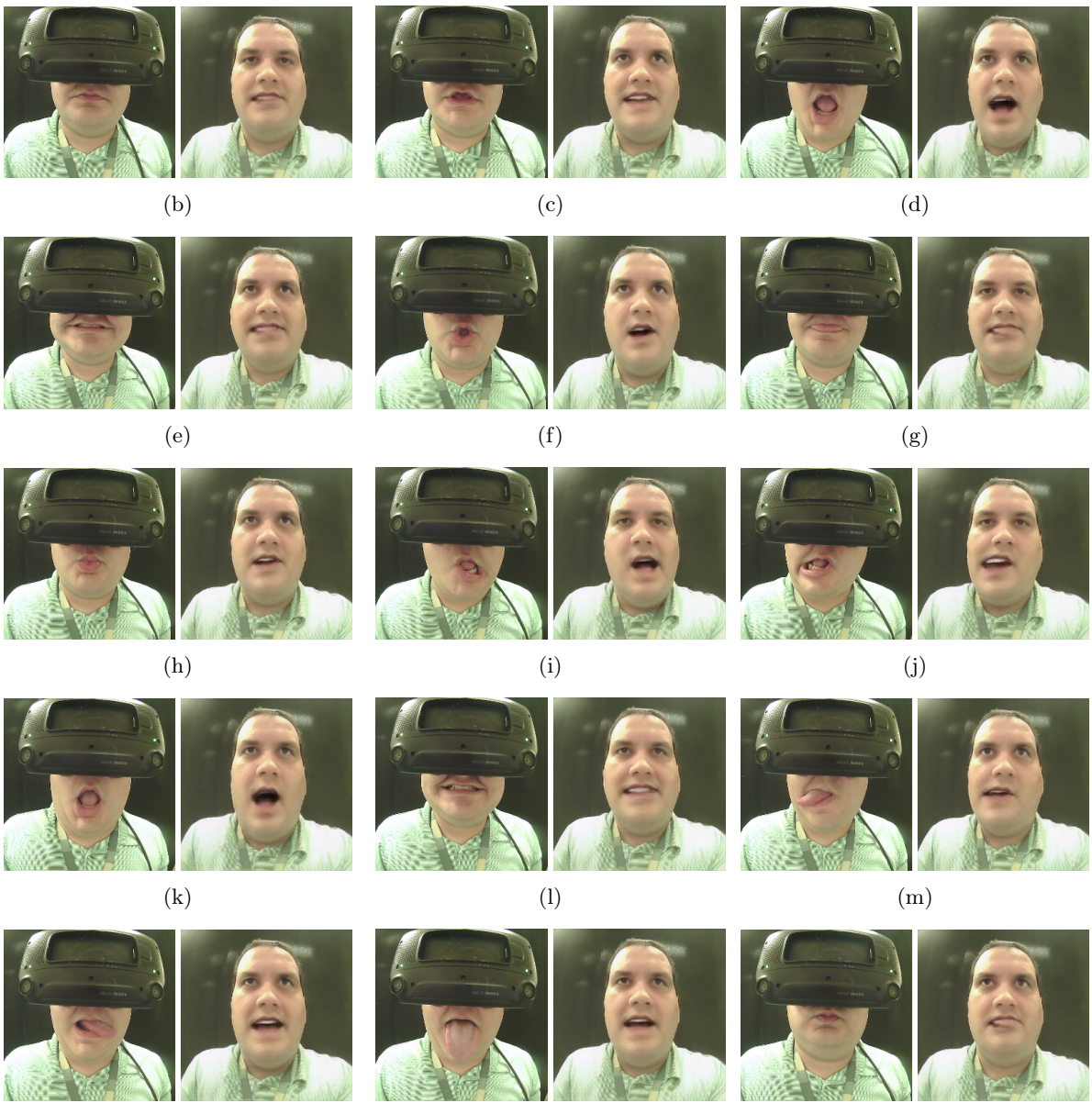
where k_{angle}, k_z, and z_0 are positive tunable constants. Fig. 27 shows the motion of the elbow when the operator turns their wrist inwards: positive z axis rotation corresponds to a negative q_{shoulder} for the right arm, which forces the elbow outwards. Turning the wrist downwards and lifting the hand upwards also force the elbow to turn outwards.

B Qualitative Evaluation of Facial Reconstruction Pipeline

Fig. 28 illustrates the performance of the proposed facial reconstruction pipeline. Examples b) through f) show that under relatively tame facial expressions common during communication, the network's output is mostly plausible and can convey some of the expressions of the operator to their remote counterparts, despite having trouble with conveying precise mouth movements or proper rendering of the operator's teeth (unseen in the source image). Examples f) through p) show some of the failure cases of this pipeline: Facial expressions far outside of its training distribution, such as i) and j), as well as any expressions that involve parts of the face not captured by facial landmarks (such as the tongue or cheeks), like m) through p) result in outputs that don't necessarily capture the operator's intent. Further, gaze direction is entirely dependent on the neutrally blinking video recording, which can sometimes result in awkward staring or stargazing, such as in example f).



(a) Source Image For Animation



(b) (c) (d)

(e) (f) (g)

(h) (i) (j)

(k) (l) (m)

(n) (o) (p)

Fig. 28: A sample of the performance of the facial reconstruction pipeline on one of the author's under varied facial expressions, with the original image on the left paired with the network's output on the right