Avatour: VR-mediated Robot Telepresence System for Exploration of the Real World

Rohan Gandhi CID: 01845920 | Kert Laansalu CID: 01954127 | Alexandra Neagu CID: 01843748 | Ishaan Reni CID: 01906148 | Jungbeom Nam CID: 01519254

Charmaine Cheuk Yin Louie CID: 01711892 | Aleera Ewan CID: 01857182 | Chris Myers CID: 01853460 | Harry Phillips CID: 01863197 | Athanasios Kantas CID: 01890458

***Abstract*—Robot teleoperation has revolutionized human engagement within hazardous environments by enabling remote completion of dangerous tasks. In parallel, virtual reality (VR) has emerged as a potential interface for enhancing such teleoperation experiences. This paper explores the integration of VR technology into remote robotic telepresence, envisioning a scenario where individuals can explore any real-world location, whether due to health constraints or financial limitations. We use a 360-degree camera mounted on the arm of a legged manipulator to enable the interactive exploration of a remote human-populated environment. We hypothesize that the physical robot component of the system with 360-degree live streaming will increase immersion by enabling interactivity and adaptability to different remote scenarios. By leveraging such technologies, users can embark on immersive journeys, experiencing the distant corners of the globe vicariously through robotic avatars.**

***Index Terms*—human-centred robotics, virtual reality, extended reality, telepresence, robot teleoperation, immersion**

I. Introduction

Virtual Reality (VR) has introduced many opportunities in the entertainment, industrial and education sectors by bridging the gap between people around the globe in a virtual environment or introducing immersive ways of teleoperating robots. As a result, the degree of immersion that users experience is highly important to the success of the task at hand.

Previous research has concentrated on investigating human-to-environment interactions, employing teleoperated robots for tasks such as environment mapping and object grasping. The goal of the project is to integrate insights from this research domain, thus enabling the development of VR-robot interfaces. This integration holds the potential to create new opportunities for individuals to explore the real physical world that is geographically distant.

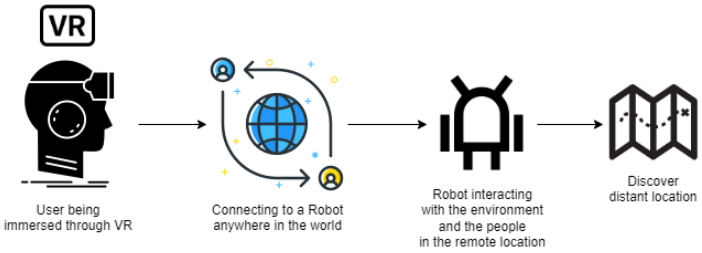

Fig. 1: Avatour idea design: Allowing users to explore remote locations through VR-teleoperation of a robot

Henceforth, Avatour, a VR-mediated robot telepresence system, is introduced to allow users to experience the real world through a VR headset. When using the system, users will connect to a physical robot located in a distant remote location to explore the real-world surroundings through unrestricted movement and interact with the environment through visual and auditory stimuli.

NOTE: All code for this project can be found on the public Github repositories under Avatour Organisation

II. Literature Review

A. Immersion in Virtual Reality

The topic of VR immersion has been actively researched, with relevant subjects such as “feeling of being there” [1] and “degree of interaction” [2], being debated. Immersion can be defined by a multitude of characteristics, both technical and psychological, and is the primary metric for measuring user satisfaction and connection felt with the virtual world.

Hudson et al. [2] present various uses of VR in enhancing tourism by connecting people to new experiences (e.g. underwater tours of aquariums). They describe immersion and social interaction as essential determinants of user satisfaction, where immersion is a compound of a “sense of harmony, escape and being in another world” [2]. The paper emphasises the role of subjective immersion as each user perceives the VR scene differently depending on their behavioural traits, sense of self or wants to escape their daily life. This research confirms that virtual experiences also strongly depend on the triple pillars of “things, surroundings and other people” [3].

B. Teleoperation of Robots through VR

Teleoperating robots bring reliable ways of remotely interacting with a separate environment in the real world. There are diverse examples of teleoperation, such as having rescuers explore disaster zones or any areas proving risks to human interventions [4], doctors hosting medical interventions remotely [5] or adults surveying their pets or children left unattended at home [6]. However, traditional teleoperation remains reliable on 2D perspectives and sensor data to display the environment that the robot has to progress through. With the improvements in Extended Reality (XR) methodologies, developers are now considering XR-mediated teleoperation.

There is current research in using XR as an innovative approach which allows for a 3D real-time perception of the environment surrounding the robot. Quesada et al. describe a novel approach of using the Microsoft Hololens 2 as the Augmented Reality (AR) user interface for controlling a robotic arm mounted on the *Boston Dynamics' Spot* [7]. The setup allows the user to interact with the environment surrounding the robot by grasping targeted objects using the robotic arm. Their results reflect the AR platform as a

highly reliable interface suitable for the teleoperation of grasping tasks. Additionally, Queseda et al. follow this up by presenting an in-depth overview of the design and evaluation of the teleoperation procedure in terms of immersion, time performance and cognitive workload [8].

Several commercial products exist and aim to build traction in VR robotic teleoperation. An example of existing teleoperation of robots through VR is TX Scara, developed by Telexistence. TX Scara aims to bring ultra-low latency VR streaming and hybrid intelligent control for reaching, grasping to pick & place of everyday objects [9].

C. VR-Mediated Robotic Telepresence

According to [10], telepresence is based on vividness and interactivity. Vividness is "the representational richness of a mediated environment" [10] and interactivity refers to "the extent to which users can participate in modifying form and content of a mediated environment in real time" [10]. There are many works studying telepresence in virtual environments or digital scans of the real world, such as exploring the use of VR in experiencing shopping inside a virtual mall [11], however, there is an apparent commercial scarcity in the realm of using VR interfaces to remotely interact with a distant real-world environment. Nevertheless, research studies by Sakashita et al. and Oh et al., highlight the viability of real-time VR robotic interaction for applications in tourism [12] and office work [13].

III. Motivation & Hypothesis

This study aims to investigate whether VR-mediated robot telepresence increases immersion in non-social environments compared to 2D screen-based interactions. The definition of immersion in this question will be similar to what is described in Section II-A.

The primary group of users would be immobilised (for example, bedridden), where their limited freedom of movement introduces complications in allowing them to travel and discover new geographically distant locations. With the system, users would wear a VR headset and connect to any unrestricted location in the world, allowing them to discover that location through the eyes of a quadruped robot. An example of a use-case for a non-social, human-(robot)-environment interaction would be a hospitalised user participating in a virtual tour adventure in a different country.

Through such use cases, Avatour is hypothesised to be an effective replacement for 2D screen-based interactions according to the following qualitative benchmarks:

- freedom of vision through 360-degree VR.
- freedom of movement via teleoperation
- feeling of motion through dynamic video streaming.
- feeling of presence through auditory and other stimuli.

These qualitative benchmarks can be implicitly measured from quantitative Human-Robot-Interaction metrics as discussed in Section V-B.

A. Similar Products

The hypothesis aims to compare 2D screen-based robotic interactions to 360-degree VR telepresence. Each of these pathways offers distinct experiences, capabilities, and applications, ranging from simple video streaming to immersive telepresence of complex environments. By decomposing these ideas, an exploration into the current state of 2D streaming and 360-degree streaming before discussing 2D telepresence and 360-degree VR telepresence is required, as it can support the intention behind the hypothesis.

2D Live Streaming Products: Since COVID-19, the market in 2D streaming services has been valued at 7.02 billion USD [14]. Major video streaming such as Microsoft Teams, Zoom or FaceTime have been used during and after the pandemic in various scenarios such as family gatherings, office meetings or university lectures. Many of these services have improved the lives of people during the pandemic by connecting people when not possible otherwise. However, after the pandemic, the use of video streaming services as a replacement for face-to-face human interaction in times of leisure or recreation is now less common.

3D Live Streaming Products: There is a range of 3D streaming products, such as the Insta360 Enterprise, which are increasingly being used in many domains including education, manufacturing and remote collaboration. The key aims of these streaming services are to bring high-quality but low-latency 360-degree video feed. This poses many challenges including preventing dizziness and nausea to optimising low latency feed when there is a lack of high-speed connectivity. Furthermore, though the 360-degree stream provides a greater sense of presence to the environment, there does exist a lack of interaction with the environment, especially in very social environments such as watching a sports match. In this example, a 360-degree VR streaming service cannot provide any interactivity with crowd members since there does not exist anything but a camera where the stream is broadcasted.

2D Telepresence: 2D telepresence products include the Double 3 robot by Double Robotics, the Ohmni® telepresence robot, and the Ava robot by AVA Robotics. These types of robots target the workspace environment and cannot adapt to various navigational settings such as climbing stairs or a diverse range of terrains outside an office.

3D Telepresence: As mentioned in Section II-C, there already exists VR-mediated robotic telepresence for specific scenarios such as pick & place of everyday objects, tourism or office interaction. To further research in this domain, this project aims to shed light on the social impact from the user's perspective by determining if immersion is increased through VR.

B. Evaluation of hypothesis

The evaluation of the system will be done through a controlled experiment with human users, where interaction data will be gathered and experiment opinions of all participants be posted.

The experiment will compare VR and web based interfaces for robot remote control and telepresence. The evaluation between the two interfaces will determine whether using a VR headset is more immersive for viewing a 360 video live streams than 2D displays. The 2D streaming app will display a 360 video as well and will allow for panning using a finger or gyroscope tilting. The app will also determine whether controlling the robot through a VR joystick or joystick component on an app will be more realistic or user-friendly.

The 360 video live streaming to the VR headset/tablet will be evaluated using latency as its primary metric of measurement. The lower the latency, the lower the time lag is between the 360

camera capturing the footage and the user viewing it through the VR headset/tablet. Therefore, a goal of moderately low latency will positively affect the immersion and feeling of "being there" for the participants.

IV. System Design Plan and Progress

A. Robot Morphology

Section III-A introduced some existent commercial robots that allow for novel 2D interfaces to be used for telepresence during meetings in an office setting. Such robots are limited by their navigational settings (wheels) and are only used in constrained areas. This is the main reason why using Boston Dynamics' Spot would introduce a greater degree of freedom in terms of manoeuvrability. Society is built for humans, which are bipedal beings, hence wheeled vehicles encounter vast challenges during movement, such as when encountering stairs or uneven ground. However, Spot is a quadruped robot capable of manoeuvring around such challenging environments. This is a crucial feature to provide users with a greater feeling of presence that is continuous across different environments. Moreover, Spot is a commercially available robot that is compatible with many add-on components such as a robotic arm, LIDAR sensors and 360-degree cameras. Therefore, Spot can be used for a variety of tasks and be specialised to a specific use case.

Figure 4 shows the final morphology of the Spot robot with the following components integrated on Spot:

- An NVIDIA Jetson Orin, with a 3D printed-protective enclosure and power supply, for onboard data processing and AI workloads.
- A Ricoh Theta camera for 360-degree video streaming.
- A DJI OSMO Gimble to stabilize the 360-degree camera.

Fig. 2: Avatour robot morphology

B. System Overview and Hardware

Figure 3 illustrates the Avatour's system architecture. The 360-degree live footage captured by the Ricoh Theta camera is transmitted to the NVIDIA Jetson Orin, which then relays it to a video streaming server. This server, in turn, broadcasts the feed through HTTP to both a tablet and the *Unity* application. Consequently, the 360-degree video stream is accessible for display on either a 2D screen via the tablet or within a 360 VR headset through the *Unity* app.

The user can send motion commands to the Spot robot through either the tablet or the VR joysticks. A Motion Tracking server receives and converts commands from either a VR joystick or tablet to appropriate ROS messages. These ROS messages are received by a 'joystick interface' node, which determines the resulting movement. It should be noted that the tablet and the VR headset are in a remote location, away from the Spot robot with the 360 camera and the Jetson processor attached to it. All key components are developed in their own Docker containers, following a microservice distribution approach, in order to ensure easy reproducibility of the software on different hardware.

C. Robot teleoperation

Both the tablet and the VR headset have two joysticks to control linear and angular velocities on Spot. Furthermore, there exist two modes of control on Spot - *standing* mode and *walking* mode. These modes are the same as the *standing* and *walking* modes provided with the default Spot controller. Implementing the teleoperation features identical to that of Spot's manufacturer was a sound and reasoned choice, considering the design had already undergone thorough design by industry specialists. In addition, these two modes were regarded to be sufficient for the motive of exploration since all poses can capture the entire surroundings of the user. Given that the robot can be teleoperated from large distances, a requirement for navigation was to maintain a sufficiently safe distance from obstacles, which also helps ease user navigation. Fortunately, Spot has in-built collision avoidance and the appropriate obstacle padding was calibrated for our purposes. Finally, a safety feature to stop control of Spot was implemented in case users required a break or desired to cease teleoperation. This control was set through a button on the tablet or a trigger on the VR joystick.

As mentioned, a motion server receives joystick commands from either the tablet or the VR headset via an HTTP request containing a JSON. This JSON is decoded into left and right velocity controls and a mode of operation. The control flag is also passed in this JSON to signal if the user has control of the robot. Once the JSON is decoded into its corresponding components, the data is restructured to appropriate ROS datatypes and published to the 'joystick interface' node. This node is not situated on Spot but on the Nvidia Jetson as it was decided that using Spot-CORE (Spot's onboard computer) was unnecessary and all commands can be sent via the Boston Dynamic's Spot SDK [15] through the Jetson's ethernet network interface. The 'joystick interface' node processes incoming commands from the appropriate topics and determines the required commands to send via the Boston Dynamic's Spot SDK [15], which initiates the intended movement. Care has also been taken to ensure that Spot returns to a safe resting state via e-stop services in the Spot SDK.

1) VR control: The VR controllers offer users a range of controls for operating Spot. These controls include the *control trigger* (right trigger), *mode switch* (left trigger), *movement control* (left trackpad/joystick), and *orientation control* (right trackpad/joystick).

The control trigger, activated by clicking the right trigger, establishes the user's control of Spot with the VR headset. Subsequent commands from the controllers are then directed to the motion server and relayed to Spot. In this context, a boolean

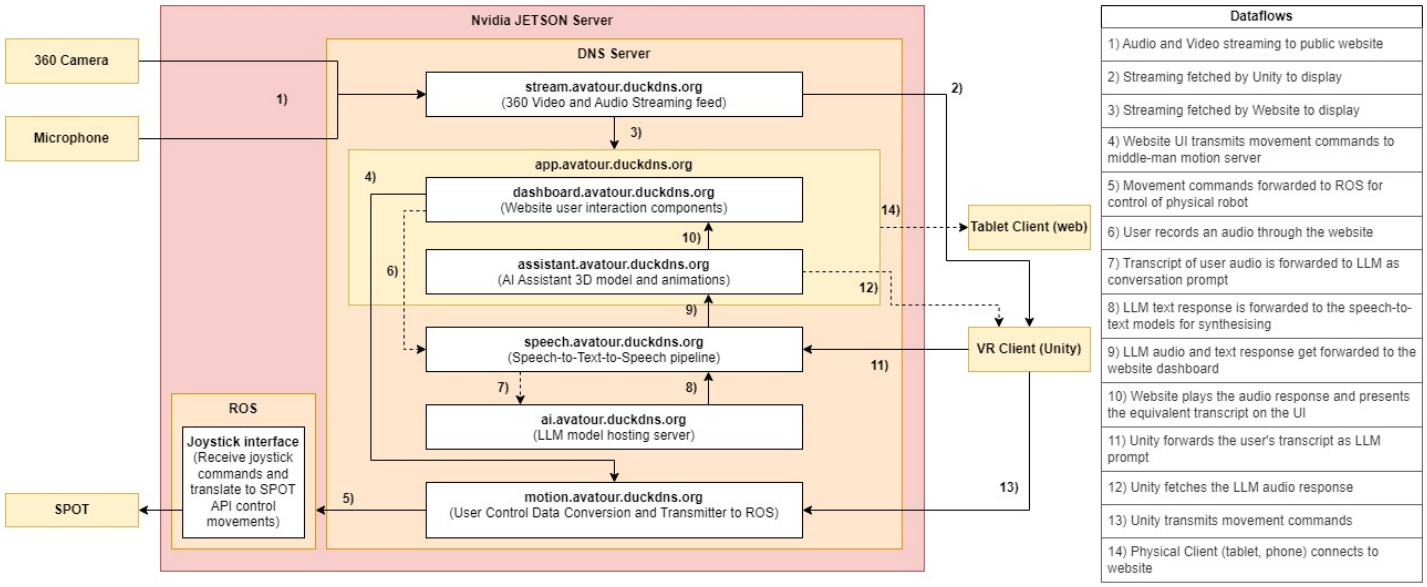

Fig. 3: System architecture of Avatour: presents the Environment-to-User data flow, such as video and audio streaming, and all the User-to-Robot interactions, such as movement commands through joysticks

value is included in the motion server's JSON to indicate whether the VR is in control (1) or not (0).

The mode switch, initiated by clicking the left trigger, toggles between standing mode (where Spot remains fixed but can lean) and walking mode, based on the user's preferences for Spot's movement. An integer value is integrated into the motion server's JSON to delineate Spot's mode: either standing mode (0) or walking mode (1), while considering the possibility of introducing additional modes in the future.

Movement control is facilitated by the left controller's trackpad/joystick, which intuitively maps to standard forward/backwards/left/right directions. Spot's orientation is managed by the right controller's trackpad/joystick, enabling users to turn Spot similarly to a yaw control. Touching either Vive Controller's trackpad translates to Cartesian X and Y coordinates, which are then incorporated into the motion server's JSON. Both trackpads can be utilized simultaneously to ensure smooth teleoperation of Spot.

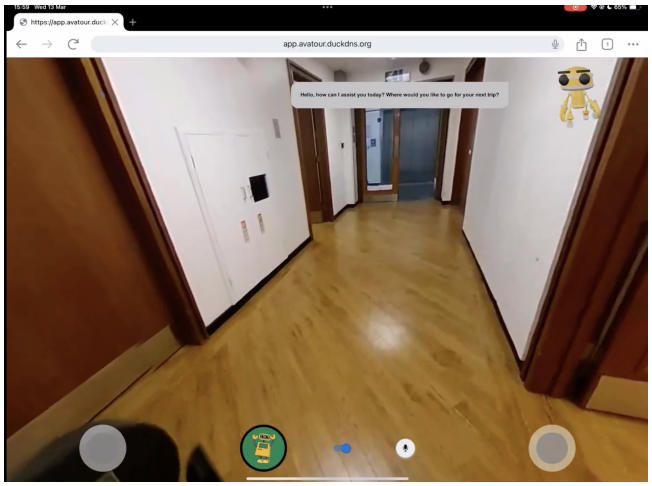

Fig. 4: Avatour Tablet Interface

2) Tablet control: The primary objective of the tablet is to establish an equivalent interface on an alternative display platform, with the intention of testing hypotheses related to immersive user experiences. This approach allows for a comparative analysis between the immersive qualities of VR control and the more traditional touch-based interaction offered by tablets. By maintaining a consistent set of functionalities, such as the control button for command initiation, switch for mode of movement and dual joysticks for controlling the movement and orientation of the robot, the tablet interface seeks to replicate the interactive capabilities of the VR controllers within a different sensory context.

Integrating a gyroscope within the tablet interface adds a critical dimension to the comparative study of immersive interfaces by enabling users to look around within the control environment by physically moving the tablet. The gyroscope functionality aims to mirror the head-tracking capabilities inherent in VR headsets, where the user's head movements are directly translated into navigational input within the virtual environment. This was implemented by hosting both the web app and the backend on the Jetson, thus originating from the same source and resolving CORS errors, and providing SSL certificates to run the URLs over HTTPS, resolving security issues preventing the web app from accessing the hardware of the device.

The switch for controlling whether the robot is active or not is a custom component designed in raw HTML and JavaScript. On the other hand, the switch and joysticks utilise pre-built React components. The map functionality uses the Google Maps API. A marker is placed on the map at the robot's location. The location is fetched every second, and the map is re-rendered on an updated location; this allows one to see their position move on the map as the robot moves. The map can be easily extended to facilitate navigation functionality.

The parallel structure of tablet and VR headset enables users to evaluate the extent to which immersion and intuitive control are influenced by the mode of interaction, be it through the spatial engagement of VR or the tactile feedback combined with gyroscope-enabled visual interactivity from a tablet.

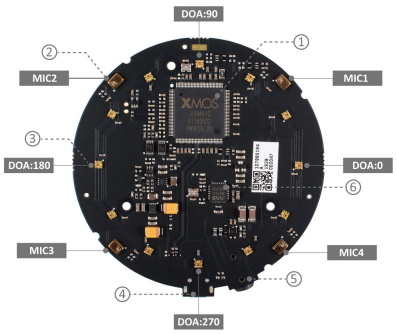

Fig. 5: Microphone Array Layout

D. Environmental Audio Recording

The audio input is captured through the Respeaker USB Mic Array, which hosts 4 microphones to enable sound to be captured from all directions. Due to the operating noise from Spot itself, our main objective with the microphone was to suppress the noise and allow other environmental and speech sounds to be delivered to our VR user. This was done by first detecting the primary Direction of Arrival (DOA) of the incoming sound, which is accessed from the tuning parameters of the microphone. This DOA is then used as input with the unit of degrees, ranging from 0 to 360 corresponding to the angles on the microphone. Figure 5 depicts the location of the microphone array system as well as the baseline DOA settings for processing.

This DOA value is then taken as the input for the sound processing function, which calibrates the DOA to our desired angle of 45 to 135 degrees. This was chosen due to the microphone being mounted in an upright position on top of Spot, with the excluded angles pointing towards Spot. The incoming sound from these excluded DOA values is first directly suppressed by multiplying the data stream by 0.3, followed by adjusting and experimenting with the parameters of the microphone. These parameters include a high-pass filter to reduce low frequency noise, the over-subtraction factor [16] for noise, as well as gain-floor [17] of the noise suppression process. These adjustments resulted in a successful filter that suppresses the whirring noises from Spot while allowing environmental sounds and human speech to pass through to the live stream feed.

E. Video & Audio Streaming

1) Streaming Process: The streaming is performed using a custom GStreamer installation and pipeline, based on NVIDIA's Deepstream SDK.

The GStreamer installation was built as a container image using the following steps:

- Starting with NVIDIA's Deepstream container image
- Uninstalling GStreamer (as the installed version is too old) and building GStreamer 1.22 from source
- Building a patched version of `libuvc` that supports the Ricoh Theta Z1 camera
- Building a custom GStreamer plugin that can use the patched `libuvc`

The GStreamer pipeline is then set up to perform the following:

- Read live audio from the USB Microphone
- Read live video from the Ricoh Theta Z1 camera

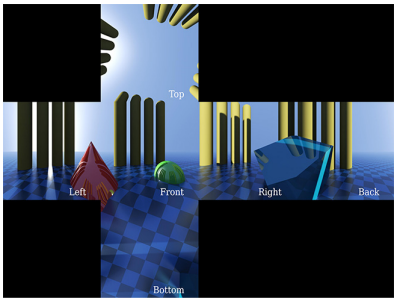

Fig. 6: Panoramic Skybox Structure

- Transcode the video to 10Mbps H.264 using Nvidia Deepstream
- Stream the combined audio and video over RTSP (Real-Time Streaming Protocol)

Using NVIDIA's Deepstream SDK allowed us to use the Jetson's hardware video encode and decode accelerators to transcode the video stream to a lower bitrate with very low latency and CPU usage. This allowed the video to be streamed over lower-bandwidth network connections, or to multiple clients.

We additionally configured the MediaMTX server software as a media router to translate between RTSP and WebRTC, and to allow the stream to be sent to multiple clients. The GStreamer pipeline sends to MediaMTX continuously, and when a client connects to MediaMTX, the stream is forwarded to the client.

2) VR/Unity: The VR system utilises an HTC Vive Pro with Unity to seamlessly integrate the 360 video camera feed into the VR headset. In Unity, a dedicated project was crafted to develop a scene, later transformed into an application for the Jetson platform. First, a project was designed to project an on-disk 360 video onto the panoramic Unity Skybox [18], which is designed to project an image or feed onto the surface of a deconstructed cube wrapped around the viewpoint of the person using the headset, as displayed in Figure 6.

To do this, a Unity Video Player component was used, which plays a feed from a video file. Then, the Video Player component was replaced with an empty component running a script to capture feed from a webcam and project this feed onto the Skybox. The Ricoh Theta 360 camera was not recognised as a webcam by the devices used. As an alternative, OBS Studio [19] was used to play the camera feed via an OBS Virtual Camera. This virtual camera was detected by Unity's WebCamTexture, which is a texture onto which live video input is rendered [20]. After this, an initial test for streaming was performed by streaming the camera feed via YouTube Live, but the latency was too large. Moving forward with the recognition that it is infeasible to use YouTube Live to stream the camera footage, GStreamer [21] was used to send the video feed from the camera to a server on the Jetson via RTSP [22]. It is anticipated that a server that can convert this feed to WebRTC [23] will be used. As with the webcam before, this feed is projected onto the Unity panoramic SkyBox texture, allowing an immersive 360-degree experience. The current streaming pipeline is presented in Figure 7.

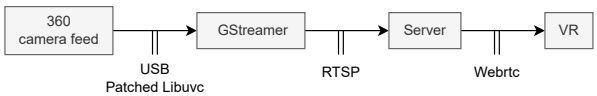

Fig. 7: Streaming Pipeline

Additionally, the HTC Vive's controllers were customised for specific functions within the system: the right controller facilitates user orientation in the VR space by utilising the joystick, eliminating the need for physical rotation, while the left controller, also through joystick manipulation, triggers HTTP POST requests to a Python server (running on the Jetson), directing the robot based on user instructions within the VR environment.

3) Web App: The web app accesses the video stream from the Ricoh THETA by instantiating a video component and then attaching a WebRTC stream accessing the stream as the source object of the video element. This video component is placed inside an *A-Frame*, a framework for rendering 360-degree videos in an interactive environment, allowing scrolling to look around and enabling access to the device hardware, such as the gyroscope.

F. Network and connectivity

The Jetson requires a network connection in order to stream video to and receive commands from the user. During the experiments, we used Imperial College London's Campus WiFi, but this could be substituted with any other connection with sufficient bandwidth, such as 4G/5G.

During the time whilst the Jetson is being operated, it is feasible that the Jetson may roam between networks, momentarily disconnect, or have its DHCP lease expire. All of these scenarios can cause the Jetson's IP address to change. The client needs to be able to immediately reconnect to the Jetson's new IP address.

This is accomplished using our Dynamic DNS Daemon. This is a small program that checks the Jetson's IP address every 500ms and updates the domain name to point to the new address if it is found to have changed. This is installed onto the Jetson as a `systemd` service, to ensure that it is reliably started up on boot and restarted if an error occurs. This ensures that the Jetson is always accessible via the network at a known location.

Additionally, the Jetson was configured to run an HTTP reverse proxy server to provide TLS-terminated access to the individual microservices. We obtained a TLS certificate from Let's Encrypt, ensuring that the certificate was trusted by all client devices.

This is important as browsers restrict a lot of functionality when running in an insecure context, especially cross-origin. Instead of configuring each microservice to serve HTTPS, clients connect up via HTTPS to the reverse proxy, which makes a local insecure connection to the microservice. Additionally, the reverse proxy server allowed us to use HTTP/3 transport, which improved connection latency for the web application.

G. AI Assistant

The system also presents the user with the opportunity of learning more information regarding the environment that is explored. The users can interact with an AI Assistant (called "Ava") by engaging in conversations regarding the explored environment, the controlled robot and the tasks needed to complete. Ava is knowledgeable about the system's morphology, the UI on both web app and VR interfaces, how the user should interact with the UI and the capabilities of the robot.

Figure 8 shows the data flow of the AI and Speech pipeline. The first step is for the user to record their question in either of the UI platforms. This audio will be transmitted to the backend pipeline over HTTPS and will be transcripted by passing it through a Speech-to-Text model inference stage. This transcription is then

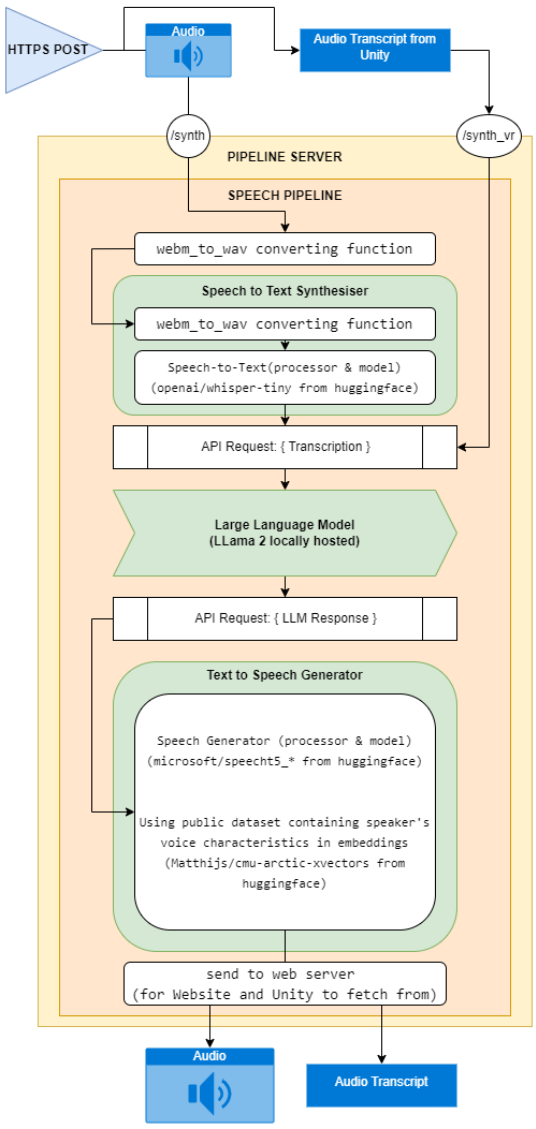

Fig. 8: AI Assistant and Speech Pipeline: Pipeline differentiates between the interface it receives user recordings from (VR or website), transcribes the audio, forwards data to the LLM, synthesises the response to a natural voice and returns generated audio and text back to the interface

forwarded to a locally hosted Large Language Model (LLM). The LLM will generate a response to the user's prompt and will send it back to the pipeline. Considering the accessibility of the system as a priority, the pipeline will transmit both the text-based response and its equivalent audio which was generated through Text-to-Speech inference models. Ava's voice was selected from the database of voices in the device (tablet or laptop) to sound natural and appropriate for our purposes. Hence, a mature UK-speaking female voice was selected.

The AI Assistant is based on Meta's pre-trained LLama 2 13b chat [24] model with a specific knowledge base regarding our task at hand. Prompt engineering techniques were used to forward the relevant information to the LLM system prompt. The LLM allows for a memory-based conversation throughout the user interaction, by saving all of the previous conversations in its prompts. The overall response time, after relatively long conversations (10 interactions), was contained under 5s for the

Fig. 9: 3D Model of Ava

entire pipeline. This is the reason why after every task, the memory of the LLM is reset to preserve quick responses to the user without perturbing the user experience.

The main challenge encountered during the development of this feature was the integration of a reasonably powerful LLM into the computing capabilities of the Nvidia Jetson. Firstly the GPUs were not being detected in the Docker containers, therefore one of the official Nvidia Jetson Container projects [25] was adapted for our use-case and code-base. Secondly, Small Language Models (SLMs) were initially used to minimise the response times, however, their limited capabilities proved to be a challenge in allowing for relevant, fluent and informational responses to be generated. Prompt engineering proved to not be enough to allow for the SLMs to provide good responses regarding the user tasks and the explored environment. Hence, by successfully allowing the containers to access the computing power of the GPUs, it allowed for more powerful LLMs to be implemented, such as LLama 2 7b chat and LLama 2 13b chat model [24].

1) Integrating with the Web App: To visually display Ava on the web app, a GLB file is used, it contains a model of the robot shown in Figure 9. A GLB file was chosen as it provided consistency with the wide range of animations available, it was also essential to include a friendly model to encourage more human interaction. Animations were chosen depending on the status of the AI pipeline (eg. walking idly).

An assistant server was created to retrieve text from the AI pipeline to display in a speech bubble next to the AI Assistant. The output of the AI pipeline is sent to the server using a HTTPS POST request and the string is stored within the server. Every 5 seconds the web app sends a POST request to a different path (`\log`) in the server and the server replies with the stored string from the AI pipeline. Using a different server path ensures that text from the AI pipeline is not sent back to the pipeline and sent to the web app. To show that the AI pipeline was processing the speech input, a "Thinking…" string is displayed, this is important as the user can view whether the system is working correctly. To enable the AI assistant to be aware of the processing status of the AI pipeline, each time an audio file is saved, it sends a POST request to the server signifying the processing status of the pipeline. A flag is raised when the pipeline is processing, once the pipeline has returned an output the flag is changed to signal completion. The text displayed in the speech bubble is dependent on the status flag. Lastly, after receiving the output from the pipeline the text is spoken out loud to the user to enable them to focus on their surroundings whilst hearing the LLM output. The Web Speech API provides a realistic text-to-speech synthesis and the reply is spoken each time the text in the speech bubble is altered.

There were several challenges associated with this system as importing and using animations from a GLB file requires several packages which were incompatible with the React app. To work around this issue HTML is embedded to the React app and model viewer was the only package required. Another challenge was to display the processing status, the initial plan for this function included sharing variables across two React files. However, during implementation, the variable was not updated regularly, which meant the status was incorrect. To solve this issue, a new plan was developed, using the assistant server to ensure that the processing status was correctly updated with each POST request.

2) Integrating with the VR: The VR implementation involved customizing a humanoid model named *Banana Man* sourced from the Unity Asset Store to serve as the AI assistant within the VR user interface. Animation clips from Adobe Mixamo were seamlessly integrated onto the humanoid model using the Unity Editor's Inspector Pane. An animation controller, functioning as a state machine, was then created to manage transitions between animations based on user interactions.

Integrating the assistant 3D model with the VR controllers required two main functionalities: movement and AI assistant capabilities. Movement control involved mapping specific controls from the controllers to enable the humanoid model's directional movement, implemented through adjustments in a Player Input script. For AI assistant capabilities, integration with the Hugging Face API facilitated voice recording triggered by the right controller, with the recorded data sent via a POST request in JSON format to the assistant server. The server provided a text response based on the transcription from the recording script, akin to the functionality of the web app.

V. Experiment Design Plans and Evaluation

A. Experiment Overview

The original experiment design consisted of two experiments that would be conducted in Prince's Gardens, within Imperial College London's campus. Given the time and regulatory constraints, the location of the experiment was moved to inside the Electrical and Electronic building. Though the open-air experiment could have potentially provided a richer insight into the potential deeper immersive and exploratory capabilities of our system, the internal experiment (on Floor 10) still provided sufficient comparison of both the web app and the VR.

B. Human-Robot Interaction Metrics

1) Immersion: As discussed in the literature review in Section II-A, immersion refers to the participant's feelings of presence and engagement in the virtual world. The main hypothesis of the experiment was to determine if the Virtual Reality system improved the participant's feelings of immersion during the teleoperation of Spot. Therefore, to measure immersion, participants would respond to statements from the validated ARI (Augmented Reality Immersion) questionnaire [26]. Although the ARI questionnaire was designed to be used in the context of augmented reality instead of virtual reality, its design was to measure immersion in location-based systems and is still suitable for this experiment.

Participants responded to the statements from the ARI questionnaire through a five point Likert scale, ranging from 'strongly disagree' to 'strongly agree'. The statements used belonged to the immersion levels of engagement and total immersion from the ARI questionnaire.

2) Technology Acceptance: The Technology Acceptance Model is used to predict the acceptance of a new technology system [27]. It is separated into the product's perceived usefulness and perceived ease-of-use to provide an insight into the 'behavioural intention of use'. This is necessary to provide an insight into the general usability of the teleoperation product from a consumer's perspective.

3) Trust Perception: The participant's trust in Spot was recorded using questions from the validated Trust-Perception HRI questionnaire [28]. This questionnaire was used to compare the participant's trust in the VR system compared to the tablet system, as well as measure the overall trust in the system. Unlike the robot partners referred to in the development of the Trust-Perception scale HRI questionnaire, Spot acts as a robot partner in that it is a robotic representation of the user and would not be in the physical presence of the user. Therefore, trust in the collaboration between the robot and the user is necessary to ensure that users are confident that the robot will not affect the safety of others in the remote location, rather than the safety of the user themselves, as is typically explored in trust studies. This is a contributing factor to the user's acceptance of the system. The statements from the trust-perception scale were responded to with a percentage between 0 and 100% to produce an overall percentage trust score.

A participant's perceived trust in a robot system can be significantly influenced by their previous experience and interaction with robotics. Therefore, their initial robot experience was also recorded from the prior experience Trust-Perception HRI questionnaire [28], before conducting the experiments.

Success in these metrics, specifically in immersion, can imply that the benchmarks discussed in Section III have been satisfied (freedom of vision, freedom of movement, freedom of motion, and feelings of presence). Feelings of presence contribute to immersion, as discussed in Section II-A, whilst the former three benchmarks contribute to a participant's feeling of autonomy whilst using the system, which further contributes to both immersion and their trust-perception of the system.

C. Experiment Design

An independent design experiment composed of remote users ($N = 4$) located in a different room in the Electrical Engineering building was carried out to determine whether a 2D screen-based interaction or a 3D VR interaction (*Independent variable*) will yield a more immersive experience (*Dependent variable*).

Given the limited number of accessible experimental participants within a short time, a within-subjects design was performed. To mitigate for any bias due to the order of use, the participants were assigned a group: A or B. Participants in Group A experienced the VR system first, whilst participants in Group B experienced the tablet system first.

The participants completed the experiment in the following 4 stages:

1) User questionnaire before tablet/VR: The participant was asked to complete a questionnaire consisting of questions relating to the participant's previous experience with robots [28]. This questionnaire was based on the Trust Perception Scale-HRI as mentioned in Section V-B3. This was necessary to provide context to the participant's trust perception (which was recorded after the experiment), as discussed in stage 4: User questionnaire after tablet/VR.

An additional questionnaire was completed before the participant's use of the tablet system and VR system respectively. This questionnaire was based on the Technology Acceptance Model [27], with questions related to the participant's perceived usefulness and perceived ease-of-use of both products. This was necessary to provide an insight into the general usability of the product from a consumer's perspective, and to identify the user's acceptance of the system.

2) Introduction to joystick controls for either tablet/VR: Before each experiment with the VR or tablet, the participant was given an overview of how to control Spot. They were instructed on the use of the two modes: *standing* and *walking*, and the different types of movement with their corresponding joystick controls. They were made aware of the AI assistant and they were also given an initial overview of the tasks that they would have to complete throughout the experiment.

3) Obstacle free navigation and exploration: The participant was instructed to complete three tasks:

- Task 1: exit the lab and enter the main corridor
- Task 2: read the door number of a specified door
- Task 3: a picture of the Queen's Tower had been placed on a wall. The participant had to find the picture and identify it, with the option to ask the AI Assistant further questions about it

The fixed nature of these tasks was necessary to ensure that the participant experiences with the VR interaction were comparable to that of the screen-based interaction. These tasks allowed the participant to explore an area, which is the basis of the system, whilst the specificity of the tasks meant that all components of the system were utilised.

Task 1 focused on joystick control, whilst task 2 relied on the quality of the video streaming. Task 3 required the participant to either use Spot's poses or the physically move their body (if using the VR system) or the tablet to exploit the 360 camera, in order to locate the image. This ensured that the participant's responses were based on the complete design, whilst allowing an observation into how different components would be used in practise, as well as providing an insight into the technical quality of the project.

4) User questionnaire after tablet/VR : After each stage (VR or tablet), the user was given another questionnaire about their experience. This questionnaire was separated into two sections. Section A focused on the measurement of immersion, from the ARI questionnaire [26], whilst section B consisted of questions relating to the measurement of the participant's trust of the system, from the Trust-Perception HRI questionnaire [28].

VI. Results and Discussion

Given the time constraints, the participant's demographic was exclusively students within the Department of Electrical and Electronic Engineering. These participants have had much experience with technology and this was considered in the results.

Statistical analysis using two group comparison tests, for example Mann-Whitney U-Test and T-test, were performed. The t-test assumes that the data is approximately normally distributed and is typically used with larger sample sizes. With a sample size of 4, the assumption of normality is difficult to satisfy. Moreover, a t-test with such a small sample size would have very low statistical power, meaning the ability to detect a true effect is

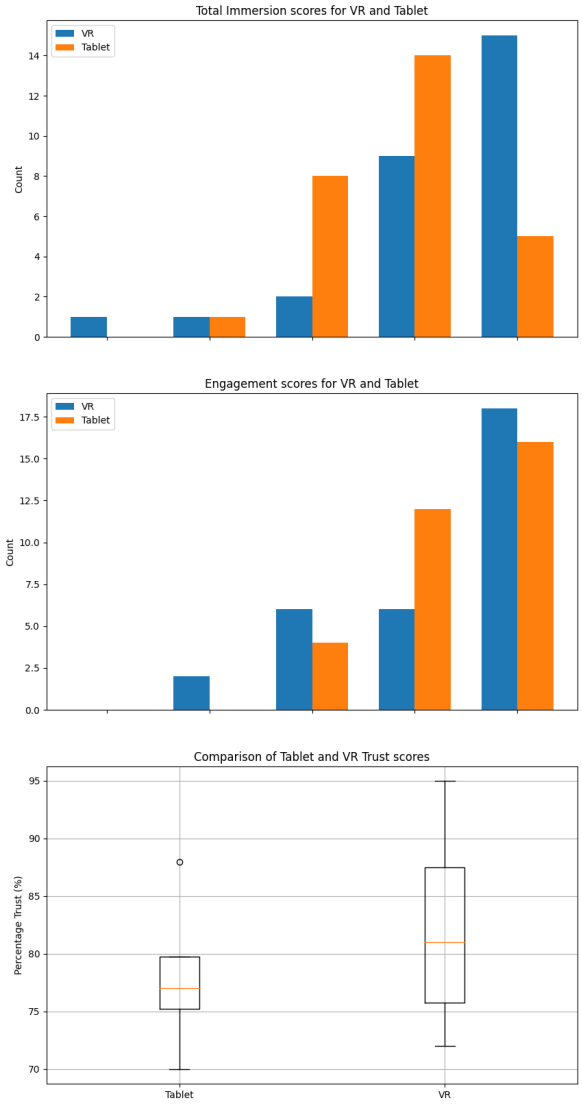

Fig. 10: ARI Total Immersion, ARI Engagement and Trust scores

limited. The Mann-Whitney U-test can be used instead since it is a non-parametric test that doesn't assume normality and is more suitable for small or non-normal samples compared to the t-test. However, with only 4 observations per group, this test also suffers from a lack of power. Nevertheless, some statistical insights can be grasped from the small sample size, with more convincing results obtainable through further experimentation with more participants.

Table I indicates the results of the questionnaire based on the Technology Acceptance Model.

	VR	Tablet
Mean	4.081633	4.071429
Median	4	4
Mode	4	4.25
IQR	0.160714	0.375

TABLE I: Technology Acceptance Model (TAM) results

Figure 10 displays the results for the ARI total immersion, ARI engagement, and trust-perception scores.

The results of the Mann-Whitney U-Tests and t-tests are described in Table II comparing to a 5% significance value ($\rho < 0.5$).

	Mann-Whitney U-Test		T-Test	
	ρ-value	Significant?	p-value	Significant?
TAM	0.07589	×	1	×
Total Immersion	6.309×10^{-5}	✓	0.3226	×
Engagement	0.4011	×	0.6156	×
Trust Perception	3.537×10^{-6}	✓	0.5239	×

TABLE II: Mann-Whitney U-Test and T-Test significance results

The reported difference in immersion for each control method is significant at $\rho < 0.05$ for the Mann-Whiteney U-Test. In addition, the trust perception difference for each control method is significant at $\rho < 0.05$. Analysing the raw results of each of these metrics, it is clear that the VR method of control potentially achieves overall more freedom of vision, feeling of motion and freedom of presence (see Section III). Engagement and Technology Acceptance did not reach statistical significance; there was insufficient evidence to declare any difference in participant experience of engagement or perceived usability and ease of use for the VR system in comparison to the tablet system. Regarding the Technology Acceptance results, the participant demographic's generally high prior technological experience and exposure could mean there was quite a high acceptance towards both systems equally. However, the small sample size is likely to have been the contributing factor to the insignificant evidence, so further experimentation would be required to determine if there were differences in Engagement or Technology Acceptance between the systems.

The t-tests for all evaluation metrics reported insignificant evidence of any difference between the two control methods. As previously mentioned, the t-test requires the assumption of a normally distributed dataset and given the sample size ($N = 4$), this assumption is violated. Nevertheless, looking at the raw scores for ARI Engagement indicates that the tablet and VR maintained similar results. This can be attributed to users finding that the control via the VR joysticks is more difficult to grasp at first than the tablet. Furthermore, small latency issues associated with using Unity compared to the web application when sending joystick commands to Spot can lead to users being less absorbed in the real-time immersive experience.

VII. Conclusion and Future Work

This paper presented the design and evaluation for Avatour, a VR-mediated robot telepresence system for the exploration of distant places in the real world. Avatour presents the discrepancy in immersion and accessibility between two user interfaces: virtual reality and traditional 2D interface. The results demonstrated an improvement in the immersive level of total immersion for the virtual reality system in comparison to the screen-based interface, however the small sample size meant that the difference in the immersive level of engagement was inconclusive. The results also demonstrated a higher level of trust in the VR system than in the screen-based system, however for more concrete findings regarding both the acceptance of the technologies and the immersive level of engagement, further experimentation with more participants is required.

A. Future work

1) VR Interface: The Vive controllers have proven to be very unintuitive as joysticks, hence the introduce of new controllers on the VR would provide a smoother and improved robot control.

2) Web App Interface: The web app would highlight components unintuitively with specific user interactions (e.g. holding down onto joysticks), hence a solution for this should be found.

The map feature that is shared between the web app and the VR UI could include more 360 interactive features on the map, such as pinning locations and showing navigation paths and provide panning and zooming integration.

3) Video & Audio Streaming: The video encoder should adjust the video bit-rate dynamically depending on the client's available bandwidth, hence adaptive bit-rate would highly help with the streaming capabilities.

Audio streaming is only currently limited to the VR clients, hence more work on the environment audio recording, transmission and directional noise cancellation would allow for better experiment comparison.

4) AI Assistant: Tailoring the knowledge base of the LLM specifically for the task and integrating multi-modality (vision through selected video frames) would enable the system to further understand the robot's surroundings, thus allowing the user to ask specific questions regarding what they are seeing.

5) Other: The Nvidia Jetson was powered by a dedicated battery. Therefore, powering it from Spot's battery would allow for better integration of the system.

REFERENCES

[1] J. Leigh, T. DeFanti, A. Johnson, M. Brown, and D. Sandin, "Global teleimmersion: Better than being there," 1997.

[2] S. Hudson, S. Matson-Barkat, N. Pallamin, and G. Jegou, "With or without you? interaction and immersion in a virtual reality experience," 2019. DOI: https://doi.org/10.1016/j.jbusres.2018.10.062.

[3] R. W. Belk, "Possessions and the Extended Self," 1988. [Online]. Available: https://doi.org/10.1086/209154.

[4] B. Dynamics, *Spot to the Rescue — Boston Dynamics*, https://bostondynamics.com/blog/spot-to-the-rescue/, [Accessed 31-01-2024], 2023.

[5] R. Eveleth, *The surgeon who operates from 400km away, bbc*, https://www.bbc.com/future/article/20140516-i-operate-on-people-400km-away, [Accessed 31-01-2024], 2014.

[6] S. Newsroom-AG, *A Day in the Life With Ballie: An AI Companion Robot for the Home*, https://news.samsung.com/us/samsung-ballie-ai-companion-robot-home-video-ces-2024/, [Accessed 31-01-2024], 2024.

[7] R. C. Quesada and Y. Demiris, "Holo-spok: Affordance-aware augmented reality control of legged manipulators," 2022. DOI: 10.1109/IROS47612.2022.9981989.

[8] R. C. Quesada and Y. Demiris, "Design and evaluation of an augmented reality head-mounted display user interface for controlling legged manipulators," 2023. DOI: 10.1109/ICRA48891.2023.10161278.

[9] [Accessed 02-02-2024]. [Online]. Available: https://tx-inc.com/en/technology/.

[10] J. Steuer, F. Biocca, M. R. Levy, *et al.*, "Defining virtual reality: Dimensions determining telepresence," 1995.

[11] H. Van Kerrebroeck, M. Brengman, and K. Willems, "Escaping the crowd: An experimental study on the impact of a virtual reality experience in a shopping mall," 2017. DOI: https://doi.org/10.1016/j.chb.2017.07.019.

[12] Y. Oh, R. Parasuraman, T. Mcgraw, and B.-C. Min, "360 vr based robot teleoperation interface for virtual tour," 2018.

[13] M. Sakashita, H. Kim, B. Woodard, R. Zhang, and F. Guimbretière, "Vroxy: Wide-area collaboration from an office using a vr-driven robotic proxy," Association for Computing Machinery, 2023. [Online]. Available: https://doi.org/10.1145/3586183.3606743.

[14] [Accessed 08-02-2024]. [Online]. Available: https://www.fortunebusinessinsights.com/industry-reports/video-conferencing-market-100293.

[15] [Accessed 20-01-2024]. [Online]. Available: https://dev.bostondynamics.com/.

[16] N. Upadhyay and A. Karmakar, "Speech enhancement using spectral subtraction-type algorithms: A comparison and simulation study," 2015. DOI: https://doi.org/10.1016/j.procs.2015.06.066. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050915013903.

[17] K. Yamato, A. Sugiyama, and M. Kato, "Post-processing noise suppressor with adaptive gain-flooring for cell-phone handsets and ic recorders," 2007. DOI: 10.1109/ICCE.2007.341477.

[18] *Panoramic skybox*, [Accessed 08-02-2024], 2024. [Online]. Available: https://docs.unity3d.com/Manual/shader-skybox-panoramic.html.

[19] *Obs studio*, [Accessed 08-02-2024], 2024. [Online]. Available: https://obsproject.com/.

[20] R. K., *Webcam processing in unity shader graph*, [Accessed 08-02-2024], 2020. [Online]. Available: https://medium.com/xrpractices/webcam-processing-in-unity-shader-graph-61455d4e3796.

[21] *Gstreamer: Open source multimedia framework*, [Accessed 08-02-2024], 2024. [Online]. Available: https://gstreamer.freedesktop.org/.

[22] B. Posey, *Real time streaming protocol (rtsp)*, [Accessed 12-02-2024], 2023. [Online]. Available: https://www.techtarget.com/searchvirtualdesktop/definition/Real-Time-Streaming-Protocol-RTSP.

[23] A. Adegbenro, *What is webrtc? (explanation, use cases, and features)*, [Accessed 12-02-2024], 2023. [Online]. Available: https://ably.com/blog/what-is-webrtc.

[24] H. Touvron, L. Martin, K. Stone, *et al.*, *Llama 2: Open foundation and fine-tuned chat models*, 2023.

[25] *Local_llm*, [Accessed 01-03-2024]. [Online]. Available: https://github.com/dusty-nv/jetson-containers/tree/master/packages/llm/local_llm.

[26] Y. Georgiou and E. A. Kyza, "The development and validation of the ari questionnaire: An instrument for measuring immersion in location-based augmented reality settings," 2017. DOI: https://doi.org/10.1016/j.ijhcs.2016.09.014.

[27] F. Davis, R. Bagozzi, and P. Warshaw, "User acceptance of computer technology: A comparison of two theoretical models," 1989. DOI: 10.1287/mnsc.35.8.982.

[28] K. E. Schaefer, "Measuring trust in human robot interactions: Development of the "trust perception scale-hri"," in R. Mittu, D. Sofge, A. Wagner, and W. Lawless, Eds. 2016. [Online]. Available: https://doi.org/10.1007/978-1-4899-7668-0_10.