Mixed Reality (Autumn 2024) - AR Remote Surgical Assitant

Supervised by: Matthias Rüger (MD, Zurich), Javier Narbona Cárceles (MD PhD, Madrid)

Luca Sichi
ETH Zürich
lsichi@student.ethz.ch

Louis Niederlöhner
ETH Zürich
lniederloehn@student.ethz.ch

Siddharth Menon
ETH Zürich
menons@student.ethz.ch

Ayshé Opan
ETH Zürich
ayopan@student.ethz.ch

Abstract

This report presents the design and development of an AR Remote Surgical Assistant, a mixed-reality application enabling surgeons to receive real-time feedback and guidance from assistants located remotely. The system integrates a Magic Leap 2 AR headset for the surgeon and a desktop-based application for the assistant, connected through WebRTC for video, audio, and data communication. Key functionalities include live video streaming, two-way audio, annotated visuals, and 2D/3D projections of shared reference models and images. The project prioritizes minimal interference with surgical tasks while fostering collaboration and training opportunities without the need for co-location. The code can be found in this github repository.

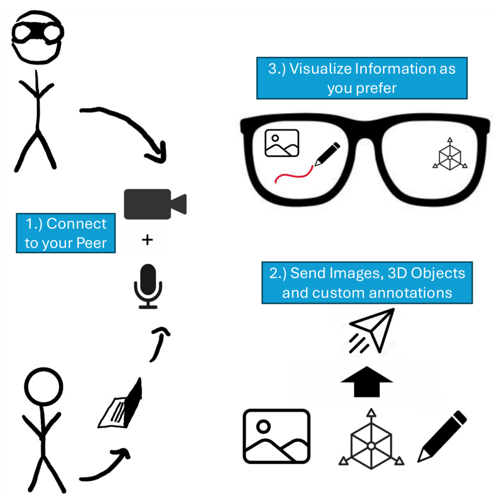

Figure 1. Intended Workflow

1. Introduction

This Remote Surgical Assistant project aims to build an augmented reality (AR) application, with the goal of allowing a surgeon to receive feedback and instructions from an assistant situated in a separate location. The application 1 would facilitate information sharing through visual and auditory cues.

More specifically, the assistant can visualize the surgeon's view through a live video stream on a desktop application deployed on their computer. Then, they can relay feedback and instructions via audio chat and drawn annotations. The surgeon, who is wearing the AR device, can hear the assistant's live audio as well as respond. Additionally, they can see the annotations, both on a 2D canvas and projected into the 3-dimensional space they are in. The assistant can also share reference images, such as MRI and X-Ray images, along with 3D reference models that the surgeon can refer to and interact with during the procedure.

The goal of the project is to build a system that facilitates training and collaboration, without the need for all participants to be present in the same location.

2. Related Works

The use of augmented reality has significant potential in the medical field, with AR systems for medical applications being discussed already in 1998 by Tang et al. [6]. Since the development of usable AR glasses, many practical applications and use cases have been proposed.

Cofano et al. [1] uses AR glasses to visualize cortical bone trajectories while maintaining focus on the surgical field. Simultaneously, a remote assistant could provide support through a 2D video stream.

To provide more intuitive guidance, Meulstee et al. [5] overlays the image data directly on the patient's body which also resolves the switching focus problem.

The potential for remote collaboration in a surgical setting is further exemplified by Davis et al. [2], introducing a system leveraging iPads to facilitate surgical assistance from remote experts. Both the surgeon and the assistant shared a composite view of video feeds, enabling real-time collaboration and education across locations.

Lim et al. [3] demonstrates ergonomic benefits of AR systems in surgical environments, improving posture and reducing physical strain by eliminating the need to frequently check separate screens.

Beyond strictly medical applications, AR systems have been explored for their general collaborative potential. Magic Leap Assist [4] presents a system where digital content and real-world environments could be shared between AR glasses and remote devices. By transmitting a mesh of the surroundings, the system enables spatial annotations and content sharing.

Similarly, Yu et al. [7] introduced the concept of duplicated reality to co-locate remote participants in a shared virtual space by creating a digital replica of the environment.

These studies show the transformative potential of AR in enhancing medical practice, enabling remote collaboration, and addressing ergonomic challenges, all of which our application tries to achieve.

3. Methods

3.1. Application Design and Development

We started planning our prototype with a reference concept idea of what the end product should look like, provided to us by our supervisors (Figure 2). As the project developed over the following weeks, we built out a system with the following desired functionality:

- Live video transmission from surgeon's headset to assistant client.
- Audio transmission to facilitate voice chat between both parties.
- The ability to screenshot and annotate images on the assistant client, as well as erase and modify annotations.
- 2D and 3D reprojected visualization of annotations on the surgeon side.
- The capability for the assistant to upload and share images and 3D model files with the surgeon.

For the purposes of this project, we built the proposed system with the surgeon-side application on a Magic Leap 2 device and the assistant application deployed and tested on Windows laptops. Both applications were built in Unity, using C# as our main programming language.

Some further details about both surgeon and assistant applications are described below. Figures 3 and 4 show the two views in the final end product.

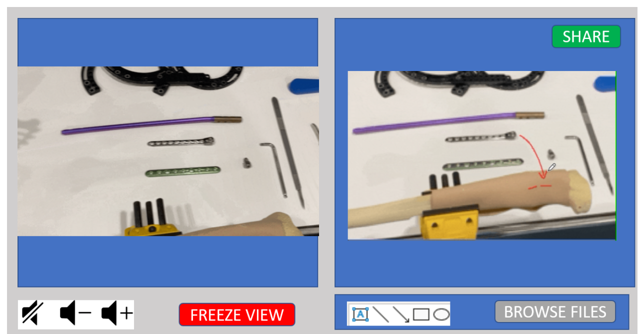

Figure 2. Reference concept of proposed application design

3.1.1 Surgeon Application

For the surgeon side, we aimed to design an interface that would create minimal interference with the surgeon's ability to perform the surgery. This includes a network connectivity UI menu that is opened and closed using a specific hand gesture (opening the left hand palm and holding for 0.2 seconds). Since the clarity of the outside world is of utmost importance to the surgeon, we reduced the dimming on the device itself. Additionally, we also removed the hand prefab models provided by the Unity XR template, so as to reduce the visual clutter on screen.

Two 2D panels, initially not visible to the surgeon, are placed in the surrounding space, which are later populated with the 2D annotations and shared image files, sent over by the assistant. These panels, along with any 3D reference model also sent by the assistant, can be interacted with and repositioned by the surgeon through direct contact or by pinch gestures. Finally, the annotations, after 3D reprojection, are also visually displayed in the world using spawned sphere primitives.

3.1.2 Assistant Application

On the assistant application, we provide two tabs: one for the live surgeon-view video stream and the other for loading and sharing images (jpeg and png supported) and 3D model files (obj files supported). On the live video stream tab, the assistant has the capability to pause/screenshot the current frame and draw annotations on the screen.

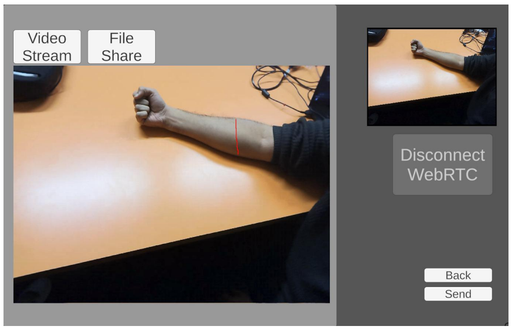

Figure 3. Assistant View

Figure 4. Surgeon View (on Magic Leap 2)

3.2. WebRTC Communication

To allow the two applications to communicate with each other, once deployed, we use communication through a WebRTC interface and a Python server hosted on a Windows laptop. WebRTC is a project that allows mobile applications to carry out real-time communication through the web. It provides us with the capabilities to send and share video, audio, and generic data streams, proving to be a relatively simple and free solution to our goals of creating a streamlined communications system.

The WebRTC package for Unity provided a simple Unity wrapper that allowed easy integration of these systems into our project. The video and audio stream options available in WebRTC are sufficient to allow us to stream the live video stream from the Magic Leap device along with the two-way audio. The additional generic data streams made available to us are used to share 3D model files from the assistant to the surgeon and depth information from the surgeon to the assistant.

3.2.1 Limitations

Due to the data size limitations inherent within WebRTC, we cannot send large files without needing to split it into smaller chunks of data. The depth data is of a fixed size that is known to us, which allows us to split them into pre-defined chunks and keep track of them as they are streamed to the assistant side for further processing (see Sec. 3.3 for more information about how the data is processed). Unfortunately, because the 3D model files can be of variable sizes, we currently do not support sharing of large model files.

To address bandwidth constraints, we implemented a reduction in the frame rate for transmitting the depth map. Transmitting the depth map at every update overwhelmed the data channels, resulting in only one operational channel. Consequently, since the depth map was divided into four segments, only one quarter of the map was successfully updated. To mitigate this issue, we opted to transmit the depth map at intervals of 0.2 seconds. This adjustment ensured consistent and reliable data transfer across all channels.

3.3. 3D Reprojection

In this part of the project, we aimed to project the annotations from the 2D image plane into 3D space. This process requires the RGB video frame, the corresponding depth frame, the annotations, the camera's rotation R and translation T, as well as the intrinsic matrices K_{depth} and K_{rgb} for the depth and RGB cameras, respectively.

To achieve this, we chose to represent each annotated pixel as a small sphere in 3D space. The process begins by querying the depth map at the specific pixel locations corresponding to the annotations. However, as the depth and RGB cameras operate in different coordinate systems, it is first necessary to transform the homogeneous annotation coordinates $(u, v, 1)$ from the RGB camera system to the depth camera system. This transformation is performed using the following equation:

$$uv_{depth} = K_{depth} K_{rgb}^{-1} uv_{rgb} \quad (1)$$

Using the transformed coordinates uv_{depth}, we queried the depth image to retrieve the corresponding depth value $d[u, v]$. The next step involved projecting the 2D point into

3D camera coordinates. This was accomplished by applying a transformation based on the intrinsic matrix of the depth camera.

$$p_{depth} = d[u, v] * K_{depth}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{2}$$

Once the point was in 3D camera coordinates, it was further transformed into world coordinates using the camera's rotation R and translation T. This transformation ensures that the 3D points are represented accurately in the global coordinate system.

$$p_{world} = R * p_{depth} + T$$

With the resulting world coordinates, we instructed the Magic Leap to render a small sphere at the appropriate location in the environment, effectively visualizing the annotations in 3D space.

3.3.1 Implementation

Unfortunately, the Magic Leap documentation does not provide access to the depth camera intrinsics through the API when using our project template with OpenXR. The available information in the documentation pertains to the deprecated MLSDK, which was not applicable to our setup. As a result, alternative solutions had to be devised for transforming the coordinates as described in Equations 1 and 2.

Our initial approach involved classical camera calibration techniques. However, since we were working with a depth camera, this method proved to be challenging, as it requires specially designed physical calibration objects that were not available to us.

Ultimately, we adopted a simpler solution for Eq. 1, assuming no lens distortion. This approach involved applying a cropping transformation. The values for the top-left corner of the crop were determined experimentally, while the dimensions of the RGB and depth images were known. As shown in Figures 5 and 6, the depth image exhibits a significantly larger field of view compared to the RGB image.

To obtain the correct value of p_{depth} as described in Eq. 2, our solution leverages trigonometry and the additional knowledge of the depth camera's field of view (FOV). For both x and y dimensions, we apply specific transformations based on the pixel's position relative to the image center. These transformations account for the nonlinear mapping induced by the FOV and ensure accurate reprojection into 3D space.

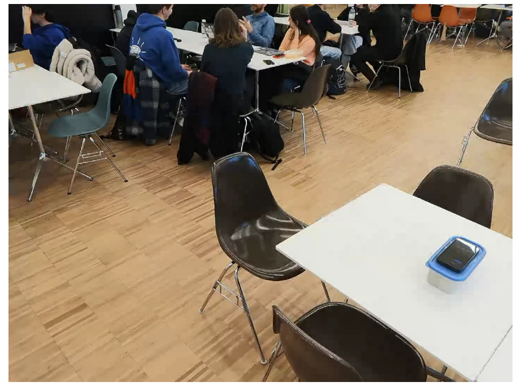

Figure 5. RGB image

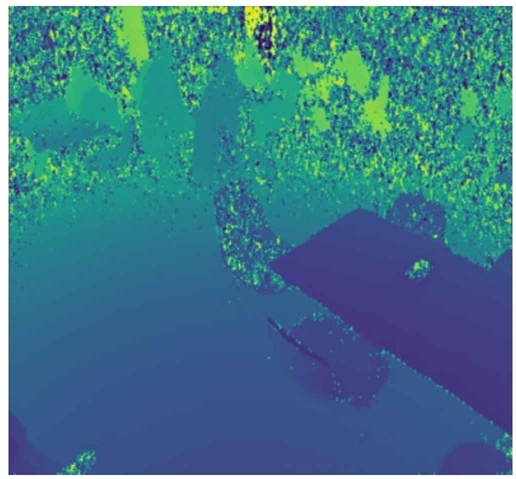

Figure 6. Visualization of the depth map

$$t = \frac{u - \frac{size}{2}}{\frac{size}{2}}$$

$$u_{new} = depth * \sin\left(\arctan(t * \tan\frac{fov}{2})\right)$$

If $u > \frac{size}{2}$, else we use

$$t = 1 - \frac{2u}{size}$$

$$u_{new} = -depth * \sin\left(\arctan(t * \tan\frac{fov}{2})\right)$$

This approach allows us to calculate the x and y coordinates, which are then used to compute p_{depth} for Eq. 2. While it is theoretically possible to calculate the camera intrinsics by hand using the field of view, as shown in Eq. 3, this assumes the principal point lies at the center of the image. We initially attempted this method but found that our custom trigonometric approach provided more accurate and fine tunable results in practice.

As observed in Figure 6, the depth map often contains significant noise, particularly in regions with specular surfaces or when the objects are beyond the depth sensor's range. To address this, we applied smoothing to the depth map using an averaging filter with a window size of 9. This step reduced noise and improved the reliability of the projected points.

$$f = \frac{size/2}{\tan(FOV/2)} \tag{3}$$

4. User Study & Results

Our study was designed to evaluate the usability, effectiveness, and potential for real-world application of the AR Remote Surgical Assistant system. Participants tested the implementation and then answered survey questions about their experience. The survey was structured into five sections: background (familiarity with AR and the medical field), usability of features, user experience, and potential future development, concluding with general feedback on the system.

Evaluation of the Magic Leap: Most of the participants were comfortable with the headset; 60% gave it a rating of 4 or 5 out of 5, and no one rated it lower than 3. However, challenges in manipulating 3D objects were noted, which made this feature less effective for some users.

The audio communication was highly rated and widely appreciated, being considered the most useful feature for understanding the assistant's instructions. Similarly, annotations received positive feedback for being effective and useful. Nonetheless, some participants highlighted the need for improved precision. Notably, one participant from the medical field aspiring to be a surgeon rated the precision of annotations as excellent, giving it a perfect score of 5 out of 5.

Overall, participants believed the system has the potential to enhance surgical collaboration in real-world scenarios, though the current level of precision may limit its applicability. One participant suggested that with the existing precision, the system could be particularly useful for students in the early stages of medical training.

For future development, participants were asked to rate the importance of proposed features on a scale of 1 to 5. The proposed features and their average ratings were as follows:

- More precise reprojection: This was rated as the most important, with an average score of 4.6.
- Assistant wearing a headset: This feature received an average score of 3.5.
- Replacing the assistant with AI: This was considered the least important, with an average score of 2.3.

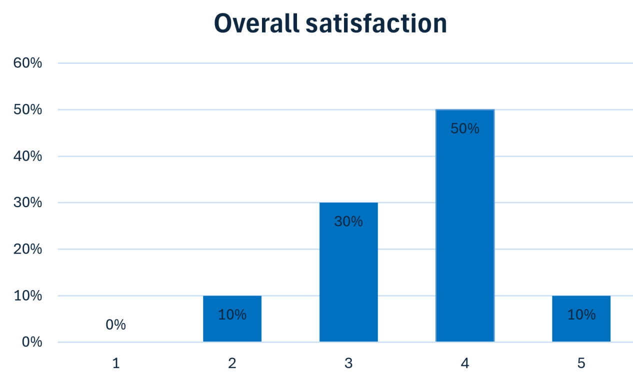

Figure 7

As shown in figure 7, people were generally satisfied with the interface overall. The person that rated it 2 is the one with the most AR experience, primarily criticizing the user interface (UI), stating that it requires redesigning.

The results of the user study suggest that while the AR Remote Surgical Assistant system demonstrates some potential, its current precision and usability limitations reduce its practicality for real-world surgical applications. Features such as audio communication and annotations were well-received, but participants identified key areas for improvement, particularly in precision and ease of 3D object manipulation. Moving forward, addressing these limitations is critical for the system to be viable in medical settings beyond educational or preparatory use cases.

5. Future Work

Looking forward towards further improvements that can be made to this application's workflow, one avenue to explore would be to improve the precision of our depth estimation and 3D reprojection calculations. To this end, it may also be useful to look into building and testing the workflow on other AR/MR devices, such as the Hololens, which may be able to provide more precise sensor information.

Improvements could also be made to the communication system, specifically aiming towards transmission of larger data files, as well as supporting loading and sharing of more file formats (such as pdfs, glb, fbx). Exploring the use of more sophisticated communication protocols could also alleviate the data size and bandwidth limitations inherent within WebRTC.

References

[1] Fabio Cofano, Giuseppe Di Perna, Marco Bozzaro, Alessandro Longo, Nicola Marengo, Francesco Zenga, Nicola Zullo,

Matteo Cavalieri, Luca Damiani, Daniya J Boges, et al. Augmented reality in medical practice: from spine surgery to remote assistance. *Frontiers in Surgery*, 8:657901, 2021. 1

[2] Matthew Christopher Davis, Dang D Can, Jonathan Pindrik, Brandon G Rocque, and James M Johnston. Virtual interactive presence in global surgical education: international collaboration through augmented reality. *World neurosurgery*, 86:103–111, 2016. 2

[3] Ae Kyeong Lim, Junsun Ryu, Hong Man Yoon, Hee Chul Yang, and Seok-ki Kim. Ergonomic effects of medical augmented reality glasses in video-assisted surgery. *Surgical Endoscopy*, pages 1–11, 2022. 2

[4] Magic Leap. *Magic Leap Assist*, 2023. Accessed: 2025-01-05. 2

[5] Jene W Meulstee, Johan Nijsink, Ruud Schreurs, Luc M Verhamme, Tong Xi, Hans HK Delye, Wilfred A Borstlap, and Thomas JJ Maal. Toward holographic-guided surgery. *Surgical innovation*, 26(1):86–94, 2019. 2

[6] Son-Lik Tang, Chee-Keong Kwoh, Ming-Yeong Teo, Ng Wan Sing, and Keck-Voon Ling. Augmented reality systems for medical applications. *IEEE Engineering in Medicine and Biology Magazine*, 17(3):49–58, 1998. 1

[7] Kevin Yu, Ulrich Eck, Frieder Pankratz, Marc Lazarovici, Dirk Wilhelm, and Nassir Navab. Duplicated reality for co-located augmented reality collaboration. *IEEE Transactions on Visualization and Computer Graphics*, 28(5):2190–2200, 2022. 2