UNIVERSITY OF OKLAHOMA
GRADUATE COLLEGE

INTRAOPERATIVE VIRTUAL SURGICAL PLANNING THROUGH MEDICAL MIXED REALITY

A THESIS
SUBMITTED TO THE GRADUATE FACULTY
in partial fulfillment of the requirements for the
Degree of
MASTER OF SCIENCE

By
Mohammad Abdul Mukit
Norman, Oklahoma
2021

INTRAOPERATIVE VIRTUAL SURGICAL PLANNING THROUGH MEDICAL MIXED REALITY

A THESIS APPROVED FOR THE

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY THE COMMITTEE CONSISTING OF

Dr. Samuel Cheng

Christian El Amm, MD

Dr. Bin Zheng

Dedication

This thesis is dedicated to my parents, my wife, my brother, and the rest of my family.

Acknowledgements

I am luckier than most graduate students for having not one but two great mentors, always watching over my academic growth and progress towards my career objectives.

I must thank Dr. Samuel Cheng, my thesis supervisor, for his guidance and for offering me the opportunity to work on such an interesting and worthwhile research project. Without his initiatives, I wouldn’t have made it to the US and got my Graduate Fellowship for my MS.

I must thank Christian El Amm, my mentor. Without his leadership and visionary medical insights, IntraOpVSP would have not been possible for me to engineer. I am grateful to him for all the experiences, accomplishments, and adventures I had over the last three years. Christian El Amm arranged for all the data and provided all the medical expertise, I needed to make IntraOpVSP.

All the committee members have my sincere gratitude for their time and consideration in reviewing this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

1 ABSTRACT

The mission is to augment every surgery. Compared to a pilot flying a plane or even a regular google-map user on his way to work/home, surgeons today have their instruments clustered behind them hanging in the wall. The Google Maps user or the pilot gets constant real-time updates regarding where they are, what to do next, and other vital data that helps them make split seconds decisions. They don't have to memorize every turn and detail of every landmark along the way. On the other hand, surgeons today must do rigorous surgical planning, memorize the specifics of each unique case, and all the necessary steps to ensure the safest possible surgery. Then they engage in complex procedures for several hours, with no targeting or homing devices or head-mounted displays to assist them. They must feel their way to their objective and hope everything goes as they planned. Through our research, we aim to change this practice. We are making the "google-maps" for surgery. Intraoperative Virtual Surgical Planning (IntraOpVSP) is one of the essential pillars of this vision. It is made for visualizing and interacting with Virtual Surgical Planning (VSP) intraoperatively. IntraOpVSP perfectly aligns VSP data with the patient’s body and through holographic display, allows the surgeons to look directly inside a patient’s body to inspect various anatomical structures. Through our research, we have identified essential functionalities that allow surgeons to align, interact with, and visualize the patient CT scan data as intuitively as possible. Since the development of IntraOpVSP, it has been used in over 35 surgeries at the University of Oklahoma, Medicine and is now in routine use. This thesis report focuses on how IntraOpVSP was created and how it was iteratively refined to become as pragmatic as possible.

2 INTRODUCTION

2.1 CHALLENGES IN EXISTING VISUALIZATION METHODS

Surgeons today rely primarily on conventional two-dimensional screens to view surgical plans and anatomical data during surgery. They entirely rely on two-dimensional displays to understand and operate on incredibly complex patient pathology. To view the critical data during surgery, they must look away from the subject or region of interest where the surgery is happening. Consequently, the surgeons must continuously shift their focus between the 2D screen and the subject. This is a very unintuitive process and forces the surgeons to take focus off the patient. For example, a surgeon may have to pause a procedure to consult relevant radiologic data, potentially disrupting the smooth flow of surgery. It is surprising that even with all the advancements in the scientific world, the surgeons are still performing surgery they used to a few decades ago.

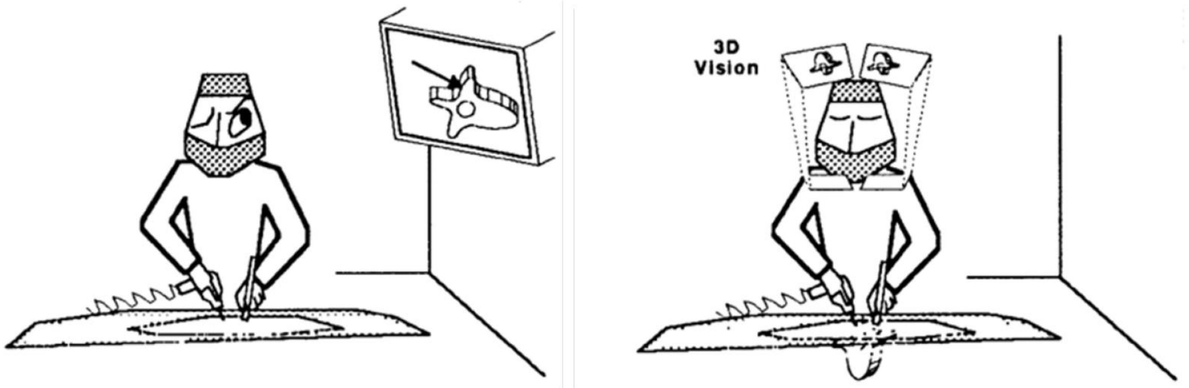


Figure 1: AR for scoliosis surgery proposed in 1995 (Peuchot et al.[1] *)*

The above figure illustrates the concept of using “Virtual Reality” (actually an AR approach) as an operative tool during scoliosis surgery [1]. Peuchot et al. proposed to superimpose a 3D transparent vision of the vertebra directly on the surgeon’s operative view. The expected advantages of such an approach include i) knowledge of the correction in the progress of a scoliotic curve, ii) safety during the placement of surgical implants, e.g. pedicular fixation, and iii) restriction in the extent of operative exposure [1]. However, nearly 25 years later, the “proposed" AR solution is still not a common practice in the OR. There have been attempts of using Virtual Reality and Augmented Reality over the last decade to address this issue.

2.2 MEDICAL VIRTUAL AND AUGMENTED REALITY

2.2.1 Introduction

Virtual Reality (VR) is an immersive display technology in which the user is submerged in a simulated environment. While conventional displays are two-dimensional, a Virtual Reality is an immersive 3D experience. The VR headset places the user inside an artificial 3D world experience. Through controllers the users are also capable for interacting with the virtual 3D world entities. However, the user loses complete sight of the real world in VR.

As the name suggests Augmented Reality or AR, augments the real world through adding 3D virtual contents in the user's perception of the real world. Unlike VR-headsets, AR-headsets don't submerge the user in a completely artificial 3D world. Instead, through AR headsets, the user can still see the real world. However, the real world's view is enhanced in the AR display via attaching interactable virtual 3D contents to physical real-world objects. AR usually consists of three fundamental features: a mixture of virtual and real-world contents, real-time interaction capability with the simulated contents, and accurate 3D registration of the virtual and real objects [2].

The displays used in AR headsets are usually video-see-through displays, in which the world is shown to the user as a video stream captured by several cameras placed on the headset. The simulated 3D contents are overlaid on the video stream. Thus the sensory information can be constructive (i.e. additive to the natural environment), or destructive (i.e. masking of the natural environment) [3]. In this way, augmented reality alters one's ongoing perception of a real-world environment, whereas virtual reality completely replaces the user's real-world environment with a simulated one [4].

2.2.2 Applications in Surgery

Augmented Reality & Virtual Reality in healthcare market size was valued at USD 2.0 billion in 2020 and is expected to expand at a compound annual growth rate (CAGR) of 27.2% from 2021 to 2028 [5]. Due to the great potential of these technologies, there have been a lot of interest in the medical device industry to develop impactful AR/VR products. This section focuses on the commercial effort by various companies to be at the forefront of VR and AR in healthcare.

Several industry leaders use produce VR solutions to perform virtual surgical planning [6]–[8]. The solutions allow surgeons to view reconstructed 3D anatomical structures in VR. It allows for interactive and collaborative virtual surgical planning. This minimizes the risk of facing any unexpected circumstance during surgery thus making the surgery safter for both the surgeons and the patients. Additionally, the immersive 3D collaborative planning helps increase efficiency of the surgeons.

Providing a surgical rehearsal platform to help train surgeons is another very useful application of VR in surgery [6], [8]–[11]. The solutions allow surgeons to practice and improve surgical techniques in the simulated surgery. The solutions also often involve haptics for tactile feedback. These visualization are also used for extensive educational purposes as well [10], [12].

Surgical Theater also developed the SyncAR system which is used during surgery [8]. In this AR system a camera stream of the operative field is captured and shown on a 2D conventional monitor with the virtual anatomy structures are simulated and overlaid on the video stream.

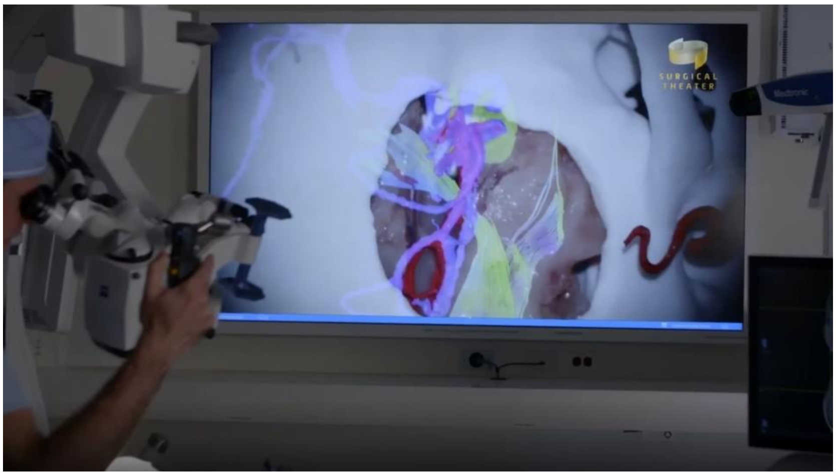


Figure 2: SyncAR system by Surgical Theater draws AR contents on a 2D display for intraoperative usage.

2.2.3 Limitations

The existing VR-AR headsets offer highly detailed and realistic rendering of the virtual content. Thus, the medical community found these technologies extremely useful for various steps before the actual surgery takes place. However, there are some limitations in other fonts that prohibit their acceptance in intraoperative applications.

Most AR-VR surgery solutions needs to be tethered to a powerful computer for render high quality virtual content. As a result, the full setup ends up being too bulky and obtrusive to be used in everyday tasks and specially during surgery. It also takes away the freedom of movement beyond a few feet. In an actual surgical theater where multiple surgeons will be working together in proximity having such an obstructive setup is impractical.

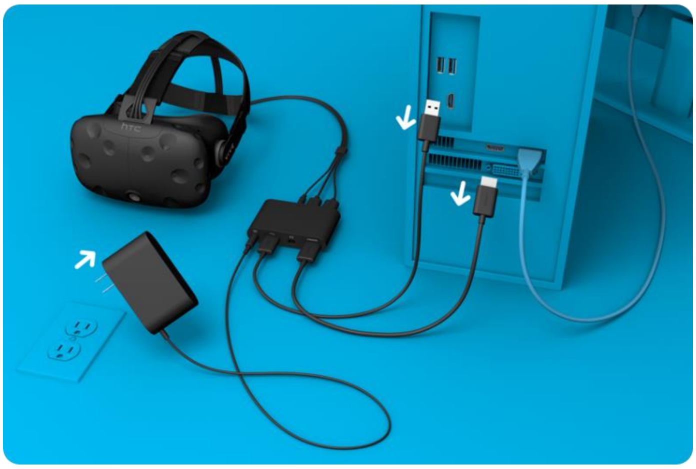


Figure 3: A conventional tethered VR Headset setup requires a power supply connection, a data streaming connection with a powerful computer and a hub that coordinates all these connections. The VR headset completely blocks the view of the real world.

Secondly, most AR-VR headsets are video see-through head-mounted display (VST-HMD). VST-HMD blocks the user's direct view of the real world. Instead, a video capture (with overlaid virtual content) of the real-world is show to the user. A failure of the display technology thus causes total vision loss of both the real world and the virtual contents. For critical surgical applications this is unacceptable. The solution is an optical see-through head-mounted display (OST-HMD). With OST-HMD, computer generated graphics can be presented to the users while they are still able to see the real world through a semi-transparent display.

Figure 4: A conventional AR headset that utilizes a video see through display. The user perceives the real-world through the video streams captured by the front facing cameras in the headset.

OST-HMDs have various advantages over VST-HMDs, for instance, they are fail-safe for critical medical procedures [13]. Even if the virtual graphics fail to deliver, the user is still able to perform the procedure normally. To use HMDs for interventional assistance, it is critical to have this fail-safe. This fail-safe is required for any practical augmented reality-based application in any industry, not just for applications in healthcare. This concept gave birth to a new technology named Mixed Reality which is described in the next section.

2.3 MIXED REALITY (MR)

Mixed reality is the next general-purpose computer flowed by mainframes, PCs and smartphones [14]. Powerful technology will no longer be in our pockets or at our desks. Through MR, it will be integrated with the human self. MR replaces the 2D screen-bound experiences by providing natural and intuitive human interactions with virtual data in our everyday life.

The term Mixed Reality was introduced in a 1994 paper by Paul Milgram and Fumio Kishino [15]. Their paper explored the concept of a *virtuality continuum* and the taxonomy of visual displays. Since then, the application of Mixed Reality has gone beyond displays to include [14]:

- Environmental understanding via spatial mapping and anchors.
- Human understanding: hand-tracking, eye-tracking, and speech input.
- Spatial sound.
- Locations and positioning in both physical and virtual spaces.
- Collaboration on 3D assets in mixed reality spaces.

Figure 5: Microsoft Hololens-2 [16] an optical see-through mixed reality headset. User always has direct natural view of the world. The AR contents are projected to the user's eyes via lasers.

The HoloLens 2 from Microsoft is currently the only fully untethered and most advanced optical-see-through mixed-reality headset available. Capabilities include hand tracking, built-in voice commands, eye tracking, spatial mapping and a larger field of view. These advancements allow precise and intuitive interaction with virtual contents. Hololens-2 users can interact with virtual objects as if they are real physical objects. They can touch, grasp, or move them with bare hands. The fully articulated hand tracking allows for this capability [17].

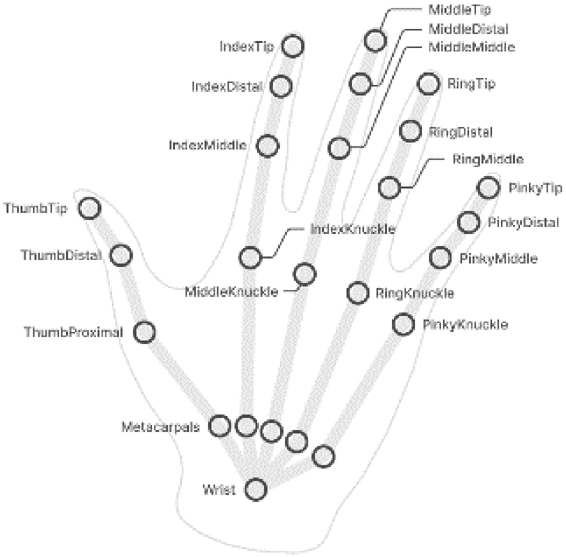


Figure 6: Hand key-points tracked by Hololens-2 in 3D [17].

Hololens-2 also allows for remote collaboration [18]. Using Microsoft HoloLens people from different remote locations can view and interact with the same simulated objects. They can even bring their own holographic projection or digital avatars into the remote meeting to allow for more intuitive collaboration.

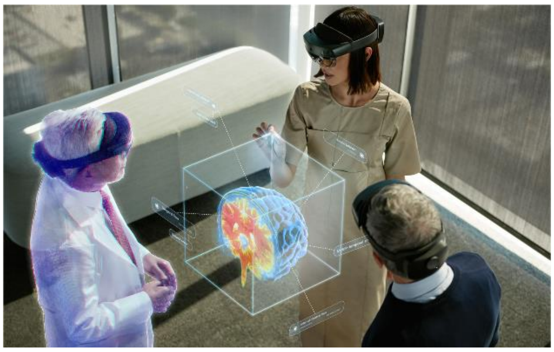


Figure 7: Through Hololens-2 remote collaboration can be done in 3D [18].

Due to its usefulness, the medical industry is in a race to incorporate Hololens-2 and mixed reality capabilities in various products that were not practical to implement via traditional AR-VR headsets. Hololens is particularly attractive for solutions surrounding intraoperative visualization and collaboration, as traditional AR-VR headsets are impractical for such usages.

2.4 MEDICAL MIXED REALITY (MMR)

2.4.1 Introduction

The goal of medical mixed reality is to allow surgeons to visualize patient data directly onto the patient's body. The data may include virtual surgical planning, CT/MRI DICOM data, interventional plans, or other measurements. Through MMR physicians are enabled to look inside the patient's body and view the internal anatomy in its actual position. This helps to planning treatment, improve diagnosis, disease assessment and procedure guidance.

Researchers and industry leaders are racing to develop intraoperative MMR solutions. For instance, the Stanford University, has recently founded the Stanford Medical Mixed Reality (SMMR) [19]. Their goal is to bring together leading researchers, clinicians, and engineers to share work, exchange ideas, collaborate and discuss opportunities and challenges. Similarly, a few industry leaders are also in a rush to develop MMR products [12], [20]–[22].

2.4.2 Challenges

The introduction of mixed reality has opened the doors for surgeons to visualize VSP intraoperatively for the first time. However, several key challenges remain to be solved. The first challenge is to design an intuitive and minimalistic user interface (UI) for surgeons to operate the MMR apps during surgery without any obstruction. The surgeons will always have their hands busy during surgery. The UI can't make surgeons disengage their hands from work even for a moment. Next, the UI can't force the surgeons to look away from the patient to navigate the UI. If the UI is like regular UI – ones with maze of buttons and menus – then the surgeon will have to take his focus off the patient and concentrate on navigating the UI. Thus, the first practical challenge is to figure out a user interaction system that allows for non-distracting and seamless use of the mixed reality solutions during surgery.

Developing accurate and easy to use registration methods for virtual anatomy models is the second challenge. It must be ensured that the holograms for the anatomy are perfectly aligned with their physical location in the patient’s body. Intelligent but computationally inexpensive solutions must be implemented in MMR headsets to achieve this.

The third challenge is to develop useful visualizations of the CT scan data. Mixed reality is a very powerful 3D visualization tool that can be used intra operatively. The data can be shown in many ways. Among them, the ones that are most useful to surgeons must be identified and implemented properly.

Finally, finding pragmatic uses of MMR in actual surgical cases is the biggest challenge. To meet this challenge experienced surgeons must contribute ideas on how MMR can be useful to them. Engineers must develop solutions that meet these needs first. A lot of sophisticated or beautiful looking solutions can be developed by an engineer. However, if those solutions are not really needed by actual surgeons or residents then those solutions are not solving any real problems. Finding which medical mixed reality problems to solve is a problem itself.

2.5 ORIGINAL CONTRIBUTIONS

In this thesis report, I present my research to address the above-mentioned challenges in MMR. Over the past few years, I and Dr. Christian El Amm, Professor at OU-medicine, have developed the IntraOpVSP system. I outline the original contributions of my research in this section.

A voice command based minimalistic UI have been developed for surgeons to avoid getting distracted during the use of IntraOpVSP. Three different registration methods have been implemented for performing registration of CT scan data with the patient’s body. Each of the methods have proved to be very useful in different circumstances during surgery. Four different modalities for intraoperative visualization of 3D VSP data have been implemented. The visualizations were found to be convenient not only for intraoperative uses but also for educational and collaborative planning purposes as well. The IntraOpVSP system has been used in over 35 surgeries at OU-medicine by the surgeons and residents. Finally, at the end of this thesis report, I present some useful ways they have made use of IntraOpVSP.

2.6 TECHNOLOGY DEPENDENCY

2.6.1 Unity

Unity3D is a game engine that allows for rapid Game development for various platforms [23]. It also allows for AR/VR/MR application development. The scripting language of choice is C# for this game engine. In this research, I use Unity3D to develop all applications.

2.6.2 Mixed Reality Toolkit (MRTK)

MRTK-Unity is a Microsoft-driven project that provides a set of components and features, used to accelerate cross-platform MR app development in Unity [24]. MRTK provides the cross-platform input system and building blocks for spatial interactions and UI. Enables rapid prototyping via in-editor simulation that allows you to see changes immediately. Operates as an extensible framework that provides developers the ability to swap out core components.

2.6.3 Virtual Surgical Planning

Virtual Surgical Planning is the process of viewing and converting the CT or MRI imaging data into several clean 3D models of different parts of the patient anatomy, with which surgeons plan their surgical steps virtually [25]. For example, from thousands of CT scan images of the skull a single 3D model of the skull can be created. Once the virtual 3D models are ready, surgeons can then plan their procedure through various VSP processing software. In IntraOpVSP, all 3D models were produced by surgeons through VSP.

3 VOICE COMMANDS

3.1 INTRODUCTION

We decided to operate the IntraOpVSP system primarily using voice commands. Usually when making MR apps it is customary to have a UI that consists of a lot of buttons or slides. However, it is our understanding that having such traditional UI is impractical for surgeons when they are performing surgery. Surgeons will always have their hands busy and engaged in the surgery. To access a specific data, if they must waste any time navigating through the maze of buttons then that defeats the purpose of IntraOpVSP. The purpose of IntraOpVSP is to become a seamless assistant to surgeons for viewing data intraoperatively. Thus, we decided that all navigations would be done through voice commands.

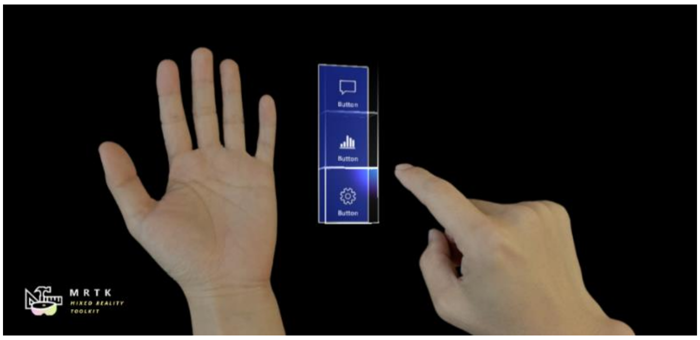
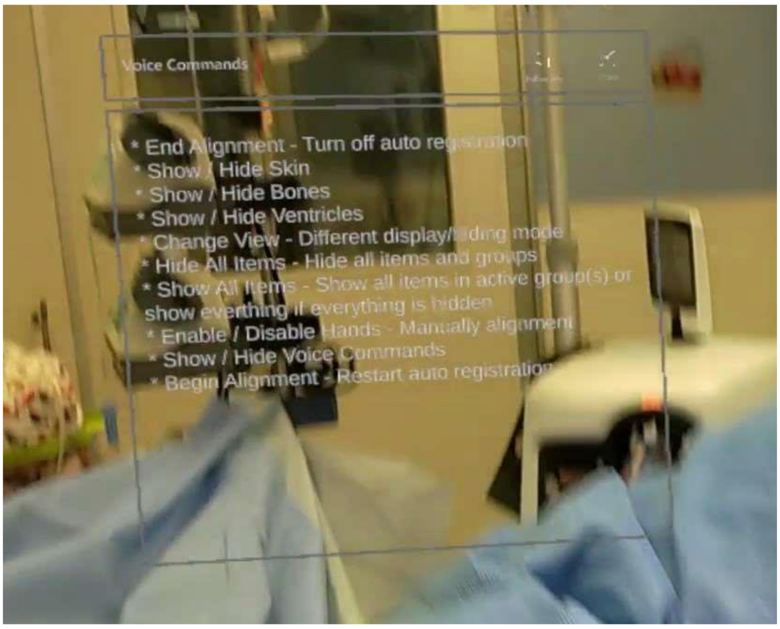


Figure 8: A sample MRTK provided UI [26]. To interact with such UI the surgeon must completely disengage their hands from their sensitive tasks.

3.2 USAGE

A "Help Menu" text box is constantly displayed "virtually" through the Hololens 2® eye display. The user can find the text box by looking straight ahead. The virtual text box will follow the users as they ambulate around the workspace, so it is always easily findable. The "Help Menu" text box contains a list of the voice commands and the "brief names" of the models, selected during the pre-operative VSP session.

To use a voice command, the user just must utter the predefined trigger words. IntraOpVSP will recognize the voice command and initiate associated callback functions. After each voice command, the Hololens 2® external speakers will play a "chime" tone to confirm execution. A "text box" with a relevant message also appears in the holographic display for 5 seconds confirming execution of the voice command. This feedback helps user understand that the voice command was successfully recognized. Voice Commands are used to display the preselected 3D models, using their “brief names”. For example, by saying “Show Skin” the user can bring up patient’s skin hologram.


Voice Commands

- End Alignment - Turn off auto registration
- Show / Hide Skin
- Show / Hide Bones
- Show / Hide Ventricles
- Change View - Different display/hiding mode
- Hide All Items - Hide all items and groups
- Show All Items - Show all items in active group(s) or show everthing if everything is hidden
- Enable / Disable Hands - Manually alignment
- Show / Hide Voice Commands
- Begin Alignment - Restart auto registration

Figure 9: A “Help Menu” containing the voice command list and their description. The virtual slate always follows the user while not obstructing the users view of the surgical field. The menu can be turned off and on at will.

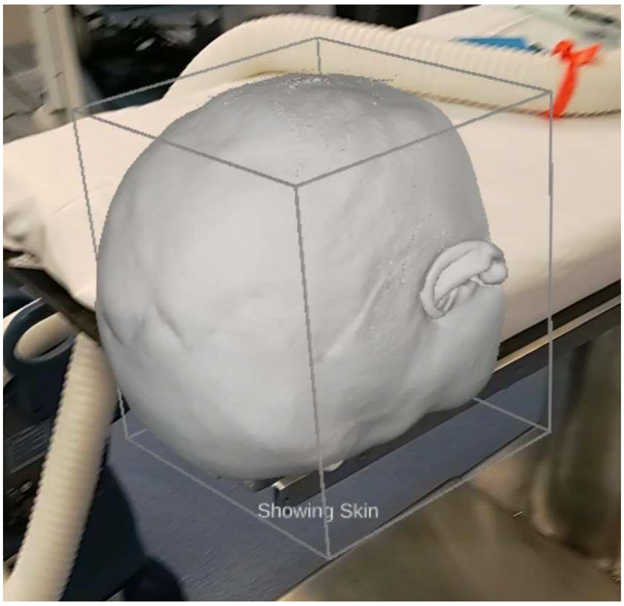

Showing Skin

Figure 10: Upon recognition of the “show skin” command a hologram of the subject’s skin is shown along with the text feedback in HoloLens.

Another common example is “Show Bone” to show the calvarium or other osseous structures. Conversely, saying “Hide Skin” will remove the displayed skin layer, and allow visualization of the underlying structures.

If the display becomes too cluttered, saying “Hide All Items” will remove all displayed models. One can hide the “Help Menu” by saying “Hide Voice Commands”. One can show it again by saying “Show Voice Commands”. The list of voice commands and the model “brief names” can be customized during the preoperative VSP session.

3.3 IMPLEMENTATION DETAILS

The list of voice commands and “brief names” is curated to avoid interference with commonly used words during surgery and avoid accidental activation. This voice command feature is added to each application using the MRTK [27]. There are three voice command recognition confidence levels: low, medium, and high. For our applications we have found the medium recognition confidence level to be most useful emphatically.

Depending on the type of case (e.g. neurosurgery, liver surgery) the set of voice commands that needs to be present in the app is different. To address this, I first save the case specific voice commands in separate files for each case. Depending on the type/case the corresponding voice command list is loaded and integrated with the app.

The speech input handler script (provided by MRTK) waits and listens for trigger words (voice commands). Upon reception of a voice command the associated callback is called that resides in a global voice command responder class. To handle all the callbacks for each voice command, I have implanted a global voice command responder object. This extra step is done, to implement additional custom features like custom sound feedbacks, text feedbacks and making sure multiple callbacks don’t conflict each other.

3.4 DISCUSSION

The recognition of voice commands by Hololens-2 is almost instantaneous without any noticeable delay. It works in crowded or noisy environment. However, if the user of the app is constantly talking (for instance while doing a presentation) and without taking a pause if the user utters a voice command, the Hololens-2 doesn’t pick up on the voice command. For the device to recognize voice commands properly it is necessary for the user to stop talking fist, pause for about a second then utter the voice command.

We have been using the app in this “voice commands only” principle for about a year. So far, the participants at OU-medicine didn’t have any problem using them while performing surgeries. We believe, that is because all the keywords were chosen to be very intuitive while not being commonly used in a conversation during surgery. In the future, I plan to incorporate some optional hand-interactable menus/buttons for specialized tasks. I also want to incorporate eye tracking for UI navigation in the future versions of IntraOpVSP.

Currently we are devising methods for through evaluation of our voice recognition system’s performance and collecting associated data through experiments. The results are planned to be published in 2022.

4 MANUAL REGISTRATION

4.1 INTRODUCTION

To visualize CT scan data aligned with the subject’s body we need to perform registration first. The most primitive way of doing this is via the manual manipulation of the simulated object using bare hands in Hololens-2. In this section, I elaborate on how I integrated this feature in IntraOpVSP for manual registration of CT scan data with respect to the subject’s body. In IntraOpVSP, manual manipulation enables freehand spatial movement of the hologram, isolated linear movement, or angular rotation movement.

4.2 USAGE

Manual manipulation capability of an AR object is activated by using the voice command “Enable Hands”. A text and sound feedback are generated assuring the user that the voice command was executed. A wireframe box appears around the hologram. The hologram is now movable.

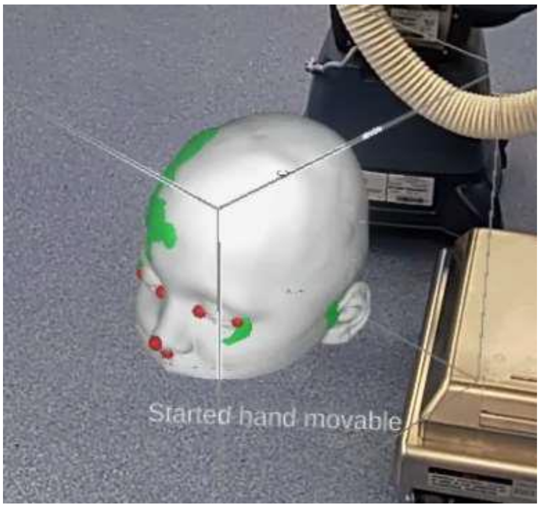


Figure 11: Upon recognition of the "Enable Hands" voice command a bounding box appears around the 3D model indicating that the object is now manipulatable.

The next step is to manipulate the 3D box that appeared around the target object. A holographic virtual ray appears to extend outwards from the operator's wrist. The virtual ray is used to point at objects, analogous to a computer mouse pointer. The operator can use a thumb-index pinch and release maneuver, analogous to a computer mouse "click". The operator can use a thumb-index pinch and hold maneuver, analogous to a computer mouse "click-and-drag". Releasing the thumb-index pinch will end the "click-and-drag".

For freehand manipulation, the user needs to point virtual ray at the hologram. After doing so a light blue iridescent color will appear over the hologram, indicating interaction with the virtual ray. This is analogous to hovering a mouse pointer over an UI item in a web browser. The operator then uses a thumb-index pinch and hold maneuver to move/rotate the object. The hologram will now follow the movement of the user's hand and wrist in 3D Space ("6 Degrees of Freedom"). By moving the hand back and forth, right and left the AR object is moved correspondingly in space. Similarly rotating wrist in different direction translates to rotation of the AR object where the pivot point would be the user's hand. Releasing the thumb-index pinch ends the "hold" on the object and places the hologram in its new location.

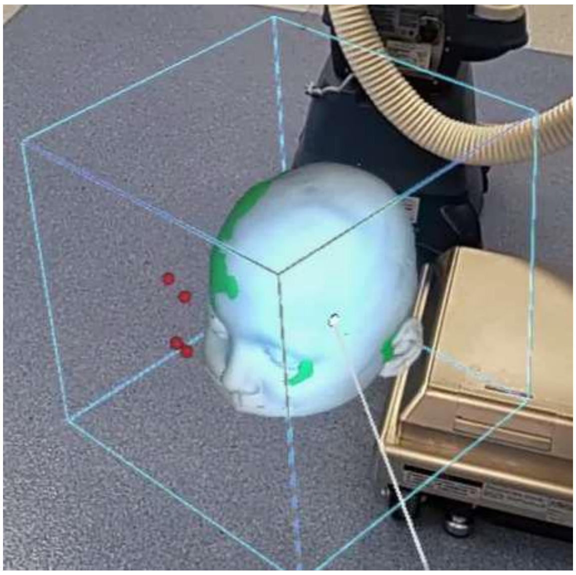


Figure 12: The 3D box can be moved and rotated by holding it via the hand-ray.

The movement strategy described so far is a coarse method of manipulating the 3D object. It not intended for fine movements/calibration of the 3D objects. The goal of this freehand manipulation is to make big non-precise movements to get the 3D objects roughly where the user wants them to be.

For precise linear or angular movements, fine-tuning knobs are implemented surrounding the 3D bounding box. The knobs are usually hidden to not obstruct the view of the 3D object all the time. To reveal them the user needs to hover the hand-ray pointer over them. Two types of knobs are available: linear movement knobs and angular movement knobs. There are 6 linear movement knobs paced at the center of each of the faces (2D planes) making up the 3D bounding box. The rotation knobs are placed at the center of each straight line making up the 3D box. The pivot point for all movements in this fine-tuning mode is set at the volumetric center of the 3D bounding box.

To perform the linear (1 Degree of Freedom) movement, the user must pinch-and-hold the knob, then move his/her hand. The movement of the hand will be translated into fine movement of the 3D object along the 3D line going through the pivot point (3D box’s center) and the knob the user is currently using. Using these knobs, the user can perform fine linear movements of the object in 3D space. Once the user lets go of the “hold” on a knob the object stops moving and remains stable at its new location.

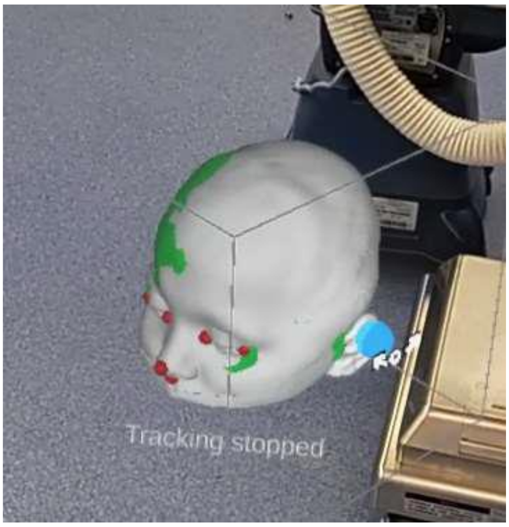


Figure 13: Hovering the hand-ray at the center of one of planes of the 3D bounding box reveals the linear movement knob. The knob allows for 1D linear movement.

For angular movement, the user points the hand ray at an edge of the wireframe box. A small iridescent blue tab appears at the center of that straight line. Using a thumb-index pinch and hold maneuver the rotation can be started. The hologram will only rotate with respect to the straight line that passes through the pivot point and is parallel to the line on which the current rotation-knob sits on. The pivot point is the center of the 3D bounding box. This translates into a one degree of freedom angular movement. Finally, releasing the thumb-index pinch places the hologram in its new location.

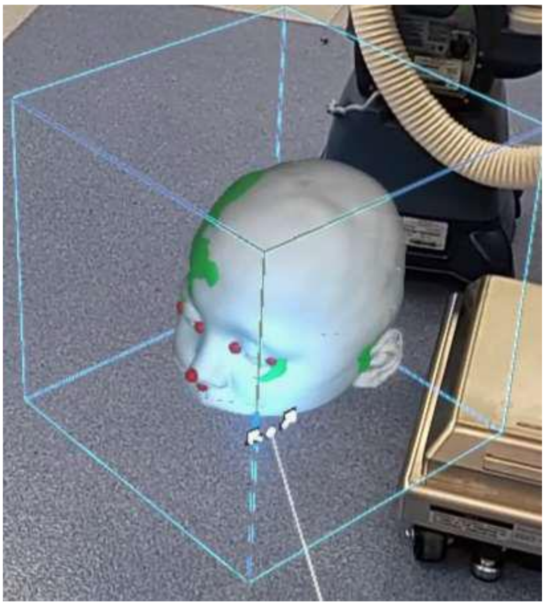


Figure 14: Hovering the hand-ray at the center of one of straight lines of the 3D bounding box reveals the angular movement knob. The knob allows for fine rotation.

The free-hand movement is designed for doing coarse movements of the virtual objects. The 6 linear movement knobs and 12 angular movements knobs are designed to allow for fine movements of 3D objects. All these features facilitate manual registration of CT scan data with respect to the actual subject's body. Once the IntraOpVSP user is satisfied with the registration, he/she can stop this feature by using the voice command "Disable Hands". The wireframe box around the hologram disappears. The hologram is no longer movable.

4.3 IMPLEMENTATION DETAILS

MRTK's Object Manipulator and Bounds Control classes were used to implement this manual manipulation feature [28], [29]. The Object Manipulator script makes an object movable, scalable, and rotatable using one or two hands [28]. The Object Manipulator script can be configured to control how the object will respond to various inputs. This script was used for implementing the free-hand coarse movement capability.

The Bounds Control script provides basic functionality for transforming objects in mixed reality [29]. A bounds control will show a box around the hologram to indicate that it can be interacted with. Handles on the corners and edges of the box allow scaling, rotating, or translating the object. The bounds control also reacts to user input. On HoloLens 2, for example, the bounds control responds to finger proximity, providing visual feedback to help perceive the distance from the object. All interactions and visuals can be easily customized. The fine-tuning feature was implemented using this script. In IntraOpVSP, the scaling is turned off. That is because we don't want to change the scale of the CT scan data.

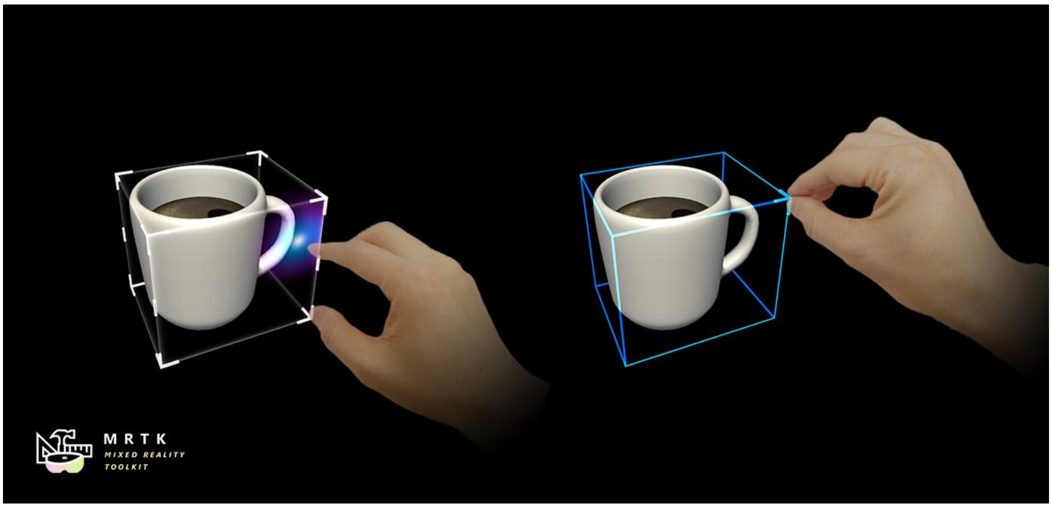
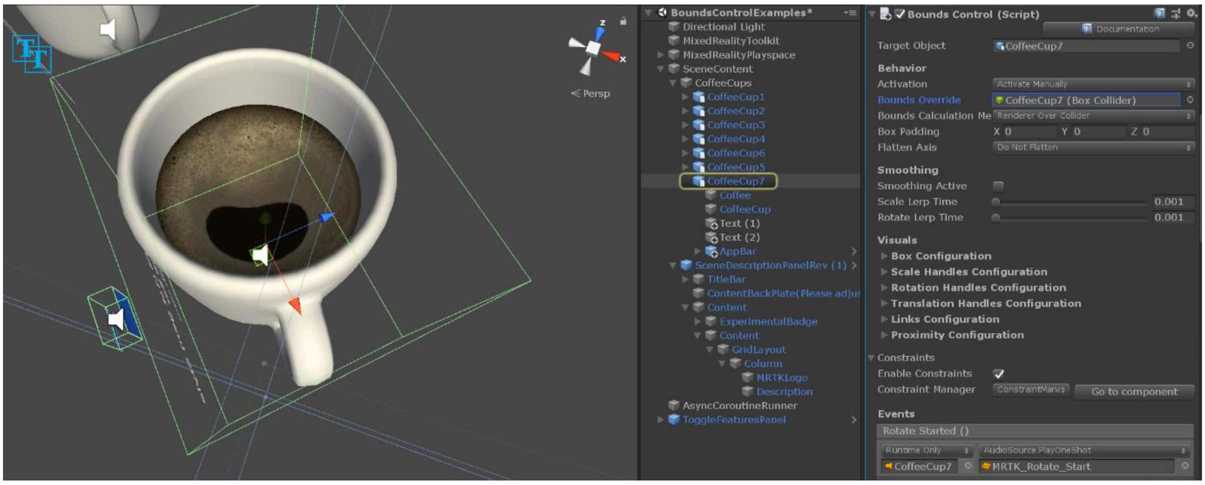


Figure 15: By using the bounds control class from MRTK objects become manipulatable. They can then be moved, rotated or scaled using near or far interaction.



Target Object: CoffeeCup7
Activation: Activate Manually
Bounds Override: CoffeeCup7 (Box Collider)
Bounds Calculation Me: Renderer Over Collider
Box Padding: X 0 Y 0 Z 0
Flatten Axis: Do Not Flatten
Smoothing Active: ☐
Scale Lerp Time: 0.001
Rotate Lerp Time: 0.001
Enable Constraints: ☑
Constraint Manager: ConstraintMan
Rotate Started ()
Runtime Only — AudioSource.PlayOneShot
CoffeeCup7 — MRTK_Rotate_Start

Figure 16: The collusion box for a sample 3D object is being shown in green. The size of this collusion box defines the size of the bounding box the bounds control script creates. Care must be taken to ensure that the green collusion box encompasses the entire 3D

While making a patient specific IntraOpVSP app, attention must be given to set the bounding box size around the CT scan. The Bounds Control script doesn't automatically set the bounding box size to accommodate for different CT scan sizes. This bounding box size must be manually set to the desired amount in the Unity editor before deploying the app. In the future a more automated solution would be implemented.

4.4 DISCUSSION

The manual manipulation is very intuitive and easy to use. Within a few minutes an approximate registration can be done by the user through this feature. We are currently collecting data to verify this manual registration system's precision. Though this system is very natural to use, it is an inconvenience as the user must spend several minutes to make sure the registration is as accurate as possible. Thus, a more automated pipeline is needed. An alternative automated registration method is presented next.

5 AUTOMATIC REGISTRATION

5.1 INTRODUCTION

To visualize CT scan data perfectly aligned with the subject's body we first need to complete registration between the virtual data and the subject's real physical body. This can be done via manual manipulation as mentioned in the previous section. However, it is not always convenient. Thus, in IntraOpVSP, I have implanted an automatic alignment registration method using computer vision.

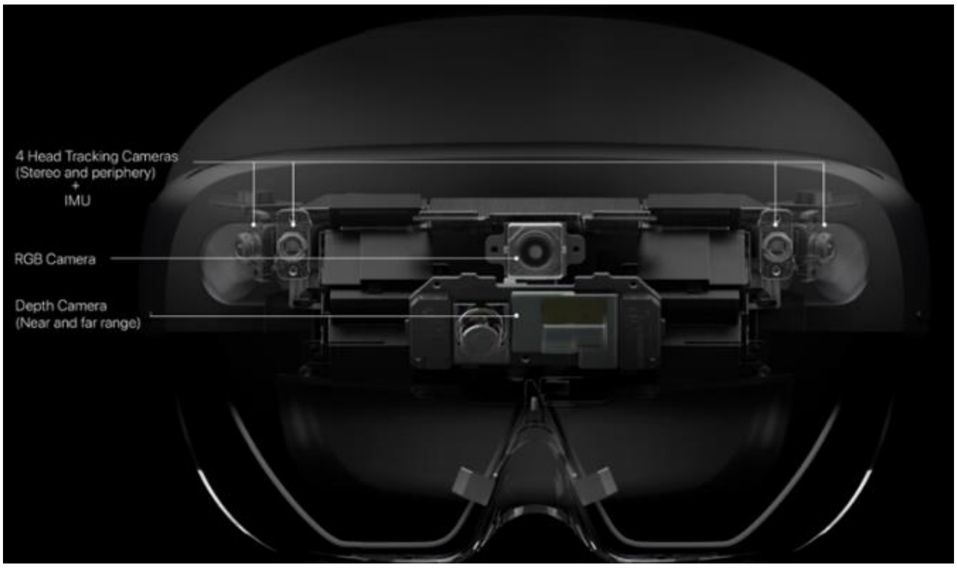


Figure 17: Outline of different sensors of Hololens-2 [30]. IntraOpVSP currently only use the RGB camera, also known as the photo-video (PV) camera [31].

5.2 USAGE

Automatic registration (alignment) calculates the 6 DoF pose of the patient's target physical anatomy and then overlays the virtual holograms accordingly. Following are the steps to complete this automatic alignment.

The user activates the automatic registration pipeline by saying "Begin Alignment". The voice command is instantaneously recognized by IntraOpVSP. An external LED light on the Hololens visor turns on to indicate activation of the automatic registration pipeline. IntraOpVSP uses the external sensors of the Hololens to locate the patient's landmarks that were selected during the pre-operative VSP session. Once landmarks are identified, IntraOpVSP will spatially align the virtual CT holograms with the patient's physical anatomy. A hologram of the patient is visible to the operator and will progressively move to align with the patient's physical anatomy. After 15-30 seconds, the hologram will stop moving, indicating maximum alignment. By saying "End Alignment" the registration pipeline is terminated. After 1-2 seconds, IntraOpVSP finalizes alignment and positions all the holograms in a new fixed location overlaying the patient's physical anatomy. The external LED light turns off.

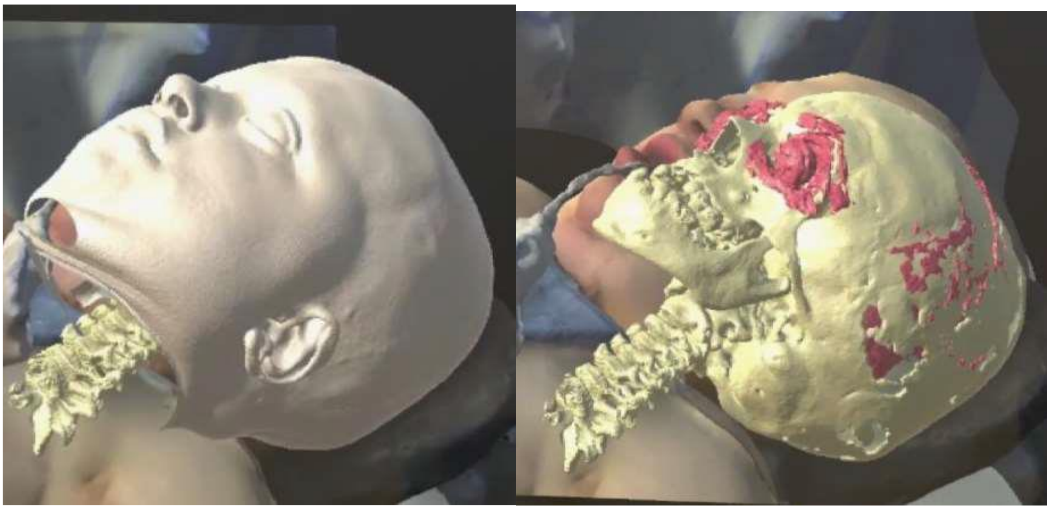


Figure 18: Automatic head 3D model alignment result after using the command "End Alignment".

5.3 IMPLEMENTATION DETAILS

In this section an example of aligning a head's CT scan with a patient's actual head will be considered as the running example. The 6 degrees-of-freedom pose estimation of the head is done via the Perspective-n-Point problem algorithm. In this problem the goal is to find the 6-DoF pose of an object with respect to a calibrated camera when we know the locations of n 3D points on the object (CT scan mesh) and the corresponding 2D projections in the captured image. The algorithm is described in detailed in this article [32].

In IntraOpVSP the images are captured using the in-built photo-video (PV) camera [31]. The camera intrinsic matrix is retrieved from captured frame's meta data. Six predetermined landmarks are used for the PnP problem. The 3D locations of these landmarks are measured from the CT scan mesh data. The 2D locations of these landmarks are estimated using OpenCV and Dlib built-in functions by operating on the captured frame. I elaborate on the individual steps more below.

IntraOpVSP uses 6 preselected points for the Solve PnP problem. These are both corners of both the eyes, the nose tip and the subnasal point. As a prerequisite, we first need the 3D locations of these six points measured from the CT scan mesh data. This is done via a 3D data processing software like MeshLab or Amira3D [33], [34]. Landmarks are measured and saved as a text file. These text files are uploaded to the IntraOpVSP system. I have implemented a script that reads these landmark locations from the text file and then loads them into unity. At this stage we have both are prerequisite data requirement fulfilled, which are the camera intrinsic matrix and the 3D locations of the landmarks.

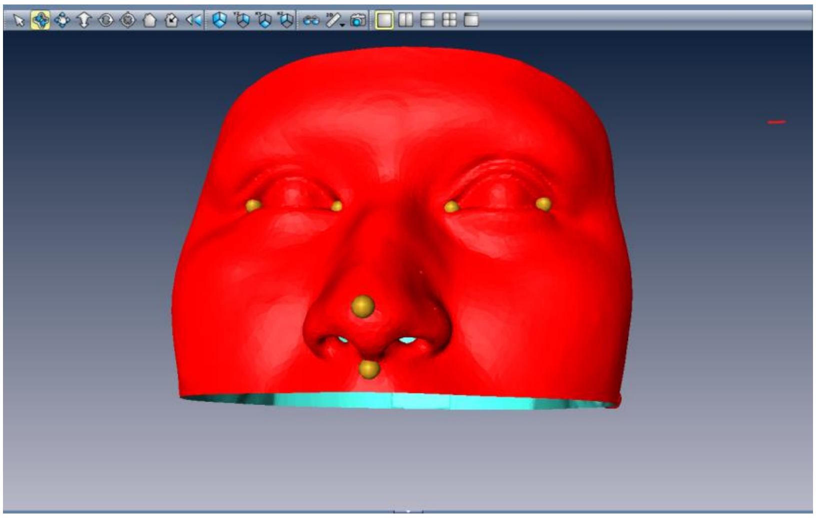


Figure 19: The 3D locations of the six predefined face landmarks are being measured using Amira3D. These measurements are called "object points" and are saved as a text file in IntraOpVSP.

Next step is to detect these 6 landmarks in the 2D captured image. IntraOpVSP starts with capturing a frame from the photo-video (PV) camera of Hololens-2 [31]. Then Dlib's face detector is used to detect the bounding box where the face is in the frame [35]. Once the face's 2D location is known, that part of image is cropped. The cropped image is then used as input to Dlib's face landmark detector, also known as shape predictor. Our trained Dlib model detects 6 preselected feature points on the face. Upon successful detection, the 2D location of these 6 landmarks are stored.

At this stage we have all the elements we need to use OpenCV's solve-PnP function [36]. These are the 2D and 3D locations of the same six preselected landmarks, and the camera calibration matrix. The function then estimates the 6 DoF pose of the physical face with respect to the camera. In Hololens-2, I also know the camera extrinsic matrix of the camera from Unity3D. Using that matrix, I calculate the pose of the face/object in the Unity3D's world coordinate space.

Now that I have the pose of the head determined, the next step is to move the CT scan mesh according to that pose. This is done using Unity3D's built-in object transformation functions. However, I have also added a low-pass-filter (lerp) in unity to smooth the movement of the virtual content / CT scan mesh data. At the end of this step the Hololens user sees the CT scan data being perfectly aligned with the actual physical head of the subject. Until the user commands the Hololens to stop pose estimation, IntraOpVSP keeps on going through the same loop and updates the pose of the head on every frame it can process.

Although in this example I have primarily talked about pose estimation of the head, this same algorithm can be adopted for any other part of the body where sufficient distinct visual features are available.

5.4 DISCUSSION

The automatic pose estimation pipeline works at around 5 FPS. The pose estimation pipeline also influences the rendering. When this auto-registration pipeline is not running, I get around 50-60 FPS of rendering in Hololens. However, when running the automatic pose estimation, the rendering FPS comes down to around 30 FPS. This drop in rendering FPS makes the IntraOpVSP a little uncomfortable until the tracking is done, and the camera is stopped. This is a point where optimization is needed in the future versions of IntraOpVSP.

The amount of time it takes to get the initial lock on pose estimation after the “Begin Alignment” command is given, is another major inconvenience. Currently it takes around 30 seconds to get the initial lock on head pose for the first time “Begin Alignment” command is used. In any subsequent usage of this command to reposition the virtual content, it takes around 10 seconds. More optimization is needed to bring, the time it takes between recognition of the “Begin Alignment” command and the completion of accurate alignment, to around five seconds. Currently I am experimenting with an alternative and much simpler pipeline to achieve that time-limit.

We are conducting experiments to verify the accuracy of the pose estimation. However, it is proving to be a very difficult challenge. There are no established protocol or practice on how to measure the alignment error of a hologram with respect to the real object that the hologram represents. Our initial experiments reveal that the average error is less than 5mm.

Though automatic registration may be very convenient for surgeons, it does not work in challenging registration cases. In such circumstance a semi-automatic, manual landmark placement-based registration solution is needed. In the following section I elaborate this method.

6 MANUAL LANDMARK PLACEMENT BASED REGISTRATION

6.1 INTRODUCTION

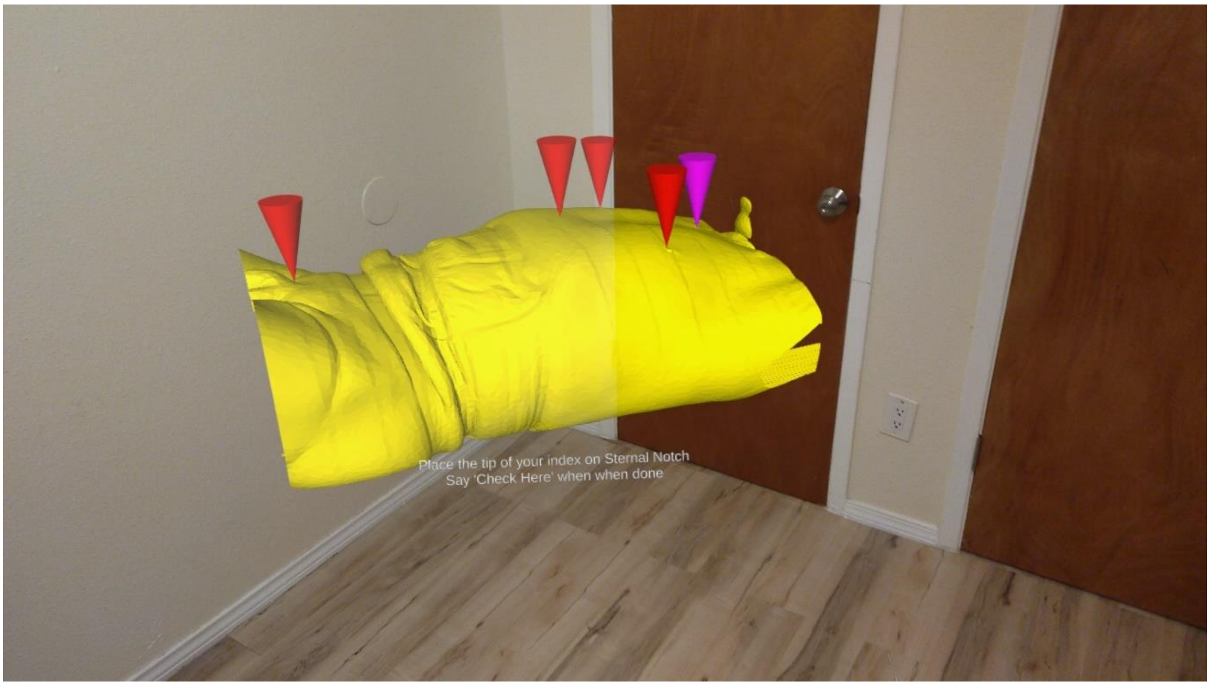
The automatic registration fails if the algorithm can't identify the landmarks automatically in the captured 2D image. For instance, if the patient is lying face down on the operating table for a surgery in the back of his head, then Hololens can't capture an image of the face to automatically detect facial landmarks. Consequentially the auto-registration algorithm can't proceed. This condition also holds true even if the patient is lying face-up, but the face is heavily deformed due to some severe accident and injuries sustained on the face.

Similarly, there are cases when the surgery is about to be performed in an area of the body where there are not much visual features available. For instance, a liver surgery. In this section the running example will be for this liver surgery case. The body of the patient will most definitely be covered by some form of cloth during the surgery. As a result, even if we had trained an algorithm to detect the 2D landmarks of a human chest and stomach area, the algorithm won't be able to work. A completely vision-based registration is therefore impractical.

I have devised a manual 3D landmark placement-based registration to handle these difficult registration cases. Simply put, instead of the computer vision algorithm trying to detect the 2D locations of the body landmarks from the captured images, the Hololens user will manually point out the 3D locations of those body landmarks directly. IntraOpVSP will calculate the pose of the human body based off that data and align the CT scans accordingly.

6.2 USAGE

The user must use the designated voice command to start this alignment process. The voice command is "Begin Alignment". In practice when we need this modality of registration in a surgery/case, we usually never need to also have the auto-registration feature present in IntraOpVSP. Thus, having the same keyword "Begin Alignment" doesn't cause any conflict. The auto-registration feature in such cases is always turned off. Upon start of alignment pipeline, a few pins/markers will appear on the subject's CT scan data as shown below.



Place the tip of your index on Sternal Notch
Say 'Check Here' when when done

Figure 20: Upon recognition of the "Begin Alignment" command, landmark pins have appeared on top of the subject's 3D scan model. The yellow region represents the CT scan of the skin. Text feedback containing instructions for the next step is also shown to the user.

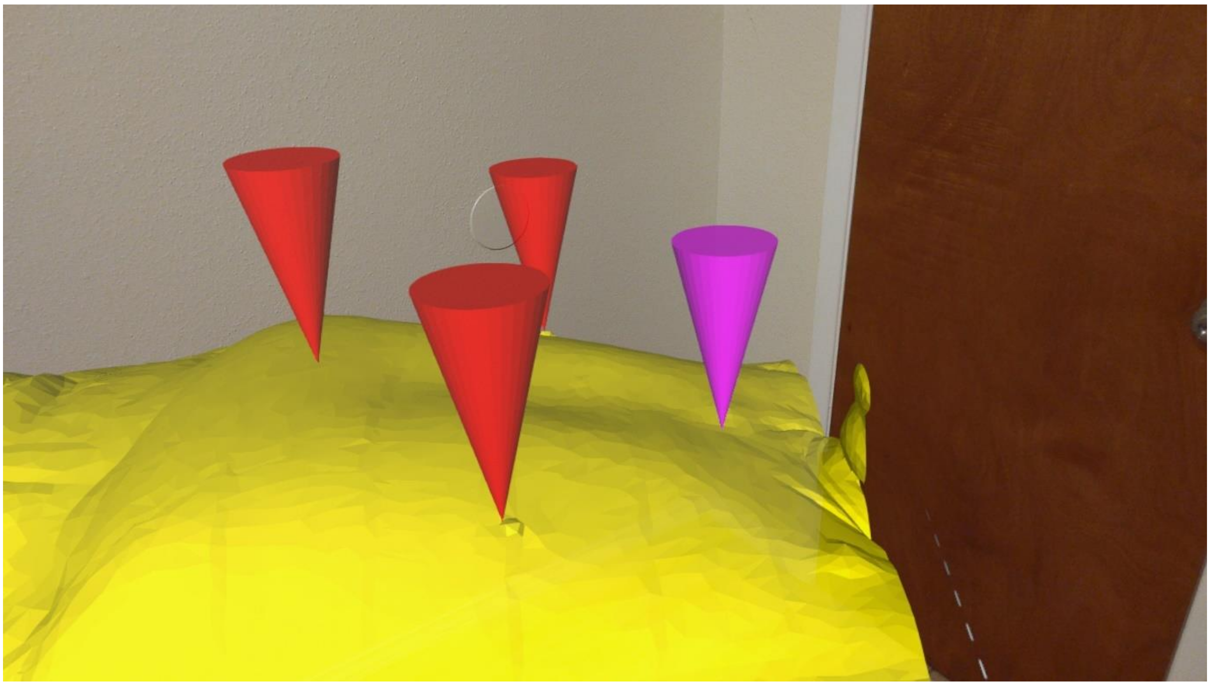


Figure 21: By default, the landmark pins are placed on predefined anatomic locations. These locations present where each of these pins should be moved to on the actual patient's body. The first landmark-pin to be placed is highlighted with purple color.

The pins represent the markers the user needs to put on the subject's body. The pins' default position on the CT scan data represents where these pins should end up in the actual physical body of the subject. Tooltips indicating the anatomical location where each of the pins should end up is also incorporated [37]. To view this tooltip, the user must touch then pins. A tooltip will pop up for a few seconds showing the name of the anatomical position where the corresponding pin should be placed.

The next step is to place the pins on the required physical locations. Upon activation of the alignment, a text message will also be constantly shown to the user. The text will ask to put the first pin at a specific anatomical location. Among all the pins, that next pin will be highlighted to the user via assigning it a different color than the rest of the pins. To place this highlighted pin on the patient's body the user needs to put his right hand's index finger on the designed anatomical spot. If touching the physical spot is not an option, the user can just bring his finger close to that spot. Once the finger is in place, the user uses another voice command to teleport the highlighted pin on top of his index finger. As a result, now that highlighted pin ends up being positioned on the subject's physical body and at the correct anatomical location.

Figure 22: Upon recognition of "Check Here" feature the highlighted landmark-pin is placed at the tip of the index finger of the operator. Upon placement, the pin is no longer highlighted. Upon touching any of the pins, a tooltip poops up showing the name of the destination anatomic location. This pin is supposed to be placed at sternal notch.

After the first pin has been placed using voice command the next pin is highlighted and the text feedback also changes helping the user determine which pin to place next. For instance, in the image below the text feedback is saying to place the currently highlighted landmark pin to Xyphoid. Just like before, the user again places his finger at the correct anatomic location and uses voice command “Check Here” to teleport the second pin. In this manner the user continues placing all the pins on the subject’s body. After all the pins have been placed, text feedback is shown to the user indicating that all pin placements are done and that IntraOpVSP is ready to calculate the pose of the body.

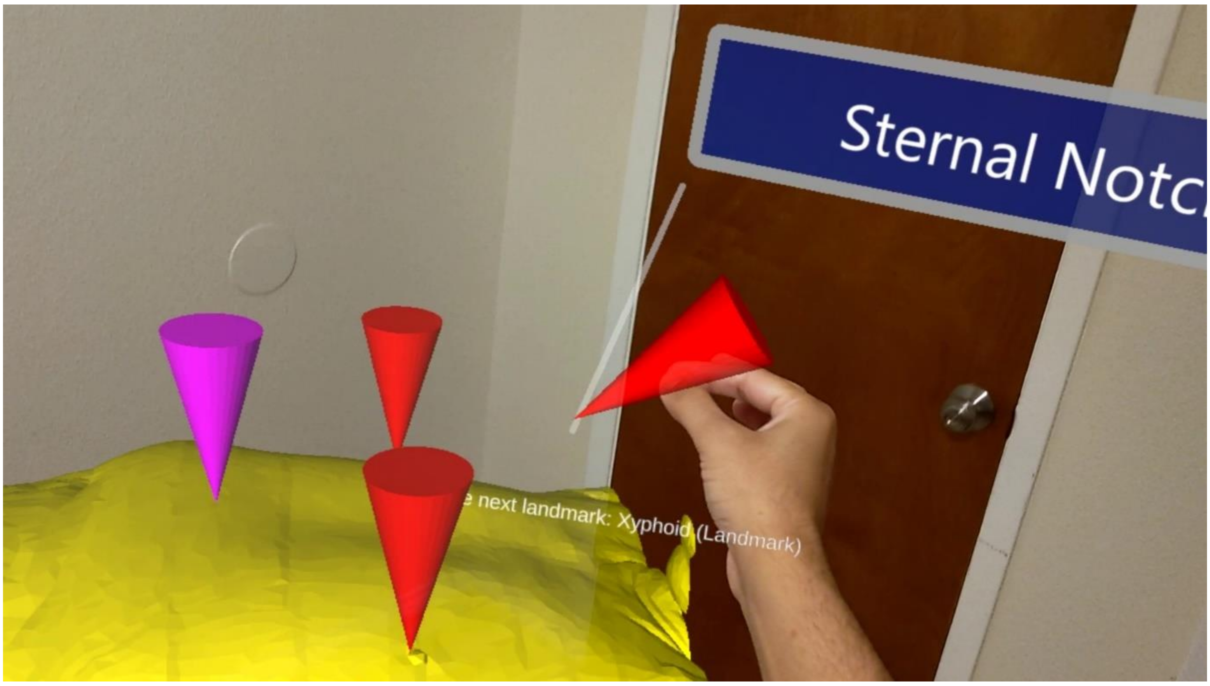


Figure 23: The user can simply grab the land-mark pins with hand and put them at their desired anatomic destinations. This helps fine-tune the placement. Upon touching the landmark, a tool-tip poops up showing the name of pin’s destination anatomic location.

If the user is not satisfied with the placement of a pin, there is option to edit it’s pose. The user will simply grab the pin with his/her hand and place the tip of the pin at the desired point on the patient’s body.

Once the user is completely satisfied with the placements of all the pins on the physical body of the patient, “End Alignment” voice command is used to end alignment process. Upon recognition of this last voice command, the IntraOpVSP will calculate the pose of the subject’s body in 6-DoF. IntraOpVSP will transform all the CT scan data accordingly. This step takes around a second of time. The operator can now see the holograms of different anatomic structures inside the subject’s physical body.

6.3 IMPLEMENTATION DETAILS

For this algorithm to work, minimum 4 landmarks on the subject's body must be determined first [38]. For our running example we have chosen five. These are Sternal Notch, Xyphoid, Base of Penis, Right Nipple, and Left Nipple. Next step is to measure the 3D locations of all these landmarks from the 3D mesh / CT scan data of the object. Just like in the case of automatic alignment, this measurement step is done manually using Amira3D or MeshLab software [33], [34]. The measurements are saved in a text file. I refer to these measured points as object points. The text file is uploaded to IntraOpVSP for the specific patient.

Once the operator has completed the landmark-pin placement process and used the voice command to complete alignment, IntraOpVSP retrieves the world coordinates of the pins. I refer to this second set of 3D points as the scene points.

Now we have both the scene points and objects points for calculating the rigid body transformation between the two sets of 3D points. Using Eigen's implementation of 3D rigid body transformation calculation, I then find the transformation matrix that would transform the object points to the pose of the scene points [39]. This represents the 6-DoF pose of the physical object to which we wanted to align our CT scan data with. Now using this calculated pose the CT scan data is transformed to the new location.

7 NORMAL VISUALIZATION

7.1 INTRODUCTION

Once the alignment is done, the hologram of the CT scan mesh data is shown in the operator's field of view. As the hologram is aligned with the anatomy of the patient, it creates an illusion that the operator can see through the skin and look at the different anatomical structures. Such as bones, veins, ventricles, tumors, vessels, liver etc. For navigating and intuitively interacting with these different structures several voice commands have been implemented. I'll present them in this section.

Often VSP for a surgery contains two versions of the same group of bones/structures. In IntraOpVSP I call them a "group". The bones/structures that make up that group is termed as "items". For instance, a VSP prepared for a surgery in the skull, may have three different group of bones as shown below.

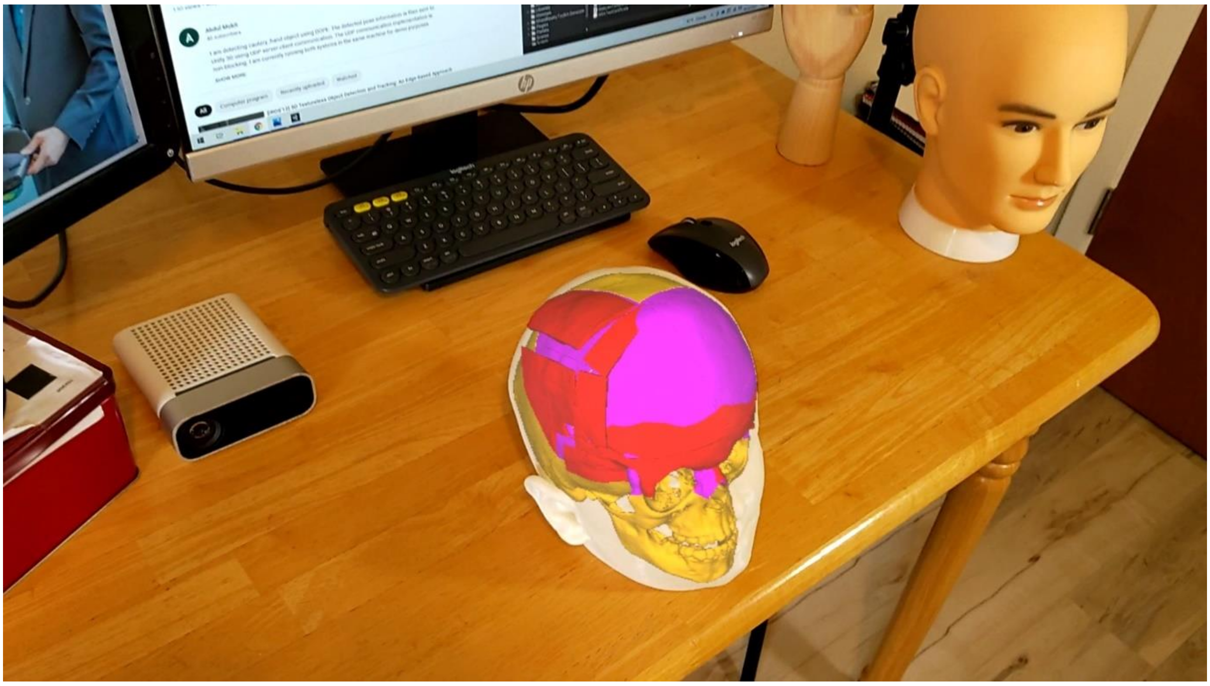


Figure 24: VSP is aligned with the 3D printed physical model of a skull. VSP is shown as hologram. Different groups of bones shown all at once but with different colors.

Different colored regions represent individual groups. The yellow group represents the base skull mode – a part which will remain unchanged after surgery. In IntraOpVSP I call them the "general" group. The purple group represents the region where the reconstructive surgery is going to take place. The red group represent the same region of interest in the patient's skull.

However, the purple group represents that region's state before the surgery has been done and the red group represents the post-operative state of the same region. In other words, the purple region and red region represent the *before* and *after* state of our region of interest. In IntraOpVSP I call these the "before" and "after" group respectively. Via surgery the surgeons aim to move the different items in the "before" group in such a way that it takes the form of the "after" group.

A group contains different items. Items represent either a bone segment or a specific anatomical structure. The "general" group in the running example has two items. These are the lower portion of the skull shown in yellow and the sinus shown in purple. In the earlier paragraph I didn't mention the sinus part to simplify explanation a bit.

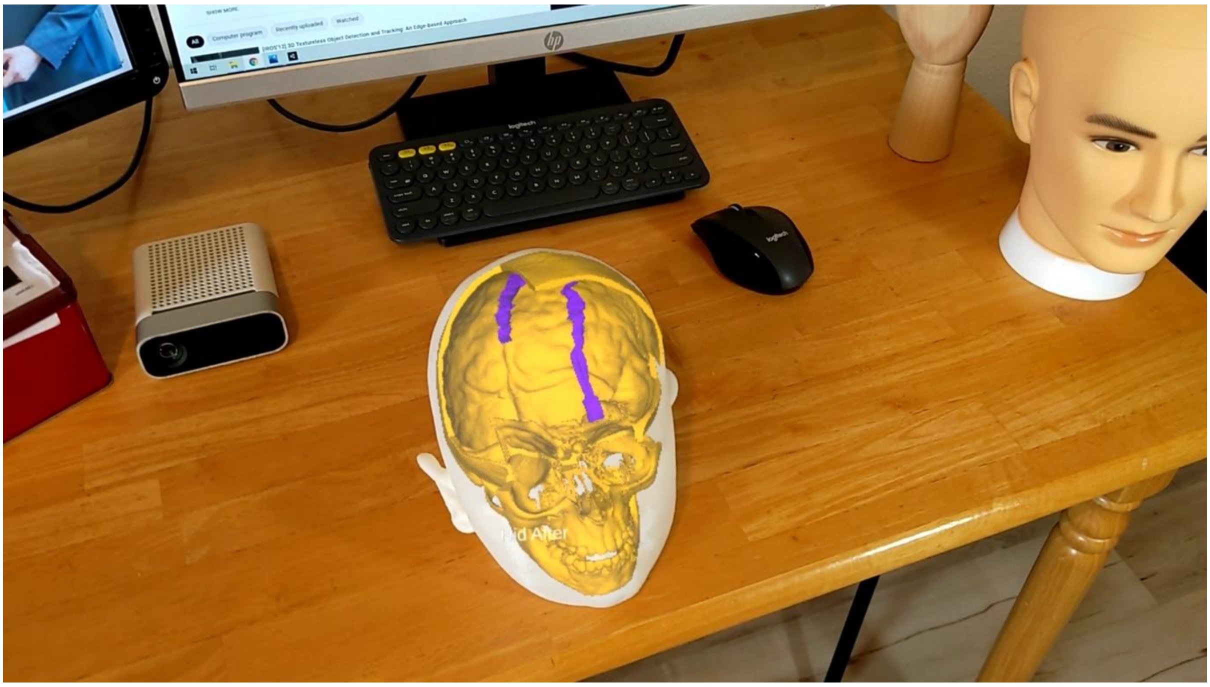


Figure 25: Only the "general group" is shown. The general group also contains an item named sinus displayed in purple. It is very important to know the pose of the sinus while surgery is being performed on the skull.

The *before* group contains several different bone segments. Each item is the result of the VSP by the surgeons. After VSP, surgeons had determined that cutting the skull in those shapes would give them the most advantage to reach their objective. The planned reconstruction is shown in the "after" group. Correspondingly the "after" group also contains moved/modified versions of the same items. To allow surgeons and students to interact with these individual items more instinctively I have also implemented "item level interaction" feature.

Figure 26: Only the "general" (yellow) and "before" (purple) group of bones are shown as VSP. Planned incision lines can be seen on the "before" group. The general group remains unchanged after surgery. Before group represents the state of the skull before surgery but is planned to be modified.

Figure 27: The red colored group is referred to as the "after" group. This group represents the destination shape of the skull after surgery. Different items making up this group can be seen.

7.2 USAGE

Different visualization commands have been implanted. First, the user may ask the IntraOpVSP to "hide all items". This command will hide all the holograms. This command is handy to clear all visualizations. Next, the user can command to iterate through each of the groups to see each group individually by saying "show next group". The user can also command the IntraOpVSP to show or hide individual groups. For instance, they can say “show before” or “hide before” to show or hide the *before* group of bones.

7.3 IMPLEMENTATION DETAILS

I have implemented several controller scripts that respond to voice commands and facilitates the show/hide functionality at a group level or item level. While designing the entire software and its features, I followed the single-responsibility principle. This allowed for easy extension and addition of features into individual classes without producing bugs in unassociated classes.

7.4 DISCUSSION

As CT scan data are shown as holograms, it creates an illusion to the user that he/she can now see inside the human body and visualize the internal anatomical structures in 3D. As discussed earlier, most AR/VR solutions have been bound in pre-operative VSP visualization. They were not usable in an intraoperative manner. Surgeons must memorize the VSP and look at them through either 2D monitors or paper prints during surgery.

Now, through IntraOpVSP they have the capability to bring up high quality 3D VSP visualization intraoperatively. The VSP data will be aligned with the patient’s anatomy. That means they never have to remove their eyes off the patient for a second. To do this they just have to ask the IntraOpVSP using voice commands. As no buttons are involved in our intuitive UI, the surgeons never have to disengage their hands form the critical tasks they were performing at those stages. This is an extremely important milestone in medical surgery. Surgeons at OU-medicine have found IntraOpVSP visualizations as a “game-changer”. A term commonly used by residents and surgeons who have tried IntraOpVSP. We are currently collecting data to perform quantitively analysis how IntraOpVSP has improved surgeon/resident performance, confidence, and efficiency.

8 ITEM VISUALIZATION AND INTERACTION

8.1 INTRODUCTION

Each item in a group can be touched and manipulated using hands. This feature has been added to enable users to instinctively interact with the holograms as if they were real objects.

Figure 28: A big item of the "before" group of bones is being moved with hand-ray.

8.2 USAGE

The way to interact is the far-interaction in MRTK. In this method, a hand ray shoots out of the hand-palm. The function of this ray is analogous to that of a mouse pointer. After the user points the ray to an item, he/she can then do the pinch-and-hold maneuver to move the items. However, the hand ray is usually turned off in IntraOpVSP. This is a practical design decision. By keeping the hand ray turned off by default, we get to prevent accidental interaction with any other virtual contents in the environment.

Figure 29: All items of both the before and after group of bones have been transformed in the space and spread around. This feature lets users inspect different items in more details.

The user uses the voice command “show hand ray” to activate the hand rays. Now they can click and drag the individual items. After the user has changed items’ positions, we can reset all items back to their original position by using the voice command “reset-items”. The callback for this command repositions all items in their original relative-locations. To turn the hand rays off again the uses has to say, “hide hand ray”.

8.3 IMPLEMENTATION DETAILS

By adding manipulation-enabling scripts to each of the items, this item-level-interaction feature was implemented in IntraOpVSP. Particularly the object manipulator class from MRTK was used. While adding this feature to each of the desired items, care must be taken to reshape the collusion bounding box surrounding each individual items. The hand rays interact with these invisible bounding boxes. Setting these boxes’ sizes incorrectly result into accidentally moving undesired items.

8.4 DISCUSSION

This feature is primarily for inspecting the items pre-operatively. During surgery the user rarely needs to use this feature. This feature is also useful for educational and pre-operative collaboration purposes.

9 GHOST VIEW

9.1 INTRODUCTION

Groups of items can be made translucent. We call this the “ghost mode” for visualization. In ghost mode, the group/item will not be entirely invisible. However, it will become translucent allowing the user to see the structures beneath it while not losing the entire perspective of the translucent segment.

9.2 USER MANUAL

Just like all other features this feature is also controlled by voice command. For the running example, by saying “ghost before” we can make the *before* group translucent. Same can be done for the *after* group. By viewing the *before* group in the ghost mode we can not only see the *before* group but also all structures beneath it.

Figure 30: “Before” group made translucent. The relative positioning of sinus (shown in purple) and the before group of the skull is more evident in this ghost mode visualization.

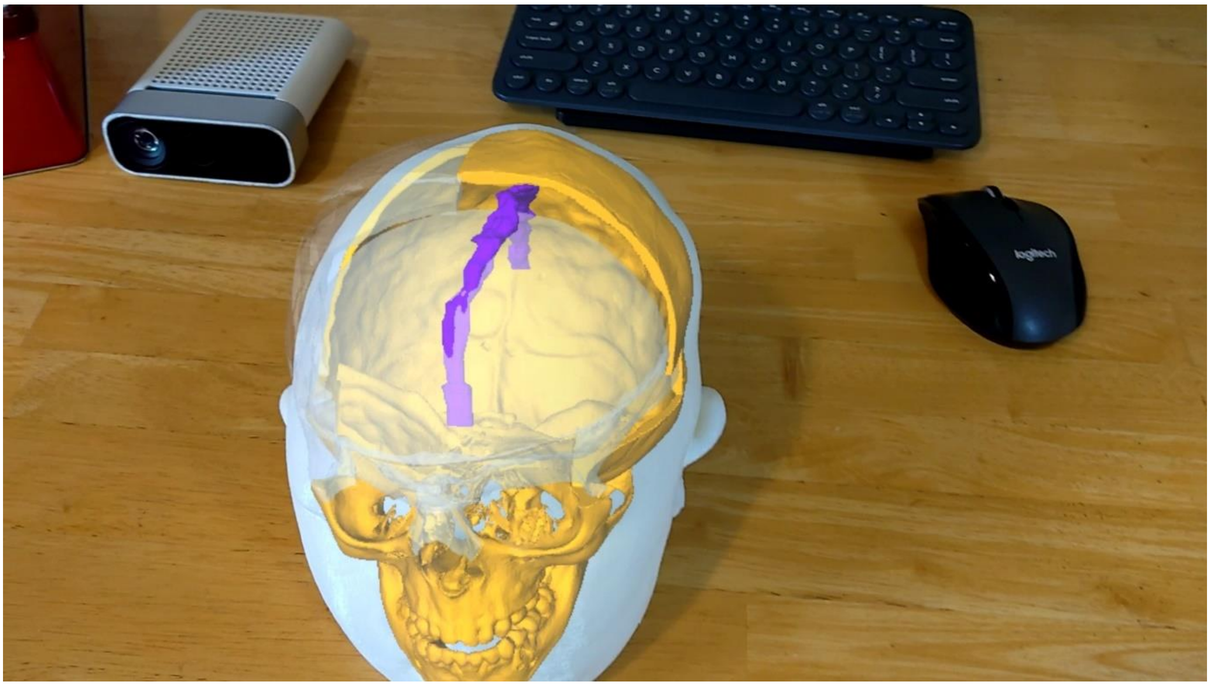


Figure 31: "After" group made translucent. The intersection of sinus (shown in purple) and the after part of the skull is now evident in this ghost mode visualization.

9.3 IMPLEMENTATION DETAILS

Transparent shaders and materials provided by MRTK were used to implement this feature [40]. I implanted some classes that facilitate attaching multiple materials to the same object. The script also implements the switching between these pre-selected materials. Each item than requires this feature must have this custom control script attached to them in unity.

9.4 DISCUSSION

This feature is primarily used for pre-operative visualization/collaboration purposes. The translucent view allows one to understand the spatial relationship between different items better. Intraoperatively this feature has low practical use. This is primarily due to the limitation of the Hololens display. In operating theater, surgeons often use very bright surgical light to brighten up the region of interest. The translucent holograms produced by Hololens are simply not bright enough to be noticeably visualized under the very bright surgical lights.

10 CROSS-SECTIONAL VIEW

10.1 INTRODUCTION

In IntraOpVSP one can view the dissection of anatomical structures. Surgeons often need to see the internal depth of bones or dissection of other anatomical structures to perform their work. This feature allows them to create a dissection view instantaneously. To present this feature a different example will now be considered in this section. In this example the bone group has been aligned with a mannequin head at first. Next, we use this slicing feature to view the dissection of the bone group at any position and from any angle we want. The user can see the depth of the bone in the dissection view.

Figure 32: 3D model of the skull has been aligned with a mannequin head.

Figure 33: Horizontal dissection shown at different heights by placing the slicing-plane at those heights.

10.2 USAGE

The user will use designated voice commands to turn this feature on and off. These are “show clip” and “hide clip”. The objects on which this feature will act on has to be predefined in Unity before compilation. Upon recognition of “show clip” command, all these virtual objects will be hidden. The virtual cutting plane can be moved using hands by the operator. Upon placing

Figure 34: A vertical dissection is created by placing the slicing-plane at a tilted position.

the virtual slicing plane through the anatomy of the patient, the surgeon can now look at a cross-section of that anatomical structure. In our running example, this anatomical structure is the entire skull of the patient.

10.3 IMPLEMENTATION DETAILS

The clipping plane and clipping box materials available in MRTK was used to implement this feature [41]. Object manipulator and solver classes from MRTK were used to control the movement of the clipping plane [42]. Clipping plane was made transparent using the translucent shader from MRTK [40].

To show the depth of bone a trick had to be done. I added a second bone mesh which is exactly like the original bone mesh. The only difference is that the second mesh is setup to render the negative surface normal associated with each face of the mesh. This trick lets us see the inner face of the bones when using the clipping plane. By keeping the slicing plane thin enough, we can create the illusion of bone-depth in IntraOpVSP. I implemented a few controller classes that facilitate the activation and deactivation of the feature.

11 DEPLOYMENTS AND LESSONS LEARNED

11.1 INTRODUCTION

The University of Oklahoma, Medicine's surgeons have tested with the IntraOpVSP in over 35 surgeries in the year 2021. These cases involve craniofacial surgery, orbital floor reconstruction, neurosurgery, liver, chest-bed reconstruction, ear reconstruction etc. Several data collection experiments are also under way to measure how IntraOpVSP is helping surgeons and residents become more confident and efficient in their work. Novel ways of visualizing data have also been discovered by OU medicine surgeon, using IntraOpVSP. In this section, I'll briefly focus on some of the ingenious ways IntraOpVSP was utilized by the OU-Medicine surgeons.

11.2 NEUROSURGERY

For Neurosurgery, intra operative view of the 3D models of skin, ventricles and skull-bone is very useful. IntraOpVSP was used to view these 3D models while performing surgery. The 3D models were anatomically aligned with the subject's head. Using different visualization control commands, different views of the data were created. Lastly, the dissection view was also utilized intraoperatively to help surgeons intuitively estimate the bone depth while making precise cuts through parts of the skull. This helps the surgeons become more confident when cutting through dense part of the skull. The visualizations also helped them identify and avoid cuts/steps that would complicate the procedure by causing avoidable blood loss. For instance, due to being able to see the ventricles through the physical skull, surgeons were able to avoid cutting important ventricles. This made the surgery more efficient and safer for both the surgeons and the patients. Through IntraOpVSP, they were able to navigate the probable future consequences of taking a step.

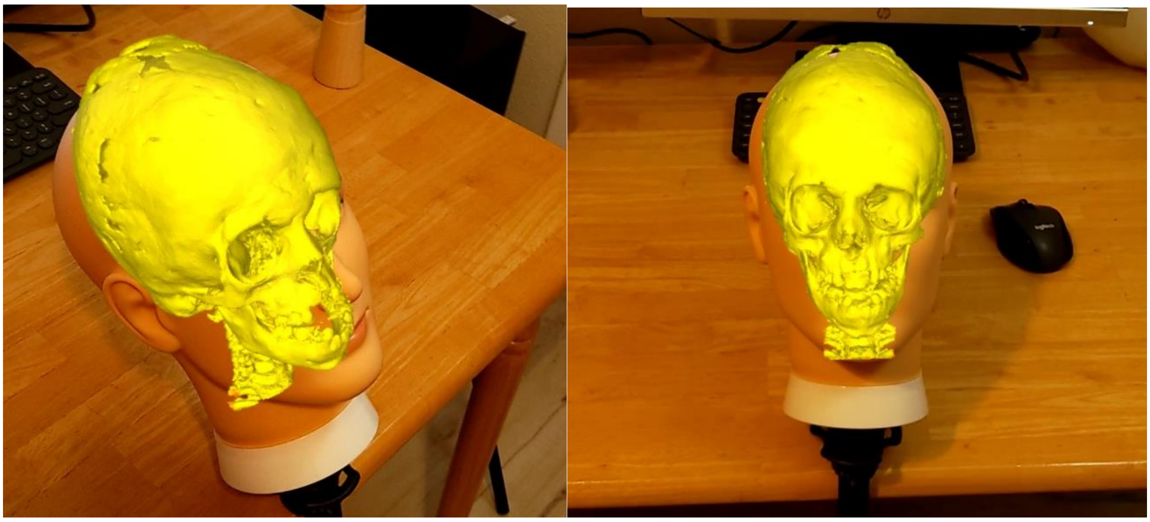


Figure 35: Showing bones only.

Figure 36: Showing ventricles only.

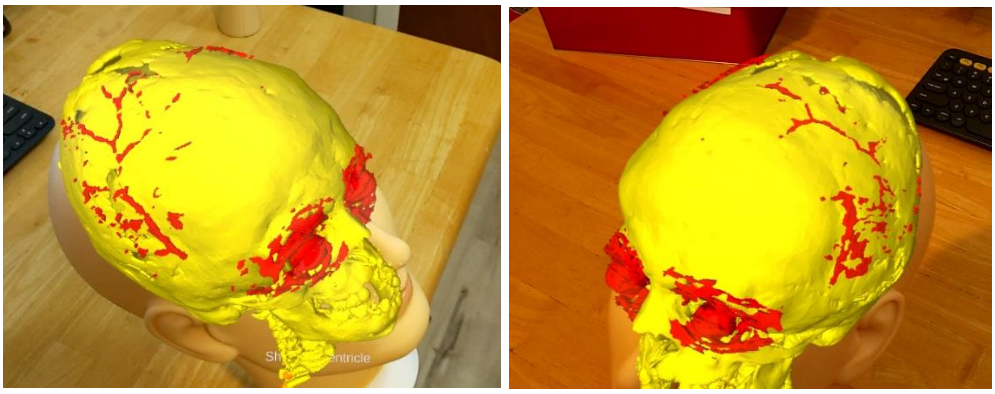


Figure 37: Showing both bones and ventricles as opaque.

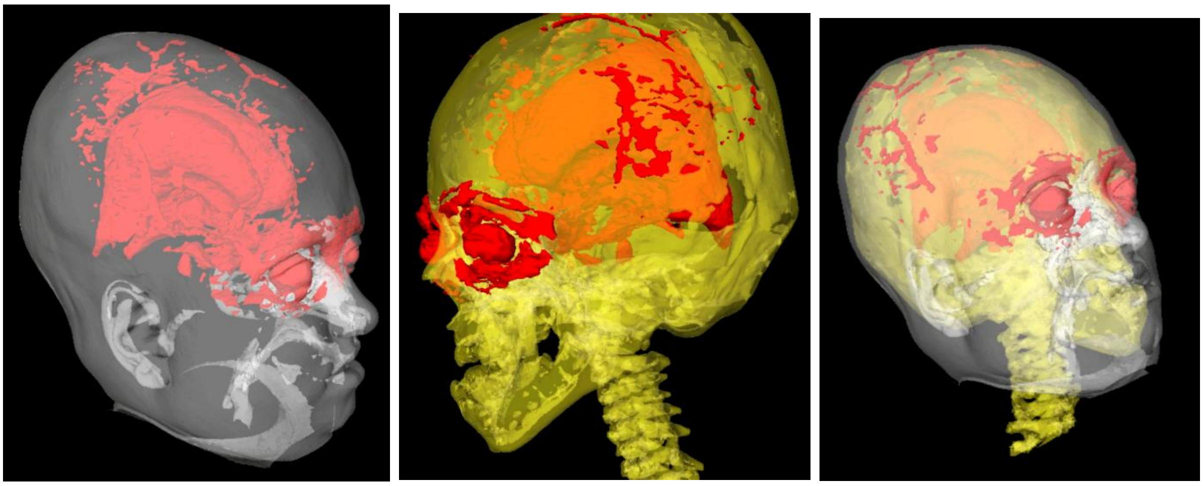


Figure 38: Different visuals can be created using the implemented voice commands. In the first image, skin is in ghost mode and ventricles is opaque. In the second image, only bones are in ghost mode and ventricles are opaque. In the third image, both skin and bones are in ghost mode but the ventricles are still opaque.

11.3 SURGICAL STEPS NAVIGATION

IntraOpVSP has been used during surgical cases to enhance the safety and efficiency of complex reconstructions. Many patients come for craniofacial reconstruction after a traumatic injury. One such surgical case, required an 18-step reconstruction of the face. IntraOpVSP overlaid the patient's VSP data on top of his real bones. Each one of those bones needed to be cut and moved in a precise direction. The device allowed surgeons to see the bones individually, then it displayed each of the cuts and each of the movements. This allowed the surgeon to verify that he had gone through all those steps. IntraOpVSP was basically walking the surgeons through the steps of surgery in mixed reality.

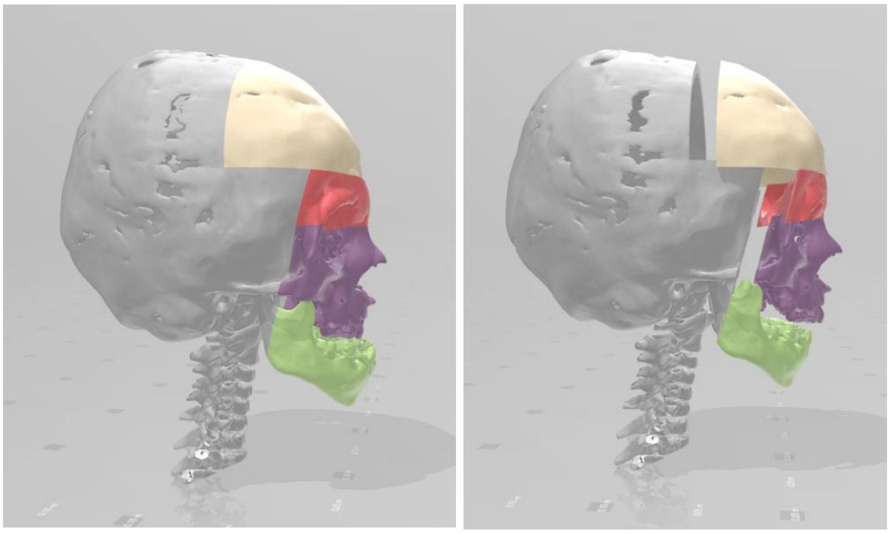


Figure 39: First image represents the planned cuts and resulting bone segments in their pre-operative state. Second image represents the same segments but moved according to planned VSP.

11.4 LIVER SURGERY

Tumor removal from liver has been performed using IntraOpVSP. After aligning the patient’s VSP data with the physical body, surgeons were able to look inside the body and directly look at the tumors at their correct spatial location. The visualization would help them to avoid damaging other structures that were wrapping around the tumors. The usage revealed the necessity of deformable 3D mesh/models of anatomic structures. After a liver is being operated on, it doesn’t maintain a static structure as shown in the holograms. This limitation of IntraOpVSP, revealed a research topic that would require significant investment to solve.

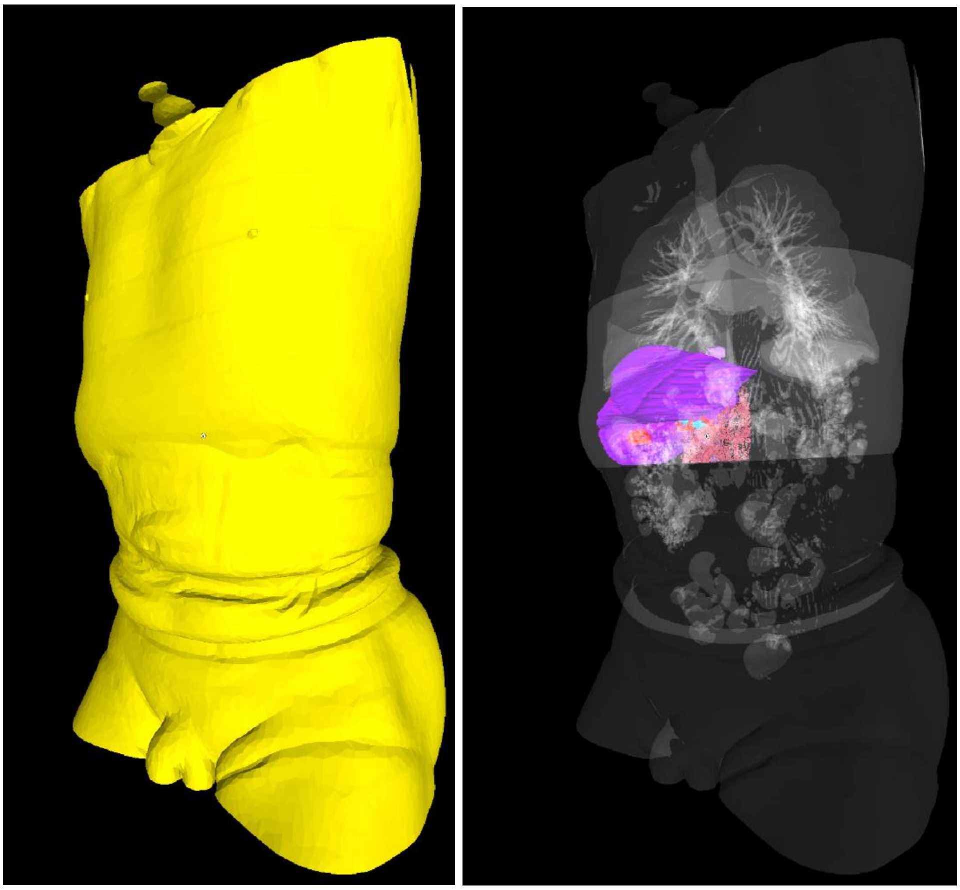


Figure 40: First image represents the opaque skin 3D model. In the second image the skin has been made translucent. As a result, now we can see the intricate internal structures of the skin model as well as the liver associated anatomical structures.

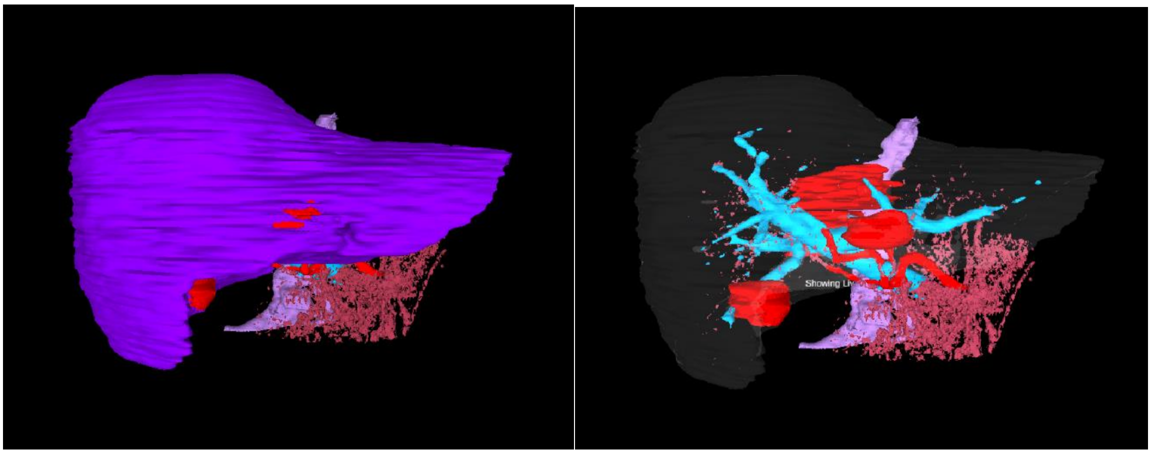


Figure 41: In the left image the liver's 3d model along with associated anatomical structures are shown in normal opaque mode. In the second image, the liver's 3D model has been made translucent. As a result, now we can see the liver's internal structures and most importantly the red blobs representing tumors.

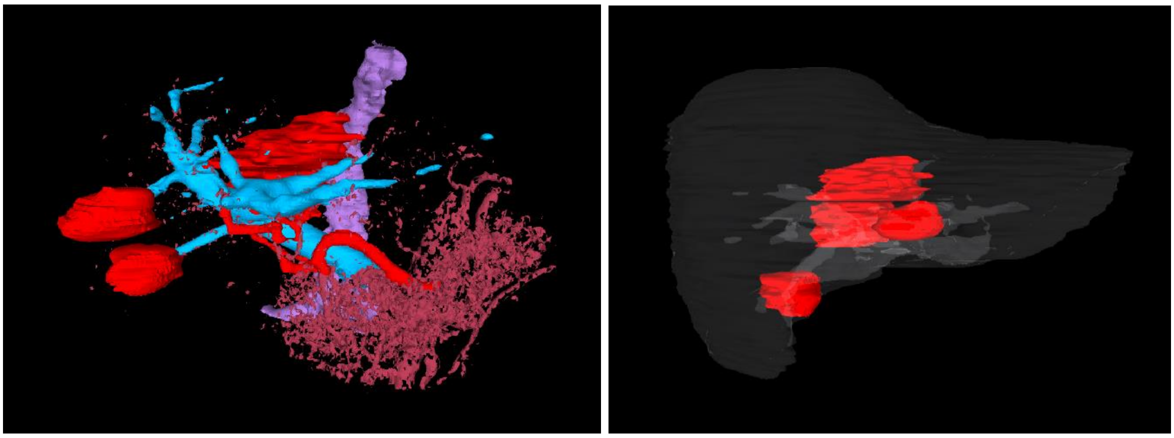


Figure 42: In the left image, the liver has been hidden completely, exposing a detailed view of the internal structures. These include artery, cava, tumors, veins, and vessels. In the image in the right all items except the tumors are hidden and the liver 3D model is made translucent again. These visualization helps surgeons perceive the geometric relationship among different items while performing surgery.

11.5 CHEST DEFORMATION CORRECTION

Chest rib-cage deformation correction was performed using IntraOpVSP. The subject's rib cage bones needed reconstruction. The manual landmark placement-based registration method allowed surgeons to align the subject's VSP to the physical body. Once aligned, the VSP helped surgeons navigate through the steps of surgery and helped identify avoidable damages to other associated anatomic structures.

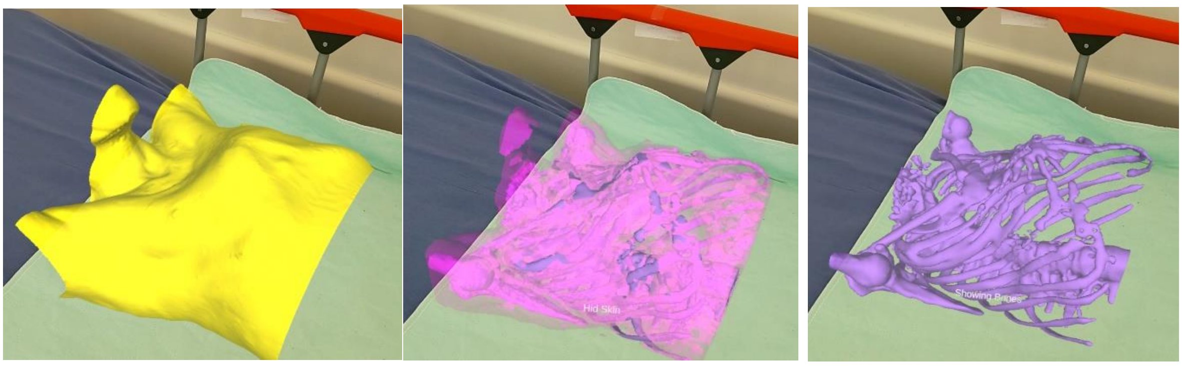


Figure 43: From left to right the three images represent 3D models of the skin, bones & muscle together, and only bones respectively.

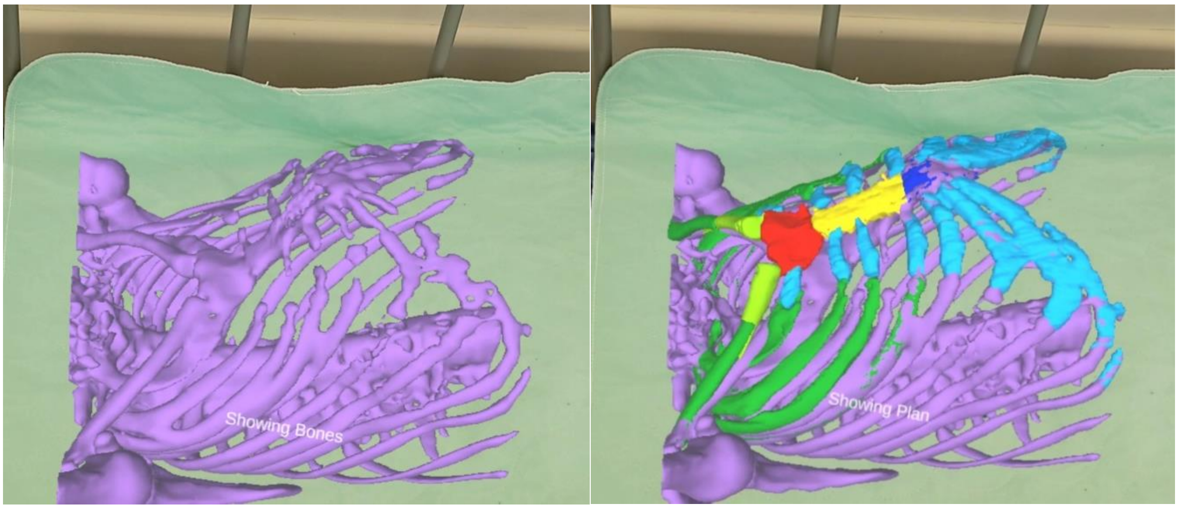


Figure 44: The left image represents the pre-operative original condition of the chest bones. The surgical plan showing the destination of different parts of the bones is represented in the second image.

11.6 EAR RECONSTRUCTION

In this case, one of the ears of the patient was missing. Traditionally the ear is reconstructed by looking at 2D printed images of the healthy ear. The surgeons must entirely rely on their intuition to recreate an ear that is as similar as possible to the healthy ear’s image. IntraOpVSP helped them reconstruct the ear by projecting a 3D hologram of how the ear should ideally look like after reconstruction at its destination anatomic location. After VSP, a mirrored version of the healthy ear was superimposed as a hologram at the destination location of the patient’s head. This is a novel way of reconstructing ears intraoperatively.

Figure 45: 3D scan of the patient’s head before surgery. The subject requires reconstructive surgery in his right ear as shown in the rightmost image.

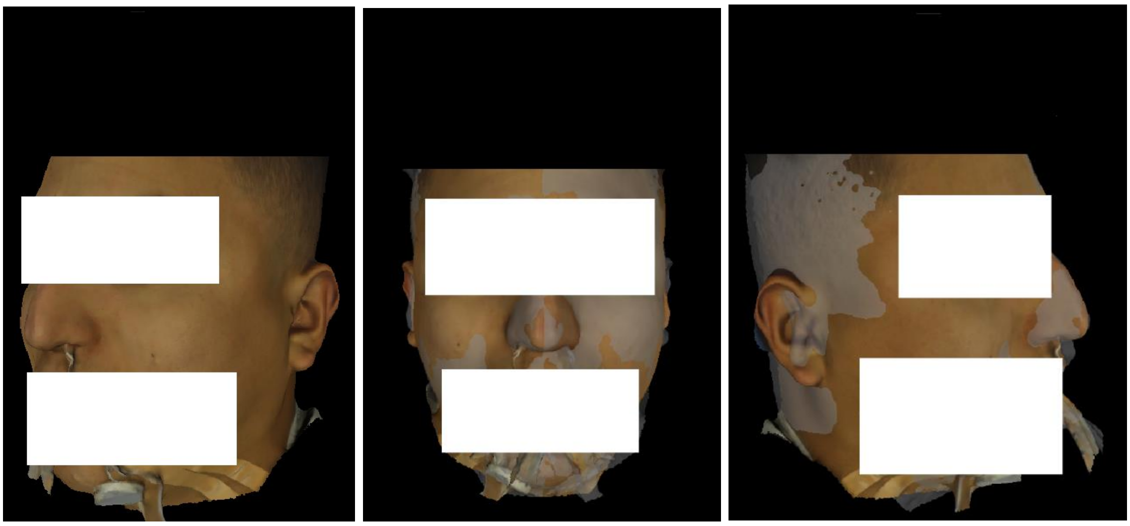


Figure 46: These images represent the destination holograms; surgeons use to perform surgery. The left-most image represents the unchanged left ear. In the center image, the "after" scan is shown as opaque and the "before" scan is shown as translucent. This visualization allows to verify how good the mirroring of the original "before" scan is. Same visualization but from the right ear's angle is shown in the last image. The surgeons now will have a 3D hologram of the ideal final shape of the right ear during performing surgery.

11.7 ORBITAL FLOOR RECONSTRUCTION

Orbital floor reconstruction was performed using IntraOpVSP. One side of the face needed reconstruction. Like the ear-reconstruction case, a mirror of the original bone 3D model was used to give surgeons a visual reference of how the damaged side should look like after reconstruction.

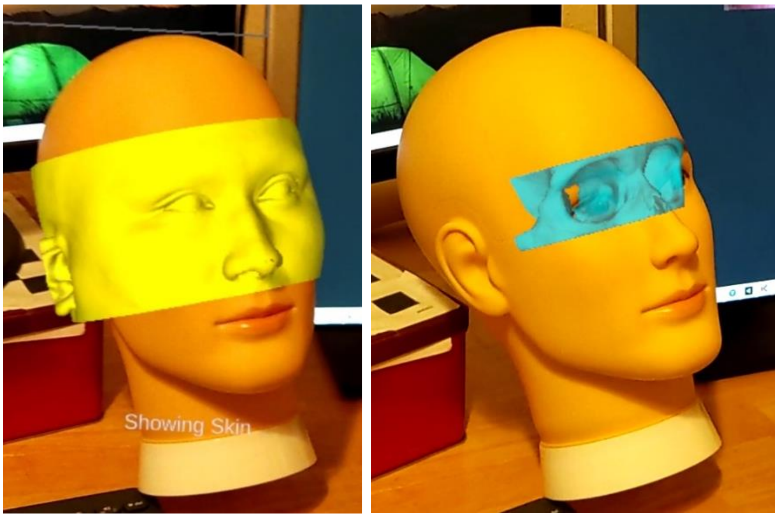


Figure 47: Skin 3D model's alignment is demonstrated in the first image. In the second image only the before group of bones is shown. The subject requires surgery at the right eye.

Figure 48: In the first image a red "combo" plate is shown. The objective of the surgery is to place the actual physical "combo" plate at same pose as this red hologram. The mirror of the "before" 3D model is shown in the second image. This "after" 3D model represents how the bones segment should look like if the reconstructive surgery was performed successfully.

12 CONCLUSION

In this research, we have made a medical mixed reality based VSP visualization software named IntraOpVSP. Through Microsoft Hololens-2, IntraOpVSP allows surgeons to look directly inside the body of the patients and investigate the internal anatomy as 3D holograms. Surgeons can visualize and interact with the 3D virtual anatomy without shifting their focus from the patient or disengaging their hands. They can align the CT scan data with respect to the subject's body in various fast and convenient ways. Finally, they can create different visualizations of the anatomy using several implemented modalities. IntraOpVSP has gone through multiple iterations of improvements based on the feedbacks from surgeons after they had used it in numerous surgeries. Our key design principle was to ensure maximum usefulness to surgeons while causing the least amount of distraction or inconvenience. IntraOpVSP will pave the way for a "google-maps" for surgery.

13 REFERENCES

[1] B. Peuchot, A. Tanguy, and M. Eude, “Virtual Reality as an Operative Tool During Scoliosis Surgery,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 905, pp. 549–554, 1995, doi: 10.1007/978-3-540-49197-2_72.

[2] H. K. Wu, S. W. Y. Lee, H. Y. Chang, and J. C. Liang, “Current status, opportunities and challenges of augmented reality in education,” *Comput. Educ.*, vol. 62, pp. 41–49, Mar. 2013, doi: 10.1016/J.COMPEDU.2012.10.024.

[3] L. B. Rosenberg, “The Use of Virtual Fixtures as Perceptual Overlays to Enhance Operator Performance in Remote Environments.” 01-Sep-1992.

[4] J. Steuer, “Defining Virtual Reality: Dimensions Determining Telepresence,” *J. Commun.*, vol. 42, no. 4, pp. 73–93, 1992, doi: 10.1111/J.1460-2466.1992.TB00812.X.

[5] “Augmented Reality & Virtual Reality In Healthcare Market Report, 2028.” [Online]. Available: https://www.grandviewresearch.com/industry-analysis/virtual-reality-vr-in-healthcare-market. [Accessed: 28-Oct-2021].

[6] “ImmersiveTouch® — Medical VR Solutions.” [Online]. Available: https://www.immersivetouch.com/. [Accessed: 02-Nov-2021].

[7] “Proprio - Vision for the future of medicine.” [Online]. Available: https://www.propriovision.com/index.html. [Accessed: 02-Nov-2021].

[8] “Virtual Reality for Surgery | Precison XR | Surgical Theater.” [Online]. Available: https://surgicaltheater.com/. [Accessed: 02-Nov-2021].

[9] “FundamentalVR: Working at the Intersection of Immersive Tech.” [Online]. Available: https://www.fundamentalvr.com/. [Accessed: 02-Nov-2021].

[10] “MAI.” [Online]. Available: https://www.mai.ai/. [Accessed: 02-Nov-2021].

[11] “Osso VR.” [Online]. Available: https://www.ossovr.com/. [Accessed: 02-Nov-2021].

[12] “Medivis | AR Medical Technology.” [Online]. Available: https://www.medivis.com/. [Accessed: 02-Nov-2021].

[13] J. Rolland, H. F.- Presence, and undefined 2000, “Optical versus video see-through head-mounted displays in medical visualization,” *ieeexplore.ieee.org*, 2000.

[14] “What is Mixed Reality? - Mixed Reality | Microsoft Docs.” [Online]. Available: https://docs.microsoft.com/en-us/windows/mixed-reality/discover/mixed-reality. [Accessed: 02-Nov-2021].

[15] P. Milgram and F. Kishino, “A Taxonomy of Mixed Reality Visual Displays Using Stereoscopic Video for Defence Teleoperation View project paramedics response bags View project A TAXONOMY OF MIXED REALITY VISUAL DISPLAYS,” *IEICE Trans. Inf. Syst.*, no. 12, 1994.

[16] “HoloLens 2—Overview, Features, and Specs | Microsoft HoloLens.” [Online].

Available: https://www.microsoft.com/en-us/hololens/hardware. [Accessed: 06-Apr-2021].

[17] "Hand tracking - Mixed Reality Toolkit | Microsoft Docs." [Online]. Available: https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/features/input/hand-tracking?view=mrtkunity-2021-05. [Accessed: 03-Nov-2021].

[18] "Introducing Microsoft Mesh | Here can be anywhere." [Online]. Available: https://www.microsoft.com/en-us/mesh. [Accessed: 02-Nov-2021].

[19] "Stanford Medical Mixed Reality (SMMR) | Incubator for Medical Mixed and Extended Reality at Stanford | Stanford Medicine." [Online]. Available: https://med.stanford.edu/immers/smmr.html. [Accessed: 02-Nov-2021].

[20] "MediView – Redefining the cancer care continuum." [Online]. Available: https://mediviewxr.com/. [Accessed: 02-Nov-2021].

[21] "Experience Collaborative Medicine - VSI HoloMedicine®." [Online]. Available: https://apoqlar.com/. [Accessed: 02-Nov-2021].

[22] "Nakamir." [Online]. Available: https://nakamir.net/. [Accessed: 02-Nov-2021].

[23] "Unity Real-Time Development Platform | 3D, 2D VR & AR Engine." [Online]. Available: https://unity.com/. [Accessed: 02-Nov-2021].

[24] "microsoft/MixedRealityToolkit-Unity: Mixed Reality Toolkit (MRTK) provides a set of components and features to accelerate cross-platform MR app development in Unity." [Online]. Available: https://github.com/microsoft/MixedRealityToolkit-Unity. [Accessed: 02-Nov-2021].

[25] J. I. Efanov, A.-A. Roy, K. N. Huang, and D. E. Borsuk, "Virtual Surgical Planning: The Pearls and Pitfalls," *Plast. Reconstr. Surg. Glob. Open*, vol. 6, no. 1, 2018, doi: 10.1097/GOX.0000000000001443.

[26] "Hand menu - Mixed Reality | Microsoft Docs." [Online]. Available: https://docs.microsoft.com/en-us/windows/mixed-reality/design/hand-menu. [Accessed: 02-Nov-2021].

[27] "Voice input - Mixed Reality | Microsoft Docs." [Online]. Available: https://docs.microsoft.com/en-us/windows/mixed-reality/design/voice-input. [Accessed: 02-Nov-2021].

[28] "Object manipulator - Mixed Reality Toolkit | Microsoft Docs." [Online]. Available: https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/features/ux-building-blocks/object-manipulator?view=mrtkunity-2021-05. [Accessed: 02-Nov-2021].

[29] "Bounds control - Mixed Reality Toolkit | Microsoft Docs." [Online]. Available: https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/features/ux-building-blocks/bounds-control?view=mrtkunity-2021-05. [Accessed: 02-Nov-2021].

[30] "HoloLens 2 hardware | Microsoft Docs." [Online]. Available:

https://docs.microsoft.com/en-us/hololens/hololens2-hardware. [Accessed: 02-Nov-2021].

[31] “Locatable camera overview - Mixed Reality | Microsoft Docs.” [Online]. Available: https://docs.microsoft.com/en-us/windows/mixed-reality/develop/advanced-concepts/locatable-camera-overview. [Accessed: 02-Nov-2021].

[32] “Head Pose Estimation using OpenCV and Dlib | LearnOpenCV #.” [Online]. Available: https://learnopencv.com/head-pose-estimation-using-opencv-and-dlib/. [Accessed: 29-Oct-2021].

[33] “Avizo Software for materials research | Thermo Fisher Scientific - US.” [Online]. Available: https://www.thermofisher.com/us/en/home/electron-microscopy/products/software-em-3d-vis/avizo-software.html. [Accessed: 02-Nov-2021].

[34] “MeshLab.” [Online]. Available: https://www.meshlab.net/. [Accessed: 02-Nov-2021].

[35] “dlib C++ Library.” [Online]. Available: http://dlib.net/. [Accessed: 03-Nov-2021].

[36] “OpenCV: Camera Calibration and 3D Reconstruction.” [Online]. Available: https://docs.opencv.org/3.4.15/d9/d0c/group__calib3d.html. [Accessed: 03-Nov-2021].

[37] “Tooltip - Mixed Reality Toolkit | Microsoft Docs.” [Online]. Available: https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/features/ux-building-blocks/tooltip?view=mrtkunity-2021-05. [Accessed: 03-Nov-2021].

[38] D. W. Eggert, A. Lorusso, and R. B. Fisher, “Estimating 3-D rigid body transformations: a comparison of four major algorithms,” *Mach. Vis. Appl. 1997 95*, vol. 9, no. 5, pp. 272–290, 1997, doi: 10.1007/S001380050048.

[39] “Eigen: Geometry module.” [Online]. Available: https://eigen.tuxfamily.org/dox/group__Geometry__Module.html. [Accessed: 03-Nov-2021].

[40] “MRTK standard shader - Mixed Reality Toolkit | Microsoft Docs.” [Online]. Available: https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/features/rendering/mrtk-standard-shader?view=mrtkunity-2021-05. [Accessed: 03-Nov-2021].

[41] “Clipping primitive - Mixed Reality Toolkit | Microsoft Docs.” [Online]. Available: https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/features/rendering/clipping-primitive?view=mrtkunity-2021-05. [Accessed: 03-Nov-2021].

[42] “Solver overview - Mixed Reality Toolkit | Microsoft Docs.” [Online]. Available: https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/features/ux-building-blocks/solvers/solver?view=mrtkunity-2021-05. [Accessed: 03-Nov-2021].