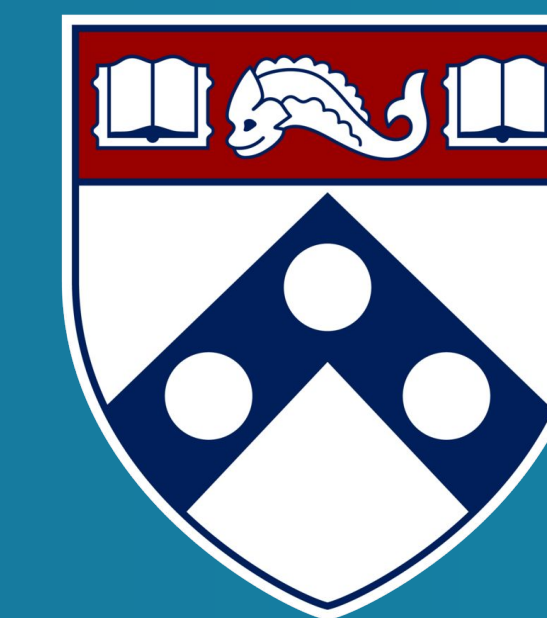


Developing a Surgical Skin Cutting System For Virtual Reality



Dineth Meegoda (SEAS '25), Kristoffel Dumon (MD), William Yi (MD), Daniel Weber

Department of Surgery, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA

Objective

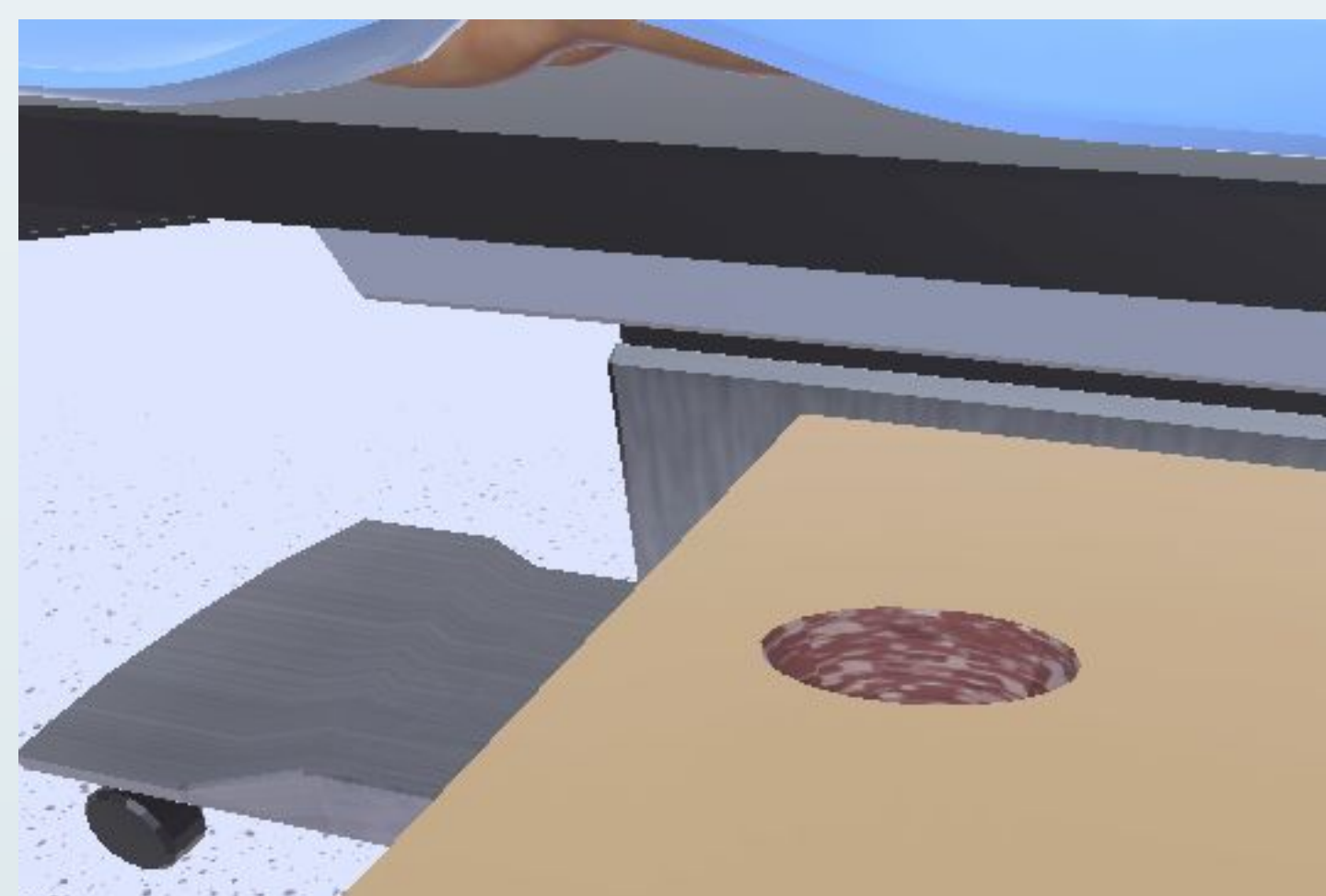
The objective of this project was to find an efficient way to generate a skin cutting system that would behave accurately to real world surgical skin cutting and maintain a minimum of 60 frames per second when testing the simulation on the Oculus Quest 2.

Background

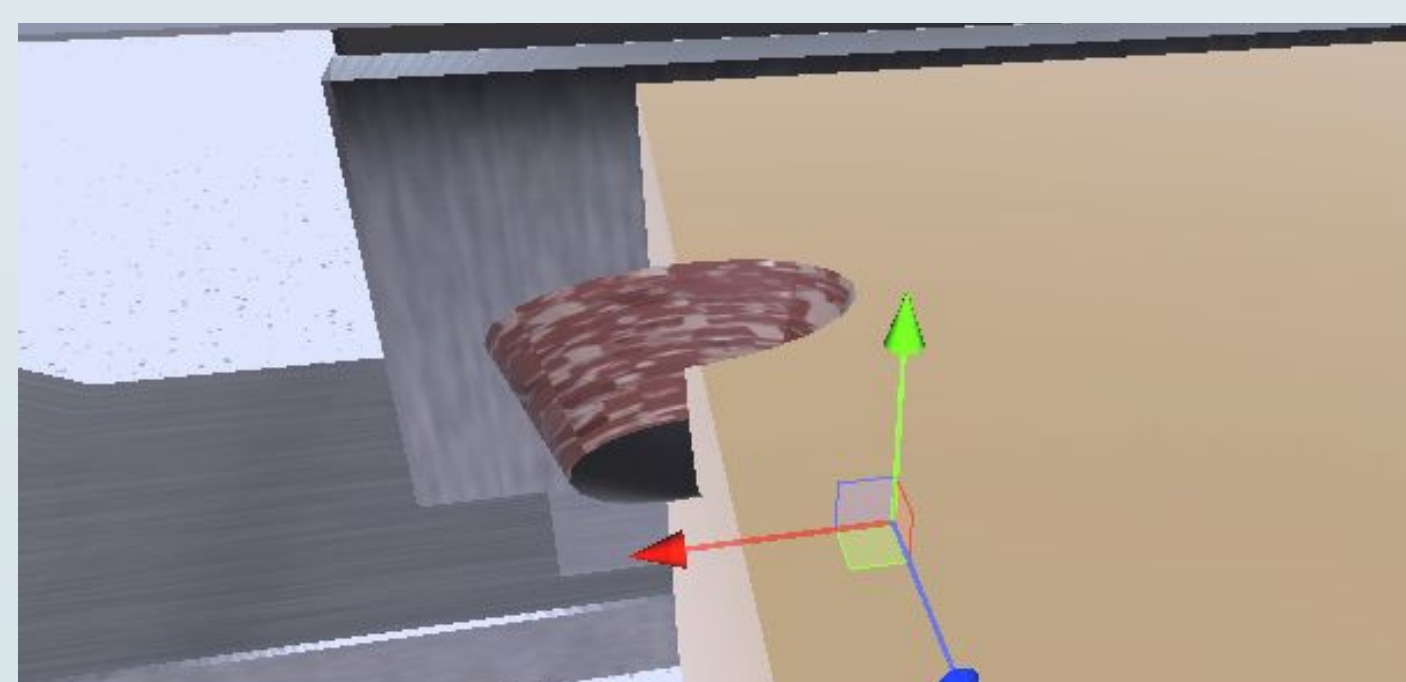
Virtual Reality (VR) has been utilized as an effective educational tool that places the learner directly into their subject matter. Due to its immersion, Virtual Reality is currently being explored in the medical, nursing, and surgical fields as a way to accurately teach and assess both the intellectual and physical aspects of each occupation. Currently, the University of Pennsylvania Hospital is working on a VR Simulation to give surgeons, residents, and medical students experience with a procedure known as a 'Surgical Airway'. It is a very precise technique that requires the surgeon to cut into the patient's throat in a very exact manner in order to provide them with an emergency airway. However, to properly simulate surgical techniques in virtual space, the cutting of the patient's skin and other organs must be replicated in a way that would intuitively follow how cutting the skin with a scalpel would behave in the real world. There are existing libraries to simulate splitting an object along a line or curve into two separate meshes, but no such library exists that allows for an object to be cut into in a surgical fashion. To replicate this is technically advanced, and even more importantly, very computationally intensive. The simulation is expected to run on the Oculus Quest 2 Headset's hardware with no support from an external computer and as a result, graphical techniques such as generating new mesh geometry when an object is 'cut' would be too intensive for the simulation to run smoothly on the Oculus.

Materials

- The Unity Game Engine was used as the game engine to run the simulation. It was used due to its XR toolkit, which is a built in API that allows for Virtual Reality Development.
- Oculus Quest 2 was the headset used for this project due to its wireless and user friendly interface. The target audience of this simulation were not expected to be familiar with the technical aspects of VR Headsets or wiring devices to computers so the project was meant to be standalone.
- A spline library written by Sebastian Lague was used to store and create a path for the cuts drawn by the user in the form of vertex paths and bezier curves.



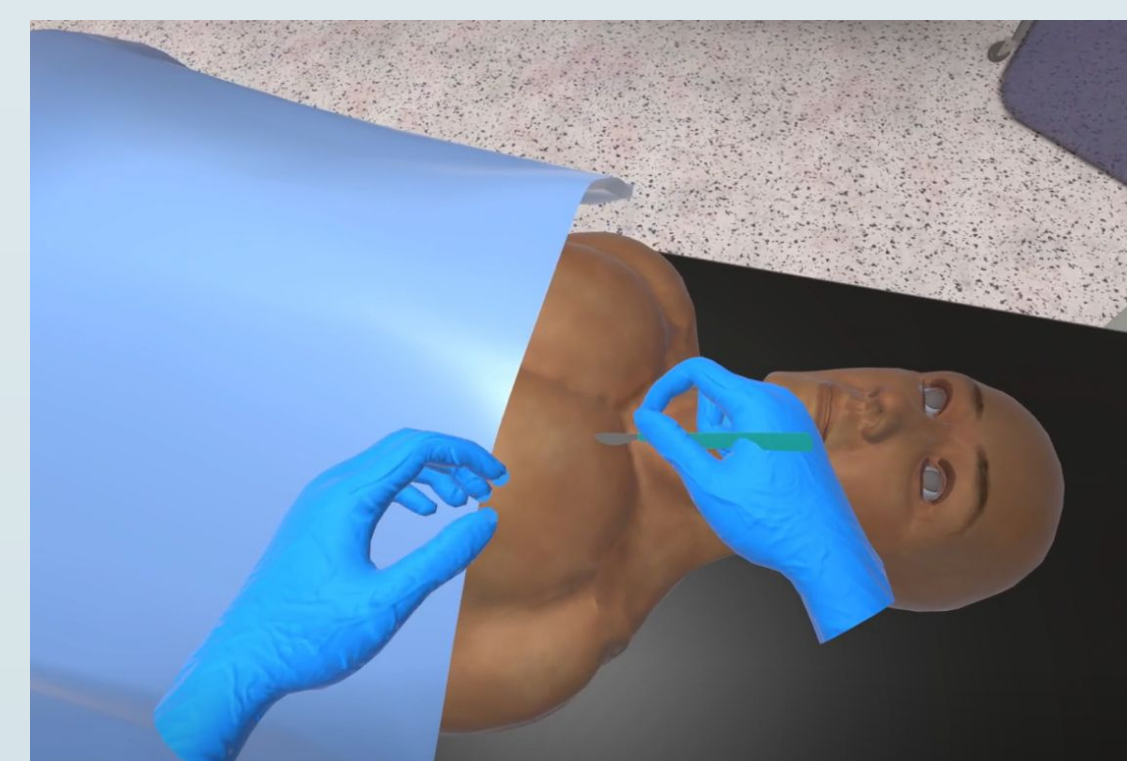
A cut hole mesh appearing as a hole inside of the cuttable object



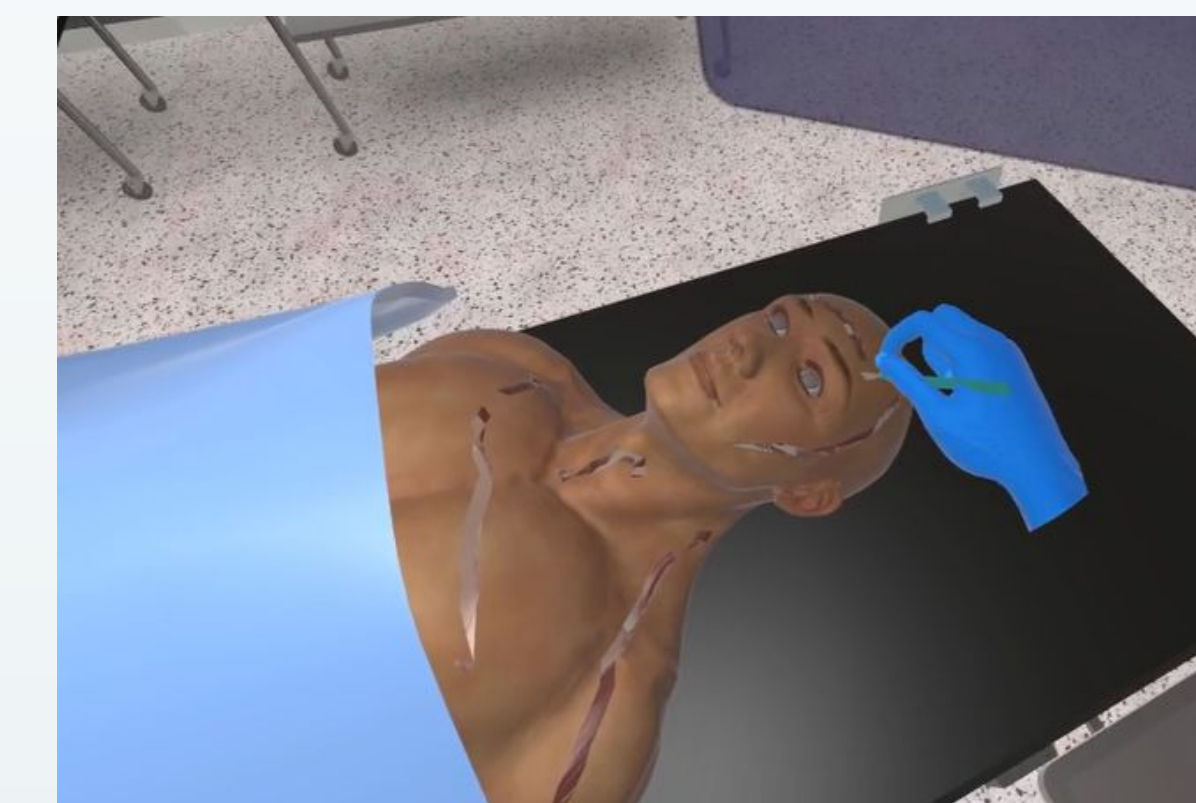
View of the cut mesh partly outside of the cuttable object

Methodology

The key technique that was used to create a cut was the use of a custom renderer that displayed objects underneath another object which created the visual illusion of seeing the object being cut. The next problem was tracking the cut and making it follow the user's hand specifically. Our simulation used Oculus' hand tracking feature which eliminated the need for a controller allowing free movement and an accurate simulation of the surgical procedure. To efficiently track three dimensional points, a hashmap was created with a formula that converted three dimensional points into a single digit that tracked its region of space. When the user entered a new region, a new point on a spline was created, allowing the simulation to only create new points when a new region denoted by the hash map. The spline library allowed us to create a continuous path for the cut using only the information of where specific points on the cut are meant to be. However, the points must map to the surface of the cuttable object because the patient's body is not a even surface. To accomplish this, a raycaster object was placed on the user's hand and fires a ray pointing towards the end of the scalpel. If the ray makes contact with a cuttable surface, that point on the surface is added to the spline. Finally, the surface of the cut had to match the normal direction of the cuttable object's surface. To solve this, the raycaster outputted the normal data of the surface that it hit and a mesh was generated with these normals and location of the surface to create the illusion of a cut.



The patient model before the user creates cuts.



The patient model after the user creates cuts.

Results & Discussion

Results & Discussion:

The average frames per second was 64. This was done over 12 tests, 3 tests for each of 4 cuttable objects, a cube, a sphere, a cylinder, and a human patient model. This cutting system creates accurate cuts in an acceptable frame rate. However, the program can slow down if too many cuts are created. On sharp edges, the vertices of the cut are not accurately marked and can create inaccurate cuts.

Future Work

- Adding functionality to the depth of the cut.
- Increasing the efficiency of the program and code.
- Merging adjacent cuts together.
- Changing the width of the cut by using the user's finger.

References & Acknowledgements

- XR Interaction Toolkit, Unity Manual. <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@2.1/manual/index.html>
- Yi William, et al. "Development and Evaluation of an Immersive Virtual Reality Pneumothorax and Chest Tube Management Simulation." Set to be published in SAGES Surgery Conference, 2023.
- Special thanks to Dr. Kristoffel Dumon and Dr. William Yi and Daniel Weber
- This project was funded by the Penn Undergraduate Research Mentoring (PURM) program and the University of Pennsylvania.