HoloSurgery

Volume rendering of medical imaging in AR

By: Ariel Yehezkely

Advisor: Miri Ben-Hen, Boaz Sternfeld.

In collaboration with:

Microsoft Garage Israel

Microsoft

THEGARAGE

Beilinson Hospital

RABIN MEDICAL CENTER
BEILINSON HOSPITAL

Contents

Introduction

Medical imaging is a pre-surgery procedure that helps surgeons diagnose patient's medical condition, plan the surgery in advanced and navigate during the surgery.

There are many different technologies of medical imaging but the two common machines in use are CT and MRI, which produce an array of 2D images with constant spacing between them. Radiologists and surgeons are used to viewing the sequence of images one by one and extract the data they need from the 2D grayscale images, but such ability is gained over time and experience.

The human mind is used to 3D processing and visual 3D content helps us understand and remember data in easier way. In addition, coloring help us distinguish between various parts in a picture and understand boundaries and relations in the scanned body parts. The nowadays medical imaging diagnosing tools can visualize 3D reconstruction of the examined body parts and helps surgeons understand the constriction of the place they are about to operate in advanced, but usually those reconstructions are used for visual purposes only and senior surgeons barely use this feature.

During a surgery, the surgeon has to remember what he saw in the pre-surgery phase and operate according his memory of the locations of the patient's organs, tumors, vessels etc., and in case of review need he cannot operate the display himself but ask a nurse to do it for him due to sterilization he must contain during surgery.

Project Goal

Creating an application for viewing 3D holograms made from medical imaging data in AR with Microsoft HoloLens for:

1. Help surgeons view the medical imaging and operate the data view during surgery by using hand gestures for efficient and comfortable usage while preserving sterilization.
2. Create 3D hologram out of the 2D pictures in order to give the surgeon 3D understand of the organs and entire body relation and have it during the surgery mixed in the real world and the real patient making an "Invisible patient" and ability to see through tissues.
3. 3D slice ability of the hologram for viewing the relevant part in the current operation.
4. Superposition of the Hologram and the real body part for better understand the inner construction of opaque tissues.

Using AR for viewing holograms on top of the real world can help surgeon's perception and understating due to the 3D properties of the human mind.

Using hand gestures can help surgeons operate the viewing system themselves and not through another person.

Viewing inside parts like vessels inside tissue give the surgeon understanding of the inner parts of organs and help the surgery procedure.

Implementation

Hardware

Microsoft HoloLens are full wearable computer headset running windows 10 OS and can run Universal Windows Applications (UWP). The headset contains depth cameras which produce spatial mapping of the headset surroundings and by placing a 3D hologram in this coordinate system the user has feel of reasonable 3D content projected into the real world.

Software

The application is a UWP application written in C# using DirectX and HLSL shaders on the HoloLens HPU graphic processor.

Algorithm

The application use Volume rendering algorithm that is well described in computer graphics sources and has many implementations and various usages.

The implementation used in this project is consist of the following steps:

1. Load Data to the GPU –
   a. 2D images Array, as 3D texture. The texture volume is inside the uvw coordinates: (0,0,0) -(1,1,1)
   b. 3D sphere vertex coordinates, and transformation matrix to the world coordinates.
   c. Head position in the world coordinates
   d. UI data.
2. In vertex shader, apply the world transform matrix on the sphere vertices and pass the data to the pixel shader.
3. In pixel shader:
   a. Calculate transformation between world coordinate system and texture coordinate system.
   b. For each pixel calculate the LOS vector between head position and relevant pixel position.
   c. Sample the texture on the LOS vector in fixed steps.
   d. Calculate the pixel's color using transfer function on the samples array.
   e. Shade the pixel color according to light model.

- Geometric shape choice – the basic shape for enclosing the texture box was a box, but because of the sharp angle between box faces, the envelop was visible from some angles. The improved algorithm use sphere that has smooth angles and lacks this artifact. The cost of using a sphere is high number of vertices.
- Transformation between world space and texture space – the sampling is made in the texture space, but the position is given in world space. For sampling the right pixel in the texture at the desired location a transformation must be done, and it's being calculated from the basic pixel properties- position in world space and in texture space that gives the transformation.

- LOS vector – in pixel shader each pixel has location in world space that is being calculated by interpolation of the vertices of the triangle that the pixel is in, the interpolation between vertex shader and pixel shader is a property of the GPU, the pixel location and head location used to calculate the LOS vector.
- Step size- sampling the texture is made in fixed steps. Choosing step size is tradeoff of calculation complexity for small steps and low accuracy for large steps.
- Texture sampling- the accuracy of the color in the desired locations on the Los vector depends on the sampling method. When sampling the texture on the LOS vector the positions for sample are not discrete like the texture voxels. Choosing the right sampling method is tradeoff of calculation complexity for high quality like in anisotropic interpolation and color accuracy in averaging methods.
- Transfer function- different transfer functions gives different visual effects.
  Threshold function will return the nearest above threshold pixel that has been found and used for surface rendering.
  Accumulative function used to see through and gain the entire data in the LOS vector for seeing inside parts too.
- Light model- accurate light model gives the user the sense of realism and blends the model in the real world using the right shading.

User interface

UI elements

The application user interface consists of 2D menus for selecting the hologram kind (Brain, Heart, Liver images), and for manipulation tools on the hologram for superposition and view modes.

Hand gestures

Application operating done by hand gestures:

1. Double Tap - click on UI elements, like choosing menu item.
2. Drag and Drop – manipulation on the hologram in according to gesture speed and distance for example, scaling, translating and slicing the hologram.

Menus

Main menu – choose between the different holograms in the system.

Manipulation tools menu – used for superposition with the real body part:

1. Translation in the world coordinates.
2. Scale
3. Rotation in 3 axes
4. Slicing- slice the hologram to see the inner image slices data.
5. Threshold- change the transfer function threshold for changing data layers.
6. Change transfer function- see vessels inside tissue in the liver hologram

Challenges and solutions

Hardware limitations

The HoloLens are wearable computer and the current edition has limited hardware capabilities. HoloLens's application nature is of 3D content for creating the illusion of holographic content and most of the applications for HoloLens are made with Unity framework to the ease of use and the overhead that Unity provides. The first attempt for creating the application was made in Unity but the Hardware limitations made the hologram very limited and with low quality.

The solution was to write shaders code and use the GPU capabilities for interpolations and for parallel computing. Working with DirectX on HoloLens is very rare, and lack of documentation and the development was unique to this application. The volume rendering algorithm had to be customized and to simplify the calculations for responsive application in one hand and for graphic quality in the other hand. In the next generation of HoloLens, the application could maybe use more complex algorithms for better graphic quality.

Memory- most of the volume rendering calculations are made in the GPU pixel shader, for working on the GPU the whole texture should be load to buffer on the GPU. Because of memory limitation, not all image sequences were able to upload fully with in RGB space, and shrinking the data was part of the application development.

Coordinate systems

To match between the real world and the augmented reality content, the device scans the environment of the user and set world coordinate system for placing the holograms in reasonable location within the space. In addition, the texture has its own coordinate space, and conversion must be made for each pixel for sampling the texture in the right position.

LOS direction

When rotating the model, the LOS also changed, probably because of the camera implementation of the 3D content in DirectX and the application for HoloLens. The solution was the rotate the head position in the other direction for creating the right feel of model rotation.