Immersive inner-tissue visualization based on Optical Coherence Tomography (OCT) scans

Orly Liba, Spring 2017

Motivation

Optical Coherence tomography (OCT) [1] is capable of imaging inside living animals with very high resolution. OCT scans produce three-dimensional (3D) volumes, rather than two-dimensional (2D) images. Therefore, some of the information and the sense of depth are lost when they are displayed as simple two-dimensional images. **The goal of this project is to visualize OCT scans in 3D on a head mounted display (HMD)**. **This will be done by adapting OCT scans to** Unity**: a broad-application platform that supports virtual reality (VR).**

Possible applications

Several applications for this method of visualization are:

- Diagnostics: Improved visualization of the 3D structures can produce better diagnostics.
- Education: Enable learning about tissue structure and anatomy using an immersive 3D visualization.
- A better and more complete representation of OCT data at conferences and papers. Enabling 3D interactive visualization of OCT data would convey the OCT scans more reliably because information is not lost as a result of projection or slicing onto a 2D image.
- Intraoperative OCT (long-term application). HMDs displaying OCT data may assist surgeons with seeing under the surface of tissue. The use of HMDs for displaying OCT data is currently being developed for retinal surgeries[2–4]. Intraoperative OCT would require scanning, reconstruction and rendering to be done in real-time.

Scope of project

This project focuses on displaying pre-acquired OCT volumes on a HMD using Unity. The benefit of using Unity is that most HMD platforms provide SDKs for Unity. Therefore, it will be relatively simple to view the OCT volumes on different HMDs and utilize their various tracking capabilities, such as orientation, position and controller tracking.

Here we demonstrate the immersive visualization of OCT volumes on YouTube's stereoscopic panorama, which can be viewed on most smart-phones and only requires a very simple HMD such as Google Cardboard. This platform uses the gyroscope and accelerometer inside the cell-phone for orientation tracking. Several demos are described at the end of this document.

Recipe for displaying OCT volumes in Unity and visualizing them in virtual-reality

A) Preliminary steps (can be done with Matlab):
   1. Reconstruct the OCT volumes[5].
   2. Detect blood vessels using speckle variance[5] or any other angiography method[6].
   3. Save the volumes as .tif stacks

B) The .tif stack includes the volume represented by pixel values in a 3D pixel location. In order to view the volume in Unity, we are going to represent the volume as a series of 2D images with transparency. These images will be the texture for a uniform three-dimensional grid, which is described in an .obj file. Here are the steps, in more detail:
   a. Convert the .tif stack to a series of 2D .png images. The .png images include a parameter called alpha which controls the transparency of the voxel (0 is fully transparent and 1 is opaque). The alpha value for each voxel is determined by a segmentation step (of the vessels or tissue) of the OCT scan. For example, in an OCT volume showing blood vessels, voxels with a low pixel value should not appear black in the final volume – we would like to see through them, so that we can see the structure of the vessels behind them. Therefore, we segment dark pixels to have low alpha and bright pixels to have high alpha. Matlab code is available for this segmentation and conversion to .png.
   b. Write the voxel locations in to an .obj file. Each voxel location in the image stack is equivalent to a vertex. Therefore, each voxel location is written to an .obj file which will “tell” Unity how to construct the volume (model), layer by layer. Matlab code is available.
   c. Assign the .png images to their vertices. Each layer in the volume is a modeled as a plane, and assigned a texture, which is the corresponding png image. The series of png images is mapped onto the planes by a .mat (material) file. The Matlab script that creates the obj, also creates the mat file.

C) Display the volume in Unity
   a. Drag the folder that includes .png, .obj, and .mat files into Unity.
   b. Materials should be automatically generated by Unity for each slice in the volume (shown in the assets folder). If these are not generated, there is probably a problem with the obj or mat files.
   c. The tricky part is how to display the volumes as intended, using the alpha transparency of the volumes. Two existing approaches are:
      i. Volume ray casting (also called volume ray marching)[7–9]
      ii. Texture-based volume rendering[8,9]

      These methods were implemented, demonstrated and compared for microscopy data visualization[10]. A commercial volume viewer for Unity is available from the Unity asset store[11], but currently it requires payment.

      The approach described in this project resembles the texture-based volume rendering method and uses specific shaders which are aready built into Unity:
      - Particles/alpha blended: this shader uses the transparency values to create transparency in the volume and looks good on the volumes of vasculature.
      - Unlit/Transparent cutout: this shader cuts out the transparent areas and shows the surface of the volume. This can be used for showing vessles, but is more useful for showing OCT derived structure.

To apply the shaders to the volume, select all the material files and change their shader in the "inspector" tab.

d. In order to look into the volume, and virtually slice through it, a nice trick is to use the near clipping plane of the camera. We can change the position and orientation of the volume relative to the clipping plane in order to virtually slice through the volume at a variety of angles.

D) Create a 360 (panorama) movie

Unity is a very broad platform and can be used for various applications. Creating a 360 movie is an example of a simple method to display the OCT volumes in "virtual reality", which includes head tracking and stereoscopic rendering. The benefit of recording a 360 video is that it can be displayed on almost any phone + cardboard. Here is a very good tutorial for creating a 360 panorama movie: https://www.youtube.com/watch?v=w-umzg_iLoY

Another option is to build the Unity project into an Android or iOS application.

E) Once you've creates the 360 movie, upload it to YouTube. After it's uploaded, open your movie from a smart-phone using the YouTube app. The movie can be displayed as a regular or stereoscopic 360 movie, both include orientation tracking, so that you can move around the volumes and look at them from different angles. In order to view it on a HMD (such as cardboard), it should be stereo-rendered by clicking the cardboard icon on the YouTube app:

Figure 1 Cardboard icon on YouTube (from https://www.pcmag.com/article2/0,2817,2490498,00.asp)

F) Possible additions:

1. Integrate a controller to control (rotate and move) the volume. Perhaps also to virtually slice the volume and/or reveal different types of OCT reconstruction (angiography, spectral analysis, elastography…)
2. Integrate viewer position tracking (360 panoramas only enable tracking of head orientation, not position)
3. Load a 4D OCT volume, meaning a volume that changes in time, and allow the user to control time progression, as well as volume position.
4. Use Matlab or Unity to design a trajectory inside the blood or lymph vessels and then render a fly-through the vasculature (like a roller-coaster).
5. Real-time OCT rendering for intra-operative purposes. Very challenging to have a high enough frame rate![4]

Results (demo)

- A study of contrast agent detection in lymphatic vessels (from Fig 5 of this 2016 publication[12]): https://www.youtube.com/watch?v=JHRgDJ4y-Vw. The left image shows blood vessels (red) in a mouse pinna before injection of a contrast agent (large gold nanorods, GNRs). The center image shows the same region after a subcutaneous injection of GNRs, which are draining through the lymph vessels (green). The right image shows the same area after a subsequent injection of GNRs with a different scattering spectrum (cyan-blue). Each volume has an area of 4 x 4 mm.

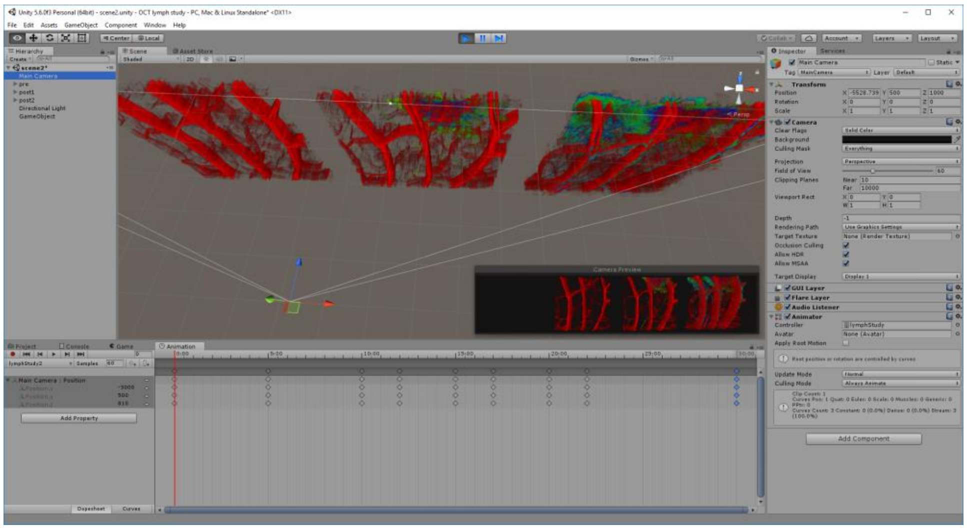

Figure 2 Creating the scene and movie with unity

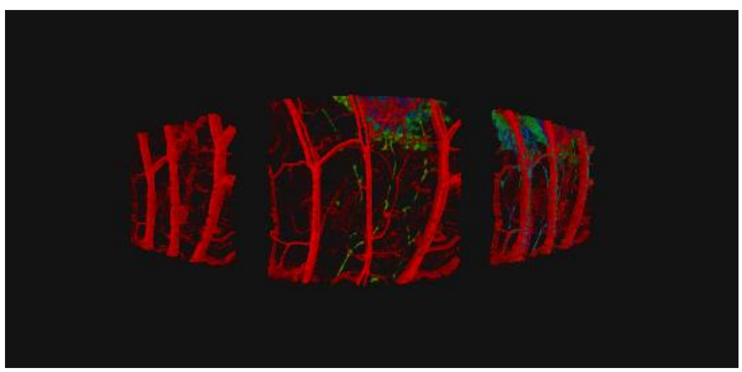

Figure 3 One frame of the 360 panorama movie

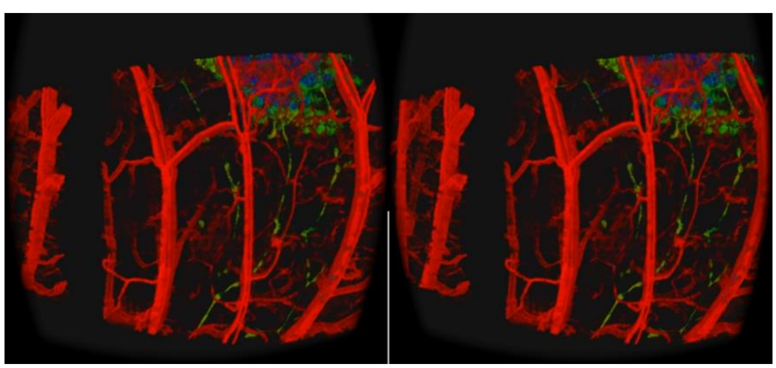

Figure 4 Stereoscopic rendered view on a cellphone

- Mouse brain vessels: https://www.youtube.com/watch?v=Yj0bVawucfo. The brain was imaged through a cranial window. The volume has an area of approximately 5 x 7 mm.

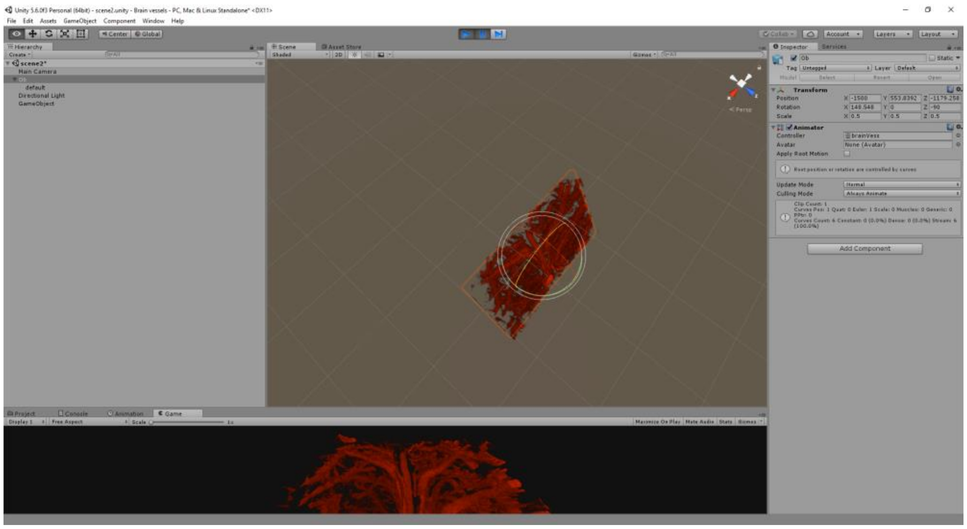

Figure 5 Creating the scene and movie with Unity

- Virtually slicing through a mouse brain which has a tumor: https://www.youtube.com/watch?v=tQI4jTj4erA. Note that the scans were captured while the mouse is alive and no slicing was actually performed. Imaging is done through a cranial window. OCT enables looking up to 2 mm deep inside tissue. The slicing was performed using the near clipping plane – as the volume crosses the clipping plane the deeper layers become visible. We can slice at different angles, by aligning the volume at an angle relative to the near clipping plane. The tumor is at the left side of the field of view and appears brighter compared to its surrounding.

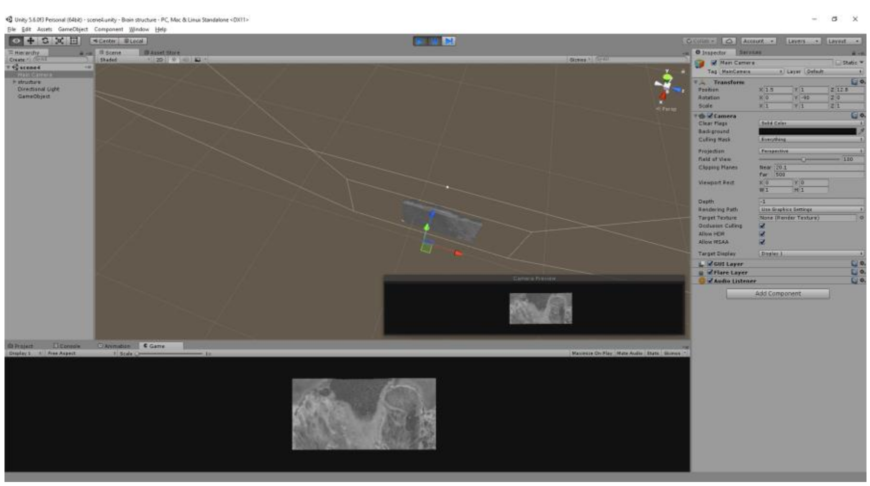

Figure 6 Creating the scene and movie with unity

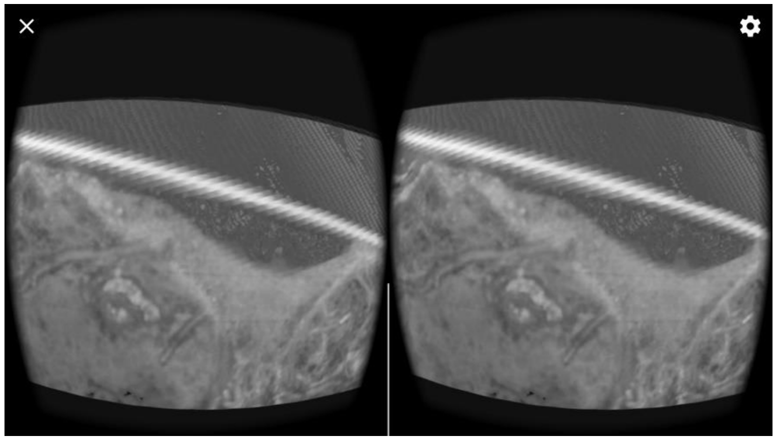

Figure 7 Stereoscopic rendered view on a cell phone

- Vasculature of mouse brain with small tumor, virtually sliced at different angles: https://www.youtube.com/watch?v=mNErrFAFBqs. This visualization uses the Unlit/ transparent cutout shader instead of the particles/alpha blend shader.

Summary and next steps

This project demonstrates a generalizable approach for adapting OCT volumes into Unity, which is a widely used platform for creating virtual-reality content. A possible next step for this project would be to create a more physical rendering for OCT volumes (such as ray marching), and not rely on built-in shaders. This project can be further extended by adding more user interaction, such as manipulating the volumes using a controller.

Bibliography

1. Huang, D. *et al.* Optical coherence tomography. *Science (80-.).* **254,** 1178–81 (1991).
2. Carrasco-Zevallos, O. M. *et al.* Review of intraoperative optical coherence tomography: technology and applications [Invited]. *Biomed. Opt. Express* **8,** 1607 (2017).
3. Viehland, C. *et al.* Enhanced volumetric visualization for real time 4D intraoperative ophthalmic swept-source OCT. *Biomed. Opt. Express* **7,** 1815 (2016).
4. Carrasco-Zevallos, O. M. *et al.* Live volumetric (4D) visualization and guidance of in vivo human ophthalmic surgery with intraoperative optical coherence tomography. *Sci. Rep.* **6,** 31689 (2016).
5. Liba, O. OCT reconstruction and spectral analysis code. Available at: https://github.com/orlyliba/OCT_Reconstruction_and_Spectral_Analysis.
6. de Carlo, T. E., Romano, A., Waheed, N. K. & Duker, J. S. A review of optical coherence tomography angiography (OCTA). *Int. J. Retin. Vitr.* **1,** 5 (2015).
7. Lorensen, W. E. & Cline., H. E. Marching cubes: A high resolution 3D surface construction algorithm. *ACM siggraph Comput. Graph.* **21,** (1987).
8. Engel, K. *Real-time volume graphics*. (A K Peters, 2006).
9. Movania, M. M. *OpenGL development cookbook : over 40 recipes to help you learn, understand, and implement modern OpenGL in your applications*. (Packt Pub, 2013).
10. Theart, R. P., Loos, B. & Niesler, T. R. Virtual reality assisted microscopy data visualization and colocalization analysis. *BMC Bioinformatics* **18,** 64 (2017).
11. Volume Viewer Pro - Asset Store. Available at: https://www.assetstore.unity3d.com/en/#!/content/83185.
12. Liba, O., SoRelle, E. D., Sen, D. & de la Zerda, A. Contrast-enhanced optical coherence tomography with picomolar sensitivity for functional in vivo imaging. *Sci. Rep.* **6,** 23337 (2016).