2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW) | 978-1-6654-8402-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/VRW55335.2022.00235

Using Direct Volume Rendering for Augmented Reality in Resource-constrained Platforms

Berk Cetinsaya* Carsten Neumann& Dirk Reiners+
University of Central Florida

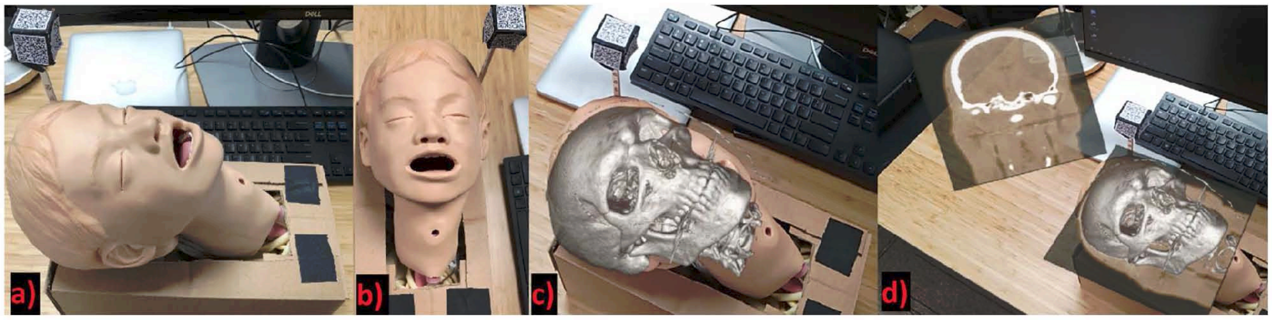

Figure 1: a-b: Mannequin head with tracker markers side view and top view. c: Holographic view. d: Holographic view with slice planes.

Abstract

Rendering a large volume is a challenging task on mobile and Augmented Reality (AR) devices due to lack of memory space and device limitations. Therefore, we implemented an Empty Space Skipping (ESS) optimization algorithm to render the high-quality large models on HoloLens. We designed and developed a system to visualize the computerized tomography (CT) scan data and Digital Imaging and Communications in Medicine (DICOM) files on Microsoft HoloLens 2. We used the Unity3D game engine to develop the system. As a result, we achieved about 10 times more frames per second (fps) on a high-quality model than the non-optimized version.

Index Terms: Computing methodologies—Modeling and simulation—Simulation types and techniques—Real-time simulation; Computing methodologies—Computer graphics—Graphics systems and interfaces—Mixed / augmented reality; Computing methodologies—Computer graphics—Shape modeling—Volumetric models

1 Introduction

Volumetric rendering is commonly used in several fields of medicine to visualize the human body [1]–[3]. Processing and analyzing volumetric data have several computational challenges since the data might be huge and needs lots of resources to render [4]. In light of the current studies, Augmented Reality (AR) can bring real improvement to volumetric data analysis and visualization. AR technologies are used in engineering, industrial, and construction projects for project visualization [5], [6].

*e-mail: cetinsaya@knights.ucf.edu
&e-mail: carsten.neumann@ucf.edu
+e-mail: dirk.reiners@ucf.edu

We present a method to render high-quality volumetric models on the HoloLens 2 that improves existing methods by about 10 times and to show the design and development cycle of the system with the analysis of the performances on HoloLens 2.

2 Methods

Ray marching is one of the basic methods for direct volume visualization. In this technique, a ray is generated for each voxel. The ray starts at the center of the camera and passes through the voxels on the imaginary image plane floating in between the camera and the volume to be rendered.

In this study, we used the Microsoft HoloLens 2 because it is widely acclaimed as the best commercial and powerful AR system [7] and provides good rendering resolution of 2048x1080 per eye and 52 degrees field of view. We also used Unity3D game engine v2019.4.26f1 and MRTK for Unity v2.7.2 for the development of the project, Visual Studio 2019 v16.10.0 for the deployment of the project into HoloLens 2 and Vuforia Engine v9.8 for tracking.

2.1 Volumetric Rendering Algorithm and Optimization

We used Matias Lavik's algorithm [8] as our main project for the volumetric rendering for being open source and for being able to render volumetric data on the Unity3D environment. We made some changes to enable the holographic rendering. He provides three different rendering modes: Direct Volumetric Rendering, Isosurface rendering, and Maximum Intensity Projection. We used Isosurface rendering for our case. Isosurface rendering draws the first voxel the ray hits, with a density higher than some threshold. However, it was not enough to render a high-quality 128x256x256 model at an acceptable frame rate. Thus, we applied additional optimization techniques to be able to render a high-quality model on HoloLens 2 without losing the quality.

First, we added an imposter, an existing technique in computer graphics, to the scene. Since the impostor allows varying the resolution (and frequency) at which the volume is ray marched from the display resolution and frequency, it is possible to increase the fps. Imposters are basically similar to billboards that represent 3D objects that you just swap them with 2D billboard representation when you don't need to render them in full 3D. Imposters are slightly more advanced than normal billboards in that they often provide different images depending on the angle from which the object is viewed. It was not still enough on its own to render smoothly, therefore; we implemented an Empty Space Skipping (ESS) [9]–[11] algorithm to the project.

ESS allows the ray marching to more efficiently traverse empty (transparent) regions of the volume that do not contribute to the rendered image. The shader does not know this information by itself. When we generate the volume, we create a pre-calculated distance map to store the shortest distance from an opaque voxel to an empty one. By this, we will send rays to the volume without rendering all voxels in the volume data. It will skip the empty ones and hit an opaque voxel immediately. We will show the performances in the results section.

3 Results

We created our volumetric model by using the VisMale Head from Visual Human Project, National Library of Medicine. The VisMale Head is a 128x256x256 CT scan of the Visible Male Head. All rendering performance numbers are for the HoloLens 2, while the precomputation of the distance map is done on a desktop workstation with an Intel Xeon E-2124G processor, 32 GB main memory, and NVIDIA Quadro P5000 graphics card.

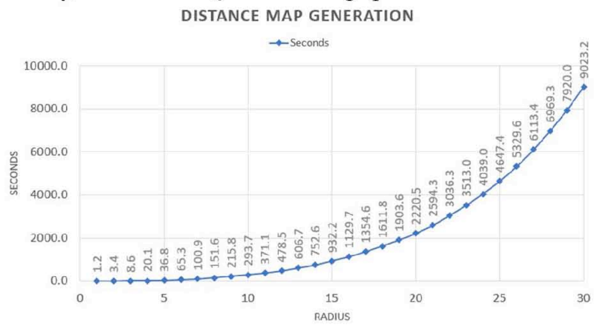

Figure 2: The times take to generate a distance map with specified radius in seconds.

To apply ESS, a radius should be defined to create a distance map. The distance map generation times can be seen in Figure 2. The radius is the ray's maximum travel distance. For instance, assume that we define the radius as 5. First, we will write 5 to all voxels in the volume. Then, our algorithm will check the voxel alpha values. When we find an opaque voxel, we write 0 in our distance map corresponding to the opaque voxel we found. After that, we start to look for the neighbors in the defined radius to calculate the distances from an opaque voxel to the closest empty voxel. After reaching an empty voxel, we store the distance value of this voxel in our distance map. If we do not reach any empty voxels by a given radius in neighbors, for our case is 5, then we will put 5 as our distance value which will be the maximum voxel value that will be skipped in our shader until reaching an opaque voxel. According to our results, the generation of a distance map with a radius of 5 will take approximately 36.8 seconds. Increasing the radius values will increase the frames per second (fps) since the skipped empty voxel numbers (not rendered voxels) will also increase.

The non-optimized model, without imposters and ESS, runs at 2-4 fps, only the imposter model runs at 18-20 fps. And, on the imposter + ESS model, we achieved to increase the fps to 25-40 which is about 10 times more than the non-optimized model without losing any quality with real-time tracking.

4 Conclusion and Limitations

We proposed a system to visualize computerized tomography (CT) scan data on Microsoft HoloLens 2 in real-time. We designed and developed an ESS optimization model for the volumetric rendering with real-time tracking. This project can be used in emergency rooms (ER) and operating rooms (OR) or to train medical students, interns, and resident physicians since research shows that AR projects increase the success rates of students [12].

There are some worst-case scenarios while using the ESS. For example, ESS would not be as helpful if there are no transparent voxels in the volumetric data or some portion of the data. However, it is still promising and worth using on AR devices since they do not have powerful GPUs to render high-quality large volume data. It is worth mentioning that we achieved 60 fps by downscaling the model by half (64x128x128) on HoloLens 2 with the trade-off quality. However, we did not include it in the results because it is obvious that if you reduce the volumetric data size, the performance increases. Our goal was the increase the fps without any quality loss on a large volume by implementing the ESS optimization model to our project and we succeeded.

References

[1] D. R. Ney, E. K. Fishman, D. Magid, and R. A. Drebin, "Volumetric rendering of computed tomography data: principles and techniques," *IEEE Comput. Graph. Appl.*, vol. 10, no. 2, pp. 24–32, Mar. 1990, doi: 10.1109/38.50670.

[2] X. Hu *et al.*, "Volumetric rendering of multimodality, multivariable medical imaging data," in *Proceedings of the 1989 Chapel Hill workshop on Volume visualization*, New York, NY, USA, May 1989, pp. 45–49. doi: 10.1145/329129.329359.

[3] E. K. Fishman *et al.*, "Volumetric rendering techniques: applications for three-dimensional imaging of the hip.," *Radiology*, vol. 163, no. 3, pp. 737–738, Jun. 1987, doi: 10.1148/radiology.163.3.3575725.

[4] M. Smelyanskiy *et al.*, "Mapping High-Fidelity Volume Rendering for Medical Imaging to CPU, GPU and Many-Core Architectures," *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 6, pp. 1563–1570, Nov. 2009, doi: 10.1109/TVCG.2009.164.

[5] X. Li, W. Yi, H.-L. Chi, X. Wang, and A. P. C. Chan, "A critical review of virtual and augmented reality (VR/AR) applications in construction safety," *Autom. Constr.*, vol. 86, pp. 150–162, Feb. 2018, doi: 10.1016/j.autcon.2017.11.003.

[6] F. S. Irwansyah, Y. M. Yusuf, I. Farida, and M. A. Ramdhani, "Augmented Reality (AR) Technology on The Android Operating System in Chemistry Learning," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 288, p. 012068, Jan. 2018, doi: 10.1088/1757-899X/288/1/012068.

[7] L. Roitman, J. Shrager, and T. Winograd, "A Comparative Analysis of Augmented Reality Technologies and their Marketability in the Consumer Electronics Segment," 2017, doi: 10.4172/2155-6210.1000236.

[8] M. Lavik, *UnityVolumeRendering*. 2021. Accessed: Dec. 21, 2021. [Online]. Available: https://github.com/mlavik1/UnityVolumeRendering

[9] J. Kruger and R. Westermann, "Acceleration techniques for GPU-based volume rendering," in *IEEE Visualization, 2003. VIS 2003.*, Oct. 2003, pp. 287–292. doi: 10.1109/VISUAL.2003.1250384.

[10] W. Li, K. Mueller, and A. Kaufman, "Empty space skipping and occlusion clipping for texture-based volume rendering," in *IEEE Visualization, 2003. VIS 2003.*, Oct. 2003, pp. 317–324. doi: 10.1109/VISUAL.2003.1250388.

[11] D. Cohen and Z. Sheffer, "Proximity clouds — an acceleration technique for 3D grid traversal," *Vis. Comput.*, vol. 11, no. 1, pp. 27–38, Jan. 1994, doi: 10.1007/BF01900697.

[12] M. Selek and Y. E. Kıymaz, "Implementation of the augmented reality to electronic practice," *Comput. Appl. Eng. Educ.*, vol. 28, no. 2, pp. 420–434, 2020, doi: 10.1002/cae.22204.