Volumetric Medical Data Visualization for Collaborative VR Environments

Roland Fischer, Kai-Ching Chang, René Weller, and Gabriel Zachmann

University of Bremen, Bremen, Germany
{rfischer,weller,zach}@cs.uni-bremen.de, kchang@uni-bremen.de
https://cgvr.informatik.uni-bremen.de

Abstract. In clinical practice, medical imaging technologies, like computed tomography, have become an important and routinely used technique for diagnosis. Advanced 3D visualization techniques of this data, e.g. by using volume rendering, provide doctors a better spatial understanding for reviewing complex anatomy. There already exist sophisticated programs for the visualization of medical imaging data, however, they are usually limited to exactly this topic and can be hardly extended to new functionality; for instance, multi-user support, especially when considering immersive VR interfaces like tracked HMDs and natural user interfaces, can provide the doctors an easier, more immersive access to the information and support collaborative discussions with remote colleagues. We present an easy-to-use and expandable system for volumetric medical image visualization with support for multi-user VR interactions. The main idea is to combine a state-of-the-art open-source game engine, the Unreal Engine 4, with a new volume renderer. The underlying game engine basis guarantees the extensibility and allows for easy adaption of our system to new hardware and software developments. In our example application, remote users can meet in a shared virtual environment and view, manipulate and discuss the volume-rendered data in real-time. Our new volume renderer for the Unreal Engine is capable of real-time performance, as well as, high-quality visualization.

Keywords: Volume Rendering · Medical Visualization · Virtual Reality · Collaborative VR · Computed Tomography · Unreal Engine

1 Introduction

Computed tomography (CT) is a vital examination tool in medicine, especially for radiologists, and widely used in clinical practice. Its use cases range from diagnosis and therapeutics to preventive medicine and screening of diseases. CT images are, for example, commonly used for visualization purposes in tumor board reviews or for postmortem imaging in forensic pathology. 3D visualization of the CT data is rarely taken advantage of yet. However, it is slowly getting more important. Due to rising processing power and continuous research in algorithms and rendering techniques, faster and more advanced 3D visualization techniques are developed. The main benefit is the more intuitive, three-dimensional visualization of the data. This makes it easier and faster to get an overview of the data and an understanding of the spatial relations, volumes, and general layout of the depicted objects. This is helpful for analyzing complex anatomy or conveying medical situations in an easy-to-understand way. Typical 3D visualization techniques are maximum/minimal intensity projection (MIP/MinIP), surface shaded display (SSD), also called indirect volume rendering, and direct volume rendering (DVR). SSD shows opaque three-dimensional surfaces, called isosurfaces, of specific objects or organs in the volume data determined by a density-dependent segmentation. DVR accounts for the possibility of multiple tissue types per voxel and maps the densities to opacities and colors using transfer functions. This results in a semi-transparent rendering [8,9].

Currently, both 2D and 3D CT reconstructions are typically viewed on 2D screens or projectors, which limits the advantages of volumetric visualizations. On the other hand, Virtual Reality (VR) devices such as the HTC Vive become popular in many fields as they provide immersive stereoscopic visualizations with intuitive user interfaces and novel cooperative multi-user capabilities. VR offers a natural progression over previous 2D telepresence tools and leads to a new quality of collaborative work, as users can meet and intuitively interact with virtual objects as well as with each other in a shared virtual 3D environment. This makes VR an important tool for the entertainment industry but also industrial, educational, and medical applications. For example, a current trend is to use VR for simulators in which users can be trained and educated realistically and in a safe virtual environment (e.g. laparoscopy, heart surgery, and even orthopedic operations [15]). These benefits and the increasing display resolutions of newer headsets make VR in general, and multi-user VR particularly, well suited for use cases like inspection and discussion of volumetric medical data and corresponding 3D visualizations as part of diagnosis or pre-operative planning [21].

VR applications and their virtual environments are typically created and powered by 3D graphics engines like Unity or the Unreal Engine which provide features such as high-quality graphics and automatic VR integration. However, they are usually mesh/polygon-based and, out of the box, do not support volume rendering.

We propose a system based on the Unreal Engine 4 in which multiple users can collaboratively inspect and interact with volume-rendered CT data in real-time within a VR environment resembling an operating room. For this purpose, we combine mesh- and volume rendering into an immersive multi-user application. This includes a custom direct volume renderer for the Unreal Engine and several optimization and lighting techniques to achieve real-time performance as well as a good visualization quality. Additionally, we have developed a custom pipeline for processing CT images allowing easy and effective visualization of multiple windows in parallel.

2 Related Work

Volume rendering is a promising tool for medical visualization as it proved to be useful for planning of surgical treatment of nasal bone fractures [27], acetabular fractures [29], virtual endoscopy [16] or the visualization of complex anatomy such as the ossicular chain in chronic suppurative otitis media [12]. Recently, a direct volume rendering approach for serial PET–CT scans that preserves anatomical consistency was presented [14]. The high computational effort of direct volume rendering can be mitigated by algorithmic optimizations, e.g., early ray termination and empty space skipping [25]. Also, the visual quality can be improved, e.g., by applying local ambient occlusion [13]. Berger et al. [2] have shown that the novel, more complex cinematic rendering technique provides a superior visualization to the classic volume rendering using ray casting, however, the significantly slower computation is still a challenge. Ryan Brucks [3] developed a custom volume rendering implementations for the Unreal Engine 4, however, it is only rudimentary and not designed for medical data, leading to artifacts.

Several evaluations show that VR can be beneficial in a wide range of medical applications, foremost simulators for training of different surgical procedures [17,23]. Often, medical imaging plays a central role in these applications: e.g., Maloca et al. [19] proposed an OpenGL-based immersive VR system for real-time volume rendering of Optical Coherence Tomography data. An accompanying study suggested that it could be helpful for education and preoperative planning. Similarly, Scholl et al. [26] developed a medical VR application for 3D visualization based on volume rendering. Real-time performance is achieved by the use of several acceleration and optimization techniques. Adams et al. [1] used the Unity 3D engine to develop an immersive VR application for medical imaging in which CT images and corresponding, segmented 3D models can be viewed and manipulated. Magdics et al. [18] also used Unity to develop an educational VR application in which volume rendering is used for visualizing Nasal Cavities. Faludi et al. [11] presented a VR application that uses not only direct volume rendering but also haptic rendering of medical data. However, non of these systems support multiple users or collaborative work, which is another popular and promising research area.

Regarding collaborative medical VR, Cecil et al. [4] developed a system for orthopedic surgery. Similarly, Paiva et al. [22] presented a VR simulator for surgical team training. Chheang et al. [5] proposed a promising collaborative VR system for planning and simulation of laparoscopic liver surgery, Christensen et al. [6] positively evaluated the feasibility of team training in VR for robot-assisted minimally invasive surgery, and Elvezio et al. [10] designed a VR system for collaborative symmetric and asymmetric interactions and found that low latencies (below 15 ms) are crucial for effective collaboration. These works, however, do not feature 3D visualization of CT data.

3 Our Approach

The goal of our system is to combine the benefits of collaborative VR and medical 3D visualization to an immersive, interactive application based on a modern, extensible open-source 3D game engine, specifically the Unreal Engine 4. As the engine does not support volume rendering out of the box, we have developed and integrated a ray-marching-based volume renderer, based on Ryan Brucks' rudimentary implementation [3], focusing on a good trade-off between speed and visual quality.

We have decided to use the Unreal Engine 4 for several reasons: first, it is known for its high graphical fidelity, second, it supports most available VR devices like the HTC Vive with a platform independent interface, and it has networking capabilities included. Moreover, due to its open-source implementation, it can be easily extended with native C++ programming but also offers an easy graphical programming interface via Blueprints. We decided to directly benefit from Unreal's networking architecture, hence, we use a client-server model enabling users to host and join sessions via a lobby system, whereby the first client acts also as a server. An overview of the whole system design is shown in Figure 1.

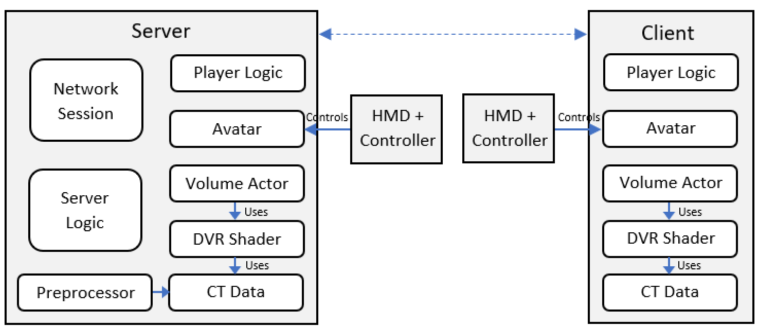


Fig. 1: System architecture of our application. The first client acts also as a server.

The CT data requires a preprocessing step to be loaded into the Unreal Engine. The processed data can be rendered seamlessly into the polygonal scene using our shader-based direct volume rendering solution. Our DVR approach achieves real-time performance guaranteeing a smooth VR experience. In the following, we will describe the individual parts of our system in detail.

3.1 Direct Volume Rendering

In order to visualize the CT data in our Unreal Engine-based virtual environment, we opted for a direct volume rendering approach based on ray marching.

Our pipeline is specifically designed for the visualization of CT data, thus, the first step is to read and process the CT DICOM files in a preprocessing phase. To map the density to opacity, we employ multiple, freely adjustable, default windows with corresponding transfer functions. The advantage of having multiple windows is that each feature captured by a window can be visualized with high contrast. To store the windows in a single grayscale image (8 bit) we decided to blend the windows similar to the RADIO algorithm by Mandell et al. [20], which maintains the relative attenuation relationships between the fundamental anatomic densities and thus accommodates radiologists and their expectations. Figure 2 depicts the underlying concept. However, any other blending algorithm would be compatible too. Additionally, the volumetric data set has to be transformed into a format suitable for import and further processing in the Unreal Engine, therefore, we arrange the individual 2D slices of the volume sequentially into sequence maps (see Fig. 3).

We have implemented the ray casting directly in a pixel shader. We use a unit cube as a geometrical proxy mesh and reconstruct the volume coordinates from the generated sequence maps. In order to avoid artifacts of box-aligned samples (left diagram in Fig. 4), we align the first sampling points to stacked view-aligned planes instead (middle diagram in Figure 4). Additionally, we precompute the sampling step length and the maximal number of samples fitting in the volume outside of the ray casting loop to reduce overhead. The calculation is based on the CT data set's proportions, the ray's accordingly adjusted starting position, and a user-adjustable factor allowing for arbitrary changes to the sampling rate. More details about the aforementioned handling of the sequence maps and the general sampling procedure can be read in Ryan Brucks' volume rendering guide for the Unreal Engine 4 [3]. We provide the possibility to apply stochastic jittering and a 2×2 ordered grid supersampling to improve the visual quality (see the right diagram in Fig. 4) .

Regarding convincing yet fast shading and shadowing, we opted to implement a couple of different and not too complex local illumination methods and compare the visual results. Firstly, at each sample position, we cast shadow rays to determine the amount of occlusion. For this purpose, we dynamically track the position of multiple light sources. This method enables proper self-shadowing

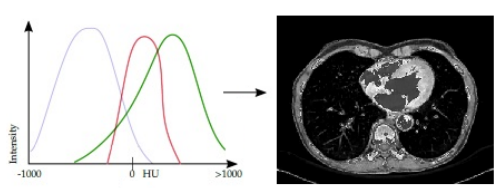


Fig. 2: Window blending according to the RADIO algorithm. Left: bone, lung, soft tissue windows. Right: blended CT image.

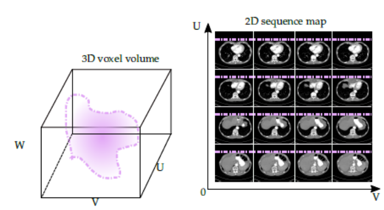


Fig. 3: Right: sequence map of CT slices. Left: corresponding reconstruction in the shader.

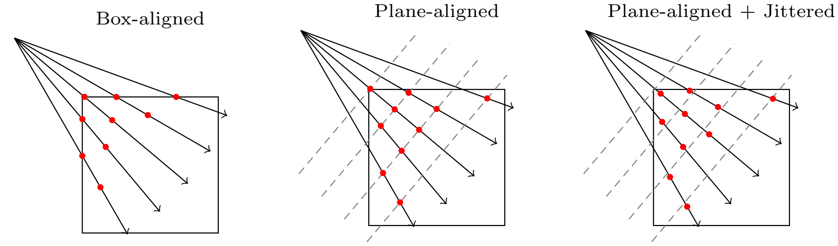


Fig. 4: Sampling positions in the volume. Left: in the naive approach the start sample positions align with the box mesh and cause patterned artifacts throughout the volume. Middle: sampling positions on equidistant view-aligned planes. Right: the sample positions are additionally jittered along the ray axis.

from multiple dynamic lights, however, is rather computational expensive. Therefore, we lower the shadow rays' sampling frequency in contrast to the primary rays'. Secondly, we implemented the classic Blinn-Phong shading model that is evaluated at each sampling position. It is rather cheap to compute and enables local lighting approximation by diffuse and specular reflections which can be configured on a per-material-basis. We approximate the needed surface normals, which are not present in CT data, based on the local gradient using the central differences technique in the preprocessing phase. Lastly, we also implemented volumetric local ambient occlusion (LAO). Here, the sampling point is shaded based on the amount of occlusion, which is estimated by the opacities of the local neighborhood. This method can be used to prevent full shadows, which may obscure fine details. Another advantage is that it is not based on gradients, which are often not well defined (e.g. in homogeneous regions) and susceptible to noise. Algorithm 1 outlines the raycasting process.

To increase the performance, we reduce the number of samples being taken by early ray termination and empty space skipping using an octree. We construct the octree using a pointer-free branch-on-need strategy and encode it in a texture during the preprocessing phase, as the data is static. During sampling, the octree is traversed top-down similar to the parametric approach described in [24].

Algorithm 1 Shader-Based Raycasting

```
for each pixel (in parallel) do
    compute firstSamplePos and maxSamples
    while maxSamples not reached and accumOpacity below 1 do
        sample position using opacity sequence map
        calculate base color depending on opacity
        update accumOpacity by composition with sample opacity
        evaluate selected lighting method(s) and update color
        update accumColor by composition with sample color
        increment sample position
```

3.2 Collaborative VR

Generally, we made use of Unreal Engines polygonal and stereo rendering capabilities and VR support to build our application. To create a believable virtual environment, thus, enhancing the immersion and the experience for the users, we build a 3D scene resembling an operation room in which the users can interact. Similarly, users are represented by static mesh avatars modeled after doctors in a medical outfit. Our avatars consist of separate models for the head and hands; their corresponding positions are tracked directly by the HMD and the accompanying controllers. To avoid issues with possibly faulty and distracting animations by inverse kinematics, we refrain from using whole-body skeletal meshes. Each user can be identified by a personal name shown over the avatar. Figure 5 shows a session with three users inspecting the CT data in the virtual operating room.

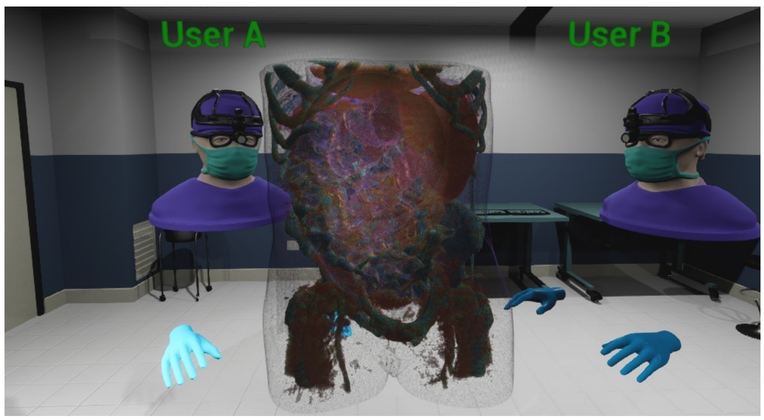


Fig. 5: Several networked users inspecting the 3D visualized CT data in a shared virtual environment.

We included a lobby system with which users can create or search for active sessions, or alternatively join one via a known IP, thus, enabling multiple of these virtual shared environments to exist in parallel. Also, although VR usage is our main focus, VR and non-VR users can mix and collaborate without restriction as we have implemented movement and interaction metaphors for both of them. For example, we implemented physical 3D buttons placed in the scene for VR users and keyboard shortcuts for non-VR users to manipulate properties of the 3D visualization. To reduce the latency between user input and perceived action, which has been shown to be crucial for a positive user experience in previous studies [28], all (inter)actions from users are executed locally first, before being sent and replicated on the server, from which they are finally broadcasted to all remaining users. Furthermore, the Unreal Engine provides some additional latency optimization techniques which help to minimize and stabilize the time needed for communication between client and server.

As a locomotion metaphor for VR users, we decided to use the classical teleportation approach, in combination with room-scale locomotion, as it minimizes the occurrence of motion sickness [7]. A problem arising from using teleportation in a multi-user environment is that the actual process of vanishing and reemerging somewhere else will be confusing for observers as it resembles the typical effects of a slow network connection or network errors. Therefore, we have implemented a particle effect, to highlight the deliberate action of the teleportation process.

To allow for collaborative work between users, we replicate not only their avatars but also the complete state of the 3D visualization of the CT data, making it a single shared object in the scene which is rendered from the individual users' viewpoint. It can be grabbed, moved, and rotated freely and naturally using the controller for optimal (re)view (see Fig. 6). Non-VR users, however, can rotate the object via an orbiting mode. To keep it simple, we do not restrict concurrent manipulation, which internally would be executed sequentially, as users can coordinate themselves. The replicated hands in VR make it easy to point specific spots or areas in the 3D visualization and to make gestures, which help in discussing the data, show findings, or plan interventions.

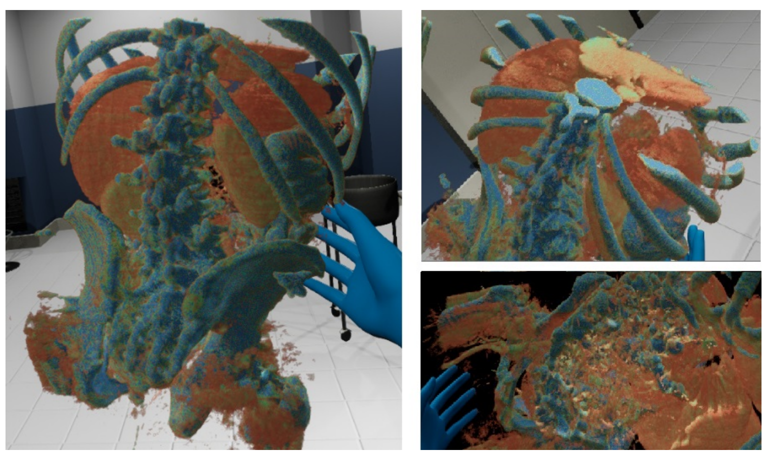
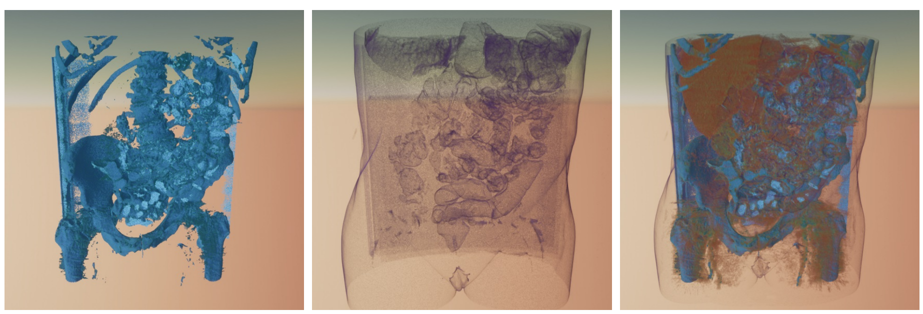


Fig. 6: Several images illustrating how the 3D visualization can be grabbed and freely be moved and rotated for a better view. The background was hidden in the image in the bottom-right.

In addition to the 3D visualization of the medical data, users in our application have the possibility to view accompanying 2D images, e.g. the raw CT data, on a virtual TV in the operating room scene. This may be useful if there is a need to quickly check for specific fine details not visible in the 3D visualization. Finally, the complete scene bar the 3D visualization can be dynamically hidden, resulting in a black background, for an undistracted contrast-rich view.

4 Results

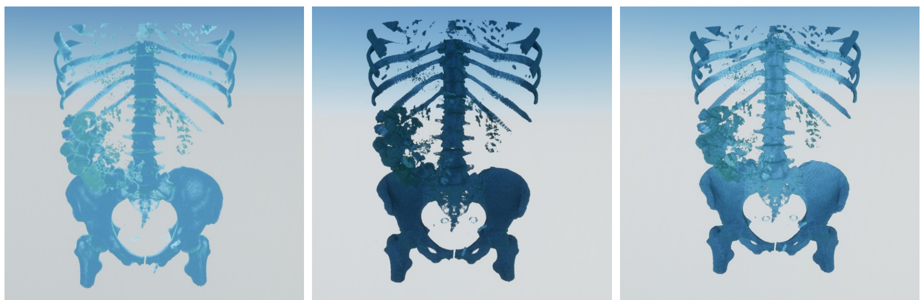
We have evaluated the quality as well as the performance of the main aspects of our approach. In order to show the quality of our volumetric renderer, we visually compare our results to two competing visualization tools. Additionally, we did extensive measurements regarding the performance under various conditions, e.g. different lighting models and optimization methods. For the evaluations, we used several real-world CT data sets obtained by a hospital. The number of slices varies between the data sets and ranges from 47 to 317. We have developed and tested our work based on the Unreal Engine 4.22. Figure 7 shows our volume renderer with different active windows. In the left image, only the bone window is applied. The middle image depicts, among others, inner structures of the liver, small intestine, colon, and skin. Finally, in the right image, all three windows (the third one being soft tissue) are simultaneously visualized. Our DVR is able to effectively render single materials like the bone as well as compositions of multiple materials simultaneously, and thus, the complete range of CT data. This helps in conveying the spatial relationships between organs and getting a good understanding of the data.



(a) (b) (c)

Fig. 7: Our volume renderer applied to a CT data set using different windows: bone (a), bowel and skin (b), and soft tissue, combined with the previous windows (c).

Figure 8 depicts our renderer with the different lighting settings. The left image shows a bone window using only shadow rays. Self-shadowing can be seen which helps in conveying depth, however, because of the limited sampling rate for shadow rays, the shadows are coarse and imprecise. In the middle image, we switched on the Blinn-Phong lighting model. A possible issue with this technique is, that, depending on the position of the light source relative to the visualized object, areas may lie completely in the shadows, and thus, can be hard to inspect if no additional ambient lighting is applied. The right image, however, shows the combination with the local ambient occlusion technique. This combination circumvents the problem of full shadows and results in the best lighting. The transition between being in complete light and full shadow is the most fine-granular and accounts for the local neighborhood providing the best depth perception and understanding of object shapes.



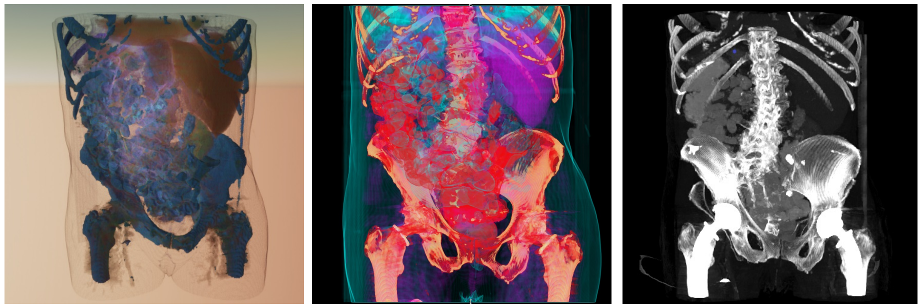
(a) (b) (c)

Fig. 8: Our volume renderer using different illumination methods: shadow rays (a), additional Blinn-Phong lighting (b), both combined with LAO (c). As can be seen, the latter enhances the depth perception by superior shadowing.

Figure 9 illustrates a comparison of our renderer (first image) with the common visualization tools RadiAnt DICOM Viewer in the standard 3D volume rendering mode (second image), and the Visualization Toolkit (VTK) with maximum intensity projection as a composition scheme (third image). The comparison shows that our renderer generates visualizations which are very effective in conveying a perception of depth and giving a clear and understandable overview of the data set as a whole. At the same time, our renderer produces precise, plastic visualizations of the individual materials. VTK uses MIP which results in relatively flat images with missing details. The advanced lighting and shading of our renderer make the assessment of the spatial relations between the objects easy. Although the results by RadiAnt are very good too, they tend to exhibit slightly stronger artifacts and a simpler shading is used.

A performance evaluation was done on a PC with Windows 10, Intel Core i7 4790 CPU, Nvidia Titan V graphics card, 32GB of system memory, and a Full HD monitor. To perform the measurements, we used the native GPU profiler of the Unreal Engine and took the average of multiple runs.

Figure 10 shows the performance of our renderer and the influence of factors such as the number of slices and different lighting methods. In all cases, our renderer outperforms the rudimentary volume rendering solution by Ryan Bruckst [3], independent of the chosen lighting models. Actually, we achieve real-time performance for VR in all our test cases.



(a) (b) (c)

Fig. 9: Comparison of our volume renderer (a) with the visualization tools RadiAnt using 3D volume rendering (b) and VTK using maximum intensity projection (c).

Additionally, we have evaluated the efficiency of our octree implementation for empty space skipping. With the octree, we measured performance improvements for all test data sets of up to 49.4 %. The average improvement was 14.7 %, while the empty space ratio varied between roughly 45 % and 55 %, except for one data set with only 36 %. This shows that our octree implementation is effective in increasing the performance, especially for high-slice data sets.

Finally, we have measured the network performance, specifically the latency. However, in order to get objective and comparable results, we avoided a real internet transmission that is highly dependent on individual factors such as the connection quality or the distance. Instead, we set up client and server on two different computers which were connected via a router and measured the round time of the network messages from client to server and back. The average time was 16.8 ms with a standard deviation of 1.6 ms that is added by our system. Obviously, in case of an internet connection, additional latency have to be added. To conclude, our application is very well suited for collaborative work as actions form other users are replicated quickly. Accordingly, the user feedback, regarding the multi-user VR experience as well as the medical visualization, is very positive so far.

5 Conclusion and Future Work

We have presented a multi-user virtual reality system for medical visualization based on a state-of-the-art game engine that is capable of 3D visualizing computed tomography data in real-time and in a high visual quality. This is achieved by our custom ray-marching-based direct volume renderer which we have implemented using shaders and integrated into the Unreal Engine. Our renderer supports different lighting models, transfer functions selection, and artifact-reducing methods. Our evaluation shows that we achieve VR capable framerates of more

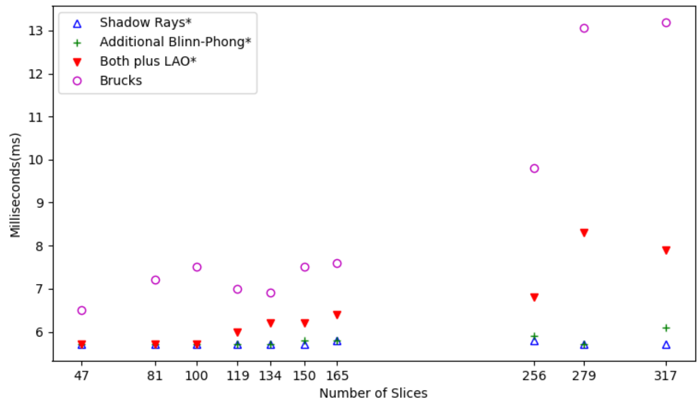


Fig. 10: The Performance using different lighting models and data sets with a varying amount of slices. Our methods are marked with asterisks, "Brucks" is a rudimentary volume rendering integration in Unreal. Even though advanced lighting models increases the computational time, or renderer is real-time capable in all cases and significantly faster than the implementation by Brucks.

than 100 Hz even for complex data sets consisting of more than 300 slices and with advanced lighting features such as ambient occlusion enabled. Our system includes a multi-user component and is designed as a shared virtual environment resembling a real operation room, thus, enabling immersive collaborative work between co-located or remote users. Thanks to the combination of the sophisticated game engine, VR, and our fast high-quality direct volume renderer, users can interact with each other and the shared visualized CT data in an immersive virtual environment and (re)view and discuss the 3D data in a comprehensive natural way. This makes our system ideally suited for pre-operative planning, possibly tumor boards, post-operative evaluation, or patient education.

For the future we plan to expand the interaction possibilities with the volume visualization, specifically, we are looking at integrating a dynamic clipping plane for a better view of internal regions and a volumetric drawing tool allowing for quick sketches and annotations inside the volume. Other improvements would be a direct integration and parallelization of the preprocessing part to speed up the workflow and allowing for a dynamic adjustment of the transfer functions. To improve the visualization of complex structures and organs that involve multiple materials support for multi-dimensional transfer functions could be added.

Acknowledgment

This work was partially funded by the German Federal Ministry of Education and Research (BMBF) under the grant 16SV8077.

References

1. Adams, H., Shinn, J., Morrel, W.G., Noble, J., Bodenheimer, B.: Development and evaluation of an immersive virtual reality system for medical imaging of the ear. In: Medical Imaging 2019: Image-Guided Procedures, Robotic Interventions, and Modeling. vol. 10951, pp. 265 – 272. International Society for Optics and Photonics, SPIE (2019)
2. Berger, F., Ebert, L.C., Kubik-Huch, R.A., Eid, K., Thali, M.J., Niemann, T.: Application of cinematic rendering in clinical routine ct examination of ankle sprains. American Journal of Roentgenology **211**(4), 887–890 (Oct 2018)
3. Brucks, R.: Creating a volumetric ray marcher. https://shaderbits.com/blog/creating-volumetric-ray-marcher (2016), accessed: 2020-07-17
4. Cecil, J., Ramanathan, P., Rahneshin, V., Prakash, A., Pirela-Cruz, M.: Collaborative virtual environments for orthopedic surgery. In: 2013 IEEE International Conference on Automation Science and Engineering (CASE). pp. 133–137 (2013)
5. Chheang, V., Saalfeld, P., Huber, T., Huettl, F., Kneist, W., Preim, B., Hansen, C.: Collaborative virtual reality for laparoscopic liver surgery training (10 2019)
6. Christensen, N., Hjermitslev, O., Stjernholm, N., Falk, F., Nikolov, A., Kraus, M., Poulsen, J., Petersson, J.: Feasibility of team training in virtual reality for robot-assisted minimally invasive surgery. pp. 1–4 (04 2018)
7. Christou, C.G., Aristidou, P.: Steering versus teleport locomotion for head mounted displays. In: Augmented Reality, Virtual Reality, and Computer Graphics. pp. 431–446. Springer International Publishing, Cham (2017)
8. Dappa, E., Higashigaito, K., Fornaro, J., Leschka, S., Wildermuth, S., Alkadhi, H.: Cinematic rendering – an alternative to volume rendering for 3d computed tomography imaging. Insights into Imaging **7** (09 2016)
9. Ebert, L.C., Schweitzer, W., Gascho, D., Ruder, T.D., Flach, P.M., Thali, M.J., Ampanozi, G.: Forensic 3d visualization of ct data using cinematic volume rendering: A preliminary study. American Journal of Roentgenology **208**(2), 233–240 (Feb 2017)
10. Elvezio, C., Ling, F., Liu, J.S., Feiner, S.: Collaborative virtual reality for low-latency interaction. In: The 31st Annual ACM Symposium on User Interface Software and Technology Adjunct Proceedings. pp. 179–181 (10 2018)
11. Faludi, B., Zoller, E.I., Gerig, N., Zam, A., Rauter, G., Cattin, P.C.: Direct visual and haptic volume rendering of medical data sets for an immersive exploration in virtual reality. In: Medical Image Computing and Computer Assisted Intervention – MICCAI 2019. pp. 29–37. Springer International Publishing, Cham (2019)
12. Guo, Y., Liu, Y., Lu, Q.H., Zheng, K.H., Shi, L.J., Wang, Q.J.: Ct two-dimensional reformation versus three-dimensional volume rendering with regard to surgical findings in the preoperative assessment of the ossicular chain in chronic suppurative otitis media. European journal of radiology **82**(9), 1519—1524 (September 2013)
13. Hernell, F., Ljung, P., Ynnerman, A.: Local ambient occlusion in direct volume rendering. IEEE Transactions on Visualization and Computer Graphics **16**(4), 548–559 (2010)
14. Jung, Y., Kim, J., Bi, L., Kumar, A., Feng, D.D., Fulham, M.: A direct volume rendering visualization approach for serial pet–ct scans that preserves anatomical consistency. International Journal of Computer Assisted Radiology and Surgery **14**(5), 733–744 (May 2019)
15. Kaluschke, M., Weller, R., Zachmann, G., Pelliccia, L., Lorenz, M., Klimant, P., Knopp, S., Atze, J.P.G., Möckel, F.: Hips – a virtual reality hip prosthesis im-

plantation simulator. In: 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR) (March 2018)
16. Krüeger, A., Kubisch, C., Strauss, G., Preim, B.: Sinus endoscopy - application of advanced gpu volume rendering for virtual endoscopy. IEEE transactions on visualization and computer graphics **14**, 1491–8 (01 2009)
17. Larsen, C., Strandbygaard, J., Ottesen, B., Sorensen, J.: The efficacy of virtual reality simulation training in laparoscopy: A systemic review of randomized trials. Acta obstetricia et gynecologica Scandinavica **91**, 1015–28 (06 2012)
18. Magdics, M., White, D., Marks, S.: Extending a virtual reality nasal cavity education tool with volume rendering. In: 2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE). pp. 811–814 (2018)
19. Maloca, P.M., de Carvalho, J.E.R., Heeren, T., Hasler, P.W., Mushtaq, F., Mon-Williams, M., Scholl, H.P.N., Balaskas, K., Egan, C., Tufail, A., Witthauer, L., Cattin, P.C.: High-performance virtual reality volume rendering of original optical coherence tomography point-cloud data enhanced with real-time ray casting. Translational vision science & technology **7**(4), 2–2 (Jul 2018)
20. Mandell, J.C., Khurana, B., Folio, L.R., Hyun, H., Smith, S.E., Dunne, R.M., Andriole, K.P.: Clinical applications of a ct window blending algorithm: Radio (relative attenuation-dependent image overlay). Journal of Digital Imaging **30**(3), 358–368 (Jun 2017)
21. Nguyen, B.J., Khurana, A., Bagley, B., Yen, A., Brown, R.K.J., Stojanovska, J., Cline, M., Goodsitt, M., Obrzut, S.: Evaluation of virtual reality for detection of lung nodules on computed tomography. Tomography (Ann Arbor, Mich.) **4**(4), 204–208 (Dec 2018)
22. Paiva, P., Machado, L., Valença, A., Batista, T., Moraes, R.: Simcec: A collaborative vr-based simulator for surgical teamwork education. Computers in Entertainment **16**, 1–26 (04 2018)
23. Pulijala, Y., Ma, M., Pears, M., Peebles, D., Ayoub, A.: Effectiveness of immersive virtual reality in surgical training - a randomized control trial. Journal of Oral and Maxillofacial Surgery **76** (10 2017)
24. Revelles, J., Ureña, C., Lastra, M., Lenguajes, D., Informaticos, S., Informatica, E.: An efficient parametric algorithm for octree traversal (05 2000)
25. Ruijters, D., Vilanova, A.: Optimizing gpu volume rendering. Journal of WSCG **14**(1-3), 9–+ (2006)
26. Scholl, I., Bartella, A., Moluluo, C., Ertural, B., Laing, F., Suder, S.: Medicvr. In: Bildverarbeitung für die Medizin 2019. pp. 152–157. Springer Fachmedien Wiesbaden (2019)
27. Song, S.W., Jun, B.C., Chae, S.R., Kim, B.G.: Clinical utility of three-dimensional facial computed tomography in the treatment of nasal bone fractures: a new modality involving an air-bone view with a volume rendering technique. Indian journal of otolaryngology and head and neck surgery : official publication of the Association of Otolaryngologists of India **65**(Suppl 2), 210–215 (Aug 2013)
28. Waltemate, T., Senna, I., Hülsmann, F., Rohde, M., Kopp, S., Ernst, M., Botsch, M.: The impact of latency on perceptual judgments and motor performance in closed-loop interaction in virtual reality. In: Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology. p. 27–35. VRST '16, Association for Computing Machinery, New York, NY, USA (2016)
29. Wang, H., Wang, F., Newman, S., Yanping, L., Chen, X., Xu, L., Qiugen, W.: Application of an innovative computerized virtual planning system in acetabular fracture surgery: A feasibility study. Injury **47**(8), 1698 – 1701 (2016)