Generating Pseudo Random Volumes for Volumetric Research

Thomas J. Clarke*
IVE, Wearable Computer Lab, University of South Australia

Wolfgang Mayer†
Industrial AI Research Centre, University of South Australia

Joanne E. Zucco‡
IVE, Wearable Computer Lab, University of South Australia

Ross T. Smith§
IVE, Wearable Computer Lab, University of South Australia

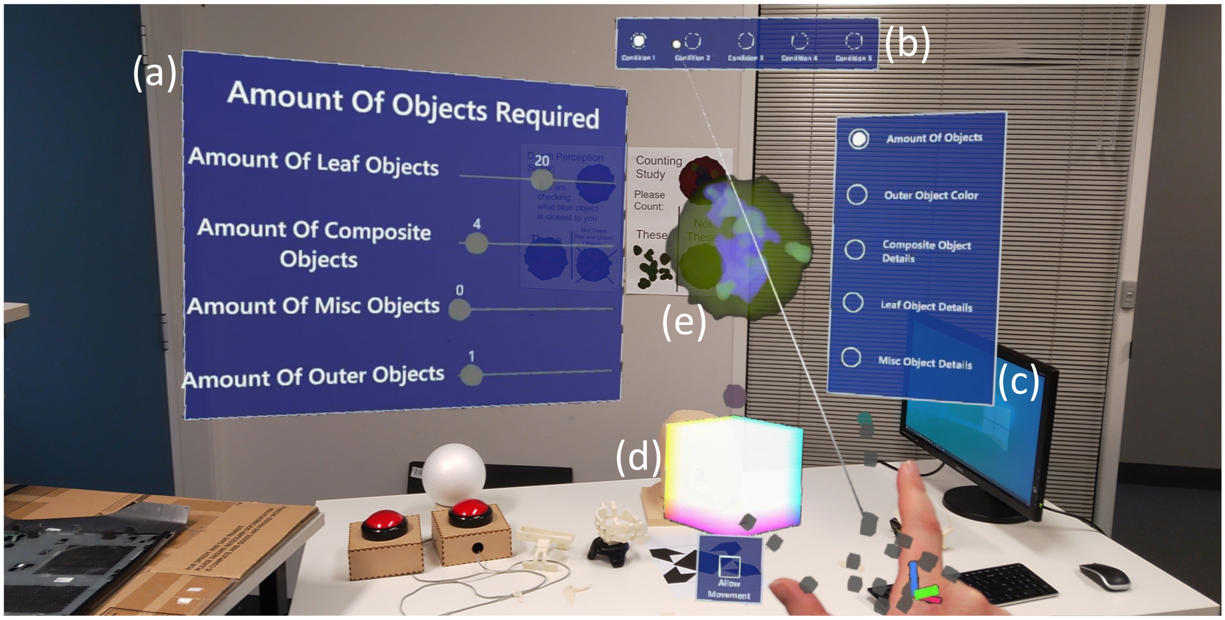

Figure 1: A photo of the interface taken via Microsoft HoloLens of the Immersive system UI. This system utilizes a generic MRTK interface. In the center of the screen is the previous volume created. (b)A radial options menu at the top of the screen allows users to choose which iteration they are editing. (c) On the right-hand side of the menu is a radial list of the different menus users can access to modify various parameters of the volumes. (a)To the left is a window where the currently selected menu lets the user change various volume parameters. (d) At the bottom of the image is a volumetric color picker.

Abstract

Volumetric data is commonly used to understand the structure of an object, like the human body, underground structures, or weather data. However, it is difficult to quite this data as it requires expensive equipment and is difficult to automate. This data is that it is expensive to create and catalog on a large enough scale for a controlled study. We present a solution to this challenge by releasing an open-sourced system designed to create volumes solely designed for user studies for immersive experiences displayed through a VR or AR headset. By utilizing a system like this, we not only have a clear understanding of how a study can be received in MR, but it is possible to have a high level of control over the data allowing for a higher degree of flexibility in the underlying data design. To facilitate this, we have designed a modular system designed to generate and validate volumes, adhere to a rule set and interact within them from a VR or AR interface and evaluate the types of volumes that could be formed.

Index Terms: Human-centered computing—Visualization—Visualization techniques—Treemaps; Human-centered computing—Visualization—Visualization design and evaluation methods

*e-mail: thomas.clarke@mymail.unisa.edu.au
†e-mail: wolfgang.mayer@unisa.edu.au
‡e-mail: joanne.zucco@unisa.edu.au
§e-mail: ross.smith@unisa.edu.au

1 Introduction

Scenes containing volumetric objects are becoming prevalent in Mixed Reality studies [6, 13, 17, 19], but it can be difficult and laborious to create such scenarios that conform to the properties required by the study design. Uncontrolled (Real world) studies can be effective and useful in understanding a real world experience a user might have [11, 18, 33], but are venerable to confounding variables and struggle to provide in-depth answers for questions relating that require precise metrics like fields like depth perception tend to research [11, 13]. Volume data, in particular, struggles with the quantity of similar data, resulting in skewed results [11].

Volumetric data sets for research purposes can be difficult to produce in large quantities. The Open Scientific Visualization Database [1] has a good range and collection of volumetric data well suited to various research problems like rendering, but they are very different and can cause unexpected differences in data [11]. Creating new datasets requires machines like MRI or CT scanners, whose access to these machines can be limited. Current solutions for generating artificial data require precise inputs, are difficult to manage, and are computationally taxing [1, 12, 22, 26]. This paper details a method of making simple-to-understand volumetric data, a method to validate that the data is clearly comprehendible, and a

[1] https://klacansky.com/open-scivis-datasets

modular system allowing for flexible options to evaluate volumetric structures.

Current alternatives for creating controlled studies using volumetric rendering include adding extra information to the limited volumes via adding artificial data. These settings do well to recreate scenarios relating to the introduction of medical instruments to a volume and determine the reliability of these visualizations, but the interactions between other objects become more difficult. Another option can be seen in Englund et al.'s work [3] that generates a series of random planes of data to view via a 2D display. Bifocal displays utilizing volumes within a cube can present noticeable optical artifacts causing some users to have imprecise depth perception when using immersive displays.

Real world volumetric data is difficult to share amongst other researchers due to both its large file size and privacy concerns [5]. Medical data privacy is very important, and there may be restrictions on what datasets can be distributed through research [7, 23]. The size of the compressed data can easily be larger than one gigabyte for a lower resolution volume and even greater when distributing raw medical data. This becomes an issue for data sharing long term since storing large data sets will likely come at a cost to either the publisher on the researchers [23]. Leading to many issues with long-term data availability.

Synthetically created data has been used before in Englund et al.'s [3, 4] and Kersten et al.'s [15] controlled experiments. Kersten et al. [15] et al. utilized a fog-like volume within a cylindrical object rendered using DVR and displayed on either a monoscopic or stereoscopic monitor while the volume rotated slightly.Englund et al.'s [3, 4] experiments generated data designed especially for the purposes of testing their study and rendered as static images displayed on a 2D display for a range of two alternative forced-choice tasks.

Our system is designed to produce reproducible and diverse results with constant but random volumes that are able suitable for a wide range of studies in mixed reality and are easy to extend and implement new visualizations. The input it requires is received via either a small set of input parameters that can be preset or can be manufactured using the Immersive UI (seen in Figure 1 and described in Section 3.2). These input parameters include all settings, from how many objects should exist, how big they should be, and how many child objects they can house. Every element will be produced to match the desired outcome of the input parameters by randomly assigning elements to avoid any similarities between the conditions. The system itself is extremely modular and can generate many different scenarios. The volumes can be output as isosurfaces or DVR allowing them to be utilized faster. The default JSON files this system generates for the volumes are generally under 100 kilobytes in size by only listing the generic properties of the visualization (shape details and Perlin noise variables) rather than recording each voxel's details. It can also be output in different methods if desired, such as a raw volume file or an isosurface.

This paper contributes the following to the field.

- A modular system that is able to create a range of volumes designed for controlled user studies in AR.
- A user interface designed for head-worn Mixed Reality(MR) devices user interactions with this system that allows participants to view the generated volumes.

The next section discusses related work on synthetic data generation (Section 2). The subsequent section introduces the system (Section 3) and its mixed reality interface(Section 3.2), which both provide an overview of important technical details and algorithms underlying the system. This includes the algorithms required this system utilizes, the required inputs and outputs, and what our immersive experience provides.

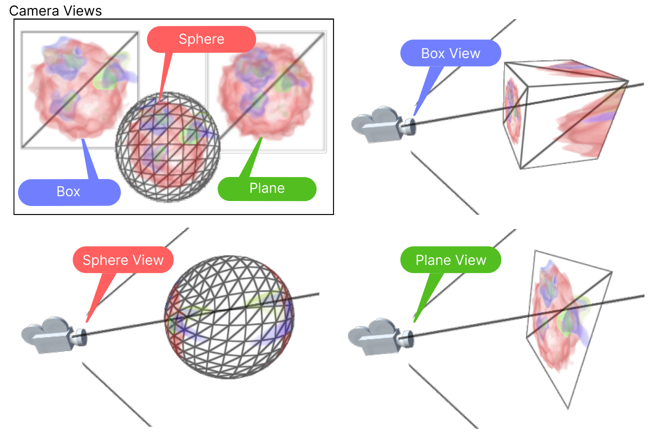

Figure 2: A third person view of Direct Volume Rendering using three different polygonal meshes. In the Box Labeled "Camera Views" are the outputs from the cameras that you can see looking at each of the different mesh. The straight lines coming from the camera represent the camera frustum.

2 Related Works

Several systems exist to generate of synthetic data. Meyer et al. [22] as large amounts of data of any type are hard to come by, and systems need to be tested using flexible mechanics and a wide range of parameters. Patki et al.'s [26] synthetic data vault also allows users to create randomly generated synthetic data based on a smaller set of the original data. These graphs and fake data can be used to train AI models or for various users present for various outcomes. However, they normally need a baseline dataset to take inspiration from and are not simple to modify for different studies [22, 26].

Bossik et al. [1] and Haghihi et al. [12] created systems for creating artificial 2D network graphs. They tend to be used for graph traversal algorithms detection and process benchmarking of graph visualizations and are not designed for studying human interaction but are able to be viewed in a number of different formats.

To this point, several papers have tried using Generative Adversarial Nets (GANs) [9] and stable diffusion [2] to create real examples of medical data [24, 28, 29, 32]. These datasets provide use cases for training staff without the need to provide scans of real people, which can benefit tasks such as publishing and sharing medical data with a wider audience [32], training radiologists [24] and research [24, 29]. These models utilize a large collection of already existing data and aim to create a new set of existing data from this which tends to be indistinguishable from the real data [29, 32] However, both GAN's and Stable Diffusion models are difficult to control the exact parameters of the models, making them problematic to use for controlled experiments but allowing for better uncontrolled experiments.

3 System Overview

The volumes produced by this system appear as seen in Figure 3. In order to make these volumes as much like MRI and CT scans as possible, we utilized Perlin noise to deform a sphere object created via a spherical Sign Distance Field (A noisy sphere) set up in a hierarchical fashion [27]. Since they are Signed Distance Fields (SDFs), we also are able to determine the angle of normals in the volume quite accurately, allowing for X-ray vision studies (as seen in Figure 6) as well as visualizations for many other experiments. Four different types of noisy spheres can be made for the base version: outer, composite, leaf and some multipurpose spheres can also be added. To improve optical focus, these objects are housed within a spherical mesh. Figure 2 shows this allows for better bifocal viewing by allowing the mesh to align with the volume.

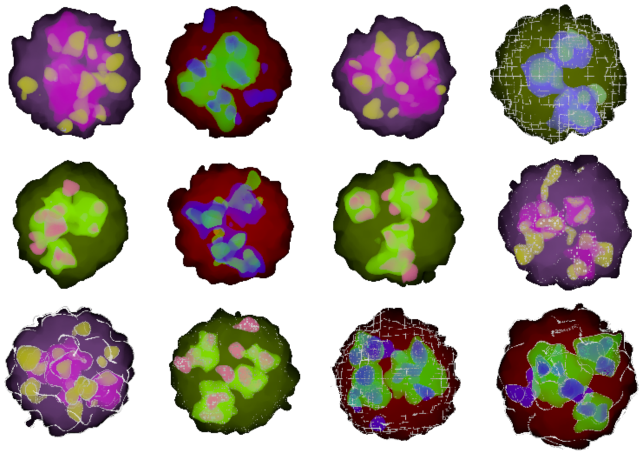

Figure 3: A small example of the types of volumes that this system is capable of producing. There are noisy spheres on the bottom and right-hand sides using various illustrative effects (Outlines, Stippling, and Hatching).

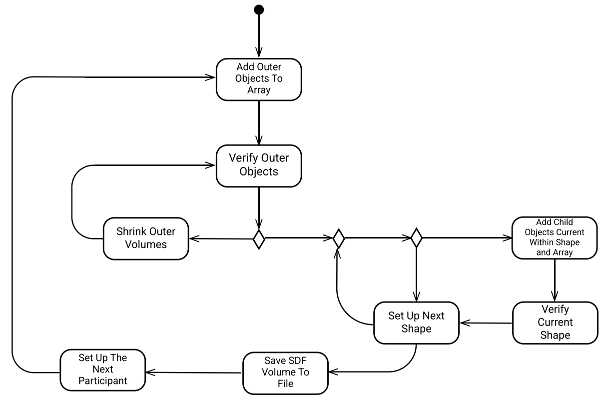

Figure 4: A activity diagram showing the transition between the various states of the Random Volume Generation System

Improving the distortion from the bi-optical, allowing for better depth perception [15].

This system is primarily aimed at researchers conducting human-computer interaction research concerned with investigating the traits and impacts of different displays and rendering styles [30, 31]. This system's volumes will need to have generic qualities that other forms of volumes would typically have. We chose to base these volumes on a similar design to MRI and CT scans, but they also share qualities with electron microscopy visualizations [25], and with minor modifications, they can be conformed to molecular systems [10], meteorological data [14, 20], as well as ground penetrating radar (GPR) data [34]. All these imaging techniques communicate information via geometrical methods, require a high degree of precision to generate, and are subjective to noise.

We also considered the previous studies that have generated artifical volumes. Englund et al.' s [3, 4] static images but wouldn't be translated well to a stereoscopic display. Kersten et al.'s [15] Perlin noise fog showed promise when using stereoscopic display. Perlin noise was adopted to generate synthetic volumes to mimic the noisy characteristics of volume-rendered objects in applications such as MRI and CT scans. This approach is advantageous since it can be rendered effectively on a variety of surfaces in real time. Volume data tends to be inherently noisy and messy, making it more difficult to interpret, so Perlin noise-based objects make logical sense [7, 34]. The ability to create a solid surface that could represent to different materials or densities would be beneficial when trying to create iso-surfaces, and the ability to have identifiable pieces components is common in MRI and CT scans.

3.1 Random Volume Generation System

The Random Volume Generation system [2] is built as a modular system built in the Unity game engine [3]. This system is designed to produce an array of volumes that can conform to a set of conditions. Figure 3 shows the volumes generated by the system are represented as a Signed Distance Field (SDF), whose shape is a "noisy sphere". The shape of the object is based on a sphere with random distortions. As such, each shape is irregular in nature, and a pseudo-random generation process determines the specific shape and appearance.

In each iteration of the generation process (shown in Figure 4), a top-level volume is created and added to the scene. Children within the volume are created recursively. The size and appearance of each volume are determined based on the probability distributions given as input to the generation process, whereby the size of the child volumes is constrained by their parent volume (if any). The number of children in the volume is drawn from a probability distribution for each volume. In addition, global constraints govern the overall number of objects of each type at each level to ensure that the synthesized scene satisfies the desired properties.

The dimensions and placement of objects are determined at random such that each volume is wholly contained in its enclosing parent volume (if any), and no objects are allowed to touch each other at any point. These placement constraints are verified by a voxel-based algorithm that examines possible intersections between volumes. The system provides a naive algorithm and a more efficient octree-based implementation. In case an object is found to violate the constraints, the object is moved to repair the situation.

3.1.1 Verification

The verification system looks at each voxel and determines if they are sitting wholly inside each other and not slipping outside their parent object or touching any other object that isn't a parent of theirs. These rules can be easily exchanging any class that inherits the Irules interface. The rules are checked via a C# based system that emulates the same properties of Unity's high-level shader language (HLSL). It is designed to be extendable to allow for any different type of SDF and even potentially a different type of shader altogether.

The verification methods for the root objects look at making sure the volume fits neatly within the mesh it is rendering within. This mesh can either be a cube mesh or a spherical mesh. The cube mesh checks all of the voxels on the outside of the mesh. While the Sphere mesh utilized A Fibonacci sphere algorithm to determine all collect a dense and evenly distributed array of points over the mesh [8]. In both conditions, if any point from any of these checks were found to have any SDFs within a range of them, it would then shrink the volume and then perform the check again.

The verification used for the leaf and composite objects can use two different verification methods. Either a brute force or progressive oct-tree searching method. Both of these can be combined as well. This can either happen linearly or in parallel for each voxel. The other method utilizes a partial linear oct-tree starting. From one predefined depth in the oct-tree from one depth to another deeper one. This will perform the check mentioned earlier in this section using a trimmed depth-first search. This search will determine if a group of voxels conform to SDF coordinates.

[2] https://github.com/tomishninja/RandomlyGeneratingVolumes
[3] https://unity.com/releases/editor/archive

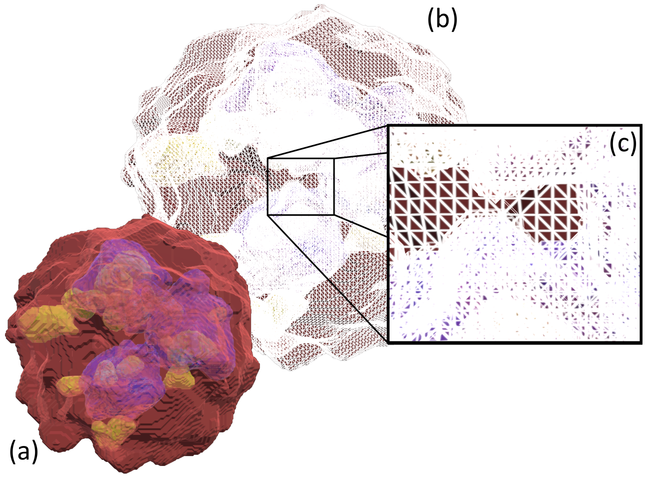

Figure 5: A breakdown of the structures of the 3D meshes generated by this system. a) presents the volume using transparent colors to show the different levels that meshes can be segmented, b) is the same volume but covered in a wireframe, c) is a close-up shot of the center b

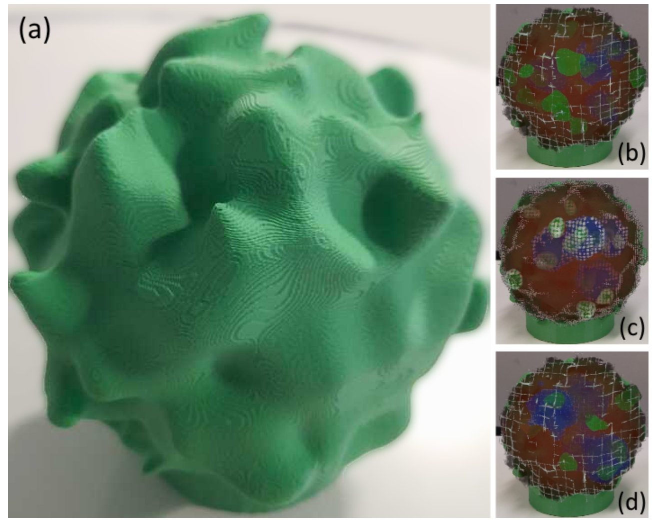

Figure 6: (a) A 3D printed version of the model made from the mesh output of this system. (b, c, and d) shows this model from the view of a Microsoft Hololens2 to create an X-ray vision effect using illustrative rendering.

3.1.2 System Outputs

The output from these files is able to provide mesh file outputs (.obj and .stl) or as a JSON to show DVR content. The meshes are created via marching running marching cubes over the volume and can be created at any resolution required. The volume can be saved as a single mesh or a collection of smaller meshes (as shown in Figure 5) spaced appropriately apart. Allowing game engines and most displays to render them natively. It can also create a 3D printable model by using a 3D printer (As seen in Figure 6).

The JSON file is designed to be read as input for a user study. It contains instructions on what condition the visualization was built for, the noise key, and any answers required for each volume, like volumetric information and the number of volumes that are contained under a certain circumstance. This modular system can be redesigned using AR or VR devices for many DVR studies.

3.1.3 System Details

All classes used to build this system utilize a high level of inheritance and utilize interfaces and abstract classes to allow for a high level of modification. An independent set of classes has been designed to set up all of the objects and relationships. While the main structure uses a set of 3 stateful systems. One to determine what type of action is required to take, another to add more shapes to the volume, and one to verify the volume.

The logic systems are functors as they tend to be designed to run more than is possible in a single frame, Impairing the users' ability to interact with the UI. Their behavior is modular, with the system's logic separated from their unique tasks. Allowing for a system that is capable of doing much more than its base functionality. Two simple examples of the methods are recursively placing an SDF within its parents so it takes as much space as possible. The other one can produce several meshes for 3D printing(seen in **??**) while also providing their corresponding noise key, allowing for comparisons with real world objects.

3.2 Immersive UI Design

This system's immersive design component is where the volumes are randomly generating various volumes while the user can change a set of parameters with the goal of creating an input file for the final system. By immersing researchers in the same environment as their users, they can experience the data set the same way their participants would while choosing parameters for various instances of the project.

The basic UI elements have been generated from the mixed reality tool kit [4] and have been laid out to keep the users focused on the volume. These elements include changing two radial menus between various conditions and updating input parameters (shown up the top and two the right of Figure 1). Most of the input parameters users can change relate to sliders (seen on the right side of Figure 1) that will allow them to choose between various sizes and amounts of objects that want to exist at a time.

One new element to this UI requires a volumetric color picker (shown down the bottom of Figure 1). The various colors for the different elements of the volume can be chosen from a 3D color picker in a similar style to Kim et al.'s [16] color pickers designed for color blending where direct volume rendering is used to identify an object color.

In the future, we intend to extend the system to enable two people in separate locations and on separate devices to interact with this system simultaneously using a technique similar to McDade et al.'s CADET [21]. As well as allow for more interactions with the interface, such as changing the outer mesh adjusting the output options, and even being able to utilize different types types of shapes.

4 Conclusion

We provide a modular system for creating and evaluating volumes designed for various AR and VR user studies which exports a volume allowing for a simple sdf treatment. The volumes that are created are generic and easy to tailor for different experimental settings. Our system includes an MR user interface in which researchers can experience the same conditions that users will experience in a study to validate the quality of the generated 3D objects. This system has the potential to simplify the generation of diverse volumetric content for AR and VR studies.

Acknowledgments

Research supported by an Australian Government Research Training Program Scholarship.

[4]https://github.com/microsoft/MixedRealityToolkit-Unity

References

[1] J. Bossek. grapherator: A Modular Multi-Step Graph Generator. *The Journal of Open Source Software*, 3(22):528, 2018. doi: 10.21105/joss.00528

[2] P. Dhariwal and A. Nichol. Diffusion Models Beat GANs on Image Synthesis. *Advances in Neural Information Processing Systems*, 11:8780–8794, 2021.

[3] R. Englund and T. Ropinski. Evaluating the perception of semi-transparent structures in direct volume rendering techniques. *SA 2016 - SIGGRAPH ASIA 2016 Symposium on Visualization*, 2016. doi: 10.1145/3002151.3002164

[4] R. Englund and T. Ropinski. Quantitative and Qualitative Analysis of the Perception of Semi-Transparent Structures in Direct Volume Rendering. *Computer Graphics Forum*, 37(6):174–187, 2018. doi: 10.1111/cgf.13320

[5] B. Fabian, T. Ermakova, and P. Junghanns. Collaborative and secure sharing of healthcare data in multi-clouds. *Information Systems*, 48(October 2022):132–150, 2015. doi: 10.1016/j.is.2014.05.004

[6] R. Fischer, K. C. Chang, R. Weller, and G. Zachmann. Volumetric Medical Data Visualization for Collaborative VR Environments. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12499 LNCS(March 2021):178–191, 2020. doi: 10.1007/978-3-030-62655-6_11

[7] C. Gillmann, N. N. Smit, E. Groller, B. Preim, A. Vilanova, and T. Wischgoll. Ten Open Challenges in Medical Visualization. *IEEE Computer Graphics and Applications*, 41(5):7–15, 2021. doi: 10.1109/MCG.2021.3094858

[8] Á. González. Measurement of Areas on a Sphere Using Fibonacci and Latitude-Longitude Lattices. *Mathematical Geosciences*, 42(1):49–64, 2010. doi: 10.1007/s11004-009-9257-x

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. doi: 10.1145/3422622

[10] D. S. Goodsell, I. S. Mian, and A. J. Olson. Rendering volumetric data in molecular systems. *Journal of Molecular Graphics*, 7(1):41–47, 1989. doi: 10.1016/0263-7855(89)80055-4

[11] A. V. Grosset, M. Schott, G. P. Bonneau, and C. D. Hansen. Evaluation of depth of field for depth perception in DVR. *IEEE Pacific Visualization Symposium*, pp. 81–88, 2013. doi: 10.1109/PacificVis.2013.6596131

[12] S. Haghighi. Pyrgg: Python Random Graph Generator. *The Journal of Open Source Software*, 2(17):331, 2017. doi: 10.21105/joss.00331

[13] F. Heinrich, L. Schwenderling, M. Streuber, K. Bornemann, K. Lawonn, and C. Hansen. Effects of Surface Visualizations on Depth Perception in Projective Augmented Reality. *Proceedings of the 2021 IEEE International Conference on Human-Machine Systems, ICHMS 2021*, pp. 0–5, 2021. doi: 10.1109/ICHMS53169.2021.9582452

[14] W. Hibbard L. 4-D Display of Meterological Data. *Interactive 3D Graphics*, pp. 23–36, 1986.

[15] M. A. Kersten, A. J. Stewart, N. Troje, and R. Ellis. Enhancing depth perception in translucent volumes. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1117–1123, 2006. doi: 10.1109/TVCG.2006.139

[16] J. Kim, J. I. Hwang, and J. Lee. VR Color Picker: Three-Dimensional Color Selection Interfaces. *IEEE Access*, 10:65809–65824, 2022. doi: 10.1109/ACCESS.2022.3184330

[17] B. Laha and D. A. Bowman. Poster: Volume Cracker: A bimanual 3D interaction technique for analysis of raw volumetric data. *IEEE Symposium on 3D User Interface 2013, 3DUI 2013 - Proceedings*, C:147–148, 2013. doi: 10.1109/3DUI.2013.6550221

[18] B. Laha, D. A. Bowman, D. H. Laidlaw, and J. J. Socha. A classification of user tasks in visual analysis of volume data. *2015 IEEE Scientific Visualization Conference, SciVis 2015 - Proceedings*, D:1–8, 2016. doi: 10.1109/SciVis.2015.7429485

[19] B. Laha, K. Sensharma, J. D. Schiffbauer, and D. A. Bowman. Effects of immersion on visual analysis of volume data. *IEEE Transactions on Visualization and Computer Graphics*, 18(4):597–606, 2012. doi: 10.1109/TVCG.2012.42

[20] D. Lin, C. Wyman, and C. Yuksel. Fast volume rendering with spatiotemporal reservoir resampling. *ACM Transactions on Graphics*, 40(6):1–18, 2021. doi: 10.1145/3478513.3480499

[21] J. McDade, A. Drogemuller, A. Jing, N. Ireland, J. Walsh, B. Thomas, W. Mayer, and A. Cunningham. CADET: A Collaborative Agile Data Exploration Tool for Mixed Reality. *Proceedings - 2022 IEEE International Symposium on Mixed and Augmented Reality Adjunct, ISMAR-Adjunct 2022*, pp. 899–900, 2022. doi: 10.1109/ISMAR-Adjunct57072.2022.00195

[22] D. Meyer, T. Nagler, and R. J. Hogan. Copula-based synthetic data augmentation for machine-learning emulators. *Geoscientific Model Development*, 14(8):5205–5215, 2021. doi: 10.5194/gmd-14-5205-2021

[23] Nature. Data sharing is the future. *Nature Methods*, 20(4):471, 2023. doi: 10.1038/s41592-023-01865-4

[24] L. X. Nguyen, P. S. Aung, H. Q. Le, S. B. Park, and C. S. Hong. A New Chapter for Medical Image Generation: The Stable Diffusion Method. *International Conference on Information Networking*, 2023-January:483–486, 2023. doi: 10.1109/ICOIN56518.2023.10049010

[25] N. Nguyen, F. Liang, D. Engel, C. Bohak, P. Wonka, T. Ropinski, and I. Viola. Differentiable Electron Microscopy Simulation: Methods and Applications for Visualization. (1):1–22, 2022.

[26] N. Patki, R. Wedge, and K. Veeramachaneni. The synthetic data vault. *Proceedings - 3rd IEEE International Conference on Data Science and Advanced Analytics, DSAA 2016*, pp. 399–410, 2016. doi: 10.1109/DSAA.2016.49

[27] K. Perlin and E. M. Hoffert. Hypertexture. *ACM SIGGRAPH Computer Graphics*, 23(3):253–262, 1989.

[28] W. H. L. Pinaya, P.-d. Tudosiu, J. Dafflon, P. F. Da, V. Fernandez, P. Nachev, S. Ourselin, and J. M. Cardoso. Brain Imaging Generation with Latent Diffusion Models. *arXiv*, pp. 1–10, 2022.

[29] Z. Ren, S. X. Yu, and D. Whitney. Controllable medical image generation via generative adversarial networks. *IS and T International Symposium on Electronic Imaging Science and Technology*, 2021(11), 2021. doi: 10.2352/ISSN.2470-1173.2021.11.HVEI-112

[30] P. Rheingans and D. Ebert. Volume illustration: Nonphotorealistic rendering of volume models. *IEEE Transactions on Visualization and Computer Graphics*, 7(3):253–264, 2001. doi: 10.1109/2945.942693

[31] N. A. Svakhine, D. S. Ebert, and W. M. Andrews. Illustration-inspired depth enhanced volumetric medical visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15(1):77–86, 2009. doi: 10.1109/TVCG.2008.56

[32] R. Togo, T. Ogawa, and M. Haseyama. Synthetic gastritis image generation via loss function-based conditional pggan. *IEEE Access*, 7:87448–87457, 2019. doi: 10.1109/ACCESS.2019.2925863

[33] V. Yoghourdjian, D. Archambault, S. Diehl, T. Dwyer, K. Klein, H. C. Purchase, and H. Y. Wu. Exploring the limits of complexity: A survey of empirical studies on graph visualisation. *Visual Informatics*, 2(4):264–282, 2018. doi: 10.1016/j.visinf.2018.12.006

[34] B. Zehner. On the visualization of 3D geological models and their uncertainty. *Zeitschrift der Deutschen Gesellschaft fur Geowissenschaften*, 172(1):83–98, 2021. doi: 10.1127/zdgg/2020/0251