Shader Development and User Interface Design for 3D Scientific Visualization in VR Headset

By

Annaliese Watson

Submitted to Integrated Science and Technology

Kean University in partial fulfillment
of the requirements for the degree of

Master's of Science in Computational Science and Engineering

May 2025

Computational Science and Engineering Graduate Program

Shader Development and User Interface Design for 3D Scientific Visualization in VR Headset

A Thesis Presented

By

Annaliese Watson

Approved as to style and content by:

____

Dr. David Joiner, Acting Dean (Primary Advisor)

____

Dr. Edward Farnum, Committee Member

____

Dr. Ensela Mema, Committee Member

Acknowledgements:

Huge thanks to the Center for Biological Imaging and the Niepielko Lab at Kean University, your support and insight brought this project to life. Grateful to the NIH (R15HD102960) for funding, and to the Dorothy Hennings College of Science and Mathematics.

Endless appreciation to Judy April, Nan Peligro and Dr. David Joiner, your guidance meant everything. Thank you to my committee, Dr. Edward Farnum and Dr. Ensela Mema, for your mentorship and feedback. Shoutout to all whose encouragement kept me going.

Abstract

This research introduces an innovative user interface tailored for the visualization and manipulation of three-dimensional (3D) hyperstack images, obtained through confocal microscopy. Focused on biological specimens such as pollen grains and fruit fly stem cells, the interface leverages virtual reality (VR) technology to overcome the limitations of traditional 2D and 2.5D visualization techniques. It offers full spatial immersion by rendering volumetric data with enhanced transparency, rotational control, zoom functionality, and interactive slicing along the x, y, and z planes. Inspired by Brett Jackson's Idea Engine, the system incorporates natural user interface (NUI) principles to support intuitive interactions, helping scientists engage more naturally with complex datasets. Developed in Unity Game Engine using customized volumetric shaders and the XR Interaction Toolkit, users can grab, slice, rotate, and zoom into data structures through hand and controller-based input. Performance testing on Oculus Quest 2, 3, and Pro headsets demonstrated stable operation at 50–72 frames per second, maintaining high-resolution integrity (1024 × 1024 × 80). Beyond supporting object detection validation and 3D pollen grain database creation, this work lays the foundation for immersive scientific visualization across disciplines providing a more natural, accessible, and efficient method for engaging with complex biological datasets across 2D, 2.5D or 3D spaces.

Introduction & Background

1.1 2D, 2.5D and 3D Visualization in Virtual Reality

Scientists rely heavily on visualization tools to interpret complex datasets. These datasets can be represented in two-dimensional (2D), two-and-a-half-dimensional (2.5D), or three-dimensional (3D) formats, each of which significantly influences how users interact with the data and derive insights. The process of developing visualization tools and interacting with these formats varies considerably.

Traditional scientific visualization methods often rely on 2D or 2.5D environments. In a 2D format, visual elements are arranged on a flat, single-layered surface, providing no perception of depth [1]. Common features include slicing through datasets or scrolling through Z-stacks—an approach frequently used in microscopy and medical imaging. A 2.5D setup introduces limited depth by vertically stacking plane layers, creating the illusion of three-dimensionality without enabling full spatial interaction [1]. These environments typically use a false perspective on flat 2D displays (e.g., computer screens), a technique popularized through video game engines. When adapted for scientific visualization, 2.5D formats offer new perspectives on complex datasets, allowing users to perceive depth and structure in ways that are not possible with static 2D images.

However, incorporating 2D and 2.5D environments into scientific analysis introduces challenges, particularly in preserving spatial fidelity, formation and data integrity [2]. These challenges stem from how virtual data is mapped to visual representations and how spatial relationships between physical and digital elements are maintained during rendering [3]. Definitions of these formats extend beyond the visual output. They consider how virtual data maps onto a representation space, the spatial relationships involved, and how interactions between the physical and digital worlds are structured. Mathematical precision is necessary to ensure that visualization methods do not distort or compromise the scientific accuracy of original datasets.

Virtual Reality (VR), by contrast, enables the creation of fully immersive 3D environments, overcoming many of the limitations of traditional formats. Unlike standard 3D visualizations displayed on flat monitors, VR allows users to navigate spatially distributed elements in real-time through headsets like the Oculus Quest and Apple Vision Pro. A true 3D environment positions data throughout a navigable three-dimensional space, such as over a hemisphere allowing for immersive and intuitive exploration [1].

VR presents an opportunity to enhance scientific data interpretation by supporting natural interactions. Through gestures and controller-based inputs, users can manipulate volumetric structures with greater spatial awareness. This level of interaction, absent in conventional 2D or 2.5D settings, is particularly valuable for researchers working with highly detailed biological data.

As scientific inquiry continues to address increasingly complex systems, traditional 2D representations are no longer sufficient. Three-dimensional imaging enables researchers to slice, rotate, and analyze data from multiple angles. This technique has become a critical asset across disciplines such as biology, medicine, and environmental science. While each field may pursue different visualization goals, the aim remains consistent: to make complex data more accessible and interpretable.

Despite significant advancements in visualization technologies, many existing tools remain confined to 2D or 2.5D environments. Even where VR is introduced, its applications are often limited to simple educational experiences rather than fully developed scientific platforms. This research seeks to bridge that gap by developing interactive and intuitive VR user interface elements, enhanced with custom volumetric shaders, specifically designed to support the visualization and analysis of confocal microscopy datasets. Throughout this paper, we define 3D as immersive views experienced through headsets such as the Oculus Quest or within CAVE environments; 2.5D as first-person perspectives rendered on flat screens, such as those found in PC video games; and 2D as inherently flat visualizations, including side-scrollers, wide third-person views, and standard data plots.

1.2 Scientific Visualization and Confocal Microscopy

Scientific visualization refers to the study and design of techniques for visually representing complex datasets in science and engineering disciplines [4]. It plays a vital role in helping scientists identify patterns, detect anomalies, and interpret large-scale data in ways that are both intuitive and informative. Well-known examples include CT scans, X-rays, and MRIs, tools that have transformed medical diagnostics by converting raw data into meaningful visual forms. To support deeper interpretation, modern visualization increasingly incorporates technology and mathematical computation. Drawing on insights from STEM fields and immersive technologies, this research explores new approaches to enhance scientific visualization and data manipulation.

In today's data-driven landscape, the retrieval and analysis of high-resolution scientific data plays a pivotal role in shaping decisions and driving discovery. Microscopic imaging reveals structural details invisible to the naked eye. One key source of such data is confocal microscopy, an advanced technique capable of capturing detailed three-dimensional representations of cellular structures. Widely used in medical and biological imaging, confocal microscopy reveals microstructures not visible through traditional light microscopy [5].

Traditional microscopy, which relies on direct optical observation, often falls short when analyzing complex biological structures. Confocal microscopy addresses this limitation by blocking out-of-focus light and enabling the capture of high-resolution, optically sectioned images. These thin slices can be digitally reconstructed into sharp, three-dimensional views of cells and tissues, allowing for detailed structural analysis [6].

Confocal microscopes are especially valuable in quantitative biology due to their spatial precision and ability to minimize background noise. These advantages make them ideal for studying intricate cellular processes and spatial relationships within biological samples. Recent advances such as multi-view imaging, super-resolution microscopy, and live-cell imaging have further expanded their capabilities. The resulting datasets, often referred to as hyperstacks, are rich, layered image volumes well-suited for advanced visualization techniques, including integration into virtual reality (VR) platforms.

Due to their complexity, hyperstack datasets require sophisticated visualization tools. By integrating scientific visualization methods with immersive VR environments, researchers can interact with these multidimensional datasets in more intuitive, spatially aware ways. This research builds on that potential by leveraging confocal microscopy as the foundational data source for a VR-based visualization system enabling scientists to explore biological structures through natural, embodied interaction.

1.3 Role of VR and Natural User Interface (NUI)

Environments that use 2.5D visualization have a few upfront interaction challenges. What interaction features can be included to decrease the learning curve? The less time spent for scientists to figure out how to use the platform, the more time can be spent going into the research of microscopic data. Despite advancements in technology, much of scientific data visualization remains confined to two-dimensional (2D) or two-and-a-half-dimensional (2.5D) displays. These approaches limit a user's ability to perceive depth and interact naturally with complex datasets, relying instead on indirect tools such as a mouse and keyboard. Effective analysis of microscopic structures often demands exploration techniques like rotating objects, zooming into details, and revealing internal cross-sections tasks that traditional interfaces struggle to support.

This rapid evolution highlights a key principle: as technology advances, the way users interact with devices becomes increasingly aligned with familiar human behaviors. We moved from pressing physical buttons to tapping directly on screens, and now to engaging with digital environments through gestures and spatial movement in augmented and virtual reality. Each transition removed layers of abstraction between the user and the system, making interaction feel more direct. As we progress toward immersive environments, the guiding question for system design becomes: what feels most natural to the user?

This shift sets the foundation for the concept of Natural User Interfaces (NUI), which aim to replicate the ease of interacting with real-world objects. A focused section of the industry is working to integrate VR into more areas with minimal onboarding time. NUI encourages developers and researchers to consider how users instinctively engage with physical, 2D, or 3D objects, making advanced digital tools more accessible to novices. Although technology continues to evolve and reshape Human-Computer Interaction (HCI), core interaction behaviors often remain consistent.

With the mix of NUI and VR challenges, we are able to use computational skills and technology to help scientists manipulate digital data. Since the immersion of VR in professional industries, many fields such as medicine, architecture, education, tourism, and sports have successfully used VR to benefit their individual goals. These studies have had extensive research to also see factors that can impact the use of VR technology in these fields [7]. Additional studies have been conducted showing that users prefer VR video experiences rather than 2D pictures in sectors such as tourism, digital marketing, museums, and journalism, as it increases emotional response and experience enjoyment [8].

With this implementation, certain optimizations are necessary to improve user experience. User Interface (UI) and User Experience (UX) are key to ensuring the effectiveness and usability of scientific platforms. A thoughtfully designed UI enables users to navigate and interact with data more effectively. Research highlights the importance of critical VR design elements, such as resource allocation, object accuracy, realism, and environmental responsiveness, all of which contribute to deeper engagement. As VR technology continues to grow, embedding interaction models and ensuring cross-platform compatibility will be essential to advancing immersive scientific visualization.

Beyond simply displaying data, another consideration is how users interact with these virtual environments. Wireframing and prototyping play an essential role in designing effective visualization platforms, as they help optimize a researcher's ability to explore, analyze, and better understand complex data. Today, many scientific visualization applications are developed using powerful engines like Unity Game Engine or Unreal Engine, which provide robust tools for building immersive environments while preserving data fidelity.

One key development goal is to create interfaces that are immediately understandable without extensive training. Natural User Interfaces (NUI) support this by allowing users to interact through familiar body movements, such as gestures [9]. As we move from basic button presses to more dynamic systems like hand gestures and voice commands in extended reality (XR), developers must shift their design thinking to prioritize ease of use and behavior-based controls [10].

As XR environments continue to mature, NUIs principles is enabling more seamless and responsive interactions. Eye movements, gestures, and voice inputs help users engage with virtual environments in ways that mimic real-world behavior [10]. For instance, natural actions like grabbing or throwing can now be replicated in VR, supporting more fluid experiences. Brett Jackson's Idea Engine exemplifies these principles in practice. His platform allows users to create and share virtual reality (VR) and mixed/augmented reality (MR/AR) experiences [11]. Jackson emphasizes thoughtful design, considering both ideal and challenging user scenarios to improve effectiveness. Key features of Idea Engine include UI panels that can be manipulated from any angle, automatic panel flipping for readability, and scrollable content controlled by simple gestures like grabbing and dragging. Rather than mirroring traditional 2D interaction methods, Jackson reimagines how we interact in spatial digital contexts. He also highlights challenges such as accessibility, user fatigue, and usability concerns, emphasizing that a successful NUI must strike a balance between innovation and user comfort. Building on these ideas, further research and intentional design are essential to making XR environments more immersive, accessible, and user centered.

1.4 Research Objective and Contribution

This study develops a virtual reality (VR) interface for the visualization and analysis of confocal microscopy datasets, leveraging layered image data provided by the Center for Biological Imaging and the Niepielko Lab at Kean University. By building on the Unity Game Engine, the project focuses on optimizing interaction design in a 3D headset environment rather than a 2.5D environment. The aim is to provide researchers with a more natural, immersive, and efficient method for engaging with complex biological data.

The user interface (UI) was designed to mirror familiar interaction styles such as grabbing, slicing, rotating, and zooming allowing users to intuitively manipulate 3D hyperstack images as if holding them in their hands. Core UI elements include contrast adjustment, transparency control, and cut-plane navigation. This approach aligns with Natural User Interface (NUI) principles, lowering the learning curve and shifting focus from system navigation to scientific exploration.

Custom volumetric shaders were developed to support transparency and cross-sectional slicing along the x, y, and z planes, preserving both structural clarity and data fidelity. The shaders were enhanced to apply transparency selectively, defined by a base point and normal vector. An interactive in-headset menu offers sliders, checkboxes, and toggles for controlling these visual parameters in real time. Interface performance was validated across Oculus Quest 2, 3, and Pro headsets, achieving stable frame rates between 50–72 frames per second (FPS). XR Interaction Toolkit integration enabled real-time responsiveness, control mapping, and cross-device compatibility.

This project addresses common challenges in volumetric scientific visualization, such as performance lag, rendering artifacts, and unintuitive input methods. Existing platforms often rely on traditional keyboard and mouse inputs, which do not translate well to immersive data analysis. Additionally, many interfaces are not designed with scientific workflows in mind, leading to inefficiencies and steep learning curves.

By introducing a system specifically tailored for confocal microscopy datasets, this work fills a critical gap: the need for immersive tools that maintain scientific accuracy, support embodied interaction, and operate smoothly in real time. Traditional 2D slice scrolling or 2.5D pseudo-3D views are often insufficient for fully conveying spatial complexity. This VR interface bridges that gap transforming raw microscopy data into a more accessible, spatially rich, and precise environment for analysis.

Ultimately, this project contributes to the field of scientific visualization by demonstrating how immersive technologies, grounded in NUI design can improve both usability and insight generation when working with complex biological structures.

Design Goals

The main objective of this project is to develop a VR-based software system that enables scientists to intuitively analyze pollen cells using interactive, immersive tools. The design process is guided by key questions:

- Does it solve the core visualization and interaction problem?
- Is it easy for new users to understand without long tutorials?
- What features will make it functional and user-friendly?

With the rise of Natural User Interfaces (NUI), interaction systems are moving toward designs that mimic real-world gestures and behaviors. This research is geared to incorporate these advancements as seamlessly as possible. By doing this it will reduce the cognitive load on users and enable scientists to focus on their data rather than learning complex tools. To explore a pollen grain effectively, users should be able to interact with it as if it were a tangible object, zooming in, rotating, and slicing through it. The VR environment supports this kind of immersive, natural interaction. The design goals of this research project fall under four major categories, functional goals, usability goals, performance goals and hardware or software considerations.

2.1 Functional Goals

Our functional goals are centered around how users will be able to analyze the data. Specifically, the design focuses on the features users need to easily manipulate to achieve a correct analysis of the pollen grain. The microscopic images are stacked on top of each other and then uploaded into Unity Game Engine as layered data. The system must allow for the following key functions:

- **Selection and Recognition of Specific Layers:**
  Users must be able to select a specific image within the dataset. This selection may be facilitated by a pointer that tracks its location to ensure the correct image is identified.
- **Cut Plane Vertex Implementation:**
  Users should be able to apply a cut plane vertex to isolate and view specific parts of the pollen grain. This allows users to slice through the 3D volume and focus only on the areas they want to analyze.
- **Transparency Control Using Volumetric Rendering:**
  An added feature to the cut plane vertex is transparency control. Users can adjust transparency levels of the sliced sections, improving visibility of internal structures without losing overall context.
- **Dynamic Cut Plane Manipulation:**
  Users must not be restricted to slicing along only the x or y axis. Instead, they should be able to manipulate the cut plane freely across any axis (x, y, or z) and rotate it at any desired angle to explore cross-sections dynamically.
- **Centralized Menu System:**
  A centralized menu will serve as the environment to host all interaction options. Inspired by Brett Jackson's Idea Engine concept, the menu should:
  - Be accessible by pressing a button on the VR controller
  - Use eye-tracking to appear directly in front of the user
  - Remain stationary in the VR environment, allowing users to move around it
  - Be readable and visible from both the front and back sides to maintain accessibility from any angle
- **Zoom Functionality:**
  A circular zoom feature should allow users to magnify a targeted area. This zoom will help users focus on specific multi-layered regions of the pollen grain, providing better clarity and precision for detailed analysis.

2.2 Usability Goals

When considering usability goals, it is important to identify natural gestures that users commonly employ to manipulate objects in an immersive environment, with the aim of reducing the cognitive learning curve. In everyday interactions, hand gestures are used naturally while communicating, while in immersive gaming environments, users typically rely on mouse input, WASD keys, and joystick buttons for navigation and control.

Some of the primary usability goals identified for this project include enabling intuitive actions such as grabbing, rotating, and slicing to manipulate the pollen grain models. Movement within the virtual space will be supported through teleportation, allowing users to travel from one point to another easily. In addition to teleportation, the system will offer alternative locomotion options, including movement via WASD keys combined with mouse control, and analog walking using the upward push of a joystick lever.

To further enhance interaction precision, gaze tracking and laser pointers will be incorporated, allowing users to easily target and select objects or regions of interest. Visual cues will also be integrated throughout the environment to enhance clarity, provide feedback, and guide users through interactions, ensuring that navigation and manipulation remain intuitive and accessible.

2.3 Performance Goals

Performance optimization is critical in maintaining a seamless and immersive user experience within virtual reality environments. One of the primary goals of this project is to ensure real-time interaction with minimal input delay, allowing users to manipulate and explore volumetric data smoothly and responsively. The system has been tested and optimized to sustain frame rates between 50 to 72 FPS on Oculus Quest 2, 3, and Pro headsets, providing a consistent experience without noticeable lag or motion sickness. To preserve the visual integrity of the data during interaction, the application is designed to prevent pixelation and rendering artifacts, particularly when users zoom in or rotate the image stack. These performance safeguards ensure that image clarity is maintained during manipulation, which is essential for accurate scientific analysis. Additionally, the system must remain stable when handling large datasets, such as those composed of 1024 × 1024 × 80 confocal microscopy image stacks. Efficient rendering and memory management techniques are employed to prevent crashes or slowdowns, ensuring that the data remains both interactive and scientifically valid throughout the user's experience.

2.4 Hardware & Software Considerations

The development of this project was carried out using the Unity Game Engine, a widely adopted platform for creating interactive 3D environments, particularly well-suited for virtual reality applications. To streamline cross-platform VR development and ensure compatibility with Oculus hardware, the XR Interaction Toolkit was utilized. This toolkit provides a robust foundation for implementing core VR interaction features such as grabbing, hovering, selecting, and interacting with user interface elements in a three-dimensional space.

The application was specifically tested on Oculus Quest 2 and Oculus Quest 3 headsets, which support room-scale tracking and is capable of running VR applications independently. These devices also allow for detailed performance evaluation in realistic usage scenarios. The primary interaction method employed in the project was through handheld VR controllers equipped with joysticks. These joysticks facilitated object manipulation, locomotion, and menu navigation, enabling a more immersive and intuitive user experience. The combination of Unity, the XR Interaction Toolkit, and Oculus hardware provided a stable and responsive environment for implementing and evaluating the VR interface for scientific data visualization.

Materials & Methods

3.1 Data Collection and Preparation

The biological specimens used for this study were provided by the Center for Biological Imaging and the Niepielko Lab at Kean University. High-resolution images of pollen grains were captured using a confocal microscope. These images were then converted into raw .jpg files for use in the VR visualization pipeline.

From this dataset, a subset of sample images was selected based on clarity and textural detail to support the development and testing of the VR interface. Each sample represented a maximized view of a specific pollen grain type, chosen to emphasize the structural complexity needed for scientific analysis and interaction design.

The selected image stacks were uploaded into the Unity Game Engine to serve as the foundational dataset for volumetric visualization. Preparing these images in Unity allowed the research team to experiment with shader applications, user interface elements, and 2.5D interaction techniques in a controlled development environment.

3.2 Why Unity? Why Oculus?

Unity was selected as the development platform because of its cross-platform flexibility, its strong support for XR (extended reality) systems, and its integration capabilities for VR development via packages like the XR Interaction Toolkit. Unity's modularity allowed fast prototyping of interactive features while offering built-in tools for managing performance optimization across devices.

The Oculus Quest 2, Quest 3, and Oculus Quest Pro headsets were chosen for their high frame rate capabilities, their standalone operation (no PC required), and their broad developer support. Their compatibility with the XR Interaction Toolkit also made it easier to implement gesture-based and controller-based interactions, ensuring that the final product could achieve intuitive, natural-feeling manipulation of scientific data in VR.

3.3 Codebase Integration

This project builds upon the existing LayeredImageViewer codebase, a set of tools originally developed by previous members of the research group. The LayeredImageViewer provides a foundation for displaying confocal microscopy image stacks in Unity, enabling image layering, camera control, and depth navigation. The framework was designed to support scientific visualization of high-resolution biological datasets in virtual reality environments.

3.4 Shader/UI Development

As part of this study, custom shaders were developed using the CG programming language to enhance and tailor visualization features within the existing system. Early shader development was tested on a basic 3D cube object to prototype core interaction functionalities, including:

- Recognizing collision points between virtual elements
- Selecting objects based on user proximity or pointer interaction
- Moving selected objects within Unity's 3D world space

Once interaction fundamentals were confirmed, the full stack of pollen grain images was imported into Unity. The existing LayeredImageViewer shader structure was then extended to integrate volumetric rendering capabilities that maintained transparency, depth cues, and data fidelity.

These enhancements enabled layered visualizations with interactive cut-planes, transparency toggles, and UI adjustments. These features supported both 2.5D and full 3D interaction modes, laying the groundwork for additional features such as menu sliders, shader toggles, and performance optimization across different headsets.

3.5 Cut-Plane Vertex Equations and Geometric Shaders

To enable users to examine internal structures of the pollen grain model, a cut-plane interaction technique was implemented using a vertex-fragment shader pipeline within Unity. The cut-plane dynamically slices through the volumetric image stack, revealing cross-sectional data based on the plane's position and orientation.

The mathematical condition for transparency at any point is based on the relative position of a fragment to the cut-plane. Specifically, we define the cut-plane using:

- $\vec{p}$: the world-space position of the fragment (the point being rendered).
- $\vec{b}$: a base point on the plane (the clip base),
- $\vec{n}$: the normal vector of the plane (the clip normal),

A fragment is made transparent if the following condition is true:

$$(\vec{p} - \vec{b}) \cdot \vec{n} > 0$$

This equation determines whether a fragment lies on the clipped side of the plane by measuring the dot product between the direction from base to fragment and the plane's normal.

The vertex shader computes this logic at key points (e.g., corners of the image planes), and Unity's rendering pipeline interpolates values across the surface. In the fragment shader, we use these interpolated world coordinates to evaluate the equation and determine if the current pixel should be rendered or discarded. This results in a smooth slicing effect across layered volumetric data.

Contrary to our initial exploration with geometry shaders, the final implementation used a vertex-fragment approach, as it is better suited for LayeredImageViewer (LIV) and more efficient for stacked 2D textures. This method allows for real-time cut-plane manipulation without compromising performance.

3.6 Transparency Manipulation & Layer Blending

To further improve visual clarity and depth perception, alpha transparency control was incorporated around the cut-plane. This was implemented through interactive sliders in the user interface, allowing users to fine-tune how visible different sections of the volumetric data appear.

Transparency values were dynamically passed into the fragment shader, where alpha levels were adjusted depending on proximity to the cut-plane. This gave users the ability to blend internal and external structures, revealing subtle features otherwise hidden in dense layers.

When multiple slices / stacked planes were rendered together, the shader automatically blended overlapping layers using Unity's alpha blending mode. This preserved the layered composition of the microscopy data while enhancing visual coherence.

The transparency controls were integrated into a VR-accessible menu, giving users real-time feedback and control as they explored the 3D dataset. These features helped simulate the look and feel of "peeling back" layers in a physical specimen.

3.7 XR Interaction Toolkit Integration

To enable seamless user interaction in the virtual reality environment, this project utilized Unity's XR Interaction Toolkit (XRI) a widely adopted package that provides a standardized framework for building cross-platform XR applications. XRI supports a variety of extended reality (XR) devices and handles common interaction patterns such as object selection, manipulation, teleportation, and user interface (UI) control. As an official Unity package, it is maintained for compatibility with most current and future XR hardware, making it a reliable foundation for scalable VR development.

The toolkit was selected for its ability to support both stationery and room-scale VR experiences through its XR Origin system, which manages camera rigging and player positioning in virtual space. This was essential for ensuring that our platform could accommodate different types of VR headsets and user movement styles without requiring custom implementations.

Within the project, XRI was integrated to support several critical interaction features:

- Hovering over and selecting objects in the VR scene
- Grabbing and dragging layered volumetric data
- Interacting with canvas-based UI elements via VR controllers
- Navigating the space using teleportation mechanics
- Providing real-time visual feedback through highlight tints and line renderers

By using this toolkit, we were able to implement consistent and responsive interactions across our application, while also keeping the system modular for future extension. This included integrating input mappings for Oculus controllers and customizing interaction behaviors to suit the layered microscopy data environment.

Results and Performance

This section presents the progression of implementation stages developed in Unity, beginning with early 2.5D user interface (UI) tests and culminating in a fully interactive 3D volumetric visualization in VR. While several stages were initially created for learning purposes, each provided insight into the technical feasibility and usability of different interaction methods. These incremental steps also offered valuable feedback on user interaction design within layered microscopy datasets.

We describe both intermediate experiments used to explore 2.5D interaction techniques and shader behavior and later implementations that integrated VR controls, cut-plane slicing, and interface enhancements. The work reflects a layered development approach: building confidence in Unity and XR tools while gradually merging custom visualization features into the final system.

4.1 Step 1 in 2.5D: Familiarity with Unity Game Engine in World Space

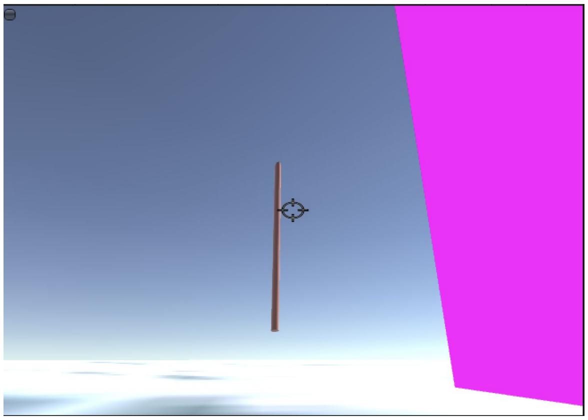

Figure 1: Unhovering and deselecting an object

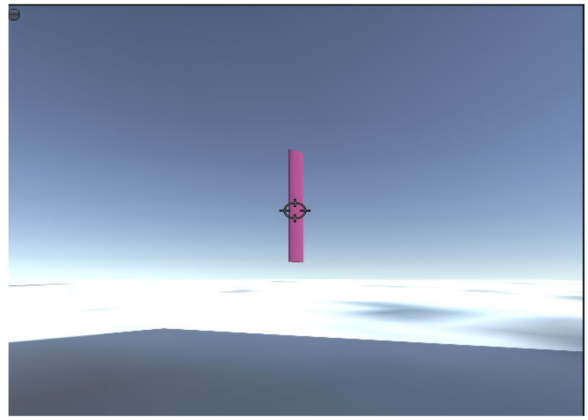

Figure 2: Hovering and selecting an object

The first implementation served as an introductory experiment to test object manipulation in a 2.5D Unity environment using standard desktop inputs. In this early build, a basic 3D cube represented a pollen grain, and interaction occurred in a flat scene rendered with a false 3D perspective.

User navigation was controlled using WASD keyboard inputs for directional movement up, down, left, and right while the mouse-controlled camera orientation. However, the setup required toggling between navigation mode and cursor mode, meaning users had to switch input states to move versus interact with objects. This added friction and disrupted the flow of interaction. Zooming in and out to examine the object and moving around the space was also a challenge. While a traditional mouse allowed for better control, operating on a laptop touchpad found that zooming was imprecise and difficult to manage, making detailed inspection of the object frustrating. Object selection was implemented through raycasting from the screen center, detecting when the virtual cursor intersected with the cube. Selection feedback was handled via image swapping, where the object visually changed upon interaction. However, this lacked the immediacy of more intuitive feedback methods like highlight outlines or glow effects, making it harder to confirm successful selection without hesitation.

Despite its simplicity, this prototype revealed the limitations of using conventional desktop input methods for scientific visualization. It offered early insight into the need for more fluid interaction models and stronger visual feedback when designing interfaces for volumetric data exploration, even in 2.5D environments.

4.2 Step 2 in 2.5D: Implementation of Layered Image Viewer as well as initial testing of cut plane shaders.

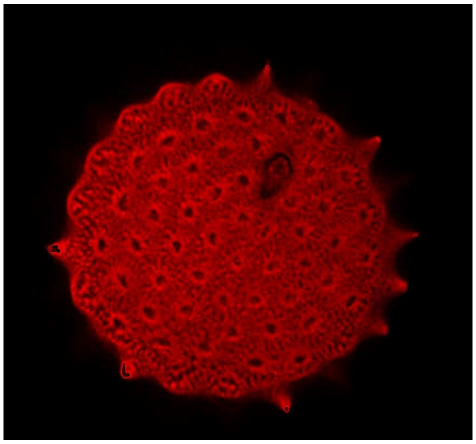

Figure 3: Sample of Ragweed Pollen Grain

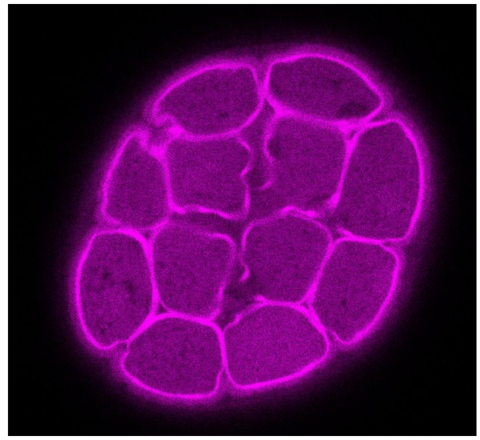

Figure 4: Sample of Golden Acacia Pollen Grain

Building on the earlier 2.5D click-and-drag interaction system, this stage focused on integrating real microscopy data into the rendering pipeline to test the behavior of cut-plane shaders. The goal was to assess how intuitive and effective the slicing interface would be when applied to layered biological images, simulating early 2.5D interaction within Unity.

Pollen grain samples, Ragweed and Golden Acacia, were selected from the confocal microscopy dataset based on their strong depth and texture characteristics. These qualities made them well-suited for evaluating how internal structural details respond to shader-based slicing. The microscopy image stacks were converted to .jpg format and imported into a Unity scene, where they were layered and aligned for interaction.

As shown in Figures 3 and 4, these sample stacks served as the test bed for observing how cut-plane shaders performed during click-and-drag exploration. This setup helped us begin bridging the gap between synthetic test objects and real-world biological data, laying the foundation for later full integration.

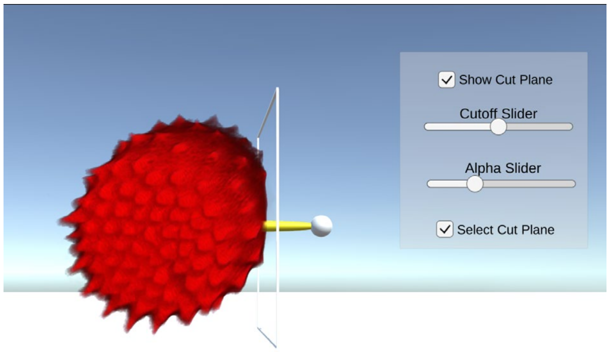

Figure 5: Initial setup of scene

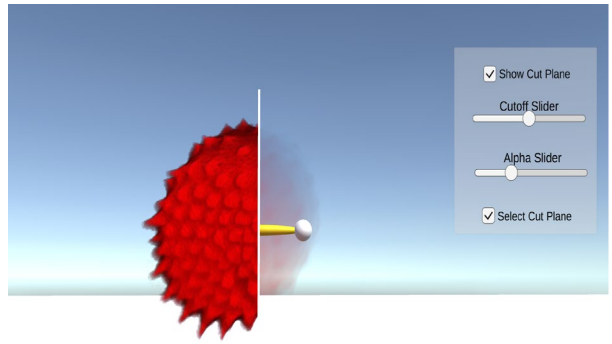

Figure 6: Cut plane vertex with visibility

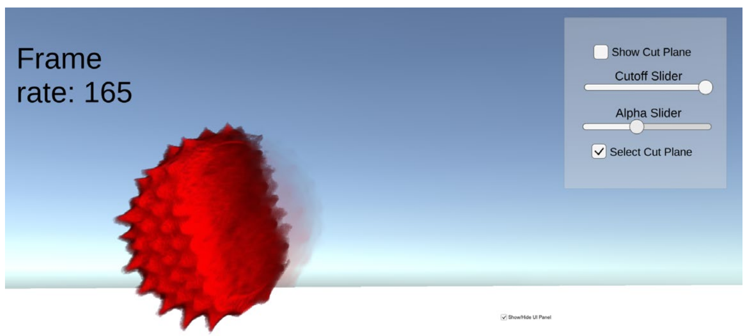

Figure 7: Cut plane vertex with visibility at angle

With the data loaded into Unity, volumetric shaders were applied to the layered images to enable real-time transparency control. These shaders allowed alpha blending, which visually merged the object's RGB values with the background color depending on user input, effectively making certain sections of the pollen grain more translucent. This blending effect is shown in Figure 5.

To enhance internal exploration of the data, a cut-plane feature was introduced, enabling users to define a vertical slicing plane through the volume. This allowed portions of the image stack to become transparent based on their spatial relation to the cut-plane. Specifically, fragments on one side of the plane for instance the right side was discarded or faded out depending on the alpha slider's value.

From a user interaction perspective, enabling and controlling the cut-plane required a multi-step process within the Unity UI. First, the user activated the cut-plane through a checkbox toggle located in the menu bar (Figure 6). Once enabled, two UI elements controlled its behavior:

- A position slider, which adjusted the plane's placement along the x-axis (sliding effect)
- A rotation slider, which tilted the plane to an angle around the x-axis (rotating effect)

Sliding the cut-plane involved dragging the position slider left or right, effectively moving the plane through the pollen model. This was straightforward and relatively easy to use. However, rotating the plane, especially in this 2.5D setup was less intuitive. Moving the rotation slider manually while interpreting how the visual output was changing in real time. Without VR input or gesture control, it lacked the natural "grab-and-rotate" motion possible in an immersive environment.

In Figure 7, the result of rotating the clip plane is shown, where tilting the cut-plane along the x-axis revealed internal perspectives of the pollen grain otherwise hidden in a purely vertical view. This UI-driven interface provided early insight into how different manipulation methods impact user understanding of volumetric data, and underscored the limitations of traditional 2.5D interaction for precise spatial exploration.

4.3 Step 3 in 3D: Integrating to VR environment using XR interactable toolkit (No volumetric data)

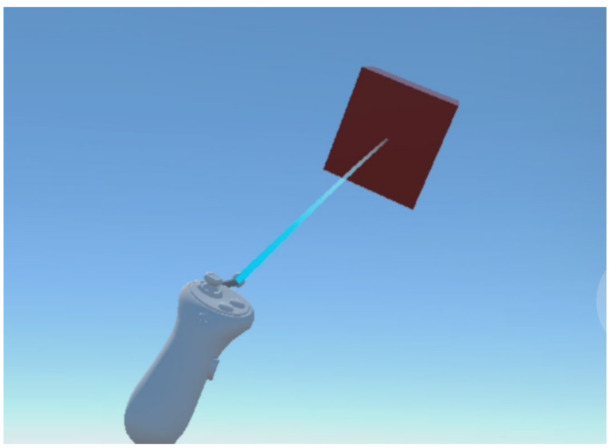

Figure 8: Grabbable object

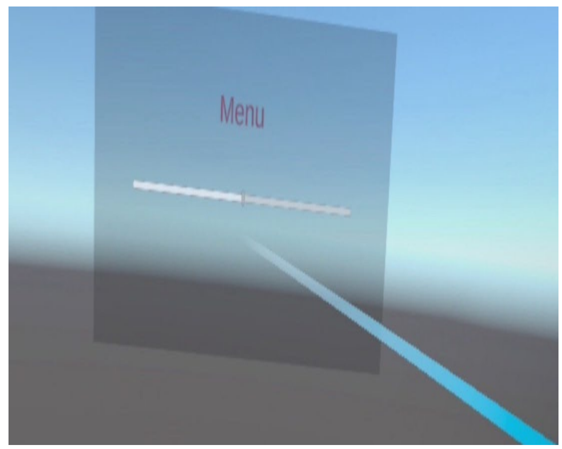

Figure 9: Interactable Sliders

While this stage was performed as a learning step towards the next version of the volumetric interface, we again will explore this to understand aspects of 3D interface are shown here.

Transitioning from 2.5D to a full VR environment involved implementing the XR Interaction Toolkit within Unity. Initial tests in XR included interacting with ready-made components, such as grabbable objects, gaze-based interactable, and UI panels.

Using VR controllers equipped with laser pointers, selecting and moving grabbable objects by pressing and holding the trigger button (Figure 8). Additionally, 2D UI elements such as sliders were adapted for VR control. As shown in Figure 9, parameters like transparency were interacting with a slider using the laser pointer, rather than pressing discrete buttons. Gaze interactors were also introduced, allowing the summoning of the menu panel at line of sight with a controller button press, reposition it as needed, and have it remain fixed relative to world space.

4.4 Step 4: Integrating the 3D controls with our Volumetric Rendering in VR

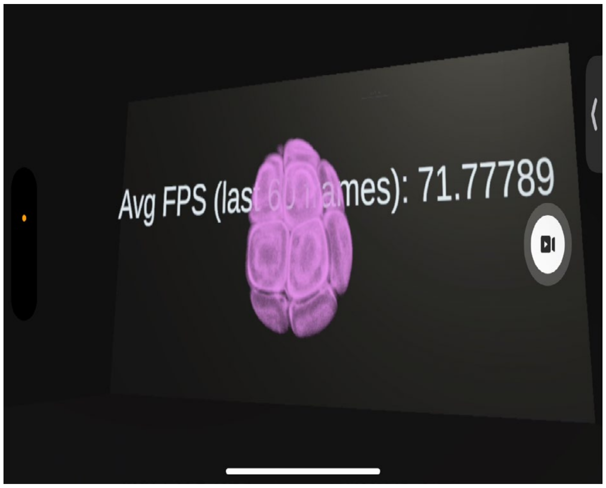

Figure 10: Pollen Grain Images

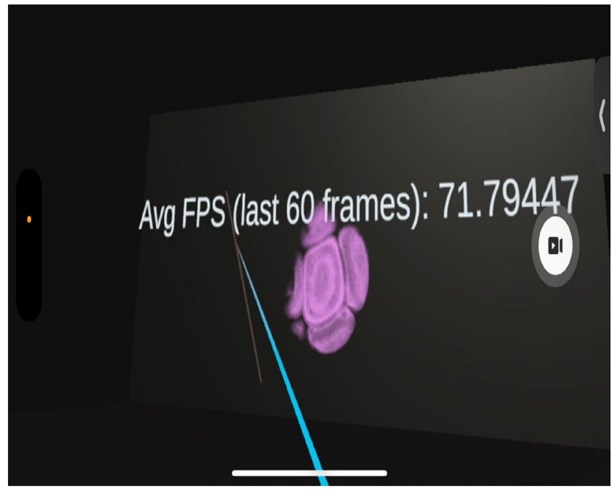

Figure 11: Manipulation cut plane vertex

Figure 12: Real-time Frame Rate

Further development involved uploading the confocal microscopy pollen images into the VR environment using the Layered ImageViewer codebase. These were placed with spacing along the y-axis, with background colors removed to create a volumetric visualization effect.

The cut plane from the 2D code was added, and interaction with the cut plane was changed to a "grab, rotate, and move" mode, where users could simply grab the cut plane and place it where desired. As sometimes this required being too close to the data to see it fully when placing the cut plane, the ability to grab the cut plane from a distance and rotate it with a thumb joystick on the Oculus controllers were also added, this was a built in feature of XR Interaction Toolkit.

As shown in Figures 10 and 11, users were able to interact with the cut-plane vertex in real time and slicing the volumetric image stack at different angles. Real-time frame rate monitoring indicated that the system performed at an average of 60 frames per second (Figure 12), ensuring smooth interaction and data integrity even with moderately sized datasets.

4.5 Step 5 in 3D: Designing a Readable and Accessible VR Menu

As the project progressed into a fully immersive VR environment, menu design became a critical component of the user experience. Inspired by Brett Jackson's Idea Engine, this stage explored how virtual menus could be repositioned, reoriented, and styled to remain accessible and readable within Natural User Interface (NUI) design. Initial prototypes used a traditional canvas-based menu panel (Unity quad object) placed directly in front of the user . However, early testing showed that a fixed menu when not aligned to the user's view could lead to disorientation and reduced usability [11]. This was particularly problematic during movement or rotation, where the menu could suddenly disappear or appear backward, forcing the user to reposition themselves physically.

To improve usability, we implemented a system that dynamically evaluated the dot product between the menu quad's normal vector and the user's camera forward vector. If the angle exceeded a certain threshold (i.e., if the user rotated behind the panel), the menu would automatically flip to face the user. This approach ensured persistent readability regardless of orientation. We explored multiple interaction models to determine the most natural layout:

- Menus that always face the user: These maintained constant readability but felt "stuck" to the head, breaking immersion.
- Floating world-space panels visible from one side: Visually clean, but easily missed when navigating around the environment.
- Dual-sided panels fixed in space: Provided balance between realism and functionality, with the added benefit of visual persistence during exploration.

Ultimately, we chose a design where the menu remains fixed in world space but is rendered double-sided, allowing it to be read from both the front and back without needing to rotate it manually. This approach aligned with NUI principles by reducing cognitive effort and increasing spatial predictability, allowing users to naturally return to the same control panel without overthinking its location or orientation.

This decision was heavily influenced by Jackson's emphasis on menu accessibility from any angle, one of the defining features of Idea Engine. Our implementation echoed his approach of auto-flipping menus and allowing gesture-driven navigation, though our interface relied on controller inputs instead of full gesture recognition.

From a user interaction perspective, this menu provided access to essential functions like toggling the cut-plane and enabling object selection. The spatial anchoring of the UI helped reinforce environmental awareness an important aspect of 3D spatial memory and supported fluid transitions between exploration and interface control.

By treating the menu as a persistent and spatially meaningful object, we took early steps toward establishing a fully embodied NUI system. While interaction was still driven by controller buttons rather than hand gestures, this phase provided meaningful insight into how VR interfaces can be designed to feel natural without relying solely on traditional screen-based UI elements.

4.6 Step 6 in 3D: Zoom in circular focus

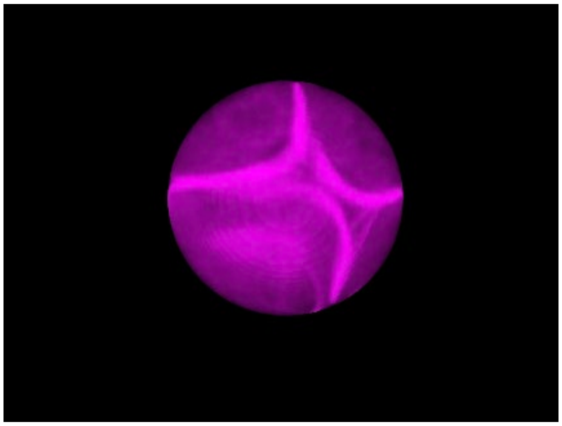

Figure 13: Zoom with Hard edge

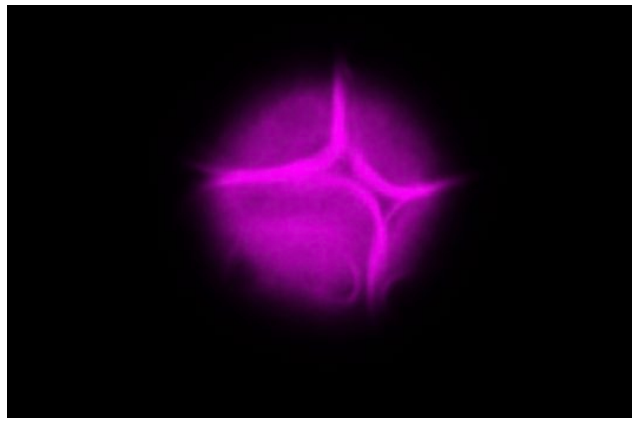

Figure 14: Zoom with Soft edge

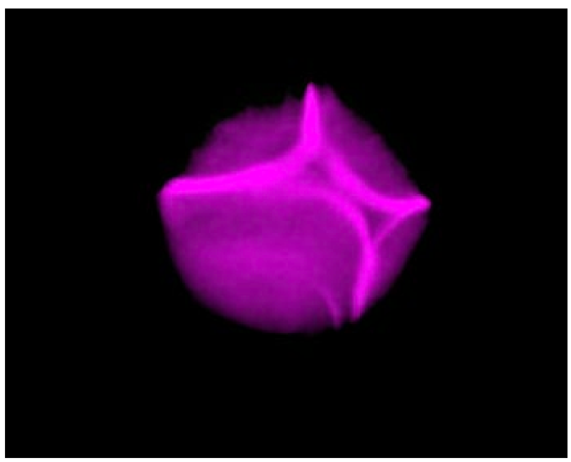

Figure 15: Exponential Zoom

The final feature implemented in this phase was a circular zoom focus, designed to enhance the user's ability to examine fine internal structures of the pollen grain without disrupting the surrounding volumetric view. This tool provided localized magnification within a circular area of the image stack, functioning similarly to a magnifying lens overlay in a 3D environment. Magnification was applied dynamically within the boundary of the zoom circle, enlarging features like surface ridges or density gradients without distorting the rest of the scene. This allowed users to preserve their spatial orientation while focusing on fine details.

Three zoom techniques were tested using Golden Acacia pollen data. Figure 13 shows the hard edge zoom, which clips the focus area with a defined boundary. Figure 14 demonstrates the soft edge zoom, which applies a smoothed step to blend the magnified region with the surrounding image. Figure 15 presents an exponential zoom, where magnification falls off smoothly from the center, maintaining visibility within a defined radius while preserving context.

This selective zoom approach addressed a common challenge in volumetric visualization: balancing global spatial awareness with local detail inspection. Unlike global zoom, which can distort perspective or disrupt orientation in VR, the circular zoom maintained immersion and fluid navigation. Importantly, the feature supports Natural User Interface (NUI) principles by enabling intuitive inspection without requiring mode switching. Users remained within the environment and interacted with zoom parameters naturally, enhancing both clarity and interactivity for volumetric scientific analysis.

DISCUSSION

The Unity-built user interface utilized customized shaders and precisely defined cut-plane parameters to ensure full functionality. Integration with Oculus Quest 2, Quest 3, and Pro headsets was achieved using the XR Interaction Toolkit, establishing a seamless connection between the Unity game scene and the Oculus headset's 3D world space. The system operated optimally at an average of 60 FPS, featuring shader optimization and controller-based cut-plane manipulation within VR. The best data performance occurred at a resolution of 1024 × 1024 × 80. Beyond this resolution, interaction quality declined, leading to frame rate drops, image distortion, and occasional system crashes.

While the software achieved strong technical performance, it also revealed several important constraints. Virtual reality (VR) for scientific visualization is still relatively new and costly, raising questions about how many institutions, researchers, and industries are willing to invest in this technology. However, VR's increasing familiarity among newer generations suggests that adoption may grow, especially as it aligns with educational, medical, and commercial trends. The software developed here, although tested with pollen grains and fruit fly cells, can be extended to other scientific applications, such as magnetic resonance imaging (MRI) scans, weather datasets, and biological structures. Moving beyond traditional 2D and 2.5D visualization, VR enables scientists to interact with layered hyperstack data in a truly immersive, hands-on environment. Rather than relying on mouse clicks or keyboard shortcuts, users can engage with data more naturally, like how they would handle physical objects.

Cost, however, remains a significant barrier. High-quality VR setups, combined with custom software development, can easily reach half a million U.S. dollars when scaled for enterprise use. Nonetheless, the hands-on, immersive interaction capabilities introduced by VR may justify investment for industries where data interpretation is critical.

Existing studies reinforce the potential of VR for serious applications. User feedback and quantitative evaluations show very few differences between real-world conditions and simulated VR environments, even under varying simulation parameters [13]. Other studies emphasize that VR simulation can successfully evaluate the usability and user experience of augmented reality (AR) applications [13]. Despite these strengths, there are known disadvantages to long-term VR headset usage, including visual fatigue, headaches, nausea, dizziness, and eyestrain [12]. Spatial awareness also poses safety risks, as users immersed in VR environments may overlook physical obstacles in their real-world surroundings [14]. Developers must account for these risks by designing spatial safety boundaries and encouraging safe usage habits. Human sensory processing also plays a crucial role in virtual environments. Research shows that proximity to stimuli can affect perception, while spatial memory, distance judgment, and peripheral vision influence user behavior in VR [15]. Systems like Meta Quest headsets offer customizable boundaries that users can reset in different environments to mitigate these risks. Human-computer interaction (HCI) principles will continue to shape VR development. Modern HCI focuses on usability, accessibility, and emotional fulfillment [16]. Designing visualization tools that are not only powerful but also easy to use, safe, and satisfying is essential to drive broader adoption of VR in scientific research.

Continued development of visualization tools in VR requires rethinking traditional user interface paradigms. The VR user interface (UI) developed in this project, designed for hyperstack microscopy images, successfully enabled transparent volumetric rendering, cut-plane manipulation, and real-time interaction through controllers and an immersive menu system. Unity's XR Interaction Toolkit and Oculus hardware provided a strong technical foundation, but further enhancements are possible. Future developments should focus on embedding UI panels more directly into the VR environment, incorporating color enhancements, additional shape-based analysis tools, and shader-based image improvements.

Looking ahead, expanding compatibility across different headsets, such as Apple's Vision Pro and other OpenXR-supported devices, is a major priority. The current implementation was optimized for Oculus devices, but broader compatibility would improve accessibility and scalability. Additionally, integrating hand-tracking capabilities to replace or supplement controllers could make interactions feel even more natural, providing users with the sensation of holding and manipulating data directly. Finally, extending the platform to mobile devices could democratize access to volumetric visualization tools, allowing users to explore 3D scientific data on portable, everyday technology. This would bridge the gap between high-end VR environments and more accessible platforms, expanding the reach of immersive scientific analysis tools.

5.1 User Interface Interactions: 2D vs 2.5D vs 3D

In the early stages of development, I began with a familiar 2D setup. Like most interfaces, this involved working on a flat screen using a mouse and keyboard. While I was able to view the stacked confocal images and see them layered, the interaction felt very removed. There was no true depth, and manipulating the images felt abstract. I could scroll and click through layers, but I was not able to interact with the data in a way that felt connected or hands-on. It was functional, but not immersive. As development progressed, I transitioned to a 2.5D environment within Unity. This added some sense of depth by positioning image layers in 3D world space, but I was still working on a laptop screen. I was able to implement key UI features like highlighting specific slices of the data and introducing the cut-plane vertex to simulate dissection. However, interaction at this level came with challenges. Using a mouse to navigate a 3D world felt awkward at times, especially when trying to zoom, rotate, or position objects along the x, y, or z axes. Zooming in required careful scrolling and positioning in the world space, which could easily lead to disorientation. While more immersive than 2D, it still didn't provide the sense of holding or truly dissecting the data.

The final phase involved uploading everything into a fully immersive 3D virtual reality environment using Oculus headsets. This made a significant difference. In VR, everything felt more natural. I could move around the data, grab it with the controller, rotate it in space, and adjust the cut-plane by actually “slicing” through the pollen grain using motion. The addition of zoom features inside VR felt intuitive, no longer was I scrolling awkwardly, but rather physically moving closer or using hand-controlled zooming functions that made the experience feel like true analysis.

From my experience, the biggest shift wasn’t just visual it was how naturally interaction felt. In 2D, you observe. In 2.5D, you start to manipulate. In 3D VR, you engage. The transition from clicking to grabbing, from scrolling to slicing, and from dragging a mouse to reaching out with a hand changed the way I could analyze the data. It brought me closer to the concept of Natural User Interfaces (NUIs), where interaction feels instinctive rather than learned. That shift marked a key evolution in the platform—not because 3D is inherently superior, but because it aligned the mode of interaction with the dimensionality of the data itself. As Edward Tufte argues in *The Visual Display of Quantitative Information*, unnecessarily increasing visual dimensions can lead to confusion rather than insight. While 2D and 2.5D environments remain ideal for many visualization tasks—such as scatter plots or simple spatial overviews, immersive 3D becomes most effective when the data itself possesses spatial depth that benefits from embodied interaction [17].

5.2 Best Practices in 2.5D and 3D Design

From the development process, several best practices emerged when transitioning from 2.5D to 3D design:

- **Preserve Depth Perception**:
  In 2.5D, stacking layers correctly and maintaining consistent world space orientation helped simulate depth, even on flat screens.
- **Minimize Disorientation**:
  Smooth, gradual zooming controls and consistent navigation cues were essential to reduce user disorientation during 2.5D exploration.
- **Prioritize Intuitive Interaction**:
  In 3D VR, movement needed to mimic natural human behaviors like grabbing and rotating, rather than abstract clicks.
- **Design for Spatial Awareness**:
  UI panels and menus in VR had to respect real-world spatial relationships, keeping objects within reachable distance and always readable.
- **Focus on User Control**:
  Giving users the ability to freely move, zoom, rotate, and slice without complex menus improved engagement and lowered the learning curve.

These practices guided the final implementation, ensuring the ability to shift seamlessly between exploration, manipulation, and analysis modes.

5.3 Challenges Encountered and Design Pivots

Several challenges emerged during development:

- **2D to 2.5D to 3D Transition**:
  Initial development was conducted in a 2.5D environment using a mouse and keyboard. Transitioning to full VR interaction required rethinking user input, particularly adapting pointer-based interactions to laser pointers, gaze tracking, and joystick navigation.
- **Interaction Consistency**:
  In early VR tests, the cut-plane control felt unintuitive when rotated in 3D space. Additional visual indicators and grab-handles were added to improve user orientation and control.
- **Menu Readability**:
  A major pivot was inspired by Brett Jackson's Idea Engine concept. Instead of a static UI panel, the menu was designed to appear in front of the user using gaze tracking and remain readable from both sides. This significantly improved usability in the VR environment.
- **Performance Optimization**:
  Fine-tuning shaders, adjusting polygon counts, and balancing rendering load were essential to maintain frame rates above 50 FPS during complex interactions.

Each pivot strengthened the user experience and aligned the system closer to Natural User Interface (NUI) principles.

Conclusion

This project successfully developed a fully immersive virtual reality (VR) platform for visualizing and interacting with scientific microscopy data, particularly pollen grains captured through confocal microscopy. Using Unity, customized volumetric shaders, and the XR Interaction Toolkit, a user interface was built that allowed users to naturally manipulate and explore hyperstack datasets through intuitive actions such as grabbing, slicing, rotating, and zooming. Functional goals such as the implementation of cut-plane manipulation, transparency adjustment, and interactive menus were achieved, and design goals focused on usability and natural interaction were successfully met.The transition from 2D to 2.5D to full 3D VR interaction significantly improved the user experience, demonstrating that immersive environments can overcome many of the limitations associated with traditional data visualization methods. Brett Jackson's Idea Engine principles also played a key role in shaping menu design and spatial interaction strategies, emphasizing the importance of maintaining comfort, clarity, and natural motion within VR.While technical challenges such as performance optimization and intuitive menu readability were encountered, strategic design pivots allowed the system to achieve high usability and maintain scientific data integrity. This project highlights the critical role of Natural User Interfaces (NUIs) in lowering the learning curve and making VR a more accessible tool for researchers.

Future work will focus on expanding hardware compatibility beyond Oculus devices, improving hand-tracking integration for more natural interactions, and adapting the platform for use on mobile devices. By continuing to refine and democratize immersive scientific visualization tools, this research paves the way for a future where complex datasets are explored not through screens, but through intuitive, hands-on virtual environments bringing researchers closer to their data than ever before. As noted by Korkut and Surer [18], while VR is increasingly adopted across many fields, the success of immersive visualization depends on critically addressing challenges related to interaction design, usability, and visualization guidelines. This project meets that need by introducing a platform tailored to both the technical and cognitive demands of scientific microscopy data. Through thoughtful integration of NUI principles, intuitive interface design, and performance-optimized rendering, it demonstrates how immersive technologies can bridge the gap between raw scientific data and user-centered exploration.

References:

[1] S. P. Feyer, B. Pinaud, S. Kobourov, N. Brich, M. Krone, A. Kerren, M. Behrisch, F. Schreiber, and K. Klein, “2D, 2.5D, or 3D? An exploratory study on multilayer network visualisations in virtual reality,” *IEEE Trans. Vis. Comput. Graphics*, vol. 30, no. 1, 2024. doi: 10.1109/TVCG.2023.3327402.

[2] J. Lemoine, V. Eric, and F. Anthony, “Evaluating usability and user experience of AR applications in VR simulation,” *Front. Virtual Real.*, vol. 3, 2022. doi: 10.3389/frvir.2022.881318.

[3] S. Zollmann, T. Langlotz, R. Grasset, W. H. Lo, S. Mori, and H. Regenbrecht, “Visualization techniques in augmented reality: A taxonomy, methods and patterns,” *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 9, 2021. doi: 10.1109/TVCG.2020.2986247.

[4] V. Natarajan, “Scientific visualization: From data to insight,” *Resonance: J. Sci. Educ.*, vol. 18, no. 7, pp. 615–629, 2013. doi: 10.1007/s12045-013-0081-8.

[5] L. Charters, “Confocal microscopy: Seeing what does not meet the eye,” *Ophthalmology Times*, vol. 45, no. 3, p. 34, 2020.

[6] J. Jonkman, C. Brown, G. Wright, K. Anderson, and A. North, “Tutorial: Guidance for quantitative confocal microscopy,” *Nat. Protoc.*, vol. 15, pp. 1585–1611, 2020. doi: 10.1038/s41596-020-0313-9.

[7] T. Kojić, R. Spang, M. Vergari, L. Meier, S. Möller, and J.-N. Voigt-Antons, “Effects of user factors on user experience in virtual reality: Age, gender, and VR experience as influencing factors for VR exergames,” *Qual. User Exp.*, vol. 8, no. 1, 2023. doi: 10.1007/s41233-023-00056-5.

[8] Y. Shuran, “Sentiment and storytelling: What affect user experience and communication effectiveness in virtual environments?” *Journalism and Media*, vol. 4, no. 3, pp. 30–41, 2022. doi: 10.3390/journalmedia4010003.

[9] C. S. González and A. F. Rodríguez, “Usability and user experience evaluation of natural user interfaces: A systematic mapping study,” *ResearchGate*, 2020. [Online]. Available: https://www.researchgate.net/publication/344864818_Usability_and_user_experience_evaluation_of_natural_user_interfaces_a_systematic_mapping_study

[10] D. Vidakovic, “From screens to Extended Reality: XR is transforming interfaces,” *Push Your Pixels*, Aug. 24, 2023. [Online]. Available: https://pushyourpixels.com/articles/from-screens-to-extended-reality-xr-is-transforming-interfaces

[11] B. Jackson, “Exploring XR user interface interactions in ‘Idea Engine’,” *Road to VR*, Jan. 4, 2024. [Online]. Available: https://www.roadtovr.com/xr-interface-interactions-idea-engine-brett-jackson/

[12] S.-M. Choy, E. Cheng, R. H. Wilkinson, I. Burnett, and M. W. Austin, “Quality of experience comparison of stereoscopic 3D videos in different projection devices: Flat screen, panoramic screen and virtual reality headset,” *IEEE Access*, vol. 9, 2021. doi: 10.1109/ACCESS.2021.3049798.

[13] J. Lacoche, E. Villain, and A. Foulonneau, “Evaluating usability and user experience of AR applications in VR simulation,” *Front. Virtual Real.*, vol. 3, 2022.

[14] F. Buttussi and L. Chittaro, “Acquisition and retention of spatial knowledge through virtual reality experiences: Effects of VR setup and locomotion technique,” *Int. J. Hum.-Comput. Stud.*, vol. 177, 2023. doi: 10.1016/j.ijhcs.2023.103067.

[15] S. Seinfeld, T. Feuchtner, J. Pinzek, and J. Muller, “Impact of information placement and user representations in VR on performance and embodiment,” *IEEE Trans. Vis. Comput. Graphics*, vol. 28, no. 3, 2022. doi: 10.1109/TVCG.

[16] R. Pushpakumar et al., “Human-computer interaction: Enhancing user experience in interactive systems,” *E3S Web Conf.*, vol. 399, 2023. doi: 10.1051/e3sconf/202339904037.

[17] E. R. Tufte, *The Visual Display of Quantitative Information*, Cheshire, CT: Graphics Press, 1983.

[18] E. H. Korkut and E. Surer, “Visualization in virtual reality: A systematic review,” *Virtual Reality*, vol. 27, no. 2, pp. 1447–1480, Jun. 2023. doi: 10.1007/s10055-023-00753-8.

[19] *XR Interaction Toolkit Documentation*, Version 3.0.1, Unity Technologies. [Online]. Available: https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@3.0/manual/index.html