IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 20 April 2026, accepted 7 May 2026, date of publication 12 May 2026, date of current version 21 May 2026.

Digital Object Identifier 10.1109/ACCESS.2026.3692426

APPLIED RESEARCH

VR BioTalk—Hands-Free Visual Analytics of Phenotyping Data Using Natural Conversation

JORGE VAZQUEZ[1], SHUWEN YANG[1], YIQUN ZHANG[2], JEFFREY DEMIEVILLE[3], BRENNAN HUPPENTHAL[4], NIRAV MERCHANT[5], VOICU POPESCU[1], (Member, IEEE), ALEJANDRA MAGANA[2], (Associate Member, IEEE), DUKE PAULI[3], AND BEDRICH BENES[1], (Senior Member, IEEE)

[1]Department of Computer Science, Purdue University, West Lafayette, IN 47906, USA
[2]School of Applied and Creative Computing, Purdue University, West Lafayette, IN 47906, USA
[3]School of Plant Sciences, The University of Arizona, Tucson, AZ 85721, USA
[4]Department of Computer Science, The University of Arizona, Tucson, AZ 85721, USA
[5]Data Science Institute, The University of Arizona, Tucson, AZ 85721, USA

Corresponding author: Bedrich Benes (bbenes@purdue.edu)

This work was supported in part by the National Science Foundation under Grant 2506783, Grant 2417510, Grant 2412928, Grant 2309564, Grant 2102120, Grant DBI-2417511, Grant DBI-0735191, Grant DBI-1265383, and Grant DBI-1743442; in part by United States Department of Agriculture-National Institute of Food and Agriculture (USDA-NIFA) under Grant 1032382 and Grant 1032672; and in part by the Department of Energy through the Biological and Environmental Research (BER) Program under Grant DE-SC0023305.

ABSTRACT We introduce, implement, and test *VR BioTalk*, a hands-free, immersive, voice-controlled visual analytics system for phenotypic data. Our system does not require any programming knowledge. Yet, it enables users to receive an interactive solution to complex tasks involving large datasets through simple verbal commands, such as ''Show me all leaves smaller than the average and calculate their leaf area index.'' We claim three main contributions: 1) preprocessing and feature extraction of point cloud data for interactive visual analytics, 2) development of a novel interface that converts user speech into commands, and 3) an immersive VR visualization that executes the commands and displays the results in VR. The speech recognition system's precision has been validated on 416 spoken commands across 13 English accents, with an accuracy of around 99.7% for transcription and 94% for command recognition. The visualization averages 63 FPS, and the system's response time is approximately 1.25 seconds. We tested *VR BioTalk* on several tasks that would otherwise require extensive programming knowledge. We tested our system with 9 participants, and the results show that *VR BioTalk* is highly usable, engaging, and easy to use, enabling experts with no programming background to explore large phenotyping datasets and generate hypotheses in natural language.

INDEX TERMS Conversational models, large datasets, point clouds, VR, visual analytics.

I. INTRODUCTION

Large biological datasets are becoming increasingly prevalent in plant science research due to the proliferation of high-quality sensors (e.g., LiDAR, high-resolution RGB, thermal, and hyperspectral cameras) and platforms such as unoccupied aerial vehicles (UAVs), small rovers, tractor-based platforms, and large phenotyping facilities. However, one of the main problems preventing researchers in agriculture and plant biology from extracting information from these datasets is the domain gap; researchers often lack the computer science knowledge to handle the volume and complexity of modern datasets.

On the one hand, efficient data extraction requires knowledge of computational pipelines and programming skills that may not be readily available to life science researchers. On the other hand, experts in computer science often cannot express biological hypotheses in computational terms [5]. Some estimates suggest that only 20-40% of acquired data is actually used to investigate research hypotheses [4].

The associate editor coordinating the review of this manuscript and approving it for publication was Ayman El-Baz.

FIGURE 1. ***VR BioTalk*** **allows the user to inspect the data by moving their head and issuing verbal commands, such as "show me plants taller than 10% of the average height and show the graph of its distribution". The system responds by providing visual output. The user does not need programming skills, yet can generate hypotheses and analyze large datasets through simple dialogue.**

For example, the University of Arizona's Field Scanner (referred to as the "gantry" Figure 3) can generate 10 TB of data daily and, during a 115-day season, can generate over 1 petabyte of data. This data volume is so large and complex that asking meaningful questions and performing fundamental exploratory analysis is nearly impossible.

There is strong evidence that visual analytics tools can support the understanding of complex data [1], [2]. This is especially true for data visualizations that correspond to real-world objects, such as plants. While photorealistic visualization is helpful, the display can utilize additional visual cues, such as different colors or transparency, to guide users to specific areas of interest. It can also overlay data with additional information, such as graphs and statistics, as shown in Figure 7. While 2D desktop displays provide sufficient detail and means of interaction, a more natural visualization (i.e., closer to human experience) is to immerse the user in the dataset using virtual reality (VR) headsets [3]. This is a form of embodied cognition [30], in which head movements are translated into direct control over the virtual camera's position and orientation, conveying to the user a sense of presence in the 3D dataset. However, one key problem with immersive visualization systems is their control [31], as they cannot use traditional keyboard-and-mouse graphical user interfaces (GUIs).

We introduce *VR BioTalk*, an immersive visual analytics system designed to bridge the gap between large phenotyping datasets and domain experts. Our approach enables interactive analysis of biological datasets through natural language. This work builds on three critical assumptions. First, immersive visualization, such as VR, enables intuitive interpretation of 3D data. It is natural to be virtually present in the dataset, enabling embodied cognition. Second, computer-based visualizations support the superposition of additional data, such as graphs, visual cues, and dynamic displays. Third, users find it difficult to formulate data analytics queries in code. Indeed, users might lack the required computer science skills, and even when they do possess them, expressing queries in code can be too slow, precluding the highly interactive query/inspect pipeline that is a prerequisite for seeding scientific insight. Instead, users should be allowed to express their data analytics requests naturally, through verbal commands in their language, for example, by saying "Show me all leaves smaller than the average and calculate their leaf area index". We demonstrate the implementation of *VR BioTalk* and its ability to perform natural-language-controlled tasks by running several complex tasks.

Information Extraction: One key task for visual analytics is information extraction from data, and *VR BioTalk* addresses this issue by working with a set of traits that can be extracted directly from point clouds and are directly mapped to the voice interface. Phenotyping facilities commonly collect images (e.g., RGB, hyperspectral, or infrared) and point clouds, either from LiDAR sensors or via photogrammetry. These unstructured data do not include explicit information about plant height, volume, leaf area index, number of leaves, or other morphometric traits. This information needs to be extracted, and the existing algorithms can be classified

depending on the input data. While many methods have been developed for trait extraction and 3D reconstruction from images [40], the present work focuses on point clouds collected by the gantry phenotyping system, which are transferable to other point cloud datasets [18].

Great effort has been made to develop robust machine-learning-based algorithms that leverage the large size of these datasets [23]. This has led to the development of various open-source software tools for high-throughput phenotyping [21], [22], which analyze data collected by sources such as UAVs [19], [20], necessitating phenotyping programming skills or software systems that can automatically extract trait information. Furthermore, data analytics of the resulting traits must be performed after extracting trait information to identify patterns, assess traits, or evaluate various genotypes. Previous research indicates a need for adaptable, cost-effective, and sophisticated data analysis infrastructures [18].

Visualization: 3D data visualization is one of the early and most important tasks assigned to computers, with now over 50 years of research. What makes phenotyping data visualization unique is the need to display large 3D datasets, often as images or point clouds [41]. Direct visualization of these datasets is challenging due to the large amount of data, which often exceeds the memory capacity of current GPUs. Bringing the insight-forming benefits of VR visual analytics to scientific applications like ours requires managing dataset complexity. The goal is to compute a subset of the original dataset that is both small enough to be rendered comfortably on the VR headset and sufficiently rich to obtain output frames of a quality that approaches that of frames rendered from the entire dataset [53]. Dataset complexity management takes multiple orthogonal approaches that can be used in conjunction. One is view frustum culling, which pre-groups dataset elements into bins and discards those outside the user's current view. Another approach is occlusion culling, which discards dataset elements that are hidden by elements closer to the user [54], [55]. A third approach is geometric simplification and level-of-detail adaptation [42], [56], which replaces the original representation of a part of the dataset with a simpler one. The goal is to find the simplest representation that preserves the original visual fidelity, for example, by replacing a cluster of distant points with a single point. Recent deep learning algorithms utilize the conversion of large datasets into deep neural data structures, such as Gaussian splats [43], [44]. These methods often cannot be used in real time, as they require training and are unsuitable for use with VR headsets.

Interaction is another important part of *VR BioTalk*. The traditional approach to user interaction for visualizing phenotyping data relies on 2D desktop views accessed via a mouse and keyboard. VR applications use head tracking to position the camera, enabling immersive interactive experiences. Traditional methods rely on hand control via 3D pointers for selection and numerical or textual input, which is tedious and error-prone, leading to fatigue, latency, and discomfort [45]. We argue that recent advances in Large Language Models (LLMs) and voice-based interfaces have the potential to overcome these problems by enabling natural-language interaction [33]. A recent work [28] demonstrates a proof of concept for complex VR interaction via simple spoken commands. *VR BioTalk* builds on the hands-free concept by leveraging advances in function calling, enabling LLMs to use external tools to execute the user's instructions [34]. Recent models, such as Llama3 [35] and Qwen2.5 [10], have been fine-tuned to respond to user requests by calling commands.

VR BioTalk aims to create a system that encompasses most of the high-throughput pipeline, enabling the use of real-world data, trait extraction, and data analytics within an immersive VR environment. Specifically, our goals are to (a) process large datasets generated from the University of Arizona Field Scanner and extract trait data from them, (b) provide an immersive environment where trait data analytics can be easily computed and visualized, and (c) allow for voice-based visual analytics that allows for complex interaction and analysis without the need for programming.

We validated the *VR BioTalk* feedback to show its real-time performance. Specifically, we evaluated *VR BioTalk* with respect to usability, cognitive load, and presence based on established questionnaires [57], [58], [59]. Evaluating usability, cognitive load, and presence in interactive systems can determine the quality of the user experience. Specifically, usability ensures that users can interact efficiently with the system, cognitive load helps determine whether the system supports the user's mental processing, and presence may influence engagement and motivation.

II. METHODS AND MATERIALS

The *VR BioTalk* pipeline has three stages (see Fig. 2). (1) The plants are grown, and raw data is collected as described in Sect. II-A, and (2) traits are extracted in a preprocessing step as described in detail in Sect. II-B. The raw data and traits are then utilized in the VR system (3), which implements visualization and interaction for Hands-Free Visual Data Analytics (Sect. II-C).

A. DATA COLLECTION

1) DATA AND FIELD EXPERIMENTS

In 2020, a population of ethyl methanesulfonate (EMS)-mutagenized BTx623 sorghum [*Sorghum bicolor* (L.) Moench], the genotype used for the reference genome, was grown and evaluated at the Maricopa Agricultural Center in Maricopa, AZ (33° 04'37" N, 111°58'26" W, elevation 358 m). The soil type is a Casa Grande sandy loam (fine-loamy, mixed, superactive, hyperthermic Typic Natrargids). The population consisted of 428 genotypes and has been previously characterized [24], [25]. The population was grown and evaluated under contrasting irrigation conditions: well-watered (WW) and water-limited (WL). The trial was planted on April 20 (Julian Day 110) in a partially

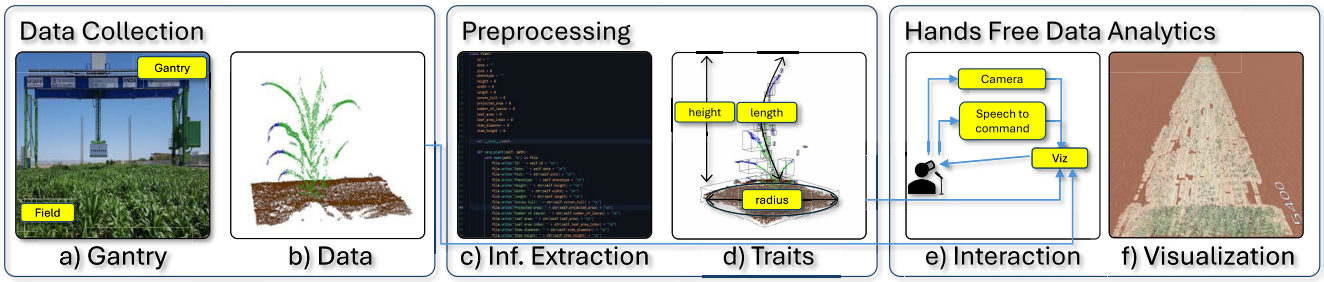

FIGURE 2. ***VR BioTalk*** **pipeline: The field (a) is scanned and the data is stored as point clouds (b). During preprocessing, several phenotypical traits are extracted and stored (c-d). During interactive visual analytics (e-f), the *VR BioTalk* engine receives the camera location and real-time voice input. The voice is converted into commands that are executed (e), and the results are visualized (f).**

replicated, incomplete-block design, with 94 genotypes replicated within each irrigation treatment, while the remaining 334 genotypes were observed only once per irrigation treatment.

The order of entries within an irrigation treatment was randomized, and 18 plots per irrigation treatment, also randomly assigned, were planted to a common check variety. Experimental units were one-row plots, 3.5 m in length with a 0.5 m alley at the end of each plot and inter-row spacing of 0.76 m; plots were thinned to a density of five plants per plot (1 plant per 0.7 linear m) after crop establishment, which occurred on June 8 (Julian day 159). Conventional cultivation practices for sorghum production (fertilizer application rate/amount, weed/insect control, etc.) in the low desert of the Southwest US were employed [26]. Weather data were recorded by the Arizona Meteorological Network (AZMET) weather station located at MAC and 738 m from the field where the experimental sorghum was grown [27]. Subsurface drip irrigation was used to establish and maintain soil moisture conditions. Pressure-compensated drip tape (DripNet PC, Netafim, Tel Aviv, Israel) was buried at a depth of approximately 0.15 m below the soil surface, and seeds were hand-planted directly above the drip line. Soil volumetric water content (SVWC) was monitored weekly using a field-calibrated neutron moisture probe (Model 503, Campbell Pacific Nuclear, CPN, Martinez, CA, USA), with measurements taken at 0.2 m increments from 0.1 m to 1.9 m depth. The collected data from across the soil profile were used to adjust the timing and duration of applied irrigation to achieve approximately 25% and 14% SVWC in the WW and WL treatments, respectively.

2) SCANNING

The University of Arizona Field Scanner (Figure 3) is a semi-autonomous gantry crane outfitted with a variety of cameras and sensors. A sensor of interest is the Fraunhofer EZRT/IIS custom structured-light 3D line scanners (Fürth, Germany). These scanners utilize the light section method to generate high-detail point clouds.

The data analyzed are from a scan conducted on 2020-07-30. On this date, the mean canopy height was approximately 0.4 m. The system operated with the laser

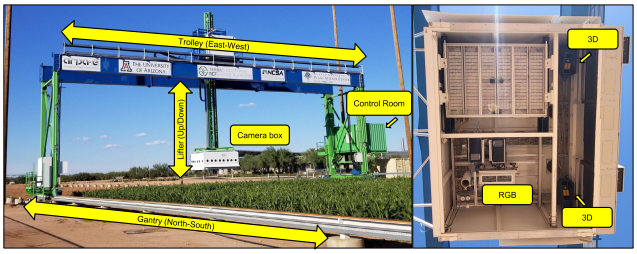

FIGURE 3. **The gantry system located in the University of Arizona provides the data for *VR BioTalk*. It moves on three axes: north-south, east-west, and up-down. The camera box (right) supports various scanning, and *VR BioTalk* uses RGB and LiDAR data.**

scanners at a distance of 3.5 m from the mode canopy height. The system moved from south to north in 0.7 m increments over a distance of 151.659 m. Starting with an east-to-west motion, the system captured measurements using the line scanners along the east-west axis for approximately 22.135 meters. At the end of the motion, the system moved north to the next position and captured the next measurement with the line scanners in the opposite direction. By repeating this process, the system captured the entire region of interest across 217 measurements.

Each measurement provides a pair of grayscale Portable Network Graphics (PNG) formatted images from both scanners (lossless LWZ compression), with one representing depth and the other representing reflectance. A JavaScript Object Notation (JSON) formatted metadata file that includes relevant fixed and variable metadata was also included. These measurements were transferred to an on-site cache server. Software running on the Field Scanner (Fraunhofer EZRT/IIS PlyWorker) downloaded these measurements from the cache server to a temporary folder, generated a point cloud in Polygon File Format (PLY) file for each scanner's data, and then uploaded the results to the cache server. Upon completion, the cache server uploads the entire session's data to CyVerse [49] for storage.

B. DATA PROCESSING AND TRAIT EXTRACTION

The raw data gathered from the gantry is processed for VR visualization and voice visual analytics. In particular,

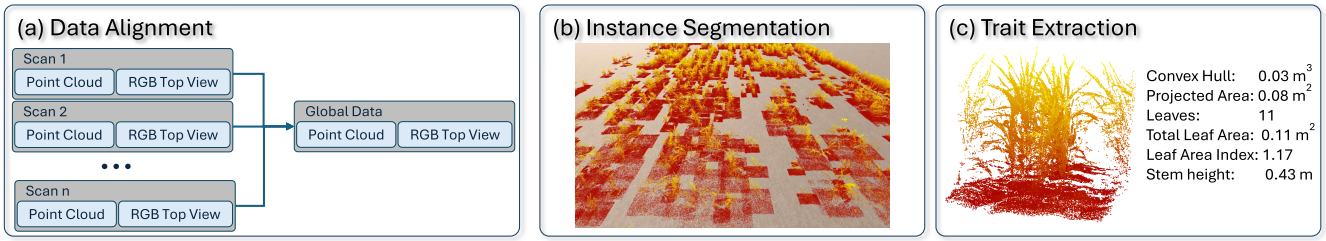

FIGURE 4. The plant traits are extracted from the raw data in three steps: (a) the individual field scans are combined into one, (b) the ground is removed, and individual plant instances are detected, and (c) traits for each plant are generated.

we extract data suitable for visualization and extract additional plant-related traits. Figure 4 shows the overview of the process, which we formalize as follows. Let $S = \{s_1, s_2, \ldots, s_n\}$ be the set of n raw scans, where each scan $s_i = (P_i, I_i, M_i)$ consists of a point cloud $P_i \subset \mathbb{R}^3$, an RGB image I_i, and metadata M_i. The preprocessing pipeline $\mathcal{F} : S \to (T, \hat{P})$ produces a trait table T and a visualization-ready point cloud $\hat{P}$ through three stages:

Stage 1 – Data Alignment. Scans are registered into a global point cloud P_g and a georeferenced orthomosaic I_g:

$$(P_g,\ I_g) = f_{\text{align}}(S). \tag{1}$$

Stage 2 – Instance Segmentation. Ground and non-focal plant points are removed, and individual plants are identified:

$$\{p_1, \ldots, p_k\} = f_{\text{seg}}(P_g,\ I_g), \tag{2}$$

where each plant $p_j = (p_j^{\text{stem}},\ p_j^{\text{leaf}})$ is further partitioned into stem and leaf point subsets.

Stage 3 – Trait Extraction. For each plant p_j, a trait vector is computed:

$$\begin{aligned}\mathbf{t}_j &= f_{\text{trait}}(p_j) \\ &= \big(h,\ r,\ l,\ cvh,\ a_\perp,\ n,\ la,\ lai,\ s_d,\ s_h\big)_j \in \mathbb{R}^{10},\end{aligned} \tag{3}$$

with symbols defined in Table 1.

1) DATA ALIGNMENT

In each scan, the gantry generates point clouds and RGB images, which must be combined into a single dataset. The data alignment is performed in three steps: (1) by generating geocorrected RGB outputs for individual plant detections, (2) by aligning multiple component point clouds, and (3) by registering and georeferencing point clouds.

Georeferenced RGB outputs with individual plant detections were generated using PhytoOracle [17]. A series of modules was used to convert raw binary data to TIFF images, apply geocorrections and image stitching, run detection models, and generate outputs that include a downsampled, full-field, georeferenced orthomosaic image and a set of uniquely identified, individual plant detections across multiple scans for the season. These outputs from RGB image processing serve as inputs to the 3D scanner data processing.

An initial alignment of the raw component point clouds was performed manually using a GUI on a subset of the data. The resulting transformation is applied to all measurements collected during the scan. Next, another GUI is used to select common landmarks present in both the downsampled RGB orthomosaic and the aligned 3D data. The resulting transformation (including rotation, scaling, and translation) is applied to all passes, correcting the GPS locations of the point clouds. RGB detections are then used to crop these geocorrected point clouds to the bounding boxes for the individual plants derived from the RGB imagery and transfer the unique identification. Subsequent registration of these clipped point clouds results in a single PLY-format point cloud for each identified plant. This point cloud contains points from the focal plant identified in RGB, points from non-focal plants that may be intruding on the bounding box, and soil points.

2) INSTANCE SEGMENTATION

Semantic segmentation was performed to remove soil and non-focal plant points from the individual plant point clouds by using the PlantSegNet [16] package. The resulting point cloud, containing only points identified as belonging to the focal plant, is used as an input for further instance segmentation using the same algorithm [16]. In this stage, segmentation categorizes points as belonging to the stem or individual leaves.

3) TRAIT EXTRACTION

An optimal way to extract meaningful traits would be to fully 3D-reconstruct the field and then extract them from the geometric models of each plant. However, full-point cloud data reconstruction remains an open research problem [29], [46], [47], [48]. Instead, we estimate certain traits directly from the point clouds (see Table 1 for the complete list). The advantage of this approach is that it can be applied to virtually any point-cloud phenotypic data and is not limited to the gantry.

The previous step segmented instances of each plant, and each plant's point belongs to one of two categories: stem or leaves. All points are used to estimate the height, radius, convex hull, and the projected area. Leaf points are used to estimate leaf area, leaf area index, and the number of leaves. The stem points are used to calculate the stem diameter and the stem height. Once all traits for each plant were calculated, we also extracted global measures for the

TABLE 1. Extracted plant trait name, symbol, unit of measurement, and description.

Name	Symbol	Unit	Description
Height	h	m	Plant height
Radius	r	m	Radius of the projected circle
Length	l	m	Plant length along the y-axis
Convex hull	cvh	m^3	Maximum convex volume occupied by the plant
Projected Area	$a_\perp$	m^2	Area of the plant projected to the ground
Number of Leaves	n	-	Number of individual leaves
Leaf Area	la	m^2	Area of all leaves
Leaf Area Index	lai	-	The amount of leaf area per unit of ground area
Stem Diameter	s_d	m	Average diameter of the stem
Stem Height	s_h	m	Stem height

entire set, such as the average, mean, and standard deviation per plot.

Height (h [m]) is the distance between the average 50 tallest and lowest points in the cloud.

Radius (r [m]) represents the radius of the projected circle that encompasses the plant. We find the plant's central axis and set r to the maximum horizontal distance from the center to the 50 points farthest from the center.

Length (l [m]) is the maximum vertical distance between the 50 points with the smallest and largest value on the y-axis.

Convex hull (cvh [m^3]) is the volume of the smallest polyhedron that contains all the points. We used the SciPy library [32].

Projected area ($a_\perp$ [m^2]) is the area of the polygon that encompasses all the points projected on the ground. We project all points onto the ground, then compute the 2D Convex Hull of the projected points and use its area.

Number of leaves (n [-]) is calculated as the number of connected components in the segmented foliage point cloud.

Leaf area (la [m^2]) is estimated by performing Poisson Surface Reconstruction [12] on each connected component in the foliage cloud and summing the area of the generated meshes.

Leaf area index (lai [-]) is calculated by dividing the total leaf area by the projected plant area:

$$lai = \frac{la}{a_\perp}.$$

Stem diameter (s_d [m]) is calculated as the diameter of a cylinder that is fit to the stem point cloud by using the Random Sample Consensus Shape Fitting algorithm (RANSAC) [11].

Stem height (s_h [m]) is the maximum vertical distance between the points on the vertical axis.

4) HARDWARE, DATA, AND PROCESSING TIME

Computer Hardware: *VR BioTalk* system was developed and deployed on a PC with an AMD Ryzen 7 9800 × 3D CPU, 32 GB of RAM, and an NVIDIA RTX 5090 GPU connected in a Virtual Local Area Network via Wi-Fi 6E with a Meta Quest 3 [39] VR headset.

We extracted the traits from the plant point clouds using Python 3.10, CloudCompare [13], and Open3D [37]. This process took approximately 10 hours for the 2020-07-30 dataset.

Images: Outputs from the 2020-07-30 RGB scan consist of 123 passes of 45 acquisitions each, totaling 5,535 acquisitions with each containing JSON-formatted metadata and BIN-formatted images from each of two cameras (left and right). Raw RGB data from this scan totals 59 GB in compressed format and 85 GB in uncompressed format. The full-scale orthomosaic image was clipped to agricultural plots and saved, totaling 23 GB. Individual plant detections were clustered across the season and saved to a CSV file, totaling 231 MB. Bounding boxes from clustered detections were used to clip plant instances from 3D data.

Point Clouds: After processing the scan with PlantSegNet [16], we end up with 10,484 focal plants, which are divided into two datasets: the aligned plot used for visualization and the segmented individual plants. The aligned plot contains, on average, 1,871,313 points and is 50.52 MB in size. The segmented plants contain, on average, 56,000 points per plant and are 1.52 MB in size. The total processed scan is 519 GB.

During trait extraction, we discarded any empty point clouds, and from the remaining set, we subsampled each segmented plant point cloud to 5,000 points to facilitate rendering in the VR headset, and we aggregated them by their plot. This reduces the 10,484 focal plants into 1,595 point cloud files, one per plot, with an average file size of 1.59 MB, totaling 1.68 GB for the entire field. Each trait file is 476 bytes, totaling 1.82 MB across all 1,595 files.

Computational Complexity: The preprocessing pipeline analyzes each plant independently. Let m denote the number of points in a single plant cloud, n_l the number of leaves,

m_l the average points per leaf, m_s the number of stem points, and k the total number of plants in the field.

The cheapest operations (soil removal, height, width, and length estimation) scan the point cloud once and are therefore $\mathcal{O}(m)$. Computing the convex hull volume and the projected area both rely on the Quickhull [32] and cost $\mathcal{O}(m \log m)$ each. Leaf instance segmentation via connected component extraction is $O(m)$, but the subsequent Poisson surface reconstruction [12] required for leaf area estimation costs $\mathcal{O}(m_l \log m_l)$ and is repeated for each of the n_l leaves. Stem diameter estimation uses RANSAC cylinder fitting [11], which runs in $\mathcal{O}(m_s \cdot t)$ for t iterations. The overall per-plant complexity is thus:

$$T_{\text{plant}} = \mathcal{O}\big(m \log m + n_l\, m_l \log m_l + m_s\, t\big), \tag{4}$$

dominated by the convex hull and Poisson reconstruction steps, and the full-field extraction runs in $\mathcal{O}(k \cdot T_{\text{plant}})$. Because each plant is processed and then discarded, the space requirement is $\mathcal{O}(m)$ at any given time. Only the ten-element trait vector $\mathbf{t}_j \in \mathbb{R}^{10}$ and a subsampled cloud of N_{vis} points are retained per plant.

C. HANDS-FREE VISUAL DATA ANALYTICS

Figure 5 shows an overview of *VR BioTalk* execution pipeline. At the core of the system is the Visualization Engine, which interprets the user's commands and visualizes the scene. Previously consolidated plant data from point clouds (right) is used for visualization, and plant traits are used for data analytics. The VR headset's position and orientation are tracked and relayed to the view control, and the voice commands are converted into commands.

1) VIEW COMPLEXITY MANAGEMENT

The inputs to the View Complexity Management are the camera position and orientation, and the point cloud. The output is a dataset that can be rendered at interactive rates on the VR headset.

Recent advances in VR technology have enabled all-in-one VR headsets with onboard tracking, networking, and graphics, allowing for untethered exploration of virtual environments. However, the rendering capabilities of the GPUs in such VR headsets are severely limited compared to those of desktop workstations. Scientific datasets, like the point cloud phenotyping dataset we aim to visualize, far exceed the rendering capabilities of all-in-one headset GPUs. Ignoring the rendering capacity of the VR headset leads to low frame rates, which are known to induce cybersickness [51]. Whereas desktop visualizations can support lower frame rates and frame-rate fluctuations, immersive visualizations must consistently update the user's frame every 10-15 ms [52]. VR headsets have native frame rates (e.g., 72 fps for Meta's Quest 3, corresponding to 14 ms frame times), which the application should aim to enforce.

Managing vegetation point cloud complexity is complicated by several factors. Effective view frustum culling relies on good view-direction coherence, which allows the cost of computing and loading the data encompassed by the user's view frustum to be amortized over hundreds of output frames. VR attempts to convey to domain experts the sense of being present in the field, allowing them to examine it freely by moving their head, with large-amplitude, fast, and unpredictable view rotations. Compared to desktop visualization, the view changes are more erratic, and delaying the frame update for view frustum culling to catch up with the user's sudden desire to see behind them is unacceptable. A significant success of VR technology is its ability to provide a large field of view, e.g., 110°× 96° for the Quest 3. Padding this field of view by another 45° in each direction to improve robustness to view prediction errors further limits any potential benefit of view frustum culling.

Occlusion culling is also challenging in large point cloud datasets of leafy fields. In an urban or indoor virtual environment, occlusion culling can safely discard most of the original dataset by leveraging large blockers such as nearby building facades or the walls of the room where the user is currently located. In our dataset, fragments of distant plants are visible to the user due to the imperfect occlusion of nearby plants, and discarding them results in approximate occlusion culling that fails to capture the true field density.

Although point clouds provide an explicit representation, opening the door to hierarchical Level of Detail (LOD) adaptation via classical space-partitioning schemes such as octrees, simplifying phenotyping point clouds remains challenging due to the complex geometry of individual plants and the large number of plants. Whereas replacing the eight children of an internal octree node with a single point helps reduce the complexity of a large leaf, repeatedly applying this approach fails to preserve the structure of a plant, a group of plants, or a field patch.

We have developed a robust approach to complexity management specifically designed for phenotyping data (Fig. 6). We first partition the dataset into near and far regions (a) with respect to the user's position. We model the near region with a circular patch of ground of radius R and center O. The far region is all the data beyond R. The data in the far region is pre-rendered from O to a cubemap (b). A cubemap is a collection of six images with the same eye O and with the image frames forming a unit cube, capturing the 3D dataset's appearance in any direction from O. The cubemap is used as a backdrop, abstracting the distance between the current user position and O (c). The data in the near region is rendered at full resolution. The near/far partitioning offers a powerful approach to complexity management. The cubemap captures distant plants with high fidelity (d), yet rendering from it is inexpensive, requiring only six squares, each textured with a cubemap face. The approximation introduced by the cubemap is that there is no motion parallax between distant and very distant plants, as the user translates. The radius R is chosen to be as large as possible while still meeting the headset's rendering budget.

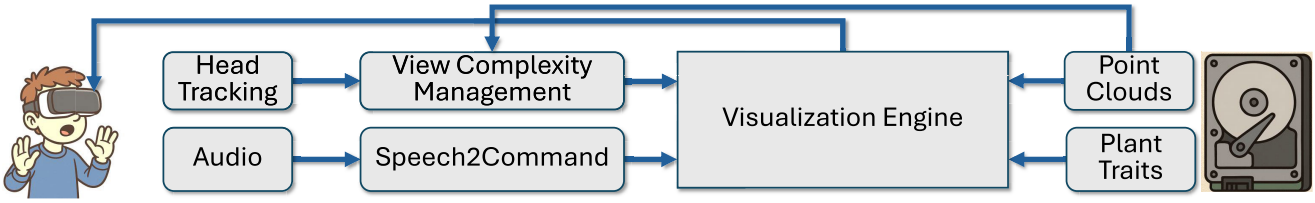

FIGURE 5. *VR BioTalk* **interaction and visualization. The VR headset's head tracking is used to select the corresponding view. At the same time, the voice is converted to commands that are interpreted by the visualization engine, which operates on the trait data and point clouds.**

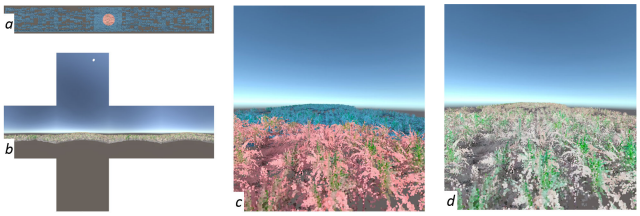

FIGURE 6. *VR BioTalk* **rendering complexity management. The point clouds are partitioned into a near (red disk in *a*) and a far region (blue in *a*). The data in the far region is prerendered to a cubemap (*b*). Output frames are generated by rendering the near region from geometry (red highlight in *c*) and the far region from the cubemap (blue highlight in *c*). The output frame *d* shows the entire field.**

Furthermore, the circular near region and the cubemap cover all user-view directions, bypassing the need for view prediction and thus avoiding artifacts from view-prediction errors.

A given near/far partition of the dataset is useful while the user is inside the near region. As the user approaches the boundaries of the near region, a new near/far partition of the dataset is needed, centered at the current user position. One of the great strengths of VR is the natural view selection enabled by head translations and physical locomotion. The user walks in the physical world, while the headset provides the images the user would see if they actually walked in the field scanned by the dataset. It is often the case that the physical space available to the user of a VR application is much smaller than the virtual environment. For example, a scientist might use a 5 m $\times$ 5 m area of open floor space in their laboratory to examine a virtual dataset that was obtained by scanning a 100 m $\times$ 10 m field. The scientist cannot go beyond the physical floor space, even though they have not reached the end of the virtual world. To address the mismatch between the physical and virtual world sizes, virtual reality applications allow the user to *teleport*, i.e., to suspend the one-to-one mapping between the virtual and physical worlds, thereby repositioning themselves in the virtual world without a commensurate translation in the physical world. The user is repositioned abruptly rather than gradually, as *flying* induces cybersickness due to visual acceleration that is not corroborated by the physical acceleration reported by the inner ear. Teleportation is well-suited to our near/far complexity management approach, with every teleportation triggering the computation of a new near-far partitioning. The near-far partitioning is dominated by rendering a 2,048 $\times$ 2,048 cubemap and takes, on average, 170 ms in our dataset.

Additionally, we maintain a Graphical User Interface (GUI) that remains fixed to the user's view, allowing them to see their transcribed audio and the executed commands.

2) Speech2command

The audio recorded by the VR set is converted into Visual Engine commands in two steps. First, the audio is converted to text (Speech2Text), and then the text is converted to a command (Text2Command). We aimed at two main goals: a wide range of accents and precise commands.

Commands: The Visualization Engine has a set of pre-defined commands with domain-limited parameters defined by the extracted traits (e.g., `get_n_plots(n, trait, order)` selects the top `n` plots in the given order based on the given trait) that were determined based on the most commonly used operations. The list can be easily expanded by other operations if needed (see the complete list in Table 2). We took these basic commands and generated an exhaustive list of variants using ChatGPT [33] with the prompt "generate a list of variants of these commands", resulting in a final list of 160 potential user queries. We then used the list of queries to generate the corresponding voice queries. We fed each query to Amazon Polly [7] and asked it to generate audio in 13 different accents, resulting in 2,080 audio samples. The output of this process is a paired list of possible commands and their combinations, along with the corresponding voice commands. We split this dataset into training and test sets in an 80:20 ratio. While the fine-tuning data consists of synthetic speech, our user study Section II-D provides implicit validation on real speech, since all nine participants issued commands in their own voices and accents and reported good usability.

We leveraged Amazon Polly [7] to generate the synthetic audio samples that were used to fine-tune Whisper [6] with Parameter-Efficient Fine-Tuning (PEFT) [9] for five epochs, using a learning rate of 10^{-4} and a batch size of eight, taking approximately 18 minutes on the previously described PC.

Speech2Text: We used the paired Visualization Engine and the voice commands to fine-tune a Speech-to-Text (STT) model to increase its precision in detecting the *VR BioTalk* commands. The STT is Whisper-large-v3-turbo [6], and we used Parameter-Efficient Fine-Tuning (PEFT) [9] to bias the

TABLE 2. List of available commands in *VR BioTalk*.

Name	Description	Parameters
get_plot_trait	Get the trait value of the selected plots or an specified one.	Plot_ID: ID of the plot to be queried. Trait: Trait to be queried.
get_n_plots	Select the top N plots based on a trait in a given order.	N: Number of plots to select. Trait: Trait to be queried. Order: Order to sort the plots by.
move_to_group	Move the selected plots to a given group.	Group: Group to move the selected plants to.
clear_group	Clear the plots in the specified group.	Group: Group to clear.
get_plots_with_condition	Select the plots that satisfy a specific condition based on a trait.	Condition: Condition to be met. Value: Value to compare against. Trait: Trait to be checked.
show_distribution	Show the distribution of the trait values of a specific group.	Group: Group to be shown. Trait: Trait to show distribution of.
show_plots	Visualize only the plots in the specified group.	Group: Group to be shown.
get_percentile_plots	Select the plots that are in the top N-th percentile based on a trait in the given order.	Percentile: N-th percentile to be obtained. Trait: Trait to query. Order: Order to sort the plots by.
compare_groups	Compare the specified trait values of two groups.	Group1: First group to be compared. Group2: Second group to be compared. Trait: Trait to compare groups on.
clarification	Ask a question to the user to clarify a previous command.	Question: What the system wants the user to clarify.

STT towards our commands and accents. The output of this step was a fine-tuned STT that can convert voice in various accents into textual commands. Without the bias, Whisper-large-v3-turbo transcribes an audio of the command "Add plants taller than 10 cm to group four" to "Add plants taller than 10 cm to grouped fall.", while the biased model transcribes it correctly.

Text2Command: The text generated by the STT is fed to an LLM, which converts it to commands. We used Ollama [8] to run the Qwen2.5 [10] LLM locally, specifically the 14B parameter model. We define each command as a tool that the model can call, and Table 3 shows the available commands and their parameters.

We used a few-shot prompting [14] to provide additional context to our model and improve its robustness. The few-shot prompting involves providing the LLM with a series of example interactions and their corresponding expected outputs. We provided the model with two example interactions: first, we presented it with the user's query, and then we described the expected command calls, including the expected values for each argument. This improved performance by 4% compared to no-shot prompting. We measured this by presenting our LLM with various users' queries and comparing the desired output with the LLM's output.

Our command template is formalized as such: each command is a tuple $c = (f, \theta)$, where f is a function from a defined set C and θ is a parameter vector that covers domain-specific enumerations (traits, groups, conditions). The LLM maps natural language to this structure.

Our set of commands could be easily extended by adding a new command that would involve three steps: (a) implement the function, (b) add it to the tool dictionary, and (c) optionally add training samples to the Speech2Text fine-tuning. Since the fine-tune biases transcription toward domain vocabulary, while the LLM interprets the semantic intent, it can handle unseen queries and map them to new tools.

3) VISUALIZATION ENGINE

The visualization engine is implemented in Unity 3D [38] version 2022.3.30f1 and runs on the Quest 3. It executes the commands generated by the Speech2Command unit. The engine maintains a current dataset for visual display, which is updated when needed; e.g., when a selection is requested. If an update is needed, it first fetches the trait and point cloud data from the database (see Figure 4 on the right), and the View Control then generates the corresponding data for the VR headset to visualize.

In addition to data selection and communication with the View Control, the Visualization Engine has several other functions (see Figure 7):

1) **Chart Generation** is used when the user asks for data comparison (e.g., "display the distribution of leaf area indices"). We use 3D charts [15] to visualize the data, which show each plot's ID and trait value.

2) **Trait Information:** The Visualization Engine can show the value of a given trait for a specific plot or a subset of plots. The trait is shown next to each plant.
3) **Selection and conditions:** One of the most powerful functions of *VR BioTalk* is the selection. The user can identify a subset and process it (e.g., "select the top 10% of the highest plants"). The selection is stored as the current dataset, and the Visualization Engine then operates on it. The selection can be modified, e.g., "add the five smallest plants to the selection"). The selection is closely coupled with conditioning. The system supports multiple selection groups.
4) **Group Comparison:** Using the system's group selection, the engine supports the comparison of different subsets of plots on a given trait.

FIGURE 7. ***VR BioTalk*** **main functions used in the visual analytics of phenotypic data.**

D. USER STUDY

We validated *VR BioTalk*'s usability, impact on users' cognitive load, and presence by conducting an IRB-approved user study.

1) STUDY FOCUS

The study involves participants interacting with *VR BioTalk* while completing four tasks. Each task focuses on interacting with a set of traits and trait groups (note that the plants are assumed to be at the same growth stage). Each task is formulated as a question, so it is not a prescription on how to interact with the system. Users were free to use any interaction with *VR BioTalk* to answer the question. The four tasks were:

T1. What is the relationship between the plant's height and its stem diameter?
T2. What is the relationship between the plant height and the number of leaves?
T3. What is the relationship between a plant's projected area and its height?
T4. Leaf area index (LAI) is the ratio of the leaf area in a plant canopy relative to the ground area beneath it. What is the relationship between LAI and the number of leaves?

2) PARTICIPANTS

We have recruited $N = 9$ participants from our university following the guidelines from Virzi [60], who concluded in their study that the minimum number of participants for usability studies is 4-5. He stated "The basic findings are that (a) 80% of the usability problems are detected with four or five subjects, (b) additional subjects are less and less likely to reveal new information, and (c) the most severe usability problems are likely to have been detected in the first few subjects."

All demographics described here were part of our pre-questionnaire. The study included four participants with a background in agronomy, two in environmental sciences, two in computer science, and one in plant biology. Four participants are doctoral students, four are master's students, and one is an undergraduate student. Participants also indicated their years of experience in their respective fields, ranging from two to 20 years. They were asked to choose how frequently they experienced immersive visualizations (e.g., VR), responding (never, rarely, occasionally, and frequently). The final set of demographics questions asked about experience working with biological data and included 12 five-point Likert-scale items (see Figure 9).

FIGURE 8. ***VR BioTalk*** **user study procedures.**

3) PROCEDURE

The expected time for each user study session was 1.5 hours, including setup, questionnaire completion, and system interaction. The four primary stages are as follows: (1) *setup:* brief introduction, pre-screening, and consent, (2) *preparation:* pre-survey and video tutorial, (3) *process:* system usage, and (4) *conclusion:* post-surveys. A $30 gift card was offered to each participant who completed the entire session. Although the participants could leave at any time, all completed the study.

4) DATA COLLECTION

Every user completed a questionnaire measuring system usability, cognitive load, and presence after interacting with the system. The System Usability Scale (SUS) is a popular measure of system usability across various contexts [57]. SUS consists of ten questions, all on the same five-point Likert scale, from strongly disagree to strongly agree [57]. The Igroup Presence Questionnaire (IPQ) is a test quantifying the presence in virtual space [58]. IPQ has a three-factor structure, involving spatial presence, realness, and involvement. There are 14 items in total, with response ranges from -3 to 3 for each item [58]. The cognitive load scale measures germane, intrinsic, and extraneous load [59]. The original example was in statistics [59], so we slightly adapted some of the items to make it relevant to our field. The cognitive load scale has 10 items, with responses ranging from 0 to 10 [59].

III. RESULTS

We first validate *VR BioTalk*'s technical capabilities, then demonstrate its visual analytics capabilities by showcasing several phenomics case studies.

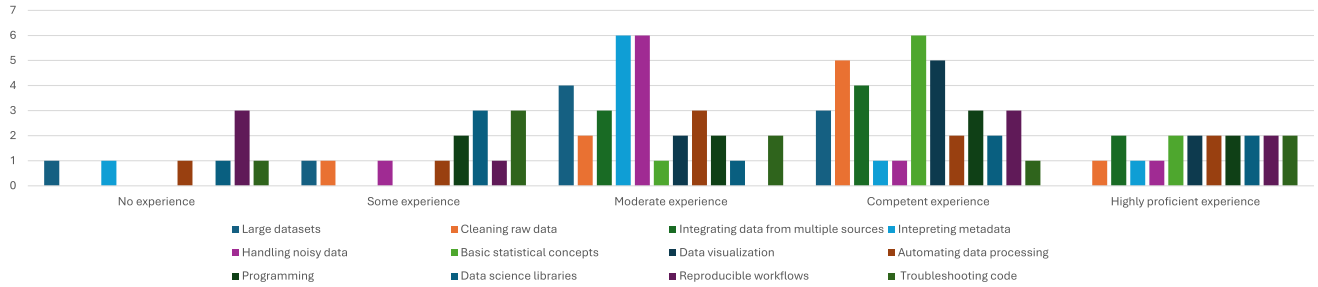

FIGURE 9. Data management experience survey results.

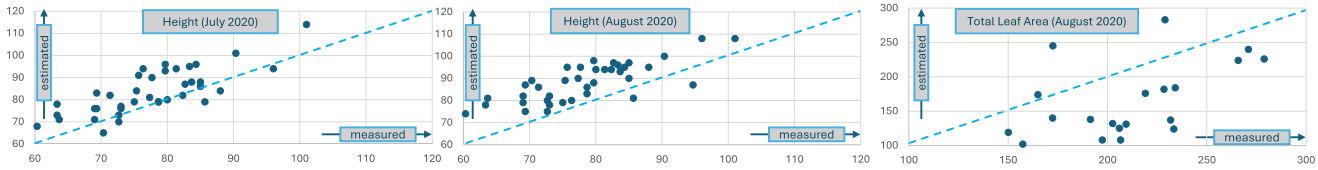

FIGURE 10. Estimated and measured plant height from July 2020 (left) and August (middle) in cm and total leaf area (right) in cm^2.

A. TRAIT EXTRACTION

We measured the effectiveness of our Trait Extraction module by comparing the average measured plant height and total leaf area for various samples with the estimated values. This was done on Sorghum data for the July and August 2020 season. Plant height was measured with a meter stick to the nearest centimeter, from the soil surface to the plant apex. For leaf area measurements, a CID Bio-Science CI-203 handheld laser leaf area meter was used. This instrument computed total leaf area using a continuous scanning procedure to provide a cumulative leaf area, in cm^2, summed across the length of the leaf.

We obtained a correlation value of 0.74 and a Root Mean Squared Error of 9.7 cm across 39 plots of the field for the July data shown in Figure 10 and a correlation value of 0.77 and a root mean square error of 11.7 cm in 38 field plots for the August data. The correlation between the measured and ground truth total leaf area was 0.42, and a Root Mean Squared Error of 99.29 cm^2. The lower leaf area correlation is attributable to Poisson reconstruction and connected component segmentation failures on small leaves, as well as error accumulating over each processing step.

B. Speech2Command

Speech2Text. We evaluated the precision of the Speech2Text module on the test set, which included 416 different spoken commands and 13 English accents. We measured its Word Error Rate (WER), which is a common metric for speech recognition systems [50]. We also measured the average inference time per command, which is the time the model takes to convert the uttered audio to text. Table 3 shows that the robust Speech2Text model performs better than the base Whisper model, without a significant trade-off in inference time.

TABLE 3. Different speech-to-text models' accuracy and average inference time.

Model	Whisper-tiny	Whisper-large-v3-turbo	Finetuned STT (Ours)
Word Error Rate (WER)	0.3022	0.2351	0.0032
Avg. Inference time per sample (s)	0.0635	0.2506	0.3407

Text2Command The performance of the command generation from text was tested by using several LLMs: Qwen 2.5 [10], Llama 3.1, Llama 3.2 [35], and DeepSeekR1 [36]. We created a dataset of 50 user compound commands that require at least two commands to be executed in a specific order. We measured accuracy by comparing the output commands against the user's intent. If the executed commands achieved the user's intent, they were marked as correct. Moreover, we measured the average inference time of the command generation from text. Table 4 shows the results. We observe that the medium models (14B, 32B) tend to perform better than smaller models, at the cost of additional inference time. We chose the Qwen2.5:14B model because it has the best performance.

TABLE 4. Different large language models' accuracy and average inference time.

Model	Qwen2.5			Llama 3.1	Llama 3.2	DeepSeekR1	
Model Size	7B	14B	32B	8B	3B	8B	14B
Accuracy (%)	76%	94%	92%	54%	52%	54%	84%
Avg. Inference time per sample (s)	0.29	0.89	1.37	0.62	0.24	0.17	0.70

Our evaluation shows that Speech2Command tends to struggle with compound commands that require more than two steps, as it tends to execute redundant commands or hallucinate commands when running it with a smaller LLM.

(e.g., ''Select the 100 tallest plants, add them to group two, and plot the leaf area distribution of group two''.)

C. VISUALIZATION

We achieved an average of 63 FPS in the visualization engine, with a minimum captured value of 23 FPS during peak processing, which occurred when visualizing and processing the whole field. The LOD rendering enables us to reduce the total number of rendered points from 11M to 2M on average, allowing us to run the client application on the headset without requiring an external PC connection or advanced graphics card.

D. VR BioTalk OVERALL PERFORMANCE

The overall time from verbally issuing the command to displaying the result is 1.25s on average for *VR BioTalk*. The minimum time was 0.63s and the maximum 1.78s. Speech2Text took, on average, 34.83% of the total time, Text2Command took 27.41%, and the visualization engine took 37.76%.

E. USER STUDY

System usability The SUS score [57] ranges from 0 to 100, with higher scores indicating greater usability as perceived by users. Our participants had an average SUS score of 71.11 with a standard deviation of 16.68, indicating a high level of usability. The higher usability value quartile was 81.25, while the lower was 66.25.

The **Igroup Presence Questionnaire** (IPQ) [58] overall scale ranges from −3 to +3 and includes several items. The eighth item measures the general presence, and *VR BioTalk* had an average of 0.67 with a standard deviation of 1.87. Items 3, 6, 9, 10, and 13 measured the spatial presence. The overall mean for spatial presence was −0.07, with a standard deviation of 1.49, indicating a neutral level of spatial presence. Items 1, 7, 11, and 14 measured the involvement. The mean for involvement was 0.64, with a standard deviation of 1.34, indicating good user involvement. Items 2, 4, 5, and 12 concerned experienced realism. The mean for experienced realism was −0.61, with a standard deviation of 0.77, indicating poor experienced realism. Figure 11 shows the box plot of all presence items.

The **cognitive load** [59] measures the mental effort while performing a task on an overall scale from 0 to 10. Items 1, 2, and 3 measured intrinsic load. Overall, the intrinsic load was 6.63 with a standard deviation of 2.50, indicating the system was neither too complex nor trivial. Items 4, 5, and 6 measure extraneous load. The extraneous load was 8.85, with a standard deviation of 2.04, indicating the system was easy to use (values for negatively worded questions were corrected so that a higher value indicates a better result). Items 7, 8, 9, and 10 measured germane load. The germane load was 5.92, with a standard deviation of 2.76, indicating that the system was inconsistent in its effectiveness at learning. Figure 11 displays the box plot of all cognitive load items, with inverted values for the negative items, so higher values mean better results.

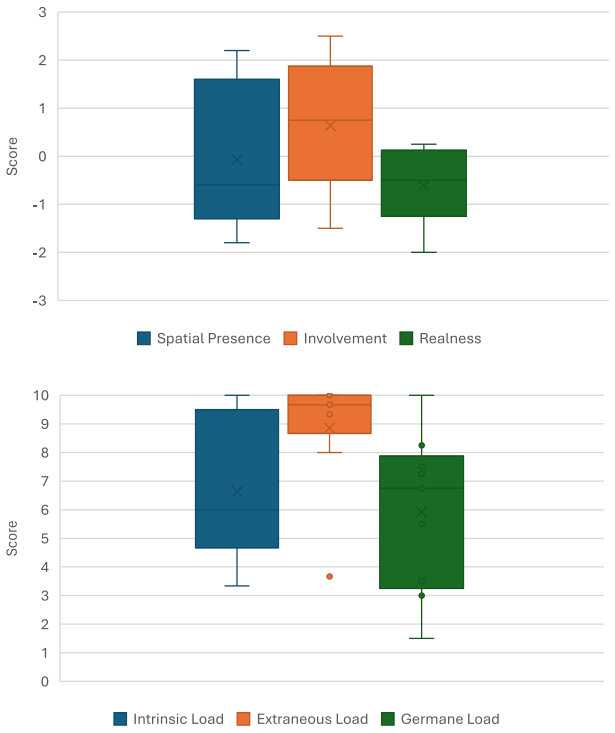

FIGURE 11. Igroup Presence Questionnaire (IPQ) (up) results divided by Spatial Presence, Involvement, and Realness subscales. Cognitive Load Survey Results (down), showcased as three subscales measuring Intrinsic, Extraneous, and Germane loads.

We also measured the average translation and rotation of the camera and the left and right hands across all tasks, as shown in Table 5. The average task completion times were 370.0 s (SD = 109.6) for T1, 228.8 s (SD = 133.7) for T2, 339.8 s (SD = 213.9) for T3, and 458.4 s (SD = 390.2) for T4. Translation and rotation magnitudes were comparable across the camera, left hand, and right hand, indicating that hand motion was predominantly passive, driven by whole-body movement during locomotion and teleportation, rather than by deliberate manual interaction.

IV. DISCUSSION

We have demonstrated that it is possible to develop a visual analytics system that accepts raw point clouds as input and enables fast (confirmed by testing), user-friendly, and meaningful visual analytics (confirmed by the user study) without requiring programming and complex interaction. *VR BioTalk* provides a fast user experience by managing complex 3D scenes and providing verbal feedback in about 1.25 seconds. We have also provided detailed validation and evaluation of each system block.

It is essential to note that many of the system's building blocks rely on current AI technology. In particular, the Speech2Text and Text2Command blocks utilize LLMs,

TABLE 5. Measured average user translation and rotation while using *VR BioTalk*.

Camera Translation (m)		Camera Rotation (deg)		Left Hand Translation (m)		Left Hand Rotation (deg)		Right Hand Translation (m)		Right Hand Rotation (deg)	
Mean	StdDev	Mean	StdDev	Mean	StdDev	Mean	StdDev	Mean	StdDev	Mean	StdDev
275.67	241.84	9,162.74	6,582.06	291.40	245.52	10,489.73	8,405.42	295.03	245.48	11,108.61	9,782.17

which are the fastest-growing area of AI. We anticipate that by the time this paper is published, the area will have already progressed in providing more precise and faster models. This does not invalidate this work, as the overall feedback and precision are likely to improve with advances in AI.

We have also demonstrated the system's functionality by presenting several test cases in a user study. The survey results highlighted *VR BioTalk*'s usability and high engagement, and it is easy to use. However, it could be improved in terms of realism and cognitive depth. Other highlights from users included the visualization of large datasets, *VR BioTalk*'s ability to ''understand'' them, and the ability to quickly compute and visualize trait value distributions.

To the best of our knowledge, there are no other end-to-end interactive, voice-controlled immersive phenotyping analytics systems for direct comparison. Therefore, a comparison would need to be made between fundamentally different workflows (e.g., scripting or manual field measurement), thereby testing the modality rather than the system itself. If a future system were developed, we suggest measuring task completion and preprocessing times, as well as command error rates and user locomotion metrics.

A. LIMITATIONS

The most significant limitation of *VR BioTalk* is its dependence on the trait extraction and the imprecision of the 3D scanner point clouds. Existing algorithms constrain the amount of information we can extract from plants because the available geometric data is limited. Plant instance segmentation often adds points from neighboring plants; leaf reconstruction fails on small leaves and sharp edges, which contributes to the reconstruction's low precision. The accuracy of the downstream analytics is bounded by the quality of the input point clouds and the existing algorithms for trait extraction. Sensor noise, occlusion artifacts, and segmentation errors propagate through the trait extraction pipeline, and their cumulative effect is most pronounced for traits that depend on surface reconstruction, such as leaf area and leaf area index. While height estimation is relatively robust to these errors because it depends only on the positions of extreme points, leaf-based traits require faithful reconstruction of thin, often overlapping structures, which remains challenging given current sensor resolution and segmentation algorithms. Quantifying and reducing this error propagation is an important direction for future work.

The trait extraction limits the expressivity of *VR BioTalk*. The system will allow the user to converse only about the extracted traits and will not understand any others. While the list is long and the number of commands is much higher, there may be essential traits that the system cannot process.

Several limitations stem from the current state of the LLMs. Although the model used for the Text2Command module is large, it still requires fine-tuning with our data. Ideally, we would use an off-the-shelf model that has been trained universally. We hope new LLMs will bring this functionality.

Another limitation of the data-capture format is the quality of the visualization for each plant. We visualize sparse point clouds that lack plant topology, with density depending on the sensor sampling rate. We had to increase the number of points to improve the level of detail in each plant, but this becomes rather difficult in a VR headset due to its limited rendering capacity. When scaling to a larger dataset, this becomes our main bottleneck. So, for larger datasets, we could subsample each plant to fewer points. The preprocessing scales linearly, so processing the dataset without parallelization would take longer. Since the runtime commands operate on the trait file, even a larger dataset would have a negligible impact on execution time, given the STT and LLM inference bottleneck.

Additionally, we have a limited set of selection queries that can be performed, as highlighted by some users in our study, who wished the system supported selecting plants within a range. Similarly, the ability to compute other plots, such as a scatter plot, and to compute the mean, median, and standard deviation for the selected plants is a requested feature. Finally, we acknowledge the limitations of our user study in demonstrating an improvement in analytical work.

B. FUTURE WORK

Future work should focus on other plant species and allow the VR application to display larger fields. This implies developing algorithms to manage the complexity of spatiotemporal data. Similarly, expanding the number of traits and available commands is a natural direction, with a focus on improving the system's phenotyping capabilities. An important future work is improving the visual quality of the results, which can be achieved by 3D reconstructing the data into 3D models. Following up on the limitations related to model size, one possible direction is to create a custom LLM that powers Text2Command, fine-tuning a smaller model to provide a lightweight alternative specialized for the task.

Additionally, an interaction paradigm in which the user can converse with the system would be an interesting direction, especially with a fine-tuned model in the phenomics literature, making it a specialized agent. Lastly, we have presented the data for a single instance. The gantry captures successive stages of plant development, and it would be interesting to see how plants develop over time, allowing the user to reason about it through verbal commands.

While we attempted to provide a comprehensive set of features, some participants suggested adding features, such as selecting a range of plants (e.g., from the 10th to the 20th percentile) and adding pairwise visualizations, such as a scatter plot. This is worth pursuing in the future, as it would allow easier comparisons between trait relationships, rather than the current approach users use, which involves forming groups and comparing them based on a single trait.

Additional future work should focus on a minimal set of features with maximum expressivity and minimal time on task. Finally, a larger study should be conducted to demonstrate and quantify the improvement in analytical work when using our system.

V. CONCLUSION

We have introduced, implemented, and tested *VR BioTalk*, a novel immersive visual analytics system that allows non-programming users to interact verbally with large biological datasets, thereby lowering the barrier to exploratory analysis for experts. *VR BioTalk* uses LLMs and speech detection to convert audio to commands that are executed by a visualization engine. Our tests show interactive feedback on contemporary computer hardware using off-the-shelf GPUs and VR systems. The main limitation of *VR BioTalk* is its reliance on preprocessing point cloud data and extracting meaningful traits from it, both of which depend heavily on contemporary information extraction algorithms. Future work should focus on temporal data, large datasets, different species, and the user experience, in particular, investigating optimal ways to convey information to users in VR immersive systems.

REFERENCES

[1] W. Cui, "Visual analytics: A comprehensive overview," *IEEE Access*, vol. 7, pp. 81555–81573, 2019.
[2] J. J. Thomas and K. A. Cook, "A visual analytics agenda," *IEEE Comput. Graph. Appl.*, vol. 26, no. 1, pp. 10–13, Jan. 2006.
[3] A. Fonnet and Y. Prie, "Survey of immersive analytics," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 3, pp. 2101–2122, Mar. 2021.
[4] C. K. Tuggle et al., "Current challenges and future of agricultural genomes to phenomes in the USA," *Genome Biol.*, vol. 25, p. 8, Jan. 2024.
[5] V. Marx, "The big challenges of big data," *Nature*, vol. 498, no. 7453, pp. 255–260, 2013.
[6] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 28492–28518.
[7] Amazon. *Amazon AWS Polly*. Accessed: Mar. 4, 2025. [Online]. Available: https://aws.amazon.com/polly/
[8] Ollama. *Ollama*. Accessed: Mar. 4, 2025. [Online]. Available: https://ollama.com/
[9] S. Mangrulkar, S. Gugger, L. Debut, Y. Belkada, S. Paul, and B. Bossan. (2022). *PEFT: State-of-the-Art Parameter-Efficient Fine-Tuning methods*. [Online]. Available:https://github.com/huggingface/peft
[10] A. Yang et al., "Qwen2.5 technical report," 2024, *arXiv:2412.15115*.
[11] R. Schnabel, R. Wahl, and R. Klein, "Efficient RANSAC for point-cloud shape detection," *Comput. Graph. Forum*, vol. 26, no. 2, pp. 214–226, Jun. 2007.
[12] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proc. 4th Eurographics Symp. Geometry Process.*, 2006, pp. 1–10.
[13] D. Girardeau-Montaut. *CloudCompare*. Accessed: Mar. 4, 2025. [Online]. Available: https://www.danielgm.net/cc/
[14] T. B. Brown et al., "Language models are few-shot learners," 2020, *arXiv:2005.14165*.
[15] Vcian. (2020). *Interactive Bar Chart*. Accessed: May 20, 2025. [Online]. Available: https://github.com/vcian/interactive-bar-chart
[16] A. Zarei, B. Li, J. C. Schnable, E. Lyons, D. Pauli, K. Barnard, and B. Benes, "PlantSegNet: 3D point cloud instance segmentation of nearby plant organs with identical semantics," *Comput. Electron. Agricult.*, vol. 221, Jun. 2024, Art. no. 108922, doi: 10.1016/j.compag.2024.108922.
[17] E. M. Gonzalez, A. Zarei, N. Hendler, T. Simmons, A. Zarei, J. Demieville, R. Strand, B. Rozzi, S. Calleja, H. Ellingson, M. Cosi, S. Davey, D. O. Lavelle, M. J. Truco, T. L. Swetnam, N. Merchant, R. W. Michelmore, E. Lyons, and D. Pauli, "PhytoOracle: Scalable, modular phenomics data processing pipelines," *Frontiers Plant Sci.*, vol. 14, Mar. 2023, Art. no. 1112973, doi: 10.3389/fpls.2023.1112973.
[18] M. M. Rahaman, D. Chen, Z. Gillani, C. Klukas, and M. Chen, "Advanced phenotyping and phenotype data analysis for the study of plant growth and development," *Frontiers Plant Sci.*, vol. 6, p. 619, Aug. 2015, doi: 10.3389/fpls.2015.00619.
[19] A. Itoh, S. N. Njane, M. Hirafuji, and W. Guo, "PREPs: An open-source software for high-throughput field plant phenotyping," *Plant Phenomics*, vol. 6, p. 0221, Jan. 2024, doi: 10.34133/plantphenomics.0221.
[20] B. Wang, C. Yang, J. Zhang, Y. You, H. Wang, and W. Yang, "IHUP: An integrated high-throughput universal phenotyping software platform to accelerate unmanned-aerial-vehicle-based field plant phenotypic data extraction and analysis," *Plant Phenomics*, vol. 6, p. 0164 May 2024, doi: 10.34133/plantphenomics.0164.
[21] C. Klukas, D. Chen, and J.-M. Pape, "Integrated analysis platform: An open-source information system for high-throughput plant phenotyping," *Plant Physiol.*, vol. 165, no. 2, pp. 506–518, Jun. 2014, doi: 10.1104/pp.113.233932.
[22] M. A. Gehan, N. Fahlgren, A. Abbasi, J. C. Berry, S. T. Callen, L. Chavez, A. N. Doust, M. J. Feldman, K. B. Gilbert, J. G. Hodge, J. S. Hoyer, A. Lin, S. Liu, C. Lizárraga, A. Lorence, M. Miller, E. Platon, M. Tessman, and T. Sax, "PlantCV v2: Image analysis software for high-throughput plant phenotyping," *PeerJ*, vol. 5, p. e4088, Dec. 2017, doi: 10.7717/peerj.4088.
[23] H. S. Naik, J. Zhang, A. Lofquist, T. Assefa, S. Sarkar, D. Ackerman, A. Singh, A. K. Singh, and B. Ganapathysubramanian, "A real-time phenotyping framework using machine learning for plant stress severity rating in soybean," *Plant Methods*, vol. 13, no. 1, p. 23, Dec. 2017, doi: 10.1186/s13007-017-0173-7.
[24] C. Addo-Quaye, M. Tuinstra, N. Carraro, C. Weil, and B. P. Dilkes, "Whole-genome sequence accuracy is improved by replication in a population of mutagenized sorghum," *G3 Genes|Genomes|Genetics*, vol. 8, no. 3, pp. 1079–1094, Mar. 2018, doi: 10.1534/g3.117.300301.
[25] A. H. Paterson et al., "The sorghum bicolor genome and the diversification of grasses," *Nature*, vol. 457, no. 7229, pp. 551–556, Jan. 2009, doi: 10.1038/nature07723.
[26] M. J. Ottman. (2016). *Growing Grain Sorghum Arizona*. [Online]. Available: https://repository.arizona.edu/handle/10150/625542
[27] P. W. Brown, "Accessing the Arizona Meteorological Network (AZMET) by Computer," Extension Report, University of Arizona, Tucson, Arizona, Tech. Rep. 8733, 1989.
[28] J. Vazquez Fernandez, J. Lee, S. Serrano Vacca, A. Magana, R. Pesam, B. Benes, and V. Popescu, "Hands-free VR," in *Proc. 20th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2025, pp. 533–542.
[29] N. Harandi, B. Vandenberghe, J. Vankerschaver, S. Depuydt, and A. Van Messem, "How to make sense of 3D representations for plant phenotyping: A compendium of processing and analysis techniques," *Plant Methods*, vol. 19, no. 1, p. 60, Jun. 2023, doi: 10.1186/s13007-023-01031-z.

[30] L. Shapiro, *Embodied Cognition*. Evanston, IL, USA: Routledge, 2019.
[31] N. Hinricher, C. Schröer, and C. Backhaus, "Design of control elements in virtual reality—Investigation of factors influencing operating efficiency, user experience, presence, and workload," *Appl. Sci.*, vol. 13, no. 15, p. 8668, Jul. 2023.
[32] P. Virtanen et al., "SciPy 1.0: Fundamental algorithms for scientific computing in Python," *Nature Methods*, vol. 17, no. 3, pp. 261–272, Mar. 2020, doi: 10.1038/s41592-019-0686-2.
[33] OpenAI. *Chat GPT*. Accessed: Mar. 4, 2025. [Online]. Available: https://openai.com/chatgpt
[34] L. Eren Erdogan, N. Lee, S. Jha, S. Kim, R. Tabrizi, S. Moon, C. Hooper, G. Anumanchipalli, K. Keutzer, and A. Gholami, "TinyAgent: Function calling at the edge," 2024, *arXiv:2409.00608*.
[35] A. Grattafiori et al., "The llama 3 herd of models," 2024, *arXiv:2407.21783*.
[36] D. Guo et al., "DeepSeek-r1: Incentivizing reasoning capability in LLMs via reinforcement learning," 2025, *arXiv:2501.12948*.
[37] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," 2018, *arXiv:1801.09847*.
[38] *Unity Technologies, Unity Game Engine, Version 2022.3.30F1*. Accessed: Mar. 4, 2025. [Online]. Available: https://unity.com/
[39] Meta. *Quest 3: New Mixed Reality Headset*. Accessed: Mar. 4, 2025. [Online]. Available: https://www.meta.com/quest/products/quest-3/
[40] S. Kolhar and J. Jagtap, "Plant trait estimation and classification studies in plant phenotyping using machine vision—A review," *Inf. Process. Agricult.*, vol. 10, no. 1, pp. 114–135, Mar. 2023.
[41] P. E. J. Kivi, M. J. Mäkitalo, J. Žádník, J. Ikkala, V. K. M. Vadakital, and P. O. Jääskeläinen, "Real-time rendering of point clouds with photo-realistic effects: A survey," *IEEE Access*, vol. 10, pp. 13151–13173, 2022.
[42] D. Luebke, M. Reddy, J. D. Cohen, A. Varshney, B. Watson, and R. Huebner, *Level of Detail for 3D Graphics*. Amsterdam, The Netherlands: Elsevier, 2002.
[43] P. Shen, X. Jing, W. Deng, H. Jia, and T. Wu, "PlantGaussian: Exploring 3D Gaussian splatting for cross-time, cross-scene, and realistic 3D plant visualization and beyond," *Crop J.*, vol. 13, no. 2, pp. 607–618, Apr. 2025.
[44] B. Shaheen, M. D. Zane, B.-T. Bui, Shubham, T. Huang, M. Merello, B. Scheelk, S. Crooks, and M. Wu, "ForestSplat: Proof-of-concept for a scalable and high-fidelity forestry mapping tool using 3D Gaussian splatting," *Remote Sens.*, vol. 17, no. 6, p. 993, Mar. 2025.
[45] A. Hameed, S. Möller, and A. Perkis, "How good are virtual hands? Influences of input modality on motor tasks in virtual reality," *J. Environ. Psychol.*, vol. 92, Dec. 2023, Art. no. 102137.
[46] J. Li, X. Qi, S. Hamidreza Nabaei, M. Liu, D. Chen, X. Zhang, X. Yin, and Z. Li, "A survey on 3D reconstruction techniques in plant phenotyping: From classical methods to neural radiance fields (NeRF), 3D Gaussian splatting (3DGS), and beyond," 2025, *arXiv:2505.00737*.
[47] X. Zhou, B. Li, B. Benes, A. Habib, S. Fei, J. Shao, and S. Pirk, "TreeStructor: Forest reconstruction with neural ranking," *IEEE Trans. Geosci. Remote Sens.*, vol. 63, 2025, Art. no. 4408419, doi: 10.1109/TGRS.2025.3558312.
[48] M. Gaillard, C. Miao, J. C. Schnable, and B. Benes, "Voxel carving-based 3D reconstruction of sorghum identifies genetic determinants of light interception efficiency," *Plant Direct*, vol. 4, no. 10, p. e0025, Oct. 2020, doi: 10.1002/pld3.255.
[49] T. L. Swetnam et al., "CyVerse: Cyberinfrastructure for open science," *PLoS Comput. Biol.*, vol. 20, no. 2, 2024, Art. no. e1011270.
[50] Y.-Y. Wang, A. Acero, and C. Chelba, "Is word error rate a good indicator for spoken language understanding accuracy," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand.*, Nov./Dec. 2003, pp. 577–582, doi: 10.1109/ASRU.2003.1318504.
[51] J.-P. Stauffert, F. Niebling, and M. E. Latoschik, "Latency and cybersickness: Impact, causes, and Measures. A review," *Frontiers Virtual Reality*, vol. 1, Nov. 2020, Art. no. 582204.
[52] M. S. Elbamby, C. Perfecto, M. Bennis, and K. Doppler, "Toward low-latency and ultra-reliable virtual reality," *IEEE Netw.*, vol. 32, no. 2, pp. 78–84, Mar. 2018, doi: 10.1109/MNET.2018.1700268.
[53] V. Popescu, S. H. Lee, A. S. Choi, and S. Fahmy, "Complex virtual environments on thin VR systems through continuous near-far partitioning," in *Proc. IEEE Int. Symp. Mixed Augmented Reality (ISMAR)*, Oct. 2022, pp. 35–43, doi: 10.1109/ISMAR55827.2022.00017.
[54] V. Popescu, E. Sacks, Z. Zhang, and J. B. Vázquez, "Complex VEs on all-in-one VR headsets through continuous from-segment visibility computation," in *Proc. IEEE Conf. Virtual Reality 3D User Interfaces (VR)*, Mar. 2025, pp. 359–369, doi: 10.1109/VR59515.2025.00060.
[55] T. Koch and M. Wimmer, "Guided visibility Sampling++," *Proc. ACM Comput. Graph. Interact. Techn.*, vol. 4, no. 1, pp. 1–16, Apr. 2021, doi: 10.1145/3451266.
[56] L. Hu, P. V. Sander, and H. Hoppe, "Parallel view-dependent level-of-detail control," *IEEE Trans. Vis. Comput. Graphics*, vol. 16, no. 5, pp. 718–728, Sep. 2010, doi: 10.1109/TVCG.2009.101.
[57] J. Brooke, "SUS—A quick and dirty usability scale," *Usability Eval. Ind.*, vol. 189, no. 194, pp. 4–7, 1996.
[58] T. Schubert, F. Friedmann, and H. Regenbrecht, "The experience of presence: Factor analytic insights," *Presence*, vol. 10, no. 3, pp. 266–281, Jun. 2001.
[59] J. Leppink, F. Paas, C. P. M. Van der Vleuten, T. Van Gog, and J. J. G. Van Merriënboer, "Development of an instrument for measuring different types of cognitive load," *Behav. Res. Methods*, vol. 45, no. 4, pp. 1058–1072, Dec. 2013.
[60] R. A. Virzi, "Refining the test phase of usability evaluation: How many subjects is enough?" *Hum. Factors: J. Hum. Factors Ergonom. Soc.*, vol. 34, no. 4, pp. 457–468, Aug. 1992.

JORGE VAZQUEZ received the B.S. degree in robotics and digital systems engineering from the Tecnológico de Monterrey. He is currently pursuing the Ph.D. degree in computer science with Purdue University. His research interests include computer graphics, virtual reality, and procedural modeling.

SHUWEN YANG received the B.S. degree in computer science from Purdue University, where she is currently pursuing the Ph.D. degree in computer science. Her research interests include computer graphics, mixed reality, and point-based rendering.

YIQUN ZHANG received the B.S. and M.S. degrees in computer and information technology from Purdue University, where she is currently pursuing the Ph.D. degree in technology with the School of Applied and Creative Computing. Her research interest includes immersive technology-based education.

JEFFREY DEMIEVILLE received the B.S. degree in biological and agricultural engineering from Texas A&M University and the M.S. degree in systems engineering from The University of Arizona. He is currently an Interdisciplinary Engineer with the School of Plant Sciences, The University of Arizona. His research interests include sensors and robotics, operations research, reliability engineering, and data processing pipelines.

BRENNAN HUPPENTHAL is currently pursuing the Ph.D. degree in computer science with The University of Arizona. Their research focuses on computer vision, 3-D reconstruction of vegetation, and trait extraction.

NIRAV MERCHANT is currently the Director of the Data Science Institute, The University of Arizona. He oversees the comprehensive computational cyberinfrastructure for biomedical research and supports projects ranging from large-scale clinical NGS analytics platforms to mobile health interventions. His research interests include data science literacy, large-scale data management platforms, data delivery technologies, managed sensor and mobile platforms, workforce development, and project-based learning.

VOICU POPESCU (Member, IEEE) received the Ph.D. degree in computer science from the University of North Carolina at Chapel Hill. He is currently an Associate Professor of computer science with Purdue University. His research interests include virtual and augmented reality, computer graphics, and visualization.

ALEJANDRA MAGANA (Associate Member, IEEE) received the Ph.D. degree in engineering education from Purdue University. She was inducted into the Purdue University Teaching Academy. She is currently the W. C. Furnas Professor of enterprise excellence with the Department of Computer and Information Technology and a Professor with the School of Engineering Education, Purdue University. She is also a Purdue Faculty Scholar. Her research investigates how model-based cognition in STEM can be better supported through educational and expert technological tools and practices. She received the NSF CAREER Award.

DUKE PAULI is currently an Associate Professor with the School of Plant Sciences, The University of Arizona. His laboratory studies the genetics of heat and drought tolerance in crop plants to reveal how stress-adaptive traits can be leveraged to develop stress-resilient crops.

BEDRICH BENES (Senior Member, IEEE) received the Ph.D. degree from Czech Technical University, Prague. He is currently a Professor and the Associate Head of computer science with Purdue University. His research interests include generative methods and the simulation of natural phenomena. He has published over 250 research articles in his research field. He is a Senior Member of ACM and a Eurographics Fellow.