EyeNavGS: A 6-DoF Navigation Dataset and Record-n-Replay Software for Real-World 3DGS Scenes in VR

Zihao Ding
Rutgers University
Piscataway, NJ, USA
zd75@rutgers.edu

Cheng-Tse Lee
National Tsing Hua University
Hsin-Chu, Taiwan
chengtse@gapp.nthu.edu.tw

Mufeng Zhu
Rutgers University
Piscataway, NJ, USA
mz526@rutgers.edu

Tao Guan
Rutgers University
Piscataway, NJ, USA
tg677@rutgers.edu

Yuan-Chun Sun
National Tsing Hua University
Hsin-Chu, Taiwan
yuanjiun911@gapp.nthu.edu.tw

Cheng-Hsin Hsu
National Tsing Hua University
Hsin-Chu, Taiwan
chsu@cs.nthu.edu.tw

Yao Liu
Rutgers University
Piscataway, NJ, USA
yao.liu@rutgers.edu

Abstract

3D Gaussian Splatting (3DGS) is an emerging media representation that reconstructs real-world 3D scenes in high fidelity, enabling 6-degrees-of-freedom (6-DoF) navigation in virtual reality (VR). However, developing and evaluating 3DGS-enabled applications and optimizing their rendering performance require realistic user navigation data. Such data is currently unavailable for photorealistic 3DGS reconstructions of real-world scenes. This paper introduces EyeNavGS, the first publicly available 6-DoF navigation dataset featuring traces from 46 participants exploring twelve diverse, real-world 3DGS scenes. The dataset was collected at two sites, using the Meta Quest Pro headsets, recording the head pose and eye gaze data for each rendered frame during free world standing 6-DoF navigation. For each of the twelve scenes, we performed careful scene initialization to correct for scene tilt and scale, ensuring a perceptually-comfortable VR experience. We also release our open-source SIBR viewer software fork with record-and-replay functionalities and a suite of utility tools for data processing, conversion, and visualization. The EyeNavGS dataset and its accompanying software tools provide valuable resources for advancing research in 6-DoF viewport prediction, adaptive streaming, 3D saliency, and foveated rendering for 3DGS scenes. The EyeNavGS dataset is available at: **https://symmru.github.io/EyeNavGS/**.

CCS Concepts

• **Computing methodologies → Virtual reality**; **Point-based models**; • **Human-centered computing → Human computer interaction (HCI)**.

Keywords

3D Gaussian Splatting, Virtual Reality, 6-DoF, User Navigation, Eye Gaze

MM '25, Dublin, Ireland

ACM ISBN 979-8-4007-2035-2/2025/10
https://doi.org/10.1145/3746027.3758265

ACM Reference Format:
Zihao Ding, Cheng-Tse Lee, Mufeng Zhu, Tao Guan, Yuan-Chun Sun, Cheng-Hsin Hsu, and Yao Liu. 2025. EyeNavGS: A 6-DoF Navigation Dataset and Record-n-Replay Software for Real-World 3DGS Scenes in VR. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25), October 27–31, 2025, Dublin, Ireland.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3746027.3758265

1 Introduction

Since its introduction in 2023, 3D Gaussian Splatting (3DGS) [21] has quickly emerged as a popular immersive media format for 3D scene representation, enabling high-fidelity, 6-degrees-of-freedom (6-DoF) exploration of complex real-world environments. Due to its fast training time and real-time rendering speed, it has received significant attention from both academia and industry [6, 14, 20, 22, 26, 30, 46]. 3DGS has unlocked new possibilities, including rendering on mobile devices with WebGL support [2, 3, 32, 34] and extending traditional video streaming paradigms to full 6-DoF volumetric content. For example, recent works such as SGSS [47] and L3GS [44] have proposed streaming approaches for static 3DGS scenes. LapisGS [38] introduced a layered 3DGS representation that supports progressive adaptive streaming. Building on LapisGS, LTS [42] proposed approaches for adaptive streaming of dynamic 3DGS scenes.

However, the development and evaluation of these 3DGS-enabled systems and applications are hampered by the lack of suitable datasets. To properly assess system performance of adaptive streaming algorithms, rendering optimizations, compression strategies, and quality of experience under real-world conditions, large-scale datasets recording authentic user interaction with 6-DoF scenes are essential. To the best of our knowledge, no publicly available dataset currently captures such 6-DoF user navigation traces for real-world scenes reconstructed by 3DGS. The absence of such datasets forces researchers to rely on synthetic traces [29, 47] or datasets collected from different 3D representations [23], which may not faithfully represent user interactions with high-fidelity 3DGS content.

To close this gap, this paper introduces EyeNavGS, the first publicly available dataset of user navigation traces. The dataset includes traces through twelve scenes. These scenes include both indoor and outdoor environments, offering diverse visual characteristics for studying user navigation behaviors and performance-quality tradeoffs in virtual reality (VR). Our contributions are summarized as follows:

- **The EyeNavGS Dataset.** We collected navigation traces of **46** participants. Traces were collected at two physical locations. Each trace includes a human user's exploration of twelve diverse indoor and outdoor scenes reconstructed by 3DGS. Each scene underwent careful initialization for tilt correction, metric scale establishment, and starting viewpoint selection to ensure perceptually comfortable VR experiences. The dataset includes per-frame head pose and eye gaze data, captured with Meta Quest Pro headsets during free world standing exploration.
- **The EyeNavGS Record-n-Replay Software.** We release our open-source software, a fork of the SIBR viewer [8] for 3DGS, enhanced with record-and-replay functionality. This fork includes capabilities for recording user traces and replaying these traces frame-by-frame for visualization, video generation, and detailed offline analysis.
- **The EyeNavGS Utility Tools.** In addition to the core software, we provide a suite of utility tools. These tools include conversion operations to ease integration with other frameworks and allow visualization of collected traces.

We anticipate this dataset and its accompanying software will facilitate more reliable and comprehensive evaluations of 6-DoF viewport prediction, view-adaptive streaming, 3D saliency, and foveated rendering for 3DGS. We also encourage collaborative expansion of this dataset, aiming to create a richer collection of data for advancing research in immersive media experiences.

2 Related Work

2.1 6-DoF Navigation Datasets

The importance of 6-DoF navigation datasets in evaluating the streaming performance and user experience in immersive environments is well-recognized in the research community in recent years. While several 6-DoF navigation datasets have been created to date, each of them has their distinct focus and limitations.

User navigation with synthetically generated environments. Khan and Chakareski [9, 23] introduced the "NJIT 6DOF VR Navigation Dataset", which recorded 6-DoF traces of three users exploring a synthetic "Virtual Museum" (sourced from the Unity Asset Store) using an HTC Vive wireless VR headset. Similarly, Chen et al. [10] collected the VRViewportPose dataset, recording viewing traces of 30 participants on three different platforms, a desktop, an Oculus Quest 2 VR headset, and an Android smartphone as they interacted with three VR games with synthetic scenes. Most recently, Ouellette et al. [33] created a point cloud video dataset with user behavior traces collected via a Meta Quest 2 headset. The environment in this dataset mainly consists of a synthetically modeled maze.

While these datasets, focusing on synthetic scenes, offer valuable insights into user navigation behaviors, they do not capture interactions within reconstructed representations of real-world scenes. Real-world characteristics, such as fine-grained textural details or subtle lighting variations, can influence user behavior [28]. Such differing user behaviors could cause researchers to draw misleading conclusions about the real-world effectiveness of 3DGS systems.

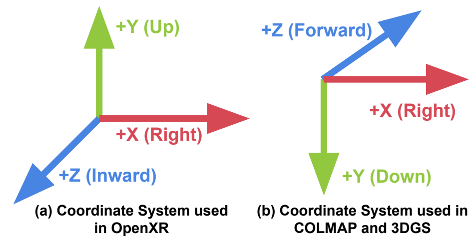


Figure 1: SIBR resolves the coordinate system mismatch between OpenXR and 3DGS with a 180-degree camera rotation around the x-axis.

User interaction with dynamic point clouds. Subramanyam et al. [41] created a 6-DoF navigation dataset where users viewed 150 frames of dynamic point cloud sequences from the 8i dataset [12] using an Oculus Rift headset. The 8i dataset features point cloud representations of individual human subjects. As a result, user navigation typically follows an "outside-looking-in" pattern. Thus, the viewing patterns are likely to be substantially different from free exploration of expansive virtual worlds. Hu et al. released volumetric video viewing behavior dataset [19] , recorded with the Meta Quest Pro headset. In this dataset, 50 participants watched 26 volumetric videos, represented as point clouds, from the FSVVD dataset [18]. A limitation of using point clouds to collect these navigation datasets lies in their their relatively low rendered visual quality. Even with careful calibration and alignment, the produced point clouds cannot render views at a photorealistic quality comparable to emerging immersive media representations, such as neural radiance fields (NeRF) [31] and 3DGS.

In summary, while several 6-DoF user navigation datasets have been collected using synthetic environments and dynamic point clouds, they do not address navigation within photorealistic reconstructions of real-world scenes. Our dataset bridges this gap by providing recorded traces of users navigating 3DGS scenes.

2.2 OpenXR and SIBR Viewer for 3DGS in VR

To experience 3DGS scenes immersively in VR, we use the open-source SIBR viewer [8]. Specifically, our data collection software is a fork of its `gaussian_code_release_openxr` branch, which renders 3DGS views to head-mounted displays (HMDs) via OpenXR [1]. OpenXR provides a standardized API for VR and augmented reality (AR) applications and defines several reference coordinate spaces. Figure 1(a) shows the OpenXR coordinate systems, which are right-handed. In particular, OpenXR defines three main types of reference spaces:

- **View Space.** The space is relative to the user's head. For stereo VR headsets, its origin is centered between two eyes. The axes are defined as +X to the right, +Y up, and -Z in the forward viewing direction.
- **Local Space.** For VR devices that support 3-DoF rotational tracking only, they only support the "Local Space", where the headset is locked to a fixed origin in the world, typically the user's starting position, with the +Y axis aligned with gravity. It is suitable for stationary or "seated" experiences where the user does not physically walk around.
- **Stage Space.** For VR devices that support the full 6-DoF tracking, they can support the "Stage Space". The "Stage Space" defines a

flat, rectangular area on the physical floor that the user can freely walk within–analogous to a performance stage. The XZ plane is aligned with the floor, the +Y axis defines the "up" direction, and the origin is fixed relative to the physical space. The "Stage Space" allows an application to use tracked physical movements (position and orientation).

SIBR VR Viewing Modes. Using the "Local" and "Stage" spaces, the SIBR `gaussian_code_release_openxr` branch supports two VR viewing modes: `seated` and `free world standing` (`fws`). The `seated` mode is for stationary use, e.g., for VR headsets that supports 3-DoF (i.e., rotational) tracking only, mapping head rotation to orientation and controller input to position.

The `free world standing` (`fws`) mode uses the "Stage Space" to map the user's tracked physical movements 1:1 to the virtual scene and requires a VR headset that supports the full 6-DoF (i.e., both rotational and positional) tracking.

For our data collection, we used Meta Quest Pro, a headset with 6-DoF tracking capabilities. We thus only used the `fws` mode to capture users' natural physical movements in our trace collection.

Coordinate Systems Mismatch. Gaussians in a 3DGS scene are trained from an initial point cloud obtained via COLMAP [36, 37]. Thus, 3DGS inherits COLMAP's coordinate system, which is a right-handed system where the +Y axis points downwards and the +Z axis points forward (as shown in Figure 1(b)). This convention conflicts with OpenXR's coordinate systems, where +Y points upwards. To resolve this mismatch, the SIBR viewer source code rotates the camera 180 degrees around the X-axis to ensure the scene is rendered upright to the user.

3 EyeNavGS Record-n-Replay Software

3.1 Scene Initialization

To prepare each trained 3DGS scene for immersive exploration, we correct each raw scene to align with human assumptions about the physical world. These corrections include i) correcting the initial quaternion to fix scene tilt, ii) selecting a per-scene scale factor to ensure objects match their real-world proportions, and iii) establishing an example starting viewing position. Values for each scene are shown in Table 1.

Scene Tilt and Orientation Correction. 3DGS scenes are trained from initial point clouds generated by COLMAP [36, 37]. However, the coordinate system reconstructed by COLMAP is not inherently gravity-aligned. Improperly oriented scenes frequently result in disorienting tilts, unnatural slopes, skewed camera behavior that degrade the sense of presence and spatial coherence within the virtual environment. To create a perceptually comfortable experience in VR, we must first align the virtual scene with gravity.

Instead of modifying the trained 3DGS `.ply` file for each scene, our solution is to apply a corrective transformation when each scene is loaded. More specifically, the initial quaternion, which rectifies any scene tilt and establishes a level ground plane, is set via the `poseInReferenceSpace` member of the `XrReferenceSpaceCreateInfo` structure.

To find the amount of scene tilt, we implemented a robust procedure using Blender [7] and the KIRI Engine add-on [24], which supports 3DGS point data. Within Blender, we inserted a reference plane perpendicular to the Y axis (which aligns with gravity) and manually adjusted the scene's orientation to ensure its ground plane is orthogonal to the virtual Y-axis and matches the reference plane. This process effectively corrected any residual tilt, ensuring that users perceive the scene as grounded and stable, avoiding perceptual illusions of being on a slope. These rotation parameters were subsequently exported as quaternions and applied at runtime during VR rendering to ensure proper alignment with the user's physical "stage" area.

Table 1: Scene Initialization Parameters

Scene	Initial Quaternion (q_x, q_y, q_z, q_w)	Scale	Example Init. Pos. (x, y, z)
truck	-0.0896, 0, 0, 0.9960	0.76	0, 2.1, -4
treehill	-0.1961, 0, 0, 0.9806	12	2, 1.4, 2
train	0.0499, 0, 0.01, 0.9987	0.36	2, -1, 6
stump	-0.3950, 0, 0, 0.9187	1	-1, 2.65, -2.5
room	-0.2334, 0, 0, 0.9724	2	0, 1.15, 0
playroom	-0.1961, 0, 0, 0.9806	2.7	0, 0.88, 0
drjohnson	-0.3699, 0, 0.5976, 0.7114	1	0, 1.5, 0
bicycle	-0.1142, 0, 0, 0.9935	1.25	1.5, 1.1, 0
nyc	-0.1483, 0, 0, 0.9888	0.64	-1.6, 4.4, 4
london	0, 0, 0, 1	0.53	18, 12, -11
berlin	0.0299, 0, -0.0599, 0.9978	0.8	-1, 1.8, -1.3
alameda	-0.1867, 0, 0, 0.9824	0.64	3, 2.6, -1

Scene Scale Calibration. Another critical limitation of raw, trained 3DGS scenes is the absence of an intrinsic real-world scale, also due to COLMAP. When rendered stereoscopically in a VR headset, this lack of calibration between scene units and physical world units can severely distort perceived object size. For example, under-scaled scenes can cause the users to feel disproportionately large, like a giant. This occurs because the physical inter-pupillary distance (IPD) becomes effectively magnified relative to the virtual world's scale. Since VR rendering inherently relies on accurate simulation of binocular disparity between the user's eyes, calibrating the scene scale is essential for preserving immersion and visual comfort.

Similar to scene tilt correction, we avoid directly modifying the trained 3DGS `.ply` files. Instead, we apply a per-scene `scale` factor at runtime that maps movements in real-world metric measurements to the scene's virtual units. Using the Blende-KIRI add-on workflow, we introduced dimensionally accurate reference objects, e.g., a 1-meter cube, into each scene. By comparing known dimensions from the real scene (e.g., width of a vehicle, rise height of staircases) to their 3DGS representations, we iteratively adjusted the scene scale within Blender's unit system. These calibrated scale factors were recorded and applied during runtime, ensuring the perceived virtual scene conforms to real-world proportions and supports perceptually correct IPD rendering for stereo vision.

Initial View Positioning. Besides tilt and scale, the initial view position also influences the user's first impression and subsequent exploration. The default origin ($[0, 0, 0]$) of a 3DGS scene often corresponds to the center of the captured volume, which can result in undesirable starting viewpoints, such as inside a tree or a wall or floating in the air. To improve user experiences, we manually selected semantically meaningful and physically plausible initial camera positions for each scene, typically floor-level regions with ample surrounding navigability. These locations were chosen to emulate natural human perspective, facilitate intuitive exploration, and avoid immediate occlusions or collisions.

3.2 Record-n-Replay Features

We extend the SIBR core rendering engine with record-and-replay features tailored for 3DGS in OpenXR.

The Record Mode. In this mode, our modified OpenXR module captures a user's 6-DoF navigation and gaze data on a per-frame basis. For each eye, the module records a comprehensive set of parameters: field-of-view (FOV), eye position, head orientation (quaternion), and, when available, gaze position and orientation (quaternions). This data is synchronized with the rendering loop and logged with precise timestamps to a structured `csv` file, ensuring a complete and accurate offline reconstruction of the user's viewing experience.

The Replay Mode. The replay mode uses recorded traces to reproduce the original VR session for analysis and view generation. During replay, the system parses the trace and injects the logged data line-by-line into the rendering pipeline, overriding the live HMD pose information. The `loadViewData()` method handles this internally by reading the `csv` trace and updating the per-frame `ViewData` structure before rendering.

To generate video output, rendered frames are captured from GPU memory using OpenGL's `glGetTexImage()` API and then converted into an OpenCV-compatible format for efficient encoding. This process generates two separate videos, one for each eye, to precisely replicate the original stereoscopic experience.

3.3 Data Output Format

Table 2 outlines the structure of the recorded `csv` traces. Each entry consists of two rows for the left and right eyes, identified by the `ViewIndex` column (0 and 1, respectively). The FOV for each eye is defined by four radian values: `FOV1` (left), `FOV2` (right), `FOV3` (top), and `FOV4` (bottom). Due to the interpupillary distance (IPD), the world coordinates of the eye positions (`Pos_X`, `Pos_Y`, `Pos_Z`) differ between the two eyes. In contrast, the head orientation, represented by the quaternions `Quat_X`, `Quat_Y`, `Quat_Z`, and `Quat_W`, remains the same for both.

The traces also include user gaze data on supported headsets (e.g., Meta Quest Pro). The **eye gaze position** in world space is given by `GazePos_X`, `GazePos_Y`, and `GazePos_Z`. These values are similar but not identical to the **eye positions** in the `Pos_*` columns. Similarly, `GazeQ_X`, `GazeQ_Y`, `GazeQ_Z`, and `GazeQ_W` provide the **eye gaze orientation** as a world-space quaternion. This can differ significantly from the head orientation, accounting for eye movement within the sockets. Finally, each frame is marked with a relative timestamp in milliseconds.

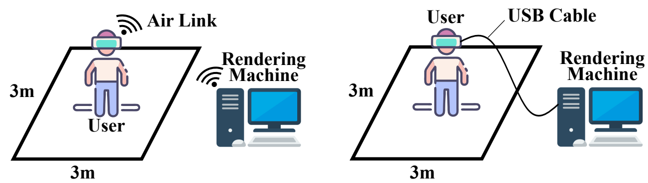
Table 3 presents sample data from two consecutive frames. Note that each frame is represented by two rows, corresponding to the left and right eye views.

4 EyeNavGS Dataset

Sites. We collected user navigation traces at two sites: Rutgers University (RU) in the New Jersey, USA and National Tsing Hua University (NTHU) in Hsin-Chu, Taiwan. The data collection protocols at both institutions received approval from their respective Institutional Review Boards (IRB).

Table 2: Columns of the Recorded User Traces in csv

Name	Description
ViewIndex	Left eye: 0; right eye: 1.
FOV1 (rad)	The left FOV angle.
FOV2 (rad)	The right FOV angle.
FOV3 (rad)	The top FOV angle.
FOV4 (rad)	The bottom FOV angle.
Position X,Y,Z	Eye position (i.e., head position offset by half of IPD) in the world space.
Quaternion X,Y,Z,W	Head/view orientation as a quaternion, in the world space (same for left and right eyes).
GazePos X,Y,Z	Eye gaze position in the world space.
GazeQ X,Y,Z,W	Eye gaze orientation as a quaternion, in the world space.
Timestamp	Time offset in ms since the left eye of the first frame is recorded.



(a) Untethered user trace collection setup at the RU site. (b) Tethered user trace collection setup at the NTHU site.

Figure 2: Real-time user trace collection setup showing the tracked space, user with HMD, HMD connection, and the rendering machine.

Figure 3: The 12 diverse 3DGS scenes used in our dataset. The first eight scenes (top row and first two of bottom row) are from the original 3DGS paper [21]. The final four scenes were trained from the ZipNeRF dataset [5, 11] to expand scene variety.

Participants. A total of 46 participants were recruited across the two sites: 22 at RU and 24 at NTHU. The participants' ages ranged from 18 to 70.

Apparatus. As detailed in Table 4, participants at both collection sites used a Meta Quest Pro headset with eye tracking enabled, navigating a 3 m × 3 m physical play area (OpenXR Stage Space). The primary distinction between the sites, illustrated in Figure 2, was the HMD connection method and GPU. The RU site provided an untethered experience via Meta Air Link with an Nvidia RTX 4090, whereas the NTHU site used a tethered USB Link cable with an Nvidia RTX 3080 Ti.

Table 3: Sample Values Extracted from a Recorded csv

View Index	FOV1 (rad)	FOV2 (rad)	FOV3 (rad)	FOV4 (rad)	Pos_X	Pos_Y	Pos_Z	$Quat_X$	$Quat_Y$	$Quat_Z$	$Quat_W$	$GazeQ_X$	$GazeQ_Y$	$GazeQ_Z$	$GazeQ_W$	$GazePos_X$	$GazePos_Y$	$GazePos_Z$
0	−0.942	0.698	−0.942	0.733	−3.669	−3.657	4.658	0.495	0.294	0.124	0.808	0.251	0.085	0.024	0.964	−3.668	−3.657	4.657
1	−0.698	0.942	−0.942	0.733	−3.513	−3.561	4.588	0.495	0.294	0.124	0.808	0.245	0.104	0.045	0.963	−3.513	−3.561	4.589
0	−0.942	0.698	−0.942	0.733	−3.669	−3.656	4.657	0.494	0.294	0.123	0.809	0.249	0.087	0.026	0.964	−3.668	−3.656	4.657
1	−0.698	0.942	−0.942	0.733	−3.512	−3.560	4.588	0.494	0.294	0.123	0.809	0.243	0.106	0.048	0.963	−3.513	−3.561	4.588

Table 4: Dataset Collection Setups at the RU and NTHU Sites

Component	RU Setup	NTHU Setup
HMD	Meta Quest Pro	Meta Quest Pro
Rendering Machine	GPU: Nvidia RTX 4090 CPU: Intel i9-14900KF	GPU: NVIDIA RTX 3080 Ti CPU: Intel i9-9920X
Connection	Wireless (Meta Air Link)	Wired (5-meter USB Link)
Area	3 m × 3 m	3 m × 3 m
OS	Windows 11	Windows 10
Participants	22	24

Stimuli. Stereoscopic views are rendered at 2160 × 2224 resolution per eye. Twelve distinct real-world scenes were used for user navigation. Eight of these scenes were selected from the thirteen pre-trained 3DGS scenes presented in the original 3DGS paper [21], shown in Figure 3. These scenes originate from three datasets: the MipNeRF360 dataset [4], the Tanks&Temples dataset [25], and the Deep Blending dataset [17]. They contain a variety of real-world scenes including indoor, outdoor, and natural environments. For our study, we used the "garden" scene from this pre-trained dataset for participant training. We excluded "kitchen", "flowers", "counter", and "bonsai" scenes due to their limited viewing volumes, making them unsuitable for VR navigation.

In addition, we also trained 3DGS representations for four scenes from the ZipNeRF dataset [5, 11], namely "alameda", "berlin", "london", and "nyc", shown in Figure 3. These scenes are mainly indoor scenes, while some of them also feature outdoor sections.

Procedure. Each participants explored twelve scenes for one minute each, in the `free world standing` (`fws`) mode. Participants were instructed to freely explore the virtual scene via natural movements such as walking and turning in the 3 m × 3 m area. No specific task was assigned to the participants during the sessions. A one-minute break was provided between each scene exploration to allow for participant rest and for the system to load the next scene.

Dataset. Our dataset is organized into scene-specific folders (e.g., "train", "truck"). Within each scene folder, each user trace is stored as a `csv` file named as: `{user}_{scene}.csv`. For instance, `user1_truck.csv` records the trace of user1 exploring the "truck" scene. For a detailed description of the dataset, please visit our project website: **https://symmru.github.io/EyeNavGS/**.

5 `EyeNavGS` Utility Tools

We developed a suite of utilities for the interoperability, reproducibility, and visualization of the collected traces. We include these utilities in the `EyeNavGS` software's `utils` folder and briefly describe them in this section.

Conversion from Virtual World Coordinates to Physical Stage Coordinates. The `EyeNavGS` dataset records user navigation traces including the head pose and eye gaze information in "virtual world coordinates". This facilitates direct replay of the traces for view generation. We provide a utility to convert these traces to "physical stage coordinates", which represent the user's movements in the physical world on a 1:1 metric scale (i.e., a one-meter physical movement in corresponds to a one-unit displacement in these coordinates). This conversion undoes the scene initialization transforms including the scene tilt correction, scene scaling, and initial view positioning, detailed in Section 3.1. The resulting "physical stage coordinates" effectively reconstruct user's movement in the 3 m × 3 m physical space. We use the converted physical stage coordinates in our dataset analysis in Section 6.

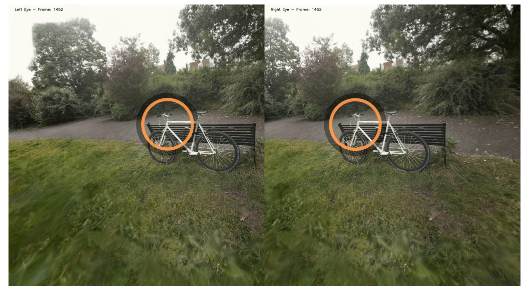


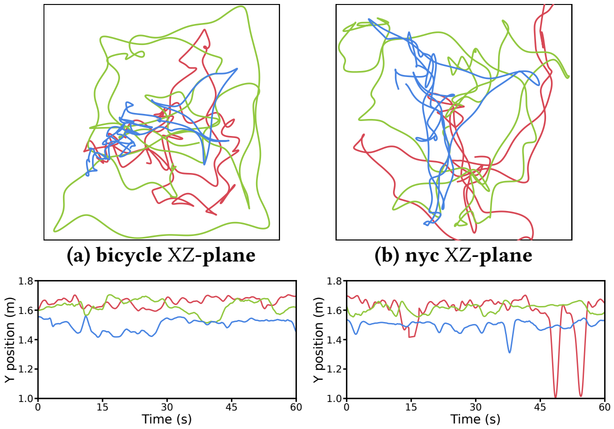
Figure 4: Example eye gaze visualization of the bicycle scene.

Compatibility with Other Frameworks. The `EyeNavGS` dataset stores user navigation traces in the `.csv` format, recording camera positions and rotation quaternions. Popular modern viewers, such as the web viewer in NeRFstudio [43] and SGSS [47], use `.json` format with a 4 × 4 homogeneous matrix for pose representation in a different coordinate system for trace replay. To enable compatibility with these frameworks, `EyeNavGS` includes two utility tools: `csv2json.cpp` for converting recorded traces to `.json` format in the correct coordinate system and `json2csv.cpp` for converting `.json` files exported from other frameworks to `.csv` format, also in the correct coordinate system, enabling replay in the `EyeNavGS` viewer. With these two tools, `EyeNavGS` ensures compatibility with other frameworks.

Eye Gaze Visualization. To visualize user attention during scene navigation, our eye gaze tool processes replayed stereoscopic videos by overlaying the recorded eye gaze data. For each frame, a visual marker (e.g., a circle) is rendered on both the left and right eye views to indicate the participant's fixation. The resulting video displays the stereoscopic view side-by-side, illustrating the user's gaze path. Figure 4 shows an example frame with eye gaze overlaid on stereoscopic rendered views.

6 Dataset Analysis

Figure 5 shows two examples of the participants' tracked position in the 3 m × 3 m physical stage area. For each figure, we plot the traces of three users' movements in the 60-second period, in different colors. Since movement along the vertical Y-axis (aligned with gravity) is much less pronounced than movement along the



(a) bicycle XZ-plane (b) nyc XZ-plane

(c) bicycle Y-axis (d) nyc Y-axis

Figure 5: User movement trajectories for an outdoor and an indoor scene, with three users per scene plotted with different colors. Top row: horizontal user movements (on the XZ plane) within the 3 m × 3 m area; Bottom row: corresponding changes of the headset height (along the Y-axis) over time.

Table 5: Statistics of Navigation Traces Per-session (60 Seconds) at Two Collection Sites

Metric	RU Site	NTHU Site
Avg. # of recorded frames	3,420	2,396
Frames per second (fps)	57.04	39.95
Total distance traveled (meters)	17.27	13.62

horizontal XZ-plane, we plot it on a separate graph for clarity. The average distances walked by the participants in the 60-second viewing sessions are reported in Table 5.

As the GPU and HMD connection methods are different at the RU and NTHU sites, the observed frame rates during trace collection also differ. As shown in Table 5, the RU site, with its more powerful GPU for rendering, averaged approximately 57 frames per second (fps), compared to approximately 40 fps at the NTHU site. The average frame rates for each scene are provided in Table 6.

Table 6: Average Frame Rates for Each Scene at Two Sites

Site	alameda	berlin	bicycle	drjohnson	london	nyc
RU	43.68	55.33	42.12	37.47	65.32	44.18
NTHU	36.68	40.62	33.15	33.81	45.86	37.66

Site	playroom	room	stump	train	treehill	truck
RU	47.53	71.37	68.80	71.30	70.91	69.75
NTHU	36.67	41.80	44.43	43.65	41.63	43.17

7 Dataset Use Cases

The EyeNavGS dataset, with its detailed 6-DoF navigation traces including head pose and eye gaze information, offers valuable opportunities for research in immersive computing systems. We describe example use cases in this section.

6-DoF Viewport Prediction and Streaming Optimization. The EyeNavGS dataset addresses the lack of 6-DoF user navigation traces in reconstructed real-world scenes, which existing datasets [9, 10, 23, 41] do not provide. The fine-grained head pose and eye gaze of reconstructed real-world scenes in VR can be used for developing 6-DoF viewport prediction algorithms. Such prediction can inform the design of adaptive media streaming algorithms to fetch only the content needed for rendering the user's viewport without wasting bandwidth on unviewed portion of the representation, e.g., [16, 44, 47].

3D Saliency and Saliency in VR. The rich per-frame eye gaze information in the EyeNavGS dataset also offers opportunities for 3D saliency research [13, 40, 45]. This detailed fixation data can be aggregated across participants to create the groundtruth 3D saliency maps of the 3DGS scenes. These maps can then be used for training 3D saliency models that better predict where the users will look in reconstructed real-world scenes. Furthermore, since the reconstructed 3DGS scene may contain imperfections such as under-constructed areas and other visual artifacts, our dataset also enables studies into how these imperfections influence user gaze and navigation behavior.

Foveated Rendering Optimization. Foveated rendering is an important technique in VR designed to reduce the rendering computation demand and improve the frame rates [27, 35, 39]. Given that the visual acuity of human vision decreases sharply away from the foveal center, foveated rendering works by reducing the shading rates in the peripheral (non-foveal) region of the user's view. This can achieve significant performance gain while with minimal impact on visual quality. Existing works have explored applying foveated rendering for 3DGS rendering, e.g., MetaSapiens [29] and VR-Splatting [15]. Our EyeNavGS dataset features per-frame eye gaze traces collected during free world standing 6-DoF navigation of 3DGS scenes, facilitating evaluation and optimization of real-world performance of these foveated rendering techniques.

8 Conclusion

In this paper, we present EyeNavGS, the first publicly available 6-DoF navigation dataset built on photorealistic 3DGS reconstructions of real-world scenes, together with an open-source record-and-replay software fork of the SIBR viewer. EyeNavGS focuses on capturing detailed and realistic behavior. It does so by collecting user traces in scenes of real-world environments, carefully calibrated (tilt, scale, and starting viewpoint) to maintain realism. Our dataset captures stereoscopic pose, field-of-view, and eye-tracking traces. In addition, multi-site data collection (46 participants across two institutions) ensure a diverse population of participants. The realism, detail, and diversity of EyeNavGS fills a critical gap for immersive media researchers, allowing user-centric evaluation in core areas such as streaming, rendering, and compression.

Acknowledgments

We appreciate constructive comments from anonymous referees. The work of Zihao Ding, Mufeng Zhu, Tao Guan, and Yao Liu is partially supported by NSF under grant CNS-2200048. The work of Cheng-Hsin Hsu is partially supported by the National Science and Technology Council (NSTC) of Taiwan through an Overseas Research Fellowship, under grant 114-2918-I-007-004.

References

[1] The Khronos Group Inc. [n. d.]. *OpenXR Overview*. The Khronos Group Inc. https://www.khronos.org/openxr/

[2] 2023. *gsplat — 3D Gaussian Splatting WebGL viewer*.

[3] antimatter15. 2023. Splat: WebGL Gaussian Splatting Renderer. https://github.com/antimatter15/splat.

[4] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. 2022. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5470–5479.

[5] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. 2023. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 19697–19705.

[6] Harun Barutcu. 2024. 404 Gen: Bringing Text to 3D Gaussian Splatting to Unity. *Radiance Fields* (17 April 2024). https://radiancefields.com/404-gen-bringing-text-to-3d-gaussian-splatting-to-unity

[7] Blender Foundation. [n. d.]. *blender.org*. https://www.blender.org/

[8] Sebastien Bonopera, Jerome Esnault, Siddhant Prakash, Simon Rodriguez, Theo Thonat, Mehdi Benadel, Gaurav Chaurasia, Julien Philip, and George Drettakis. 2020. sibr: A System for Image Based Rendering. https://gitlab.inria.fr/sibr/sibr_core

[9] Jacob Chakareski, Mahmudur Khan, Tanguy Ropitault, and Steve Blandino. 2020. 6DOF virtual reality dataset and performance evaluation of millimeter wave vs. free-space-optical indoor communications systems for lifelike mobile VR streaming. In *2020 54th Asilomar Conference on Signals, Systems, and Computers*. IEEE, 1051–1058.

[10] Ying Chen, Hojung Kwon, Hazer Inaltekin, and Maria Gorlatova. 2022. VR Viewport Pose Model for Quantifying and Exploiting Frame Correlations. In *Proc. IEEE INFOCOM*.

[11] Daniel Duckworth, Peter Hedman, Christian Reiser, Peter Zhizhin, Jean-François Thibert, Mario Lučić, Richard Szeliski, and Jonathan T Barron. 2024. Smerf: Streamable memory efficient radiance fields for real-time large-scene exploration. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–13.

[12] Eugene d'Eon, Bob Harrison, Taos Myers, and Philip A Chou. 2017. 8i voxelized full bodies-a voxelized point cloud dataset. *ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document WG11M40059/WG1M74006* 7, 8 (2017), 11.

[13] Elias Ennadifi, Thierry Ravet, Matei Mancas, Mohammed El Amine Mokhtari, and Bernard Gosselin. 2023. Enhancing VR Gaming Experience using Computational Attention Models and Eye-Tracking. In *Proceedings of the 2023 ACM international conference on interactive media experiences*. 194–198.

[14] Guangchi Fang and Bing Wang. 2024. Mini-splatting: Representing scenes with a constrained number of gaussians. In *European Conference on Computer Vision*. Springer, 165–181.

[15] Linus Franke, Laura Fink, and Marc Stamminger. 2025. Vr-splatting: Foveated radiance field rendering via 3d gaussian splatting and neural points. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 8, 1 (2025), 1–21.

[16] Serhan Gül, Sebastian Bosse, Dimitri Podborski, Thomas Schierl, and Cornelius Hellge. 2020. Kalman filter-based head motion prediction for cloud-based mixed reality. In *Proceedings of the 28th ACM international conference on multimedia*. 3632–3641.

[17] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. 2018. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics (ToG)* 37, 6 (2018), 1–15.

[18] Kaiyuan Hu, Yili Jin, Haowen Yang, Junhua Liu, and Fangxin Wang. 2023. FSVVD: A dataset of full scene volumetric video. In *Proceedings of the 14th Conference on ACM Multimedia Systems*. 410–415.

[19] Kaiyuan Hu, Haowen Yang, Yili Jin, Junhua Liu, Yongting Chen, Miao Zhang, and Fangxin Wang. 2023. Understanding user behavior in volumetric video watching: Dataset, analysis and prediction. In *Proceedings of the 31st ACM International Conference on Multimedia*. 1108–1116.

[20] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2024. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*. 1–11.

[21] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics* 42, 4 (2023), 1–14.

[22] Bernhard Kerbl, Andreas Meuleman, Georgios Kopanas, Michael Wimmer, Alexandre Lanvin, and George Drettakis. 2024. A hierarchical 3d gaussian representation for real-time rendering of very large datasets. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–15.

[23] M. Khan and J. Chakareski. 2020. NJIT 6DOF VR Navigation Dataset. https://www.jakov.org. https://www.jakov.org Accessed: 2025-05-06.

[24] KIRI Engine. 2025. *3DGS Render Blender Addon - KIRI Engine*. https://www.kiriengine.app/blender-addon/3dgs-render

[25] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. 2017. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)* 36, 4 (2017), 1–13.

[26] Muhammed Kocabas, Jen-Hao Rick Chang, James Gabriel, Oncel Tuzel, and Anurag Ranjan. 2024. Hugs: Human gaussian splats. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 505–515.

[27] Brooke Krajancich, Petr Kellnhofer, and Gordon Wetzstein. 2023. Towards attention–aware foveated rendering. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–10.

[28] Simon Lessels and Roy A Ruddle. 2004. Changes in navigational behaviour produced by a wide field of view and a high fidelity visual scene. In *Proceedings of the 10th Eurographics Symposium on Virtual Environments*. Eurographics, 71–78.

[29] Weikai Lin, Yu Feng, and Yuhao Zhu. 2025. MetaSapiens: Real-Time Neural Rendering with Efficiency-Aware Pruning and Accelerated Foveated Rendering. In *Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1*. 669–682.

[30] Jean-Eudes Marvie and Pascal Gautron. 2025. Real-Time GPU-Accelerated Gaussian Splatting with NVIDIA DesignWorks Sample vk_gaussian_splatting. *NVIDIA Technical Blog* (23 April 2025). https://developer.nvidia.com/blog/real-time-gpu-accelerated-gaussian-splatting-with-nvidia-designworks-sample-vk_gaussian_splatting/

[31] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*.

[32] mkkellogg. 2024. GaussianSplats3D: Three.js-based Gaussian Splatting renderer. https://github.com/mkkellogg/GaussianSplats3D.

[33] Jérémy Ouellette, Jashanjot Singh Sidhu, and Abdelhak Bentaleb. 2025. Maze-Lab: A Large-Scale Dynamic Volumetric Point Cloud Video Dataset With User Behavior Traces. In *Proceedings of the 16th ACM Multimedia Systems Conference*. 298–304.

[34] Panagiotis Papantonakis, Georgios Kopanas, Bernhard Kerbl, Alexandre Lanvin, and George Drettakis. 2024. Reducing the memory footprint of 3d gaussian splatting. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 7, 1 (2024), 1–17.

[35] Anjul Patney, Marco Salvi, Joohwan Kim, Anton Kaplanyan, Chris Wyman, Nir Benty, David Luebke, and Aaron Lefohn. 2016. Towards foveated rendering for gaze-tracked virtual reality. *ACM Transactions On Graphics (TOG)* 35, 6 (2016), 1–12.

[36] Johannes Lutz Schönberger and Jan-Michael Frahm. 2016. Structure-from-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

[37] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. 2016. Pixelwise View Selection for Unstructured Multi-View Stereo. In *European Conference on Computer Vision (ECCV)*.

[38] Yuang Shi, Géraldine Morin, Simone Gasparini, and Wei Tsang Ooi. 2024. Lapisgs: Layered progressive 3d gaussian splatting for adaptive streaming. *arXiv preprint arXiv:2408.14823* (2024).

[39] Rahul Singh, Muhammad Huzaifa, Jeffrey Liu, Anjul Patney, Hashim Sharif, Yifan Zhao, and Sarita Adve. 2023. Power, performance, and image quality tradeoffs in foveated rendering. In *2023 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*. IEEE, 205–214.

[40] Vincent Sitzmann, Ana Serrano, Amy Pavel, Maneesh Agrawala, Diego Gutierrez, Belen Masia, and Gordon Wetzstein. 2018. Saliency in VR: How do people explore virtual environments? *IEEE transactions on visualization and computer graphics* 24, 4 (2018), 1633–1642.

[41] Shishir Subramanyam, Irene Viola, Alan Hanjalic, and Pablo Cesar. 2020. User centered adaptive streaming of dynamic point clouds with low complexity tiling. In *Proceedings of the 28th ACM international conference on multimedia*. 3669–3677.

[42] Yuan-Chun Sun, Yuang Shi, Cheng-Tse Lee, Mufeng Zhu, Wei Tsang Ooi, Yao Liu, Chun-Ying Huang, and Cheng-Hsin Hsu. 2025. LTS: A DASH streaming system for dynamic multi-layer 3D Gaussian splatting scenes. In *Proceedings of the 16th ACM Multimedia Systems Conference*. 136–147.

[43] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. 2023. Nerfstudio: A Modular Framework for Neural Radiance Field Development. In *ACM SIGGRAPH 2023 Conference Proceedings (SIGGRAPH '23)*.

[44] Yi-Zhen Tsai, Xuechen Zhang, Zheng Li, and Jiasi Chen. 2025. L3GS: Layered 3D Gaussian Splats for Efficient 3D Scene Delivery. *arXiv preprint arXiv:2504.05517* (2025).

[45] Yao Wang, Qi Dai, Mihai Bâce, Karsten Klein, and Andreas Bulling. 2024. Saliency3D: a 3D saliency dataset collected on screen. In *Proceedings of the 2024 Symposium on Eye Tracking Research and Applications*. 1–6.

[46] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. 2024. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 19447–19456.

[47] Mufeng Zhu, Mingju Liu, Cunxi Yu, Cheng-Hsin Hsu, and Yao Liu. 2025. SGSS: Streaming 6-DoF Navigation of Gaussian Splat Scenes. In *Proceedings of the 16th ACM Multimedia Systems Conference*. 46–56.