

Unity based virtual reality for detector and event visualization in JUNO experiment

Authors: Huang, Kai-Xuan, Song, Ms. Tianzi, Mr. Yuning Su, Chengxin Wu, Wang Xuesen, Prof. Yumei Zhang, You, Prof. Zhengyun, I understand the translation requirements and am ready to begin when you provide the Chinese text.

Date: 2025-06-18T00:00:00+00:00

Abstract

Detector and event visualization are crucial components of high-energy physics (HEP) experimental software. Virtual Reality (VR) technologies and multimedia development platforms such as Unity offer enhanced display effects and flexible extensibility for visualization in HEP experiments. In this study, we present a VR-based method for detector and event displays in the Jiangmen Underground Neutrino Observatory (JUNO) experiment. This method shares the same detector geometry descriptions and event data model as those in offline software and provides necessary data conversion interfaces. The VR methodology facilitates an immersive exploration of the virtual environment in JUNO, enabling users to investigate detector geometry, visualize event data, and tune the detector simulation and event reconstruction algorithms. Additionally, this approach supports applications in data monitoring, physics data analysis, and public outreach initiatives.

Full Text

Preamble

Unity-Based Virtual Reality for Detector and Event Visualization in the JUNO Experiment

Kai-Xuan Huang,¹ Tian-Zi Song,¹ Yu-Ning Su,¹ Cheng-Xin Wu,¹ Xue-Sen Wang,² Yu-Mei Zhang,^{2,*} and Zheng-Yun You^{1,†}

¹School of Physics, Sun Yat-sen University, Guangzhou 510275, China

²Sino-French Institute of Nuclear Engineering and Technology, Sun Yat-sen University, Zhuhai 519082, China

Detector and event visualization are crucial components of high-energy physics (HEP) experimental software. Virtual Reality (VR) technologies and multimedia development platforms such as Unity offer enhanced display effects and flexible extensibility for visualization in HEP experiments. In this study, we present a VR-based method for detector and event displays in the Jiangmen Underground Neutrino Observatory (JUNO) experiment. This method shares the same detector geometry descriptions and event data model as those in offline software and provides necessary data conversion interfaces. The VR methodology facilitates an immersive exploration of the virtual environment in JUNO, enabling users to investigate detector geometry, visualize event data, and tune the detector simulation and event reconstruction algorithms. Additionally, this approach supports applications in data monitoring, physics data analysis, and public outreach initiatives.

Keywords: virtual reality, event display, Unity, detector geometry, JUNO

* Yu-Mei Zhang, zhangym26@mail.sysu.edu.cn

† Zheng-Yun You, youzhy5@mail.sysu.edu.cn

Introduction

Visualization techniques are essential in every aspect of modern high-energy physics (HEP) experiments. The “Roadmap for HEP Software and Computing R&D for the 2020s” [1] and the HEP Software Foundation Community White Paper [2] specifically discuss recommendations and guidelines for visualization tools, such as Virtual Reality (VR) technologies [3] in future software development, particularly regarding interactivity, detector geometry visualization, and event display. Compared to traditional visualizations, VR techniques offer a truly immersive perspective, which enhances interactive experience with a better understanding of detector geometry and event information. In recent years, some HEP experiments have developed VR applications for event display and outreach. These include the Belle2VR software [4, 5] for the BelleII experiment [6], the ATLASrift platform [7, 8] for the ATLAS experiment [9], the CMS VR application [10] for the CMS experiment [11], and the Super-KAVE program [12, 13] for the Super-K experiment [14].

The development of VR applications typically involves game engines, such as Unity [15] or Unreal Engine [16]. Unity is a cross-platform engine that supports the development of games, videos, animations, and architectural visualizations. It has been employed for detector visualization and event display in various HEP experiments, including BelleII, BESIII [17], ALICE [18], ATLAS [19], JUNO [20], and the Total Event Visualizer (TEV) of the CERN Media Lab [21], all of which achieve excellent visualization effects.

The Jiangmen Underground Neutrino Observatory (JUNO) [22–24] is situated underground in southern China with a 650-meter rock overburden. The primary scientific goal of JUNO is to determine the neutrino mass hierarchy. Over an approximately seven-year operational period, JUNO is expected to determine the

neutrino mass hierarchy with a significance of 3σ [25], and to measure the neutrino oscillation parameters Δm_{31}^2 , Δm_{21}^2 , and $\sin^2_{12}$, achieving a precision of 0.2% for Δm_{31}^2 , 0.3% for Δm_{21}^2 , and 0.5% for $\sin^2_{12}$ [26, 27], respectively. Additionally, the JUNO experiment is capable of investigating various types of neutrinos, including earth neutrinos, atmospheric neutrinos, solar neutrinos, and supernova neutrinos [22]. Its excellent energy resolution and large fiducial volume provide promising opportunities to explore numerous essential topics in neutrino physics.

In this study, we develop a VR-based event display tool using Unity for JUNO. This software is compatible with various platforms through Head-Mounted Displays (HMDs) [28] and offers functionalities including the VR-based visualization of the JUNO detector, event displays for different types of data, interfaces for reading and converting event data information, and Spatial User Interface (Spatial UI) control features.

The rest of this paper is organized as follows. In Section II, we introduce VR-based software for HEP experiments. Section III describes the software methodologies, including the JUNO VR framework, the data flow of detector geometry and event data conversion, as well as interaction methods with Spatial UI. The visualization of detector units and event data in the VR-based tool is introduced in Section IV. The potential for further applications is discussed in Section V. Finally, the performance of the software is introduced in Section VI.

II. Visualization and VR

A. Unity and VR in HEP Experiments

In HEP experiments, physicists typically develop detector descriptions and event visualization tools within offline software frameworks. These event display tools are usually built upon widely-used HEP software, such as Geant4 [29] or ROOT [30], which provide user-friendly visualization capabilities that facilitate software development. With the upgrades to ROOT and its EVE package [31], the development of event display tools has become more efficient. Several recent HEP experiments, including ALICE, CMS [11], BESIII [32], JUNO [33, 34], and Mu2e [35], adopt ROOT EVE for developing event display software. However, due to the limited visualization technique support in ROOT, its display capabilities do not fully meet the diverse requirements of physicists, and most ROOT applications remain confined to the Linux platform.

To enhance visualization quality, interactivity, and multi-platform support, several event display tools are developed based on external visualization software. Unity is widely applied in the field of HEP, being used in projects including BelleII, BESIII, ALICE, ATLAS, and JUNO. Unity is a professional video and game development engine based on C#, and visualization software built on Unity offers several advantages:

- **Impressive visualization quality.** Unity, a widely adopted professional

3D engine in the industry, offers advanced visual capabilities that surpass those of traditional software used in HEP, such as ROOT. Additionally, its continuous updates enable HEP visualizations to stay aligned with cutting-edge developments in graphics technology.

- **Cross-platform support.** The comprehensive multi-platform support of Unity enables seamless export and deployment of projects across a range of operating systems, including Windows, Linux, macOS, iOS, Android, and web browsers. This functionality ensures that the same visualization project can be accessed across various platforms, minimizing development effort and streamlining maintenance tasks.
- **High-quality VR rendering and performance optimization.** Unity supports modern graphics technologies such as real-time lighting, global illumination, and physics-based rendering. Light behaves according to the principles of physics, including energy conservation and Fresnel reflections [36], resulting in more realistic and immersive graphical effects in VR. These features are crucial for enhancing details like lighting, shadows, textures, and environmental interactions, significantly improving the user's sense of immersion. Additionally, Unity optimizes VR performance by rendering separate images for each eye, providing a dual-eye perspective while maintaining smooth rendering and minimizing motion blur and latency.
- **VR HMDs compatibility.** Unity supports most popular VR HMDs, including Meta Quest 2 and Quest 3 [37], HTC Vive [38], Valve Index [39], and Vision Pro [40]. Through the extended reality interaction toolkit in Unity, developers can easily create interactive applications for various devices without device-specific coding. Additionally, Unity provides a fast turnaround during the development cycle. Projects can be executed immediately, running quickly on VR devices for easier debugging, without the need to compile and link executable files [41].

Compared to 3D-based event visualization software, VR technology significantly enhances the visual experience of the user. VR applications are typically conducted through HMDs. According to Steam VR hardware statistics [42], more than half of users utilize the Meta Quest 2 and Quest 3. These devices, based on the Android operating system, offer sufficient immersion and are widely used across various fields, including gaming, social interaction, and education. Equipped with accelerometers, gyroscopes, and cameras, these devices can track the user's head and hand movements, enabling interaction and navigation within virtual environments. Additionally, the controllers facilitate interaction with Spatial UI in the virtual environment. VR technology provides synthesized sensory feedback, creating a strong sense of immersion and presence within a simulated environment.

Most HEP experiments are typically conducted in underground or restricted areas, which are generally inaccessible during data collection. VR technology enables the public to explore these experiments in an immersive environment to observe detector operations and event data collection. This offers a fundamental

understanding of the types of scientific research being conducted in HEP, which is highly beneficial for both educational and outreach purposes. Furthermore, by simulating particle emissions and their interactions with detectors, VR provides physicists with an immersive platform for refining offline simulations and reconstruction software [43–46]. It can also enhance simulation accuracy. For JUNO, considering the deformation of the stainless steel truss, offsets need to be applied to the PMT positions based on limited survey data [47–49]. Overlap checks and position tuning using the VR event display tool will be particularly helpful. Additionally, VR enables physicists to analyze rare events as though they are physically present within the inner detector environment, which provides an alternative approach for data analysis and may inspire creativity.

B. VR Application in HEP

In recent years, VR applications have been developed in several HEP experiments for event visualization and outreach. These software packages include Belle2VR [5] for the BelleII experiment, ATLASrift [7, 8] for the ATLAS experiment, and Super-KAVE [12, 13] for the Super-K experiment.

Belle2VR is an interactive VR visualization tool developed with Unity, designed to represent subatomic particle physics. This application allows users to explore the BelleII detector and observe particle jets generated in high-energy e^+e^- collisions. The Super-KAVE application immerses users in a scaled representation of the Super-K detector, allowing them to explore the virtual space, switch between event datasets, and change visualization modes [12, 13]. In addition to providing VR modes for exploring the detector and standard event displays, the application features a supernova event visualization technique, simulating the conversion of a star into a supernova. This leads to the occurrence of thousands of neutrino events within approximately ten seconds. It serves as a valuable outreach tool, offering a new example of visualization techniques for various neutrino particle physics applications. ATLASrift, a VR application developed for the ATLAS experiment, is primarily used for data visualization and outreach [9]. Users can move around and inside the detector, as well as explore the entire underground experimental cavern and its associated facilities, including shafts, service halls, passageways, scaffolds, and more.

III. Methodologies

VR technology provides an immersive experience. However, the development of comprehensive event display software utilizing VR for HEP experiments still involves significant challenges. The first challenge is to convert the detector geometry, typically based on Geant4 simulations, into a format such as FBX [50] that can be imported into Unity. Given that detectors usually consist of tens of thousands of components, manually creating the geometry would impose a significant workload. Another significant challenge is extracting and converting event information into a structure compatible with Unity. In HEP experiments, the fundamental information for event display is typically defined

by the offline software and stored in ROOT format. However, because Unity does not support direct reading of ROOT files, a dedicated conversion process is required. Additionally, a bijective mapping is established to link the detector unit identifiers used in the offline software [51] with the names assigned to the corresponding geometries in Unity.

This section introduces the software architecture and data flow in the JUNO VR program. We describe the process of detector geometry conversion, the exchange of essential event information from offline software to Unity, and the strategy of matching detector units. Additionally, we discuss the construction of the Spatial UI and provide an overview of its functionality.

A. Software Structure and Data Flow

The event display software should provide visualization capabilities, including detector geometry, event data information at different levels, and interactive controls. For JUNO VR software visualization, the first step involves converting and importing the detector geometry and event data information into Unity for display, followed by the development of interactive controls. As shown in Figure 1 [Figure 1: see original paper], the JUNO event display software consists of four components:

- **Detector geometry conversion.** The geometric models of the detector are constructed using Geant4 in the detector simulation, initially stored in a Geometry Description Markup Language (GDML) file [52]. The GDML file is then automatically converted to the FBX format using the GDML-FBX conversion tool [17, 53], which is compatible for import into Unity.
- **Event data conversion.** The Event Data Model (EDM) [54] encompasses various types of event information exchanged between different components of JUNO online and offline software, including data acquisition, simulation, calibration, and reconstruction. The event information for JUNO VR event display is extracted from the offline software EDM [55]. By combining the detector identifier and Unity geometry name matching rules, the detector information is remapped, generating event information that Unity can directly import and that conforms to the geometry hierarchy in Unity.
- **Detector and event information visualization.** The detector geometry, simulation, and reconstruction information, as well as the hit information and their associations, are visualized in Unity. By adjusting the material properties and combining Unity's layers, lighting, and rendering effects, an immersive and outstanding visualization experience in VR mode is achieved.
- **Spatial UI and interactive control.** The Spatial UI is designed to facilitate the visualization and interaction with the detector and event information. It includes the sub-detector geometry panel and the event display panel, which allow users to control the display of sub-detectors, switch between event types, and manage the event display process. Inter-

active control is enabled through the Meta Quest 3 controller, with distinct functions assigned to the joystick and various buttons. These functions include controlling the visibility of each panel, navigating within the 3D virtual detector environment, and switching perspectives.

B. Detector Geometry Conversion

The detector geometry in HEP experiments is typically complex, consisting of up to millions of detector units. The description of these detectors is commonly developed using specialized geometric languages, such as GDML and Detector Description for High-Energy Physics (DD4hep) [56, 57]. The JUNO experiment, along with BESIII, PHENIX [58], and LHCb [59], uses GDML to describe and optimize the geometry of detectors for conceptual design and offline software development. GDML is a detector description language based on Extensible Markup Language (XML) [60] that describes detector information through a set of textual tags and attributes, providing a persistent description of the detector. The geometry description files of detectors typically include essential information about the detector model, such as lists of materials, positions, rotations, solids, and the hierarchical structure of the detector.

Since the GDML format does not directly support import into Unity, some previous HEP applications involving Unity typically required manual construction of geometric models. Given that HEP detectors are usually highly complex, the creation of 3D detector models in Unity becomes particularly challenging. However, Unity does support direct import of several 3D file formats, including FBX, DAE [61], DXF [62], and OBJ [63]. Among these, FBX stands out as a widely used 3D asset format, due to its ability to handle intricate scene structures. This includes not only geometry but also animations, materials, textures, lighting, and cameras, which makes it a highly suitable choice for HEP applications involving complex 3D models.

A method that can automatically convert GDML or DD4hep to FBX format is essential for detector construction in Unity. Several researchers have proposed automated methods for converting GDML files to FBX files, significantly facilitating Unity-based development. For instance, the BESIII collaboration group suggests using FreeCAD [64], a 3D CAD and modeling software, in conjunction with CAD data optimization software Pixyz [65], with the STEP [66] format as an intermediate conversion format [17]. The CMS collaboration group employs SketchUp software for auxiliary data conversion [67]. Recently, methods were also proposed to directly convert GDML files to FBX files [53]. This research, based on the above method, enables a fast and automatic conversion process from GDML to FBX, which can be completed in just a few minutes, saving significant time in the conversion process. This approach becomes particularly beneficial during the recent geometric updates of the JUNO detector at the commissioning stage, enabling the swift conversion of the updated FBX file, which includes the latest geometry model of the real detector units after installation.

C. Event Data Conversion

In HEP experiments, the event data is typically stored in files with binary raw data format or ROOT format. ROOT, an efficient data analysis framework, is widely adopted for high-performance data input and output operations. However, since Unity cannot directly read ROOT files, it is necessary to extract the required event information based on the EDM and convert it into a text format that Unity can process. The essential information for event display comprises three main components: detector unit hits, Monte Carlo (MC) truth, and reconstruction data. The detector unit hits include the hit time and hit charge for each detector unit like a PMT. MC truth provides detailed truth information such as simulated vertices and photon trajectories (including 3D coordinates and propagation with time), which facilitate a deeper analysis of particle direction and relative velocity. Reconstruction data typically contain the reconstructed vertex positions, energy information, and additional track information for muon events like direction. Together, this information serves as the foundation for developing event display functionalities and interactive control modules based on Spatial UI.

Furthermore, the identifiers used for detector units in offline software may differ from the names of the geometric objects in Unity. In HEP experiments, the detector identifier system assigns a unique ID to each detector unit and plays a critical role in various applications including data acquisition, simulation, reconstruction, and analysis. Therefore, establishing an accurate mapping between the detector identifiers in offline software and the geometric objects like PMTs in Unity is essential to ensure the accurate display of an event. Based on EDM readout rules and leveraging the mapping between the identifier module and the geometric objects in Unity, an automated readout and conversion interface is developed to export event display information.

D. Spatial UI and Interactive Control

The Spatial UI serves as the interface facilitating interaction between the user and the VR application. For the JUNO VR project, we develop two Spatial UIs: the sub-detector geometry control panel and the event display control panel, as shown in Figure 2 [Figure 2: see original paper].

The sub-detector geometry panel primarily controls the visualization attributes of the geometries of various sub-detectors, including Central Detector (CD) large PMTs, CD small PMTs, Top Tracker, and water pool PMTs. Detailed information about the sub-detectors of JUNO is provided in Section IV A. In addition to the sensitive detectors like PMTs, an “Other structure” toggle controls the display of passive structures, such as the steel structure, the acrylic ball, the PMT support structures, and the liquid filling pipelines. Additionally, the “Data type” drop-down is used to switch between different types of events collected during real data-taking or from simulation. The “Photon trail mode” toggle enables switching of display modes for photon paths, either represented

by green lines or in a manner closely resembling particle motion.

The event display panel is designed to implement the core functionality for event visualization, which includes a toggle for switching display mode between simulation and data types, a slider for controlling the display of an event with its timeline evolution, a drop-down for selecting different types of events, and a button to play the event animation. A “Draw Hit” button initiates the animation of the full event hit process, which plays within a period of time window, with the time slider moving in sync with the event timeline, enabling users to track the current time of the event.

Interactive control is achieved through the use of controllers, gesture operations, eye-tracking, and other input methods in HMDs. The following discussion focuses on testing interactive control for the Meta Quest 3. For other HMDs, the cross-platform support provided by the extended reality interaction toolkit in Unity minimizes development differences between various devices. Simple adaptations based on the specific features of the HMDs are sufficient for operation. The controller buttons resemble a typical gamepad, with the addition of side buttons. The X&Y buttons on the left controller are used to control the visibility of the sub-detector geometry panel. When displayed, the position of this panel is based on the user’s orientation and appears at the front left of the user’s view. Users can drag or hide the panel to avoid obstructing their view when visualizing events. The A&B buttons on the right controller are used to control the visibility of the event display panel. When displayed, the panel appears at the front right of the user’s view. Based on the gyroscope and accelerometer hardware of the Meta Quest 3, these planes are always oriented perpendicular to the user’s view orientation.

For JUNO VR software, multiple types of datasets are provided, including radioactive background, Inverse Beta Decay (IBD) [68], cosmic ray muons, and other types of events. The event display dataset is designed to encompass both simulated and real data event types. Simulated events are produced with the JUNO offline software to facilitate detector simulation, commissioning, and the optimization of reconstruction algorithms. Since JUNO has not yet commenced formal data taking, real data events are instead obtained from the Data Challenge dataset [69], which has data structures identical to those expected during actual operation. With the event data conversion interface, the datasets with various types of data are ready to be displayed in the Unity-based visualization and VR software.

The joystick on the left controller controls the user’s 3D movement, based on both the controller’s input and the user’s view orientation. For example, when the user’s head orientation is directed towards the upper right, pushing the joystick upwards moves the user in the virtual space toward that direction. Figure 3 [Figure 3: see original paper] illustrates the user’s viewpoint during motion in the JUNO VR application. The event depicted is a simulated muon event. Additional details presented in the figure are described in detail in Section IV. The joystick on the right controls the user’s viewpoint direction. Addition-

ally, the user can change their head orientation to switch perspectives. The side button is used for interaction confirmation. Furthermore, when interacting with the Spatial UI, if the controller's laser pointer touches the corresponding component, audio and highlight feedback are provided, making the interaction smoother for user control.

IV. Visualization in JUNO

This section is dedicated to introducing the visualization effects in the JUNO VR application, including detector geometry, hit distribution for different types of events, as well as MC truth information and display of event reconstruction outputs.

A. Detector Units

The schematic design of the JUNO detector is illustrated in Figure 4 [23]. The detector includes the water pool, the CD [47], and the Top Tracker [70]. The CD is the heart of the JUNO experiment and is filled with 20 ktons of liquid scintillator [71, 72] to serve as the target for neutrino detection. The liquid scintillator is housed within a spherical acrylic vessel, which has a thickness of 120 mm and an inner diameter of 35.4 meters. This vessel is supported by a spherical stainless steel structure with an inner diameter of 40.1 meters. To detect photons, the CD is equipped with a total of 17,612 20-inch PMTs and 25,600 3-inch PMTs. Surrounding the CD is the water pool containing 35 ktons of highly purified water, which effectively shields the detector from external radioactivity originating from the surrounding rocks. The water pool is also instrumental in vetoing cosmic ray muons, with 2,400 20-inch PMTs deployed as part of the water Cherenkov detector. The Top Tracker, located at the top of the water pool, plays a key role in measuring and vetoing muon tracks [73, 74].

As described in Section III B, the JUNO detector geometry is converted from the GDML file, and matched between the identifier module and Unity geometry for each detector unit. The visualization effects of the whole JUNO detector in the VR application are shown in Figure 5 [Figure 5: see original paper]. The light blue cylindrical structure represents the water pool, with the water pool PMTs positioned outward, as indicated by the yellow portion of the spherical structure. At the top of the water pool, the reddish-brown structure represents the Top Tracker detector. From the interior view in the JUNO VR, the spherical acrylic vessel is shown in light gray, as depicted in Figure 2, although it is close to fully transparent in reality to allow more photons to pass through. Surrounding this vessel is the stainless steel structure, shown as dark gray in Figure 5. The CD detector PMTs, oriented toward the center of the sphere, are designed such that only the white tail structures of every PMT are visible in Figure 5.

Owing to the hardware capabilities of the Meta Quest 3, there is no requirement to optimize the grid of detector units or replace them with simplified geomet-

ric shapes. Most of the geometric details of the detector units are preserved, achieving effects that are difficult to accomplish in event displays based on ROOT. Additionally, for the detector units, in order to more closely replicate the effect of real PMTs, we have assigned different material properties to the detector units, including visualization attributes such as color, reflectivity, and metallicity, to achieve the best display effect.

B. MC Simulation Event Display

MC simulation is crucial for detector design and assists physicists in evaluating the detector's performance and tuning reconstruction algorithms. There are various kinds of signal and background events in JUNO, while currently we primarily focus on the radioactive backgrounds, IBD signals, and muon events.

The IBD event, $\bar{\nu}_e + p \rightarrow e^+ + n$, is the major signal event for detecting electron anti-neutrinos in the JUNO experiment [22, 23]. JUNO identifies and reconstructs IBD events by detecting signals from positron and neutron captures. This dual-signal characteristic helps effectively identify anti-neutrino signal events while suppressing the huge background events. For the IBD event, there are both positron and neutron signals, whose photon paths are displayed in green and red, respectively, as shown in Figure 6 [Figure 6: see original paper]. The detector units that are triggered are color-coded from cyan to dark blue based on the number of hits in the event, with bluer colors indicating a higher number of hits. The PMTs that are not triggered are displayed in yellow by default. Furthermore, in the time evolution of an event, the color of fired PMTs changes with time according to the associated timing information. The neutron-induced photon paths are delayed by approximately 170 μs relative to those from the positron, and this delay can be visualized using the time slider in the JUNO VR environment.

One major background event type is the cosmic-ray muon events. Muons are secondary particles produced by high-energy cosmic rays in the earth's atmosphere and possess strong penetrating power. Despite JUNO being located 650 m deep underground, a small fraction of muons can still penetrate the overlying shielding and enter the detector, generating muon events. Figure 7 [Figure 7: see original paper] presents the event information for the simulated muon event. Photon trajectories are represented by light green lines. These paths gradually extend over time, depicting the propagation of photons. In the simulated event shown, the directions of these photon paths may change, indicating their interactions with the detector materials. For the muon event, as a muon penetrates the detector, it continuously produces photons while depositing its energy in the liquid scintillator.

Event reconstruction plays a key role in JUNO data processing, reconstructing the vertex and energy of an event, which is essential in determining the neutrino mass hierarchy. For point-like events like IBD signals, almost all the photon paths originate from the same event vertex. Figure 8 [Figure 8: see original

paper] shows the reconstructed vertex and the MC truth. The initial particle production vertex (red sphere), derived from MC truth, indicates where the positron is created. The weighted energy deposit vertex (green sphere) marks the positron's annihilation point in the liquid scintillator. The reconstructed vertex (purple sphere) is produced by the event reconstruction algorithm. The reconstruction bias (light yellow line) represents the discrepancy between the reconstructed vertex and the energy deposit vertex. A shorter distance indicates a more accurate reconstructed vertex. In the ideal scenario, the reconstructed vertex will converge to the true vertex.

C. Real Data Event Display

For the real-data event, we utilize the Data Challenge dataset [69], whose data structures and processing pipeline are identical to those employed during data taking. This ensures that the software will function seamlessly once the experiment enters formal operation. The event composition in this dataset is the same as that in the MC simulation, encompassing radioactive-background events, IBD signals, and muon events.

Figure 9 [Figure 9: see original paper] presents event information for a muon event derived from the real data events. The reconstructed muon travels through the detector along the magenta line. The left and right sides represent the reconstructed incident and exit points of the muon. A time offset is established by dividing the track distance by the speed of light. Users can observe the trajectory of the muon through the Spatial UI. Since the exact point of photon emission along the path cannot be determined, photon information is not displayed in this mode. Using the reconstructed hit time, the corresponding point on the trajectory is linked to the relevant PMT unit. Once the photon particles arrive at the PMT units, those triggered PMTs will change color accordingly.

Moreover, by exploiting Unity's robust visualization capabilities, a specialized mode is developed to simulate photon paths using particle-like effects instead of simple line trajectories to display the propagation of particles more realistically.

V. Applications

The JUNO VR software provides an immersive interactive experience, allowing users to intuitively understand the detector structure and event information. Some features and applications of the visualization software are listed below.

Data quality monitoring. The data quality monitoring system [75–78] is designed to identify data issues promptly, ensuring the acquisition of high-quality data. During the future data-taking phase, event information can be real-time and automatically extracted from the reconstructed files from the data quality monitoring system. Based on Unity-supported databases, such as SQLite, event information can be transmitted from the data quality monitoring server to JUNO VR software. This enables immersive visualization of detector operation status and event information during the data-taking phase. For example,

an animation of a real-time data-acquisition event is automatically played every 30 seconds. Through immersive visualization, shifters can easily monitor anomalies, such as hot or dead PMT channels.

Physics analysis. Physical analysis involves in-depth research of neutrino events to extract physical parameters, validate theoretical models, search for rare signals, and uncover new phenomena. This requires detailed analysis of large volumes of complex data. Through the VR interface, researchers can reconstruct an immersive view of the event in three-dimensional space, allowing them to freely explore the data, observe event details from multiple perspectives, and identify potential patterns and anomalies.

Outreach. For HEP experiments, their complex theoretical and experimental contents are usually difficult for the public and students to understand. Based on the VR application, students can understand the structure of the JUNO detector and the processing of signal and background events through interactive operations, thereby enhancing engagement and understanding of the physics and principles of HEP experiments. Visualization programs including VR stand out in the field of education and public outreach. Due to Unity's cross-platform support and compatibility with various HMDs, the completed project can be exported to different platforms and utilized with different HMDs, meeting the requirements of various outreach scenarios.

VI. Performance

In experimental evaluations conducted on the mainstream VR device—the Meta Quest 3, the JUNO VR application is capable of processing a variety of event types and demonstrates sufficient computational performance. During testing, the device's CPU utilization remains below 70%, GPU utilization remains below 40%, and the display maintains a stable refresh rate of 72 frames per second. The software's interactive response primarily depends on the event type. For muon events, which contain a larger volume of hit information, the latency when switching between events is approximately 3 seconds; for IBD and radioactive background events, it is approximately 1 second. The event display of the JUNO VR application undergoes rigorous testing, and the application is capable of processing both simulated and real data events.

VII. Summary

VR technology greatly enhances the visualization effects of HEP experiments. A JUNO VR application for detector and event visualization is developed using Unity. By converting GDML to FBX format, efficient construction of the complex detector geometry in Unity is achieved. An event data conversion interface is created based on matching the detector identifier module and the detector geometry hierarchy in Unity. Through the Spatial UIs, users can easily control the display of various subsystems for detector and event visualization.

With the ongoing construction of the JUNO experiment, the VR event display software is successfully developed, and more features are expected to be added in future updates. VR technology offers an immersive, interactive experience, and it holds great potential in areas such as offline software development, data taking, physics analysis, education, and public outreach.

List of Abbreviations

HEP - High-energy Physics
VR - Virtual Reality
JUNO - Jiangmen Underground Neutrino Observatory
TEV - the Total Event Visualizer
HMDs - Head-Mounted Displays
Spatial UI - Spatial User Interface
GDML - Geometry Description Markup Language
EDM - Event Data Model
DD4hep - Detector Description for High-Energy Physics
XML - Extensible Markup Language
MC - Monte Carlo
IBD - Inverse Beta Decay
CD - Central Detector

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 12175321, W2443004, 11975021, 11675275, and U1932101), Strategic Priority Research Program of the Chinese Academy of Sciences (No. XDA10010900), National Key Research and Development Program of China (Nos. 2023YFA1606000 and 2020YFA0406400), National College Students Science and Technology Innovation Project, and Undergraduate Base Scientific Research Project of Sun Yat-sen University.

Kai-Xuan Huang and Tian-Zi Song contributed equally to this work.

References

- [1] HEP Software Foundation Collaboration. “A Roadmap for HEP Software and Computing R&D for the 2020s” . Comput. Softw. Big Sci. 3.1 (2019), p. 7. DOI: 10.1007/s41781-018-0018-8.
- [2] M. Bellis et al. “HEP Software Foundation Community White Paper Working Group -Visualization” (2018). DOI: 10.48550/ARXIV.1811.10309.
- [3] I. Wohlgenannt, A. Simons, and S. Stieglitz. “Virtual reality” . Business & Information Systems Engineering 62 (2020), pp. 455–461. DOI: 10.1007/s12599-020-00000-0.
- [4] M. Bender, T. Kuhr, and L. Piilonen. “Belle II virtual reality projects” . EPJ Web Conf. 214 (2019). Ed. by A. Forti, L. Betev, M. Litmaath, et al., p. 02028. DOI: 10.1051/epjconf/201921402028.
- [5] Z. Duer, L. Piilonen, and G. Glasson. “Belle2VR: A Virtual-Reality

- Visualization of Subatomic Particle Physics in the Belle II Experiment” . IEEE Computer Graphics and Applications 38.3 (2018), pp. 33-43. DOI: 10.1109/MCG.2018.032421652.
- [6] Belle-II Collaboration. “Belle II Technical Design Report” (2010). DOI: 10.48550/arXiv.1011.
- [7] ATLAS Collaboration. “Virtual Reality and game engines for interactive data visualization and event displays in HEP, an example from the ATLAS experiment” . EPJ Web Conf. 214 (2019). Ed. by A. Forti, L. Betev, M. Litmaath, et al., p. 02013. DOI: 10.1051/epjconf/201921402013.
- [8] I. Vukotic, E. Moyse, and R. M. Bianchi. “ATLASrift - a Virtual Reality application” . Meeting of the APS Division of Particles and Fields. 2015. DOI: 10.48550/arXiv.1511.00047.
- [9] ATLAS Collaboration. “The ATLAS Experiment at the CERN Large Hadron Collider” . JINST 3 (2008), S08003. DOI: 10.1088/1748-0221/3/08/S08003.
- [10] CMS Collaboration. Leveraging Virtual Reality for Visualising the CMS Detector. PoS (ICHEP2024) 1171. Available at: <https://pos.sissa.it/476/1171/>. Accessed on: June 16, 2025.
- [11] CMS Collaboration. “The CMS Experiment at CERN LHC” . JINST 3 (2008), S08004. DOI: 10.1088/1748-0221/3/08/S08004.
- [12] B. Izatt, K. Scholberg, and R. P. McMahan. “Super-KAVE: An immersive visualization tool for neutrino physics” . 2013 IEEE Virtual Reality (VR). 2013, pp. 75-76. DOI: 10.1109/VR.2013.6549370.
- [13] E. Izatt, K. Scholberg, and R. Kopper. “Neutrino-KAVE: An immersive visualization and fitting tool for neutrino physics education” . 2014 IEEE Virtual Reality (VR). 2014, pp. 83-84. DOI: 10.1109/VR.2014.
- [14] Y. Suzuki. “The Super-Kamiokande experiment” . Eur. Phys. J. C 79.4 (2019), p. 298. DOI: 10.1140/epjc/s10052-019-6796-2.
- [15] W. Goldstone. Unity game development essentials. Packt Publishing Ltd, 2009.
- [16] A. Sanders. An introduction to Unreal engine 4. AK Peters/CRC Press, 2016.
- [17] K.-X. Huang, Z.-J. Li, Z. Qian, et al. “Method for detector description transformation to Unity and application in BESIII” . Nucl. Sci. Tech. 33.11 (2022), p. 142. DOI: 10.1007/s41365-022-01133-8.
- [18] ALICE Collaboration. “ALICE: Physics performance report, volume I” . J. Phys. G 30 (2004). Ed. by F. Carminati, P. Foka, P. Giubellino, et al., pp. 1517-1763. DOI: 10.1088/0954-3899/30/11/001.
- [19] J. Pequeno. CAMELIA webpage. Available at: <https://pdgusers.lbl.gov/~pequeno/camelia>. Accessed on: March 15, 2025.
- [20] J. Zhu, Z.-Y. You, Y.-M. Zhang, et al. “A method of detector and event visualization with Unity in JUNO” . JINST 14.01 (2019), T01007. DOI: 10.1088/1748-0221/14/01/T01007.
- [21] C. M. Lab. CERN TEV visualization framework webpage. Available at: <https://gitlab.cern.ch/CERNMediaLab/>. Accessed on: March 15, 2025.
- [22] JUNO Collaboration. “JUNO physics and detector” . Prog. Part. Nucl. Phys. 123 (2022), p. 103927. DOI: 10.1016/j.pnpnp.2021.103927.

- [23] JUNO Collaboration. “JUNO Conceptual Design Report” (2015). DOI: 10.48550/arXiv.1508.
- [24] F. An et al. “Neutrino Physics with JUNO” . J. Phys. G 43.3 (2016), p. 030401. DOI: 10.1088/0954-3899/43/3/030401.
- [25] A. Abusleme et al. “Potential to identify neutrino mass ordering with reactor antineutrinos at JUNO” . Chin. Phys. C 49.3 (2025), p. 033104. DOI: 10.1088/1674-1137/ad7f3e.
- [26] JUNO Collaboration. “Sub-percent precision measurement of neutrino oscillation parameters with JUNO” . Chin. Phys. C 46.12 (2022), p. 123001. DOI: 10.1088/1674-1137/ac8bc9.
- [27] J. P. Athayde Marcondes de André, N. Chau, M. Dracos, et al. “Neutrino mass ordering determination through combined analysis with JUNO and KM3NeT/ORCA” . Nucl. Instrum. Meth. A 1055 (2023), p. 168438. DOI: 10.1016/j.nima.2023.
- [28] J. E. Melzer and K. Moffitt. Head mounted displays.
- [29] GEANT4 Collaboration. “GEANT4 - A Simulation Toolkit” . Nucl. Instrum. Meth. A 506 (2003), pp. 250–303. DOI: 10.1016/S0168-9002(03)01368-8.
- [30] R. Brun, A. Gheata, and M. Gheata. “The ROOT geometry package” . Nucl. Instrum. Meth. A 502 (2003). Ed. by V. A. Ilyin, V. V. Korenkov, and D. Perret-Gallix, pp. 676–680. DOI: 10.1016/S0168-9002(03)
- [31] M. Tadel. “Overview of EVE: The event visualization environment of ROOT” . J. Phys. Conf. Ser. 219 (2010). Ed. by J. Gruntorad and M. Lokajicek, p. 042055. DOI: 10.1088/1742-6596/219/4/042055.
- [32] Z.-J. Li, M.-K. Yuan, Y.-X. Song, et al. “Visualization for physics analysis improvement and applications in BESIII” . Front. Phys. (Beijing) 19.6 (2024), p. 64201. DOI: 10.1007/s11467-024-1422-7.
- [33] Z.-Y. You, K.-J. Li, Y.-M. Zhang, et al. “A ROOT Based Event Display Software for JUNO” . JINST 13.02 (2018), T02002. DOI: 10.1088/1748-0221/13/02/T02002.
- [34] M.-H. Liao, K.-X. Huang, Y.-M. Zhang, et al. “A ROOT-based detector geometry and event visualization system for JUNO-TAO” . Nucl. Sci. Tech. 36.3 (2025), p. 39. DOI: 10.1007/s41365-024-01604-0.
- [35] Mu2e Collaboration. “Mu2e Technical Design Report” (2014). DOI: 10.2172/1172555.
- [36] Unity Technologies. Standard Shader. Available at: <https://docs.unity3d.com/2023.2/Documentation/MarStandardShader.html>. Accessed on: March 15,
- [37] M. Aros, C. L. Tyger, and B. S. Chaparro. “Unraveling the Meta Quest 3: An Out-of-Box Experience of the Future of Mixed Reality Headsets” . HCI International 2024 Posters. Cham: Springer Nature Switzerland, 2024, pp. 3–8. DOI: 10.1007/978-3-031-61950-2_1.
- [38] HTC Corporation. HTC Vive Official Website. Available at: <https://www.vive.com>. Accessed on: March 15, 2025.
- [39] V. Corporation. Valve Index Official Website. Available at: <https://www.valvesoftware.com/en/index>. Accessed on: March 15, 2025.
- [40] R.-Z. Cheng, N. Wu, M. Varvello, et al. “A First Look at Immersive Telepresence on Apple Vision Pro” . Proceedings of the 2024 ACM on Internet

- Measurement Conference. IMC '24. Madrid, Spain: Association for Computing Machinery, 2024, pp. 555–562. DOI: 10.1145/3646547.3689006.
- [41] Unity Technologies. Unity User Manual. Available at: <https://docs.unity3d.com/Manual/index.html>. Accessed on: March 15, 2025.
- [42] V. Corporation. Steam Hardware & Software Survey. Available at: <https://store.steampowered.com/hwsurvey>. Accessed on: March 15, 2025.
- [43] G.-H. Huang et al. “Improving the energy uniformity for large liquid scintillator detectors” . Nucl. Instrum. Meth. A 1001 (2021), p. 165287. DOI: 10.1016/j.nima.2021.165287.
- [44] Z.-Y. Li et al. “Event vertex and time reconstruction in large-volume liquid scintillator detectors” . Nucl. Sci. Tech. 32.5 (2021), p. 49. DOI: 10.1007/s41365-021-00885-z.
- [45] Z. Qian et al. “Vertex and energy reconstruction in JUNO with machine learning methods” . Nucl. Instrum. Meth. A 1010 (2021), p. 165527. DOI: 10.1016/j.nima.2021.165527.
- [46] Z.-Y. Li, Z. Qian, J.-H. He, et al. “Improvement of machine learning-based vertex reconstruction for large liquid scintillator detectors with multiple types of PMTs” . Nucl. Sci. Tech. 33.7 (2022), p. 93. DOI: 10.1007/s41365-022-01078-y.
- [47] JUNO Collaboration. “The design and technology development of the JUNO central detector” . Eur. Phys. J. Plus 139.12 (2024), p. 1128. DOI: 10.1140/epjp/s13360-024-05830-8.
- [48] T. Lin et al. “Simulation software of the JUNO experiment” . Eur. Phys. J. C 83.5 (2023). [Erratum: Eur.Phys.J.C 83, 660 (2023)], p. 382. DOI: 10.1140/epjc/s10052-023-11514-x.
- [49] Z. Deng. “Status of JUNO Simulation Software” . EPJ Web Conf. 245 (2020). Ed. by C. Doglioni, D. Kim, G. A. Stewart, et al., p. 02022. DOI: 10.1051/epjconf/202024502022.
- [50] Autodesk. FBX webpage. Available at: <https://www.autodesk.com/products/fbx/overview>. Accessed on: March 15, 2025.
- [51] C.-X. Wu and Z.-Y. You. “Detector identifier and geometry management system in JUNO experiment” . PoS ICHEP2024 (2025), p. 1049. DOI: 10.22323/1.
- [52] R. Chytrcek, J. McCormick, W. Pokorski, et al. “Geometry description markup language for physics simulation and analysis applications.” IEEE Trans. Nucl. Sci. 53 (2006), p. 2892. DOI: 10.1109/TNS.2006.
- [53] A. Iusupova and S. Nemnyugin. “Geometry import into virtual reality visualization engine for HEP experiments at BM@N” . Nucl. Instrum. Meth. A 1067 (2024), p. 169619. DOI: 10.1016/j.nima.2024.
- [54] T. Li, X. Xia, X.-T. Huang, et al. “Design and Development of JUNO Event Data Model” . Chin. Phys. C 41.6 (2017), p. 066201. DOI: 10.1088/1674-1137/41/6/066201.
- [55] JUNO Collaboration. “Modern Software Development for JUNO offline software” . EPJ Web Conf. 295 (2024), p. 05015. DOI: 10.1051/epjconf/
- [56] M. Frank, F. Gaede, C. Grefe, et al. “DD4hep: A Detector Description Toolkit for High Energy Physics Experiments” . J. Phys. Conf. Ser. 513 (2014). Ed. by D. L. Groep and D. Bonacorsi, p. 022010. DOI: 10.1088/1742-

6596/513/2/022010.

- [57] Z.-Y. Yuan et al. “Method for detector description conversion from DD4hep to Filmbox” . Nucl. Sci. Tech. 35.9 (2024), p. 146. DOI: 10.1007/s41365-024-
- [58] PHENIX Collaboration. “PHENIX detector overview” . Nucl. Instrum. Meth. A 499 (2003), pp. 469–479. DOI: 10.1016/S0168-9002(02)01950-2.
- [59] LHCb Collaboration. “The LHCb Detector at LHC” . JINST 3 (2008), S08005. DOI: 10.1088/1748-0221/3/08/S08005.
- [60] T. Bray, J. Paoli, and C. Sperberg-McQueen. Extensible Markup Language (XML) 1.0. Available at: <http://www.w3.org/XML/1998/06/xmlspec-report-19980910.htm>. Accessed on: March 15,
- [61] K. Group. COLLADA Document Schema and Reference (Version 1.5). Available at: <https://www.khronos.org/collada/>. Accessed on: March 15,
- [62] Autodesk. Drawing Exchange Format (DXF) Reference. Available at: <https://archive.ph/20121206003818/http://www.autodesk.com/dxf>. Accessed on: March 15, 2025.
- [63] M. Reddy. Wavefront OBJ File Format. Available at: <http://www.martinreddy.net/gfx/3d/OBJ.spec>. Accessed on: March 15, 2025.
- [64] F. Developers. FreeCAD webpage. Available at: <https://www.freecadweb.org>. Accessed on: March 15, 2025.
- [65] P. Software. Pixyz Studio Software. Available at: <https://www.pixyz-software.com/studio>. Accessed on: March 15, 2025.
- [66] S. Kemmerer. STEP: The Grand Experience. en. 1999. DOI: 10.6028/NIST.SP.939.
- [67] T. Sakuma and T. McCauley. “Detector and Event Visualization with SketchUp at the CMS Experiment” . J. Phys. Conf. Ser. 513 (2014). Ed. by D. L. Groep and D. Bonacorsi, p. 022032. DOI: 10.1088/1742-6596/513/2/022032.
- [68] P. Vogel and J. F. Beacom. “Angular distribution of neutron inverse beta decay, anti-neutrino(e) + p $\rightarrow$ e+ + n” . Phys. Rev. D 60 (1999), p. 053003. DOI: 10.1103/PhysRevD.60.053003.
- [69] T. Lin and W.-Q. Yin. “Offline data processing in the First JUNO Data Challenge” (2024). DOI: 10.48550/arXiv.2408.00959.
- [70] JUNO Collaboration. “The JUNO experiment Top Tracker” . Nucl. Instrum. Meth. A 1057 (2023), p. 168680. DOI: 10.1016/j.nima.2023.
- [71] M. Yu, W.-J. Wu, Y.-Y. Ding, et al. “A Monte Carlo method for Rayleigh scattering in liquid detectors” . Rev. Sci. Instrum. 93.11 (2022), p. 113102. DOI: 10.1063/5.0119224.
- [72] M. Yu, W.-J. Wu, N. Peng, et al. “Measurements of Rayleigh ratios in linear alkylbenzene” . Rev. Sci. Instrum. 93.6 (2022), p. 063106. DOI: 10.1063/5.
- [73] K. Li, Z. You, Y. Zhang, et al. “GDML based geometry management system for offline software in JUNO” . Nucl. Instrum. Meth. A 908 (2018), pp. 43–48. DOI: 10.1016/j.nima.2018.08.008.
- [74] S. Zhang, J.-S. Li, Y.-J. Su, et al. “A method for sharing dynamic geometry information in studies on liquid-based detectors” . Nucl. Sci. Tech. 32.2 (2021), p. 21. DOI: 10.1007/s41365-021-00852-8.
- [75] ATLAS collaboration. “ATLAS offline data quality monitoring” . J. Phys.

Conf. Ser. 219 (2010). Ed. by J. Gruntorad and M. Lokajicek, p. 042018. DOI: 10.1088/1742-6596/219/4/042018.

[76] CMS collaboration. “CMS data quality monitoring: Systems and experiences” . J. Phys. Conf. Ser. 219 (2010). Ed. by J. Gruntorad and M. Lokajicek, p. 072020. DOI: 10.1088/1742-6596/219/7/

[77] J.-F. Hu, Y.-H. Zheng, X.-D. Sun, et al. “A data quality monitoring software framework for the BESIII experiment” . Chin. Phys. C 36 (2012), pp. 62–66. DOI: 10.1088/1674-1137/36/1/010.

[78] Daya Bay Collaboration. “Onsite data processing and monitoring for the Daya Bay Experiment” . Chin. Phys. C 38 (2014), p. 086001. DOI: 10.1088/1674-1137/38/8/086001.

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv –Machine translation. Verify with original.