An event visualization software based on Phoenix for the CEPC experiment

Authors: Zeng, Yujie, Song, Ms. Tianzi, Xuesen Wang, Zhang, Prof. Yumei, Dr. Tao Lin, You, Prof. Zhengyun, You, Prof. Zhengyun

Date: 2025-08-11T00:52:22+00:00

Abstract

In high-energy physics~(HEP) experiments, visualization software plays a pivotal role in detector design, offline software development, and event data analysis. The visualization tools integrate detailed detector geometry with complex event data models, providing the researchers with invaluable insights into experimental results. Phoenix is an emerging general-purpose visualization platform for the current and next-generation HEP experiments. In this study, we develop an event display software based on Phoenix for the CEPC experiment. It offers necessary functionalities for visualizing detector geometries and displaying event data, allowing the researchers to optimize detector design, test simulation and reconstruction algorithms, and analyze event data in a visualized way. Additionally, we discuss the future applications of the event display software, including its usage in online monitoring and the potential to build virtual reality projects for enhanced data visualization.

Full Text

Preamble

An Event Visualization Software Based on Phoenix for the CEPC Experiment

Yu-Jie Zeng,[1] Tian-Zi Song,[1] Xue-Sen Wang,[2] Yu-Mei Zhang,[2],∗ Tao Lin,[3],† and Zheng-Yun You[1],‡
[1]School of Physics, Sun Yat-Sen University, Guangzhou 510275, China
[2]Sino-French Institute of Nuclear Engineering and Technology, Sun Yat-Sen University, Zhuhai 519082, China
[3]Institute of High Energy Physics, Chinese Academy of Sciences, Beijing 100049, China

In high-energy physics (HEP) experiments, visualization software plays a pivotal role in detector design, offline software development, and event data analysis. These tools integrate detailed detector geometry with complex event data models, providing researchers with invaluable insights into experimental results. Phoenix is an emerging general-purpose visualization platform for current and next-generation HEP experiments. In this study, we develop an event display software based on Phoenix for the CEPC experiment. It offers necessary functionalities for visualizing detector geometries and displaying event data, allowing researchers to optimize detector design, test simulation and reconstruction algorithms, and analyze event data in a visualized manner. Additionally, we discuss future applications of the event display software, including its usage in online monitoring and the potential to build virtual reality projects for enhanced data visualization.

Keywords: Visualization, Event display, Phoenix, CEPC

Introduction

The Circular Electron Positron Collider (CEPC) is an international scientific facility proposed by the Chinese particle physics community in 2012 [1]. Designed to explore fundamental physics, particularly as a Higgs factory, the CEPC will be situated in China within a circular underground tunnel approximately 100 kilometers in circumference. This double-ring collider will circulate electron and positron beams in opposite directions through separate beam pipes, with detectors installed at two interaction points.

The CEPC is designed to operate in three distinct modes: H mode (e+e− → ZH), Z mode (e+e− → Z), and W mode (e+e− → W+W−). These modes correspond to center-of-mass energies of 240 GeV, 91 GeV, and 160 GeV, respectively [2, 3]. It will serve as an advanced factory of Higgs, Z, W bosons and top quarks, enabling precision measurements of the Higgs boson properties [4] along with those of the mediators of the weak interaction, the W and Z bosons [5, 6]. It also provides an opportunity to search for physics beyond the standard model (BSM), allowing scientists to discover new physics and unknown phenomena [7].

As the CEPC project progresses [8], a series of software tools must be developed, with event display software being an indispensable component [9]. In modern HEP experiments, detector structures are becoming increasingly complex to achieve higher measurement accuracy and detection efficiency. This necessitates sophisticated visualization tools to aid scientists in understanding detector structures, analyzing physical events, and optimizing simulation and reconstruction algorithms [10–12].

To develop event visualization tools for HEP experiments, the ROOT software [13] and its EVE package [14] have been widely used. However, as a native C++ application, ROOT lacks advanced web-based features such as cross-platform compatibility and ease of deployment. The JSROOT [15] package, a web-based extension of ROOT, addresses many of these limitations by enabling browserside visualization through WebGL and JSON data formats. However, there remains a need for a unified framework that integrates modern 3D rendering capabilities with user-friendly interfaces. Phoenix is a web-based event display framework specifically designed for HEP experiments. It is lightweight and highly cross-platform, built on top of JSROOT and the three.js library [16]. With these components, Phoenix can provide detailed 3D display of detector geometry and events along with a user-friendly interface, making it an excellent choice for developing visualization tools for CEPC. Therefore, in this study we develop an event display software for CEPC using the Phoenix framework. The software not only provides basic visualization functions for detectors and events, but also includes a variety of additional features, allowing users to change visualization effects through a graphical user interface (GUI) and examine tracks, hits, and Monte Carlo (MC) information. Hence, the Phoenix-based event display software will be an effective tool for detector design, offline software development, and physics analysis in the CEPC experiment.

The remainder of this paper is structured as follows. In Section II, we introduce the Phoenix framework for visualization in HEP experiments. In Section III, we present the structure and data flow of the CEPC event display based on Phoenix. In Section IV, we describe the visualization features of the CEPC event display. Applications are discussed in Section V. Finally, Section VI provides a summary.

II. Phoenix Framework

Phoenix is a web-based event visualization framework for HEP experiments [17]. It is based on the TypeScript language and Angular.js framework [18], and uses the popular three.js library [16] for 3D support. In 2017, the HEP Software Foundation (HSF) visualization white paper called for a common event format and common tools to aid visualization [19], and the Phoenix framework was adopted as an extensive framework for event and geometry display.

The Phoenix framework currently supports built-in geometries and event data for ATLAS [20], CMS [21], LHCb [22], and TrackML [23]. Several demonstrations have also been created for FASER [24], FCC-ee, and FCC-hh [25].

Compared to native applications built on ROOT software and its EVE package, the Phoenix framework offers several advantages. First, it is highly cross-platform. Unlike other event display software built as native C++ applications, Phoenix is a web-based framework that can run on any modern web browser. This framework is built as a web application and can be opened simply by visiting a URL. Second, it provides high-level support for HEP experiments. As an event visualization framework for HEP experiments, Phoenix supports visualizing various types of physics objects, including tracks, hits, calorimeter cells, vertices, compound objects, and missing energy. Phoenix has also integrated the JSROOT [15] package, which is part of the ROOT framework, to provide support for visualizing 3D ROOT geometry. In addition to the ROOT format, displaying 3D geometries like OBJ [26], glTF [27], and JSON are also supported. Third, it is extensively designed as experiment-agnostic, aiming to support as many experiments as possible. It provides a standard interface to create a visualization application, thus adding a new experiment demonstration only requires the developer to create a simple extension. Fourth, Phoenix offers rich user interface options, including labeling functionality for physics objects, shareable URLs for predefined events and configurations useful for outreach activities or physics briefings, and built-in animations that make event displays interactive and engaging. All these configurations can be saved, reloaded, applied by default, or passed through URL parameters.

Therefore, the Phoenix framework is a competitive choice for developing event visualization tools. In fact, Phoenix has become the official web event display for ATLAS and is also used by FCC, LHCb, and Belle II experiments [28].

III. Methodologies

An event display software for HEP experiments should provide detailed visualization of detector structure and event data, featuring the ability to show important information such as detector hits, showers, reconstructed tracks, event information, and a GUI to control the display. To achieve this, the CEPC event display software adopts the Phoenix framework and uses the CEPC offline software [29] to provide detector geometry and event data.

The structure of the Phoenix-based CEPC event display software is shown in Fig. 1. The workflow can be divided into the following steps. First, the CEPC offline software provides detector geometries in DD4hep [30, 31] compact XML format. It also generates event data through simulation or reconstruction and stores them in EDM4hep [32, 33] format. DD4hep and EDM4hep are both parts of Key4hep, a framework providing all necessary ingredients for future HEP experiments [34, 35]. Second, since the Phoenix framework does not directly support DD4hep XML format geometries and EDM4hep format event data, a series of conversions are needed [36, 37]. The geometries are converted to ROOT format using the ROOT software, and the event data is converted to JSON format while preserving essential information for visualization. Finally, the ROOT geometry files and event data are loaded into the web display tool, rendering the detector and events while essential information for analysis like particle energy or tracker hits is also displayed in the user interface (UI).

A simple but practical GUI is available for users to control the workflow. Fig. 2 shows the user interface of the event display software. The right corner of the figure shows the Phoenix menu, which can be used to control the visibility of each detector object; the bottom of the figure shows the Phoenix icon-bar, which can be used to upload or export files, change display modes, and enable or disable other high-level features. Users can also use GUI controls to upload geometry and event data files for visualization, change display effects, and choose which parts of the event or detector to visualize. The CEPC and Phoenix logos are displayed on the left and top, and users can visit the CEPC and Phoenix homepages by clicking the logos.

IV. Visualization

A. Visualization of Detector

To enable visualization of the CEPC detector within the Phoenix framework, a series of conversion steps are undertaken [38]. Initially, the detector geometries are stored in DD4hep XML description within the CEPC offline software. By running Geant4 [39] simulations, the detector geometries are loaded to perform simulations, and with the G4GDMLParser module provided by the Geant4 toolkit, the geometries can be exported from the simulation in Geometry Description Markup Language (GDML) format [40, 41]. This step ensures that all geometric details of the detector components are accurately captured and preserved.

Subsequently, these GDML geometries are converted into ROOT format using the geometry export functionalities provided by the ROOT framework. This conversion is necessary because Phoenix does not natively support GDML as a detector input format. The conversion process from DD4hep XML format to ROOT format retains the full geometric complexity of the detector, with no explicit simplification applied to the original design. Although the ROOT format internally represents geometry using Constructive Solid Geometry (CSG), the JSROOT package automatically converts this CSG-based geometry into surface-based representations for rendering. To manage performance during 3D visualization, JSROOT also allows setting upper limits on the number of visible nodes and surfaces, effectively mitigating potential rendering bottlenecks. As a result, even highly complex detector models can be visualized interactively without significant performance degradation.

In the context of CEPC detector visualization within the event display software, the detector is composed of several key components, including but not limited to the Vertex Detector (VXD) [42], the Tracker [43, 44], the Electromagnetic Calorimeter (ECAL) [45], the Hadron Calorimeter (HCAL) [46], and the Muon Detector (Muon). The geometry of each component is individually generated and loaded separately into Phoenix, allowing users to selectively visualize specific parts of the detector through the Phoenix menu.

With the geometries loaded into the Phoenix framework, they are immediately visualized in the software, and users can freely change their perspective, zoom in or out of the 3D model, and clip the geometries to observe details inside the detector. As the CEPC detector is still under design, multiple versions of geometry are available in the software for comparative analysis. Fig. 3 illustrates the visualization of several versions of the CEPC detector within Phoenix. Fig. 3(a) shows the conceptual design report (CDR) version of detector geometry. Fig. 3(b) displays the latest technical design report (TDR) version of detector geometry under development [47]. In addition, Fig. 3 presents detailed section views of CEPC detector geometries, sequentially showing the VXD, Tracker, ECAL, HCAL, Electromagnetic Coil [48], and Muon detector from the innermost to the outermost layers [3].

Additionally, the Phoenix framework offers users the flexibility to select specific sections of the detector for visualization. For instance, Fig. 4 illustrates the display of CEPC sub-detectors. Due to performance constraints, many detailed aspects of the detector geometries may be omitted in the default view. However, users can manually change the opacity of the 3D model or activate the "Wireframe" mode through the user interface, as shown in Fig. 5. This mode reveals the mesh structures of the detector geometries, thereby providing a detailed view of the geometric configurations while also rendering the 3D models transparent. This feature is particularly useful for gaining deeper insights into the internal structure and layout of complex detector components.

B. Visualization of Events

In HEP experiments, it is standard practice to store event data in ROOT format due to its efficiency and flexibility. However, since Phoenix does not natively support ROOT event format for visualization, an additional step involving conversion of event data formats is required before visualization can occur. The Phoenix framework provides support for loading event data from JSON format files. To convert ROOT event format to JSON format, the tool EDM4hep2json, which is provided by the EDM4hep [49] toolkit, is utilized. This tool is capable of converting EDM4hep event data to JSON format while preserving the essential event information required for visualization.

Specifically, the EDM4hep2json toolkit effectively "dumps" the EDM4hep event data into JSON format, ensuring that all relevant event information is retained and usable within Phoenix. This conversion process bridges the gap between the event data storage format and the visualization requirements of the Phoenix framework, enabling comprehensive and interactive visualizations of events in the CEPC detector.

Once converted to JSON format, users can manually upload the event file to the event display software, and the first event in the event data file will be immediately visualized. Users can also choose which event to visualize through the user interface. Fig. 6 provides an illustrative example of the CEPC detector along with a single event as visualized by the Phoenix event display software. The figure highlights three primary physical event objects: tracks representing the trajectories of charged particles, depicted as red lines; calorimeter cells indicating energy deposits within a calorimeter, shown as small transparent cubes; and clusters representing groupings of energy deposits in a calorimeter, represented by yellow pointing cuboids.

In addition to these, the software supports visualization of several other physics objects, including jets (simplified geometric cones indicating the direction of energy flow), hits (individual detector measurements illustrated as points in space), and vertices (origins of tracks displayed as dots). The software further enhances user experience by enabling selective visualization of event objects through its intuitive interface. Users can disable certain object types, such as calorimeter cells, to focus on higher-level features like clusters. Since all event objects are stored as collections in the EDM4hep event data file, users have the option to enable or disable their visualization, allowing for focused analysis on specific aspects of the event.

Moreover, the software facilitates detailed analysis by providing information about selected objects. When "Object selecting" mode is activated, users can select objects to view detailed information in the info panel located in the top left corner. For example, selecting a cluster reveals its position, energy, and associated reconstructed particle properties (including charge, energy, mass, and momentum), as shown in the top-left corner of Fig. 6(c). This functionality is invaluable for thorough event analysis.

C. Visualization of Simulation Information

Monte Carlo information, generated from simulated particle interactions, provides ground-truth data for validating reconstructed objects (e.g., matching tracks to MC particles). In modern HEP experiments, MC information also plays an important role in event analysis, detector design, and algorithm optimization [11, 50]. It can help physicists evaluate detector performance, test reconstruction and analysis processes, and understand signal events and background sources. Since the CEPC experiment is still in the planning phase, MC simulation is the primary source of event data, and utilizing MC information is key to optimizing detector design and reconstruction algorithms [12].

On the basis of the Phoenix framework, we extend the original set of visualization primitives to display particles. Our event display software now supports detailed visualization of MC information. The EDM4hep event data model stores MC information as an individual collection, and the software displays it along with other event objects. For collider experiments such as CEPC, the MC information includes detailed properties of MC simulated particles, enabling physicists to validate reconstruction processes by comparing reconstructed tracks with MC truth information of particles. The event display software extracts information about MC particles, such as their vertex, momentum, charge, and endpoint, to calculate and draw their trajectories within the detector. As an example, Fig. 7 displays an event with both its reconstructed information and particle MC truth information. The similarity between the two sub-figures distinctively validates the correctness of reconstruction through visual comparison.

In the event display software, MC truth information is visualized as a set of tracks. To achieve better visualization effects, the software divides MC particles into three categories. Charged particles interact with the magnetic field and are detected by the Tracker. They are drawn as blue and yellow curves to show their possible tracks according to MC information. The curves are interpolated to show their trajectories more precisely in the magnetic field. Curve colors represent charge: negatively charged particles are drawn as yellow curves, while positively charged particles are assigned blue color. Neutral particles (except for neutrinos) may interact with calorimeters, leaving energy deposits in the detector. They are plotted as straight black lines. Neutrinos, which in most cases do not interact with the detector and are seen as missing energy and missing momentum in event analysis, are represented by black dashed lines.

Additionally, the tracks visualize complete spatial information of MC particles, including their vertices and endpoints. The endpoint of each track is marked by a sphere, which can help physicists determine the moving direction of MC particles. Furthermore, like other event objects, the display of MC information can be enabled or disabled through the user interface, and the tracks of MC particles can also be selected to provide more detailed information.

V. Features and Applications

A. Features Compared to Other Event Display Software

The CEPC event display software is forked from the original Phoenix framework and developed as an independent branch, which not only inherits the advantages and features of the Phoenix framework but also introduces new functionalities to meet requirements for detector design and event analysis. Compared to the original Phoenix framework, the CEPC event display software provides full support for visualizing detector geometries and events, and includes extra features to provide more diverse visualization and information, such as display of MC truth information and enhanced geometry clipping capabilities. Compared to ROOT-based [51–53] or Unity-based [54–56] event display software, its advantages become even more obvious, making it the prime choice for CEPC event display in some specific situations. Generally speaking, the CEPC event display software has the following prominent features.

First, it is easily accessible. Based on three.js, the event display software can be deployed either locally or on a web server, and users can access it with any web browser that supports WebGL [57]. This includes most modern web browsers, such as Chrome, Firefox, Edge, Android browser, and iOS Safari. Similar to other software, browsers rely on GPUs to accelerate rendering of 3D models, and thus the event display software requires modern GPU support. However, with the aid of advancing front-end technologies, it minimizes the need for powerful hardware and can operate efficiently even on mobile devices.

Second, it is lightweight. One outstanding feature of the Phoenix-based CEPC event display software is its lightweight nature. Unlike the ROOT framework, which heavily relies on third-party libraries, it can be easily bundled into a package and distributed to other devices. The software bundle occupies only 85 MB of storage space, enabling users to deploy the application on a personal computer without installing external software packages. Meanwhile, the Unity editor also allows building the source code into a standalone application. However, the resulting applications are generally not cross-platform and usually occupy too much memory.

Third, it is fast. The CEPC event display software demonstrates great performance advantages. To evaluate its performance, we deployed the software on a personal computer with an Intel i5-12400 processor and an Intel Iris Xe integrated graphics card, and accessed it via the Chrome browser. The software renders the detector and events smoothly, achieving a refresh rate exceeding 30 frames per second (FPS), with average memory usage of about 800 MB. During testing, we observed that the Iris Xe GPU utilization increased from 1% to approximately 18% during active visualization, indicating that the rendering process leverages both CPU and GPU resources for efficient performance. Notably, performance is closely related to the complexity of detector geometry and event data. When rendering simpler geometries, the refresh rate can be even higher. In comparison, the typical refresh rate of ROOT-based event display software is about 10 FPS [52, 53], and high latency between users and computing servers can significantly reduce its performance.

Finally, it offers comprehensive visualization features. Instead of writing basic logic for event display, the CEPC event display software inherits the functionalities of the Phoenix framework, which provides complete features for detector and event visualization. Furthermore, the software extends the original set of functions of the Phoenix framework to fulfill the needs for deeper analysis, making it a practical choice for CEPC event display.

B. Potential Applications

As previously introduced, the CEPC event display software provides a convenient and flexible way to visualize detectors and events. It has great potential for applications and further developments. Some possible applications of the event display software are listed below.

First, detector design and event analysis. As an event display software, its primary development goal is to assist researchers in analyzing detector geometry and event data. The software provides necessary features to display CEPC events and detectors, and has the flexibility to display other 3D geometries with EDM4hep format events, enabling users to gain insights about detector structure and event signals. Researchers can receive feedback from the software and consequently optimize detectors and algorithms.

Second, online monitoring. Since the CEPC event display software is developed using the web-based Phoenix framework, it is naturally suitable for online tasks. Instead of writing complex logic to fetch data from the internet, the event display software can be easily extended to access URLs to display online event data. Since the software can run in any modern web browser, users can even check experiment status with a smartphone. This will definitely help physicists instantly modify facility status and enhance high-quality experimental data taking.

Third, further development. As introduced previously, the Phoenix framework is highly modular and extendable. The changes and extensions we develop for the event display software can be easily applied to other applications, even to other HEP experiments such as BESIII [58] and JUNO [59]. Moreover, the 3D library three.js [16] used by Phoenix has provided support for Virtual Reality (VR) [60] and Augmented Reality (AR) [61] features, enabling further development of the software to build VR/AR applications for the CEPC experiment.

VI. Summary

With the aid of the Phoenix framework, we have developed a web-based event visualization software for the CEPC experiment. It is purely based on front-end web technologies and can be easily deployed and run on any web server or personal computer. This software provides necessary facilities for visualizing and displaying CEPC detectors and events, allowing researchers to gain deep insights into detector structures and event characteristics, thereby enhancing continuous optimization of detectors, simulation, and reconstruction algorithms. It also has great potential for a variety of applications, such as online monitoring and VR/AR programs.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Nos. 12175321, W2443004, 11975021, 11675275, and U1932101), 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.

References

[1] X. Lou. "The Circular Electron Positron Collider". *Nature Review Physics* 1 (2019), pp. 232–234. DOI: 10.1038/s42254-019-0047-1.

[2] The CEPC Study Group. "CEPC Conceptual Design Report: Volume 1 - Accelerator" (2018). DOI: 10.48550/arXiv.1809.00285.

[3] The CEPC Study Group. "CEPC Conceptual Design Report: Volume 2 - Physics & Detector" (2018). DOI: 10.48550/arXiv.1811.10545.

[4] J. de Blas et al. "Higgs Boson Studies at Future Particle Colliders". *JHEP* 01 (2020), p. 139. DOI: 10.1007/JHEP01(2020)139.

[5] M. Farina, G. Panico, D. Pappadopulo, et al. "Energy helps accuracy: electroweak precision tests at hadron colliders". *Phys. Lett. B* 772 (2017), pp. 210–215. DOI: 10.1016/j.physletb.2017.06.043.

[6] J. Peng, M. Zhao, X. Wang, et al. "Prospect for measurement of CP-violating observables in B0s K± decays at a future Z factory" (2025). DOI: 10.48550/arXiv.2502.11172.

[7] Y. Fang and S. Xin. "Physics Highlights at CEPC". *Nuclear and Particle*

Physics Proceedings 330-335 (2023). Higgs Potential 2022, pp. 15–19. DOI: https://doi.org/10.1016/j.nuclphysbps.2023.04.003.
[8] J. Gao. "The Status of the CEPC Project in EDR" (2025). DOI: 10.48550/arXiv.2505.04663.
[9] The HEP Software Foundation and Albrecht, J. and Alves, A.A. and others. "A Roadmap for HEP Software and Computing R&D for the 2020s". *Computing and Software for Big Science* (2019). DOI: https://doi.org/10.1007/s41781-018-0018-8.
[10] M. Ruan, H. Zhao, G. Li, et al. "Reconstruction of physics objects at the Circular Electron Positron Collider with Arbor". *The European Physical Journal C* 78 (2018).
[11] M. Liu, W. Li, X. Huang, et al. "Simulation and reconstruction of particle trajectories in the CEPC drift chamber". *Nucl. Sci. Tech.* 35.8 (2024), p. 128. DOI: 10.1007/s41365-024-01497-z.
[12] X. Ai. "Acts: A common tracking software" (2019). DOI: 10.48550/arXiv.1910.03128.
[13] R. Brun and F. Rademakers. "ROOT — An object oriented data analysis framework". *Nucl. Instrum. Meth. A* 389.1 (1997), pp. 81–86. DOI: https://doi.org/10.1016/S0168-9002(97)00048-X.
[14] M. Tadel. "Overview of EVE – the event visualization environment of ROOT". *Journal of Physics: Conference Series* 219.4 (2010), p. 042055. DOI: 10.1088/1742-6596/219/4/042055.
[15] Javascript ROOT. URL: https://root.cern/js/.
[16] Mrdoob. JavaScript 3D library three.js. URL: https://threejs.org/.
[17] E. Moyse, F. Ali, E. Cortina, et al. "The Phoenix event display framework". *EPJ Web of Conferences* 251 (2021), p. 01007. DOI: 10.1051/epjconf/202125101007.
[18] Google. AngularJS — Superheroic JavaScript MVW Framework. URL: https://angularjs.org/.
[19] M. Bellis, R. M. Bianchi, S. Binet, et al. "HEP Software Foundation Community White Paper Working Group — Visualization" (2018). DOI: 10.48550/arXiv.1807.07177.
[20] G. Aad, B. Abbott, D. Abbott, et al. "The ATLAS experiment at the CERN Large Hadron Collider: a description of the detector configuration for Run 3". *Journal of Instrumentation* 19.05 (2024). DOI: 10.1088/1748-0221/19/05/p05063.
[21] The CMS Collaboration, S. Chatrchyan, G. Hmayakyan et al. "The CMS experiment at the CERN LHC". 3.08 (2008), S08004. DOI: 10.1088/1748-0221/3/08/S08004.
[22] P. Owen and N. Serra. "Status and prospects of the LHCb experiment". *The European Physical Journal Special Topics* (2024). DOI: 10.1140/epjs/s11734-023-01010-4.
[23] P. Calafiura, S. Farrell, H. Gray, et al. "TrackML: A High Energy Physics Particle Tracking Challenge" (2018), pp. 344–344. DOI: 10.1109/eScience.2018.00045.
[24] FASER Collaboration, Akitaka Ariga et al. "FASER: ForwArd Search ExpeRiment at the LHC" (2019). DOI: 10.48550/arXiv.1901.04468.

[25] I. Agapov, M. Benedikt, A. Blondel, et al. “Future Circular Lepton Collider FCC-ee: Overview and Status” (2022). DOI: 10.48550/arXiv.2203.08310.

[26] Adobe. OBJ Files: Uses & 3D Applications. URL: https://www.adobe.com/products/substance3d/discove are-obj-files.html.

[27] T. K. G. Inc. gltf Overview. URL: https://www.khronos.org/api/index_{2017}/gltf.

[28] A. Pappas, B. Couturier, and S. Ponce. “New web based event data and geometry visualization for LHCb”. *Journal of Physics: Conference Series* 2438 (2023), p. 012109. DOI: 10.1088/1742-6596/2438/1/012109.

[29] CEPC offline software prototype. Institute of High Energy Physics, Chinese Academy of Sciences. URL: https://code.ihep.ac.cn/cepc/CEPCSW.

[30] M. Frank, F. Gaede, M. Petric, et al. AIDASoft-/DD4hep: v01-25-01. Version v01-25-01. 2023. DOI: 10.5281/zenodo.7673417.

[31] 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.

[32] B. H. Frank Gaede1 Gerardo Ganis. “EDM4hep and podio - The event data model of the Key4hep project and its implementation”. *EPJ Web of Conferences* (2021). DOI: 10.1051/epjconf/202125101008.

[33] F. Gaede, T. Madlener, P. Declara Fernandez, et al. “EDM4hep - a common event data model for HEP experiments”. *PoS ICHEP2022* (2022), p. 1237. DOI: 10.22323/1.414.1237.

[34] G. Ganis, C. Helsens, and V. Völkl. “Key4hep, a framework for future HEP experiments and its use in FCC”. *Eur. Phys. J. Plus* 137.1 (2022), p. 149. DOI: 10.1140/epjp/s13360-021-02213-1.

[35] T. Li, X. Huang, W. Huang, et al. “Core software for the super tau-charm facility”. *Mod. Phys. Lett. A* 39.40 (2024), p. 2440012. DOI: 10.1142/S0217732324400121.

[36] T.-Z. Song, K.-X. Huang, Y.-J. Zeng, et al. Detector description conversion and visualization in Unity for high energy physics experiments. 2025.

[37] Z.-Y. Yuan, T.-Z. Song, Y.-J. Zeng, et al. “Method for detector description conversion from DD4hep to Filmbox”. *Nuclear Science and Techniques* 35.9 (2024), p. 146. DOI: 10.1007/s41365-024-01506-1.

[38] Z. Y. You, Y. T. Liang, and Y. J. Mao. “A method for detector description exchange among ROOT GEANT4 and GEANT3*”.* Chinese Physics C* 32.7 (2008), p. 572. DOI: 10.1088/1674-1137/32/7/012.

[39] S. Agostinelli, J. Allison, K. Amako, et al. ”Geant4—a simulation toolkit“. *Nucl. Instrum. Meth. A* 506.3 (2003), pp. 250–303. DOI: https://doi.org/10.1016/S0168-9002(03)01368-8.

[40] R. Chytracek, J. Mccormick, W. Pokorski, et al. ”Geometry Description Markup Language for Physics Simulation and Analysis Applications“. *IEEE Transactions on Nuclear Science* 53.5 (2006), pp. 2892–2896. DOI: 10.1109/TNS.2006.881062.

[41] Y. T. Liang et al. ”A uniform geometry description for simulation, reconstruction and visualization in the BESIII experiment“. *Nucl. Instrum. Meth. A* 603 (2009), pp. 325–327. DOI: 10.1016/j.nima.2009.02.

[42] W. Dai et al. ”A Monolithic Active Pixel Sensor with a novel readout architecture for vertex detector in particle physics“. *JINST* 20.03 (2025), p. C03019. DOI: 10.1088/1748-0221/20/03/C03019.

[43] Y. Chang, H. R. Qi, X. She, et al. ”Design and commission of TPC prototype integrated with 266 nm UV laser“. *JINST* 20.02 (2025), P02025. DOI: 10.1088/1748-0221/20/02/P02025.

[44] W. Pan et al. ”Drift chamber with cluster counting technique for CEPC“. *PoS ICHEP2024* (2025), p. 1101. DOI: 10.22323/1.476.1101.

[45] H. Zhang, X. Zhao, C. Yuan, et al. ”Design and Simulation of a Stereo Crystal Electromagnetic Calorimeter for the CEPC“. *EPJ Web Conf.* 320 (2025), p. 00059. DOI: 10.1051/epjconf/202532000059.

[46] Y. Shi, Y. Zhang, M. Ruan, et al. ”Design and optimization of the CEPC scintillator hadronic calorimeter“. *JINST* 17.11 (2022), P11034. DOI: 10.1088/1748-0221/17/11/P11034.

[47] W. Abdallah et al. ”CEPC Technical Design Report: Accelerator“. *Radiat. Detect. Technol. Methods* 8.1 (2024). [Erratum: *Radiat.Detect.Technol.Methods* 9, 184–192 (2025)], pp. 1–1105. DOI: 10.1007/s41605-024-00463-y.

[48] M. Wang, F. Ning, L. Zhao, et al. ”Conceptual Design of CEPC Detector Magnet“. *IEEE Trans. Appl. Supercond.* 33.6 (2023), p. 4500608. DOI: 10.1109/TASC.2023.3275213.

[49] EDM4hep webpage. URL: https://github.com/key4hep/EDM4hep.

[50] W. Fang, X. Huang, W. Li, et al. ”Refined drift chamber simulation in the CEPC experiment“. *EPJ Web Conf.* 295 (2024), p. 03034. DOI: 10.1051/epjconf/202429503034.

[51] Z. J. Li, M. K. Yuan, Y. X. Song, et al. ”Visualization for physics analysis improvement and applications in BESIII“. *Frontiers of Physics* 19.6 (2024), p. 64201. DOI: 10.1007/s11467-024-1422-7.

[52] Z. You, K. Li, Y. Zhang, et al. ”A ROOT based event display software for JUNO“. *Journal of Instrumentation* 13.02 (2018), T02002–T02002. DOI: 10.1088/1748-0221/13/02/t02002.

[53] M. H. Liao, K. X. Huang, Y. M. Zhang, et al. ”A ROOT-based detector geometry and event visualization system for JUNO-TAO“. *Nuclear Science and Techniques* 36.3 (2025). DOI: 10.1007/s41365-024-01604-0.

[54] Unity Real-Time Development Platform. https://unity.com/.

[55] J. Zhu, Z. You, Y. Zhang, et al. ”A method of detector and event visualization with Unity in JUNO“. *Journal of Instrumentation* 14.01 (2019), T01007–T01007. DOI: 10.1088/1748-0221/14/01/t01007.

[56] J. Li and Z. You.”Event Visualization with Unity in BESIII experiment“. *PoS ICHEP2024* (2025), p. 1047. DOI: 10.22323/1.476.1047.

[57] Mozilla. WebGL: 2D and 3D graphics for the web. URL: https://developer.mozilla.org/en-US/docs/Web/API/WebGL_{API}.

[58] M. Ablikim et al. ”Design and Construction of the BESIII Detector“. *Nucl. Instrum. Meth. A* 614 (2010), pp. 345–399. DOI: 10.1016/j.nima.2009.12.

[59] A. Serafini.”The JUNO experiment: status and physics potential“. *PoS NOW2024* (2025), p. 004. DOI: 10.22323/1.473.0004.

[60] J. Zheng, K. Chan, and I. Gibson.”Virtual reality“. *IEEE Potentials* 17.2

(1998), pp. 20–23. DOI: 10.1109/45.666641.
[61] J. Carmigniani and B. Furht."Augmented Reality: An Overview". *Handbook of Augmented Reality*. Ed. by B. Furht. New York, NY: Springer New York, 2011, pp. 3–.

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

Source: ChinaXiv — Machine translation. Verify with original.