Study on Detector Geometry Transformation and Visualization in Unity

Kaixuan Huang, Yumei Zhang, Zhengyun You*

Sun Yat-sen University, Guangzhou 510275, China

E-mail: `youzhy5@mail.sysu.edu.cn`

Abstract. In High Energy Physics (HEP) experiments, it is useful for physics analysis and outreach if the event display software can provide sophisticated visualization effects. Unity is professional software that provides 3D modeling and animation production. GDML format files are commonly used for detector descriptions in HEP experiments. In this work, we present a method for automating the import of GDML files to the Unity engine modeling through geometric transformation. With the conversion of the file format, the complicated geometric description of the HEP detector in a GDML file can be imported directly into the Unity engine using an FBX file. This format conversion was applied to BESIII experiments to achieve an automated geometry conversion to build geometry models in the Unity engine.

1. Introduction

In High Energy Physics (HEP) experiments [1], the Geometry Description Markup Language (GDML) [2] is a popular language to describe detector geometry for large scientific apparatus [3, 4]. In recent years, the Unity engine [5], a 3D interactive display tool, has been widely used for game development, education, and scientific research. There have been quite a few high-energy physics experiments developing event display software based on the Unity engine to obtain better visualization.

However, since the GDML detector description is not supported for direct import in the Unity engine, These softwares are usually developed by constructing detector geometry in Unity. With this approach, there are some disadvantages. First, the geometric models of scientific apparatus are usually complex, consisting of tens of thousands of detector units, which requires a relatively large amount of work; second, it is difficult to maintain since the models also need to be manually modified when the detector geometry is updated.

This paper proposes a method to automate the conversion of GDML files to Filmbox (FBX) [6] files, which can be imported into the Unity engine. The difference between GDML files and FBX files lies in how the geometry is constructed. Since GDML contains solids and a geometry tree, the geometry is built in a hierarchy with daughter volumes placed in its mother volume with references of position and rotation [2]. However, FBX, a standard format for 3D software, displays geometry by meshes, so a conversion process is necessary.

In the industry market, there are tens of popular 3D file formats. However, none is commonly supported in both HEP software and Unity. So we need to find a data flow path that starts from GDML and ends in Unity with minimum conversion steps. We design a conversion method based on the Unity engine. This method converts GDML detector description to STEP files [7] with the help of the FreeCAD software [8], then to FBX files that the Unity engine supports with the help of the Pixyz Studio [9]. It is used to build geometry models in the Unity engine. This conversion method solves consistency and maintenance problems compared to manually creating detector geometry in Unity. It can also be applied to other experiments where the geometry file format is GDML. Creating geometric models in Unity also allows subsequent developments, such as event display and data quality monitoring.

2. Detector geometry in Unity

The Unity engine is commonly used in 3D creation, including games, architecture, and other fields. It is a component-based object system that enables software development for multiple platforms and provides technical support for augmented reality (AR) and virtual reality (VR).

2.1. Unity for detector visualization

The main advantages of the Unity engine are its multi-platform support and excellent display effects. As a professional 3D software, the Unity engine has a variety of material property parameters that can be adjusted, and this diversity can be applied to geometry and case visualization for a better display effect.

2.2. Unity based event display

The ATLAS experiment [10] is one of the four large detectors at the Large Hadron Collider (LHC) at CERN. A Unity-based event display software named CAMELIA (Cross-platform Atlas Multimedia Educational Lab for Interactive Analysis) [11] has been developed for the ATLAS experiment. When no event information is loaded, the software displays the geometric model of the detector with outstanding visual effects based on the actual material information of the geometry. When events are loaded, CAMELIA displays the process of particle collision and the geometric information of different particles and jets.

The Jiangmen Underground Neutrino Observatory (JUNO) experiment [12] uses reactor neutrinos to measure neutrino mass ordering. The experimental setup consists of a 35.4 m diameter acrylic sphere, with about 18,000 20" and 25,000 3" photomultiplier tubes (PMT) placed on a grid outside the acrylic sphere and immersed together in a 43.5 m diameter and 44 m high pool [13]. The experiment can simultaneously detect neutrino signals from various sources [14,15]. In the JUNO experiment, a Unity-based detector model and event visualization software named ELAINA (Event Live Animation with Unity for Neutrino Analysis) [16,17] has been developed. After importing the information of simulated events, ELAINA can display the neutrino trajectory and mark the PMT hits in different colors to represent their hit time.

2.3. Methodology

A method of detector geometry conversion from GDML to FBX files for Unity will be described in this section. The method is realized with the help of two software tools, FreeCAD [8] and Pixyz Studio [9]. FreeCAD is a common software for engineering drawings, with a workbench that supports importing GDML files [18]. Based on FreeCAD and GDML workbench [19], it is possible to convert GDML to STEP, a standard format in the CAD field. This step converts the solids information into mesh information in STEP format. Pixyz Studio transforms STEP to FBX, and the latter can be imported into the Unity engine as the base model for subsequent development.

As shown in Figure 1, GDML can be converted to FBX data format by the FreeCAD and the Pixyz Studio software. This method has the following advantages. First, the automated conversion avoids building models manually in Unity. Second, this method uses GDML to build the geometry model, so the geometry model created in the Unity engine is consistent with the geometry in offline software.

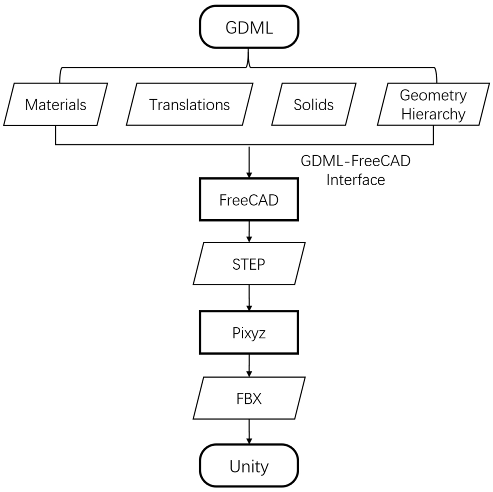

Figure 1. Conversion of detector geometry from GDML to Unity

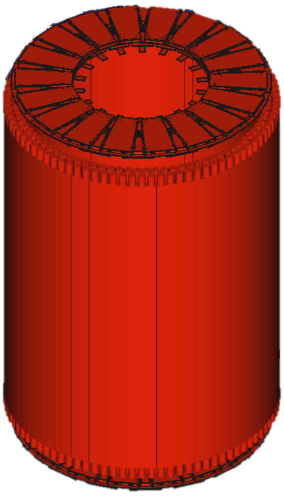

Figure 2. BESIII TOF in FreeCAD.

For example, visualization of the BESIII TOF sub-detector in FreeCAD and Pixyz is shown consistently. The BESIII experiment is running at the Beijing Electron Positron Collider [20]. The BESIII detector consists of a multilayer drift chamber (MDC), a time-of-flight system (TOF), an electromagnetic calorimeter (EMC), and a muon identification module (MUC) [21]. Here we use the BESIII TOF detector as an example of geometry conversion. A display of the BESIII TOF detector in FreeCAD software is shown in Figure 2.

2.4. Detector information in Unity

This method keeps the necessary geometric information of the GDML file and additional information while converting GDML to FBX. The additional information [2] includes materials and a geometry tree. The hierarchical structure of the geometry tree can also be automatically converted to the Unity engine. The hierarchical relationship provides convenience for determining the sub-module information of the geometry, which facilitates the subsequent development of event display based on the Unity engine.

The materials information can also be converted to the Unity engine. According to the required display, by manually adjusting the material property parameters in the Unity engine, different materials can have different display effects, such as color, transparency, reflectivity, etc, which will be helpful in further application development, such as event display and virtual reality.

3. Conversion of geometry data

A complete GDML contains the basic information about the geometry model of the detector: positions, rotations, materials, solids, a geometry tree, etc. In HEP experiments, the complex geometry model of the detector is usually composed of dozens of basic simple structures named solids, including boxes, tubes, trapezoids, etc. Based on the description of the geometry tree, different solids are combined in different volumes to form the detector modules. The materials part is the makeup information of the objects, which is consistent with the real materials of the detector.

3.1. GDML-FreeCAD geometry conversion

This method is based on the engineering drawing software FreeCAD and the format conversion software Pixyz Studio. In the GDML-FreeCAD interface [19], not all of the solid shapes currently defined in Geant4 or ROOT are fully supported. The interface only supports about ten kinds of shapes. Most of them are basic CSG [22] shapes. However, detectors often need to use complicated shapes, such as Polygon, Polyhedra, Twistedtubs, IrregBox, and Boolean shapes. We update the interface to allow the correct conversion of all shapes used in the BESIII GDML detector description. The updated content includes these unique solids, geometric Boolean operations, rotation methods, etc.

After adding conversion functions for the interface, the graphics tested in ROOT, FreeCAD software, and the Unity engine are consistent. This method does not need to create geometry in Unity manually, and the GDML file is automatically converted into a model in the Unity engine.

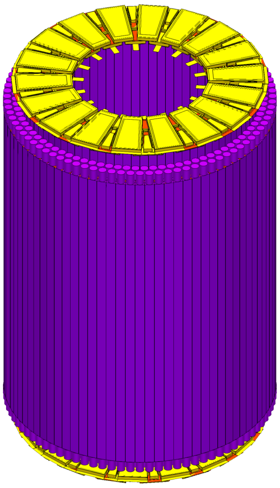

Figure 3. BESIII TOF in Pixyz Studio.

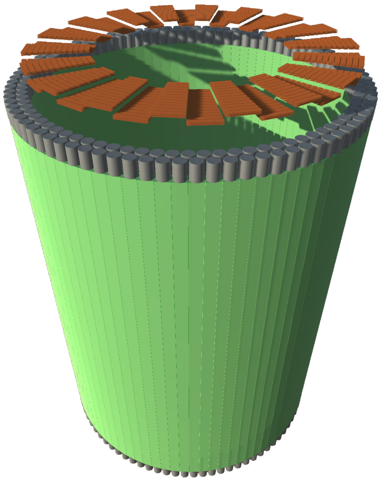

Figure 4. BESIII TOF in Unity.

3.2. FreeCAD-Unity geometry conversion

The FreeCAD software supports exporting the modeling in several popular 3D formats. STEP is chosen for further geometry transformation. Pixyz Studio can read the STEP file exported by FreeCAD and convert it into FBX format. The display of BESIII TOF in Pixyz Studio is shown in Figure 3. By importing FBX into the Unity engine, the display of the BESIII TOF detector is shown in Figure 4.

With this method, the geometry model built by GDML displayed in the Unity engine is the same as in Geant4 or ROOT in terms of geometry and hierarchy information.

3.3. Modeling in Unity

Following the above method, the material, file tree, geometry name, and the hierarchical structure information are preserved in the Unity engine. Since the geometry model imported into the Unity engine does not contain the material information of the color, transparency, and reflectivity of the material surface, these parameters can be set manually or with scripts in Unity to make the display in Unity matches the actual or more sophisticated visual effects. This is the advantage of using the professional 3D software, the Unity engine, for detector geometry visualization.

4. Summary

A method of converting detector description from GDML to Unity is presented, which reduces the workload in constructing the detector model in Unity applications while keeping the consistency of detector description between GDML and Unity. With the support of geometry conversion, more applications such as event display, data quality monitoring, and virtual reality programs can be developed in Unity.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No.11975021, 12175321, U1932101), National Key Research and Development Program of China under Contracts (No.2020YFA0406300, 2020YFA0406400), the Strategic Priority Research Program of Chinese Academy of Sciences (XDA10010900).

References

[1] M Bellis et al. Hep software foundation community white paper working group — visualization, 2018.

[2] R Chytracek, J Mccormick, W Pokorski, and G Santin. Geometry description markup language for physics simulation and analysis applications. *IEEE Transactions on Nuclear Science*, **53**(5):2892–2896, 2006.

[3] 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):572–575, 2008.

[4] K-J Li, Z-Y You, Y-M Zhang, et al. Gdml based geometry management system for offline software in juno. *Nucl. Instrum. Meth. A*, **908**:43–48, 2018.

[5] W Goldstone. *Unity game development essentials*. Packt Publishing Ltd, 2009.

[6] Fbx webpage. https://www.autodesk.com/products/fbx/overview.

[7] S J Kemmerer et al. Step: The grand experience. 1999.

[8] Freecad webpage. https://www.freecadweb.org.

[9] Pixyz studio software. https://www.pixyz-software.com/studio.

[10] The ATLAS Collaboration. The ATLAS experiment at the CERN large hadron collider. *JINST*, **3**(08):S08003–S08003, aug 2008.

[11] J Pequenao and USDOE. Atlas multimedia educational lab for interactive analysis, 4 2008.

[12] R Gioacchino et al. Status and prospects of the JUNO experiment. *J. Phys. Conf. Ser.*, **888**:012022, sep 2017.

[13] S Zhang et al. A method for sharing dynamic geometry information in studies on liquid-based detectors. *Nucl. Sci. Tech.*, **32**(21), 2021.

[14] Z-Y Li et al. Event vertex and time reconstruction in large-volume liquid scintillator detectors. *Nucl. Sci. Tech.*, **32**(49), 2021.

[15] Z Qian et al. Vertex and energy reconstruction in juno with machine learning methods. *Nucl. Instrum. Meth. A*, **1010**:165527, 2021.

[16] Z-Y You et al. A ROOT based event display software for JUNO. *JINST*, **13**(02):T02002, feb 2018.

[17] J Zhu, Z-Y You, Y-M Zhang, et al. A method of detector and event visualization with unity in JUNO. *JINST*, **14**(01):T01007–T01007, jan 2019.

[18] M Pinto and P Gonçalves. Guimesh: A tool to import step geometries into geant4 via gdml. *Computer Physics Communications*, **239**:150–156, 2019.

[19] KeithSloan et al. Freecad gdml workbench, 2022. https://github.com/KeithSloan/GDML.

[20] M Ablikim et al. Design and construction of the besiii detector. *Nucl. Instrum. Meth. A*, **614**(3):345–399, 2010.

[21] Y-T Liang et al. A uniform geometry description for simulation, reconstruction and visualization in the besiii experiment. *Nucl. Instrum. Meth. A*, **603**(3):325–327, 2009.

[22] D H Laidlaw, W B Trumbore, and J F Hughes. Constructive solid geometry for polyhedral objects. *SIGGRAPH Comput. Graph.*, **20**(4):161–170, aug 1986.