DAWN for GEANT4 Visualization

Satoshi Tanaka and Minato Kawaguti

Department of Information Science, Fukui University
Bunkyo 3-9-1, Fukui-shi, Fukui 910, JAPAN

We have developed a 3D graphics renderer DAWN, which is available to visualize GEANT4 simulation of high energy experiments. DAWN is suitable for precise visualization for presentation etc. It also enables remote visualization of the GEANT4 simulation via internet and stand alone re-visualization.

Key words: high quality; remote visualization; re-visualization.

1 Introduction

In this paper, we describe our graphics system "DAWN", which is available to visualize high energy experimental simulation. Recent progresses of DAWN have been done mainly concerned with simulator GEANT4 [1] under development.

The GEANT4 package includes drivers to several graphics systems to support varieties of requirements in visualization [2],[3],[4],[5],[6]. DAWN is a renderer for high-quality output or precise visualization. It has already been in use among developers of GEANT4 for the following purposes: (1) for generating high-quality PostScript figures for presentation, (2) for constructing geometries of detectors avoiding collisions, (3) for visualizing simulation with complicated detectors, and (4) for remote visualization of simulation. DAWN has several characteristic features, which make itself available for these purposes.

2 Review of the report at CHEP'95 [4]

DAWN is a 3D graphics renderer supporting analytical hidden-line/surface removal. It calculates out all visible parts of polygons and lines first, and then sends them to a selected drawing device as vector data. It allows very high-quality resolution, independent of a device in use. Such a feature is ideal for fine outputs for presentation and documentation. Only visible portions of 3D data are stored in memory, while invisible polygons and lines are automatically abandoned. Many useful functions are also available. For example, line haloing, automatic detection of outlines, and mixture of wireframe/surface visualization. DAWN also supports the conventional Z-buffer rendering for quick previewing.

3 Connection with GEANT4

DAWN reads data of 3D primitives (boxes, tubes, etc) from a (1) file, (2) pipe, or (3) socket, and outputs their visualized figure to X Window's and/or to PostScript files. GEANT4 simulation is visualized based on the client-sever model. DAWN runs as an independent process of GEANT4 and becomes a *visualization server*, while GEANT4 is a client. GEANT4 sends 3D primitives to request visualization of them, and DAWN responds to the requests.

4 Transferred data

DAWN understands data of 3D primitives with both the CSG and B-rep formats. It corresponds that GEANT4 stores 3D primitives with either of the formats [7]. As for CSG data, at present, DAWN understands data of the following classes: G4Box (box), G4Cons (cone or cone segment), G4Para (parallelepiped), G4Sphere (sphere or sphere segment), G4Trd (symmetric trapezoid), G4Trap (asymmetric trapezoid), G4Tubs (tube or tube segment). As for B-rep data, GEANT4 has an interface to convert them into ones understandable for DAWN. G4Polyline (a set of successive lines) is understandable to visualize steps and trajectories of particles. G4Circle (circle) and G4Square (square) are understandable to visualize hits.

Figure 1 is an example of visualizing GEANT4 simulation. The geometry data described with G4Box and G4Tubes are taken from the benchmark test included in the GEANT4 package. Figure 2 and Figure 3 visualize parts of the ATLAS detector. The former is described with G4Trd and the latter with G4Tubs and G4Cons.

5 Remote visualization and stand alone re-visualization

Remote visualization: Since GEANT4 and DAWN can be connected via TCP/IP socket, it is quite straightforward to run the two in different hosts

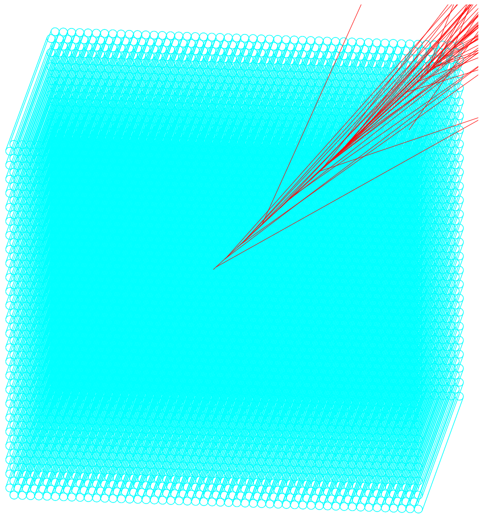

Fig. 1. Visualizing GEANT4 simulation based on its benchmark test

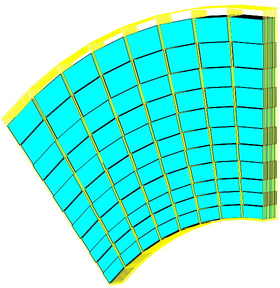

Fig. 2. Visualizing parts of the ATLAS detector I

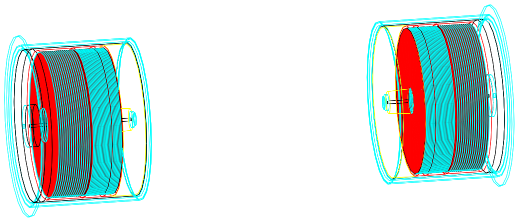

Fig. 3. Visualizing parts of the ATLAS detector II

connected with either local network or internet. It enables distributed processing of simulation and visualization. For example, a user can run GEANT4 at a remote host to use a big database there, and run DAWN at his local host to generate visualized figures quickly. Such *remote visualization* makes it avoid to transfer large data describing figures. The transferred data are very compact. Its size is one or two-order smaller than the corresponding PostScript codes.

Stand alone re-visualization: 3D data sent from GEANT4 are saved to a file named "g4.prim". DAWN can read the file by itself later without help of GEANT4 and re-visualize them (*stand alone re-visualization*). Camera position, drawing style (surface or wireframe etc), drawing device (X, PostScript, or OpenGL) and other visualizing parameters can be specified by a user on a multi-page GUI panel. These re-visualizing parameters need not be the same ones as in the visualization done in connection with GEANT4 before.

6 Conclusions

We have been developing our 3D graphics renderer DAWN for high-quality and precise visualization of high energy experimental simulation. Recent progresses have been done mainly concerned with GEANT4, e.g., for precise and remote visualization of GEANT4 simulation and its stand alone re-visualization. Such characteristic features of DAWN profit GEANT4 users as well as other graphics systems, which are complementary. DAWN is included in the GEANT4 package, and it is also obtainable from:
`ftp://i1nws2.fuis.fukui-u.ac.jp/pub/graphics/fukui_graphics_system/`

The author (ST) thanks J. Allison, K. Amako, S. Giani, A. Walkden, other GEANT4 developers, and our students for fruitful discussion. He also thanks Prof. H. Yamamoto and Prof. H. Yoshida for valuable suggestions.

References

[1] S. Giani, GEANT4: a world-wide collaboration to build Object Oriented HEP simulation software, in: *Proceedings of CHEP97* (to be published).

[2] J. Allison, Visualization in GEANT4 presented at HEPVIS96, `http://hepwww.ph.man.ac.uk/~johna/`

[3] J. Allison and S. Tanaka, The GEANT4 Visualization System, in: *Proceedings of CHEP97* (to be published).

[4] M. Kawaguti and S. Tanaka, *3D Graphics Module for Modeling, Visualization and Analysis,* in: *Proceedings of CHEP95* (to be published), Fukui University Preprint FU-HEP/IS-95-01.

[5] G. Barrand, OPACS and GEANT4, in: *Proceedings of CHEP97* (to be published), `http://www.lal.in2p3.fr/opacs/The_o_packages.html`.

[6] Joe Boudreau et al., hepvis (a set of Open Inventor C++ classes derived from SoShape), `http://www.lal.in2p3.fr/opacs/The_o_packages.html`.

[7] J. Apostolakis, Geometry in Geant 4, in: *Proceedings of CHEP97* (to be published).