VISUALIZATION

Mihaly Novak
based on material provided by
M. Asai (SLAC) & A. Dotti (SLAC) & J. Perl (SLAC) & L. Garnier (CNRS)
DAWN
OpenGL
HepRep/HepRApp

DAWN
RayTracer

OpenInventor

HepRep/FRED

gMocren
What can be visualized?

- Simulation data can be visualized
  - Geometrical components
  - Particle trajectories and tracking steps
  - Hits of particles in the geometry
  - Scored energy, dose, etc.

- Other user defined objects can be visualized
  - Polylines
    - such as coordinate axes
  - 3D Markers
    - such as eye guides
  - Text
    - descriptive character strings
    - comments or titles

- Geant4 visualisation documentation: Visualisation Documentation
A variety of choices

<table>
<thead>
<tr>
<th>Driver</th>
<th>Variant</th>
<th>High quality print</th>
<th>Interactive</th>
<th>browse geometry hierarchies</th>
<th>Direct access to G4 kernel</th>
<th>Make movies</th>
<th>Web</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenGL</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Xm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qt</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Win32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OpenInventor</td>
<td>Xt</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Win32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DAWN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VRML</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HepRep</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gMocren</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RayTracer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACSII File</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Controlling visualization

- Your Geant4 code stays basically the same no matter which driver you use.
- Visualization is performed either with commands or from C++ code.
  - For the present tutorial, we confine ourselves to command-driven visualization.
- Some visualization drivers work directly from Geant4:
  - OpenGL
  - OpenInventor
  - RayTracer
  - ASClITree
- For other visualization drivers, you first have Geant4 produce a file, and then you have that file rendered by another application (which may have GUI control):
  - HepRepFile
  - DAWNFILE
  - VRML2FILE
  - gMocrenFile
- You can open more than one driver at a time:
  - For example, do a quick check in OpenGL, then save the same event for a beautiful DAWN plot.
Controlling which drivers are available

- Six of the visualization drivers are always included by default (since they require no external libraries):
  - HepRepFile
  - DAWNFFile
  - VRMLFILE
  - RayTracer
  - gMocrenFile
  - ASCIIITree

- Other visualization drivers are included only if you request them in your `cmake` options.

- You can also add your own visualization driver
  - Geant4's visualization system is modular.
    By creating just three new classes, you can direct Geant4 information to your own visualization system.
Simplest command example

- Visualize your geometry in OpenGL
  - /vis/open OGL
  - /vis/drawVolume

- Most examples come with a visualization macro more complete
  - good starting point
Screenshots on the visualization drivers

- **Qt with OpenGL**
  - Requires that you install Qt
  - Visualization driver + user interface
  - You can customize the interface to control any Geant4 command

- **OpenGL**
- **OpenInventor**
- **HepRep**
- **DAWN**
- **VRML**
- **RayTracer**
- **gMocren**
- **ASCIITree**
1) **Qt Driver with OpenGL visualization**

- Recent developments focused on Qt User Interface and Visualization
- Geant4 **Qt + OpenGL Documentation**
- Demo...
Output from G4 kernel (support for search, MT)

Drift Chamber 4 has 5 hits.
Layer[0] : time 34.206100915825 (ns) --- local (X,Y) -224.18650513171, -0.21355242280892
Layer[1] : time 36.3760752815 (ns) --- local (X,Y) -251.45832124829, -1.2334283123023
Layer[2] : time 38.04693415785 (ns) --- local (X,Y) -278.64316463582, -2.2065434949855
Layer[4] : time 41.387329111728 (ns) --- local (X,Y) -33.34494486292, -4.22315735119901

EX Calorimeter has 7 hits. Total Edep is 967.12321716029 (MeV)
Hadron Calorimeter has 0 hits. Total Edep is 0 (MeV)
Run terminated.
Number of events processed : 1
User=0.078 Real=0.098 Sys=0.018
... write Root file : h5.root - done
WARNING: 1 event has been kept for refreshing and/or reviewing.
"/vis/view/keptEvents" to review them.
"/control/shell is
Visualization, one tab per viewer

Interactivity with mouse: rotate, zoom, move, pick
Toolbar and menubar controlled by icons.mac file, add your own without coding.
Help tree browser, free text search

Command /run/particle/addProcManager
Guidance: add process manager to specified particle type

Parameter | Guidance | Type | Omnicharge | Default | Range
---|---|---|---|---|---
1 | particleType | s | Tuo | | 

Drift Chamber 2 has 5 hits.
Layer[0] : time 34.796100216325 (nsec) --- local (x,y) -224.16650513171, -0.21355242280892
Layer[1] : time 38.37640752816 (nsec) --- local (x,y) -251.45832124828, -1.2334283123023
Layer[2] : time 38.04694157875 (nsec) --- local (x,y) -276.6416463582, -2.2065434910955
Layer[4] : time 41.387329111728 (nsec) --- local (x,y) -333.34694482692, -4.22315375111901
EX Calorimeter has 7 hits. Total Edep is 967.12237589017 (MeV)
END Calorimeter has 0 hits. Total Edep is 0 (MeV)
Run terminated.
Run Summary
Number of events processed : 1
User=0.07s Mean=0.09s Sys=0.01s
...write noun file : %5.root - done
WARNING: 1 event has been kept for refreshing and/or reviewing.
"/vis/reviewOptEvents" to review them.
/control/shell is

Session : 

16
run/beamOn 1
.control/shell is

History, re-select command

Drift Chamber 2 has 5 hits.
Layer[0] : time 34.7961009216825 (nsec) --- local (x,y) -224.16650513171, -0.213552422280892
Layer[1] : time 36.37640752814 (nsec) --- local (x,y) -251.45832124828, -1.234328323023
Layer[2] : time 38.04669457875 (nsec) --- local (x,y) -278.64164635682, -2.206834919055
Layer[4] : time 41.387329111728 (nsec) --- local (x,y) -333.34694482892, -4.22315375111901
EK Calorimeter has 7 hits. Total Edep is 987.12227138691 (MeV)
Hadron Calorimeter has 0 hits. Total Edep is 0 (MeV)
Run terminated.
Run Summary
Number of events processed : 1
User=0.07s Meas=0.09s Sys=0.01s
... write Root file : E5.root - done
WARNING: 1 event has been kept for refreshing and/or reviewing.
"/vis/reviewKeepEvents" to review them.
//control/shell is
Scene tree: edit visualization (e.g. hide volumes)
2) OpenGL

/vis/open OGL

- Features
  - Control directly from Geant4
  - Uses GL libraries that are already included on most Linux and Windows systems
  - Rendered, photorealistic image with some interactive features
    - zoom, rotate, translate
  - Fast response (can usually exploit full potential of graphics hardware)
  - Save as pixel graphics or vector EPS
  - Live movies
  - Geant4 Documentation: OpenGL
OpenGL with Motif Control

- Somewhat obsolete now that Qt can take over this functionality
  - but still supported
  - requires that you have Motif and link against this in your Geant4
3) OpenInventor

- `/vis/open OIX` or `/vis/open IOWin32`

**Features**

- Control from the OpenInventor GUI
- Requires addition of OpenInventor libraries (freely available for most Linux systems and Windows)
- Rendered, photorealistic image
- Many interactive features
  - zoom, rotate, translate
  - click to “see inside” opaque volumes
  - click to show attributes (momentum, etc., dumps to standard output)
- Fast response (can usually exploit full potential of graphics hardware)
- Expanded printing ability (vector and pixel graphics)
- Geant4 Documentation: [OpenInventor](#)

**Warning:**
OpenScientist (implementing our OI driver) is discontinued, but you could still try to use it.
4) HepRep

- /vis/open HepRepFile

Features
- The HepRepFile driver creates an XML file in HepRep1 format. This can be viewed with HepRApp HepRep Browser.
- The HepRepXML driver creates a HepRep file in HepRep2 format that can be viewed with WIRED4 plugin to JAS3 Analysis System or the FRED event display.
- Requires one of the above browsers (freely available for all systems)
- Wireframe or simple area fills (not photorealistic)
- Many interactive features
  - zoom, rotate, translate
  - click to show attributes (momentum, etc.)
  - special projections (FishEye, etc.)
  - control visibility from hierarchical (tree) view of data
- Hierarchical view of the geometry
- HepRApp and WIRED4 can export to many vector graphic formats (PostScript, PDF, etc.)
- Geant4 Documentation: HepRep

Warning: Issues with recent java versions
5) DAWN

- /vis/open DAWNFILE

- Features
  - Create a .prim file
  - Requires DAWN, available for all Linux and Windows systems
  - DAWN creates a rendered, photorealistic PostScript image
  - No interactive features once at PostScript stage
  - Highest quality technical rendering - vector PostScript
  - View or print from your favorite PostScript application
  - DAWN file can serve as input of 2 application programs: DAWNCUT and DAVID
  - Geant4 Documentation: DAWN

DAWNCUT and DAVID

- A standalone program, DAWNCUT, can perform a planar cut on a DAWN image
  - DAWNCUT takes as input a .prim file and some cut parameters. Its output is a new .prim file to which the cut has been applied.

- Another standalone program, DAVID, can show you any volume overlap errors in your geometry
  - DAVID takes as input a .prim file and outputs a new .prim file in which overlapping volumes have been highlighted.

http://geant4.kek.jp/~tanaka/
6) VRML

/-vis/open VRML1FILE or
/-vis/open VRML2FILE

- **Features**
  - Create a file to view in any VRML browser (some as web browser plug-ins i.e. can be viewed at a remote host)
  - Requires VRML browser (many different choices for different operating systems)
    - FreeWRL
  - Rendered, photorealistic image with some interactive features
    - zoom, rotate, translate
  - Limited printing ability (pixel graphics, not vector graphics)
  - Geant4 Documentation: VRML
7) RayTracer

- `/vis/open RayTracer`

## Features
- Create a jpeg file (and with RayTracerX option, also draws to x window)
- Forms image by using Geant4’s own tracking to follow photons through the detector
- Can show geometry but not trajectories
- Can render any geometry that Geant4 can handle (such as Boolean solids) - no other Vis driver can handle every case
- Supports shadows, transparency and mirrored surfaces
- Geant4 Documentation: [RayTracer](#)
RayTracerX

- You have the option of
  - /vis/open RayTracerX

- Builds same jpeg file as RayTracer, but simultaneously renders to screen so you can watch as rendering grows progressively smoother

- Means you can abort and retry the rendering with different view parameters without having to wait for the complete refinement of the image
8) gMocren

- Great tool available for volume visualization
- From JST/CREST project (Japan) to improve Geant4 for medical physics
- The gMocrenFile driver creates a gdd file then gMocren able to visualize
  - Volume data (including overlay of more than one set)
  - Trajectories
  - Geometry
- Runs on
  - Windows, Linux and Mac
  - Based on a commercial package but offered freely to all Geant4 users
  - [http://geant4.kek.jp/gMocren](http://geant4.kek.jp/gMocren)
- Installation is straightforward, follow the Download link on the above page
  - First run gMocren's one-click installer
  - Then, inside <gMocren-dir>/gtk, you will find the one-click installer for gtk
- To export Geant4 visualization to gMocren files
  - /vis/open gMocrenFile
  - /vis/scene/add/psHits
  - /vis/viewer/flush
  - many other options available with /score/draw ... commands
- Geant4 Documentation: [gMocren](http://geant4.kek.jp/gMocren)
9) ASCIIITree

- `/vis/open ATree`

- **Features**
  - Text dump of the geometry hierarchy
  - Not graphical
  - Control over level of detail to be dumped
  - Can calculate mass and volume of any hierarchy of volumes

- **Geant4 Documentation:**
  [ASCIIITree](http://example.com/)

---

**/vis/viewer/flush**

```
"worldPhysical":0
"magneticPhysical":0
"firstArmPhysical":0
"hodoscope1Physical":0
"hodoscope1Physical":1 (repeated placement)
"hodoscope1Physical":2 (repeated placement)
"hodoscope1Physical":3 (repeated placement)
"hodoscope1Physical":4 (repeated placement)
```

Can be set to **various levels of detail**

- `/vis/ASCIIITree/verbose <verbosity>`
  - 0: prints physical volume name.
  - 1: prints logical volume name.
  - 2: prints solid name and type.
  - 3: prints volume and density of solid.
  - 4: calculates and prints mass(es) of volume(s) in scene.

By default, shows only daughters of first placement and not repeat replicas.

Add 10 to the above to also show repeated placements and replicas.
At verbosity level 4, ASCIITree calculates the mass of the complete geometry tree taking into account daughters up to the depth specified for each physical volume.

The calculation involves subtracting the mass of that part of the mother that is occupied by each daughter and then adding the mass of the daughter, and so on down the hierarchy.

/vis/ASCIITree/Verbose 4
/vis/viewer/flush
"HadCalorimeterPhysical":0 / "HadCalorimeterLogical" / "HadCalorimeterBox"(G4Box), 1.8 m3, 11.35 g/cm3
  "HadCalColumnPhysical":-1 (10 replicas) / "HadCalColumnLogical" / "HadCalColumnBox"(G4Box), 180000 cm3, 11.35 g/cm3
    "HadCalCellPhysical":-1 (2 replicas) / "HadCalCellLogical" / "HadCalCellBox"(G4Box), 90000 cm3, 11.35 g/cm3
      "HadCalLayerPhysical":-1 (20 replicas) / "HadCalLayerLogical" / "HadCalLayerBox"(G4Box), 4500 cm3, 11.35 g/cm3
      "HadCalScintiPhysical":0 / "HadCalScintiLogical" / "HadCalScintiBox"(G4Box), 900 cm3, 1.032 g/cm3

Calculating mass(es)...
  Overall volume of "worldPhysical":0, is 2400 m3
  Mass of tree to unlimited depth is 22260.5 kg
Movies: time development of the event

- You can make movies that show time development of an event
  - i.e., a shower in slow motion

- Based on technique of “time-slicing”, breaking trajectories into individual slices, each with a time attribute.
  - requires newer visualization features, rich trajectory and some extensions to the OpenGL driver
  - you can run these animations directly from Geant4, does NOT involve stitching together a movie by hand

- A collection of example movies has been prepared by John Allison: http://www.hep.man.ac.uk/u/johna/pub/Geant4/Movies/


- Geant4 Documentation: movies
10 GeV pion

3 ns

Mpeg4 encoding with QuickTime Pro

http://www.hep.man.ac.uk/u/johna/pub/Geant4/Movies/pi-10Gevpi+neutronSideView.mp4
Tutorials and references on the Web!

- **DAWN**
  - DAWNCUT
    - [http://geant4.kek.jp/~tanaka/DAWN/About_DAWNCUT.html](http://geant4.kek.jp/~tanaka/DAWN/About_DAWNCUT.html)
  - DAVID
    - [http://geant4.kek.jp/~tanaka/DAWN/About_DAVID.html](http://geant4.kek.jp/~tanaka/DAWN/About_DAVID.html)
  - And more...

- **gMocren**
  - [http://geant4.kek.jp/gMocren](http://geant4.kek.jp/gMocren)

- **HepRApp**

- **OpenGL**

- **OpenScientist**

- **Qt**
• And here is a good time to discuss your graphics (not bitmapped graphics). A viewer's zoom feature, the image will...

• Now that you have some basic functionality created during the previous tutorial we can create a file called: example

• The output appears as included automatically

• One last command for the detector was drawn: run/beamOn1

The detector will then...

• Run DAWN on this file:
  
dawn g4_01.prim

• Go to the DAWN GUI's page 1 and...

• On the same page, change Camera XYZ coordinate (you can type in a specific number to change the new value).

• You should end up with an image and...
Summary

- Many visualization drivers are available

- The most recent: Qt + OpenGL
  - Visualization
  - GUI interface

- Movies

- Tutorials are available on the Internet
Backup
There are actually two OpenGL modes, **OGLS** and **OGLI**

- **Stored mode**: creates graphical database (display lists). Redrawing is faster because Geant4 only needs to resend parts that have changed. Nothing is lost on simple operations like change of viewing angle.
- **Immediate mode**: draws only to screen, no “memory”; detector can be redrawn after view changes but event data is lost.

And if you have Qt and Motif built and configured, even more options:

- **OGLSX**, **OGLIX**, **OGLSWin32** and **OGLIWin32** are basic OpenGL (stored, immediate, stored for Microsoft Windows, immediate for Microsoft Windows)
- **OGLSQt** and **OGLIQt** are OpenGL with Qt
- **OGLSXm** and **OGLIXm** are OpenGL with Motif

When you just use `/vis/open OGL`,

- We provide you the most advanced OpenGL that you currently have configured - Qt if you have it, otherwise Motif, otherwise basic OpenGL
- We give you Stored mode unless starts to use too much memory, in which case we switch to Immediate mode
- We worry for you about whether you're on Windows or not

To explicitly specify stored or immediate, but leave other decisions to us:

- `/vis/open OGLS`
- `/vis/open OGLI`
Introduction to Geant4 Visualization

John Allison: [http://www.hep.man.ac.uk/u/johna/](http://www.hep.man.ac.uk/u/johna/)  Movie with narration

10 GeV proton

![Image of particle tracks](image-url)