Geant4 Review 2007: Visualization

Geant4 Review 2007
16-20 April at CERN
Joseph Perl, SLAC
Sorry, Vis Talks always need two pages of Cover Graphics
Contents

- Introduction: Focus on Usability
- Status: Many Different Visualization Solutions
- New Features: Enhanced Trajectory Drawing
- New Features: Trajectory and Hit Filtering
- New Features: Smooth and Rich Trajectories
- New Features: Event Keeping
- New Features: Movies
- Coming by End of 2007: Visualization of Fields
- Additional Focus for this Year: Tools for Visualization of Voxel Data
- Additional Planned Developments: What else is In the Pipeline
- Help and Documentation: How we Keep our Users Informed
- Extra Slides: Details on the Seven Visualization Drivers
Focus on Usability

- From the very beginning, Geant4 visualization has been very flexible. The experienced programmer could exploit the full flexibility of C++ and a well-designed series of abstract interfaces to code any desired visualizations.

- The work of the Geant4 Visualization group over the last few years has been to identify those features that users want most, and make them easily achievable from simple interactive commands.

- So, in the last year or so we have added:
  - Trajectory Modeling commands
  - Trajectory and Hit Filtering commands
  - Smooth and Rich Trajectory commands
  - Event Keeping
  - Time-Development Animation, etc.

- More great new features will be coming, such as:
  - Visualization of Fields
Status: Many Different Visualization Solutions
Design by Interfaces

The first thing you may notice about Geant4 visualization is that there are a large number of different visualization systems.

This is a natural result of Geant4 being a toolkit and not a single application.

To support user communities who incorporate Geant4 into their own pre-existing software frameworks, Geant4 visualization is built around a set of well defined interfaces.

- These interfaces make it straightforward to connect Geant4's core visualization tools to any visualization system
  - able to drive advanced systems that can natively display complex solids such as Geant4's cut cylinders
  - able to drive more basic systems that do not understand such solids (system can ask Geant4 visualization to deconstruct complex solids into simpler polygons)
- For those users who want a ready-made visualization solution from Geant4, these same interfaces have made it straightforward for us to provide a variety of solutions, each with particular areas of strength.

- Interfaces discussed in detail in paper just finished this month and currently awaiting publication.
Geant4 Ships with Seven Visualization Drivers

- Geant4 comes with Seven visualization drivers:
  - OpenGL
  - OpenInventor
  - HepRep (using HepRApp, WIRED4 or FRED)
  - DAWN
  - VRML
  - RayTracer
  - ASCIITree

- The set of available drivers may seem like overkill, but the design of Geant4 visualization has been such that many drivers have required very little maintenance over the last few years while still filling unique niches for our users (examples are DAWN and VRML).

- The multiplicity of drivers does not place much burden on users since the set of commands the user sees are consistent regardless of the driver, and the user only links those drivers that they need.

- The drivers are discussed in detail in the extra slides at the end of this talk. For now, I'll just point out a few of the competing demands on Geant4 visualization, and how different visualization drivers meet different demands.
Quick response with flexible camera control to study geometries, trajectories and hits

- OpenGL and OpenInventor

- Widely available world standards for fast, photo-realistic rendering.
High-quality Output for Publications

- DAWN

- Developed specifically for Geant4. Not fast, but the highest quality rendering available anywhere. Excellent for publications.
Interactive Picking to Get More Information on Visualized Objects

- **HepRep**

  Reuses interactive HepRep browser applications already developed for other user communities (BaBar and GLAST). Explore hierarchies, pick to show attributes, apply cuts and labels.
Understand Complex Boolean Solids and Transparent or Reflective Surfaces

- RayTracer

- Reuses Geant4’s own tracking to shoot photons through the geometry
3D Format Suitable for Web Distribution

- VRML

- Many VRML browsers available, some as web plug-ins
Tools to Understand Geometry Hierarchies

- ASCIITree, HepRep
- /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"), 1.8 m³, 11.35 g/cm³
  - "HadCalCellPhysical":-1 (2 replicas) / "HadCalCellLogical" / "HadCalCellBox"("G4Box"), 1.8 m³, 11.35 g/cm³
- "HadCalLayerPhysical":-1 (20 replicas) / "HadCalLayerLogical" / "HadCalLayerBox"("G4Box"), 1.8 m³, 11.35 g/cm³
- "HadCalScintiPhysical":0 / "HadCalScintiLogical" / "HadCalScintiBox"("G4Box"), 1.8 m³, 11.35 g/cm³
Seven Visualization Drivers

• No Single Visualization Solution Can Meet all of Our Demands
  • Quick response with flexible camera control
  • High-quality Output for Publications
  • Interactive Picking to Get More Information
  • Complex Boolean Solids and Transparent or Reflective Surfaces
  • 3D Format Suitable for Web Distribution
  • Tools to Understand Geometry Hierarchies

• By exploiting the same interface design that we need anyway to support visualization systems of existing frameworks
  • we are able to take advantage of the best features of several different visualization drivers
  • with a common set of user commands
  • and minimal maintenance for many of the drivers

• We take advantage of the best features of many pre-existing visualization systems without having to reinvent those systems.

• Many more details on the drivers in the extra slides section at the end of this presentation
New Features:
Enhanced Trajectory Drawing
Enhanced Trajectory Drawing

- Ability to change trajectory drawing model through interactive commands

- Lets you, for example,
  - declare that trajectories should be color-coded by charge,
  - then change to have them color-coded by particle type

- Eliminates the most common reason users had to code their own trajectory classes

- Project Lead: Jane Tinslay
Example A01, five events, draw by various models
Sample Commands: generic trajectory model

# Create a generic model (will get default name of generic-0)
# From here we can set overall defaults for things like line color,
# whether to show step points or just the trajectory line, etc.
/vis/modeling/trajectories/create/generic

# Configure the generic model to colour all trajectories cyan and to show step points
/vis/modeling/trajectories/generic-0/default/setDrawStepPts true
/vis/modeling/trajectories/generic-0/default/setStepPtsSize 16
/vis/modeling/trajectories/generic-0/default/setLineColour cyan
/vis/modeling/trajectories/generic-0/default/setStepPtsColour red
Sample Commands: drawByCharge model

# Create a drawByCharge model (will get default name of drawCharge-0)
/vis/modeling/trajectories/create/drawByCharge

# Create another drawByCharge model with an explicit name of testChargeModel
/vis/modeling/trajectories/create/drawByCharge testChargeModel

# We can now go on to configure these two different drawByCharge models and then
# can choose to visualize using either one of them at any time

# Configure drawByCharge-0 model
/vis/modeling/trajectories/drawByCharge-0/set 1 red
/vis/modeling/trajectories/drawByCharge-0/set -1 red
/vis/modeling/trajectories/drawByCharge-0/set 0 white

# Configure testCharge model through G4Colour components
/vis/modeling/trajectories/testChargeModel/setRGBA 1 0 1 1 1
/vis/modeling/trajectories/testChargeModel/setRGBA -1 0.5 0.5 0.5 1
/vis/modeling/trajectories/testChargeModel/setRGBA 0 1 1 0 1

# List available models
/vis/modeling/trajectories/list

# select drawByCharge-0 to be current
/vis/modeling/trajectories/select drawByCharge-0
Sample Commands: drawByParticleID model

# Create a drawByParticleID model named drawByParticleID-0
/vis/modeling/trajectories/create/drawByParticleID

# Configure drawByParticleID-0 model
/vis/modeling/trajectories/drawByParticleID-0/set gamma red
/vis/modeling/trajectories/drawByParticleID-0/set proton yellow
/vis/modeling/trajectories/drawByParticleID-0/setRGBA e+ 1 0 1 1
Example A01, five events, drawByAttribute models
Sample Commands: drawByAttribute

/vis/modeling/trajectories/create/drawByAttribute
/vis/modeling/trajectories/drawByCharge-0/verbose true
/vis/modeling/trajectories/drawByCharge-0/setAttribute CPN
/vis/modeling/trajectories/drawByAttribute-0/addValue brem_key eBrem
/vis/modeling/trajectories/drawByAttribute-0/addValue annihil_key annihil
/vis/modeling/trajectories/drawByAttribute-0/addValue decay_key Decay
/vis/modeling/trajectories/drawByAttribute-0/addValue muIon_key muIoni
/vis/modeling/trajectories/drawByAttribute-0/addValue eIon_key eIoni
/vis/modeling/trajectories/drawByAttribute-0/brem_key/setLineColour red
/vis/modeling/trajectories/drawByAttribute-0/annihil_key/setLineColour green
/vis/modeling/trajectories/drawByAttribute-0/decay_key/setLineColour cyan
/vis/modeling/trajectories/drawByAttribute-0/eIon_key/setLineColour yellow
/vis/modeling/trajectories/drawByAttribute-0/muIon_key/setLineColour magenta
Enhanced Trajectory Drawing: Releases

- **Since Release 8.0:**
  - Color by Charge
  - Color by Particle Type

- **Since Release 8.1**
  - Color by Origin Volume
  - Control over more than just color, including
    - whether to show just trajectory line, or just trajectory points, or both
    - width of trajectory lines, type of marker to use for points, point size, etc.

- **Since Release 8.2**
  - Color by Any of the HepRep-Style Attributes, such as:
    - Initial volume name
    - Magnitude of momentum
    - Number of trajectory points
    - User defined attributes
    - Creator process name & type (if using Rich Trajectory)
    - Next volume name (if using Rich Trajectory)
New Features:
Trajectory and Hit Filtering
Trajectory and Hit Filtering

- Display user-defined subset of trajectories
  - Solves problems with large graphics files, busy events

- Two modes of operation
  - Rejected trajectories not drawn at all
  - Rejected trajectories drawn but marked invisible
    - Useful in interactive graphics browsers such as HepRApp or OpenInventor, where you have the option to toggle visibility later from the browser.
    - Drawback is that the files remain large, since all of the data is still in the file.

- Similar structure to enhanced trajectory drawing
  - Set of simple filter models
  - Similar Interactive creation/configuration structure

- You can chain multiple filters

- Project Lead: Jane Tinslay
HandsOn5, McGill tutorial, 1000 events, Attribute Filter

Momentum (MeV)

- 0-2.5
- 2.5-5
- 5-7.5
- 7.5-10
- 10-10.25
- 12.5+

IMag > 2.5 MeV

IMag > 2.5 MeV, particle = gamma

April 2007
Sample Macro Commands

/vis/modeling/trajectories/drawByAttribute-0/setAttribute IMag
/vis/modeling/trajectories/drawByAttribute-0/addInterval interval1 0.0 keV 2.5 MeV
/vis/modeling/trajectories/drawByAttribute-0/addInterval interval2 2.5 MeV 5 MeV
/vis/modeling/trajectories/drawByAttribute-0/addInterval interval3 5 MeV 7.5 MeV
/vis/modeling/trajectories/drawByAttribute-0/addInterval interval4 7.5 MeV 10 MeV
/vis/modeling/trajectories/drawByAttribute-0/addInterval interval5 10 MeV 12.5 MeV
/vis/modeling/trajectories/drawByAttribute-0/addInterval interval6 12.5 MeV 10000 MeV
/vis/modeling/trajectories/drawByAttribute-0/interval1/setLineColourRGBA 0.8 0 0.8 1
/vis/modeling/trajectories/drawByAttribute-0/interval2/setLineColourRGBA 0.23 0.41 1 1
/vis/modeling/trajectories/drawByAttribute-0/interval3/setLineColourRGBA 0 1 0 1
/vis/modeling/trajectories/drawByAttribute-0/interval4/setLineColourRGBA 1 1 0 1
/vis/modeling/trajectories/drawByAttribute-0/interval5/setLineColourRGBA 1 0.3 0 1
/vis/modeling/trajectories/drawByAttribute-0/interval6/setLineColourRGBA 1 0 0 1

/vis/filtering/trajectories/create/attributeFilter
/vis/filtering/trajectories/attributeFilter-0/setAttribute IMag
/vis/filtering/trajectories/attributeFilter-0/addInterval 2.5 MeV 1000 MeV
/vis/filtering/trajectories/create/particleFilter
/vis/filtering/trajectories/particleFilter-0/add gamma
Hit Filtering

- The Attribute-Based filtering discussed above for Trajectories was implemented generically so that it can apply to any class which implements the Geant4 generic attributes method:

  ```cpp
  const std::map<G4String, G4AttDef>* GetAttDefs() const;
  std::vector<G4AttValue>* CreateAttValues() const;
  ```

- Whatever your Hit class, you can basically get interactive hit filtering for free

- To activate, add a filter call to G4VVisManager in Draw method of hit class

  ```cpp
  void MyHit::Draw() {
      ...
      if (! pVVisManager->FilterHit(*this)) return;
      ...
  }
  ```
Trajectory and Hit Filtering: Releases

- Since Release 8.1
  - Filter by Charge
  - Filter by Particle Type
  - Filter by Origin Volume

- Since Release 8.2
  - Filter by Any of the HepRep-Style Attributes, such as:
    - Initial volume name
    - Magnitude of momentum
    - Number of trajectory points
    - User defined attributes
    - Creator process name & type (if using Rich Trajectory)
    - Next volume name (if using Rich Trajectory)

- Future Releases
  - Filter by creation time of particle
  - more
New Features:
Smooth and Rich Trajectories
G4SmoothTrajectory and G4RichTrajectory

- Most users use the standard G4Trajectory. But two other trajectory classes have been available in Geant4 tracking for some time.
  - Smooth Trajectory adds auxiliary points to allow smoother line in visualization (not Geant4 Steps, no physics at auxiliary points)
    - Useful when trajectories are highly curved due to magnetic fields
  - Rich Trajectory encodes additional information at every step point

- Project Lead: John Allison, Joseph Perl
Regular versus Smooth Trajectory

- **Regular Trajectory** is a polyline made up of only the actual steps used by Geant4.
- **Smooth Trajectory** includes additional points to make the polyline appear smoother.
- These additional points are not Geant4 steps. They are only used by visualization.
Smooth Trajectory for Vis in Magnetic Fields

- Yellow dots are the actual step points used by Geant4
- Magenta dots are auxiliary points added just for purposes of visualization
Rich Trajectory

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TED</td>
<td>0.008508...</td>
</tr>
<tr>
<td>Pos-Z</td>
<td>-6.15832...</td>
</tr>
<tr>
<td>Pos-Y</td>
<td>0.087018...</td>
</tr>
<tr>
<td>Pos-X</td>
<td>0.141176...</td>
</tr>
<tr>
<td>PTDS</td>
<td>Electroma...</td>
</tr>
<tr>
<td>PDS</td>
<td>msc</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TED</td>
<td>0.008298...</td>
</tr>
<tr>
<td>Pos-Z</td>
<td>-6.16151...</td>
</tr>
<tr>
<td>Pos-Y</td>
<td>0.084928...</td>
</tr>
<tr>
<td>Pos-X</td>
<td>0.121435...</td>
</tr>
<tr>
<td>PTDS</td>
<td>Electroma...</td>
</tr>
<tr>
<td>PDS</td>
<td>msc</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ch</td>
<td>-1 e+</td>
</tr>
<tr>
<td>IMag</td>
<td>0.429479...</td>
</tr>
<tr>
<td>PDG</td>
<td>11</td>
</tr>
<tr>
<td>ID</td>
<td>21</td>
</tr>
<tr>
<td>PN</td>
<td>e-</td>
</tr>
<tr>
<td>PID</td>
<td>1</td>
</tr>
<tr>
<td>IMom-Z</td>
<td>0.153368...</td>
</tr>
<tr>
<td>IMom-Y</td>
<td>0.353216...</td>
</tr>
<tr>
<td>IMom-X</td>
<td>0.190181...</td>
</tr>
<tr>
<td>IVN</td>
<td>firstArmPh...</td>
</tr>
<tr>
<td>NTP</td>
<td>26</td>
</tr>
<tr>
<td>IVN</td>
<td>firstArmPh...</td>
</tr>
</tbody>
</table>
Smooth and Rich Trajectories Now Easier to Use

- Up to Release 8.1, users who wanted to use these trajectories have had to write their own tracking action and explicitly instantiate one of these special kinds of trajectory.

- Release 8.2 made this possible from an interactive command:
  - `/vis/scene/add/trajectories`
  - `/vis/scene/add/trajectories smooth`
  - `/vis/scene/add/trajectories rich`
  - `/vis/scene/add/trajectories smooth rich`

- Remember that if you actually want to display the trajectory points, you need to issue the appropriate trajectory modeling commands, such as:
  - `/vis/modeling/trajectories/create/generic`
  - `/vis/modeling/trajectories/generic-0/default/setDrawStepPts true`
  - `/vis/modeling/trajectories/generic-0/default/setStepPtsSize 16`
  - `/vis/modeling/trajectories/generic-0/default/setStepPtsColour red`
New Features:
Event Keeping
Keep Event at end of Run

- Before Geant4 release 8.2, event data was discarded at the end of the run. Implications for visualization were that one could not go back and adjust the visualization after the run.
  - Transient data was written during end of run.
  - Transient data was unavailable for subsequent drawings.

- Prompted by this visualization issue, Geant4 changed how the event pointer is maintained. After the run one can now:
  - Change visualization drivers and redraw the same event to the new driver
  - Change trajectory modeling or filtering and redraw the event
  - If run was for more than one event, review events in visualization one by one
    - User can implement code to keep only those events that match some specific topology (specific hit or trigger pattern).
    - Makes it easy to do a large run and then recall for visualization only those events that are of interest.
New Features: Movies
Movies that Just Involve Changing Camera Position

- Making movies with just changes to camera position has been possible for years using macros.
- Conversion to mpeg somewhat complicated
  - documented since release 8.2
  - or see
    http://geant4.slac.stanford.edu/Presentations/vis/HowToMakeAMovie.ppt (and .pdf)
Movies: Stitched Together from Multiple Stills

QuickTime and a YUV420 codec decompressor are needed to see this picture.

http://www.hep.man.ac.uk/u/johna/pub/Geant4/Movies/g4RayTracer.mpg
New Technique: Time Development of the Event

- New features since release 8.2 allow you to do a new kind of Geant4 movie.
  - 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/

- How-To Presentation:
  http://geant4.slac.stanford.edu/Presentations/vis/HowToMakeAMovie.ppt
  http://geant4.slac.stanford.edu/Presentations/vis/HowToMakeAMovie.pdf

- Project Lead: John Allison
QuickTime and a decompressor are needed to see this picture.
What was in the Previous Movie

- http://www.hep.man.ac.uk/u/johna/pub/Geant4/Movies/pi-10Gevpi+neutronZoom.mp4

- 10 GeV/c π⁻ on lead (in a lead-liquid-argon calorimeter, exampleN03 with QGSP physics)
  - A plethora of slow pions, protons and neutrons
  - Three fast π⁻ and one fast π⁺ that subsequently interacts again
  - Neutrons (yellow) hang around for several ns

- Green circle is the light front
  - Careful viewer will notice that near the end of the event, some particles appear to cross the light front. This was a bug in the way the light front sphere was rendered in perspective view, fixed in release 8.2
Coming by end of 2007: Visualization of Fields
Additional Focus for This Year:
Tools for Visualization of Voxel Data
gMocren

- Geant4 has been interfaced to a commercial tool for visualization of voxel data.
- Free to all Geant4 users.
- Does not actually function as integral part of the Geant4 visualization system.
  - Implemented as a G4 example, outputs data in form appropriate for gMocren
Other Voxel Visualization Tools

- Improve how voxel data is handled by the rest of the visualization system.
- Important as we move from primarily a HEP user base to a base that includes Medical and Shielding studies.
- Requires features such as ability to interactively slice the visualization.
- Do not attempt to reproduce the full functionality of already existing systems such as gMocren, but rather support and enhance the interfaces to such systems.
Additional Planned Developments (1 of 2)

- Additional trajectory models and filters
  - show random subset of trajectories.
  - filter by fraction of primary energy
  - filter by how many generations from primary
  - model by what interaction type created particle
  - model hue by particle type and at same time model intensity by momentum.

- Additional ways of using attributes beyond HepRep driver:
  - Support "pick to show attributes" in OpenGL
  - Filtering of geometry according to attributes
  - Commands to Label trajectories or hits (as can currently only be one in HepRep browsers)
Additional Planned Developments (2 of 2)

- Extend functionality current in only some drivers to other drivers:
  - 2D and 3D text
  - Asymmetric scaling
  - Background color
  - Window location
  - Smooth shading
  - Control of auxiliary edges
  - Store camera information from one driver to use in another

- Other improvements:
  - Develop Web-based DAWN rendering service
  - Explore dynamic loading of visualization drivers

- Area of concern:
  - Complex Boolean shapes not all visualizable beyond RayTracer.
Help and Documentation
Well Integrated, Focused set of Communications Tools

- Over the past two years, we have refined our set of tools to communicate with users.
- Improved interactive command guidance:
  - `/help/vis` is now the single source for all up to date guidance
- Completed updated and revised visualization sections of:
  - User Guide for Application Developers
  - User Guide for Toolkit Developers
- Developed a set of introductory presentations:
  - Introduction to Geant4 Visualization
  - Geant4 Visualization Commands
  - Geant4 Advance Visualization
- Developed a set of Step by Step Tutorials:
  - OpenGL
  - DAWN
  - HepRep/HepRApp
- Removed other documents to avoid potential for conflicting information:
  - Readmes and Howtos
- A technical paper documenting the overall design of the Geant4 Visualization System is in final stages of submission to Computer Physics Communications
Summary

- Seven Visualization Drivers, all well supported
- Recent focus on adding interactive features to accomplish what users used to have to do by recoding in C++:
  - Trajectory Modeling commands
  - Trajectory and Hit Filtering commands
  - Smooth and Rich Trajectory commands
  - Time-Development Animation, etc.

- Major new feature coming this year:
  - Visualization of Fields

- Major additional focus for this year:
  - Improve tools for visualization of voxel geometries

- Additional planned developments:
  - Continue to improve ease of use

- Area of concern:
  - Complex Boolean shapes not all visualizable beyond RayTracer

- Help and documentation:
  - Well integrated set of documents
References and Further Resources
Geant4 Visualization Resources

Complete list of what has been done since release 8.0 and what is to do

- [http://geant4.slac.stanford.edu/Presentations/vis/workplans/VisToDoList_20061011d.doc](http://geant4.slac.stanford.edu/Presentations/vis/workplans/VisToDoList_20061011d.doc) (and .pdf)

Geant4 Installation Guides


Hands on HepRApp Tutorial


Hands on DAWN Tutorial


Hands on OpenGL Tutorial


Introduction to Geant4 Visualization


Geant4 Visualization Commands


Geant4 Advanced Visualization


On-line Documentation on Geant4 Visualization:


List of Visualization Commands:


For Questions or Comments: Geant4 Visualization Online Forum:

References

- OpenScientist Home Page

- HepRep: a generic interface definition for HEP event display representables
  http://www.slac.stanford.edu/~perl/heprep

- Fred: oh no, another event display (a HepRep client)
  http://www.fisica.uniud.it/~glast/FRED

- HepRApp HepRep Browser
  http://www.slac.stanford.edu/~perl/HepRApp

- WIRED4 JAS Plugin
  http://wired.freehep.org

- DAWN Hot Information
  http://geant4.kek.jp/Geant4/vis

- DAWN Home Page
  http://geant4.kek.jp/~tanaka/DAWN/About_DAWN.html

- DAWNCUT Home Page
  http://geant4.kek.jp/~tanaka/DAWN/About_DAWNCUT.html

- DAVID Home Page
  http://geant4.kek.jp/~tanaka/DAWN/About_DAVID.html

- Satoshi Tanaka’s GEANT4 Ritsumeikan University Group Home Page (more information on DAWN, sample PRIM files, images, etc.)
  http://geant4.kek.jp/~tanaka/
Extra Slides:
Details on the Seven Visualization Drivers
OpenGL

- /vis/open OGLIX

- 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)
  - Limited printing ability (pixel graphics, not vector graphics)

- Supported by John Allison with Guy Barrand
OpenGL with Motif Control

- If you don’t have Motif, all control is done from Geant4 commands:
  - /vis/open OGLIX or OGLSWin32
  - /vis/viewer/set/viewpointThetaPhi 70 20
  - /vis/viewer/zoom 2
  - etc.

- But if you have Motif libraries, you can control Geant4 from Motif widgets:
  - /vis/open OGLIXm
Hidden Line Removal

- OpenGL supports hidden line removal.
- You can control whether this removal is done and whether trajectories and hits are affected by this feature.

- By default, hidden line removal is disabled

- To turn on hidden line removal
  - `/vis/viewer/set/hiddenEdge 1`
  - This hides edges of geometry, but lets trajectories through.

- To hide trajectories and hits as well
  - `/vis/viewer/set/hiddenMarker 1`
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)

- Supported by Guy Barrand
OpenInventor: Start from Geant4

- With OpenInventor, start from Geant4, but then some control from OpenInventor GUI
OpenInventor: More GUI Control

- You can also choose to control the Geant4 run from OpenInventor.
HepRep

/vis/open HepRepFile

Features

- Create a file to view in the
  - HepRApp HepRep Browser
  - WIRED4 JAS Plugin
  - or FRED Event Display
- 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.)

Supported by Joseph Perl and Mark Donszelmann
HepRApp: Pick to Show Physics Attributes

- Picked on this volume to show:
  - Material
  - Density
  - Radlen
  - etc.

- Picked on this trajectory to show:
  - Particle ID
  - Charge
  - Momentum
  - etc.
HepRApp: Labeling by Any Attribute
HepRAp: Cut by Any Attribute

[Image of software interface showing data browser and cut control options]
WIRED4: Part of JAS

- Install JAS (Java Analysis Studio)
- Then download WIRED4 Plug-in
- http://wired.freehep.org/index.html
FRED: Fox Ruby Event Display

- An additional HepRep-compatible browser developed by members of the GLAST space telescope collaboration.
- Includes the fast rotations and beautiful rendering of GL plus HepRep interactivity.
- Allows scripting to change any attribute based on logic involving other attributes, hence things like "color by momentum" are scriptable.
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

- Supported by Satoshi Tanaka
  - Recently ported DAWN, DAWNCUT and DAVID to FedoraCore 4.0 (g++ 4.0)
DAWN Examples

- From a repository of beautiful images at
DAWN makes True Vector PostScript

- So when you zoom in with your PostScript browser, the images retain high resolution
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.

- Details at http://geant4.kek.jp/~tanaka/
/vis/open VRML1FILE or /vis/open VRML2FILE

Features
- Create a file to view in any VRML browser (some as web browser plug-ins).
- Requires VRML browser (many different choices for different operating systems).
- Rendered, photorealistic image with some interactive features
  - zoom, rotate, translate
- Limited printing ability (pixel graphics, not vector graphics)

Supported by Satoshi Tanaka
VRML

- Geant4 creates VRML File
  - `/vis/open VRML1FILE` or `/vis/open VRML2FILE`
- View file in a VRML Browser
  - Many free options, for example, here is one from octaga.com
RayTracer

- /vis/open RayTracer

Features
- Create a jpeg file
- 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

As of release Geant4.8.0, also now RayTracerX
- /vis/open RayTracerX
- Simultaneously renders to screen and to jpeg file, so that you can watch as the rendering grows progressively smoother

Supported by Makoto Asai and John Allison
RayTracer Shows Shadows
RayTracer Supports Transparency
RayTracer Handles Mirrored Surfaces

Mirrored Surfaces
RayTracer Handles Boolean Solids
RayTracerX

- New since Geant4.8.0
- In addition to
  - /vis/open RayTracer
- 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.
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

- Supported by John Allison
ASCIITree

- ASCIITREE is a visualization driver that is not actually graphical, but that dumps the hierarchy as a simple text tree.
  - `/vis/open ATree`

- `/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/ASCIITree/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
We have Seven Visualization Drivers
with complimentary strengths.
All well supported.

- OpenGL
- OpenInventor
- HepRep (using HepRAApp, WIRED4 or FRED)
- DAWN
- VRML
- RayTracer
- ASCIIITree