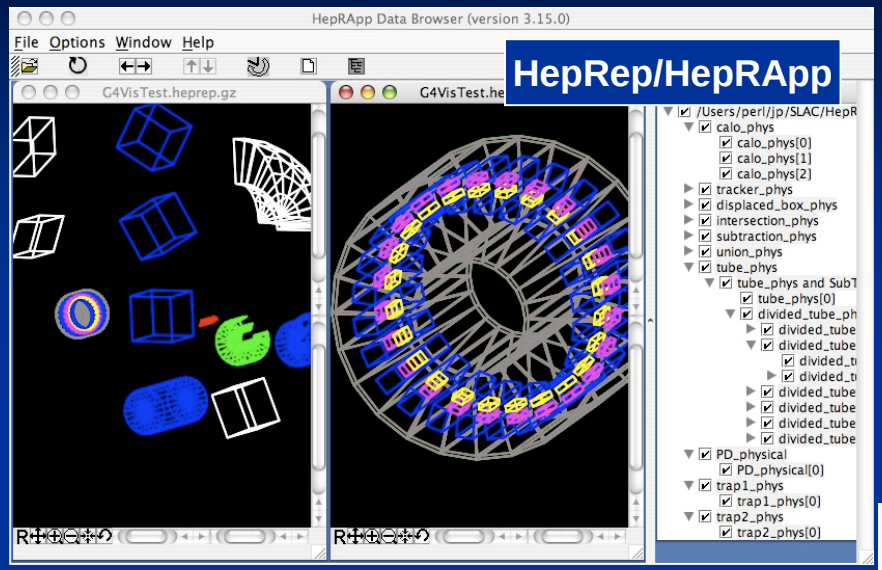
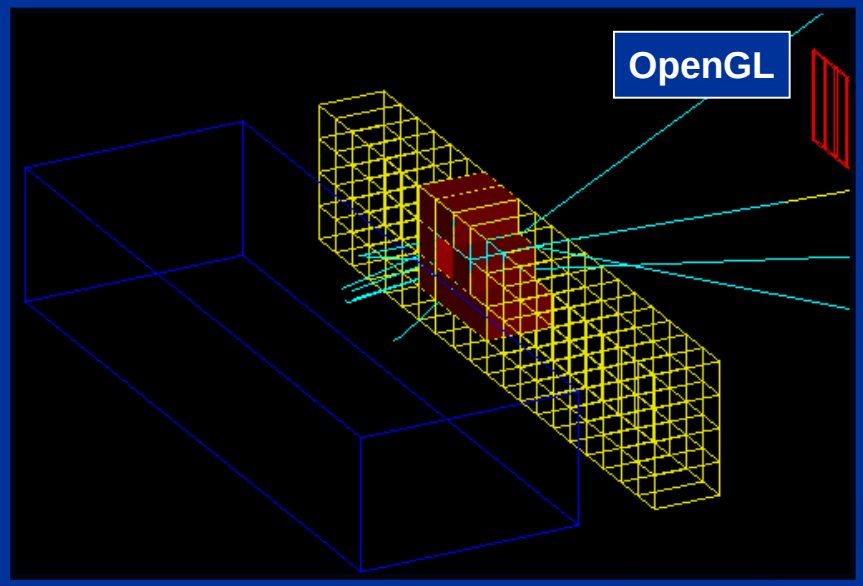
Geant4 Visualization Commands

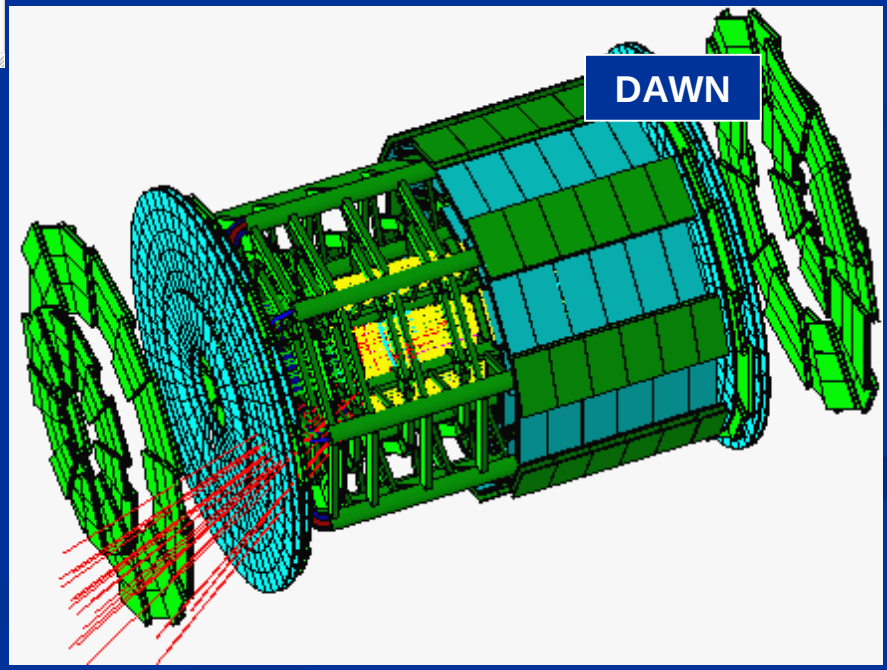
Paul Guèye, HU

Courtesy: Joseph Perl, SLAC

Basic Visualization Commands







How this Document Fits with Other Tutorial Materials

- This presentation can be used on its own, but gives the most comprehensive introduction to Geant4 visualization when used as part of the following full set of documents:
  - Introduction to Geant4 Visualization
  - Geant4 Installation Guides
  - Geant4 Visualization Tutorial using the HepRApp HepRep Browser
  - Geant4 Visualization Tutorial using the DAWN Event Display
  - Geant4 Visualization Tutorial using the OpenGL Event Display
  - Geant4 Visualization Commands
  - Geant4 Advanced Visualization
  - See the URLS at the end of this presentation

- This presentation discusses seven visualization drivers:
  - OpenGL
  - OpenInventor
  - HepRep
  - DAWN
  - VRML
  - RayTracer
  - ASCIITree

Visualization Commands

- Open a visualization driver, such as:
  - /vis/open HepRepFile
- Add the detector geometry
  - /vis/drawVolume
- If using an immediate viewer, such as OpenGL, set camera parameters and drawing style (wireframe/surface), such as:
  - /vis/viewer/set/style wireframe
  - /vis/viewer/set/viewpointThetaPhi 70 20
- Declare what data should be added to the scene (default is to just add full set of detector volumes)
  - /vis/scene/add/trajectories
  - /vis/scene/add/hits
- Run simulation with appropriate options to store trajectory information:
  - /run/beamOn 1
- Execute the visualization (done automatically with each /run/beamOn, but needed by some drivers if you want to output geometry without running an event):
  - /vis/viewer/flush
- If using an external viewer, such as for HepRepFile or DAWNFILE:
  - import the .heprep or .prim file into HepRApp or DAWN, set camera parameters, drawing style, etc., view the visualization

Example Visualization Command Sequences

- Visualize a detector in OpenGL (Linux or Mac):
  - /vis/open OGLIX
  - /vis/drawVolume
- Visualize a detector in OpenGL (Windows):
  - /vis/open OGLSWin32
  - /vis/drawVolume
- Visualize trajectories and hits for 10 events using HepRep/HepRApp
  - /vis/open HepRepFile
  - /vis/drawVolume
  - /vis/scene/add/trajectories
  - /vis/scene/add/hits
  - /run/beamOn 10

Command Guidance

- Complete guidance on all commands is available from the command line:
  - Idle> help
  - Command directory path : /
  - Sub-directories :
  - 1) /control/ UI control commands.
  - 2) /units/ Available units.
  - 3) /geometry/ Geometry control commands.
  - 4) /tracking/ TrackingManager and SteppingManager control commands.
  - 5) /event/ EventManager control commands.
  - 6) /run/ Run control commands.
  - 7) /random/ Random number status control commands.
  - 8) /particle/ Particle control commands.
  - 9) /process/ Process Table control commands.
  - 10) /vis/ Visualization commands.
  - 11) /mydet/ A01 detector setup control commands.
  - 12) /hits/ Sensitive detectors and Hits
  - 13) /gun/ Particle Gun control commands.
  - Commands :
  - Type the number (0:end, -n:n level back) :

Guidance Detail

- Guidance is hierarchical, providing full detail on all commands.
  - Sub-directories :
  - 1) /vis/ASCIITree/ Commands for ASCIITree control.
  - 2) /vis/GAGTree/ Commands for GAGTree control.
  - 3) /vis/heprep/ HepRep commands.
  - 4) /vis/rayTracer/ RayTracer commands.
  - 5) /vis/scene/ Operations on Geant4 scenes.
  - 6) /vis/sceneHandler/ Operations on Geant4 scene handlers.
  - 7) /vis/viewer/ Operations on Geant4 viewers.
  - Commands :
  - 8) enable * Enables/disables visualization system.
  - 9) disable * Disables visualization system.
  - 10) verbose * Simple graded message scheme - digit or string (1st character defines):
  - 11) drawTree * (DTREE) Creates a scene consisting of this physical volume and produces a representation of the geometry hierarchy.
  - 12) drawView * Draw view from this angle, etc.
  - 13) drawVolume * Creates a scene consisting of this physical volume and asks the current viewer to draw it.
  - 14) open * Creates a scene handler ready for drawing.
  - 15) specify * Draws logical volume with Boolean components, voxels and readout geometry.

- Command /vis/open
- Creates a scene handler ready for drawing.
- The scene handler becomes current (the name is auto-generated).
- Parameter : graphics-system-name
  - Parameter type : s
  - Omittable : False
  - Candidates : ATree DAWNFILE GAGTree HepRepXML HepRepFile RayTracer VRML1FILE VRML2FILE OGLIX OGLSX
- Parameter : window-size-hint
- pixels
  - Parameter type : i
  - Omittable : True
  - Default value : 600

Details of the /vis/open Command

- To Open a Driver
  - /vis/open <driver name>
- for example
  - /vis/open OGLIX
  - /vis/open HepRepFile
  - /vis/open DAWNFILE
- The set of available drivers is listed when you first start Geant4, but you can also get this list with the command:
  - help /vis/open

- You can open more than one driver at a time:
  - /vis/open OGLIX
  - /vis/open HepRepFile
  - /vis/viewer/list
  - /vis/viewer/select viewer-0
  - /vis/viewer/select viewer-1

Details of the /vis/viewer/… Commands

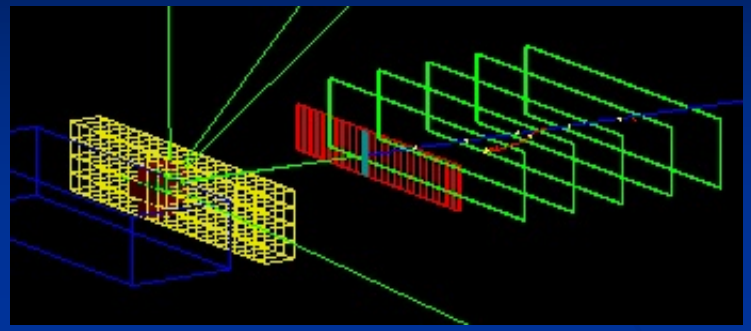
- To Set Camera Parameters and Drawing Style.
  - Only needed if using an immediate viewer, such as OpenGL
  - For HepRepFile or DAWNFILE, these sorts of adjustments are made later, in the HepRApp or DAWN viewer programs
- Reset viewpoint
  - /vis/viewer/reset
- Set view angles
  - /vis/viewer/set/viewpointThetaPhi <theta_angle> <phi_angle>
  - for example
    - /vis/viewer/set/viewpointThetaPhi 70 20
- Set drawing style
  - /vis/viewer/set/style <style>
  - for example
    - /vis/viewer/set/style wireframe
    - /vis/viewer/set/style surface
  - but note that this will not affect volumes that have style explicitly forced by “setForceWireframe” or “setForceSolid” commands in the c++ code

More Details of the /vis/viewer/… Commands

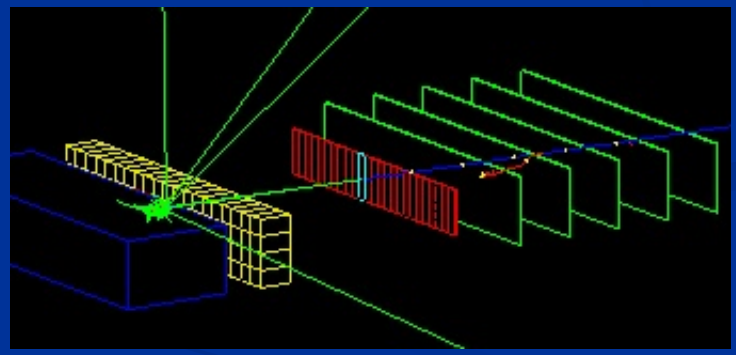
- Zoom
  - /vis/viewer/zoom <scale factor>
  - for example
    - /vis/viewer/zoom 2.
- Some drivers support different zoom along different axes (e.g., zoom more in X and Y but not in Z)
  - /vis/viewer/scale <3 vector of scale factors>
- Some drivers allow you to section the view, that is, cut it away along a specified plane (but this generally works only for simple geometries)
  - /vis/viewer/set/sectionPlane [on|off]
    <3 vector of point> [unit of point] <3 vector of plane normal>
  - e.g., for a y-z plane at x = 1 cm:
    /vis/viewer/set/sectionPlane on 1 0 0 cm 1 0 0

Hidden Line Removal

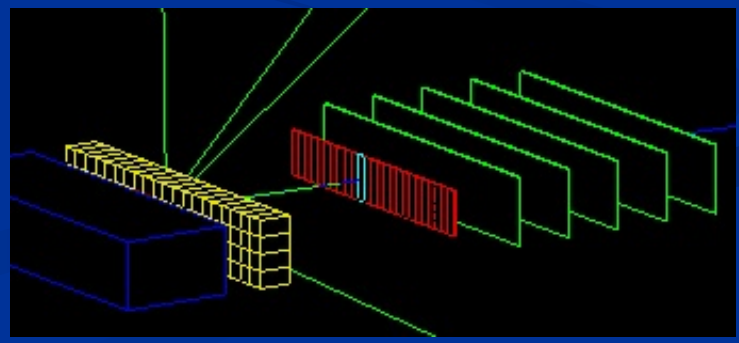
- OpenGL supports hidden line removal.
- You can control whether this removal is done and whether trajectories and hits are affected by this feature.
- By default, hidden line removal is disabled



- 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



Controlling Detail Level of Detector Geometry

- By default, /vis/drawVolume will draw the entire detector geometry. This is equivalent to the commands:
  - /vis/scene/create
  - /vis/scene/add/volume world
- You can specify additional arguments to limit the amount of geometry detail shown:
  - /vis/scene/add/volume [<physical-volume-name>] [<copy-no>] [<depth-of-descending>]
    - 1st parameter: volume name (default "world").
    - 2nd parameter: copy number (default -1 meaning first occurrence of physical-volume-name is selected.
    - 3rd parameter: depth of descending geometry hierarchy (default G4Scene::UNLIMITED (-1)).
- Still more arguments can be given to specify a clipping volume.
  - vis/scene/add/volume world -1 -1 box km 0 1 0 1 0 1 will draw the world with the positive octant cut away.

Even more Control over Level of Detail in Detector Geometry

- Additional commands allow finer control including whether or not to draw Boolean components, voxels and readout geometries:
  - /vis/specify <logical-volume-name> [depth-of-descent] [<booleans-flag>] [<voxels-flag>] [<readout-flag>]
  - /vis/scene/add/logicalVolume <logical-volume-name> [<depth-of-descending>] [<voxels-flag>] [<readout-flag>]

- Culling allows you to specify that covered daughters or low density volumes are omitted:
  - /vis/viewer/set/culling global|coveredDaughters|invisible|density [true|false] [density] [unit]
  - HepRepFile will still include these culled objects, but just make them initially invisible.
    - Idea is that you might later decide you want to see these.
    - To really omit them from the HepRepFile, as you may wish to do to make the file smaller, set the environment variable before you run:
      - G4HEPREPFILE_CULL=1

Details of Visualizing Trajectories and Hits

- To add trajectories or hits to the scene
  - /vis/scene/add/trajectories
  - /vis/scene/add/hits
- Run using the command
  - /run/beamOn
- If you place a number after beamOn, the run will go for that many events
  - /run/beamOn 10

Accumulating Trajectories and Hits

- By default, you will get a drawing after each event. To instead get just one drawing with all of the accumulated events from that run
  - /vis/scene/endOfEventAction accumulate
- This overrides the default
  - /vis/scene/endOfEventAction refresh

- To even suppress that one drawing from the end of the /run/beamOn, use
  - /vis/scene/endOfRunAction accumulate
- This overrides the default
  - /vis/scene/endOfRunAction refresh
- When you actually want to draw, you then have to explicitly issue the command
  - /vis/viewer/flush

Reviewing Kept Events

- If you have accumulated several events in your visualization, you can still go back afterwards and view the events individually. For each event, you can execute various vis commands to rotate, zoom, output to a different vis driver, etc.
  - /vis/reviewKeptEvents
  - Each time you type “continue”, you will get to the next kept event.
- To quit reviewing events:
  - /vis/abortReviewKeptEvents
  - and then again type “continue”

- You can also use a command or c++ calls to force keeping of specific events regardless of how visualization is accumulating them.
  - e.g., keep events based on a particular hit or trigger pattern
- From the command line:
  - /event/keepCurrentEvent
- From C++
  - G4EventManager->KeepTheCurrentEvent()
- This feature makes it easy to do a large run and then recall for visualization only those events that are of interest

Compound Commands

- To allow you to work quickly, Geant4 visualization lets you issue the equivalent of several common commands at one time by using a “compound command”.
- Some of the commands you have already seen in this presentation are actually compound commands:
  - /vis/open
    - /vis/sceneHandler/create
    - /vis/viewer/create
  - /vis/drawVolume
    - /vis/scene/create
    - /vis/scene/add/volume
  - /vis/viewer/flush
    - /vis/viewer/refresh
    - /vis/viewer/update

Geant4 Visualization in Standalone Mode

- The Geant4 Visualization system can be used on its own - without the rest of Geant4.
- Build something “by hand” from the Geant4 geometry primitives and placement apparatus, but without any of the main parts of Geant4 such as detector construction, run manager or physics list.
- Still preserves all of the interactive apparatus of the visualization system.
- From Geant4 release 8.0 there is an example: /examples/extended/visualization/standalone

```
// Simple box...
    pVisManager->Draw(      G4Box("box",2*m,2*m,2*m),
                                G4VisAttributes(G4Colour(1,1,0)));

// Boolean solid...
    G4Box boxA("boxA",3*m,3*m,3*m);
    G4Box boxB("boxB",1*m,1*m,1*m);
    G4SubtractionSolid subtracted(      "subtracted_boxes",&boxA,&boxB,
                                G4Translate3D(3*m,3*m,3*m));
    pVisManager->Draw(      subtracted,
                            G4VisAttributes(G4Colour(0,1,1)),
                                G4Translate3D(-6*m,-6*m,-6*m));
```

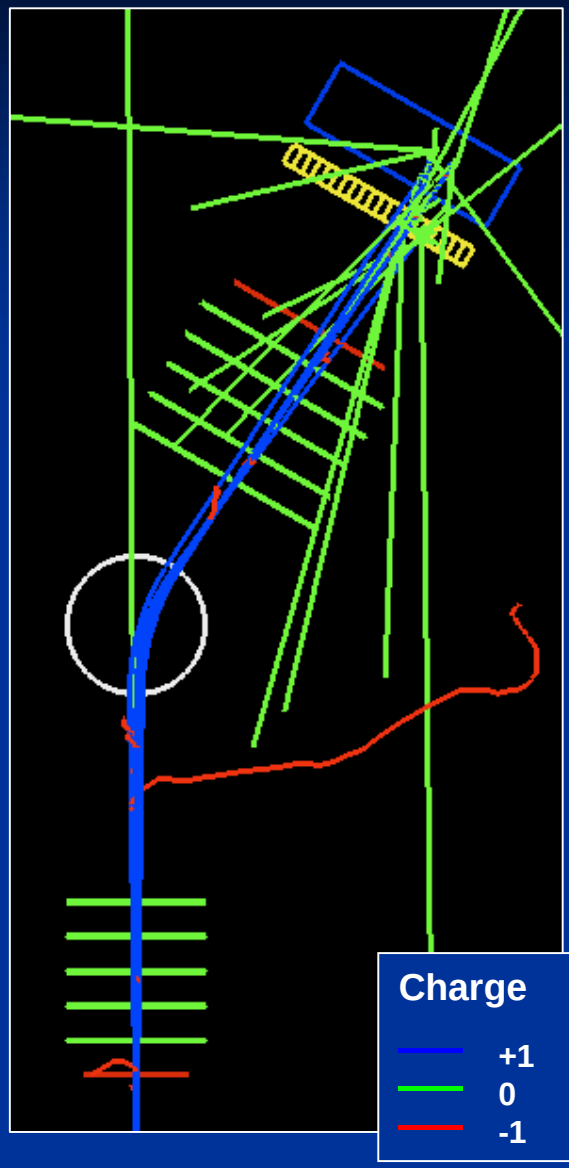
Complete List of Commands

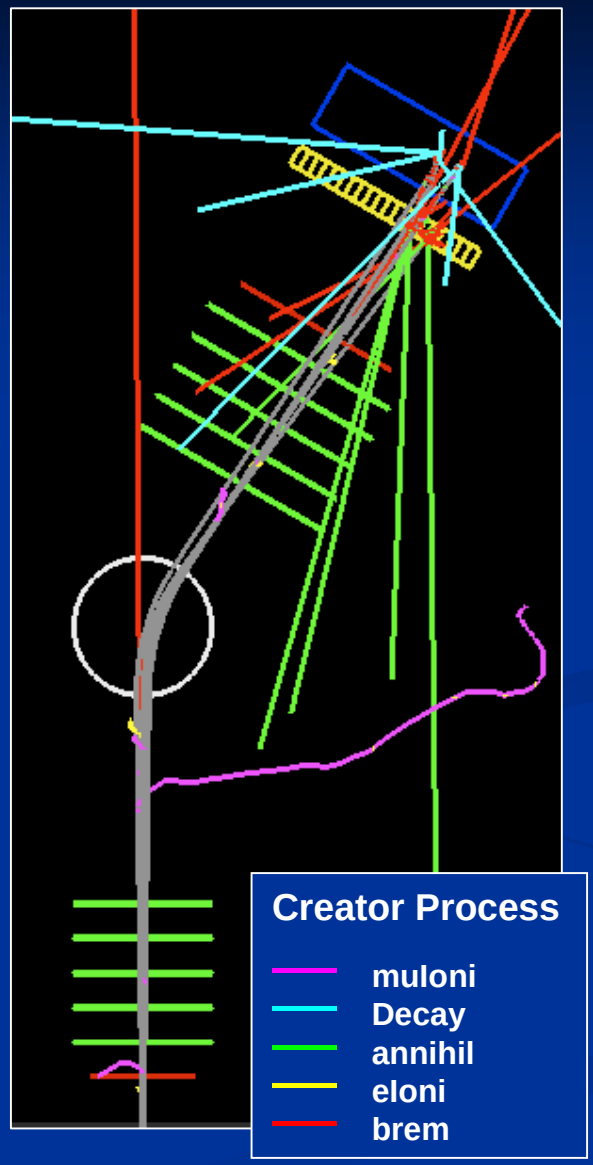
- This presentation has shown only a very small subset of Geant4 vis commands. Even for those commands shown, only a few of the options have been presented.
- Each visualization driver may have its own set of additional commands.
- To see the complete set of commands, use the interactive command guidance (i.e., type help and then type the appropriate number for “vis”).
- Note that many of the command details are only loaded into the help system once you start using the given command
  - e.g., when you first look at the help for /vis/modeling, you will see only
    - /vis/modeling/trajectories/create
    - /vis/modeling/trajectories/list
  - But once you have done your first
    - /vis/modeling/trajectories/create/drawByParticleID
  - you will see many subcommands such as
    - /vis/modeling/trajectories/drawByParticleID-0/set
    - /vis/modeling/trajectories/drawByParticleID-0/setRGBA
    - etc.

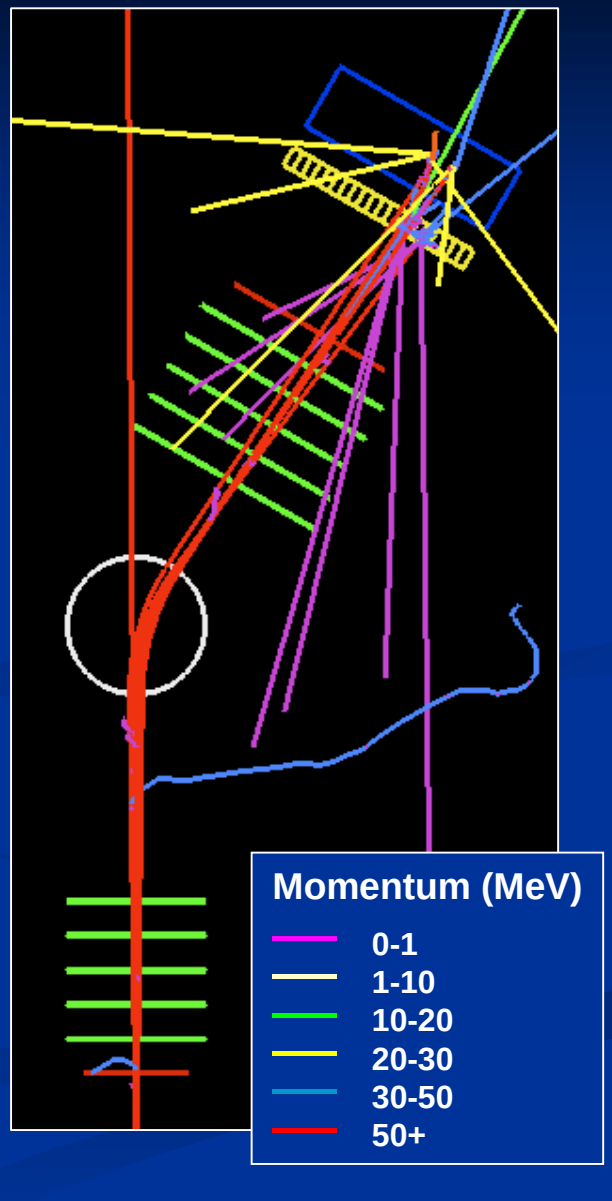
And More ….

See: http://geant4.slac.stanford.edu/SLACTutorial07/Visualization3.ppt

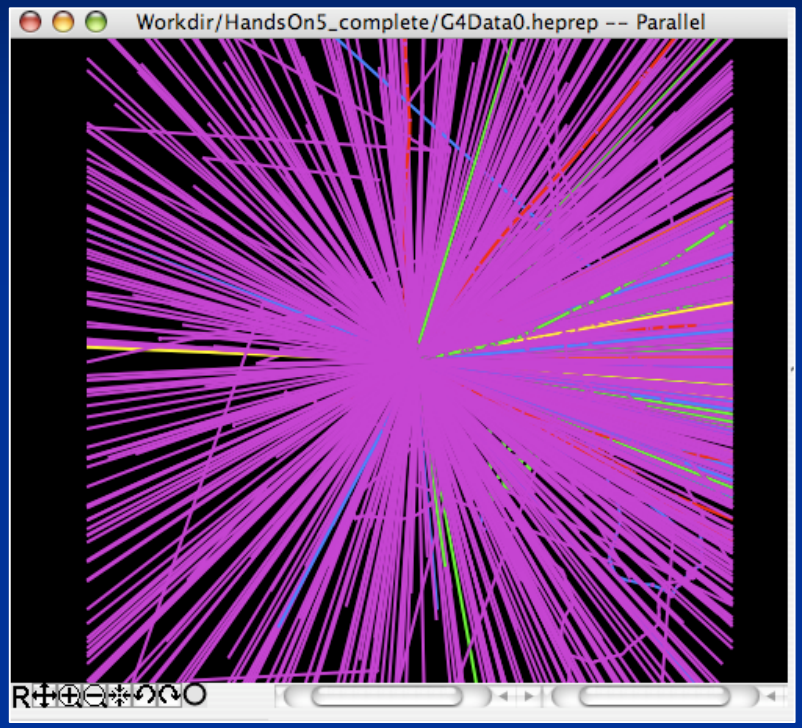
Example A01, five events, drawBy various models





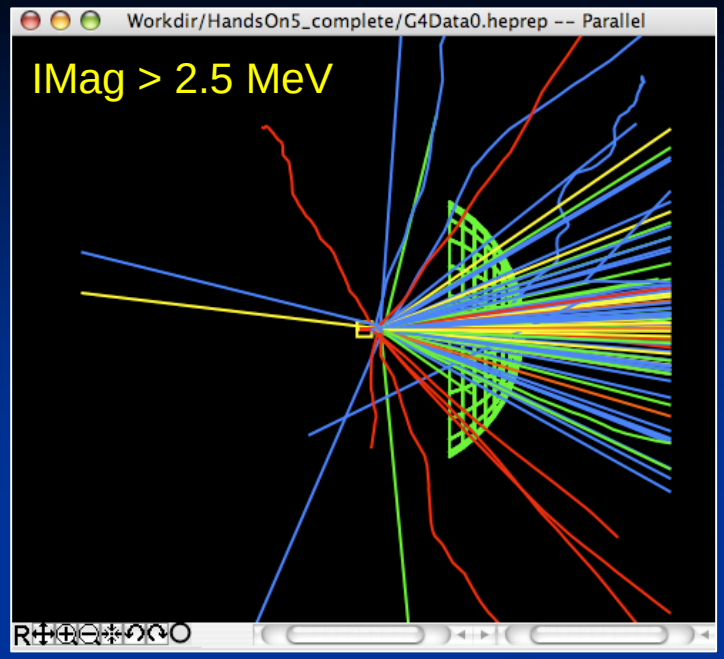


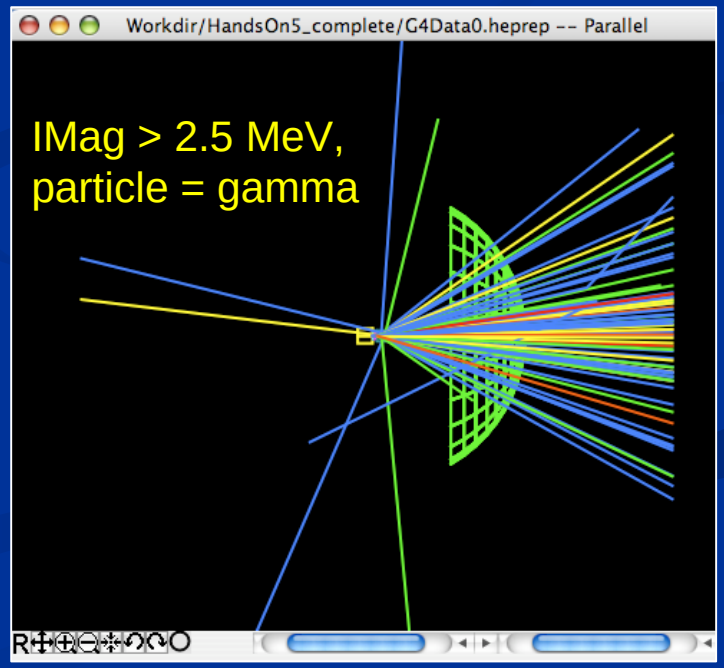
HandsOn5, McGill tutorial, 1000 events, Attribute Filter



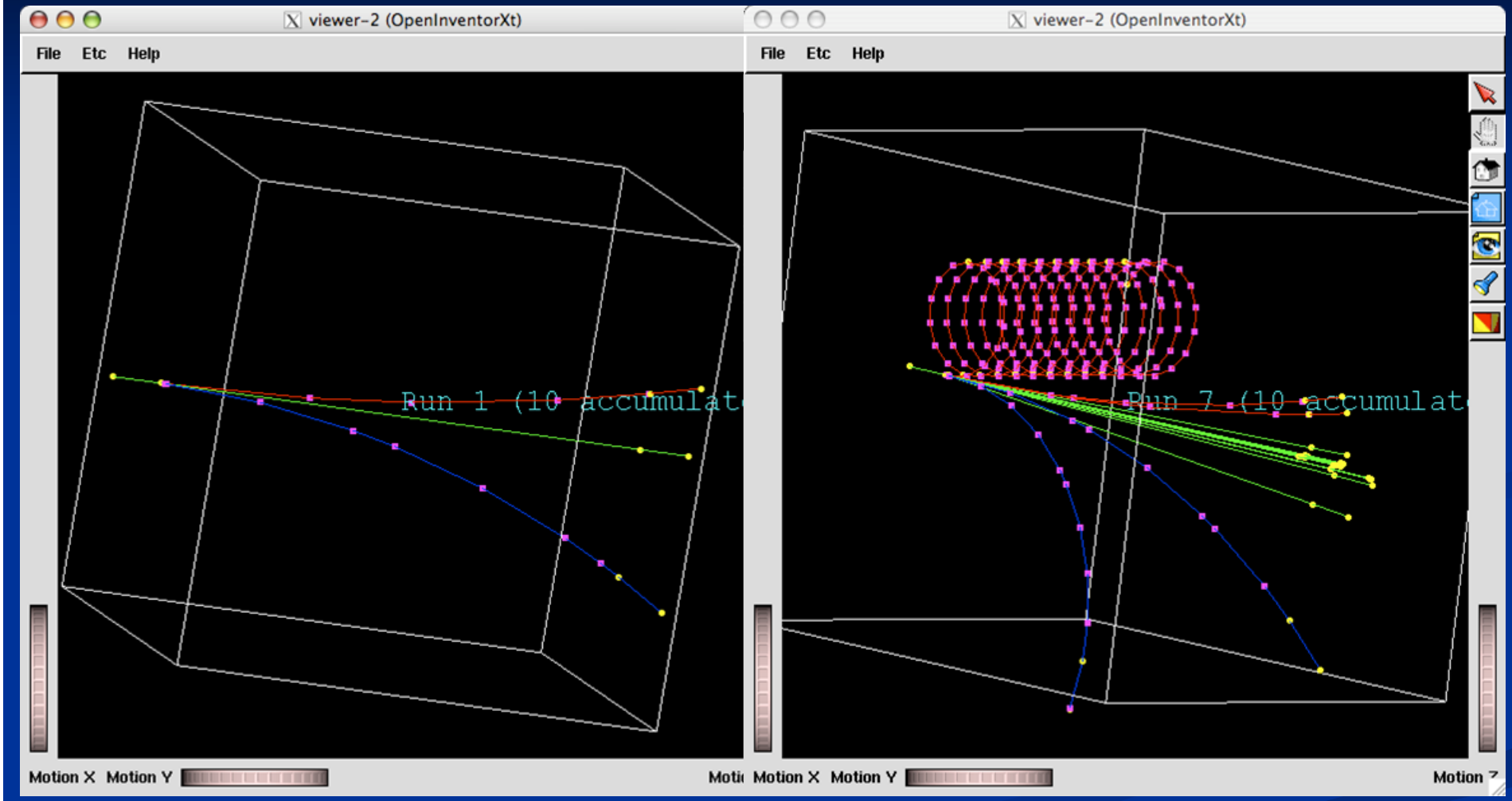
Momentum (MeV)

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





Smooth Trajectory Makes Big Difference for Trajectories that Loop in a Magnetic Field



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

Geant4 Visualization Resources

Geant4 Installation Guides

Øhttp://geant4.slac.stanford.edu/installation

Hands on HepRApp Tutorial

Øhttp://geant4.slac.stanford.edu/Presentations/vis/G4HepRAppTutorial/G4HepRAppTutorial.html

Hands on DAWN Tutorial

Øhttp://geant4.slac.stanford.edu/Presentations/vis/G4DAWNTutorial/G4DAWNTutorial.html

Hands on OpenGL Tutorial

Øhttp://geant4.slac.stanford.edu/Presentations/vis/G4OpenGLTutorial/G4OpenGLTutorial.html

Geant4 Visualization Commands

Øhttp://geant4.slac.stanford.edu/Presentations/vis/G4VisCommands.ppt (and .pdf)

Geant4 Advanced Visualization

Øhttp://geant4.slac.stanford.edu/Presentations/vis/G4VisAdvanced.ppt (and .pdf)

How to Make a Movie

Øhttp://geant4.slac.stanford.edu/Presentations/vis/HowToMakeAMovie.ppt (and .pdf)

Visualization Chapter of the Geant4 User's Guide for Application Developers

Øhttp://geant4.web.cern.ch/geant4/UserDocumentation/UsersGuides/ForApplicationDeveloper/html/

List of Visualization Commands:

Øhttp://geant4.web.cern.ch/geant4/UserDocumentation/UsersGuides/ForApplicationDeveloper/html/AllResources/Control/UIcommands/_vis_.html

For Questions or Comments: Geant4 Visualization Online Forum:

Øhttp://geant4-hn.slac.stanford.edu:5090/HyperNews/public/get/visualization.html