



Version 11.2.1

Geant4 Tutorial Course

Jefferson Lab, Newport News, VA, USA
March 25-29 2024

Visualization

Maurizio Ungaro
Jefferson Lab

Original Slides created by Joseph Perl (SLAC)



Version 11.2.1

Geant4 Tutorial Course

Jefferson Lab, Newport News, VA, USA
March 25-29 2024

Visualization

Maurizio Ungaro
Jefferson Lab

Original Slides created by Joseph Perl (SLAC)

Introduction

- ◆ G4 vis is an object-oriented multi-driver system. The core **vis** system converts the G4 scene (geometry, trajectories, etc.) into G4 graphics primitives. A driver renders the primitives and brings additional features, such as clipping, volume rendering, file export, etc.
- ◆ You can choose your driver, even have different drivers in the same interactive session, and multiple views of the G4 scene.

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 also be visualized:
 - Polylines
 - coordinate axes
 - 3D Markers
 - Eye guides
 - Text
 - descriptive character strings
 - comments or titles

New or Improved and retired VIS drivers

New in Geant4 11.0 and further developments

- ◆ Qt3D (*John Allison*): limited functionality but nice
- ◆ Open Inventor Qt Viewer
- ◆ ToolsSG (TSG) (*Guy Barrand*): working nicely
 - Most features of the OpenGL drivers
 - Also supports histos plotting
 - Full-screen driver, TOOLSSG_OFFSCREEN—always built, default in batch mode
- ◆ **Retired (removed) in Geant4 11.1**
 - HepRep/Wired (HepRepFile/HepRApp is retained)
 - VRML1 (VRML2 is retained)
 - The “network” drivers (those that communicate with their browser via BSD sockets)
 - VRML2 (VRMK2FILE is retained)
 - DAWN (DAWNFILE is retained)
- ◆ **Work in progress:**
 - Qt6. Migration in progress - quite problematic, it's a rewrite.
 - VTK (very promising, multi-lab support, mobile version)
- ◆ **Future (and warning):**
 - Apple DEPRECATED OpenGL in favor of Metal. We have to look.

A variety of choices

Driver	Variant	High Quality Print	Interactive	Geo Hierarchies	Access to G4 Kernel	Movies
OpenGL	X	Green	Green	Red	Green	Green
	Xm	Green	Green	Red	Green	Green
	Qt	Green	Green	Green	Green	Green
	Win32	Green	Green	Red	Green	Green
OpenInventor	Xt	Green	Green	Red	Green	Red
	Win32	Green	Green	Red	Green	Red
DAWNFILE		Green	Red	Red	Red	Red
VRML2		Red	Green	Red	Red	Red
HepRepFile		Red	Green	Green	Red	Red
gMocren		Red	Green	Red	Red	Red
RayTracer		Green	Red	Red	Red	Red
ASCII		Red	Red	Green	Red	Red

Comput. Phys. Comm. 178 (2008) 331-365


Controlling visualization

- ◆ Your Geant4 code stays basically the same no matter which driver you use
- ◆ Visualization is performed either with UI 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
 - ASCIITree
- ◆ 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 drivers 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

- ◆ Many visualization drivers are included by default (since they require no external libraries):
 - HepRepFile
 - DAWNFILE
 - VRML2FILE
 - RayTracer
 - gMocrenFile
 - ASCIITree
- ◆ Other visualization drivers are included only if you request them in your cmake options.
 - OpenGL
 - **Qt with OpenGL**: Requires that you install Qt. **Have customizable interface.**
 - Open Inventor Qt (OIQt). Requires additional software.
- ◆ 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.

 **NOTE:** `/vis/list` will show the registered graphics systems

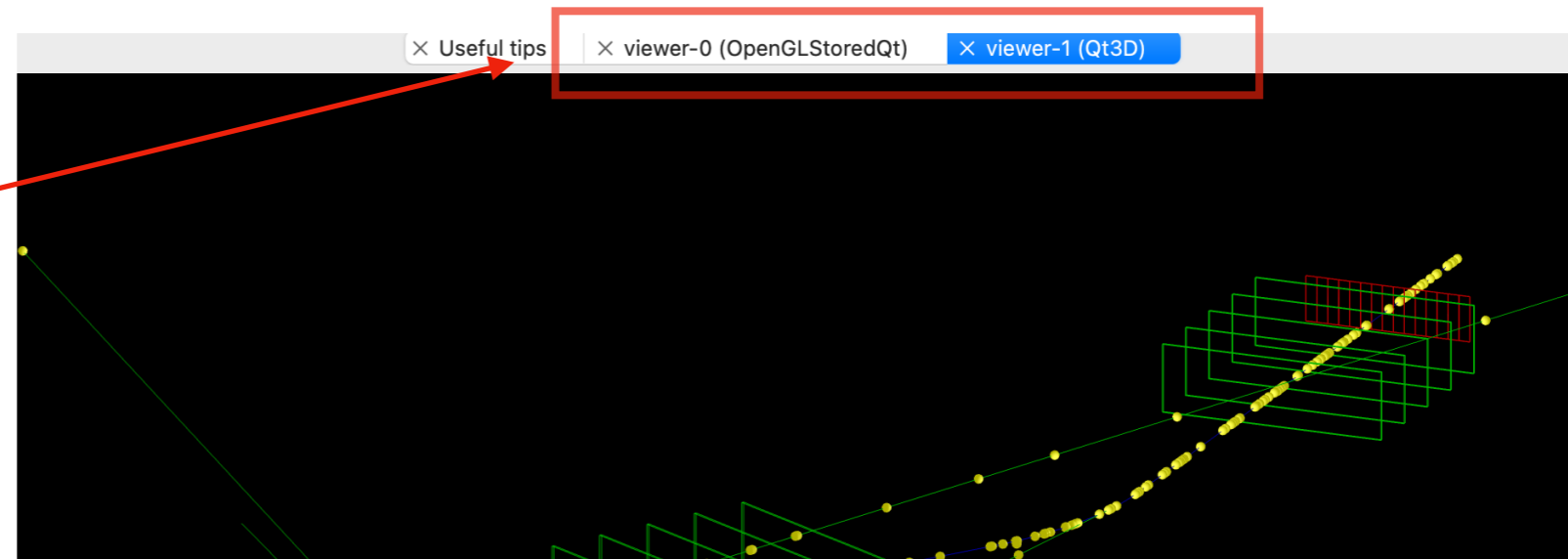
Simple Command Example

- ◆ Open a driver (i.e. OGL, RayTracer, ASCIIITree)

- `/vis/open <drivername>`

- ◆ Visualize your geometry in OpenGL (default: OpenGLStoredQt if Qt is installed)

- `/vis/open OGL`
- `/vis/open Qt3D`
- `/vis/drawVolume`



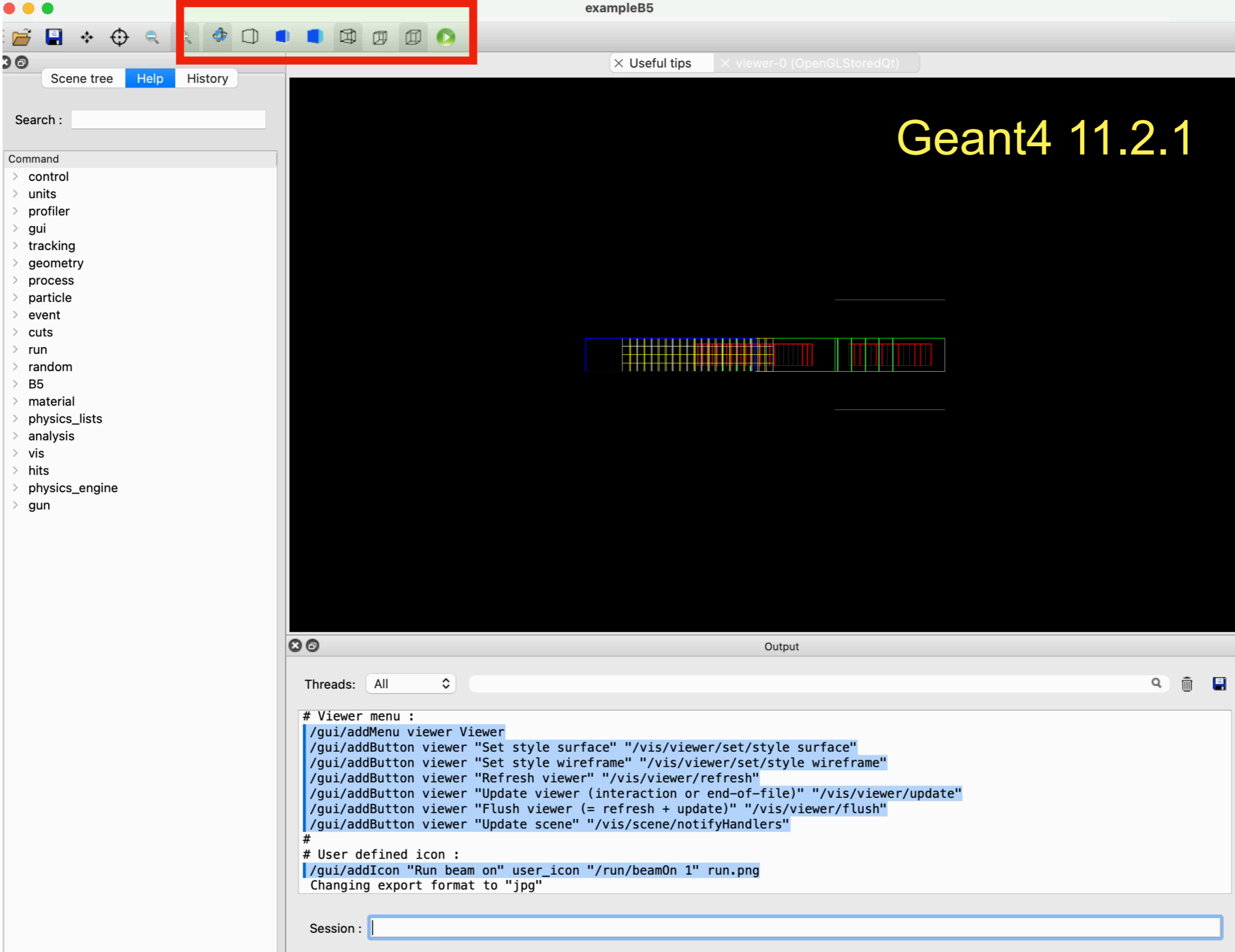
- ◆ Most examples come with a visualization macro

- Typically named vis.mac
- Checkout the examples, in particular:

- **B1**: basic visualization
- **B5**: basic plotting of quantities via ToolsSG (`/control/execute plotter.mac`)

Qt Driver with OpenGL visualization

Interactivity with mouse: rotate, move, zoom, style, right click



The screenshot displays the Geant4 Qt Driver interface. The main window, titled "exampleB5", shows a 3D visualization of a particle detector structure. The text "Geant4 11.2.1" is displayed in the upper right corner of the visualization area. A toolbar at the top of the window contains various icons for interaction, with a red box highlighting the rotation, translation, and zoom controls. On the left side, there is a "Scene tree" panel with a search bar and a list of commands. The "Output" panel at the bottom shows a list of commands and their execution status.

```
# Viewer menu :  
/gui/addMenu viewer Viewer  
/gui/addButton viewer "Set style surface" "/vis/viewer/set/style surface"  
/gui/addButton viewer "Set style wireframe" "/vis/viewer/set/style wireframe"  
/gui/addButton viewer "Refresh viewer" "/vis/viewer/refresh"  
/gui/addButton viewer "Update viewer (interaction or end-of-file)" "/vis/viewer/update"  
/gui/addButton viewer "Flush viewer (= refresh + update)" "/vis/viewer/flush"  
/gui/addButton viewer "Update scene" "/vis/scene/notifyHandlers"  
#  
# User defined icon :  
/gui/addIcon "Run beam on" user_icon "/run/beamOn 1" run.png  
Changing export format to "jpg"
```

Qt Driver with OpenGL visualization

Interactive commands to G4 Kernel, kernel suggestions

The screenshot displays the Geant4 Qt Driver interface. On the left, a 'Scene tree' panel lists various components like 'control', 'units', 'profiler', 'gui', etc. The main window shows a 3D visualization of particle tracks in a dark environment. Below the visualization is an 'Output' window showing a list of commands and their parameters. A red box highlights the command suggestions in the 'Session' input field, which are listed as a tree structure starting with '/control/'.

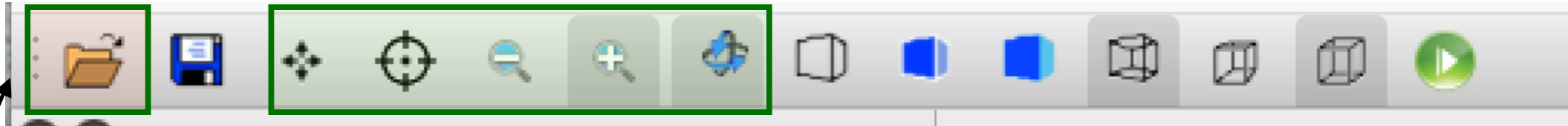
```
# Viewer menu :  
/gui/addMenu viewer Viewer  
/gui/addButton viewer "Set style surface" "/vis/viewer/set/style surface"  
/gui/addButton viewer "Set style wireframe" "/vis/viewer/set/style wireframe"  
/gui/addButton viewer "Refresh viewer" "/vis/viewer/refresh"  
/gui/addButton viewer "Update viewer (interaction or end-of-file)" "/vis/viewer/update"  
/gui/addButton viewer "Flush viewer (= refresh + update)" "/vis/viewer/flush"  
/gui/addButton viewer "Update scene" "/vis/scene/notifyHandlers"  
#  
# User defined icon :  
/gui/addIcon "Run beam on" user_icon "/run/beamOn 1" run.png  
Changing export format to "jpg"  
  
Session : /  
/control/  
/units/  
/profiler/  
/gui/  
/tracking/  
/geometry/  
/process/  
/particle/  
/event/  
/cuts/  
/run/  
/random/  
/B5/
```

- Commands suggestions as you type
- Parameters help for the command



Qt Driver with OpenGL visualization

Toolbar and menubar controlled by icons.mac file, add your own without coding



Open
Macro
File

```
# open/save icons
/gui/addIcon "Open macro file" open /control/execute
/gui/addIcon "Save viewer state" save /vis/viewer/save

# Cursors style icons
/gui/addIcon "Move" move
/gui/addIcon "Pick" pick
/gui/addIcon "Zoom out" zoom_out
/gui/addIcon "Zoom in" zoom_in
/gui/addIcon "Rotate" rotate

# Surface Style icons
# Surface Style icons
/gui/addIcon "Hidden line removal" hidden_line_removal
/gui/addIcon "Hidden line and hidden surface removal" hidden_line_and_surface_removal
/gui/addIcon "Surfaces" solid
/gui/addIcon "Wireframe" wireframe

# Perspective/Ortho icons
/gui/addIcon "Perspective" perspective
/gui/addIcon "Orthographic" ortho
```

Available Default Commands:

```
open save move rotate pick zoom_in zoom_out wireframe solid hidden_line_removal
hidden_line_and_surface_removal perspective ortho exit
```

Qt Driver with OpenGL visualization

(Thread selectable) Output from G4 Kernel

The screenshot displays the Geant4 Qt Driver interface. On the left is a scene tree with a search bar and a command list. The main window shows a 3D visualization of particle tracks (green lines) and detector components (blue and red boxes) on a black background. At the bottom, an output log window shows the following text:

```
Thread: All
G4WT8 > Layer[4] : time 41.555718686334 (nsec) --- local (x,y) -285.12084428753, -9.5765250770173
G4WT8 > EM Calorimeter has 9 hits. Total Edep is 920.24546579154 (MeV)
G4WT8 > Hadron Calorimeter has 0 hits. Total Edep is 0 (MeV)
G4WT4 > Drift Chamber 2 has 5 hits.
G4WT9 > Layer[1] : time 8.3390450887793 (nsec) --- local (x,y) 18.579194896481, 4.9148781219574
G4WT9 > Layer[2] : time 10.006920981662 (nsec) --- local (x,y) 21.798992213686, 7.3993983174673
G4WT9 > Layer[3] : time 11.674799468178 (nsec) --- local (x,y) 25.385437479748, 9.5115867856361
G4WT9 > Layer[4] : time 13.342684893914 (nsec) --- local (x,y) 29.315309837406, 11.499661359699
G4WT9 > Drift Chamber 2 has 5 hits.
G4WT10 > ... set ntuple merging row mode : row-wise - done
G4WT4 > Layer[0] : time 34.730300171222 (nsec) --- local (x,y) -206.58292859963, -0.97864111892766
G4WT5 > Layer[0] : time 35.817348668063 (nsec) --- local (x,y) -298.49155045027, -177.36695350034
```

Select thread

Search log

Qt Driver with OpenGL visualization

Help tree browser, with text search, history

Scene tree Help History

Search : logo

Command	Match
vis/scene/add/logo	
vis/scene/add/logo2D	

Command /vis/scene/add/logo
Guidance : Adds a G4 logo to the current scene.
If "unit" is "auto", height is roughly one tenth of scene extent.
"direction" is that of outward-facing normal to front face of logo.
If "direction" is "auto", logo faces the user in the current viewer.

If "placement" is "auto", logo is placed at bottom right of screen when viewed from logo direction.

	Parameter	iuidanc	Type	Ommittable	Default	Range	Candidate
1	height		d	True	1		
2	unit		s	True	auto		
3	direction	auto [-]x [-]y [-]z	s	True	auto		
4	red		d	True	0		
5	green		d	True	1		
6	blue		d	True	0		
7	placement		s	True	auto		auto manual
8	xmid		d	True	0		
9	ymid		d	True	0		
10	zmid		d	True	0		
11	unit		s	True	m		

Moveable borders

Search Field (press enter)
Search Results

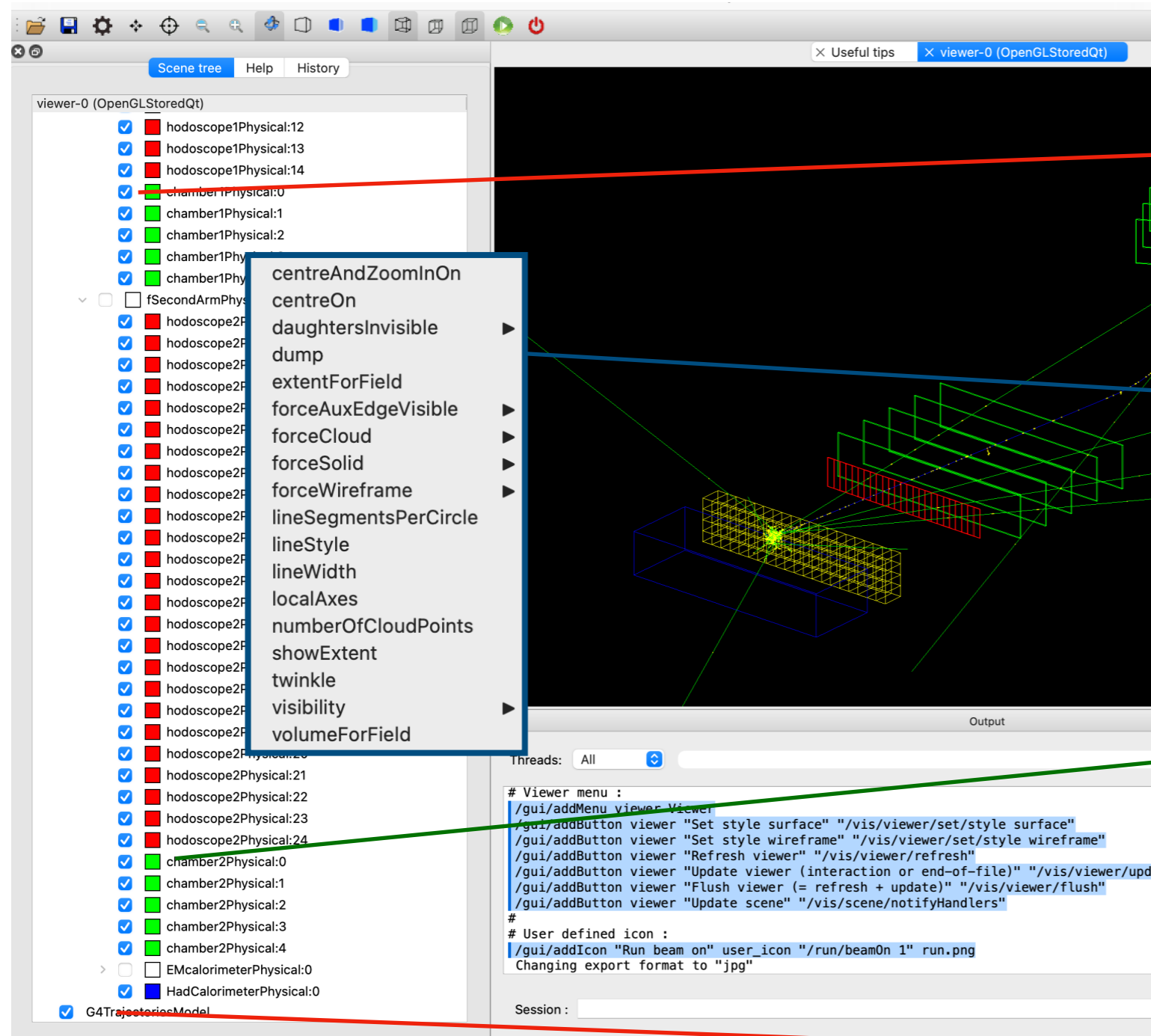
Selection Help
Command description

Parameters Help

Guidance
Omittable boolean
Default value
Range

Qt Driver with OpenGL visualization

Scene tree: edit visualization attributes



Mouse click to show / hide

Right click for many more options

Double mouse click to change color

Mouse click to show / hide trajectories

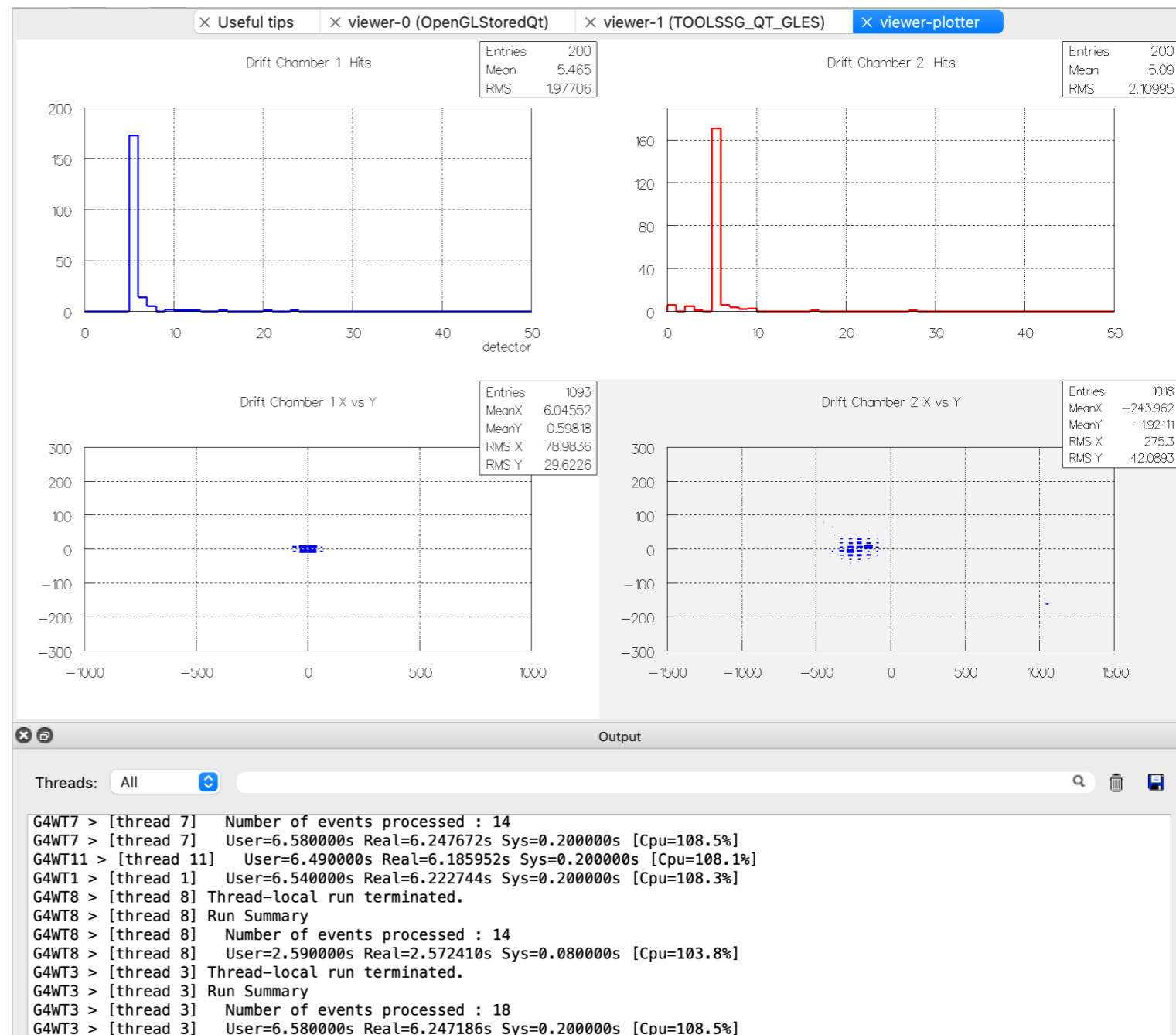
Right / click dump for properties

ToolsSG Driver

◆ Introduced in 2021, now part of g4tools

- Binds to various renderers: OpenGL, X11, Xt, Windows, Qt5, Qt6
- Permits some plotting
- Used in G4/analysis for batch plotting
- See `plotter.mac` in B5

```
/vis/open TSGQt  
/control/execute plotter.mac  
/run/beamOn 200
```



Open Inventor Qt Viewer

◆ Released in Geant4 11.0

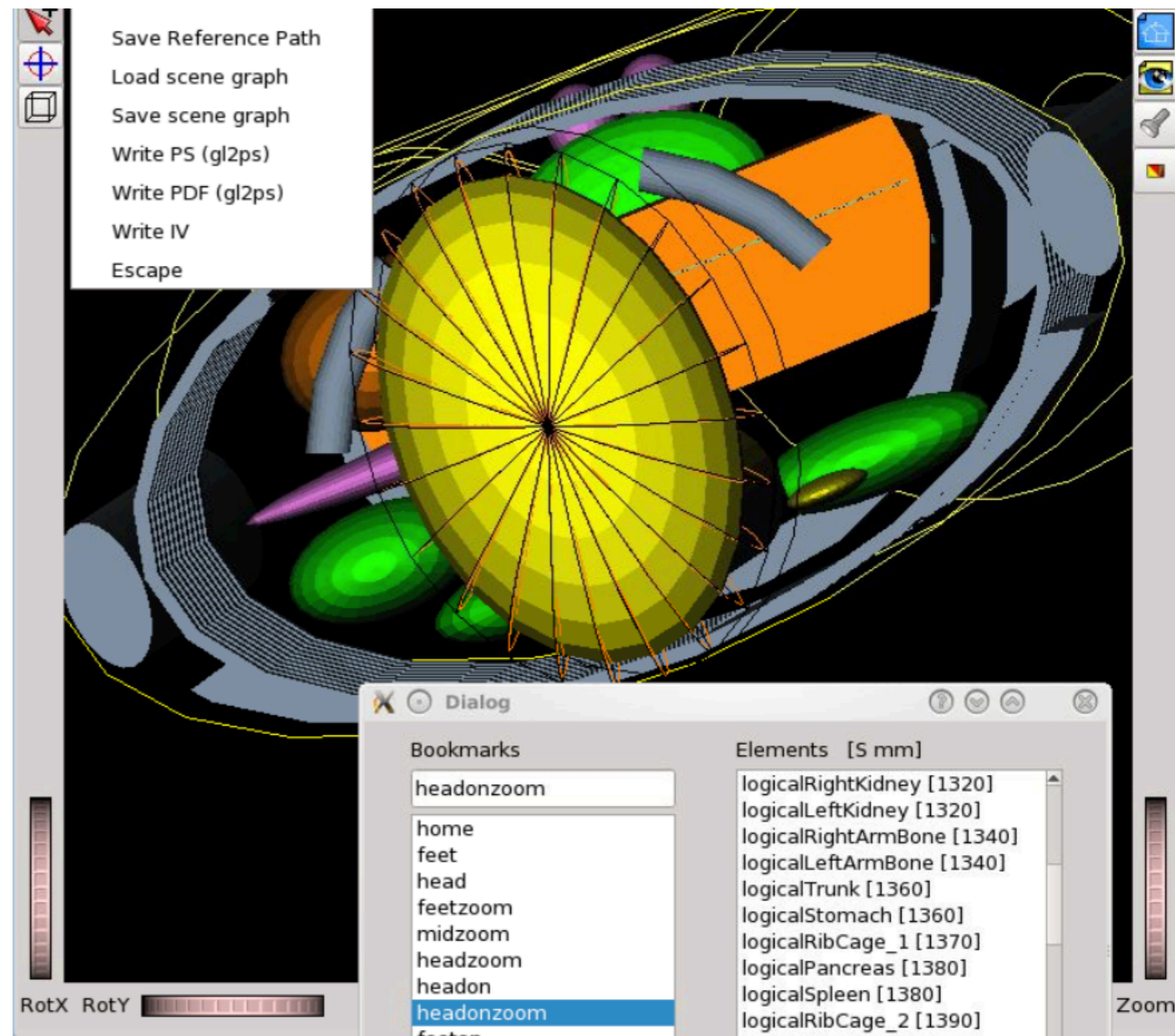
◆ How to use:

- Install Qt5
- Install Coin3D
- Build with cmake
-DGEANT4_USE_OPENINVENTOR_QT

```
/vis/open 0I
```

◆ Some unique capabilities

- Full 3d manipulation and interactivity via mouse/trackpad,
- Navigation along a reference path defined by a trajectory or by a 3d polyline, with precise camera movements and rotations.
- fly-through animation, seek function, mouse-over readouts of volumes and trajectories, operates within the Qt UI or can be detached from it as a resizable window.



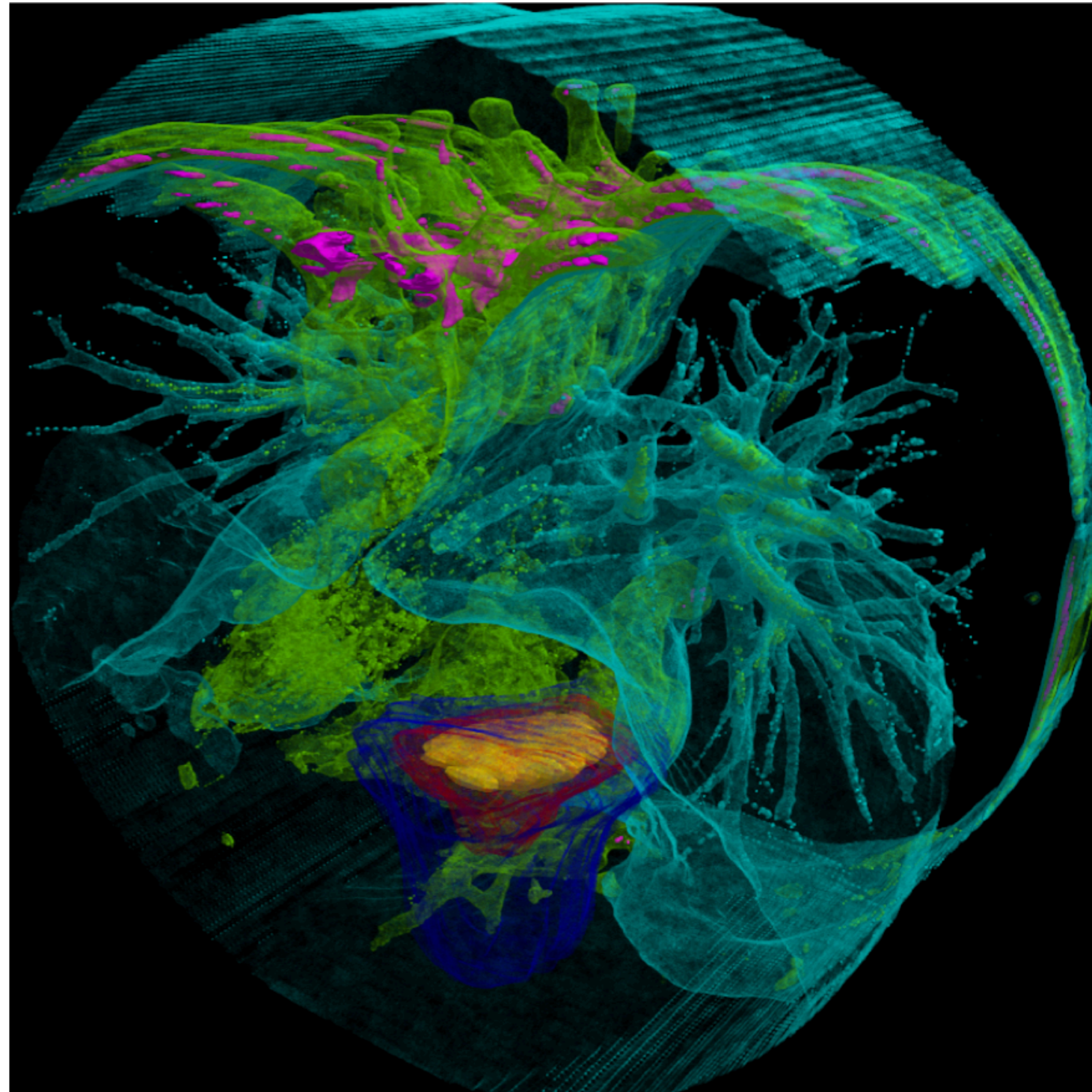
gMocren Visualization Driver

- ◆ From JST/CREST project (Japan) to improve Geant4 for medical physics
- ◆ Based on a commercial package but offered freely to all Geant4 users
- ◆ How to use:
 - Install gMocREn
 - Export visualization to gMocren Files:

```
/vis/scene/create  
/vis/open gMocrenFile  
/vis/scene/add/psHits  
/vis/viewer/flush  
/run/beamOn 1
```

- ◆ Issues with new Os being workout

Dose on human tissue



- ◆ For more information check the [Geant4 Visualization Driver Help on gMocren](#)

ASCII Tree

- ◆ 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
- ◆ How to use:

```
/vis/open ATree
```

```
/vis/viewer/flush
```

```
/vis/ASCIITree/verbose <verbosity>
```

```
/vis/viewer/flush
```

Print infos

Set verbosity level

```
"cellPhysical":76 /  
  "cellLogical" (SD="EMcalorimeter") /  
    "cellBox"(G4Box), 6.75 L , 4.51 g/cm3  
      (G4_CESIUM_IODIDE),  
        6.75 L , 30.4425 kg
```

```
"HadCalorimeterPhysical":0 /  
  "HadCalorimeterLogical" /  
    "HadCalorimeterBox"(G4Box), 1.8 m3 , 11.35 g/cm3  
      (G4_Pb),  
        1.8625e-07 mm3, 2.11393e-06 mg
```

Calculating mass(es)...

Overall volume of "worldPhysical":0, is 2400 m3
and the daughter-included mass to unlimited depth
is 22056.2 kg

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 total mass and volume

5: show individual volume masses and volumes

6: include volume transformations, more formatted log

7: include facets counts

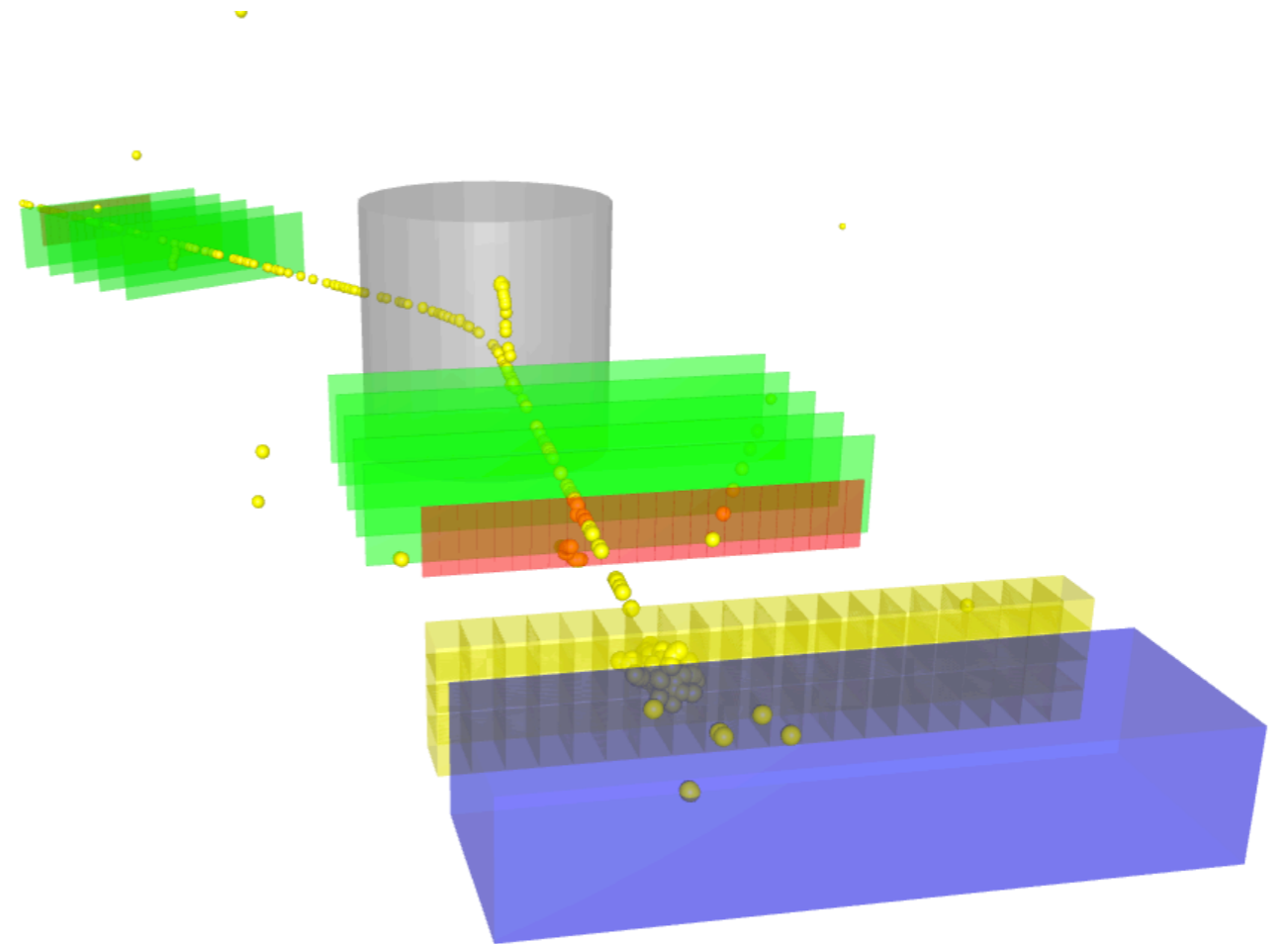
+10 to show repeated placements and replicas.

VRML2 Visualization Driver

- ◆ Requires VRML browser (many different choices for different operating systems)
 - App example [FreeWRL](#)
 - Online example: [3dviewer.net](#)
- ◆ Create a file to view in any VRML browser (some as web browser plug-ins)
- ◆ Rendered, photorealistic image with some interactive features
 - zoom, rotate, translate
- ◆ Limited printing ability (pixel graphics, not vector graphics)
- ◆ How to use:

```
/vis/open VRML2FILE  
  
/vis/viewer/flush
```

A WRL file will be created

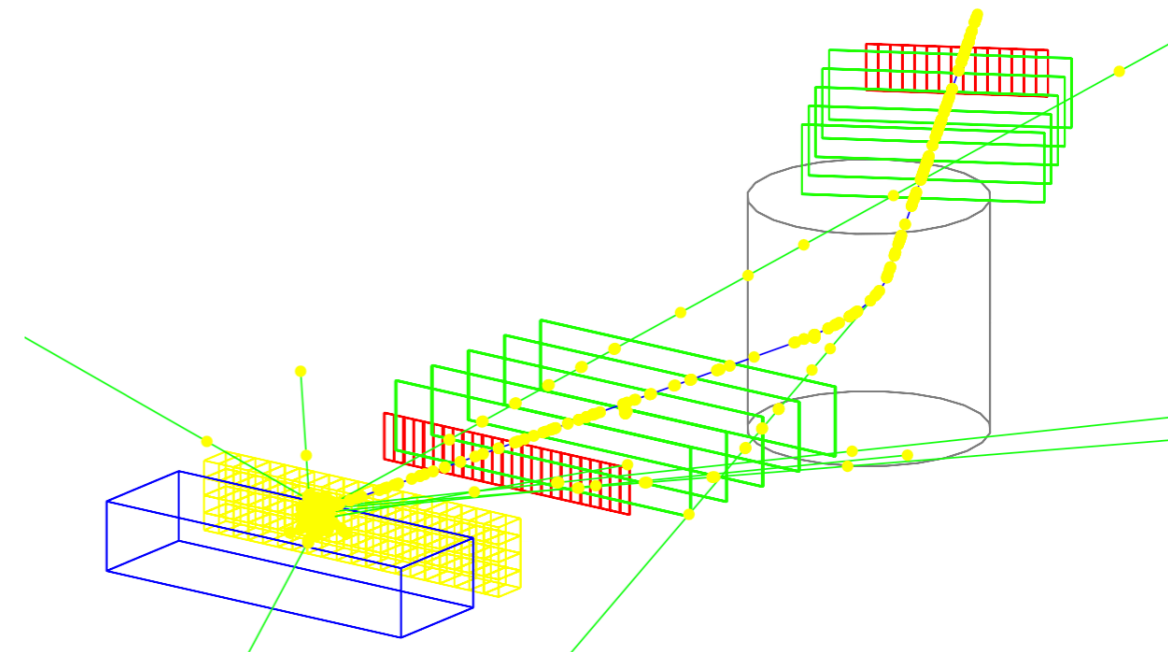
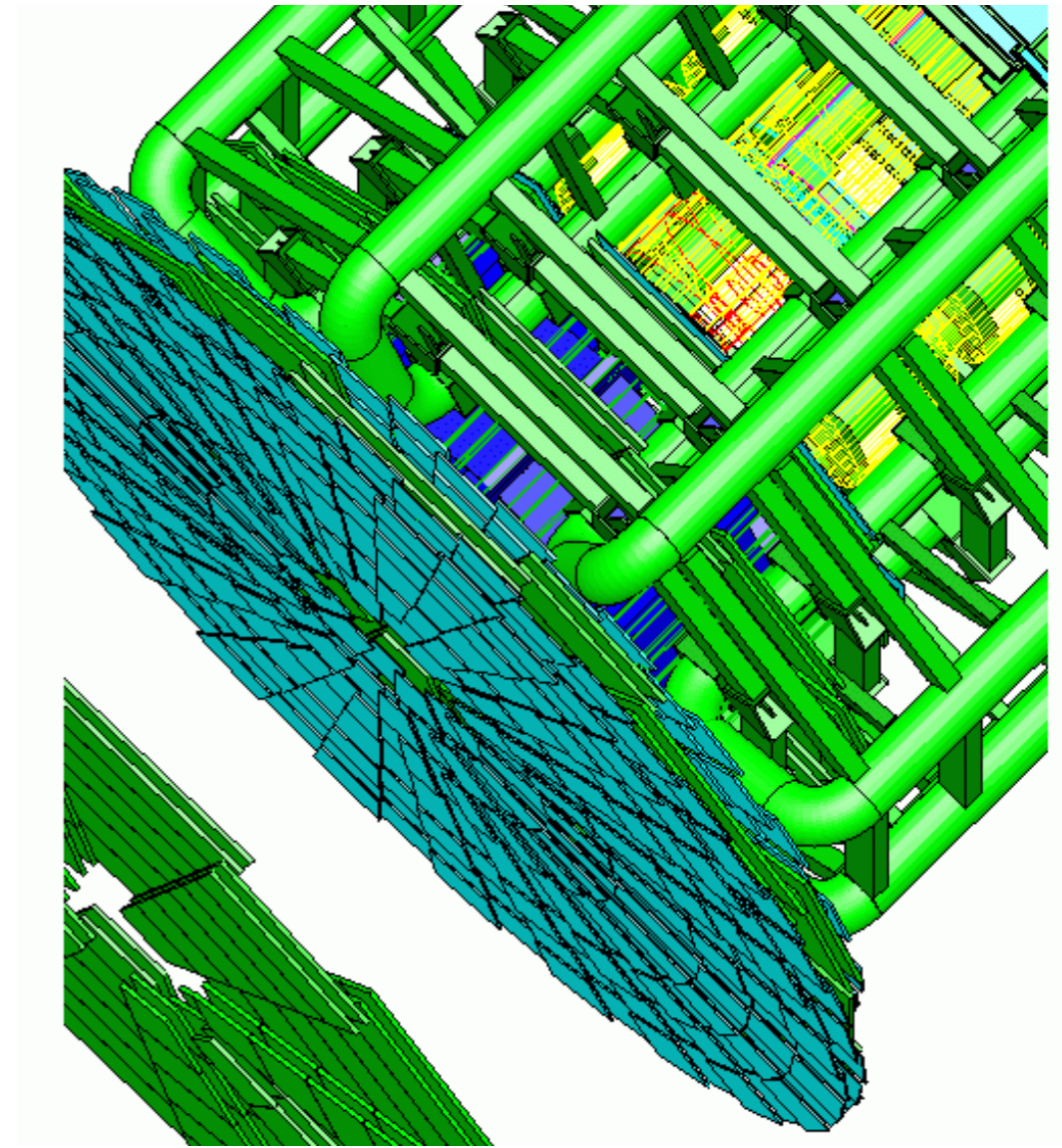


DAWNFILE Visualization Driver

- ◆ Creates a .prim file
- ◆ Requires DAWN to render
- ◆ Rendered, photorealistic PostScript image
- ◆ No interactive features once at PostScript stage
- ◆ How to use:

```
/vis/open DAWNFILE  
/vis/viewer/flush
```

- ◆ Also check DAWNCUT for planar cuts on a DAWNFILE

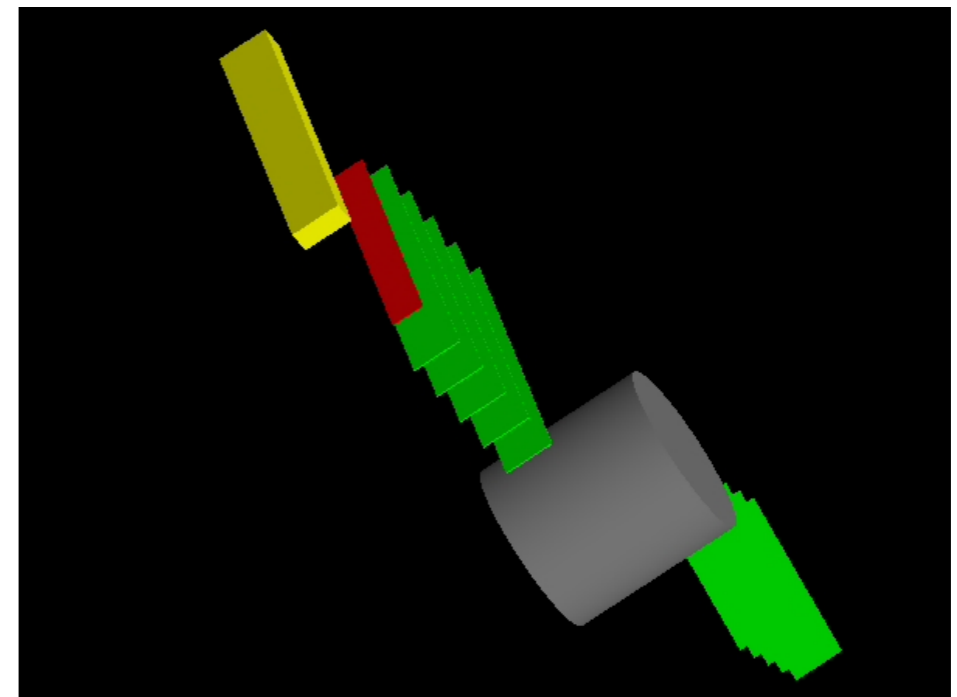
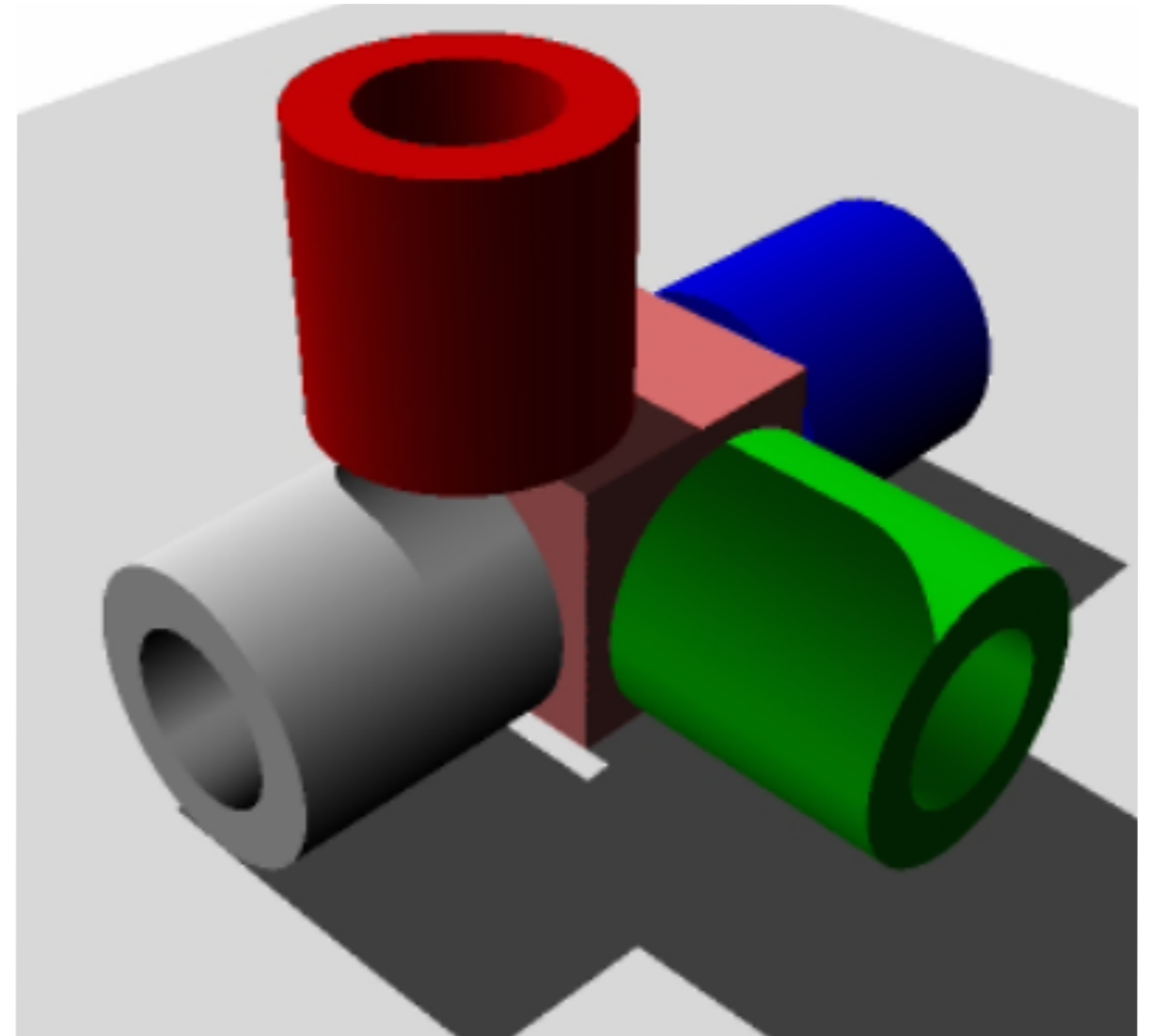


RayTracer Visualization Driver

- ◆ 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
- ◆ How to use:

```
/vis/open RayTracer  
  
/vis/viewer/flush  
  
/vis/viewer/set/viewpointThetaPhi 20 30  
  
/vis/viewer/flush
```

- ◆ Check/search guidance for additional help

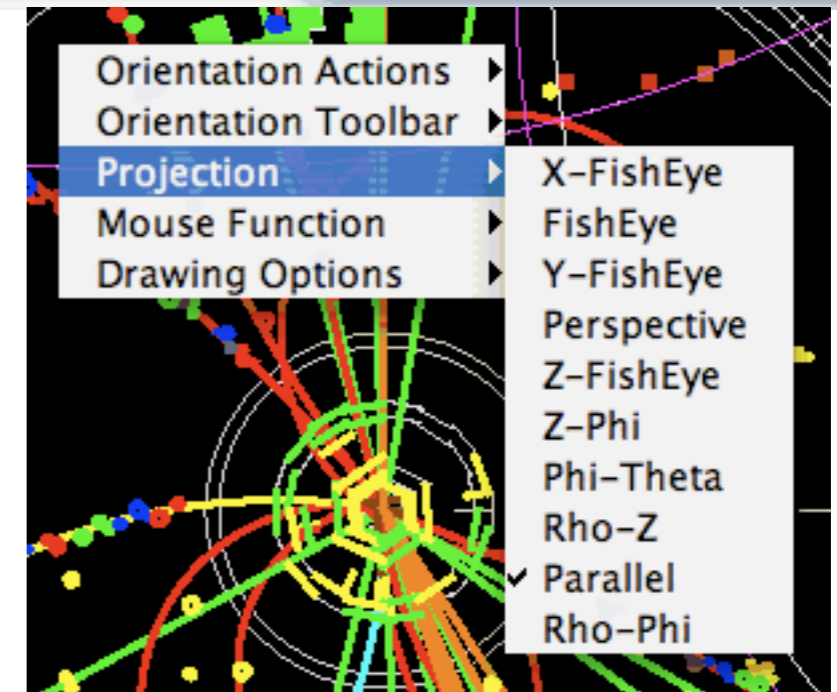
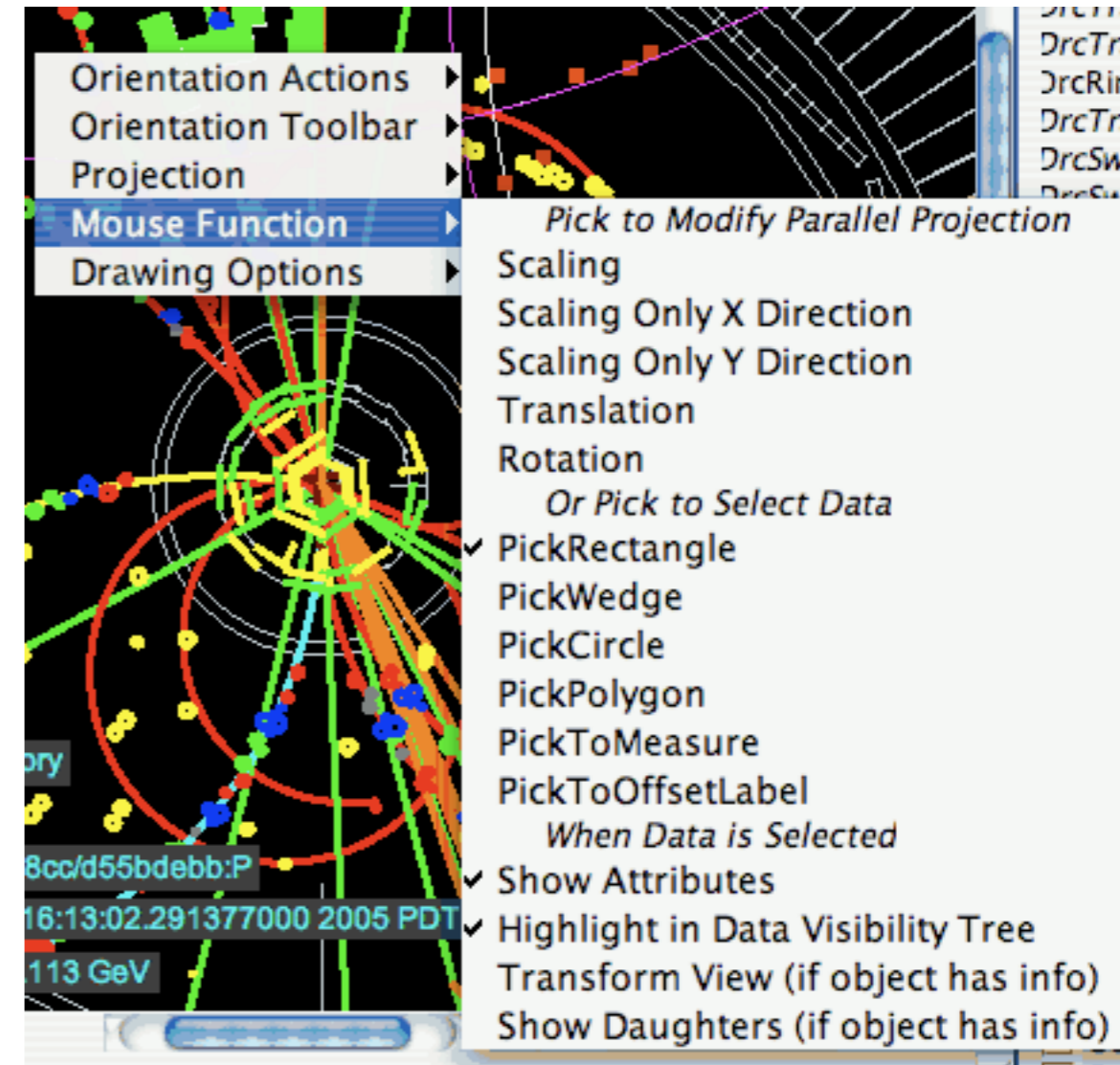


HepRepFile Visualization Driver

- ◆ Create a `.heprep` file to view in the
 - `HepRApp HepRep` Java app
- ◆ Wireframe or simple area fills (not photorealistic)
- ◆ Many interactive features
- ◆ Hierarchical view of the geometry
- ◆ How to use:

```
/vis/open HepRepFile  
/vis/viewer/flush
```

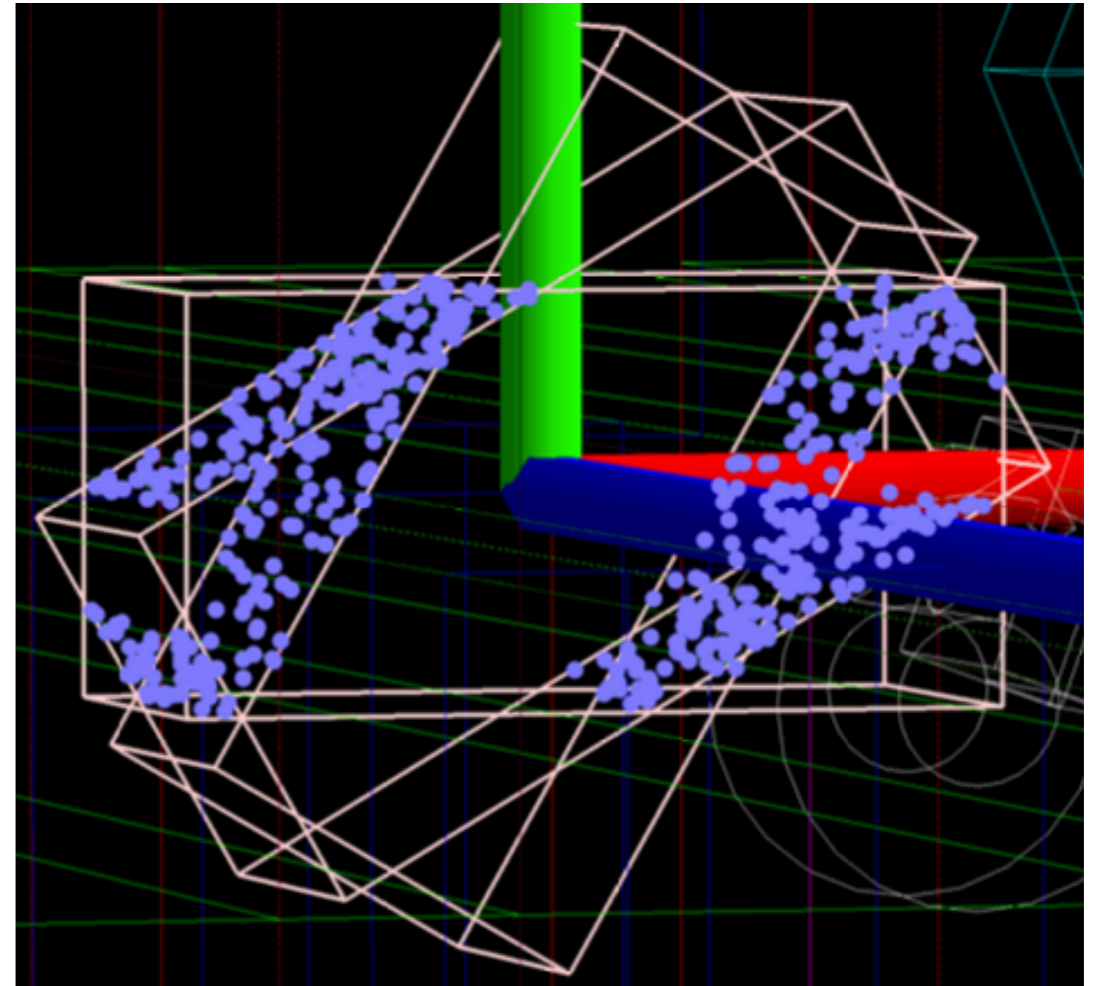
! **Warning: Issues with recent java versions being worked out**



Goodies: Visualization of Overlaps

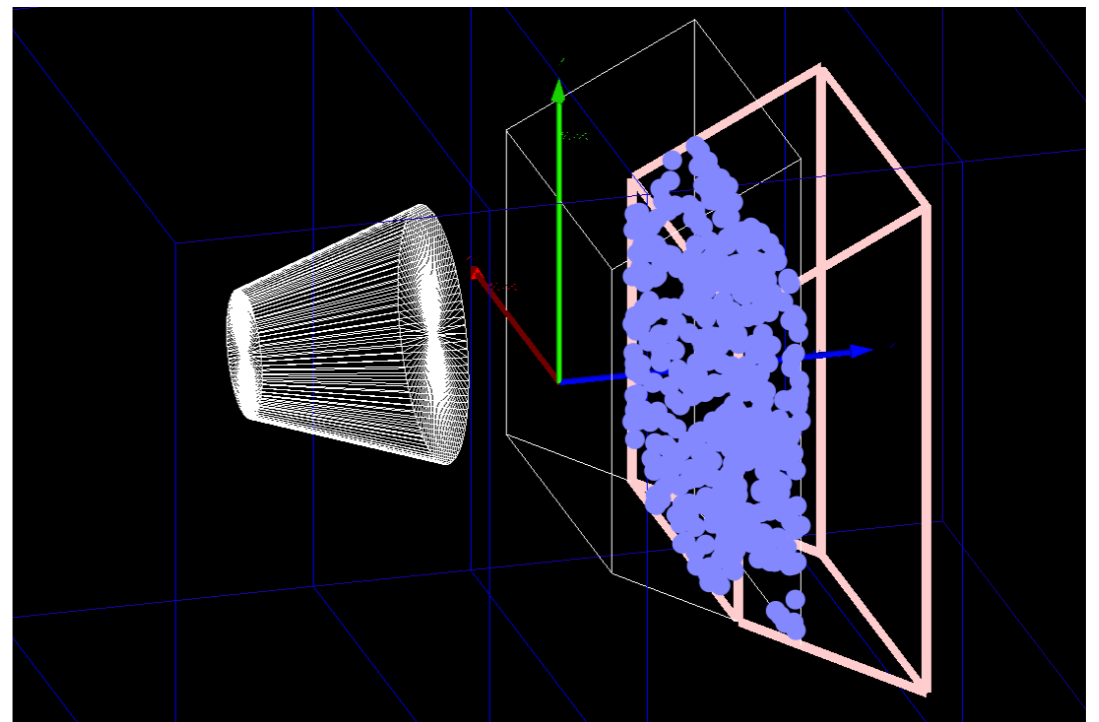
```
/vis/drawLogicalVolume <volumeName>
```

- ◆ By default will print and draw overlaps
- ◆ To my knowledge this will check only for direct daughters overlaps.



- ◆ Example B1, with added Shape translated

```
/vis/drawLogicalVolume Envelope
```



Goodies: Visualization of Fields

```
/vis/scene/add/magneticField
```

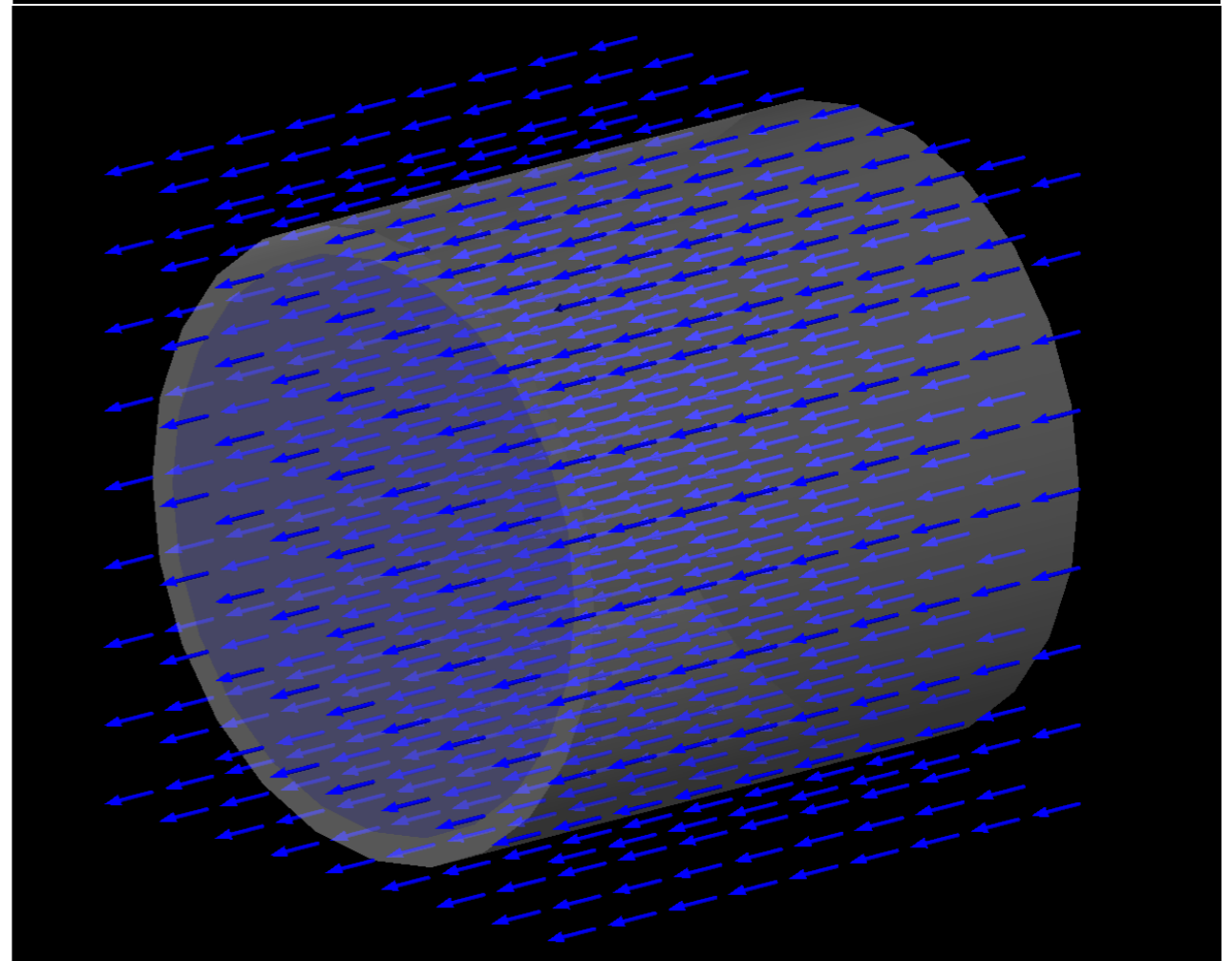
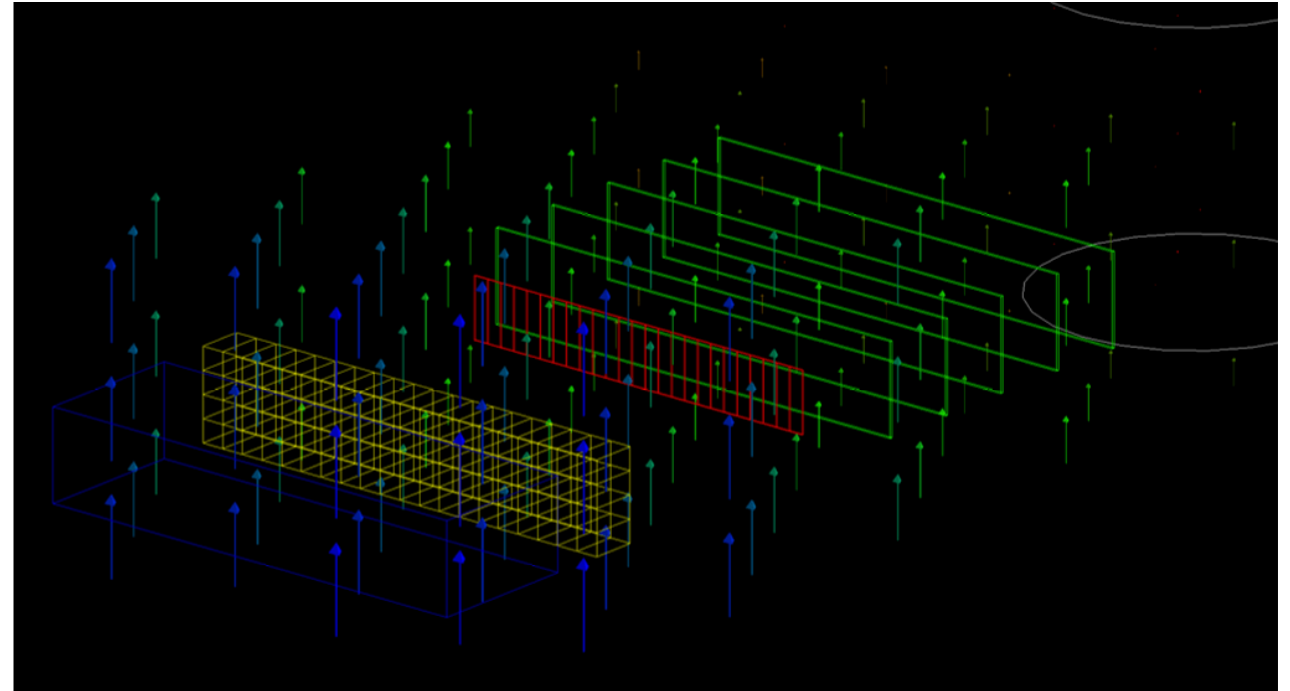
```
/vis/set/extentForField  
/vis/set/volumeForField
```

With these commands you can limit the extent over which the field is drawn.

This would help, for example, if drawing over the whole scene produced so many arrows or lines that it clutters the scene.

Optional parameter specify lines 'density'

See guidance for further explanation.

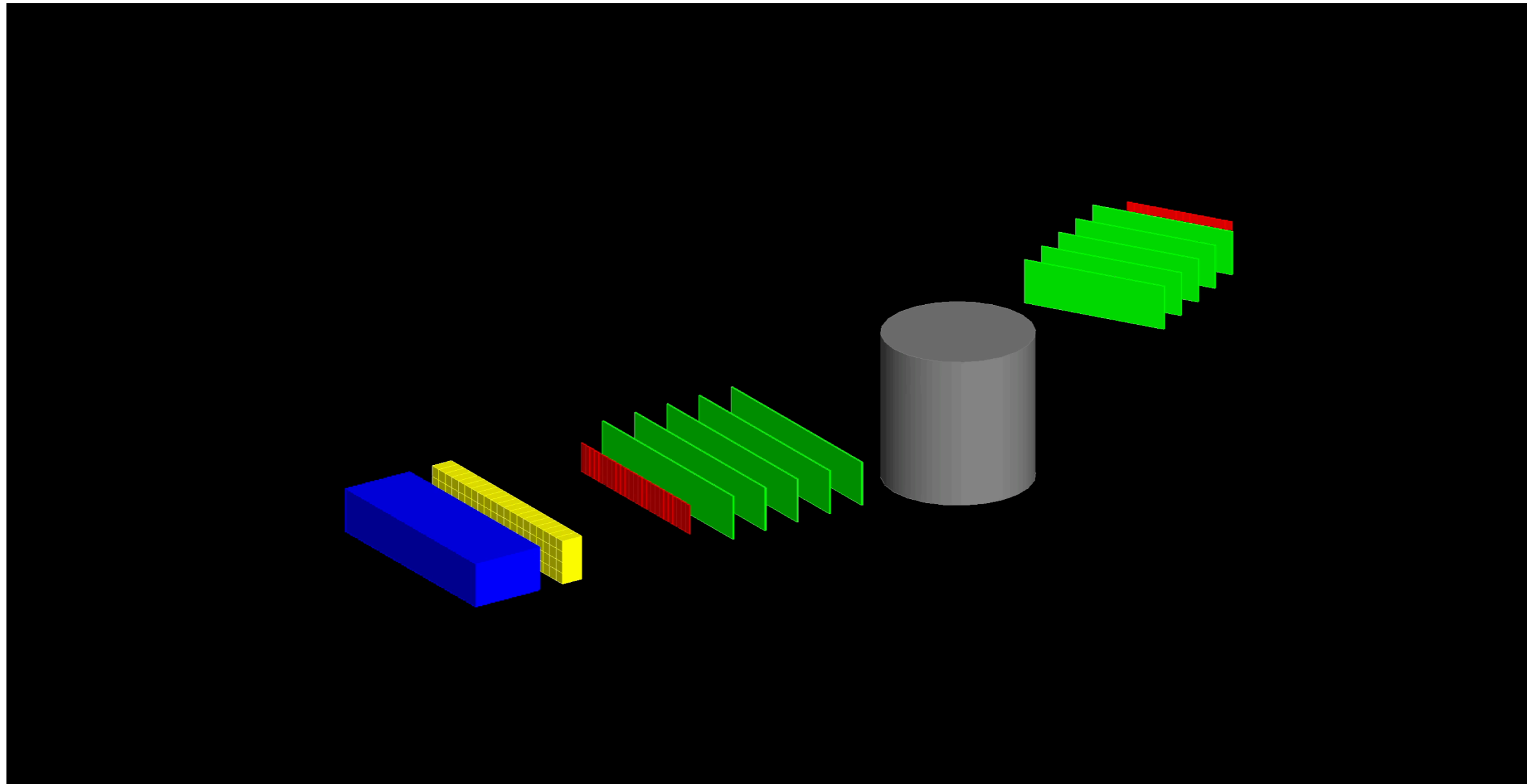


Goodies: Centering (and twinkle)

```
/vis/viewer/centreOn
```

```
/vis/viewer/centreAndZoomInOn
```

⚠ **NOTE:** 'twinkling' is switched off for complex geometries. This because each frame with different rightness of the volume requires a rebuild



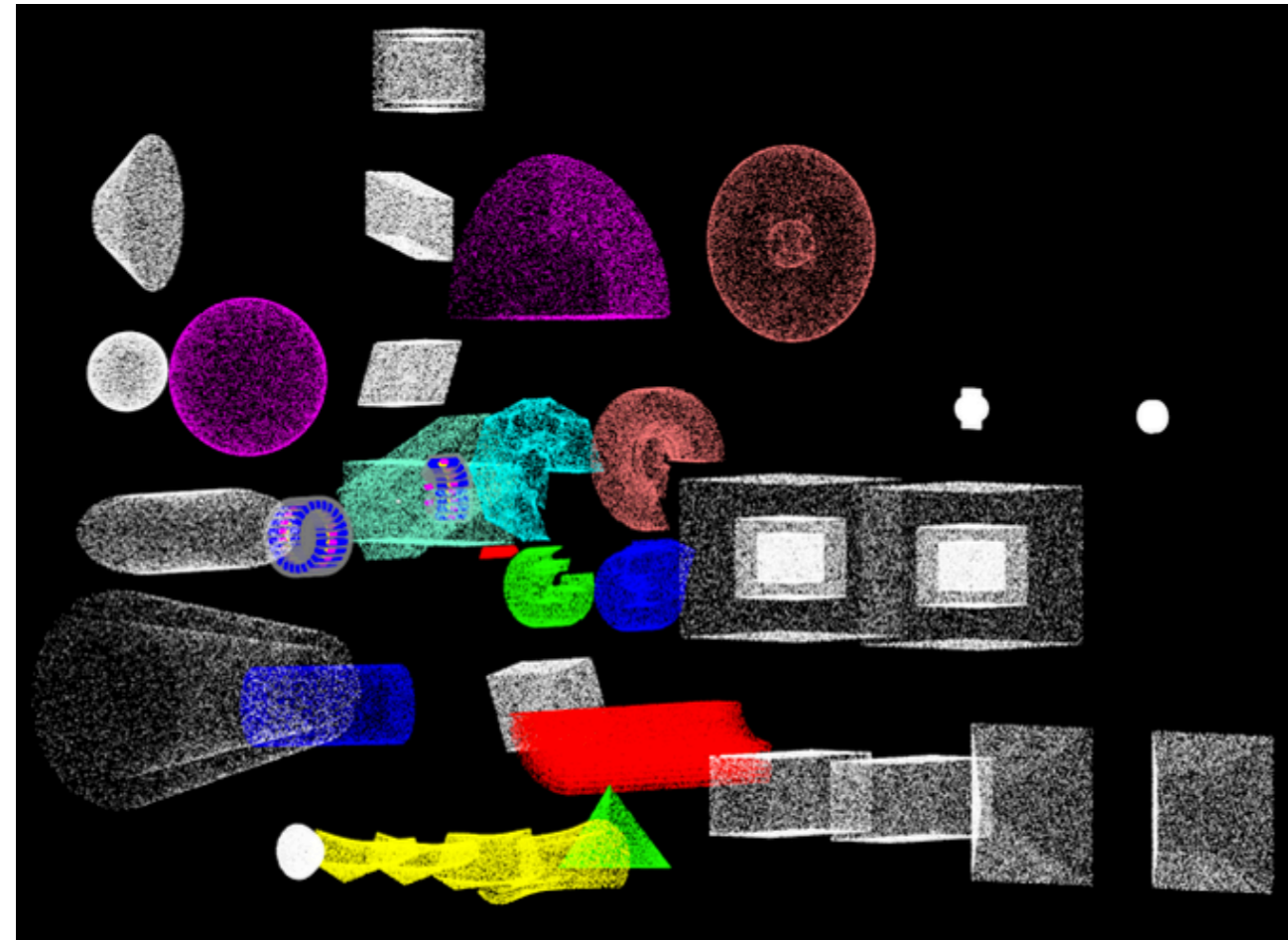
```
/vis/viewer/centreAndZoomInOn chamber2Physical 1
```

Goodies: Cloud Drawing Style

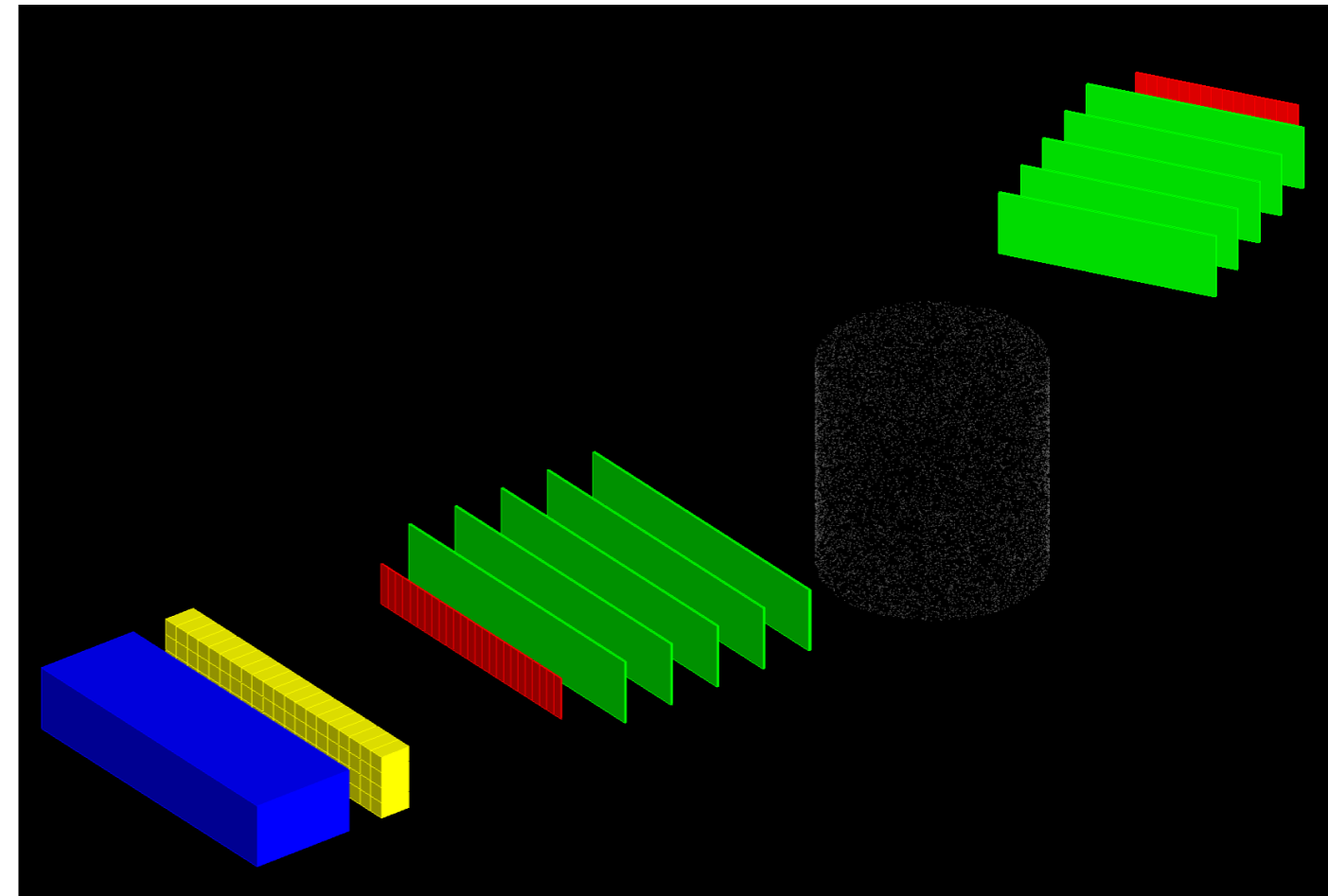
```
/vis/viewer/set/style cloud
```

```
/vis/geometry/set/forceCloud
```

- ◆ The solid is represented by a polymarker of dots.
- ◆ The default number of points is 1000.
 - This can be changed with `/vis/viewer/set/numberOfCloudPoints`



```
/vis/viewer/set/style cloud
```




```
/vis/geometry/set/forceCloud
```

Summary

Geant4 provides several *vis* drivers that can display data and objects.

- ◆ OpenGL interface quite advanced
- ◆ If Qt is installed (highly recommended) then Qt + OpenGL is the Geant4 default
- ◆ Old drivers may need updates from developers (gMocren, HepRepFile)
- ◆ Promising future technologies: ToolsSG, VTK

 **NOTE:** Much more information in the [Geant4 Visualization Documentatio](#)

