2022 - Updates in Vis/UI

Geant4 workshop - Rennes 2022

Laurent Garnier - Vis group

General updates

- Qt6 is on the way, hope it will be ready for next release
- Todo list :
  - Move Qt scene tree to be able to use it with all drivers
  - VTK driver prototype to complete

Open Inventor Qt Viewer

Frederick Jones, TRIUMF

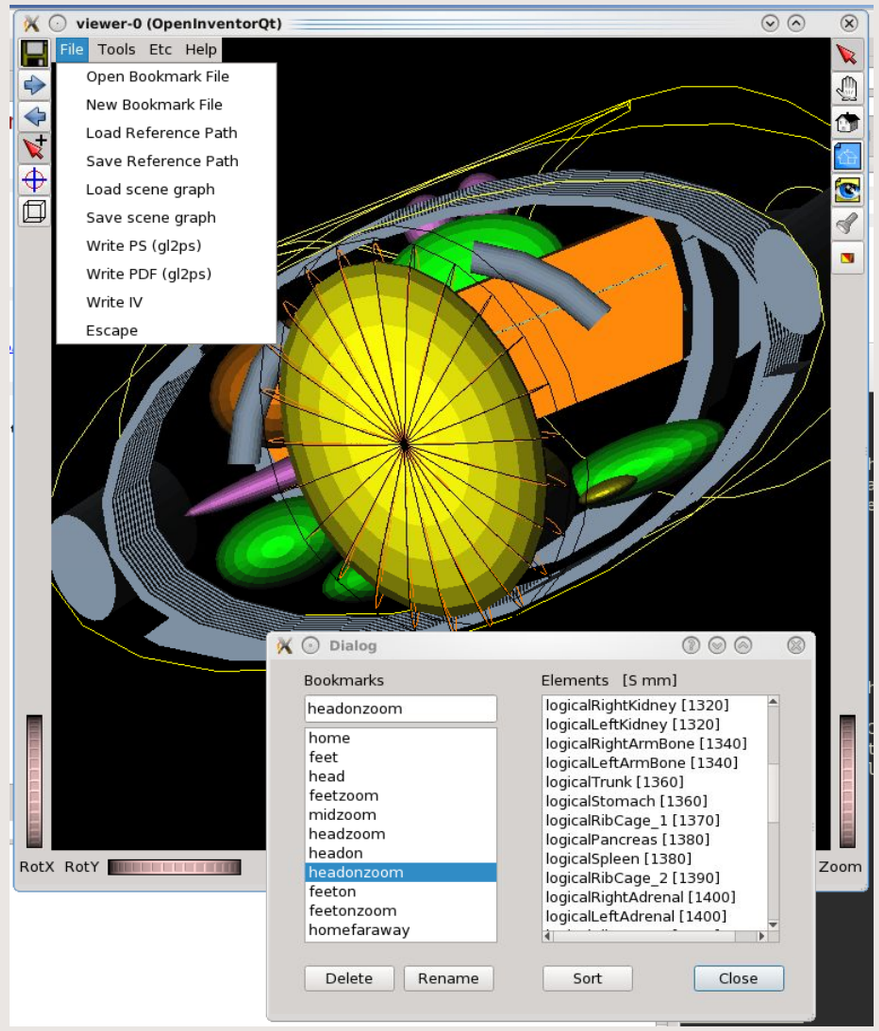
The Qt-based viewer (OIQt) was completed and released in Geant4 11.0.

How to use:

- Install Qt5 (most people will already have this to support the Qt UI).
- Install Coin3d libraries Coin 4.0 and SoQt 1.6 (or newer versions) using your package manager OR by downloading the sources for Coin and SoQt from https://coin3d.github.io/ and following the build instructions for Linux or MacOS.
- Build with cmake –DGEANT4_USE_OPENINVENTOR_QT ...
- To open the viewer: /vis/open OI either in terminal or UIQt session
- MacOS note: this vis driver is intended to use the native OpenGL on MacOS. There is no requirement for XQuartz or any X11 or Mesa libraries.

Some unique capabilities

- Full 3d manipulation and interactivity via mouse/trackpad, viewer buttons, menus, and a bookmark/navigation panel.
- Bookmarking: similarly to a web browser, any 3d view can be “bookmarked” and revisited later with a single mouse-click on an editable list which is persistent between sessions. Useful for geometry development/debugging and interactive slide shows.
- Navigation along a reference path defined by a trajectory or by a 3d polyline, with precise camera movements and rotations.
- Others, including 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.



Is anybody using it? Waiting for user comments and questions!

Visualization on Windows Subsystem for Linux (WSL2)

WSL comes with Windows 10 and 11 and enables installation and operation of Linux (various distributions) without the need for a virtual machine or dual boot system.

Installation is easy: wsl --install –d Ubuntu-20.04

[set wsl version to 2 and install kernel update]

bash [opens a terminal window]

My Experience (using with Windows 10):

- Building and installing Geant4 is identical to doing it on a native Linux system.
- Ubuntu is quite stripped down and many packages had to be installed to support Geant4.
- Existing documents for “Geant4 on WSL” were helpful but occasionally wrong.

What about visualization? Short answer: everything works!

- WSL2 strategy: The application connects to an X server with the GLX extensions. The server is a Windows application and hence uses the Windows 3d graphics drivers.
- Best X server: GWSL. Didn’t work for me: XMing (never); VcXsrv (eventually broke).
- Built and tested all the interactive drivers: OGLQt, Open Inventor, Vtk, and ToolsSG, without any problems, even running all of them at once. Very fast OpenGL rendering.
- WSLg (Windows 11 only) offers seamless graphics without the need for an X server.

Is this important for Geant4?

- It could be an alternative way to support Geant4 on Windows. However, it would have to be operable within the existing workflows and supporting applications used by Windows Geant4 users.
- It allows Geant4 visualization access to high-performance 3d graphics hardware and drivers that are commonly used on gaming and professional Windows systems.

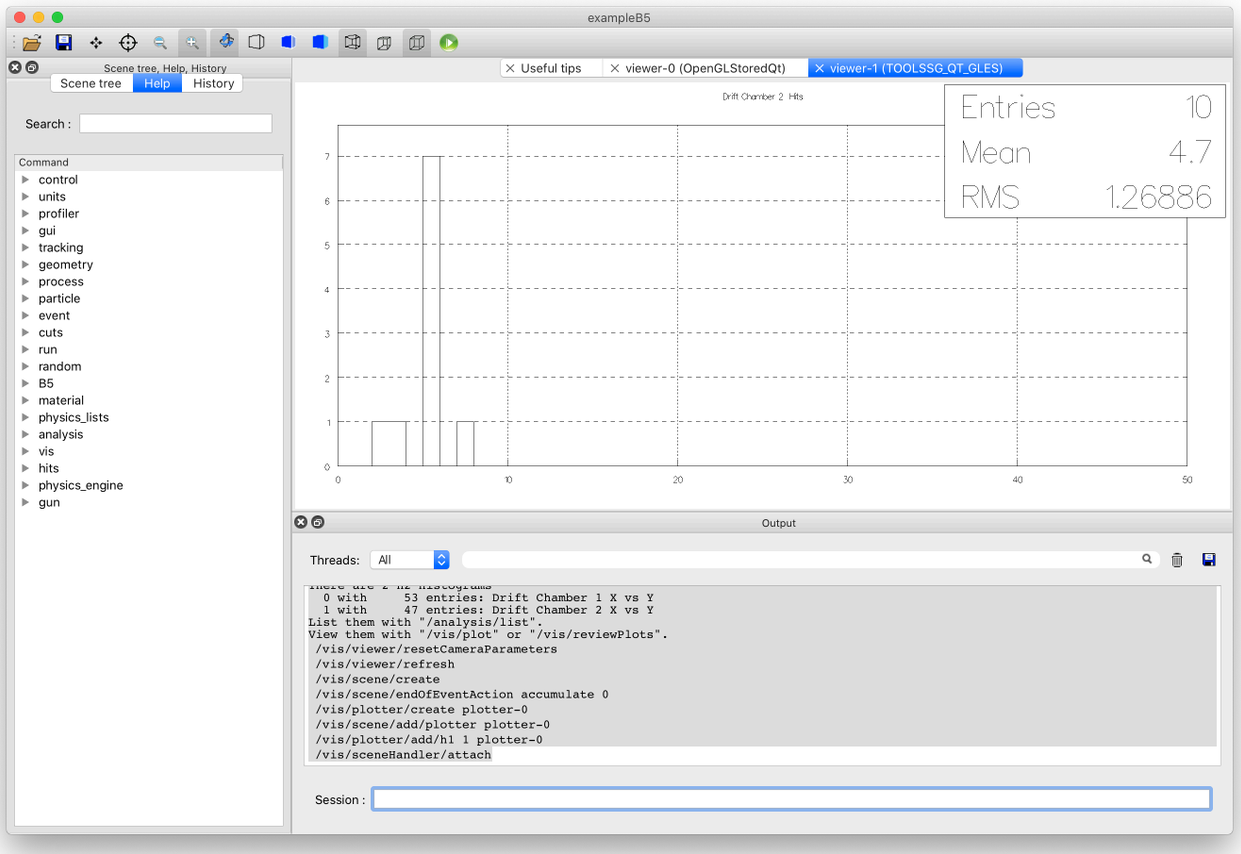
Plotting

- Guy Barrand, Ivana Hrivnacova, John Allison
- If your app uses `G4Analysis`, at end of job you will see:

```
10 events have been kept for refreshing and/
or reviewing.
  "/vis/reviewKeptEvents" to review one by one.
  To see accumulated, "/vis/enable", then "/vis/
viewer/flush" or "/vis/viewer/rebuild".
There are 2 h1 histograms
  0 with     10 entries: Drift Chamber 1 # Hits
  1 with     10 entries: Drift Chamber 2 # Hits
There are 2 h2 histograms
  0 with     53 entries: Drift Chamber 1 X vs Y
  1 with     47 entries: Drift Chamber 2 X vs Y
List them with "/analysis/list".
View them with "/vis/plot" or "/vis/reviewPlots".
```

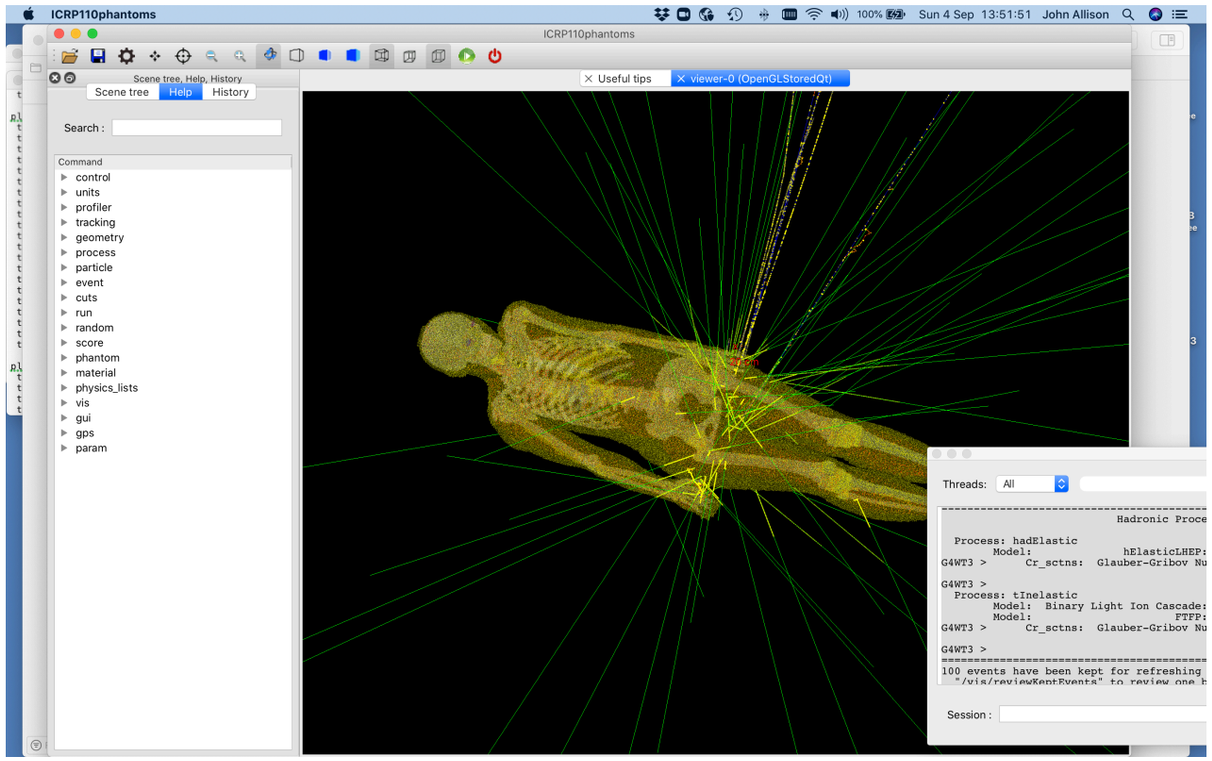
Plotting (2)

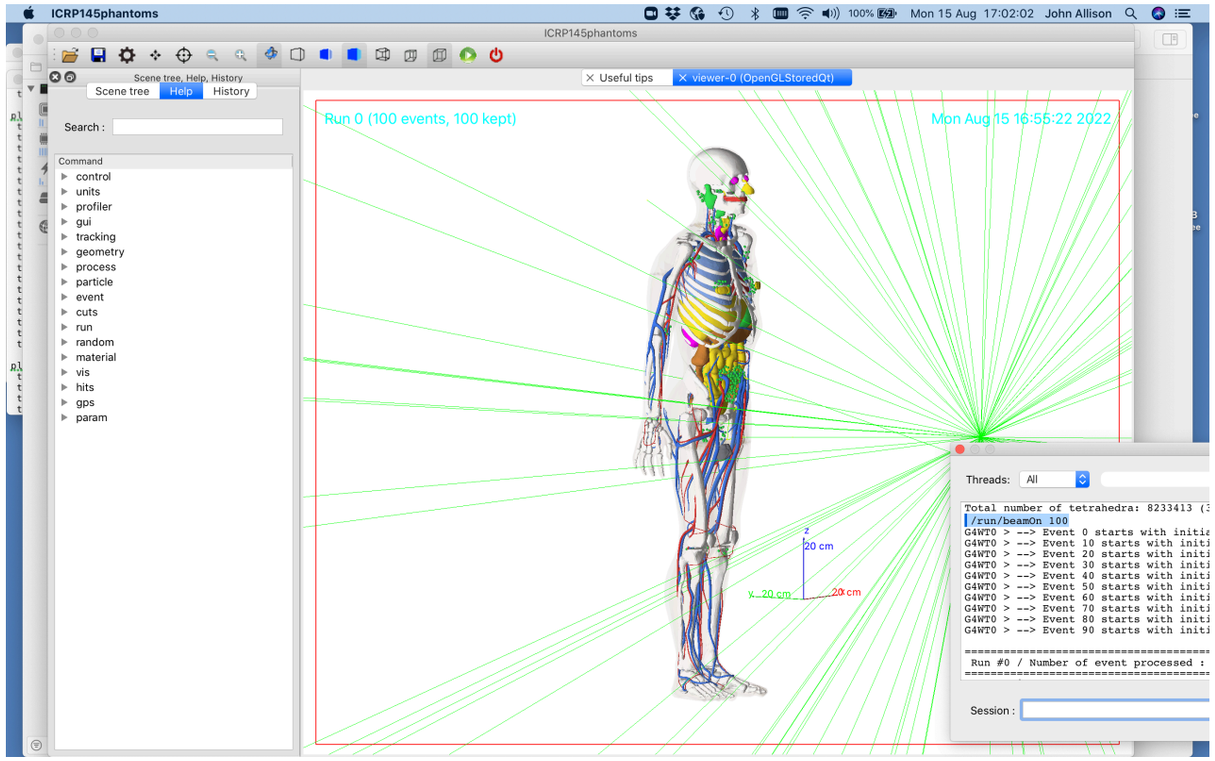
- So try
  - `/vis/plot h1 1`
    ```
    WARNING: Current viewer not able to draw plots.
      Try "/vis/open TSG", then "/vis/plot h1 1" again.
    ```
- So try
  - `/vis/open TSG`
    `/vis/plot h1 1`



Special Mesh Rendering

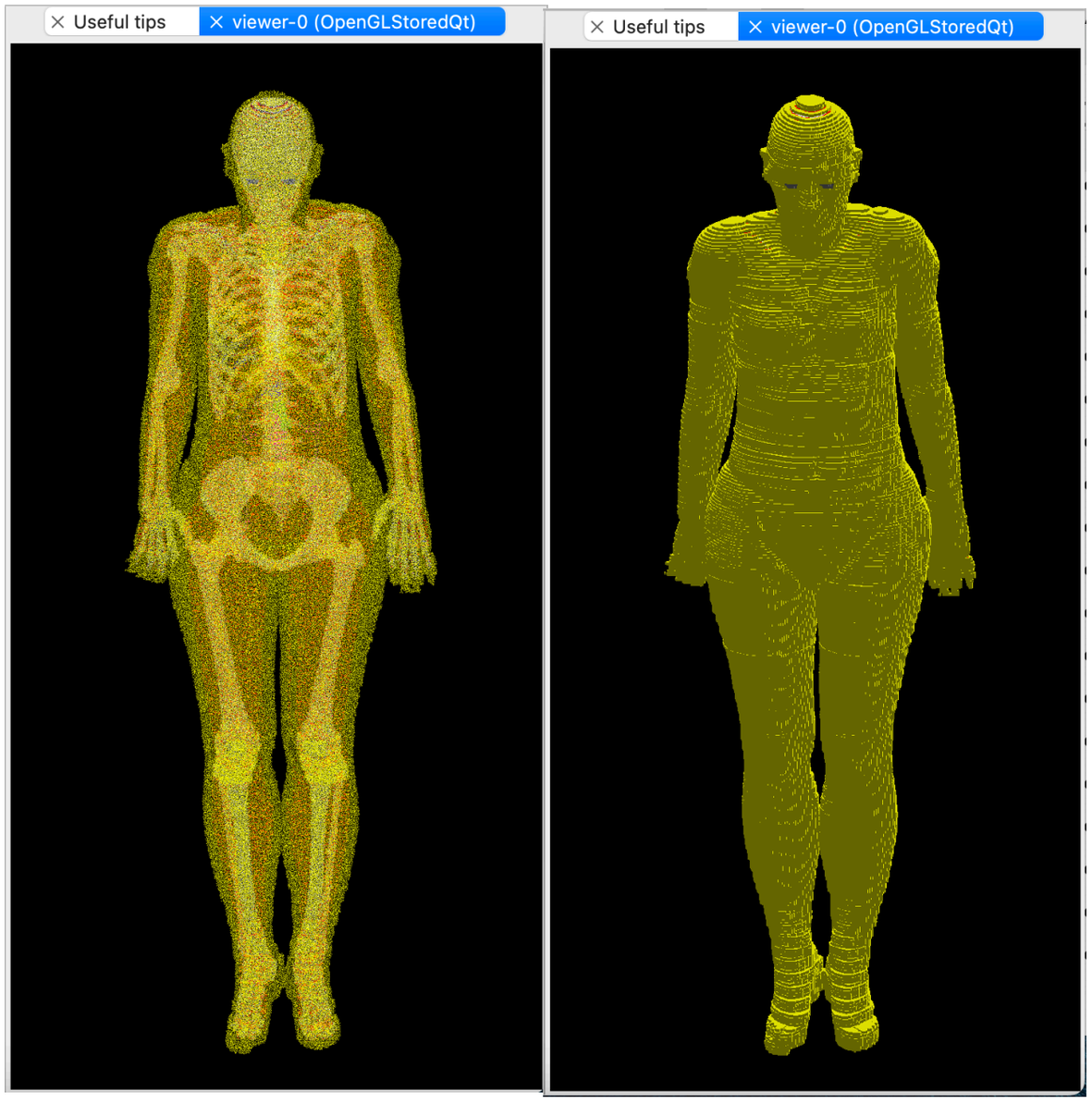
- Evgueni Tcherniaev, John Allison: to be presented in the vis parallel session
- A G4VParameterised can be treated as a G4Mesh and rendered as dots of surfaces
- Used in examples/advanced/ICRP110_HumanPhantoms: e.g., 3,885,291 dots
- Used in examples/advanced/ICRP145_HumanPhantoms: e.g., 8,233,413 `G4Tets` represent 187 organs, reduced to 4,807,770 external faces





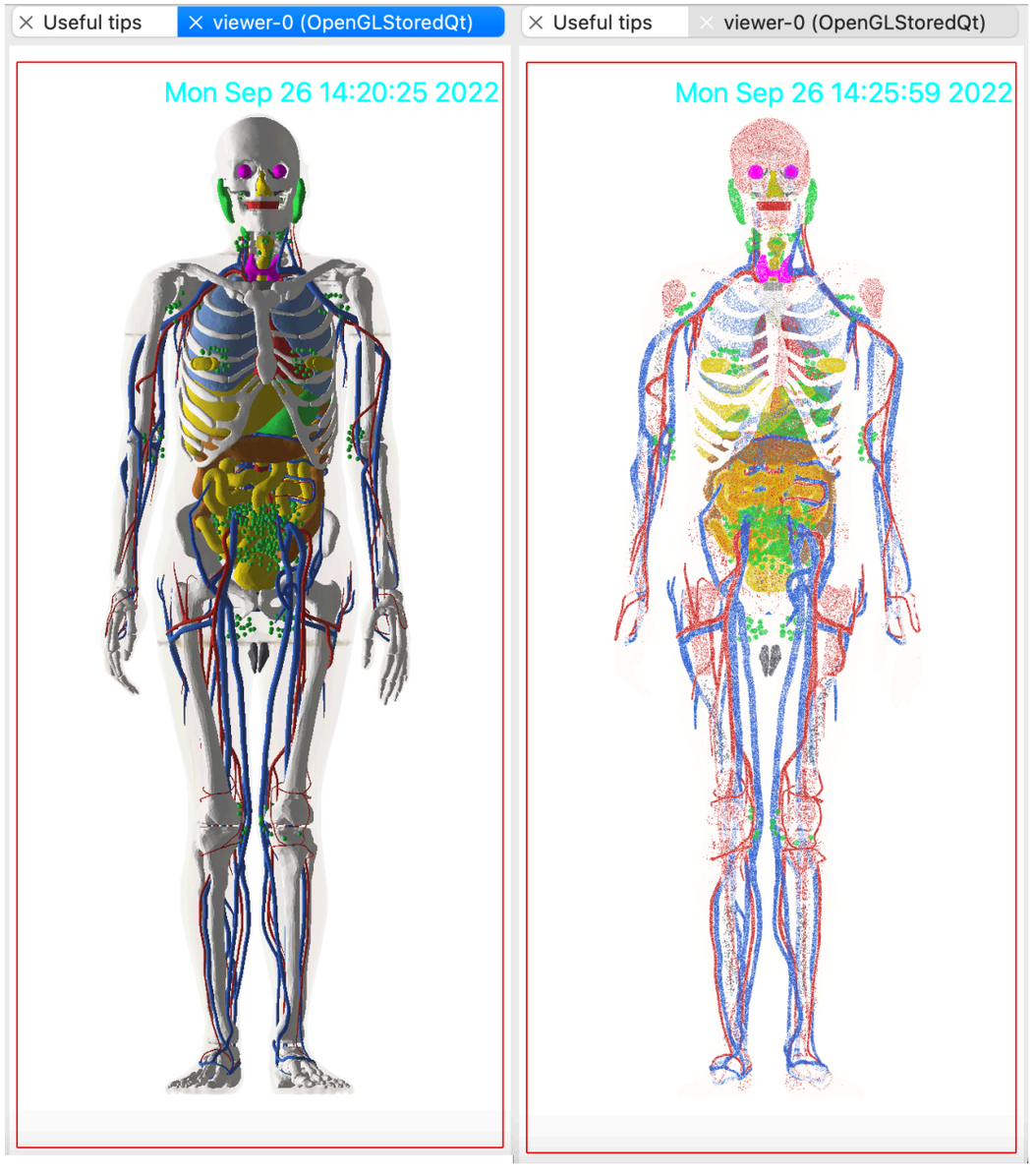
Visualization of rectangular Mesh in ICRP110_HumanPhantoms

- The example is based on the publication: ICRP, 2009. Adult Reference Computational Phantoms. ICRP Publication 110. Ann. ICRP 39 (2).
- Contains a G4VNestedParameterisation of 299x137x348 = 14,255,124 boxes
- 3,885,291 (23,311,746 faces) of which are “visible” representing 52 organs
- one dot per box are sorted into 52 G4Polymarker objects of different materials (and colour)
- 3,885,291 dots are rendered at about 5 fps on MacBook Pro (Retina, Mid 2012), 2.7 GHz Quad-Core Intel Core i7, 16 GB
- Original 23,311,746 faces reduced to to 3294818 facets (14%), which are rendered with approximately the same speed



Visualization of tetrahedral Mesh in ICRP145_HumanPhantoms

- The new advanced example is based on the publication:
  ICRP, 2020. Adult mesh-type reference computational phantoms. ICRP Publication 145. Ann. ICRP 49(3)
  C.H. Kim, Y.S. Yeom, N. Petoussi-Henss, M. Zankl, W.E. Bolch, C. Lee, C. Choi, T.T. Nguyen, K. Eckerman, H.S. Kim, M.C. Han, R. Qiu, B.S. Chung, H. Han, B. Shin
- Contains a one-level `G4PVParameterised` of 8,233,413 `G4Tets` (32,933,652 faces) representing 187 organs
- By eliminating internal shared faces, tetrahedra are converted to 187 G4Polyhedron objects by material (and colour) with only 4,807,770 faces (14% of original number of faces), which are rendered at about 2 fps on MacBook Pro (Retina, Mid 2012), 2.7 GHz Quad-Core Intel Core i7, 16 GB
- 8,233,413 dots are rendered at about the same speed



“Twinkling”

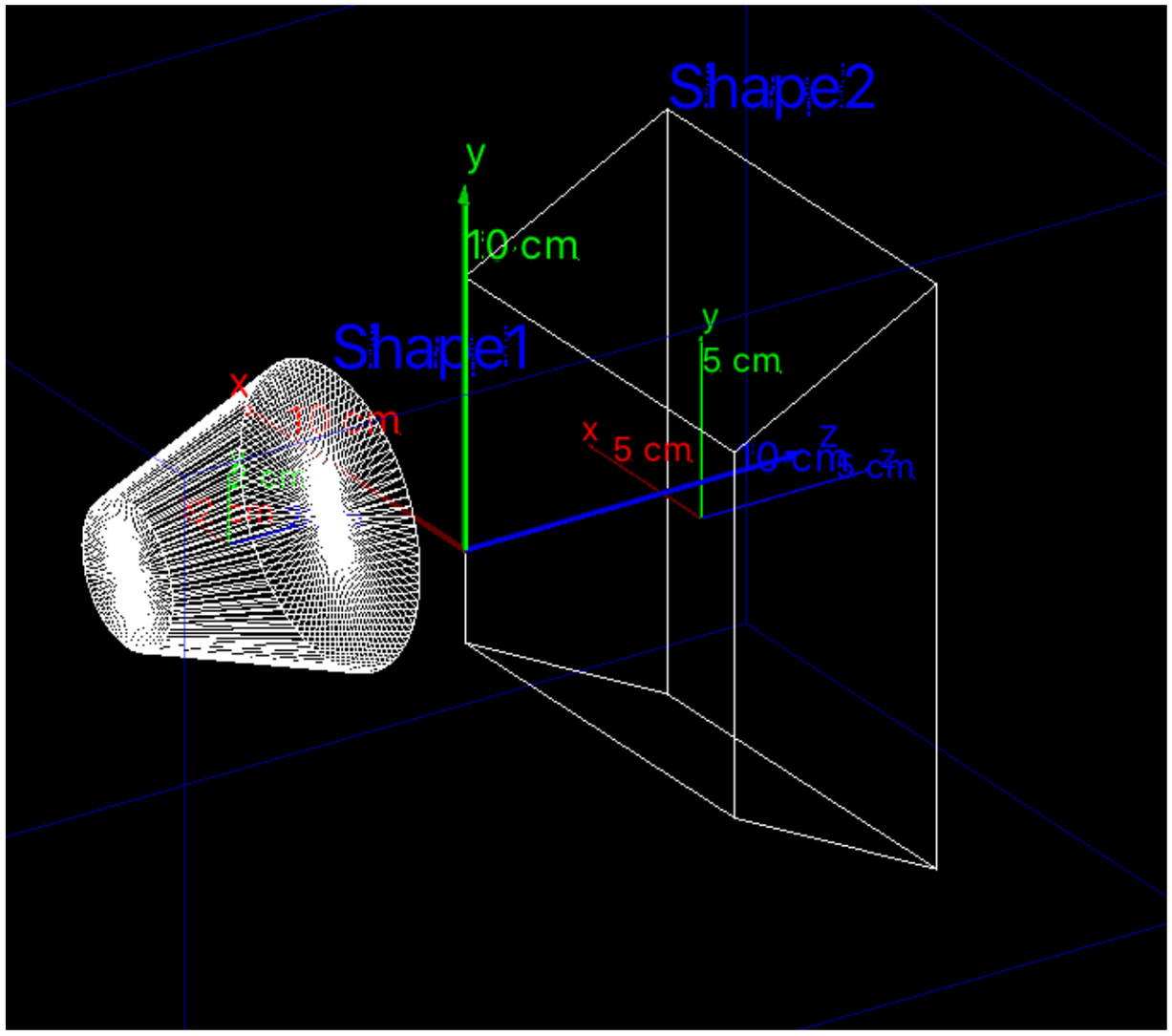
- When you centre or zoom in on a volume:

  `/vis/viewer/centreOn` and `centreAndZoomInOn`
  `/vis/viewer/centreOn` and `centreAndZoomInOn`

  the volume “twinkles” (varies on brightness) for a short time to attract your attention
- Restriction: the feature is switched off for complex geometries, i.e., those for which a rebuild of the graphics database takes more than $0.1\ s$. This because each frame with different brightness of the volume requires a rebuild.

Local axes

- /vis/scene/add/localAxes
- /vis/touchable/localAxes



New/retired vis drivers

- New in Geant4 11.0 and further developed for 11.1
  - **Qt3D** (John Allison): limited functionality but nice
  - ToolsSG (TSG)(Guy Barrand): (almost) fully working, nice
    - Also supports plotting
  - **Open Inventor Qt** (OIQt)(Fred Jones): Also very nice, requires users
    - Includes “bookmarking” and “navigation”
  - **Vtk** (Stewart Boogert, Laurie Nevay): prototype version only
- To be 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)
- **gl2ps**: the vis version has been removed; we use the version in g4tools