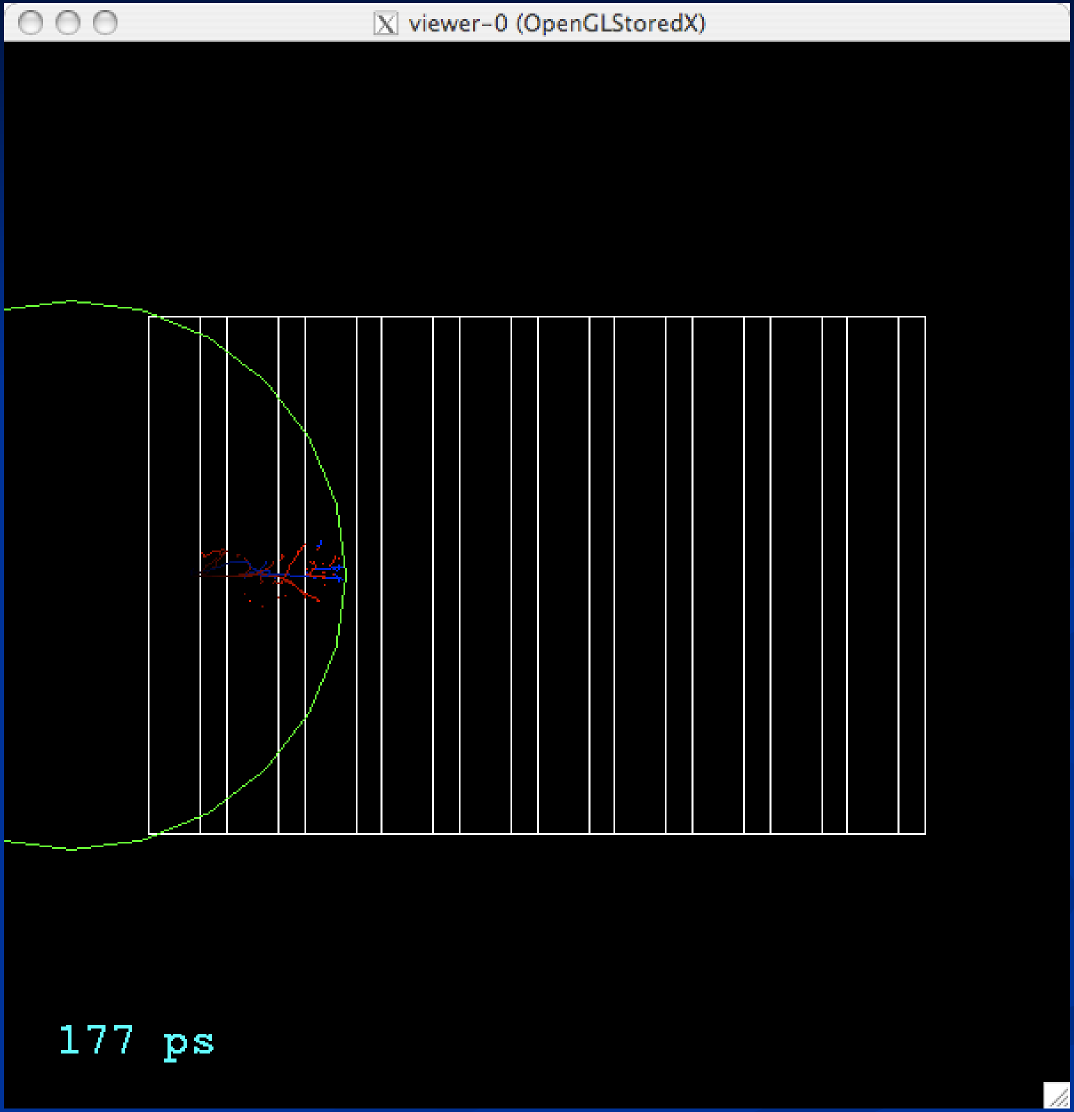
Geant4 Visualization Special Topic: How to Make a Movie



Joseph Perl

SLAC

Acknowledgement

- The techniques presented here were pioneered by John Allison.
- This talk draws heavily on the documentation and examples that John presented at last month’s Geant4 Collaboration Workshop in Lisbon.

Part 1: The Old Technique
Stitching Still Pictures Together to Make a Movie

Stitching Still Pictures Together to Make a Movie

- Making movies with just changes to camera position has been possible for years using macros
  - Produce a large number of still images
  - Stitch them together into a movie file (mpeg)
- Conversion to mpeg somewhat complicated
  - Documented since release 8.2
  - New section of the Application Developers Guide, section 8.10.
- I'll show you here how to do it with OpenGL, but you can also do it using other drivers that can output pictures, such as DAWNFILE and RayTracerX

Movies: Stitched Together from Multiple Stills

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

It’s All About Macros

- Main Macro
  - /control/verbose 2
  - /vis/open OGLSX
  - /vis/drawVolume
  - /vis/viewer/reset
  - /vis/viewer/set/style surface
  - /vis/viewer/set/projection perspective 50 deg
  - /control/alias phi 30
  - /control/loop movie.loop theta 0 360 1
- Inner Macro, movie.loop
  - /vis/viewer/set/viewpointThetaPhi {theta} {phi}
  - /vis/viewer/zoom 1.005
  - /vis/viewer/refresh ←

Omit /vis/viewer/refresh on auto-refresh devices such as OGLIX and OGLSX

- At each step of the macro
  - the volume rotates a little
  - zooms in a little
  - and then refreshes the OpenGL window
- Demo live from Geant4

To Create a Movie File, Start by Saving the Individual Frames

- Main Macro stays the same:
  - /control/verbose 2
  - /vis/open OGLSX
  - /vis/drawVolume
  - /vis/viewer/reset
  - /vis/viewer/set/style surface
  - /vis/viewer/set/projection perspective 50 deg
  - /control/alias phi 30
  - /control/loop movie.loop theta 0 360 1
- Inner Macro, movie.loop, just has one extra command to output to file
  - /vis/viewer/set/viewpointThetaPhi {theta} {phi}
  - /vis/viewer/zoom 1.005
  - /vis/ogl/printEPS
  - /vis/viewer/refresh ← Omit /vis/viewer/refresh on auto-refresh devices such as OGLIX and OGLSX
- At each step of the macro
  - the volume rotates a little
  - zooms in a little
  - and then an eps file is produced

From EPS to MPEG (slide 1 of 2)

- To turn the many eps files into a single mpeg, you first need to generate an mpeg parameters file. To do this, you will use the script:
  - make_mpeg2encode_parfile.sh
  - which can be found in
    - geant4/UserDoc/UserGuildes/ForApplicationDeveloper/html/Visualization/

- Run the script as:
  - make_mpeg2encode_parfile.sh G4OpenGL_*eps

- Edit the resulting mpeg2encode.par file to specify file type and size
  - Before "/* horizontal_size */" put a size, such as:
    - 600 /* horizontal_size */>
  - Before "<   /* vertical_size */" put a size, such as:
    - 600 /* vertical_size */>
  - Before /* aspect_ratio_information" put a ration, such as:
    - 1 /* aspect_ratio_information 1=square pel, 2=4:3, 3=16:9, 4=2.11:1 */

From EPS to MPEG (slide 2 of 2)

- You'll need an eps conversion facility called "convert" that you can get from:
  - imagemagick.org
- Batch convert the eps files to ppm files using a shell command such as:
  - for i in G4OpenGL*eps; do j=`basename $i .eps`; command="convert $i $j.ppm"; echo $command; $command; done
- Finally, run the mpeg encoder using the parameters file and the ppm files:
  - mpeg2encode mpeg2encode.par G4OpenGL.mpg
  - mpeg2encode is free software from:
    - http://www.mpeg.org/MSSG/
- The resulting mpeg file can be run in any mpeg player.
  - For example, on a Mac, opening G4OpelGL.mpg starts a QuickTime player.
- The above instructions used OpenGL, but you can also make a movie using other drivers that can output pictures, such as DAWNFILE and RayTracerX.
  - For details on making movies from any of those drivers, see section 8.10 of the Geant4 Application Developers Guide.

Part 2: the New Technique
Time Development of the Event
Live from OpenGL

New Technique: Time Development of the Event

- New features since release 8.2 allow you to do a new kind of Geant4 movie.
  You will be able to 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 Stored driver
    - /vis/open OGLSX
      or for windows
      /vis/open OGLSWin32
- You can run these animations Directly from Geant4. You do NOT need to output any files or stitch them together into an mpeg.

Movies: Live from Geant4 OpenGL

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 pi- on lead (in a lead-liquid-argon calorimeter, exampleN03 with QGSP physics)
  - A plethora of slow pions, protons and neutrons
  - Three fast pi- and one fast pi+ 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

You’ll Need Geant4 Release 8.2 or Newer

- Because you’ll need the latest Rich Trajectory code
  - so that it can instrument each trajectory step with a time value
- And the latest Enhanced Trajectory Drawing code
  - so that it can slice the trajectory into many different graphics primitives based on time slices
- And the latest Geant4 OpenGL Driver
  - so that it can show one time slice at a time
- And the latest Trajectory and Hit Filtering
  - so you can get all of those annoying low energy gammas out of the picture
- The techniques shown in this talk were first unveiled by John Allison at the 2006 Geant4 Collaboration Workshop in Lisbon
  - See John’s example movies at: http://www.hep.man.ac.uk/u/johna/pub/Geant4/Movies/

The New Vis Tutorial Examples

- There are two example macros in the visTutor part of example N03. This presentation is based on those examples.
  - /examples/novice/N03/visTutor/exN03Vis12
  - /examples/novice/N03/visTotur/exN03Vis13

- Before you try to make a move, just build and run exampleN03 normally and check that you have OpenGL working. You'll need the "stored" version of OpenGL for this work:
  - /vis/open OGLSX
    or on Windows
    /vis/open OGLSWin32

- Make sure something appears in the OpenGL window when you do a simple visualization command such as:
  - /vis/drawVolume

The New Trajectory Commands

- The examples use various forms of the following commands:

- Add trajectories using the new Rich Trajectory option
  - /vis/scene/add/trajectories rich

- Create a trajectory model to color by charge
  - /vis/modeling/trajectories/create/drawByCharge

- Specify the new time-slicing option in the trajectory model
  - /vis/modeling/trajectories/drawByCharge-0/default/ setTimeSliceInterval 0.1 ns
    - Chops trajectory into pieces, each assigned a time range
    - Remember that particles are often relativistic and travel 30 cm/ns

- Filter trajectories to remove gammas
  - /vis/filtering/trajectories/create/particleFilter
  - /vis/filtering/trajectories/particleFilter-0/add gamma
  - /vis/filtering/trajectories/particleFilter-0/invert true

The OGL Commands

- At present, the time-dependent animation only works in the Stored Mode OpenGL drivers
  - /vis/open OGLS*

  So the commands are specific to /vis/ogl

- Specify time window
  - /vis/ogl/set/startTime 2 ns 1 ns
  - Items are displayed if their time range* and viewer's time window overlap
  - Default time range is -infinite to infinity, so normal items such as detector geometry are always drawn

- Specify how fast slices should fade, gives a "vapor trail" effect
  - /vis/ogl/set/fade 1

- Specify whether the light wavefront should be shown
  - /vis/ogl/set/displayLightFront true -90 0 0 mm

- Specify whether the current time should be displayed
  - /vis/ogl/set/displayHeadTime true

The Time-Slice Loop

- Simplest form, from exN03Vis12.mac
  - /control/alias timeRange 0.1
  - /control/loop loop.mac startTime -{timeRange} 0.7 0.005
- where loop.mac specifies start time and range for which slices to show
  - /vis/ogl/set/startTime {startTime} ns {timeRange} ns

- Slightly more complicated form, from exN03Vis13.mac, adds panning so that camera follows the light front and specifies time slices by end time rather than start time
  - /control/alias timeRange 0.1
  - /control/alias dx 0.3
  - /control/loop loop.mac endTime 0 0.7 0.001
- where loop.mac specifies end time for which slices to show
  - /vis/viewer/pan {dx} 0 mm
  - /vis/ogl/set/endTime {endTime} ns {timeRange} ns

Running the Vis Tutorial Examples

- examples/novice/N03/visTutor/exN03Vis12.mac
  does three animations in a row:
  - Draw by charge with trajectory points
  - Draw by particle ID (remove g’s)
  - p-m-e decay

- examples/novice/N03/visTutor/exN03Vis13.mac
  does one animation, more complex since it includes camera panning
  - 10 GeV EM shower showing light front
  - Camera follows (pans) at speed of light

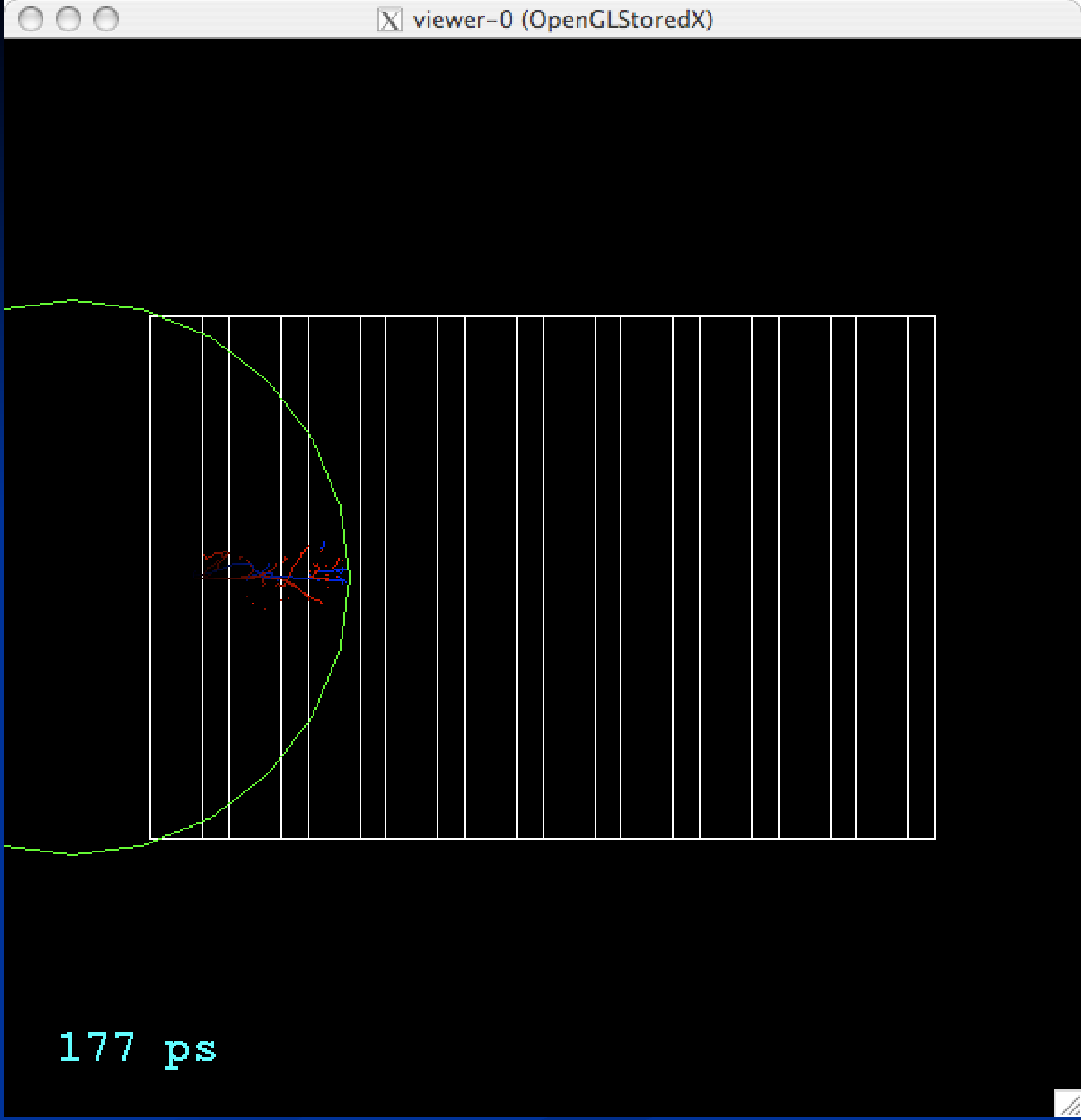
- To run either of them:
  - Make sure you have Geant4 Release 8.2 or newer
  - Build examples/novice/N03
  - Run N03 with one of the above macros, such as:
    - $G4WORKDIR/bin/Darwin-g++/exampleN03 visTutor/exN03Vis12.mac

Still from `exampleN03 visTutor/ exN03Vis12.mac`

1 GeV electron into lead-liqAr calorimeter

Demo live from Geant4

Unfortunately text is lost making a movie file (a current OpenGL eps “feature”)



Some More Example Animations

- The following is a set of example animations prepared by John Allison.
- You can check them out yourself from John’s movies page:
  - http://www.hep.man.ac.uk/u/johna/pub/Geant4/Movies/

50 MeV electron into lead-liqAr calorimeter

e^- red

e^+ blue

green

Yellow circles are step points -- geometrical boundaries or physical processes

Single50MeV.mpg

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

10 GeV electron into lead-liqAr calorimeter over 800 ps Single10GeVSlow.mpg

Gammas filtered out

```
/vis/filtering/
trajectories/create/
particleFilter
/vis/filtering/
trajectories/
particleFilter-0/add
gamma
/vis/filtering/
trajectories/
particleFilter-0/
invert true
```

Electrons red
Positrons blue

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

Same 10 GeV electron, camera follows by panning at light speed

Single10GeVFollowed.mpg

Shows light front (wave front of light starting at same time and place)

```
/vis/ogl/set/
displayLightFront
true -90 0 0 mm
```

Time range 800 ps

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

100 MeV $^{+}$

pi+100MeVmu+e+.mpg

Stops, decays to

$^{+}$

Stops, dallies (meanlife 2 s), decays to e^{+}

Annihilates to gamma-gamma

Compton scattering

Duration 4 s

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

10 Gev $^-$

pi-10GeVEMShower.mpg

N04 hadronic physics

Interacts early (potential confusion with EM shower)

Produces EM shower, presumably via charge exchange to ^{0}

Neutrons also produced

$^+$ magenta
$^-$ cyan
n yellow
 green
Others grey
Duration 2 ns

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

10 Gev $^-$

pi-10GeVNeutrons.mpg

N04 hadronic physics

Produces neutron chain reaction

filtered
e- red
e^+ blue
$^+$ magenta
$^-$ cyan
n yellow
green
Others grey

Duration 2 ns

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

Another 10 GeV $^-$

pi-10Gevpi+neutronSideView.mp4

3 fast $^-$

$^+$ interaction

3 ns

Mpeg4 encoding
with QuickTime Pro

QuickTime and a
decompressor
are needed to see this picture.

Same

pi-10Gevpi+neutronZ

Perspective view

Slow zoom

Detector drawing suppressed

3 ns

QuickTime and a
decompressor
are needed to see this picture.

Geant4 Visualization Resources

Geant4 Installation Guides

- http://geant4.slac.stanford.edu/installation

Hands on HepRApp Tutorial

- http://geant4.slac.stanford.edu/Presentations/vis/G4HepRAppTutorial/G4HepRAppTutorial.html

Hands on DAWN Tutorial

- http://geant4.slac.stanford.edu/Presentations/vis/G4DAWNTutorial/G4DAWNTutorial.html

Hands on OpenGL Tutorial

- http://geant4.slac.stanford.edu/Presentations/vis/G4OpenGLTutorial/G4OpenGLTutorial.html

Geant4 Visualization Commands

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

Geant4 Advanced Visualization

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

How to Make a Movie

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

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

- http://geant4.web.cern.ch/geant4/UserDocumentation/UsersGuides/ForApplicationDeveloper/html/

List of Visualization Commands:

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

For Questions or Comments: Geant4 Visualization Online Forum:

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