# *EsbRootView Progresses since Zagreb 2019*

Virtual ESSnuSB annual meeting

17 November 2020

Guy Barrand, CNRS/IN2P3/IJCLAB

# *EsbRootView / guiding ideas (1)*

- Have an event display able to run natively on all nice interactive devices that we have in hands today, by exploiting as much as possible local graphics capabilities of them.
- I have the graphics technology to do that for Linux/X11, Windows/Win32, macOS/Cocoa, iOS and Android.
- C++, local GL-ES and a scene graph logic of my own (strongly inspired by the great OpenInventor).
- Thesaurus of code and expertise accumulated for long, now on github under the generic name « softinex ».
- (Used also in G4/g4tools for doing IO and plotting).

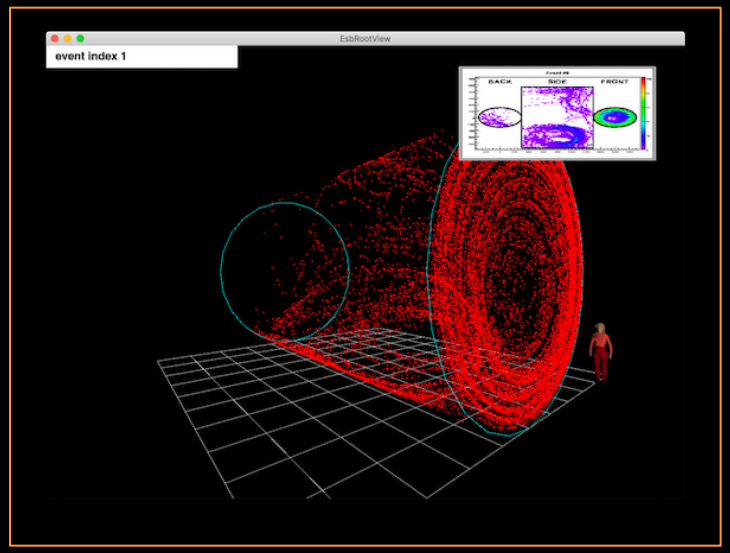
# *EsbRootView / guiding ideas (2)*

- Data access : today existing various HEP frameworks/stacks, handling detector and event models (and the critical IO), because though/targeted for the batch (and then Linuxes), are not ported natively on the « interactive » operating systems.
- (iOS and Android are science fiction for them). Which is a sad fact for anyone interested in HEP, interactivity and visualisation ☹.
- (This will not change before long).
- BUT, I can read detector and event ROOT files in a highly and light portable way. (softinex/inlib/rroot code) ☺

Use inlib/rroot over EsbRoot files and the softinex graphics classes to build EsbRootView.

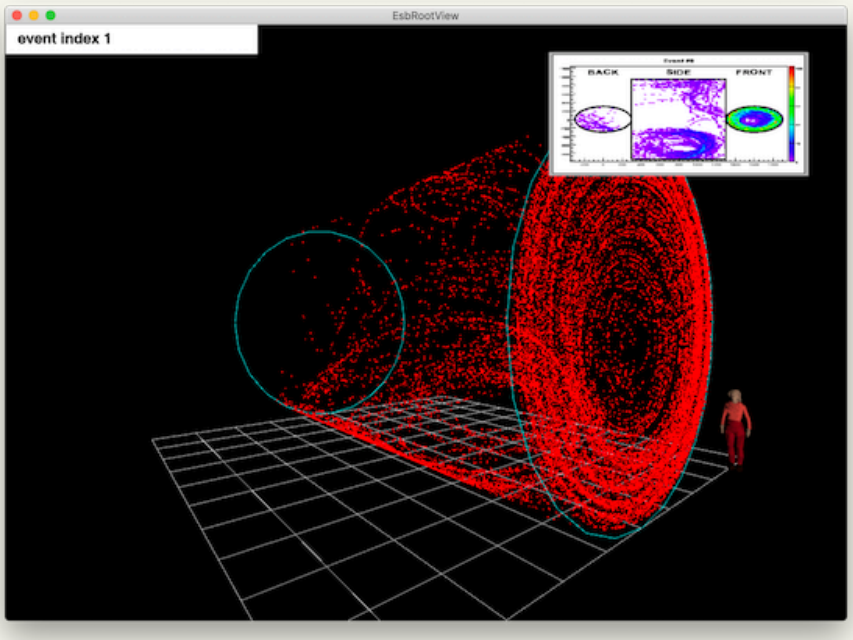
# *EsbRootView / 1.0.0*

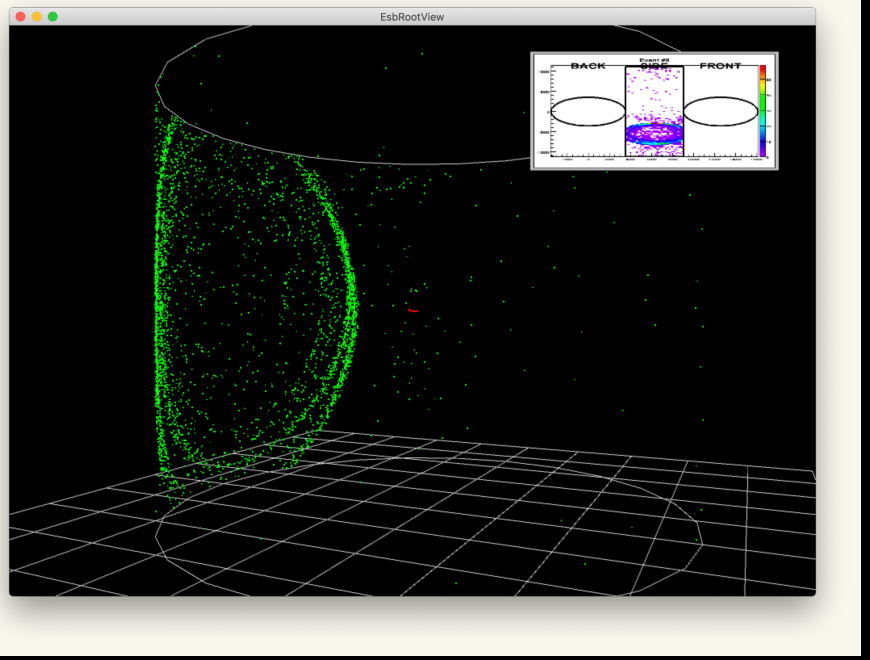
- Proof of concept/demonstrator that it is feasible.
- Read the geo_full.root and evetest.root of first release of EsbRoot.
- Show the “wc“ cylinder and WCDetectorPoints only.
- Released May/2019 on github/gbarrand.

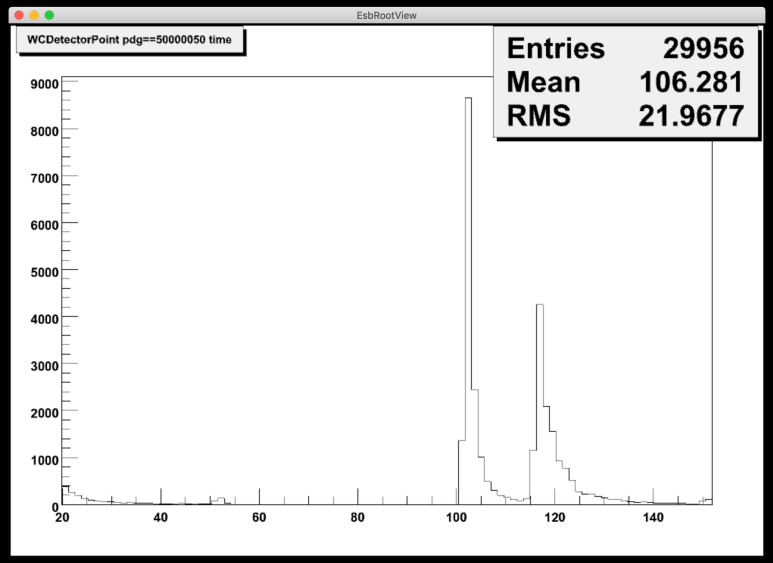


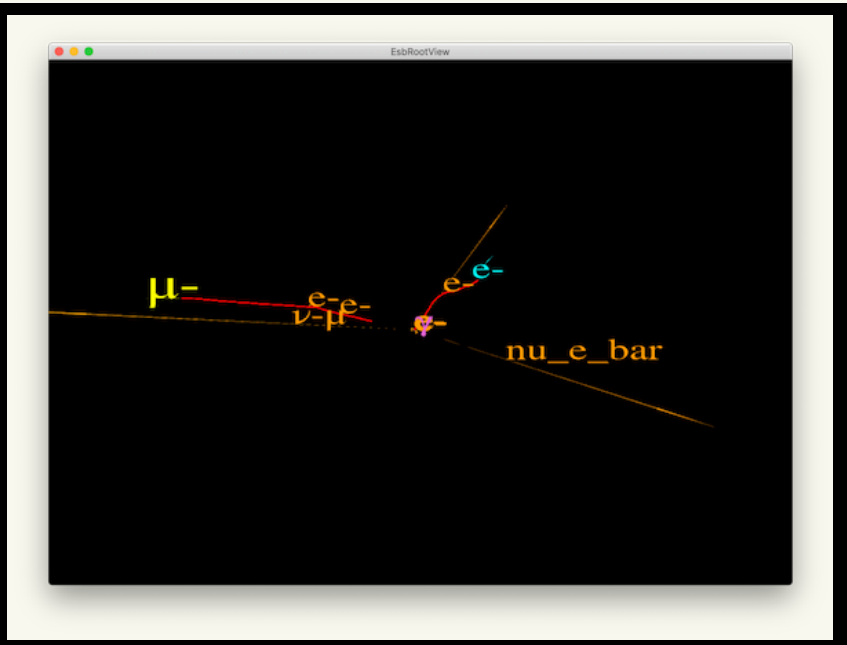
# *EsbRootView / 2.0.0/ More physics*

- Released end September 2019.
- Now available from : https://gbarrand.github.io
- Can cover neard, fard, fgd setups with the same program.
- MCTrack point and « arrow » representations.
- WCDetectorPoint, FgdDectorPoint and FgdHit : point rep.
- FgdHit : solid and wire-frame cube representations.
- Evolution in time for MCTrack/t, [WC,Fgd]DetectorPoint/time.
- A « cut/filter » mechanism that permits to have a fine tuning of what we want to see: it helps a lot in understanding an event.
- A bash-like scripting to customize startup and event scenes (see web pages).

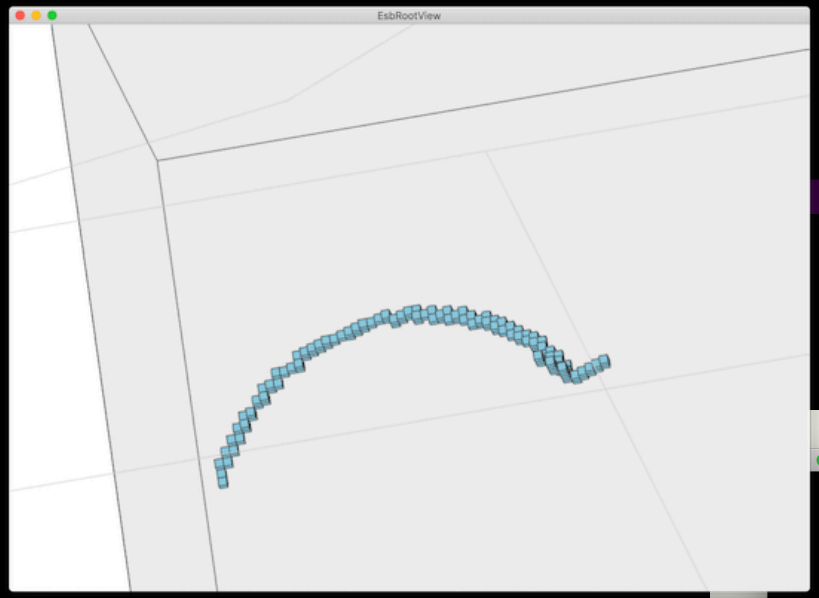

EsbRootView
event index 1
BACK
SIDE
FRONT

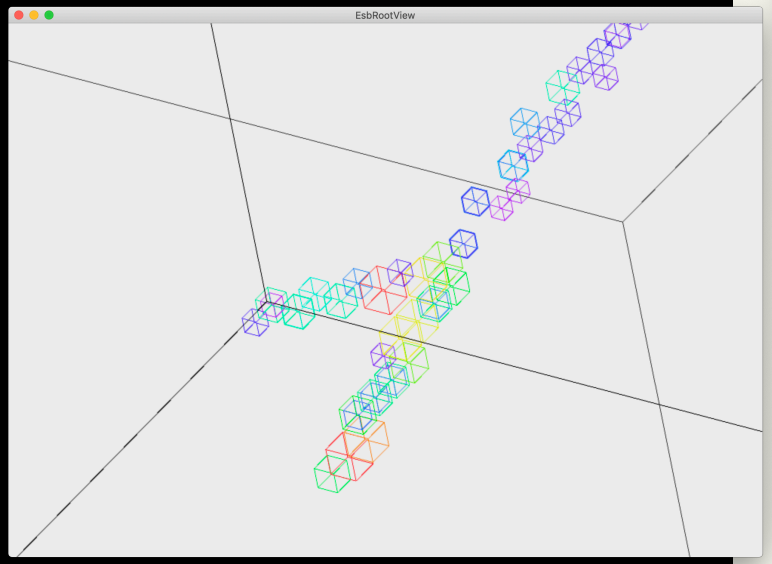

EsbRootView
BACK
FRONT

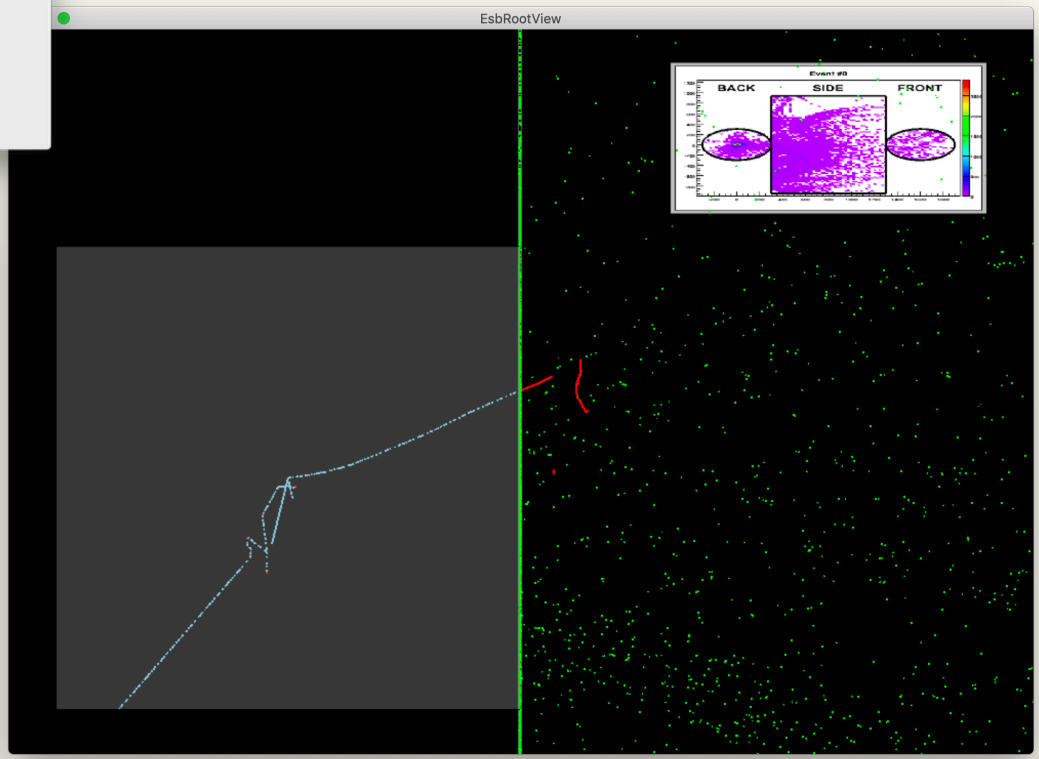

EsbRootView
WCDetectorPoint pdg==50000050 time
Entries 29956
Mean 106.281
RMS 21.9677
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
20
40
60
80
100
120
140


EsbRootView
μ-
e-
ν-μ-
e-
e-
nu_e_bar

# *Fgd points and hits*



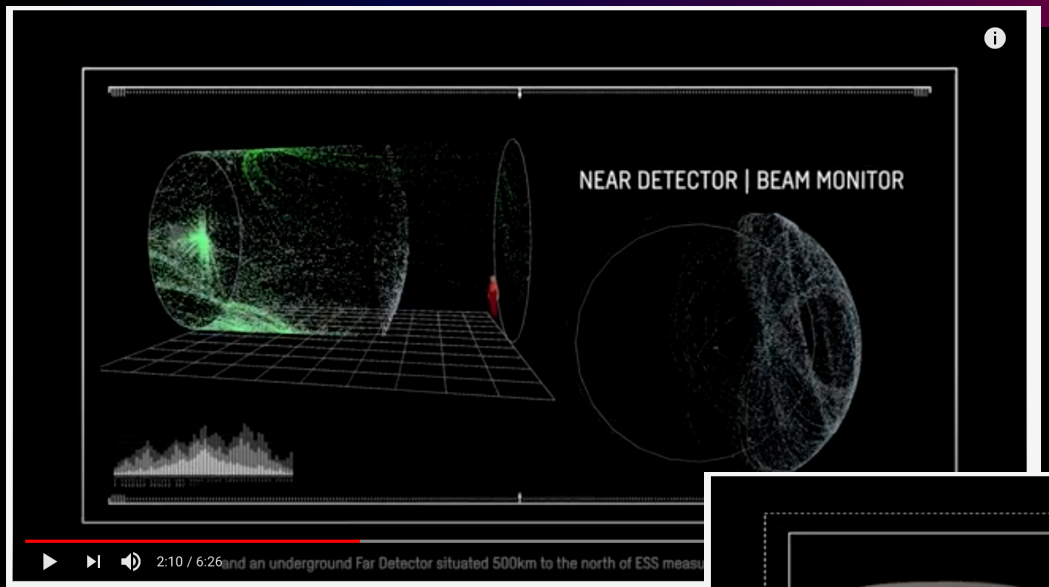



# *EsbRootView / 3.x*
# *Progresses since Zagreb*

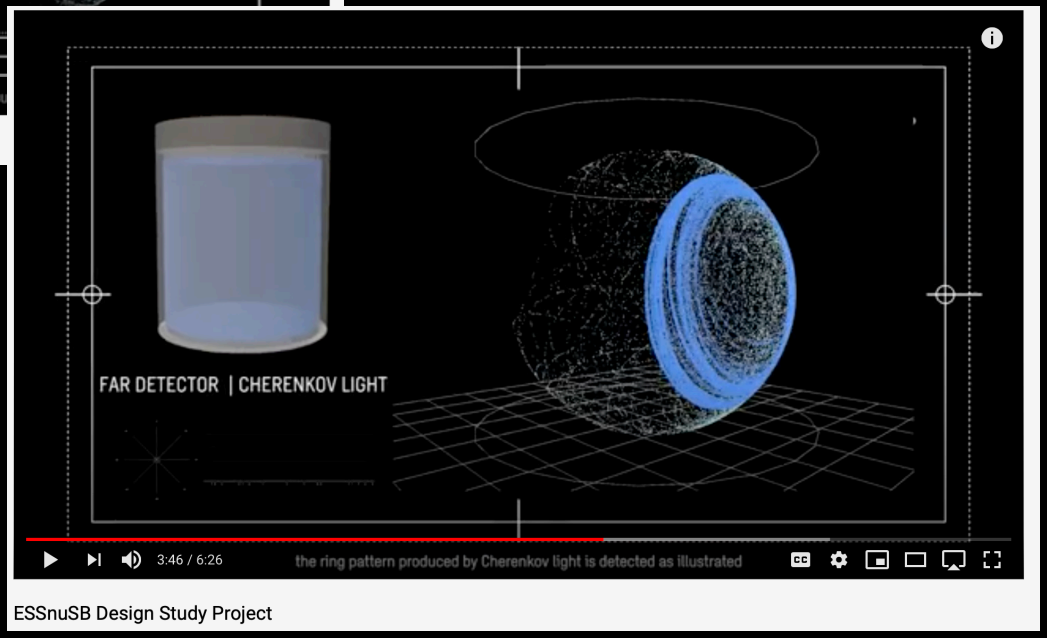
# *Optical photons deployment in time*

- From the MCTrack infos (position + momentum) of optical photons coming from G4, do a “straight line” tracking of them per time slice => very nice animations !
- Round trips with Tord to produce sequences for outreach videos.
- For neard and fard, around 10 seconds included in the “ESSnuSB Design Study Project” video available on YouTube.

# *ESSnuSB Design Study Project video*




ESSnuSB Design Study Project




ESSnuSB Design Study Project

# *A WebAssembly version*

- Have a 3.1.0 ported on WebAssembly.
- Permit to have the display running in web browsers on most devices (including iOS and Android).
- The display is running locally in a virtual machine embedded in a web browser (the wasm). The graphics is done by using WebGL.
- Need to have a good connection to load things !
- For the moment it is the “full display”,  we may think to have a more light and “funny oriented” version for outreach…

# *Apple/Metal and M1*

- Apple WWDC June 2018 : deprecation of Apple OpenGL.
- In favour of their own “Metal” rendering library.
- Due to the impact of Apple concerning interactivity, we can’t ignore that.
- During summer, have a inlib “renderer” class dealing with Metal.
- I have now a version of EsbRootView running straight with Cocoa (for the windowing) + Metal (for the rendering). Not yet released.
- (It had been a pain to have that).
- We are ready to enjoy the “M1” processors announced last week for Macs.
- Experience gained here will be reused in Geant4/vis system…

# *Conclusions*

- No “more physics” since Zagreb.
- But strong technical progresses around animations, handling the web and handling Apple new software and devices.

- CHEP people announced last week a “virtual CHEP” in May with publication of proceedings; a definite opportunity to submit a “EsbRootView display” paper to describe progresses done up so far (and then have a reference).