Warsaw University of Technology

ALICE

ALICE Track visualisation options for LHC Run 3

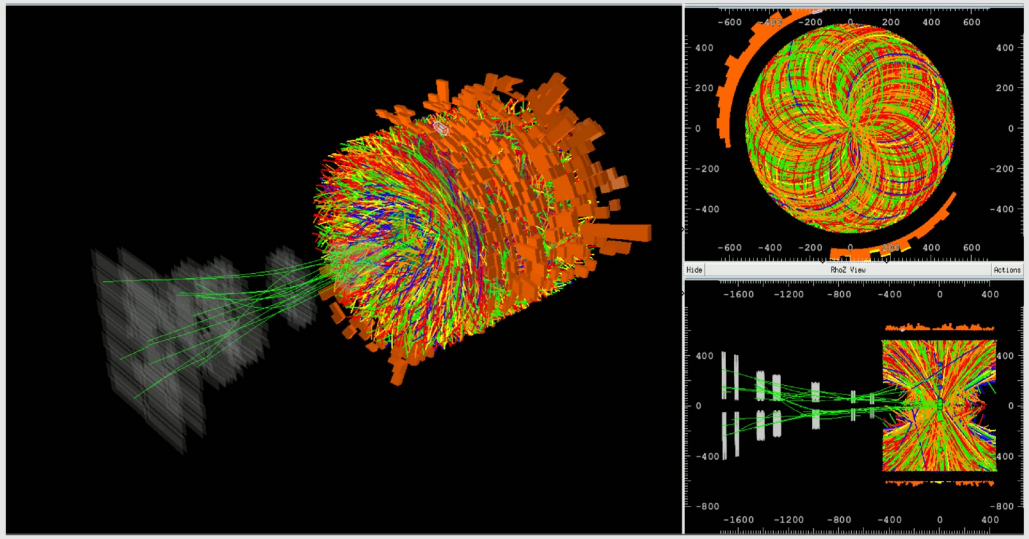
Julian Myrcha, Piotr Nowakowski, Przemysław Rokita for the ALICE Collaboration

Agenda

- Run 3 demands
- Time in visualisation
- Progressive visualisations
- Do we really need to visualise tracks
- Improvements in track visualisations
- Results

Visualisation

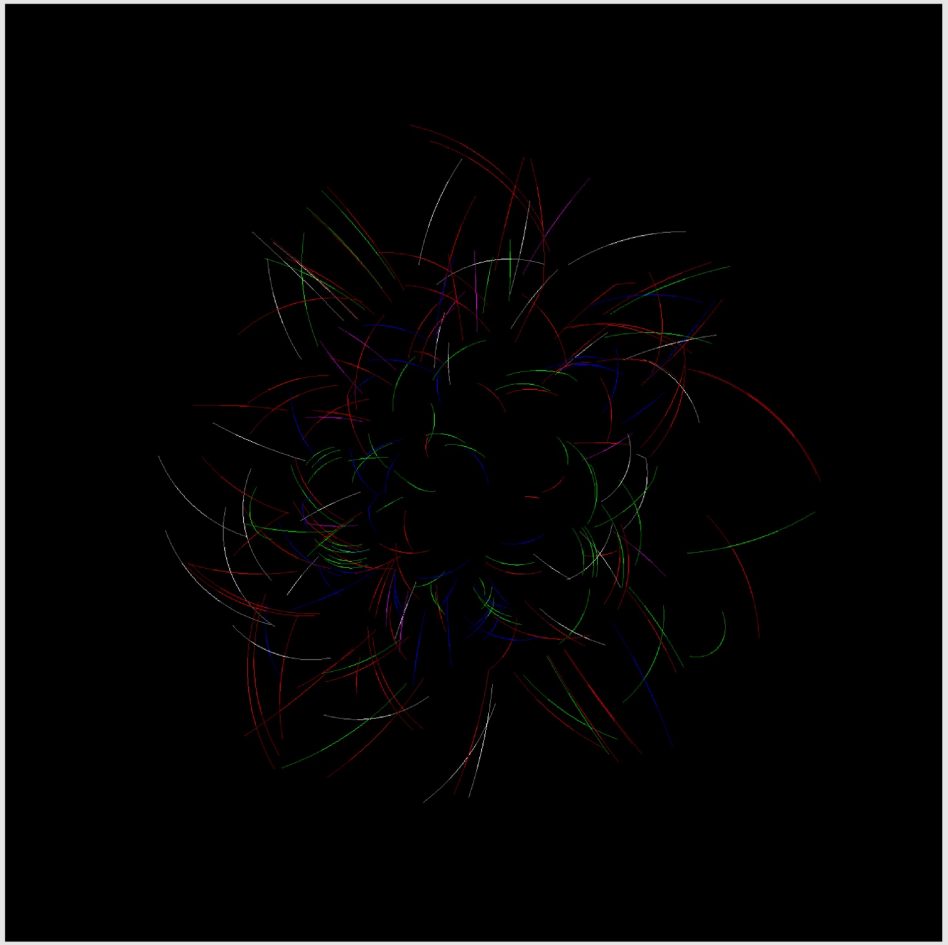
- Current – drawing tracks
- Pros
  - Looks nice
  - Easy to imagine for non-professionals
- Cons
  - Difficult to see details if there are many tracks



ALICE run 3

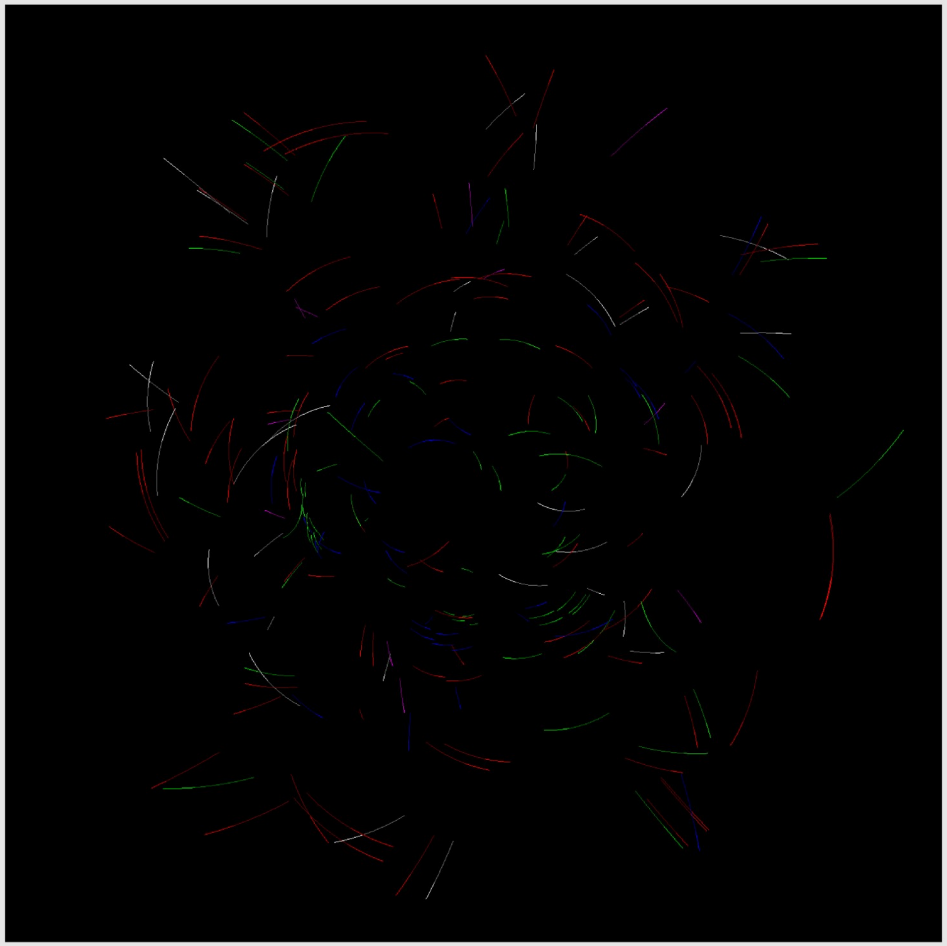
ALICE Run 3

- real time filters
- time dimension
- Hierarchical navigation
- Possibility to debug algorithms and detectors



Visualisation – dynamics

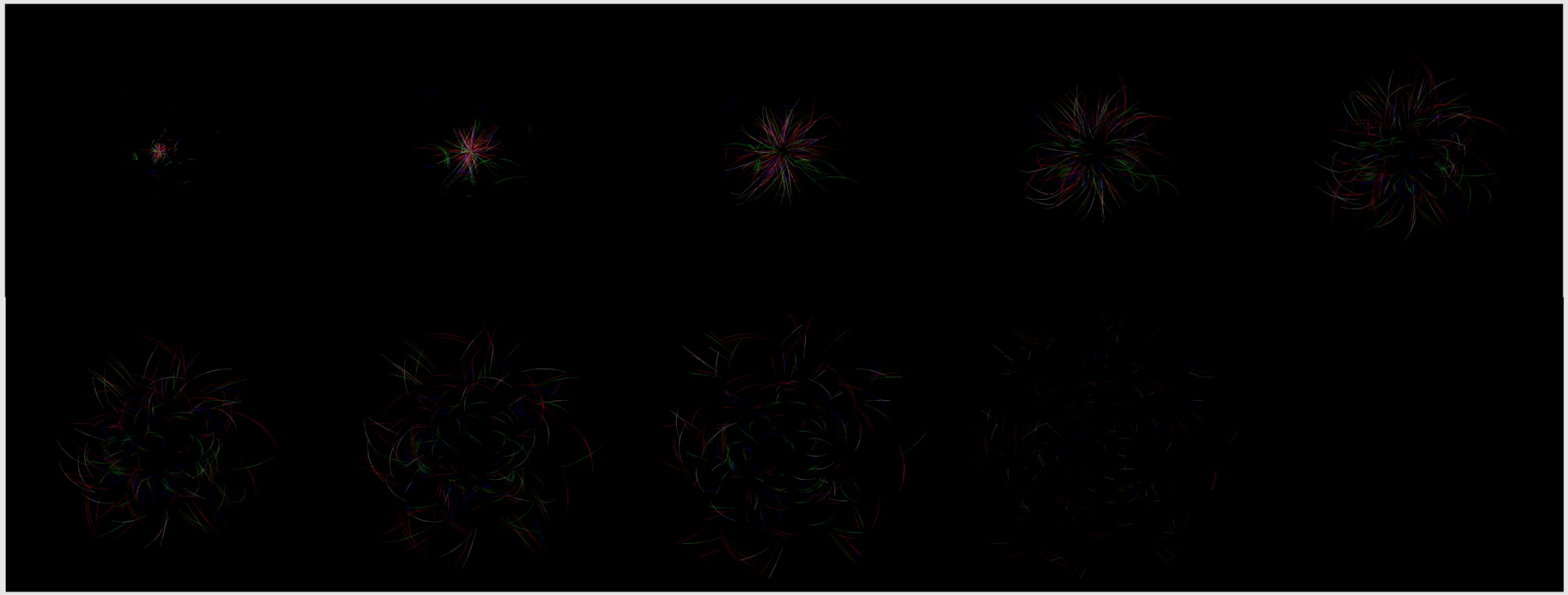
- We see a cumulative snapshot of the event
- It is interesting to observe how the system evolve in the time
  - Adding track filtering
  - Adding the animation – the incremental drawing
  - Drawing consecutive frames on the same image, but moved by (time frames)



Incremental drawing

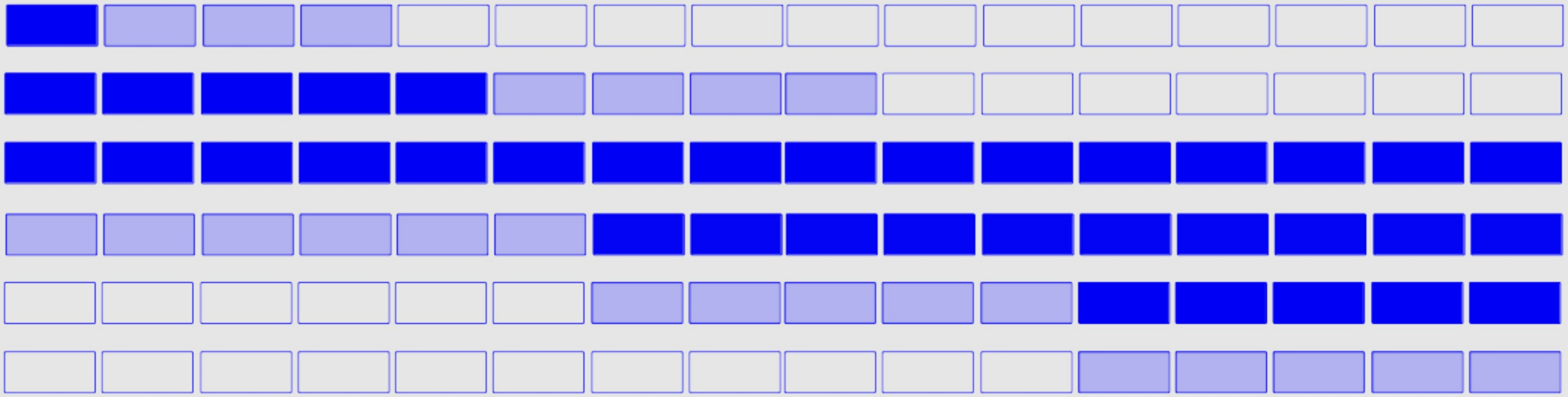
Incremental drawing

- Current visualization shows whole tracks
- Track animation may improve visual attractiveness for visitors



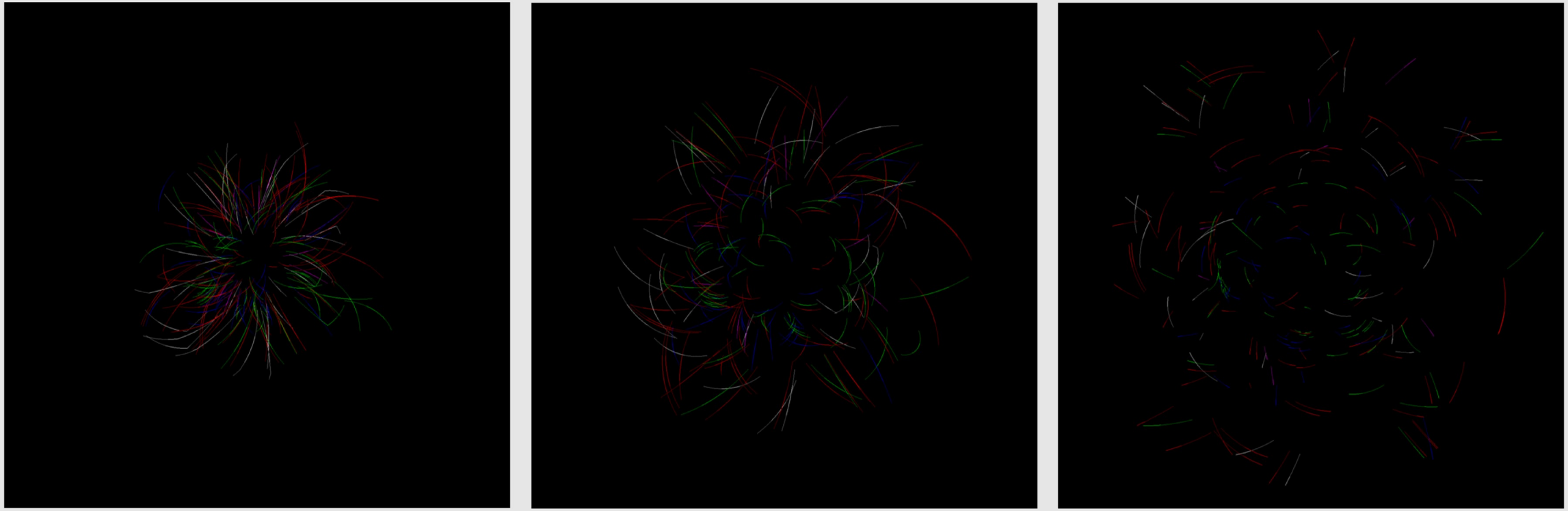
Incremental drawing

- Current visualization shows whole tracks
- Track animation may improve visual attractiveness for visitors
- Below the same track displayed for six points in time



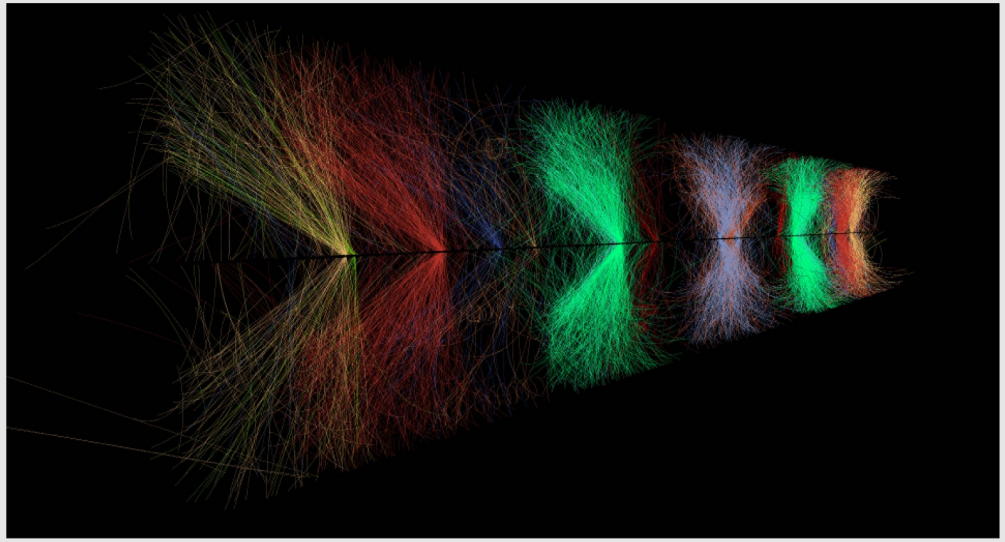
Incremental drawing

- Tracks parts appear and disappear



Visualisation – Time frames

- We can show visualisation evolving by drawing several snapshots on the same visualisation
- May be the same or different events
- This technique valid for tracks and for non-tracks visualisation



https://alice-o2.web.cern.ch/node/171

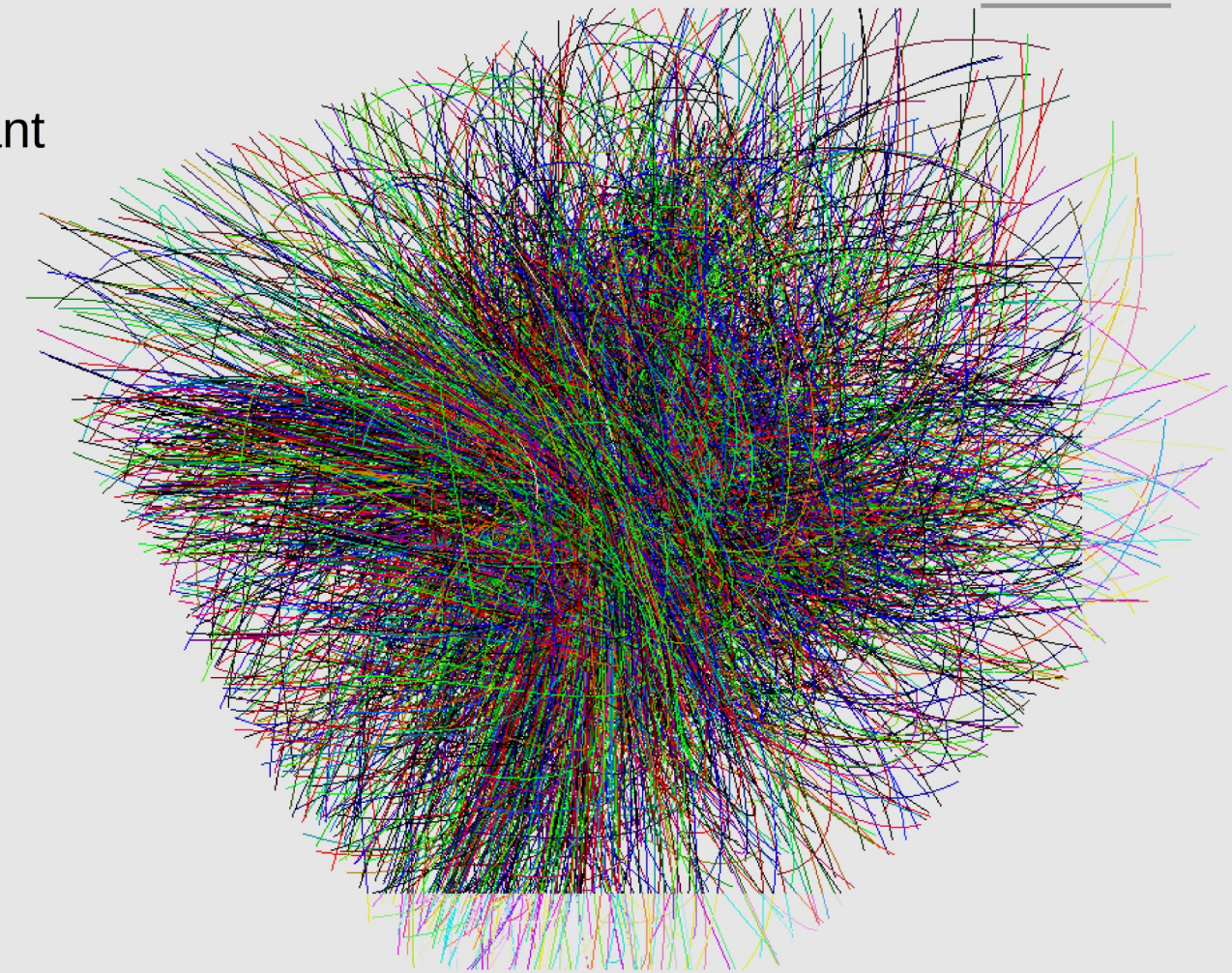
Visualisation – without a tracks

- We can abandon drawing tracks altogether
- Straightforward – live (cumulative?) statistics
- Combined – draw non-tracks information spatially (energy, particle types)
- Drawing volumes instead of the tracks
- Change in paradigm will be welcome – but to be found
- Example from the past were calorimeter towers

Improvements on track drawing

Track visualisation

- Improvements in track drawing still important
  - Event registered 25-11-2015 (Pb-Pb)
  - 6364 particles
  - Window size: 1280x720px
  - Measured frame rate in 10 seconds, averaged on 10 measures



Algorithms – OpenGL + Vulcan

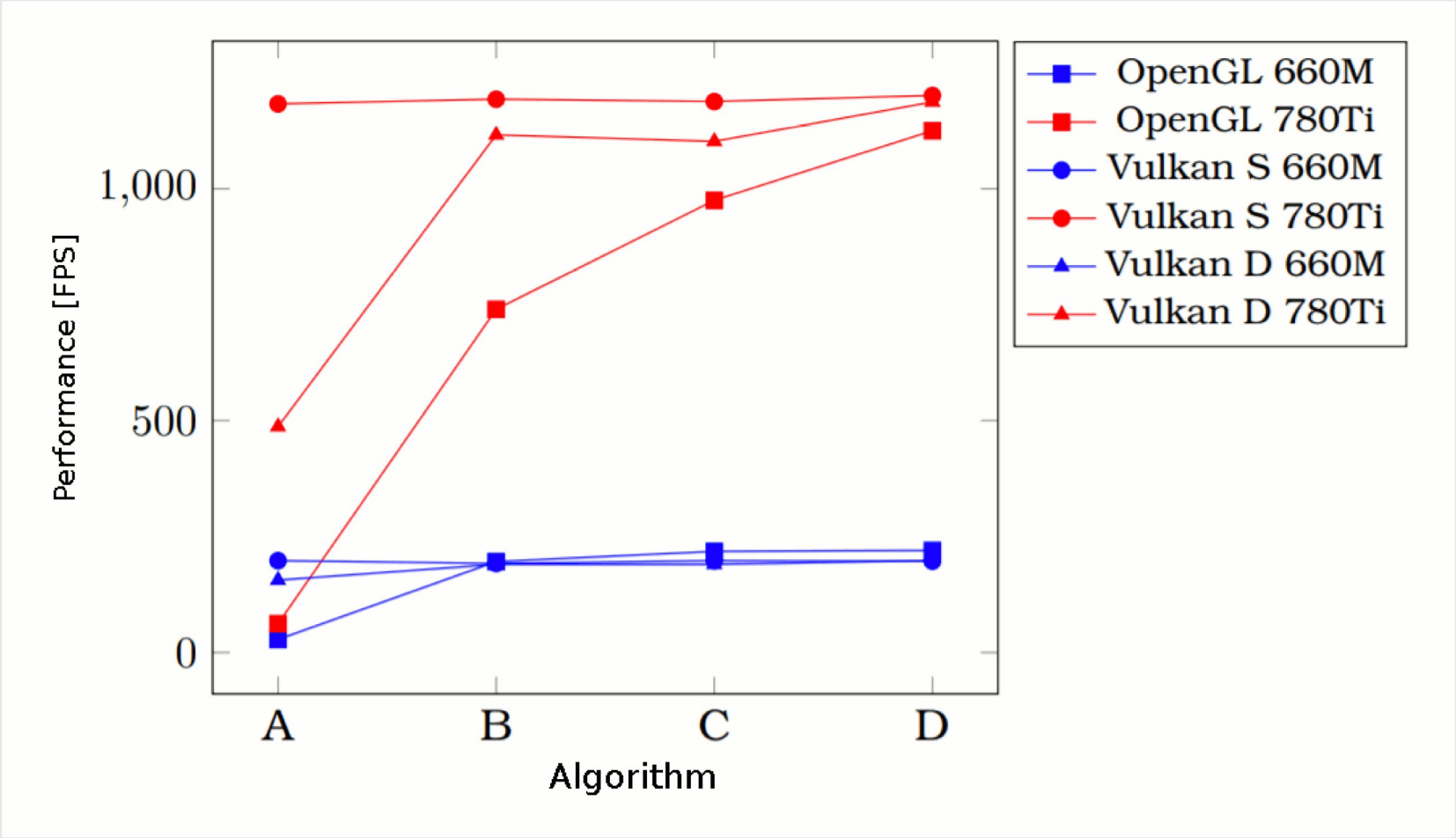
- Two technologies were compared
  - OpenGL (A 25 year old, but evolving standard, Linux,Windows, IOS)
  - Vulcan (A new Graphics API for Linux/Windows giving a much more control over the visualisation hardware)
- Four versions of drawing tracks
  - Version A (independent paths)
  - Version B (single buffer)
  - Version C (single command)
  - Version D (indirect drawing)

Support from the card vendors

Graphic Card	OpenGL 4	Vulkan	Metal
Intel (integrated)	*Ivy Bridge* (2012)	*Broadwell* (2015)*	
NVIDIA (dedicated)	*Fermi* (2010)	*Kepler* (2012)	
AMD (integrated)	*Llano* (2011)	*Graphics Core Next* (2011)	
AMD (dedicated)	*TeraScale 2* (2009)	*Graphics Core Next* (2011)	
Apple Inc.**	iMac, Mac Pro (2010)		iMac, Mac Pro (2015)

* For Linux here are open-source driver supporting *Ivy Bridge* (2012)
** Apple warns that it will discontinue support for OpenGL, but will support OpenGL ES for some time

Track drawing performance results



Summary

- Adding information about changes in time is a new factor worth to be considered
- Not only tracks may be visualized, time for new ideas
- There are many ways how tracks may be drawn
- Vulkan is no superior over properly written OpenGL
- Apple threatened dropping support for OpenGL, so maybe we should also drop support (for desktop graphics or for Apple)