Polyplopia

(from gr., πολύς - polús, "many,", and ὄψ-ops , "vision")

Coincident atmospheric shower events in IceCube

Juan Carlos Díaz Vélez - Simulation Workshop - 19 October, 2020

Polyplopia

(from gr., πολύς - polús, "many,", and ὄψ-ops , "vision")

Coincident atmospheric shower events in IceCube

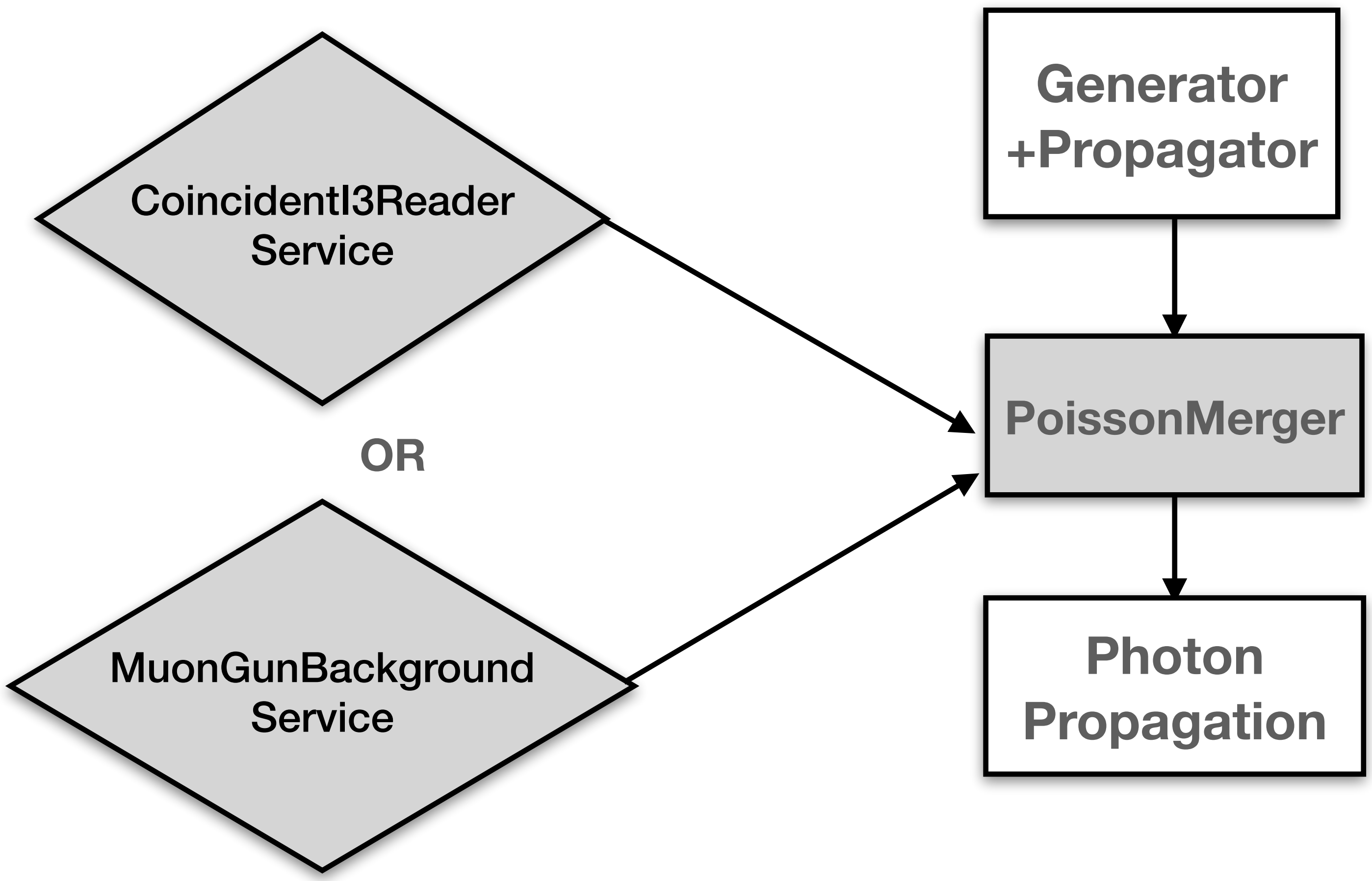
- poly**plopia::PoissonMerger**
  - Injects background event read from a separate file on top of primary events in the chain by sampling from a Poisson distribution over a time window Δt.
  - Also makes use of a *CoincidentEventService* that could be drop-in replaced with other event services such as a MuonGun-based service.
  - Writes a separate I3MCTree with background particles.
  - Writes a combined I3MCPE map for signal and background.

- poly**plopia::MPHitFilter**
  - Removes events that don't produce light in the detector and removes branches of I3MCTrees whose particles don't produce enough PEs in the detector,
  - Reduces the storage requirements.

- It is then up to the trigger-sim to split up Q-frames into P-frames events based on triggers.

CoincidentEventService

Inject cosmic-ray background from CORSIKA, generated with a "natural" spectrum/composition model (Hörandel polygonato) read from I3File.



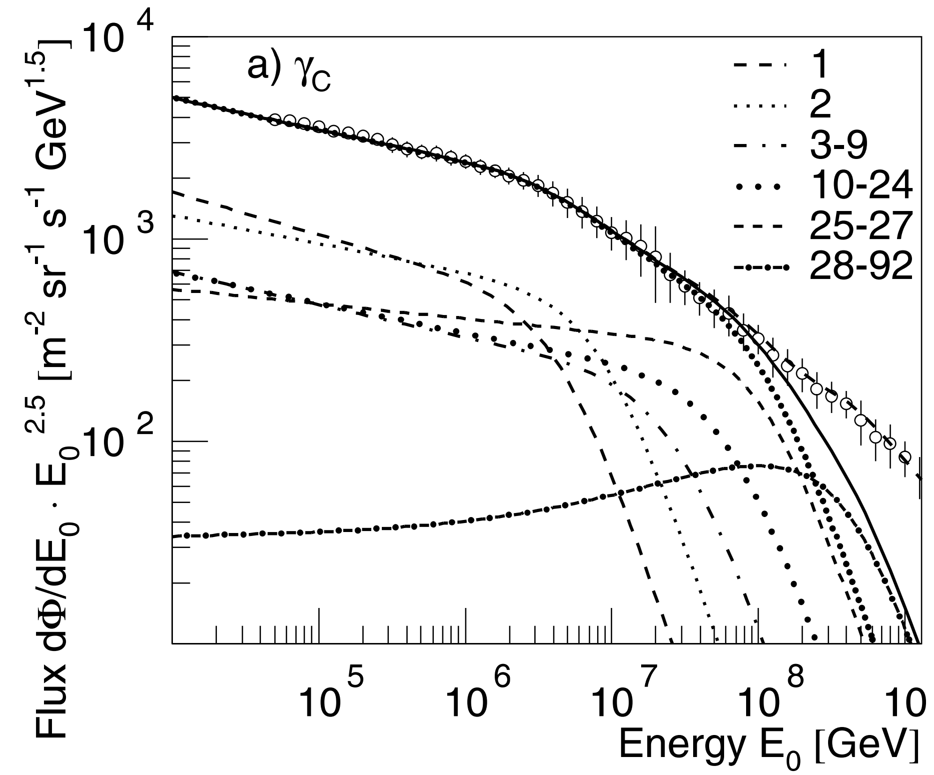


Fig. 11. Average all-particle energy spectrum. The line through the data represents a fit of the sum spectrum for elements with 1 6 Z 6 92 according to the poly-gonato model with rigidity dependent cut-off for common γ_c. The dotted line shows the spectrum for 1 6 Z 6 28. In addition, energy spectra for groups of elements are shown. Above 10^8 GeV the dashed line reflects the average spectrum.

An alternative to using full CORSIKA cosmic ray simulations is to replace the background shower file by a muon stream service implemented from the MuonGun generator configured to produce a muon spectrum and bundle multiplicity comparable to the one that results from the Polygonato cosmic-ray spectrum and mass distribution.

This approach, although less, accurate provides a much faster way to produce background coincidences and can produce muons on demand, thus saving a lot of computation time.

Number of coincident showers

Inject cosmic-ray background on top of primary or signal events in the main simulation chain by sampling from a Poisson distribution with a probability

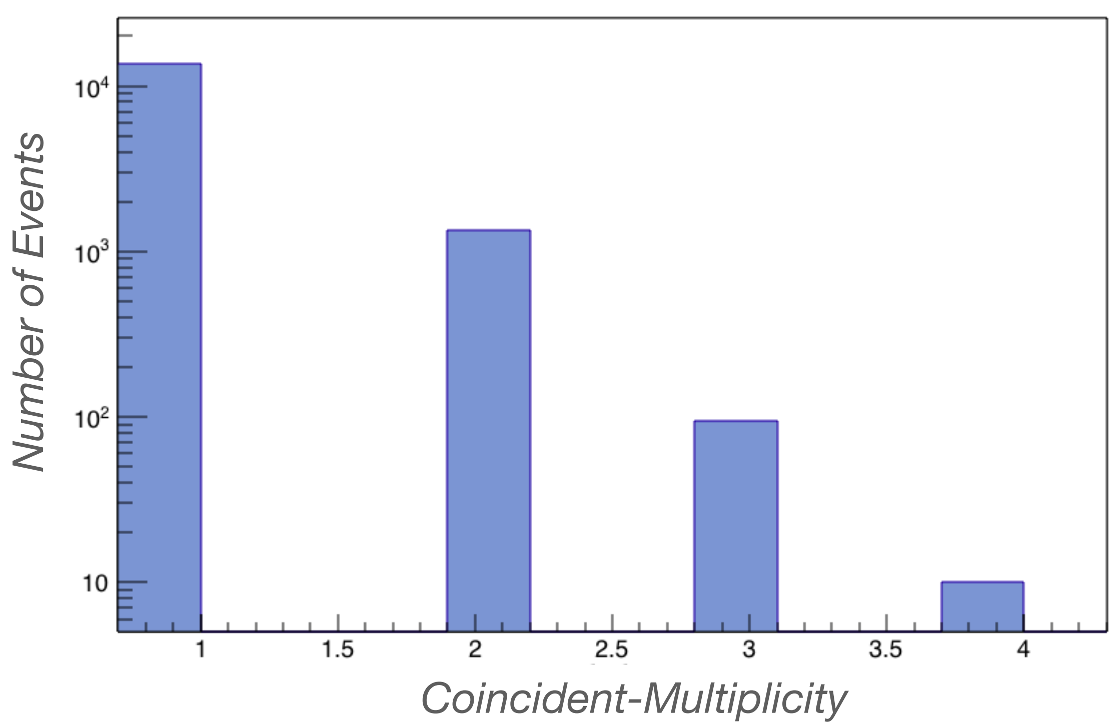
$$f(k,\lambda) = \frac{e^{-\lambda}\lambda^k}{k!}, \lambda = R\Delta t$$

where $\lambda = R\,\Delta t$, is the average number of CR shower muons (bundles) entering the detector volume within a time window Δt, given a rate R, and k is the number of coincident showers in that same interval. Background showers are injected with a uniform time distribution over the interval Δt.

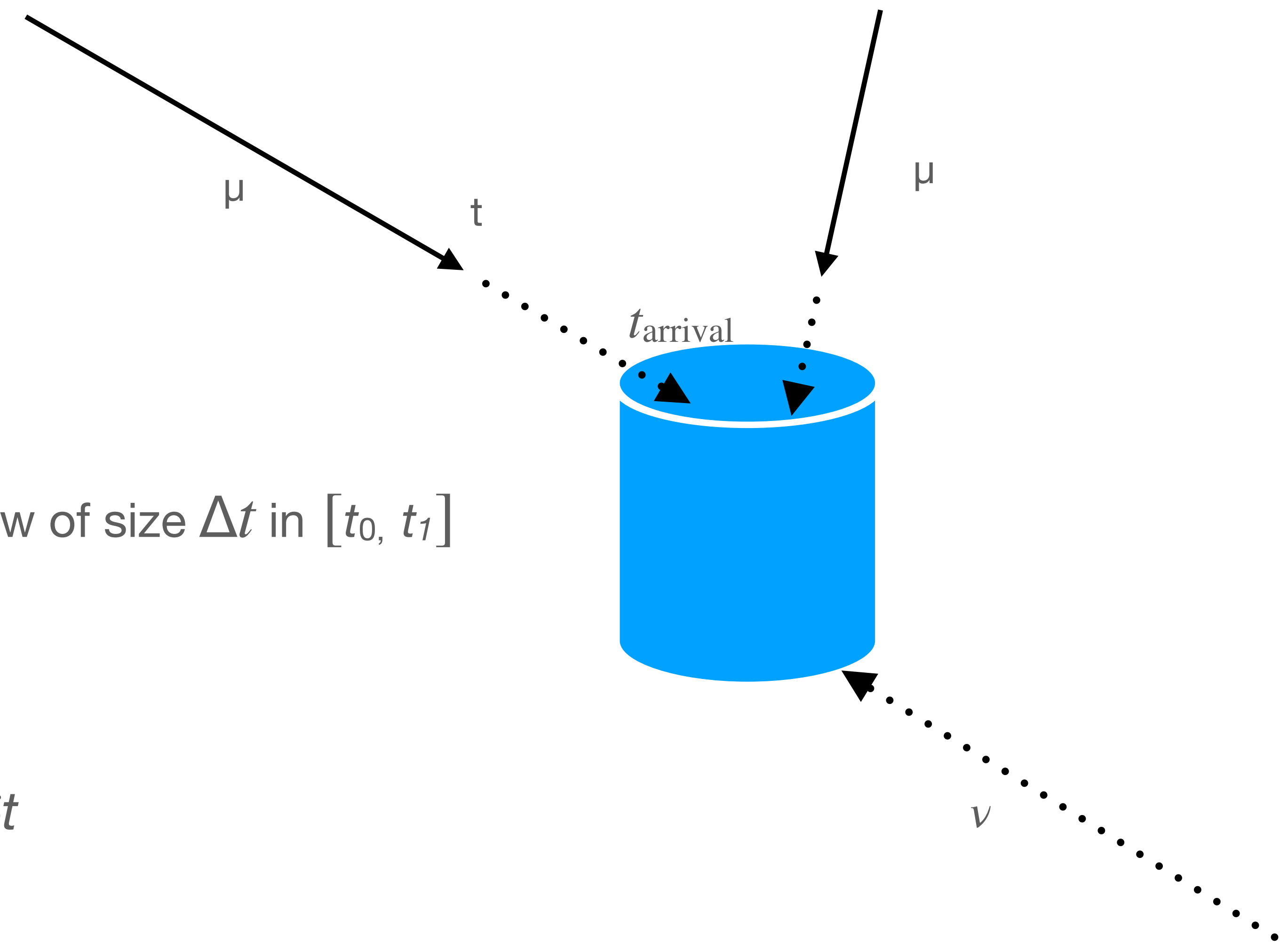
The default time window Δt is 40 μs but can be made arbitrarily large given enough memory needed to store each shower element, CPU needed to propagate each background shower, and GPU to propagate the emitted photons.

The background showers are assumed to be randomized in terms of energy and composition.

Same approach is used for injecting cosmic-ray muon background into both neutrino signal events as well as weighted cosmic-ray shower events. In the case of the latter, a single weighted cosmic-ray shower is treated as a "signal" but in this case k in would be replace by k-1 in order to avoid over-counting.



Arrival Times



In order to merge events prior to propagation we need an estimate of arrival time at detector

$$t_{\text{arrival}} = t + d/c$$

We sample a uniform distribution within the time window of size Δt in $[t_0, t_1]$

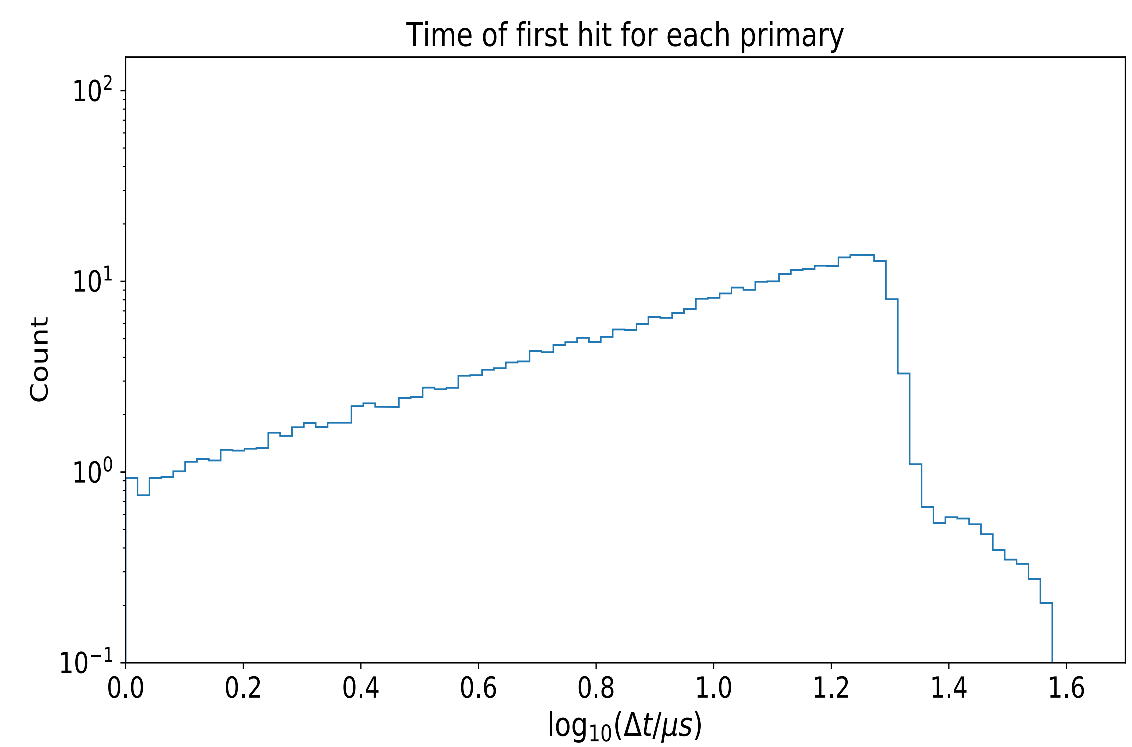
$$\delta t = \text{rand::uniform}(0, \Delta t)$$

And offset all objects in subtree s.t. $t_{\text{arrival}} \rightarrow t_0 + \delta t$

$$t' = t - t_{\text{arrival}} + t_0 + \delta t$$

$$t' = t_0 + \delta t - d/c$$

```
double TimeAtDetector(const I3Particle& p){
    return p.GetTime() - std::min(0., ((p.GetDir()*p.GetPos())/ p.GetSpeed()));
 }
```



Arrival Times

Events are randomized around the signal event with

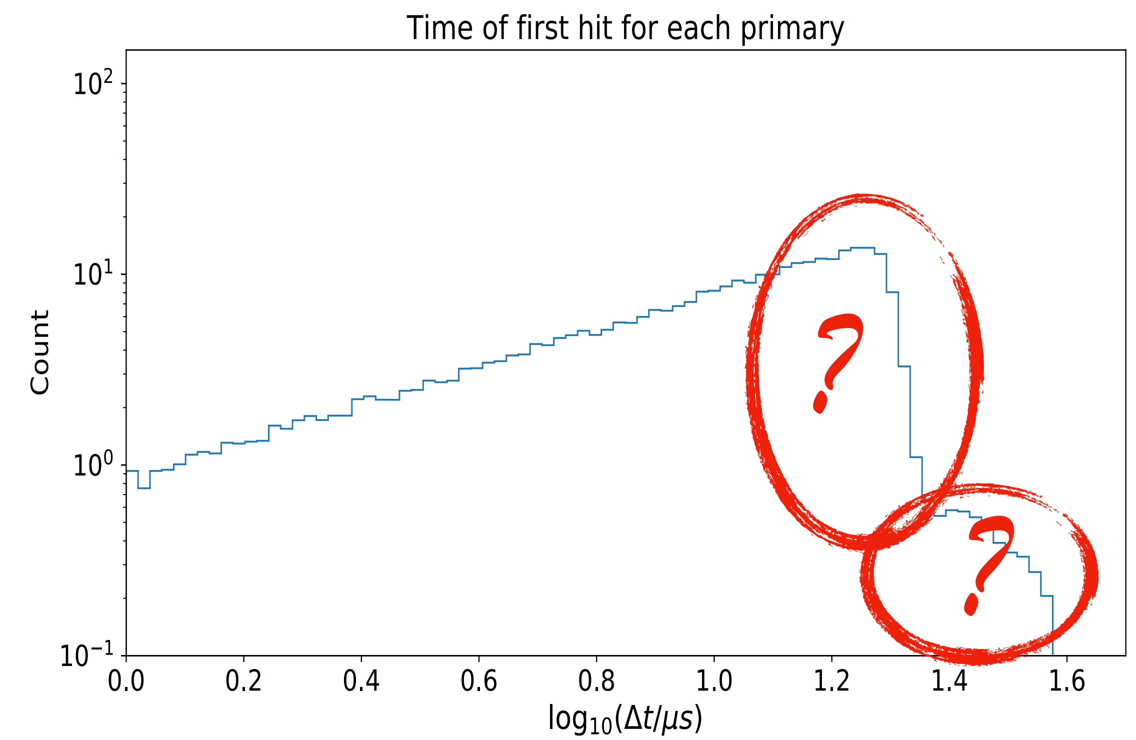
$$[-\Delta t/2, \Delta t/2]$$

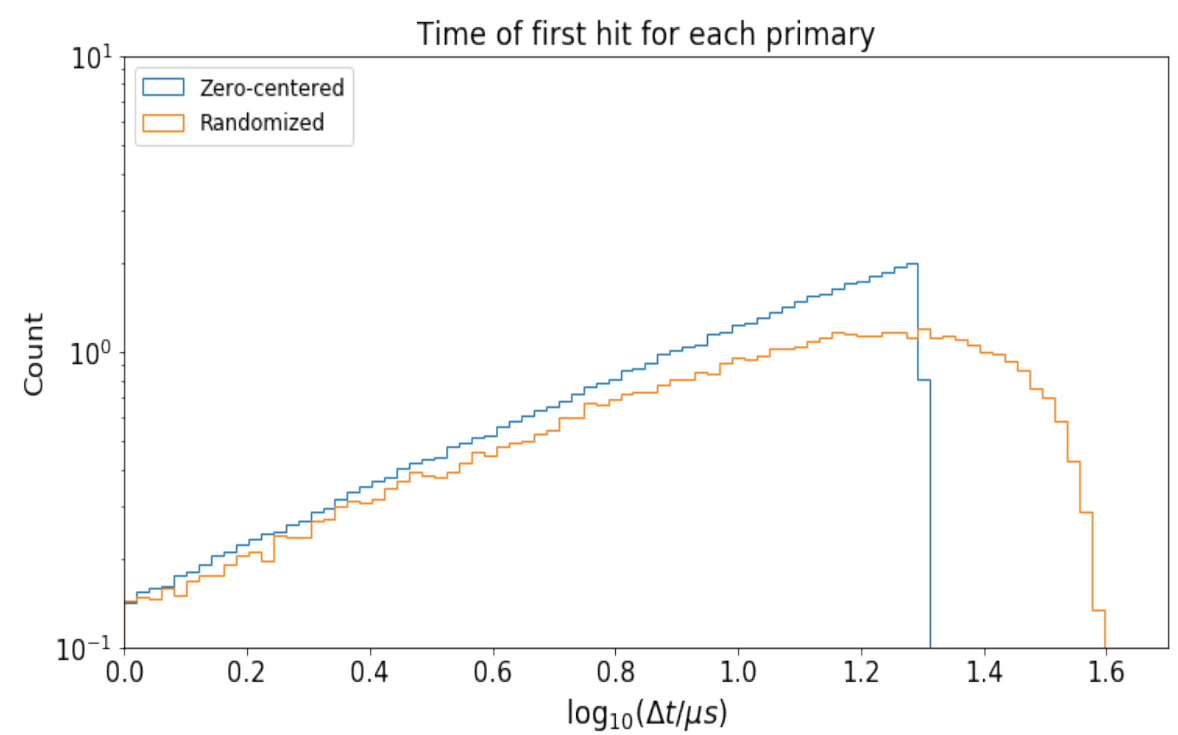
hence the signal event is always at t=0 and the maximum $\delta t = \Delta t/2$.

Is this correct?... probably not

What about the tail?

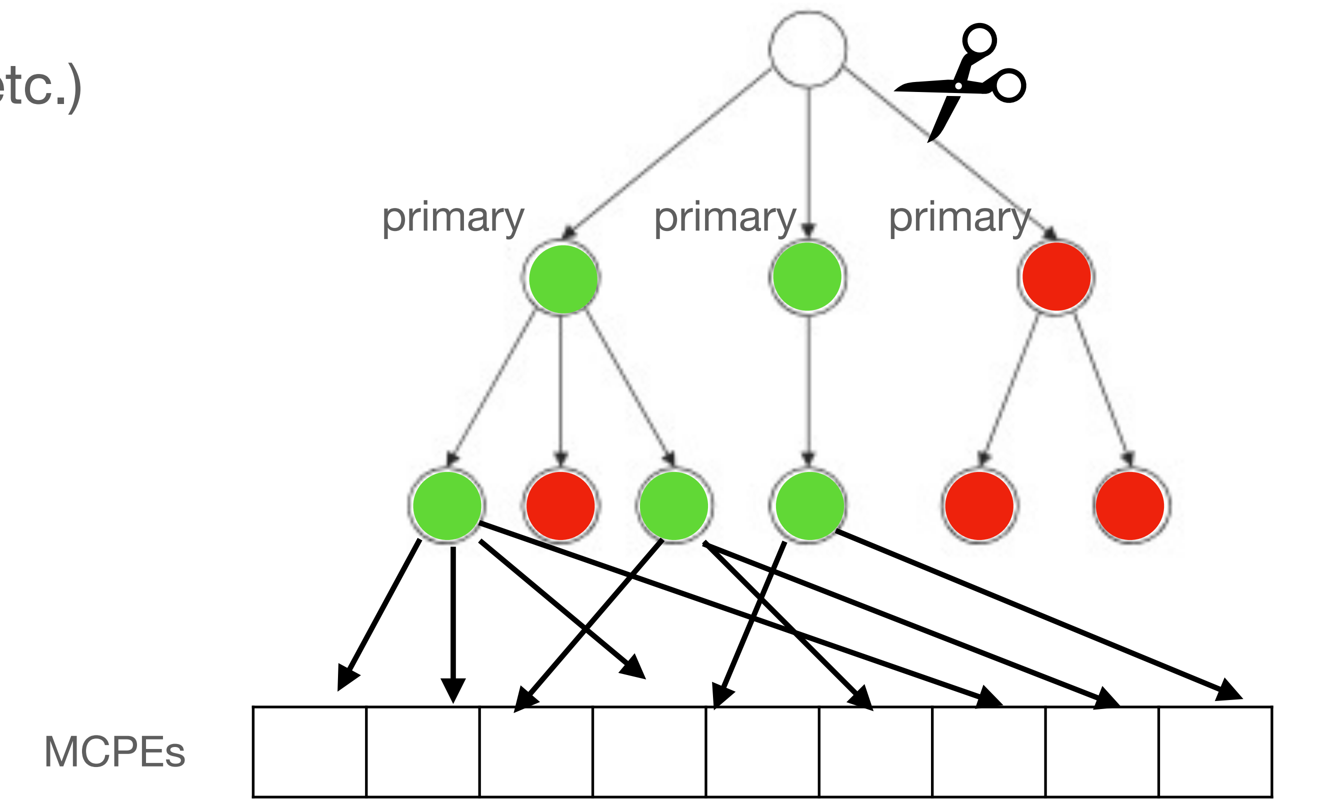
> A: This is just the spread of hit times for individual bundles





MPHitFilter

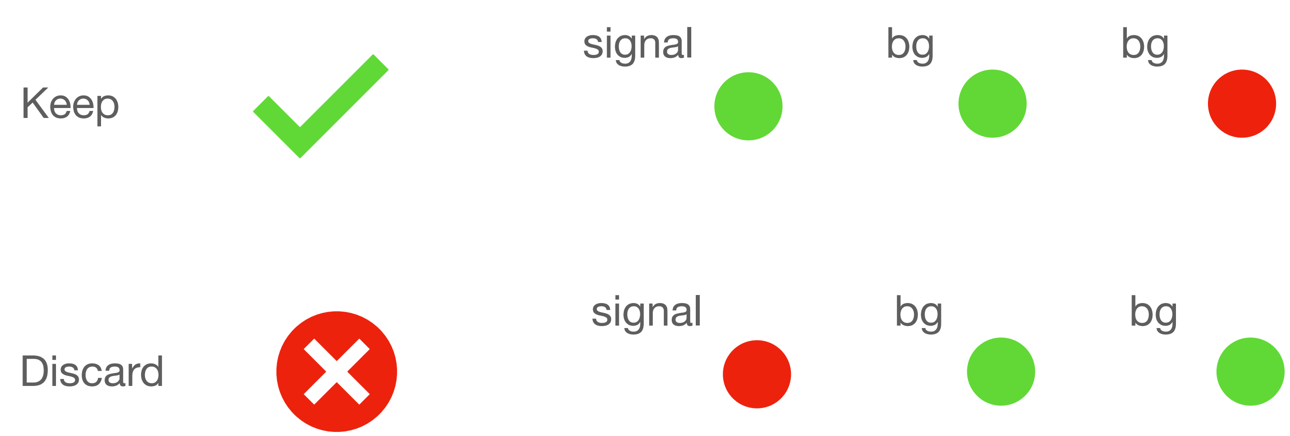
1. Polyplopia: Combine I3MCTrees (MMCTrackList, etc.)
2. Photon propagation (PPC, CLSim)
3. MPHitFilter
    a. Remove primary branches with no hits
    b. Discard frame if does not contain "signal" primary
    c. Write PolyplopiaInfo & PECounts to frame



You can run Polyplopia:

A. at generation-level then a single photon propagation. This ensures you are using the same ice model, DOMoversizing, hole ice, etc.

B. after propagating photons and removing events with no hits. This is more efficient in GPU utilization.



MPHitFilter

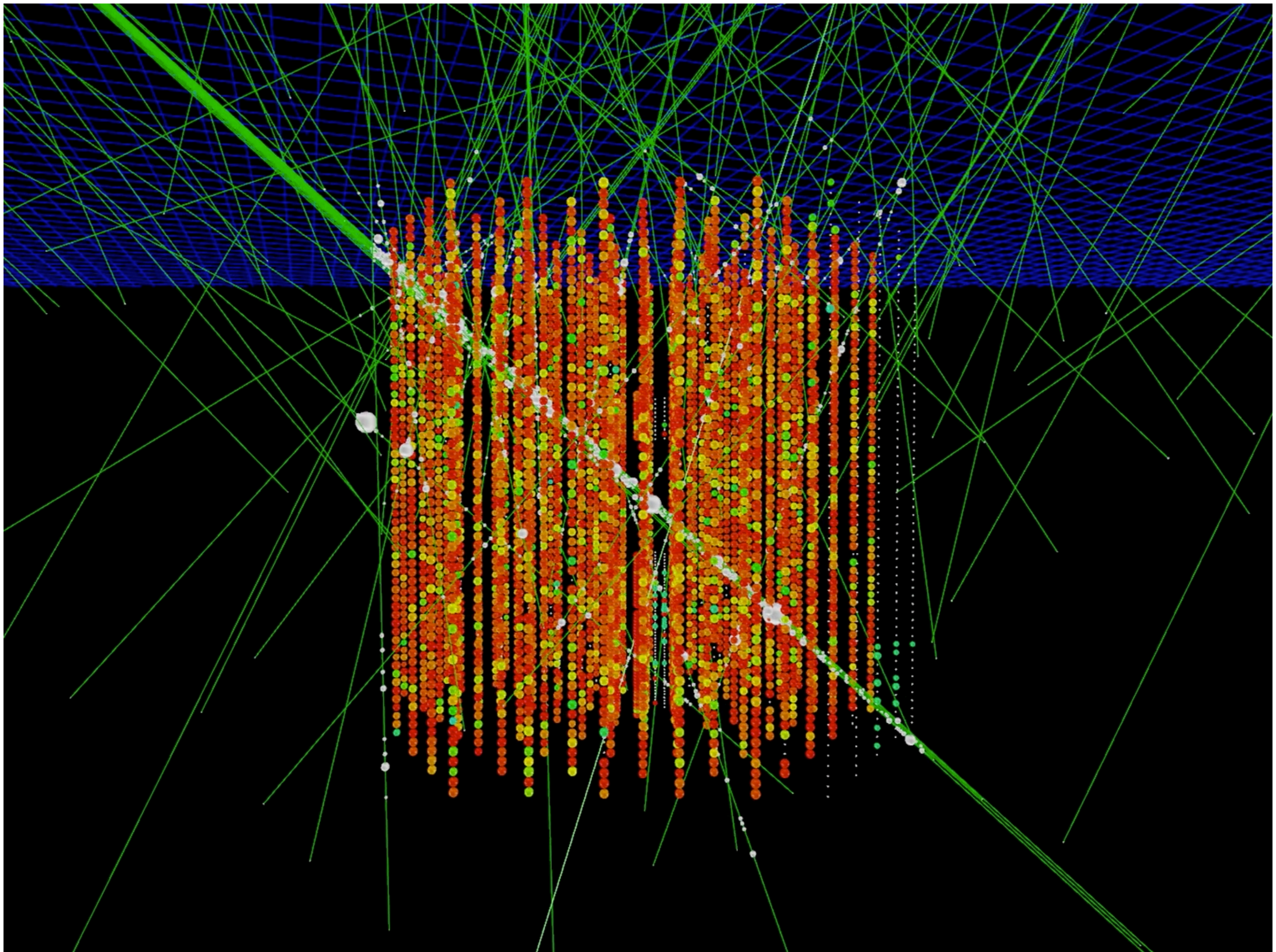
```
[gtx-00]$ dataio-pyshovel merged_pes.i3
 I3 Data Shovel                                                      Press '?' for help

 Name                           Type                                        Bytes
 BackgroundI3MCPESeriesMap      I3Map<OMKey, vector<I3MCPE> >               41
 BackgroundI3MCPESeriesMapPa... I3Map<OMKey, map<I3ParticleID, vector<unsigned in... 41
 BackgroundI3MCTree             TreeBase::Tree<I3Particle, I3ParticleID, i3hash<I... 32
 BackgroundI3MCTreePEcounts     I3Map<unsigned int, unsigned int>           47
 BackgroundI3MCTree_preMuonProp TreeBase::Tree<I3Particle, I3ParticleID, i3hash<I... 32
 BackgroundI3MCTree_preMuonP... I3GSLRandomServiceState                     85
 BackgroundMMCTrackList         I3Vector<I3MMCTrack>                        40
 I3MCPESeriesMap                I3Map<OMKey, vector<I3MCPE> >               41
 I3MCPESeriesMapParticleIDMap   I3Map<OMKey, map<I3ParticleID, vector<unsigned in... 41
 I3MCTree                       TreeBase::Tree<I3Particle, I3ParticleID, i3hash<I... 2902
 I3MCTree_preMuonProp           TreeBase::Tree<I3Particle, I3ParticleID, i3hash<I... 422
 I3MCTree_preMuonProp_RNGState  I3GSLRandomServiceState                     85
 I3MCWeightDict
 MMCTrackList
 NuGPrimary
 PhotonSeriesMap
 PolyplopiaInfo
 PolyplopiaPrimary
 SignalI3MCPEs
 SignalI3MCTree                 TreeBase::Tree<I3Particle, I3ParticleID, i3hash<I... 2902
```

```
PolyplopiaInfo [I3Map<string, int>]:
[BackgroundI3MCPESeriesMapCount => 20,
BackgroundI3MCTreePrimaries => 1,
I3MCPESeriesMapCount => 118,
I3MCTreePrimaries => 1,
Multiplicity => 2]
```

Polyplopia

(from gr., πολύς - polús, "many,", and ὄψ-ops , "vision")

Coincident atmospheric shower events in IceCube



http://icecube.wisc.edu/~juancarlos/video/10msec_w_noise.mp4