Online Event Selection using GPUs for the Mu3e Experiment

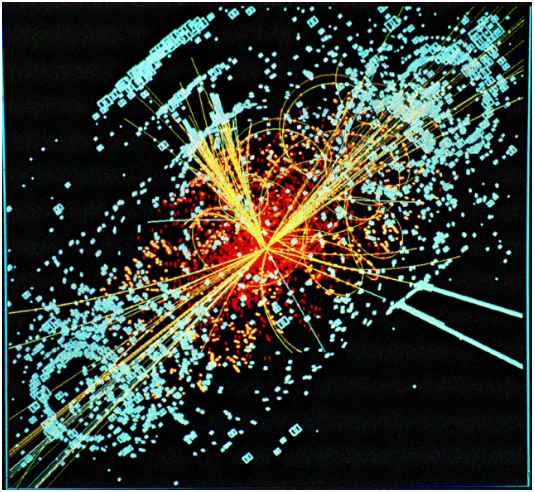
Valentin Henkys, Bertil Schmidt, and Niklaus Berger
henkys@uni-mainz.de

Departments of Computer Science and Nuclear Physics,
Johannes Gutenberg-University Mainz
For the Mu3e Collaboration

ISPDC 2022

Motivation

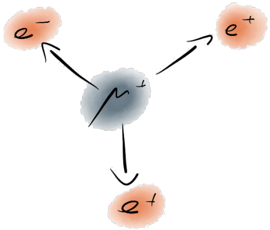
- Search for physics beyond the Standard Model
- $\rightarrow$ Need for increasingly precise experiments
- $\rightarrow$ High data rates expected:
  - 40 TBps LHCb [1],
  - 3 TBps ALICE [2],
  - 80 Gbps Mu3e Phase I [3]
- $\rightarrow$ Perfect for FPGAs and GPUs



Higgs Event. Source: CERN, CC-BY-SA

Mu3e Experiment [3]

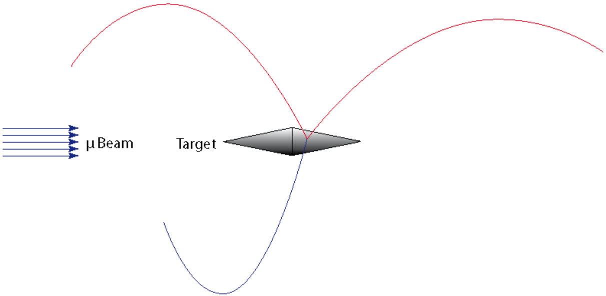
- Search for the Lepton Flavor Violating decay $\mu^+ \rightarrow e^+ e^+ e^-$
    - Previous upper bound on branching ratio was set by SINDRUM [4] to 10^{-12}
- Located at Paul-Scherrer-Institute (PSI) Switzerland
- 1 year of data taking for Phase I
- muon rate of $1 \cdot 10^8 \mu/\text{s}$ for Phase I, $\geq 1 \cdot 10^9 \mu/\text{s}$ for Phase II



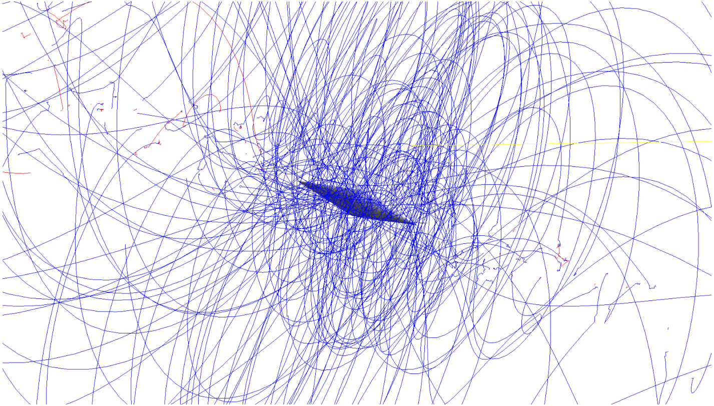
Detector: Beam setup

Signal event:

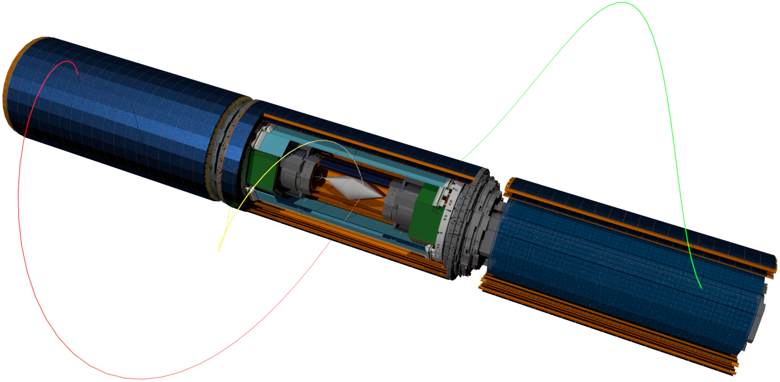
- Momentum and Energy conserved
- Same point of origin (*event vertex*)



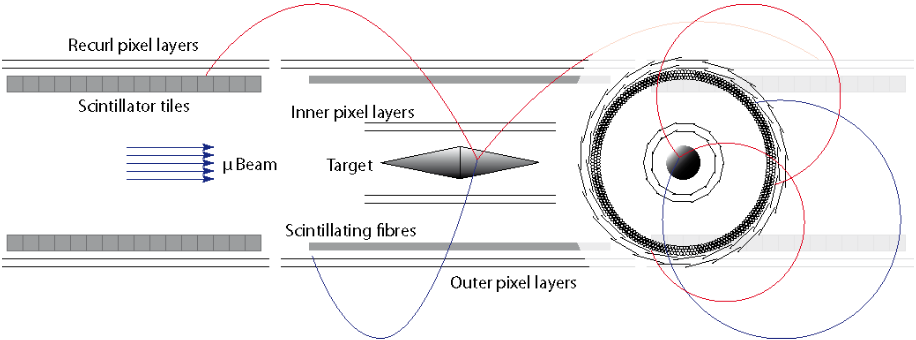
Detector: One Frame



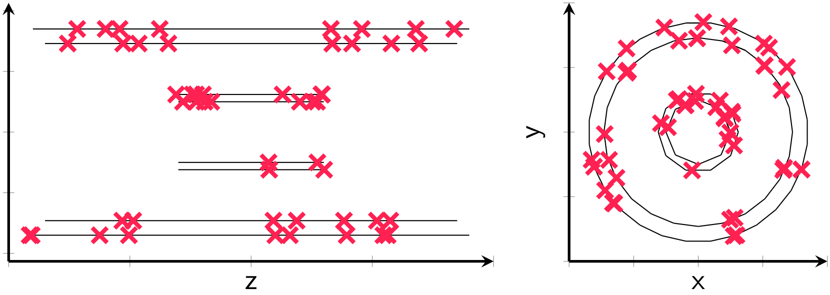
The Detector



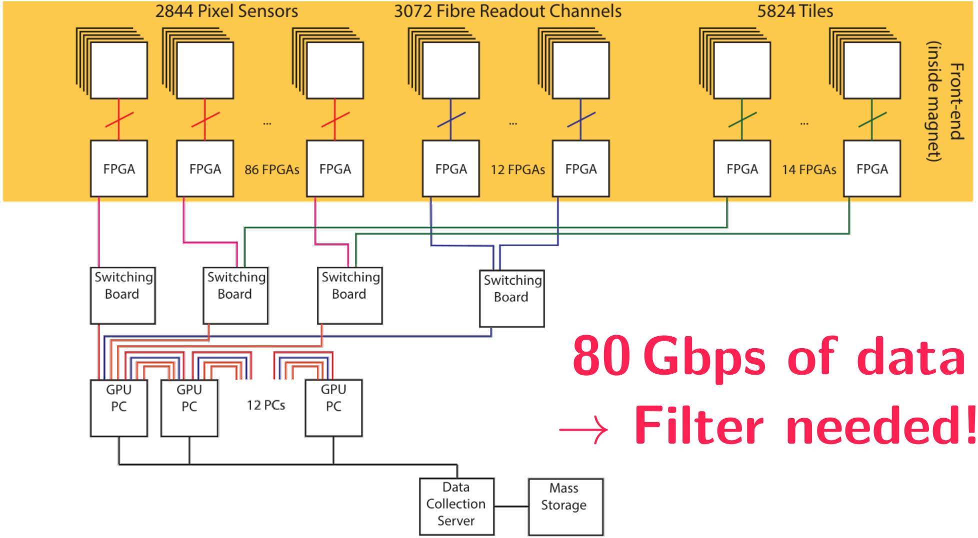
Detector: Setup Explained



Detector: One Frame of Data



Detector: Readout



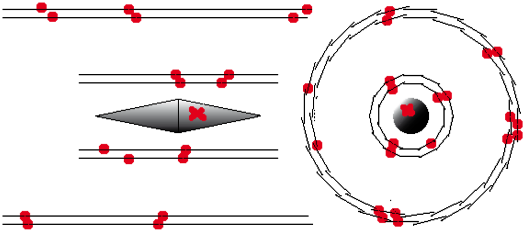
80 Gbps of data

→ Filter needed!

Datarate Reduction by Filtering

Online Event Selection

- Goals:
  - Identify signal-like frames
  - Reduce data rate by at least a factor of 100
  - Run in real time on GPUs
  - Based on [5]
- Outline:

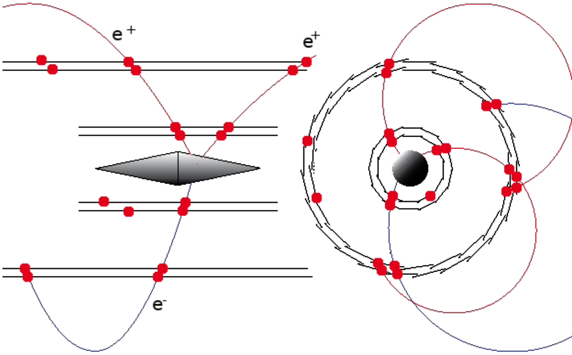


3. Check for possible event vertex

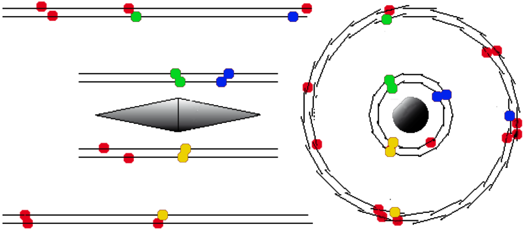
Online Event Selection

- Goals:
  - Identify signal-like frames
  - Reduce data rate by at least a factor of 100
  - Run in real time on GPUs
  - Based on [5]
- Outline:

  2. Find particle tracks
  3. Check for possible event vertex



- Goals:
  - Identify signal-like frames
  - Reduce data rate by at least a factor of 100
  - Run in real time on GPUs
  - Based on [5]
- Outline:
  1. Reduce input problem size
  2. Find particle tracks
  3. Check for possible event vertex



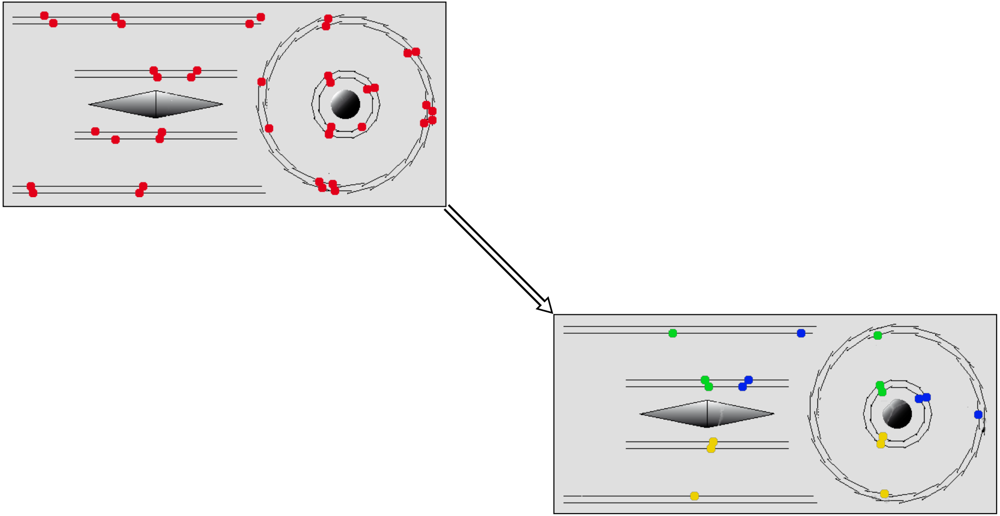
Online Event Selection

- Goals:
  - Identify signal-like frames
  - Reduce data rate by at least a factor of 100
  - Run in real time on GPUs
  - Based on [5]
- Outline:
  1. Reduce input problem size
  2. Find particle tracks
  3. Check for possible event vertex

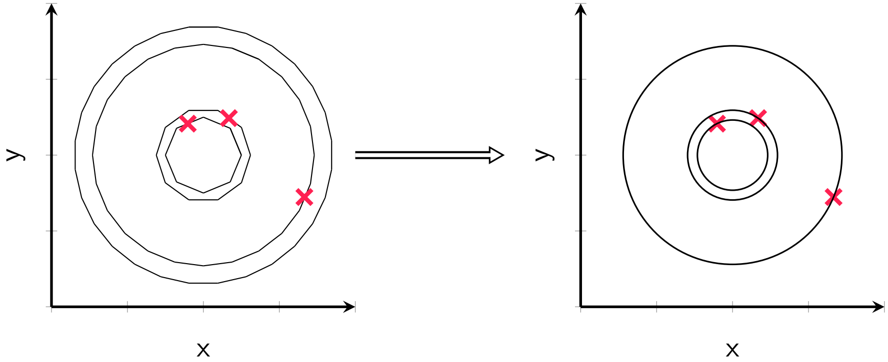
Algorithm 1: Main kernel

```
// Distribute frames over CUDA blocks
for all frame ∈ frames do in parallel
    load_into_shared_memory(frame)
    __syncthreads()
    selection_cuts(frame)
    __syncthreads()
    track_reconstruction(frame)
    __syncthreads()
    vertex_fit(frame)
    __syncthreads()
    store_frame(frame)
end
```

Selection Cuts: Reducing Input Problem Size



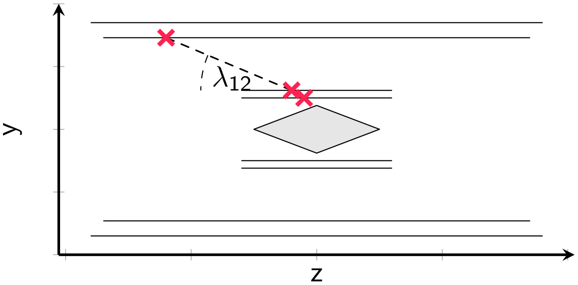
Selection Cuts: Cylindrical "Transformation"



Longitudinal Cut

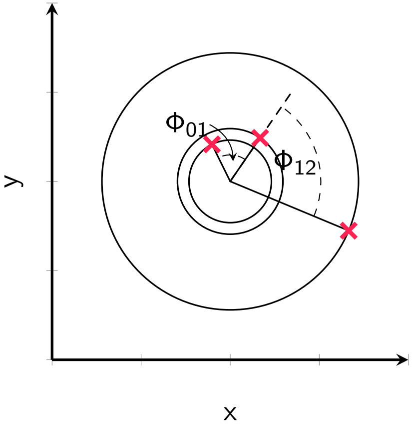
$$\tan \lambda_{i,j} = \frac{z_j - z_i}{r_{t,j} - r_{t,i}}$$

$$\Delta\lambda = \tan \lambda_{12} - \tan \lambda_{01}$$



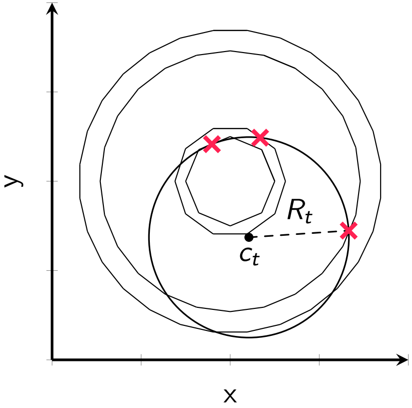
Transversal Cuts

$$\cos\Phi_{ij} = \frac{\mathbf{h}_{t,i}\cdot\mathbf{h}_{t,j}}{r_{t,i}\cdot r_{t,j}}$$

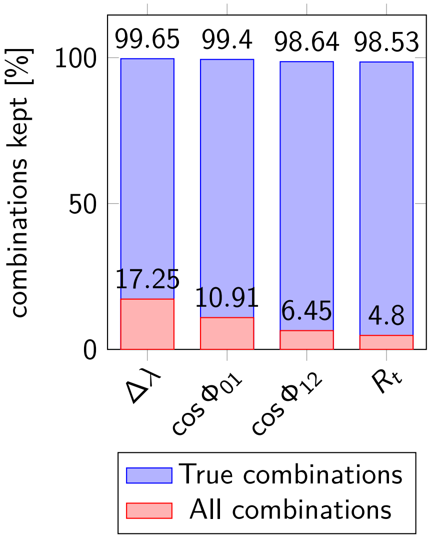


Transverse Cut - the Circle Solution

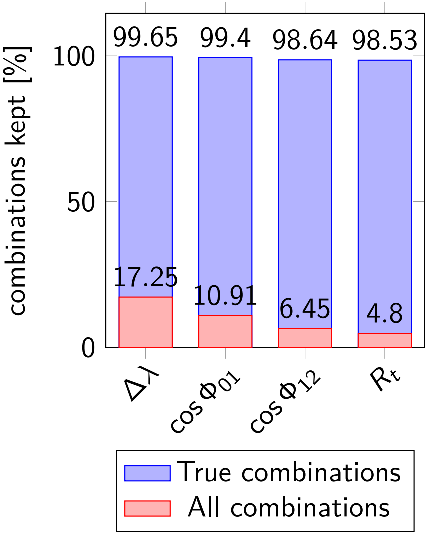
$$R_t = \frac{d_{01}d_{12}d_{20}}{2\left|(\mathbf{h}_0 - \mathbf{h}_1) \times (\mathbf{h}_2 - \mathbf{h}_1)\right|_z},$$
$$\text{with } d_{ij} = |\mathbf{h}_i - \mathbf{h}_j|$$



Selection Cuts: Efficiency



Selection Cuts: Efficiency



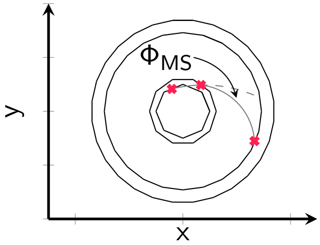
Algorithm 3: Selection Cuts

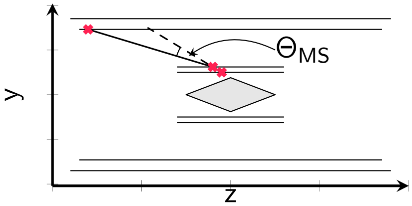
```
// Distribute across threads
for all hit_combinations(frame) do in parallel
    if test_Δλ(hit_0, hit_1, hit_2) then
        if test_Φ_01(hit_0, hit_1) then
            if test_Φ_02(hit_1, hit_2) then
                if test_r_{t,c}(hit_0, hit_1, hit_2) then
                    if num_cuts < cuts_max
                     then
                        save_cut(hit_0, hit_1, hit_2)
                        num_cuts++
end
```

Track Reconstruction: Triplet Fit [6]

- Assumption: Multiple Scattering in pixel layers only source of uncertainty
- Scattering angles have a mean of 0
- Objective:

$$\chi^2(\kappa) = \frac{\Phi_{MS}(\kappa)^2}{\sigma_\Phi^2} + \frac{\Theta_{MS}(\kappa)^2}{\sigma_\Theta^2} \quad (1)$$

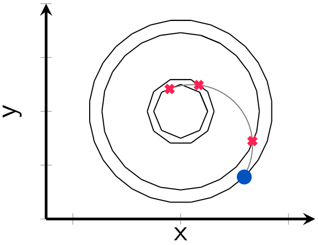




Triplets Fit

- Propagate first helix to fourth layer and use closest hit as extension
- Combining multiple triplet solutions to one solution

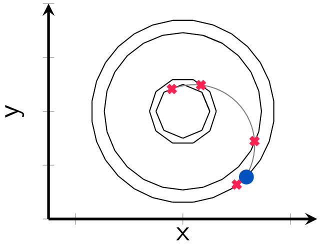
$$\chi^2_{\text{global}}(\kappa) = \sum_{t}^{n_{\text{triplets}}} \chi^2_t(\kappa). \qquad (2)$$



Triplets Fit

- Propagate first helix to fourth layer and use closest hit as extension
- Combining multiple triplet solutions to one solution

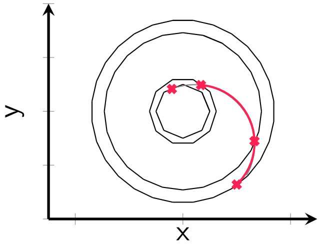
$$\chi^2_{\text{global}}(\kappa) = \sum_{t}^{n_{\text{triplets}}} \chi^2_t(\kappa). \qquad (2)$$



Triplets Fit

- Propagate first helix to fourth layer and use closest hit as extension
- Combining multiple triplet solutions to one solution

$$\chi^2_{\text{global}}(\kappa) = \sum_{t}^{n_{\text{triplets}}} \chi^2_t(\kappa). \qquad (2)$$



Algorithm 4: Track Reconstruction

`// Distribute across threads`

for all $hit_0, hit_1, hit_2 \leftarrow cuts$ **do in parallel**

  $3hit\_track \leftarrow fit\_helix(hit_0, hit_1, hit_2)$

  $hit_3 \leftarrow find\_closest\_layer3\_hit(3hit\_track)$

  $full\_track \leftarrow fit\_helix(3hit\_track, hit_3)$

  if $full\_track.\chi^2 < \chi^2_{max}$ **then**

    if $num\_tracks < max\_tracks$ **then**

      $store\_track(full\_track)$

      $num\_tracks{+}{+}$

    end

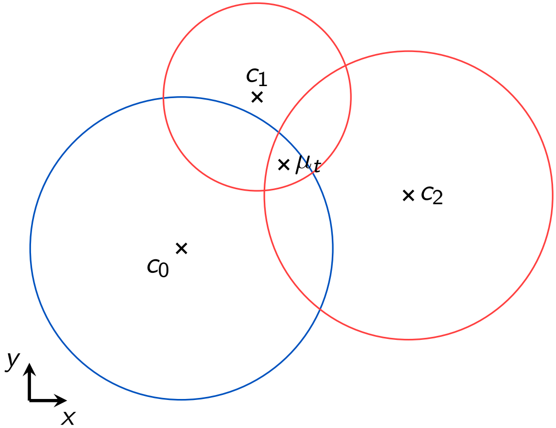
  end

end

Vertex Fit

For all e^+, e^+, e^- combinations:

1. Check for energy conservation
   $E_{e^+} + E_{e^+} + E_{e^-} = E_\mu$
2. Project onto transverse plane
3. Find $2D$ event vertex, choose vertex with smallest error
4. Go back to $3D$ and find $3D$ vertex position
5. Check distance to target
6. Check momentum conservation



Algorithm 5: Vertex Fit: Part 1

```
// Distribute across threads
for all e0+, e1+, e- ← tracks do in parallel
    if test_E_tot(e0+, e1+, e-) then
        store_track_comb(e0+, e1+, e-)
        num_track_combs++
end
__syncthreads()
if num_track_combs > max_tracks then
    return
```

Algorithm 6: Vertex Fit: Part 2

`// Distribute across threads`

for all $e_0^+, e_1^+, e^- \leftarrow$ *track_combs* **do in parallel**
    intersections $\leftarrow$ *get_intersections*(e_0^+, e_1^+, e^-)
    vertex $\leftarrow$ *calc_vertex_estimate*$($*intersections*$)$
    if *vertex_estimate*.$\chi^2 > \chi^2_{max}$ *or keep_frame* **then**
        continue
    end
    if *vertex_estimate.target_dist*$() <$ *target_dist*$_{max}$ *or keep_frame* **then**
        continue
    end
    if *vertex_estimate.total_momentum*$(e_0^+, e_1^+, e^-) >$ *momentum*$_{max}$ *or keep_frame* **then**
        continue
    end
    keep_frame $\leftarrow$ *true*
end

Test Setup

1080Ti:

- Intel(R) Core(TM) i7-5930K
- NVIDIA Geforce GTX 1080Ti
- 16GB RAM

2080Ti:

- AMD Ryzen Threadripper 2970WX
- NVIDIA Geforce RTX 2080Ti
- 128GB RAM

Test data:

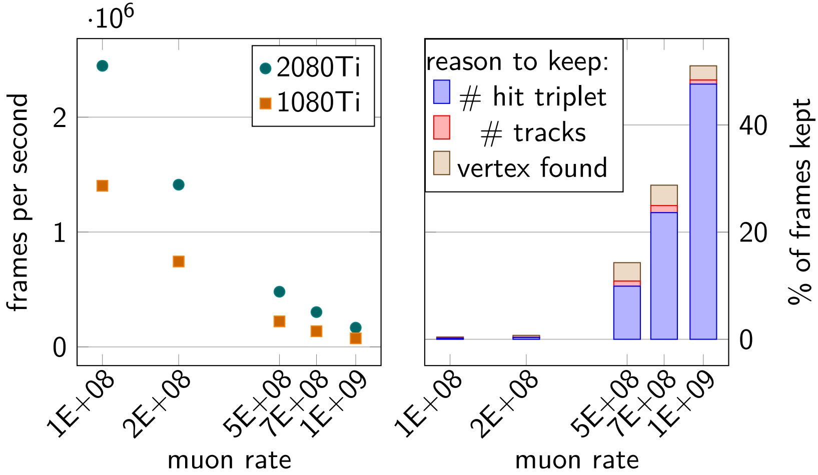
- Mu3e Simulation Framework based on Geant4
- 64ns frame length
- $1.302 \cdot 10^6$ frames per second need to be processed

Algorithm Accuracy

- Muon rate $1 \cdot 10^8 \mu/\mathrm{s}$

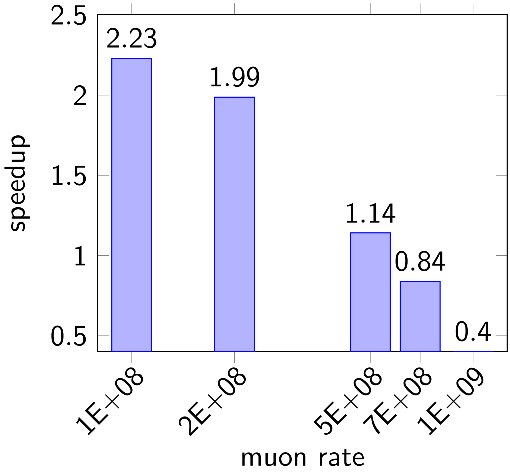
	approx. % kept
signal tracks	97.45
true tracks	94.56
signal frames	94.42

Benchmarks: Muon Rates



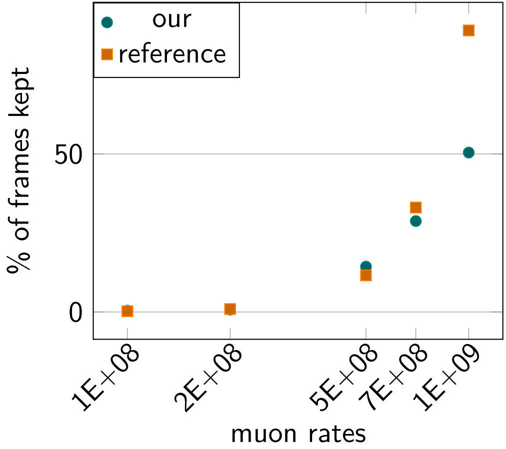
Benchmarks: Speedup

	Our	Reference [5]
Selection Cuts	✓	×
Track Reconstruction	✓	✓
Vertex Fit	✓	✓
Static cut off at 1024 possible triplet combinations	×	✓



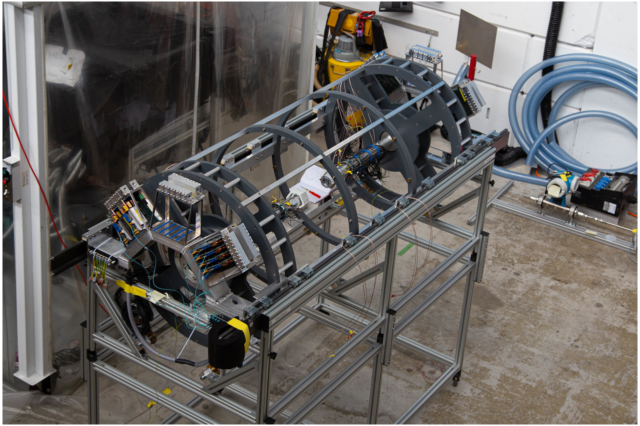
Benchmarks: Speedup

	Our	Reference [5]
Selection Cuts	✓	×
Track Reconstruction	✓	✓
Vertex Fit	✓	✓
Static cut off at 1024 possible triplet combinations	×	✓



Conclusion

- Full implementation of the *Online Event Selection* for Mu3e Phase I
- Performance goals reached, with a spedeup of 2 over previous version.
- Efficiency kept, but could be improved in future work.



[1] Placido Fernandez Declara u. a. “A Parallel-Computing Algorithm for High-Energy Physics Particle Tracking and Decoding Using GPU Architectures”. In: *IEEE Access* 7 (2019), S. 91612–91626. DOI: 10.1109/ACCESS.2019.2927261. arXiv: 2002.11529 [physics.ins-det].

[2] David Rohr. “Usage of GPUs in ALICE Online and Offline processing during LHC Run 3”. In: *EPJ Web of Conferences* 251 (2021), S. 04026. DOI: 10.1051/epjconf/202125104026.

[3] K. Arndt u. a. “Technical design of the phase I Mu3e experiment”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 1014 (Okt. 2021), S. 165679. ISSN: 01689002. DOI: 10.1016/j.nima.2021.165679.

[4] U. Bellgardt u. a. “Search for the decay $\mu^+ \to e^+e^+e^-$”. In: *Nuclear Physics B* 299 (1 März 1988), S. 1–6. ISSN: 05503213. DOI: 10.1016/0550-3213(88)90462-2.

References II

[5] D. vom Bruch. “Pixel Sensor Evaluation and Online Event Selection for the Mu3e Experiment”. Diss. Heidelberg University, 2017.

[6] Niklaus Berger u. a. “A New Three-Dimensional Track Fit with Multiple Scattering”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* (2016). URL: http://arxiv.org/abs/1606.04990%0Ahttp://dx.doi.org/10.1016/j.nima.2016.11.012.

Thank you!