Luminosity Monitor DAQ

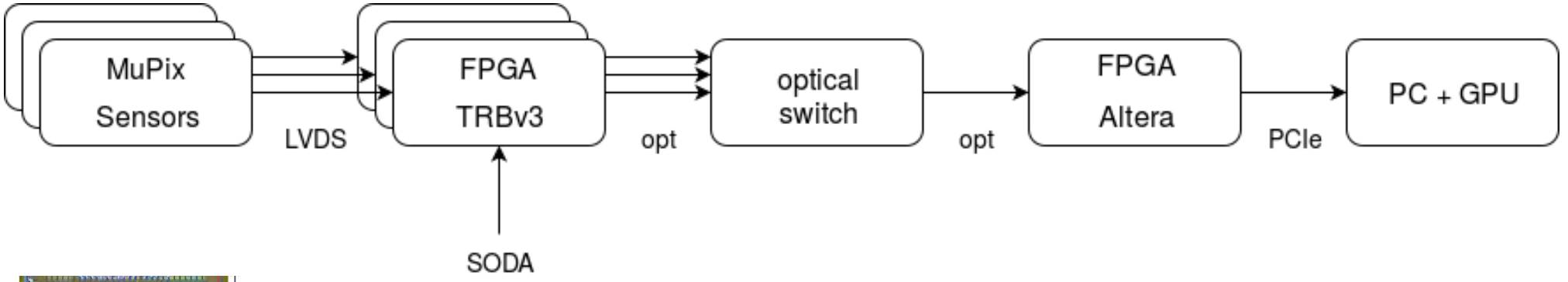
Stephan Maldaner

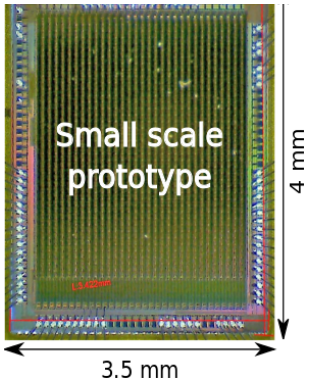
Helmholtz-Institut Mainz
Johannes Gutenberg Universität Mainz

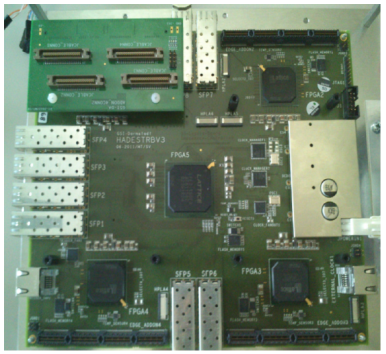
March 6th, 2018
PANDA Collaboration Meeting Darmstadt

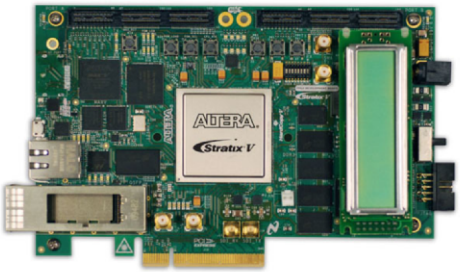
HIM
Helmholtz-Institut Mainz

DAQ





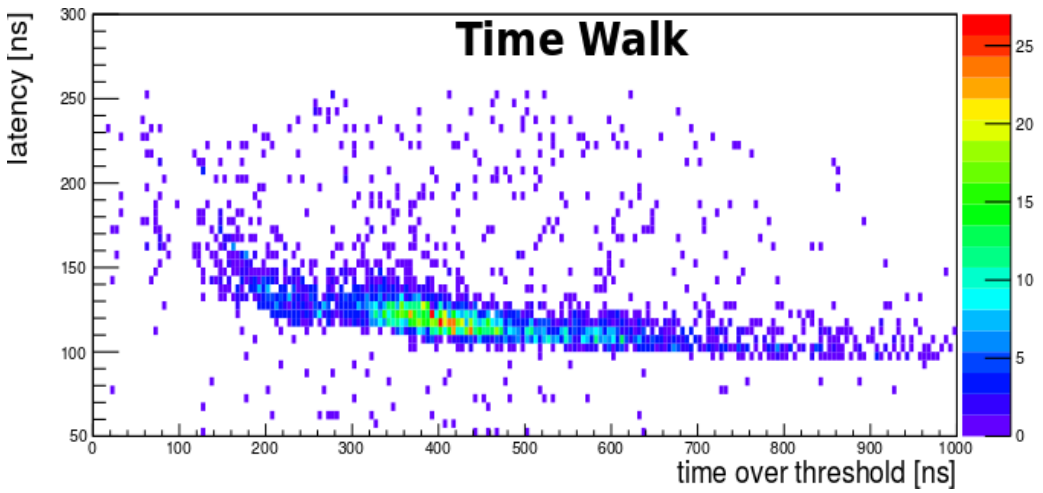




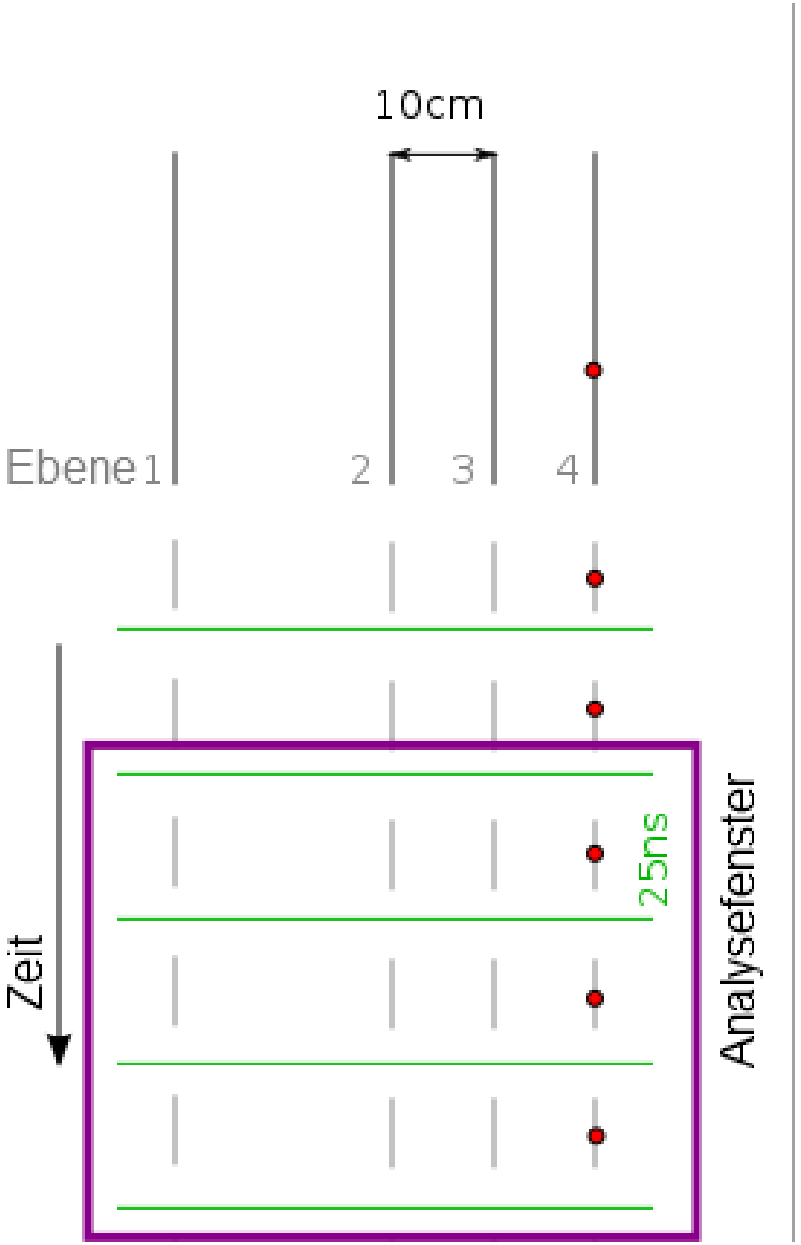
Silicon pixel sensors

1. Data stream from MuPix via LVDS links
2. Slow Control via SPI like bus
3. ~ 20 sensors per TRBv3
4. Merging of TRBv3 data streams
5. Data stream to PC via PCIe
6. Nvidia GTX 980 Ti for tracking (CUDA)

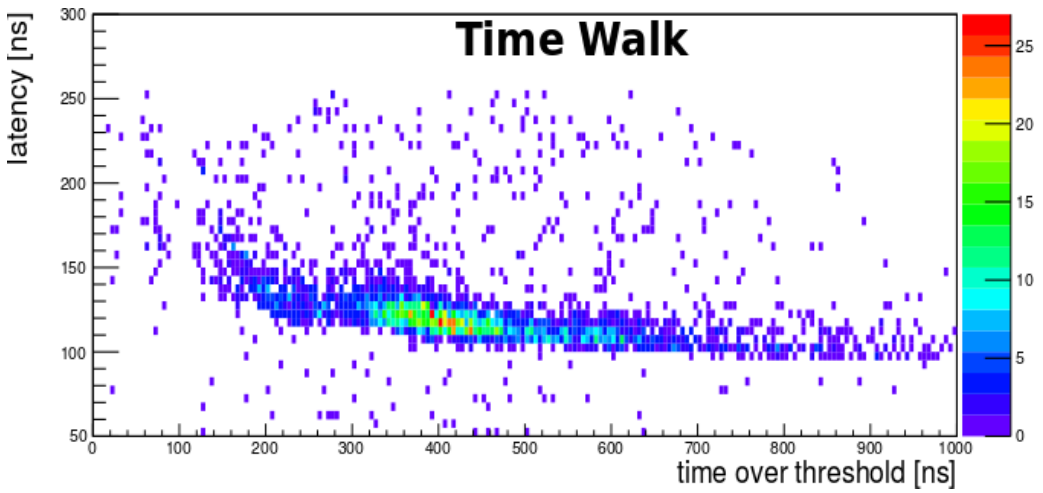
Zeitliche Trefferverteilung

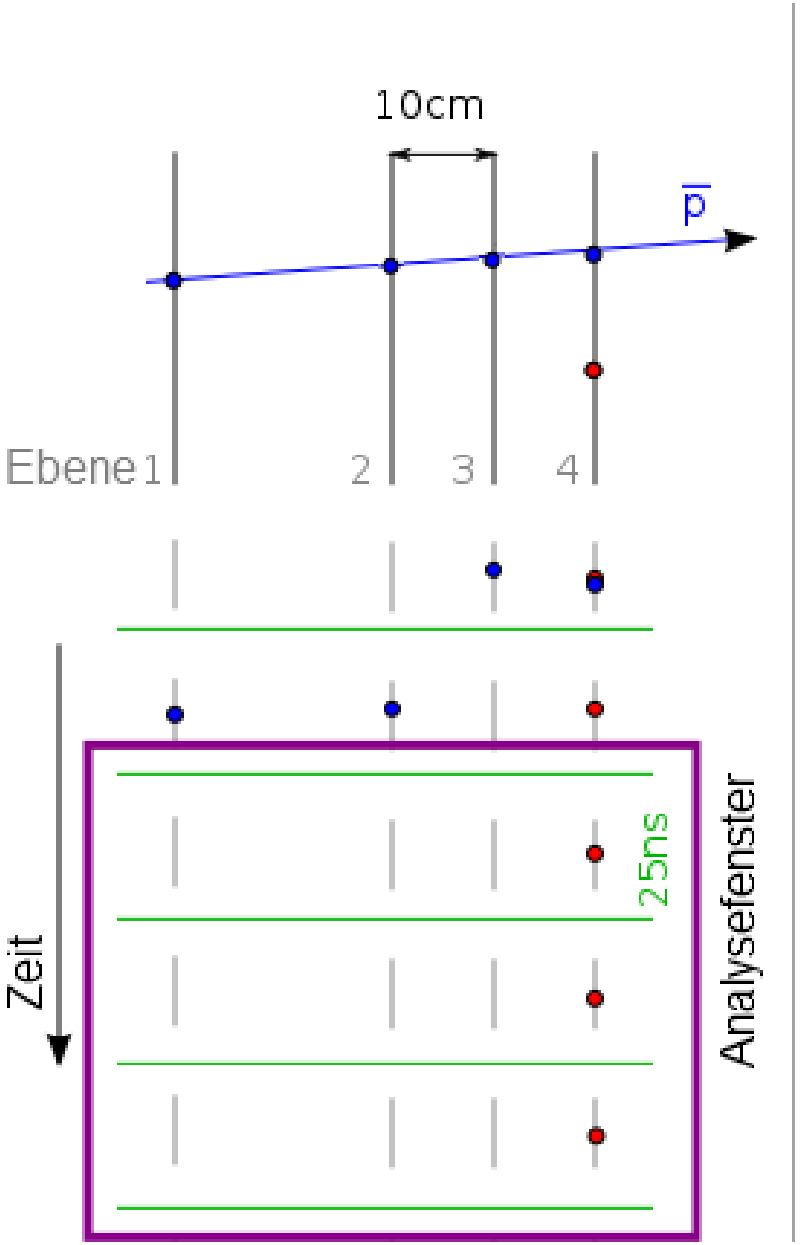


- Different charge deposition in sensors leads to different latencies Δt=75 ns
- Timestemp with 25 ns resolution
  - ➔ Timeframe consists of 3 timestamps
  - ➔ „Moving“ window to get all tracks
- Avoid duplication by requesting hit in first timstamp of timeframe



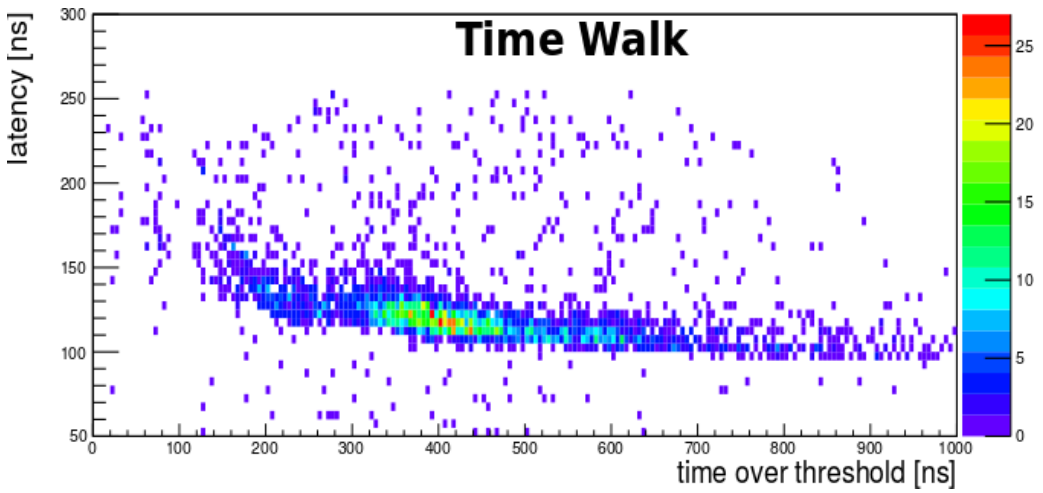
Zeitliche Trefferverteilung



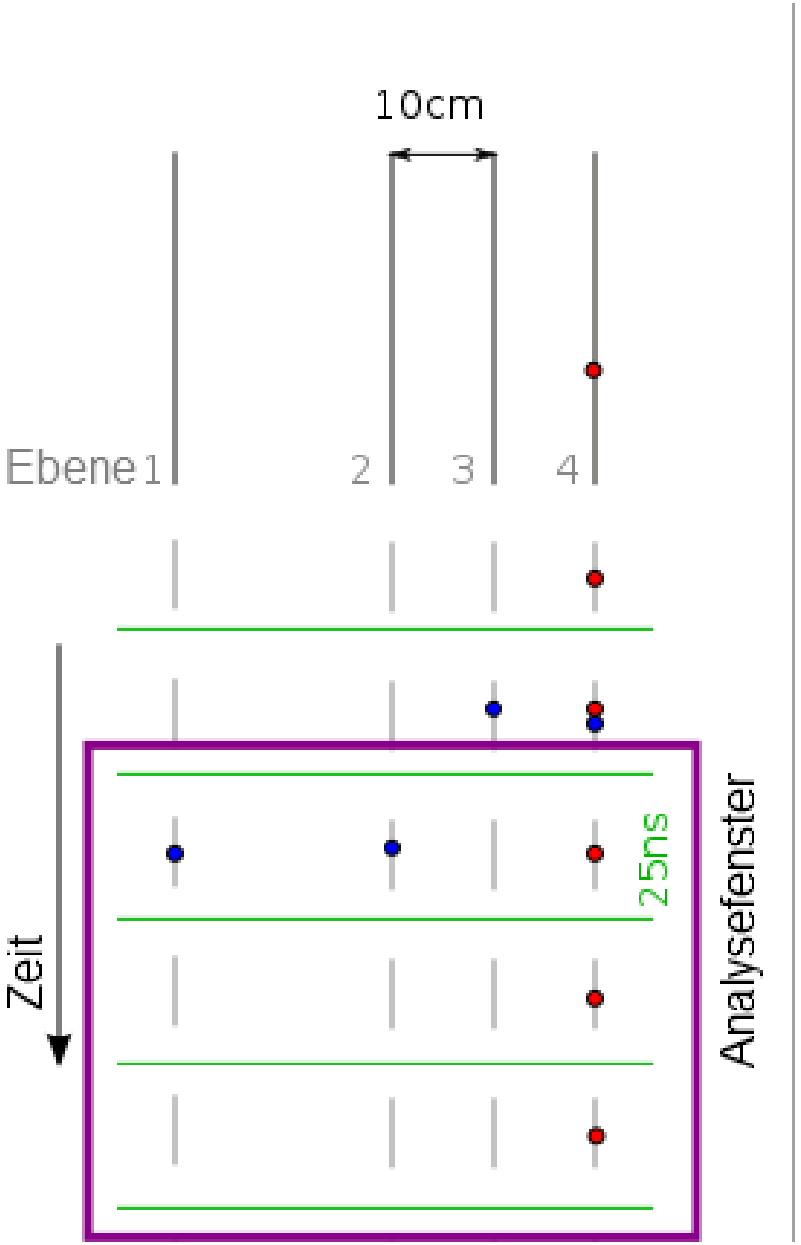


- Different charge deposition in sensors leads to different latencies Δt=75 ns
- Timestemp with 25 ns resolution
  - ➔ Timeframe consists of 3 timestamps
  - ➔ „Moving“ window to get all tracks
- Avoid duplication by requesting hit in first timstamp of timeframe

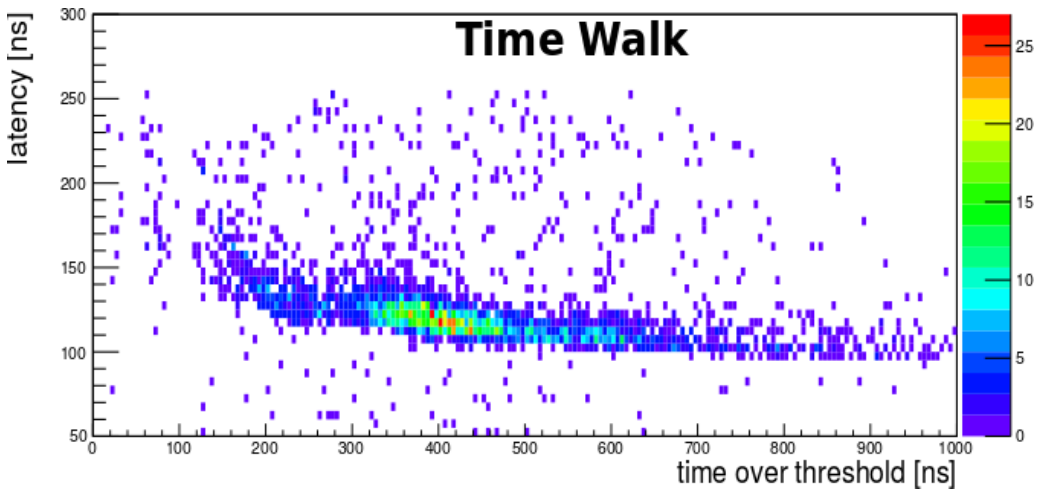
Zeitliche Trefferverteilung



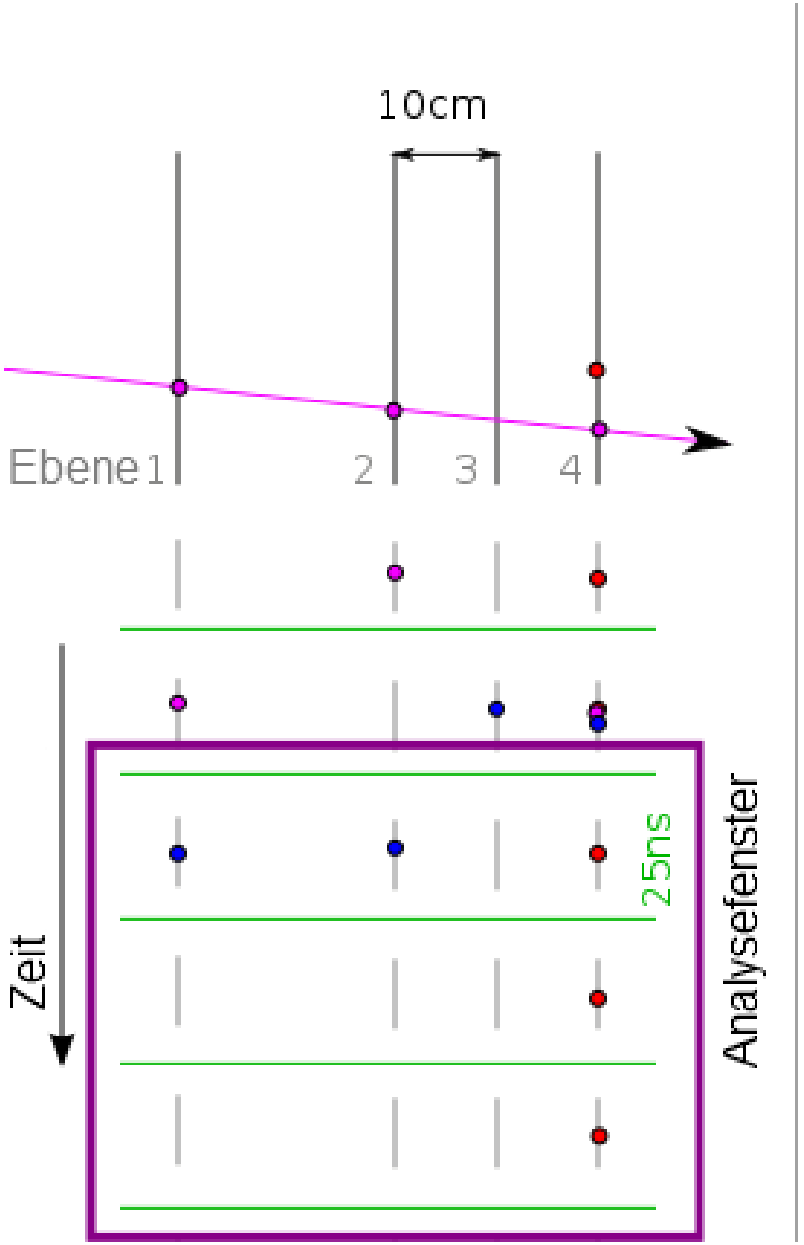
- Different charge deposition in sensors leads to different latencies Δt=75 ns
- Timestemp with 25 ns resolution
  - ➔ Timeframe consists of 3 timestamps
  - ➔ „Moving“ window to get all tracks
- Avoid duplication by requesting hit in first timstamp of timeframe



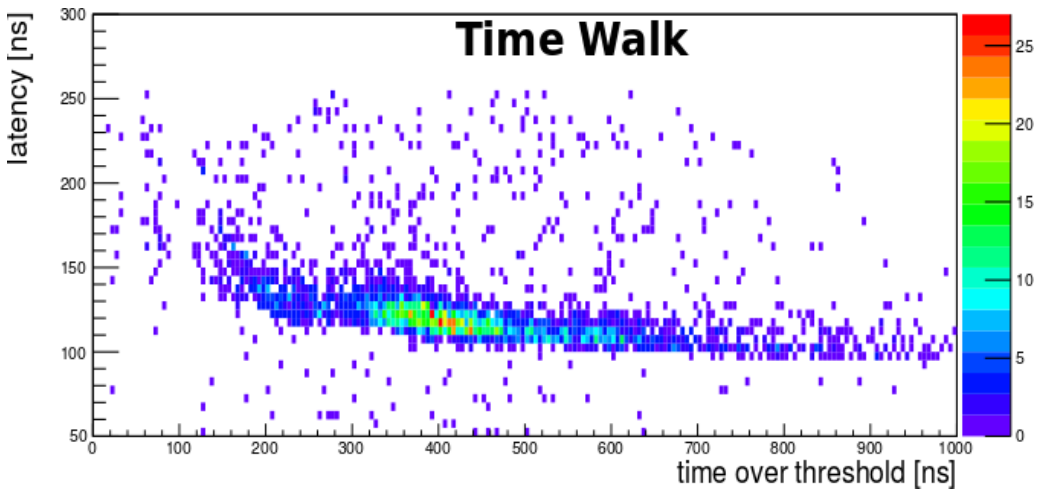
Zeitliche Trefferverteilung



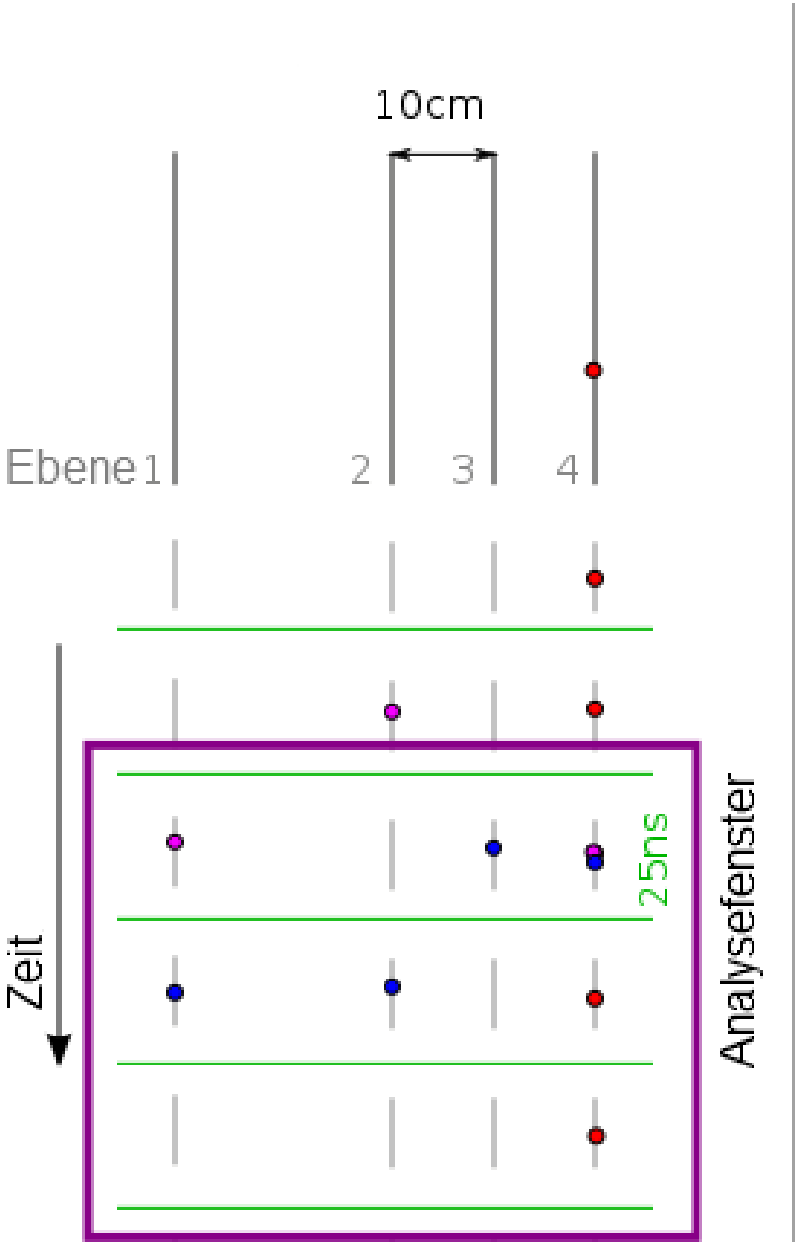
- Different charge deposition in sensors leads to different latencies Δt=75 ns
- Timestemp with 25 ns resolution
  - ➔ Timeframe consists of 3 timestamps
  - ➔ „Moving“ window to get all tracks
- Avoid duplication by requesting hit in first timstamp of timeframe



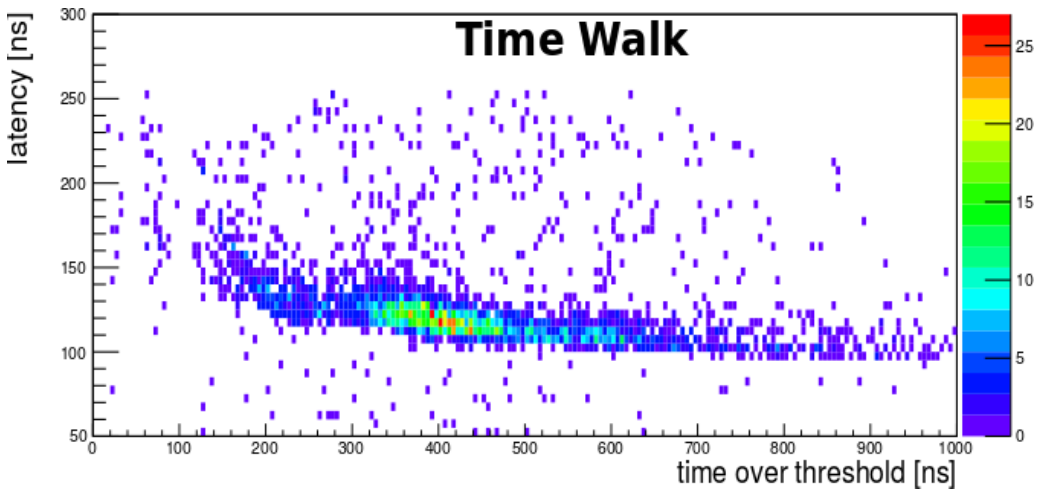
Zeitliche Trefferverteilung



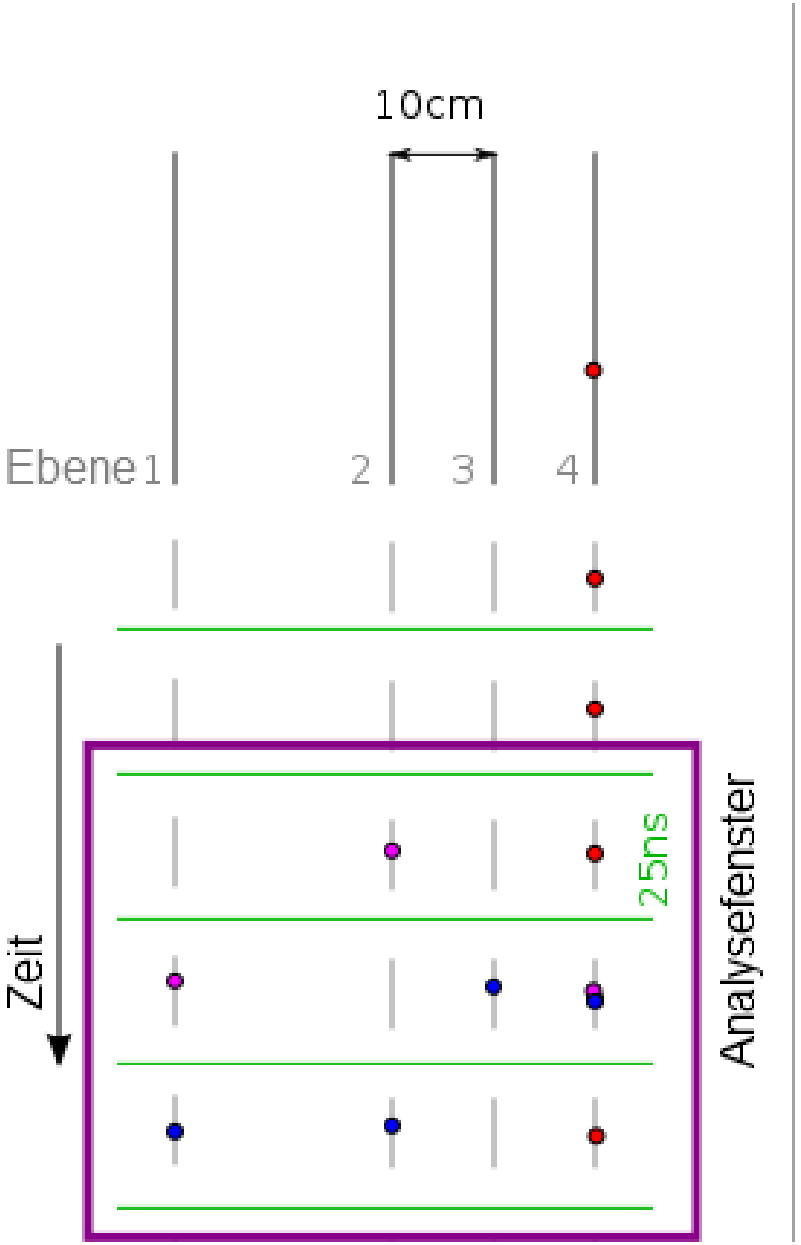
- Different charge deposition in sensors leads to different latencies Δt=75 ns
- Timestemp with 25 ns resolution
  - ➔ Timeframe consists of 3 timestamps
  - ➔ „Moving“ window to get all tracks
- Avoid duplication by requesting hit in first timstamp of timeframe



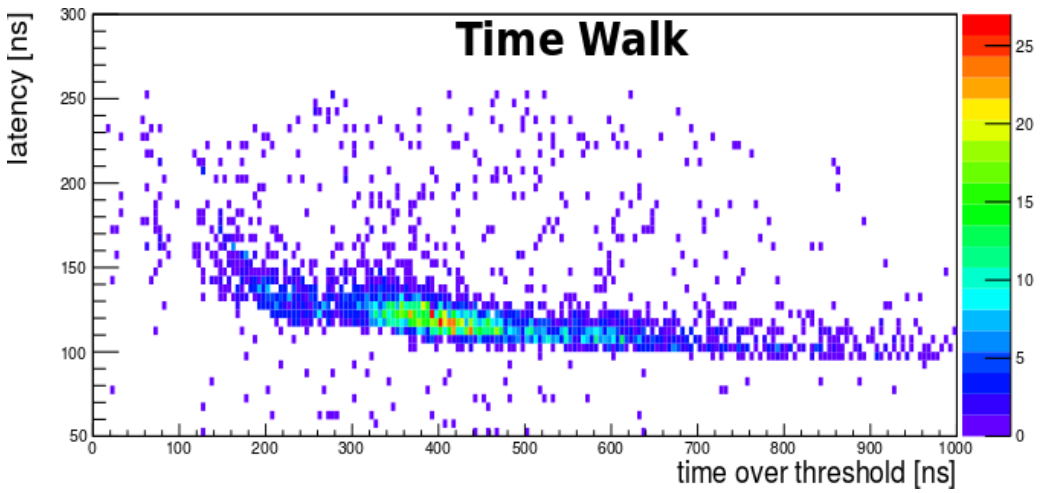
Zeitliche Trefferverteilung



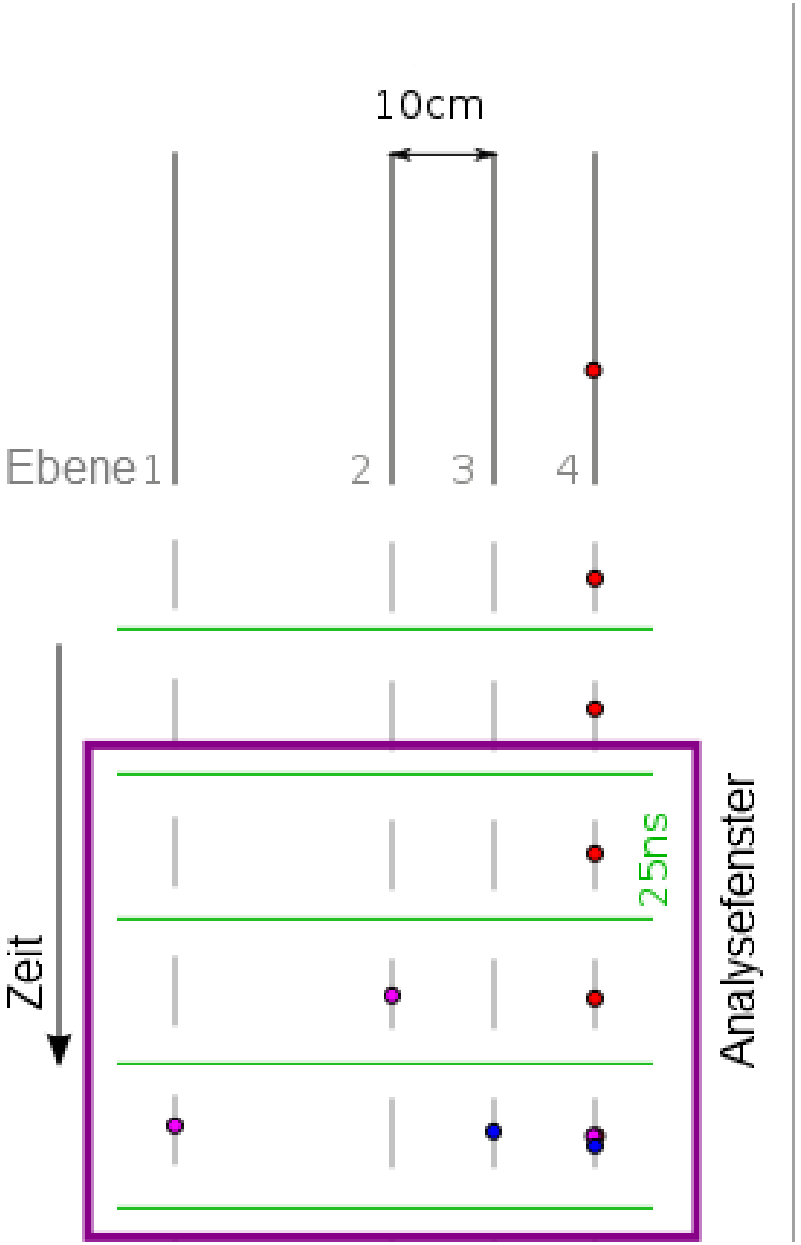
- Different charge deposition in sensors leads to different latencies Δt=75 ns
- Timestemp with 25 ns resolution
  - ➔ Timeframe consists of 3 timestamps
  - ➔ „Moving“ window to get all tracks
- Avoid duplication by requesting hit in first timstamp of timeframe



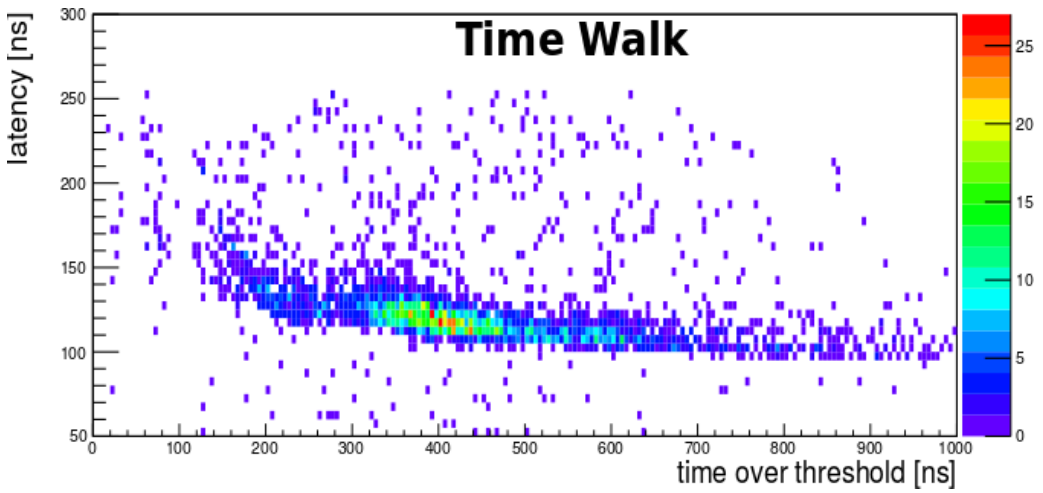
Zeitliche Trefferverteilung



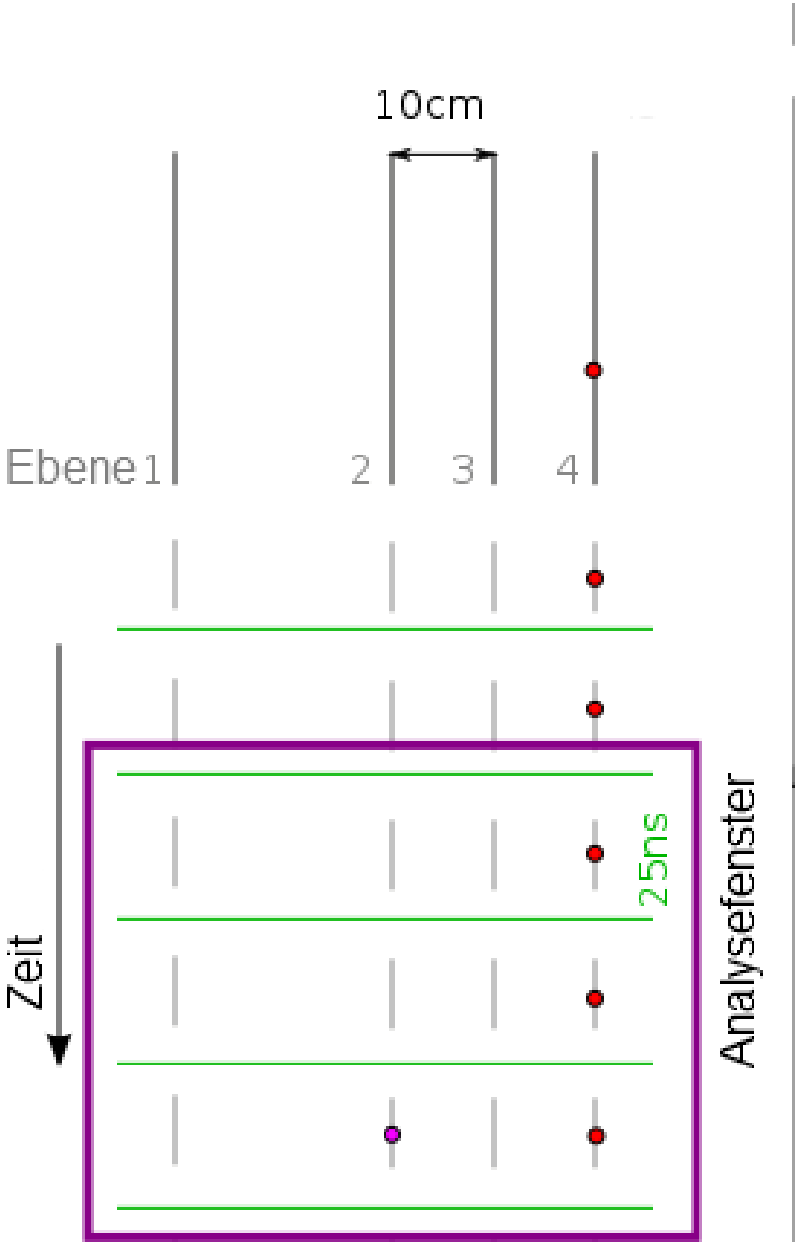
- Different charge deposition in sensors leads to different latencies Δt=75 ns
- Timestemp with 25 ns resolution
  - ➔ Timeframe consists of 3 timestamps
  - ➔ „Moving“ window to get all tracks
- Avoid duplication by requesting hit in first timstamp of timeframe



Zeitliche Trefferverteilung

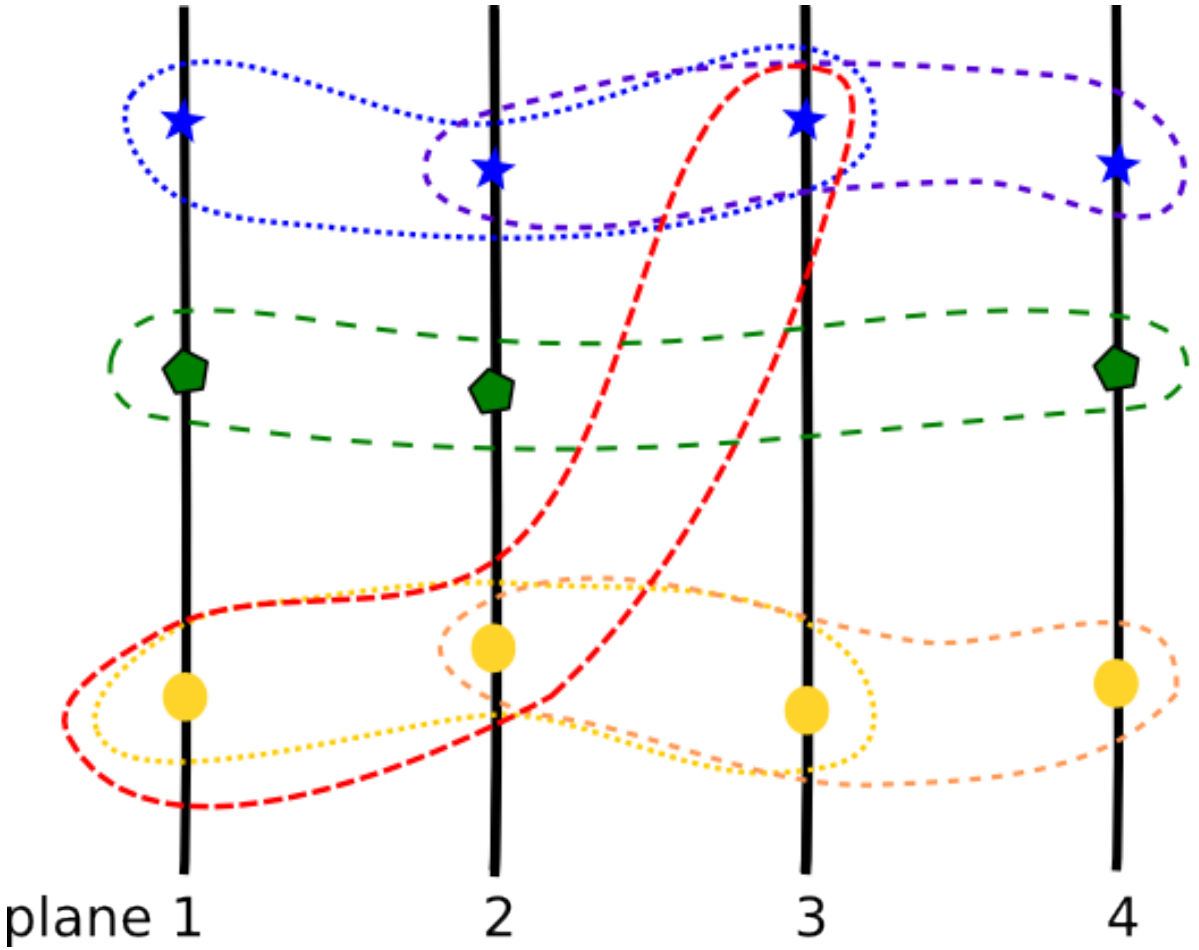


- Different charge deposition in sensors leads to different latencies Δt=75 ns
- Timestemp with 25 ns resolution
  - ➔ Timeframe consists of 3 timestamps
  - ➔ „Moving“ window to get all tracks
- Avoid duplication by requesting hit in first timstamp of timeframe



Triplets

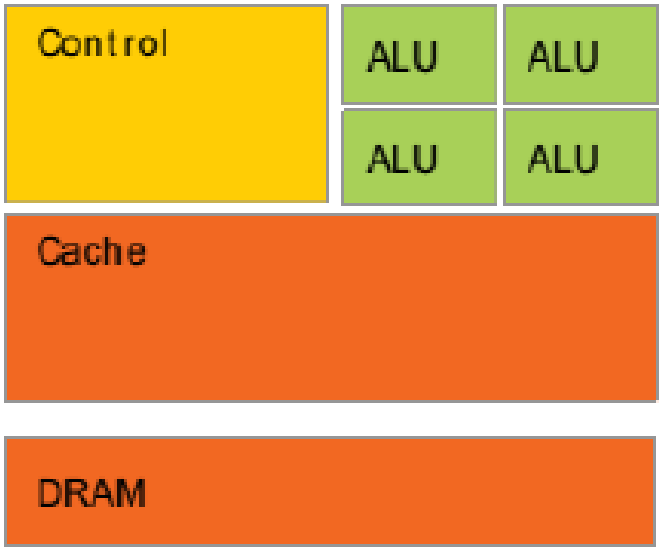
- Three hits form a triplet
- Angle determines correctness of triplets
- Combine triplets to tracks
- One missing hit allowed



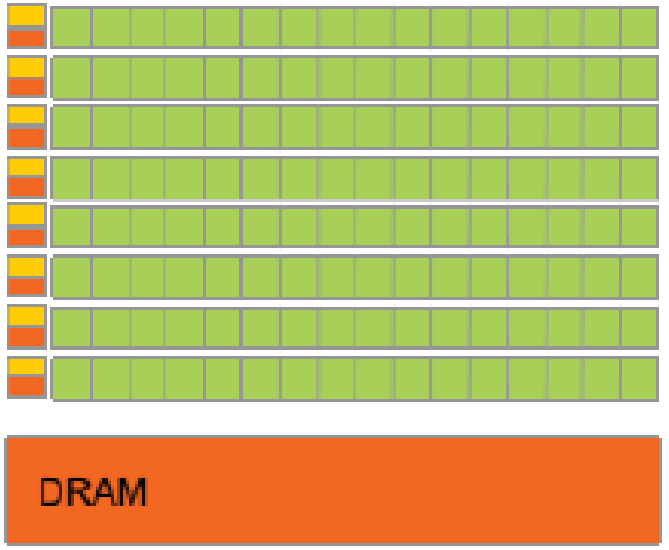
CPU vs GPU

- Complex computing units (ALUs)
- Few ALUs
- Bigger cache
- Smarter cache
- Branch prediction

- Simple ALUs
- Many ALUs
- Smaller Cache
- Simple control logic
- Consumer GPUs
  → Single precision only



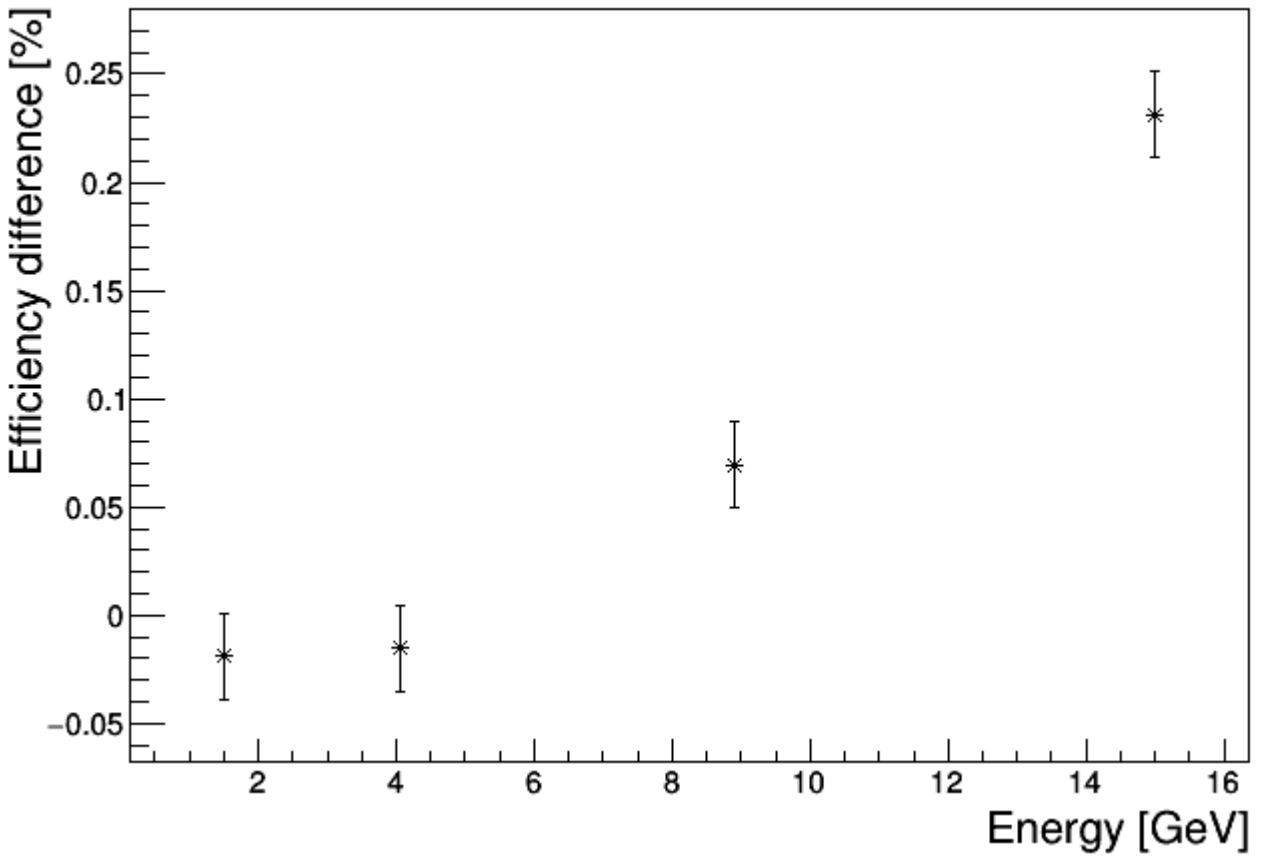
CPU



GPU

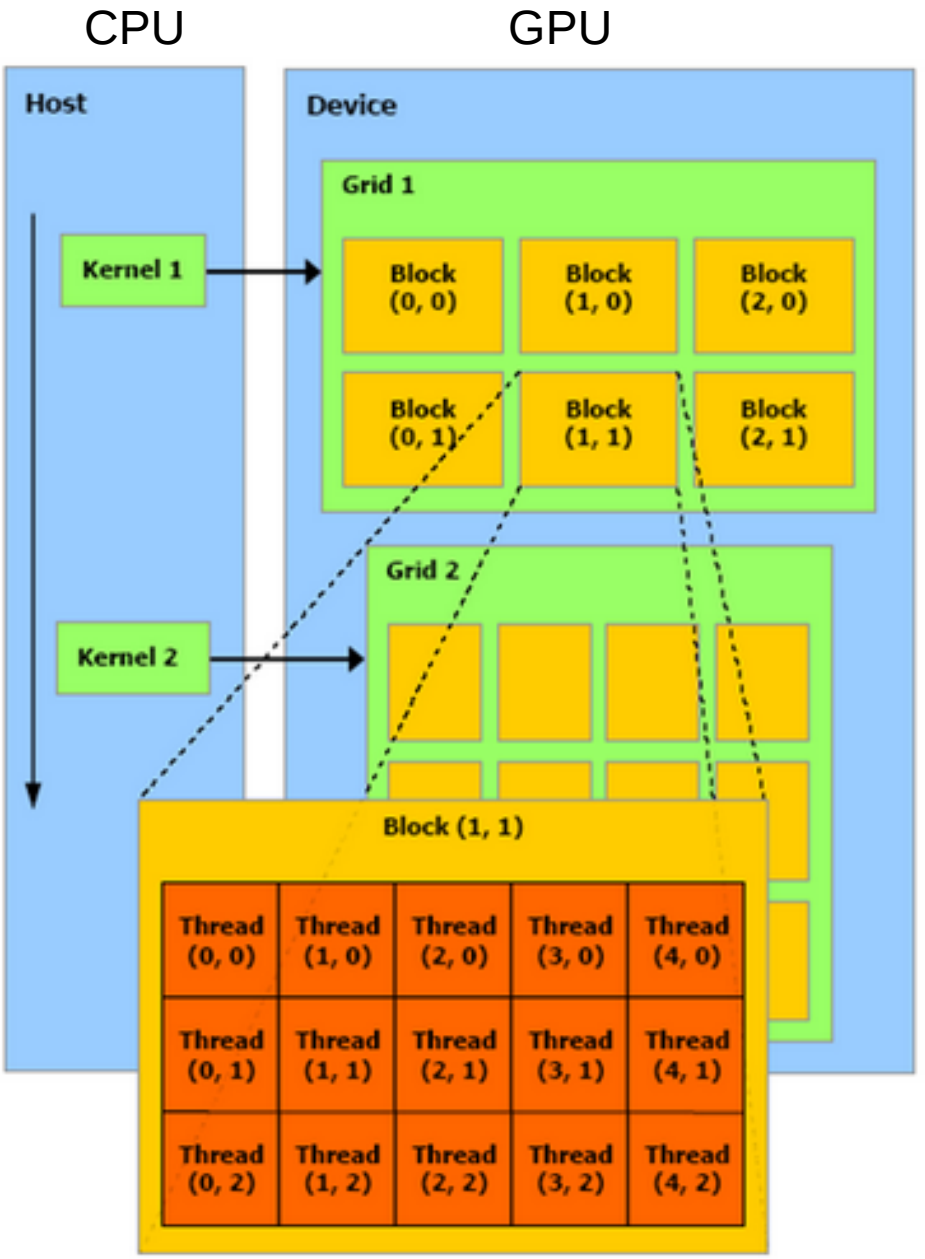
Single vs. double precision

- minimal difference between single and double precision
  ➔ consumer GPUs sufficient

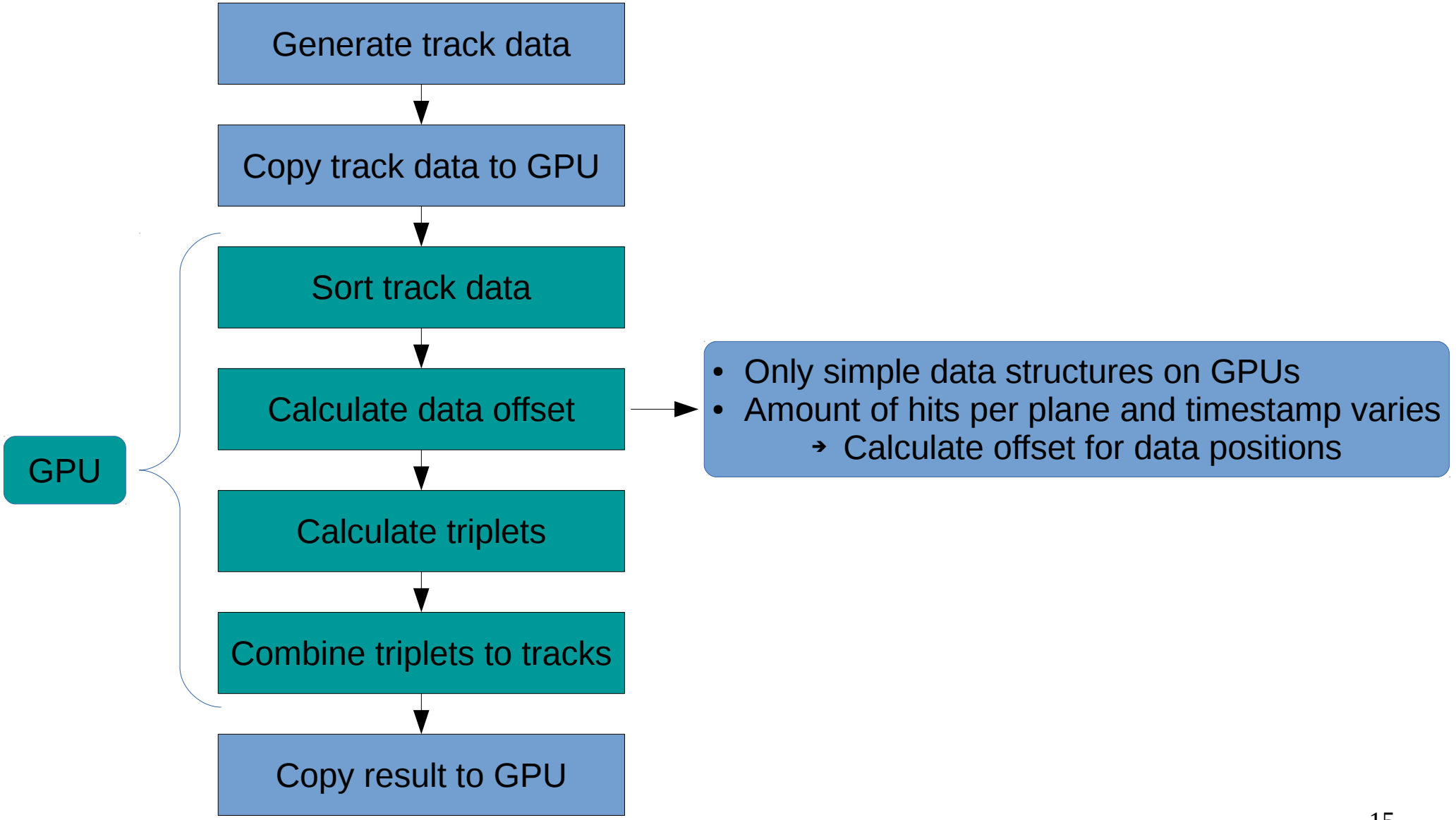


GPU (CUDA) Software Structure

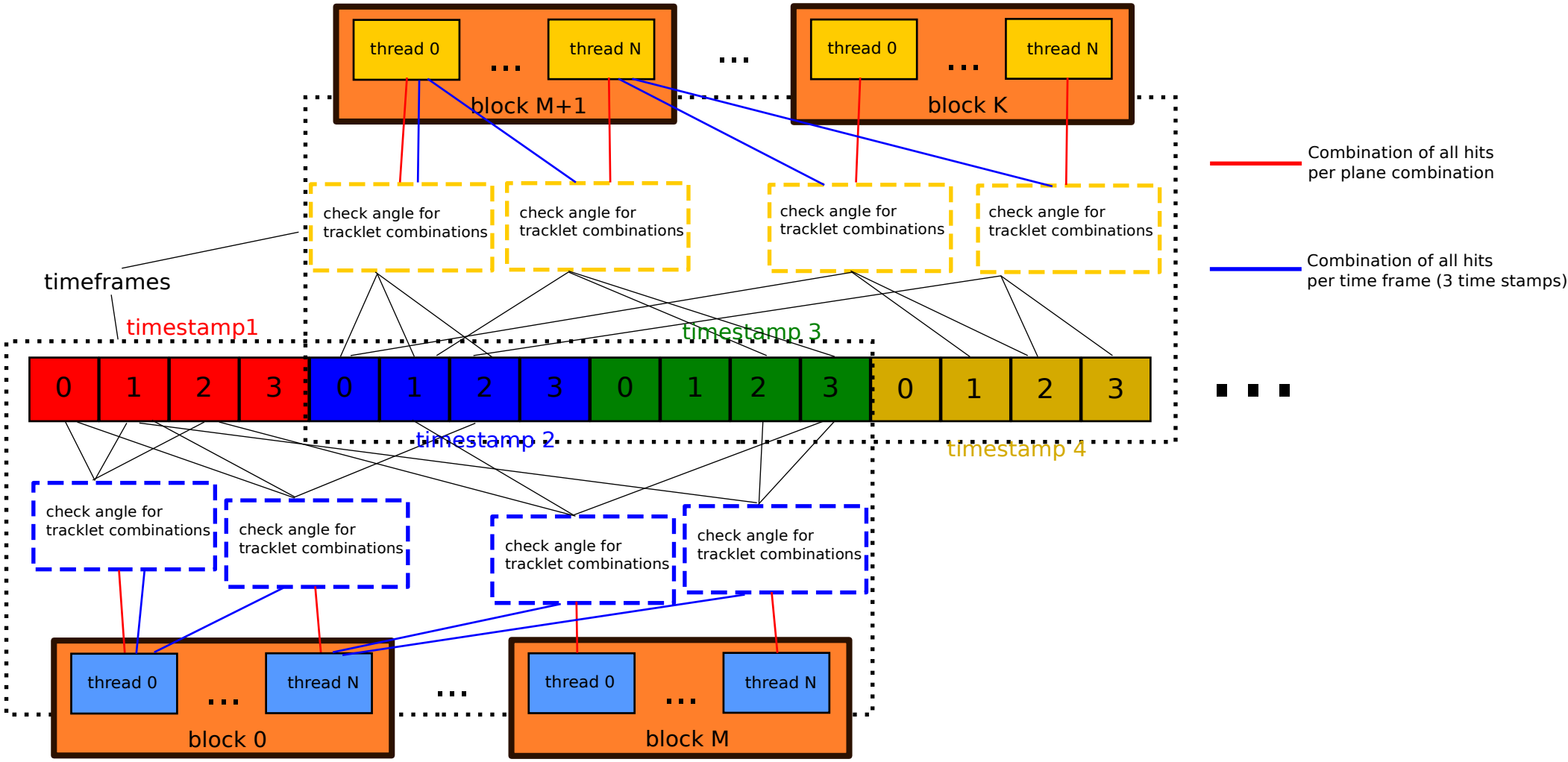
Amount of blocks and
threads in kernel variable
→ Optimize values for problem



Current program steps



Triplet calculation on GPU



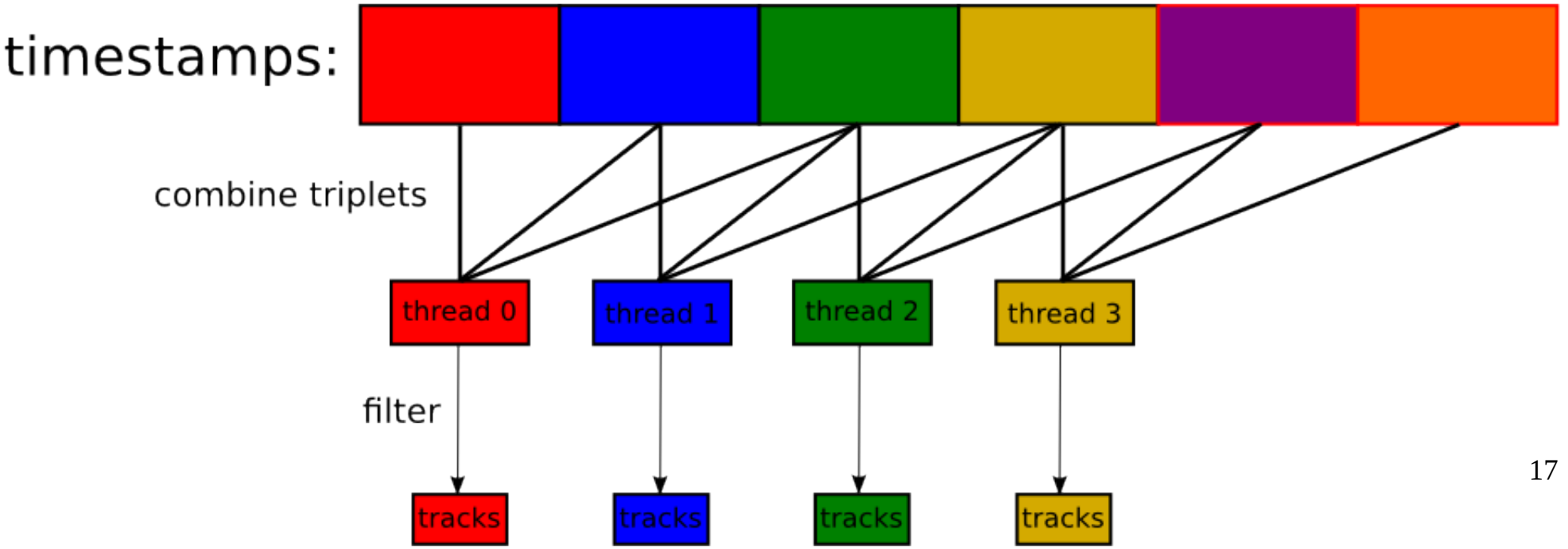
Combine Triplets

- Threads construct tracks belonging to timestamps

  1.Start with triplet containing hit of current timestamp

  2.Match to triplets from same timestamp and 2 right neighbors

  3.Track is assigned according to leftmost timestamp
- Uncombined triplets saved as short tracks to leftmost timestamp
- Last 2 timestamps have to be fed into algorithm again



Timing of steps

- 10000 timeframes
- 1 Track per timestamp
- 512 Threads 100 Blocks (for the last 3 steps)
- Nvidia GTX 980 Ti
- Total average time per track 2,87 µs

copy track data to GPU	sort track data
0,02 µs	0,07 µs

Later on FPGA

Average time per track

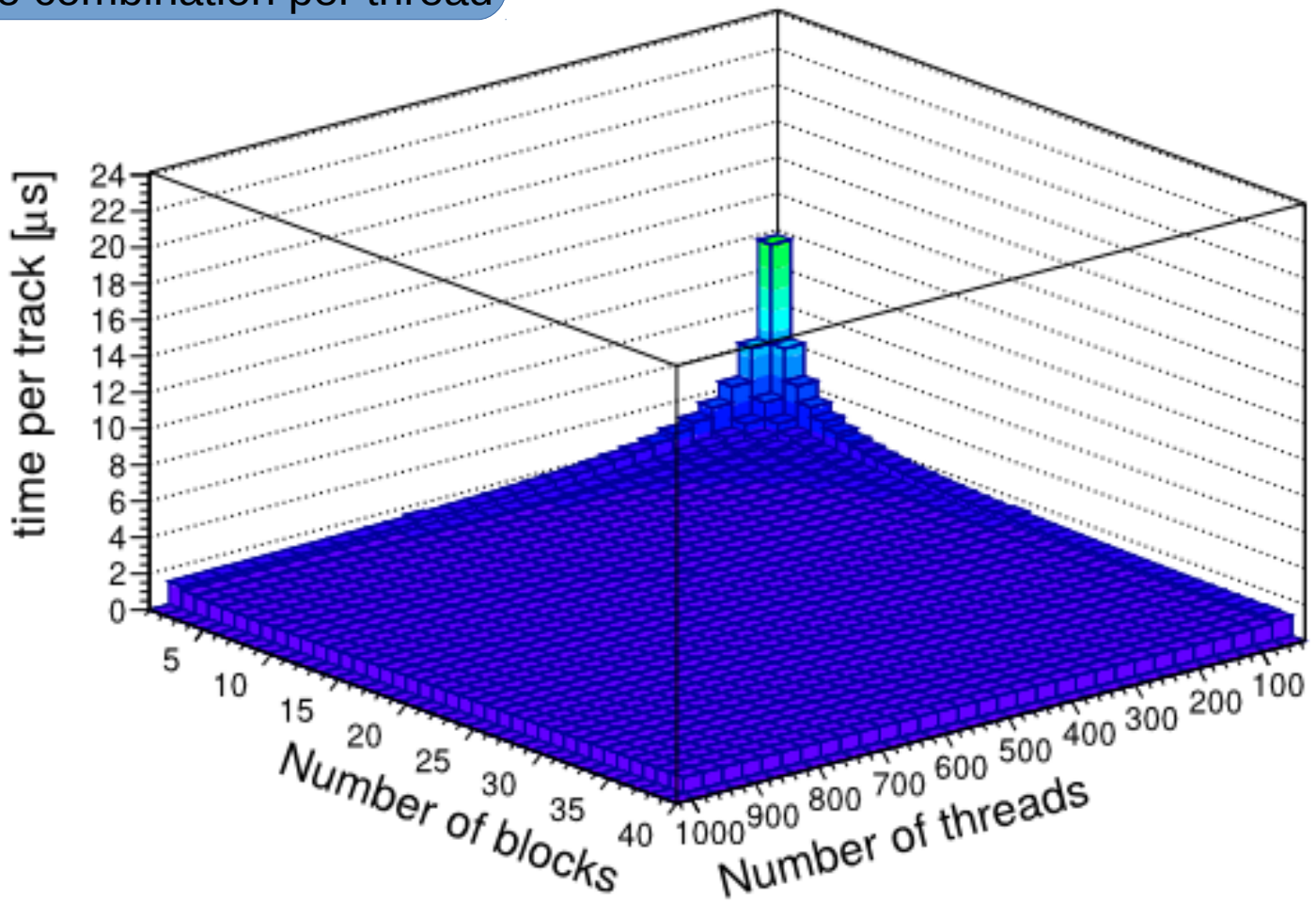
Calculate offset	Calculate triplets	combine triplets to tracks
0,86 µs	0,91 µs	1,01 µs

1 plane combination per thread faster than 1 timeframe combination per thread

- Sequentielle Variante: 22,6 µs
- Intel Xeon CPU E5-1607

Timing dependency on GPU parameters

- 10000 timeframes
- 1 track per timestamp
- 1 plane combination per thread



Future Tasks

- Measure Timing of algorithm for different parameters
- Compare GPU algorithm to parallel CPU code
- Measure delay from SODANET source to FEE
- Integrate altera development board