Data Reconstruction in NA61/SHINE

Bartosz Maksiak

2020-10-30

Introduction

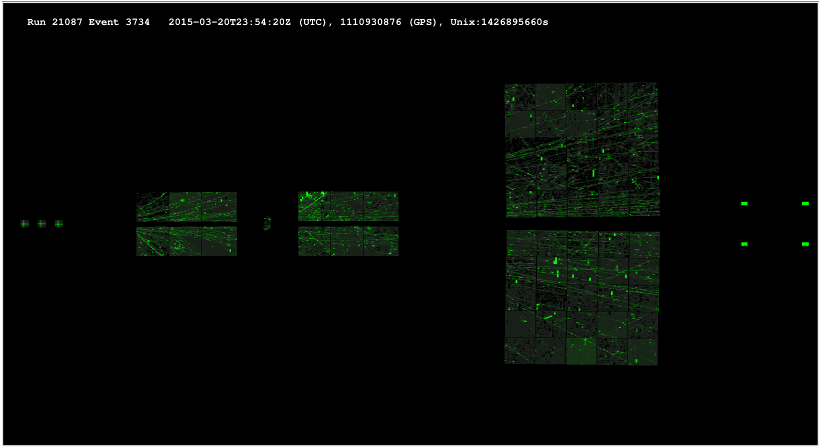
Data reconstruction is the process that converts information about electronic signals from detectors to structures with more physical meaning like clusters, tracks, vertices and events. Those will contain info about:

- Whether the collision (we call it "event") was interesting for us
- What was the precise location of the collision ("vertex")
- How many particles ("tracks") were produced in the collision
- Where are the points ("clusters") left by particles coming through Time Projection Chambers (TPCs)
- What were the parameters of produced particles (momenta, deposited energy)

Or, in more graphical way...

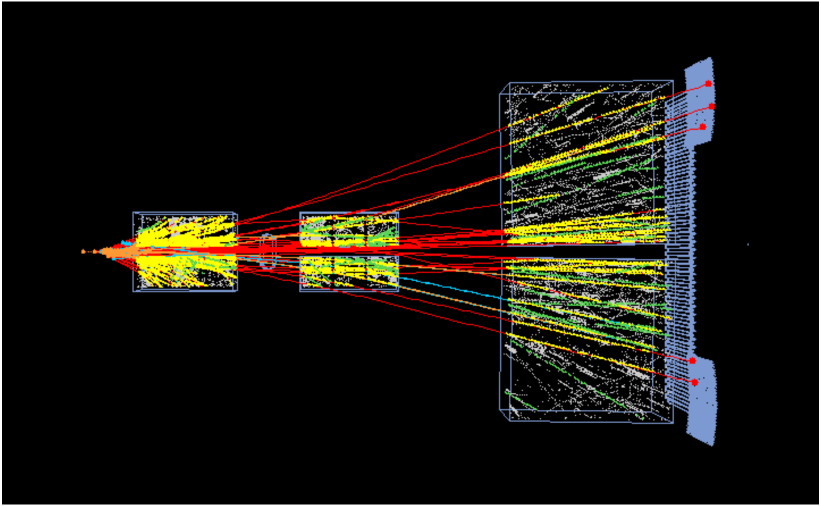
Introduction

… how do we get from this…



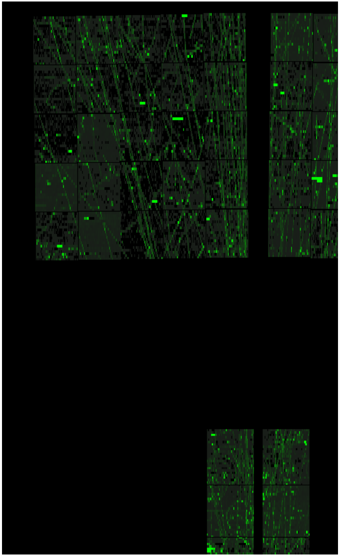
Introduction

... to this



Raw data

- During data taking the data is formed into RAW files.
- In a big simplification: the NA61/SHINE detector works like a photo camera, but instead of CCD pixels, we have sensors (different ones for different parts of the detector: pads for TPC, pixels for ToF, modules for PSD, etc.).
- When a good moment comes (i.e. trigger system gives a "green light" for gathering data), all detector parts will record everything that comes through it and save all signals into the RAW files.
- At this moment we do not know anything about the nature of the collision nor produced particles. There are just values from sensors.



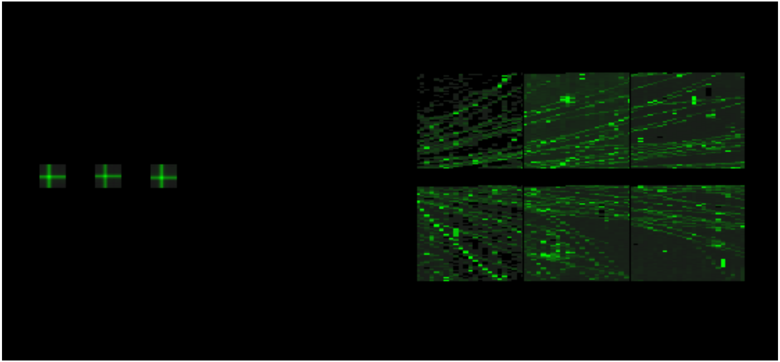
Raw data

- RAW files are sent to CASTOR – CERN data space stored on magnetic tapes
- One event in RAW file takes up to few megabytes. The size depends on the colliding system and beam momentum.
- We store few hundred (200-500) events in one RAW file. Due to historical reasons the size of one RAW file is 1 GB.

Note: In order to get precise and exact information about tracks and events, the detector parts need to undergo the calibration process and store calibration constants in our calibration database. Calibration process will be covered in the next presentation, calibration database will be described more later in this talk.

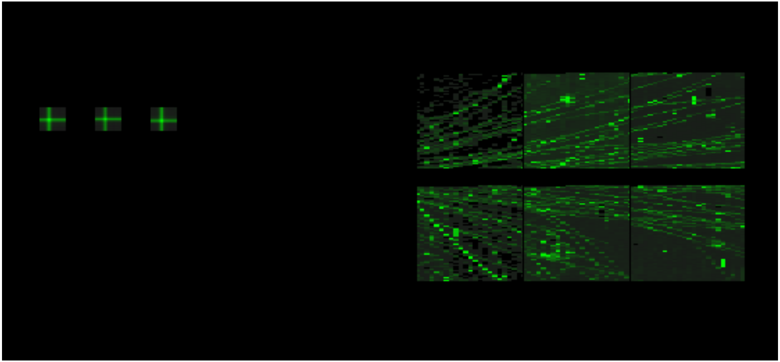
Vertex finding

- For us, a good collision (event, interaction) is such, that one particle from beam hits one particle from the target. This is ensured mostly by properly configured trigger system.
- A precise position of the interaction point has to be found. This is done by using measurements from three Beam Position Detectors (BPD).
- The main interaction point we call Main Vertex. Its position will be needed later.



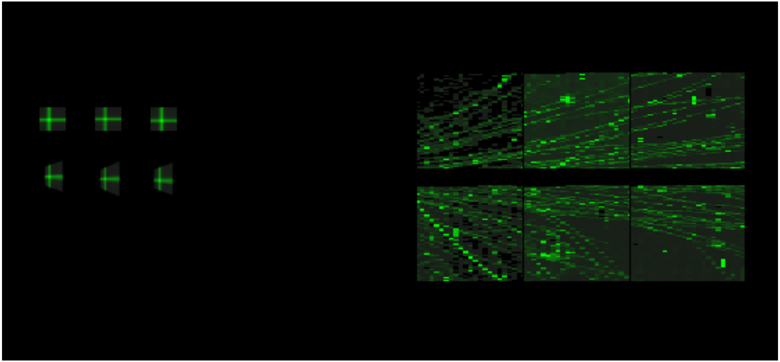
Vertex finding

- For us, a good collision (event, interaction) is such, that one particle from beam hits one particle from the target. This is ensured mostly by properly configured trigger system.
- A precise position of the interaction point has to be found. This is done by using measurements from three Beam Position Detectors (BPD).
- The main interaction point we call Main Vertex. Its position will be needed later.



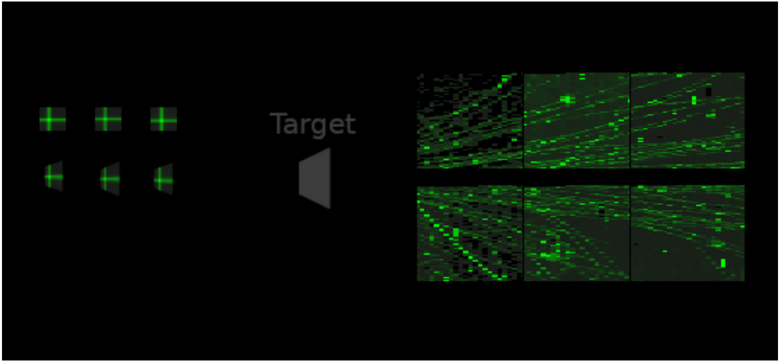
Vertex finding

- For us, a good collision (event, interaction) is such, that one particle from beam hits one particle from the target. This is ensured mostly by properly configured trigger system.
- A precise position of the interaction point has to be found. This is done by using measurements from three Beam Position Detectors (BPD).
- The main interaction point we call Main Vertex. Its position will be needed later.



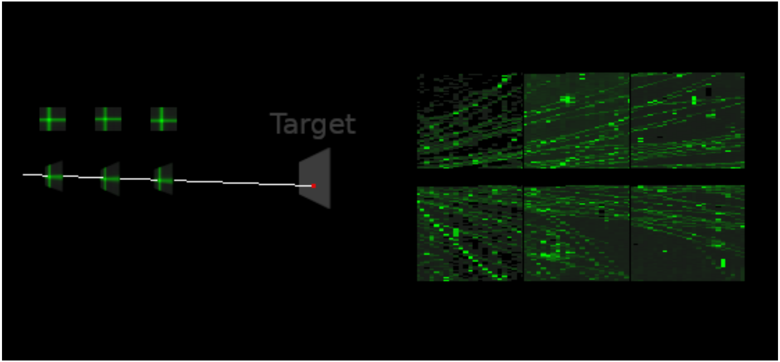
Vertex finding

- For us, a good collision (event, interaction) is such, that one particle from beam hits one particle from the target. This is ensured mostly by properly configured trigger system.
- A precise position of the interaction point has to be found. This is done by using measurements from three Beam Position Detectors (BPD).
- The main interaction point we call Main Vertex. Its position will be needed later.



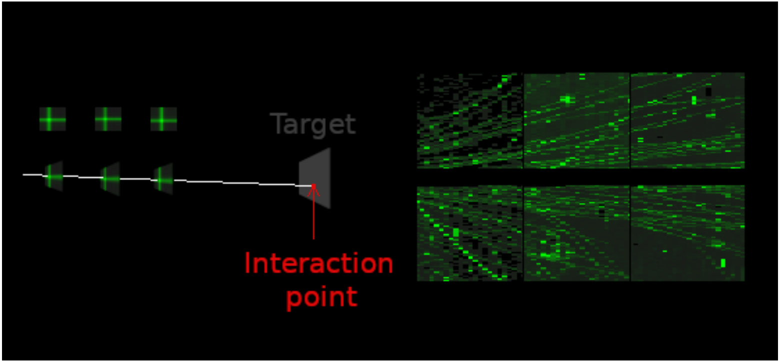
Vertex finding

- For us, a good collision (event, interaction) is such, that one particle from beam hits one particle from the target. This is ensured mostly by properly configured trigger system.
- A precise position of the interaction point has to be found. This is done by using measurements from three Beam Position Detectors (BPD).
- The main interaction point we call Main Vertex. Its position will be needed later.



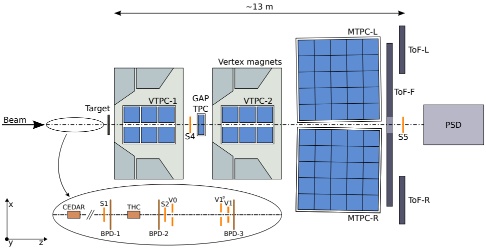
Vertex finding

- For us, a good collision (event, interaction) is such, that one particle from beam hits one particle from the target. This is ensured mostly by properly configured trigger system.
- A precise position of the interaction point has to be found. This is done by using measurements from three Beam Position Detectors (BPD).
- The main interaction point we call Main Vertex. Its position will be needed later.



Formation of clusters

1. The particles produced in the collision go through Time Projection Chambers (TPCs). They ionize particles of gas which fills the TPCs.
2. Ionized particles are drifting (following the lines of electric drift field) and land on the readout electronics (on pads) and leave signals there.
3. Since we know precise position of a given pad, we can learn the XZ position of the particle.
4. Because ionized particles drift along Y axis, we get additional information on Y position based on time of particle drift.
5. Now we know the 3D point in space where the particle passed!



Problems of tracking

- Now, since we have geometrical points within TPCs, we may start the tracking procedure, which in a big simplification is just a connect-the-dots puzzle.
- However, there are some problems:
  - We register in TPCs only charged particles. Neutral ones are invisible, unless they decay into charged particles (so-called V0).
  - The "dots" have to be connected throughout up to four TPCs and there are gaps between them, so some assumptions (extrapolations) to their paths have to be made.
  - VTPC1 and VTPC2 chambers are inside magnetic field, so charged particles will be bent by magnetic field, so their tracks will be curved.
- All these problems are handled by reconstruction algorithms and several stages of the tracking.

Tracking process

1. Firstly, local tracking is done: within each TPC the cluster points are connected to form lines or curves.
2. Then, global tracking goes:
   2.1 A line (local track fragment) from MTPC is found and it is extrapolated to VTPC2.
   2.2 In VTPC2 the algorithm searches for lines/curves that end close to where the extrapolation from MTPC landed. Longer candidate track is formed.
   2.3 Situation repeats: Extrapolation from VTPC2 goes to GTPC or VTPC1 where algorithm searches for good candidate track fragments to join the long track – a global track.
   2.4 If the global track ends nearby the vertex, it will be considered as primary track (i.e. coming from the main collision).
   If the global track ends somewhere far from vertex, it will be considered as secondary track (e.g. being a product of a decay of a V0 particle).

Fitting to vertex

- If a global track ends nearby main vertex, it becomes a primary track.
- Primary tracks are being fitted to the main vertex.
- After fit, the tracks are getting their "official" momentum. The momentum vector that is obtained during that fit is the momentum that will be used later when analyzing data.

- Other global tracks considered as secondary tracks (products of particle decays) will be later "attached" to the different, secondary vertices.

Other reconstruction activities

Tracking is the most essential part of the reconstruction, but there are additional processes that have to be done in order to have fully reconstructed data. Below are the examples:

- **Finding V0 vertices** – the tracks that start not from main vertex will be considered as products of V0 decay, a special algorithm tries to connect those tracks and determine from decay of what neutral particle they were created.
- **Calculation of potential points** – this process determines how many pads should get the signal from passing particle assuming that we already know the particle momentum. This is a good help in finding artifacts of reconstruction that we call "split tracks" and "merged tracks".

Final file

- When reconstruction finishes, we get a file in the SHOE format (SHOE stands for SHine Offline Event)
- The SHOE format is based on STL structures, but may be browsed using ROOT. The file contains information on:
  - Beam position.
  - Vertex parameters: e.g. their position, their type (primary vertex, secondary vertex, etc.), the tracks that are coming from given vertex.
  - Track parameters: e.g. momentum, positions of clusters they left and their ionization energy (essential for more accurate identification).
  - Information from reconstruction of other parts of detector: ToF, PSD, etc.
  - Raw data if needed.
- SHOE file is a file ready for further processing like calibration or analysis.
- Reconstructed SHOE files are sent to CERN storage spaces: EOS (disk space) and CASTOR (magnetic tapes).

Calibration database

The reconstruction may use only data from RAW file, but then it will be inaccurate. It needs additional information on precise measurements such as conditions of the detector in a moment of recording an event like atmospheric conditions of the air and gas in TPCs (temperature, pressure, amount of water, hydrogen, oxygen, etc.)

Also, the reconstruction is not done just once. Single reconstruction would give inaccurate results. In order to get results closer to the true ones, the detector parts need to undergo the calibration process. Variables that need to be precisely calibrated are for example:

- Precise position (geometry) of BPDs and TPCs
- Drift velocity inside TPCs
- Starting moment of collecting data in a given TPC (T0)
- Measurements of particle energy loss ($\mathrm{d}E/\mathrm{d}x$)

All those values are stored in the calibration database (based on structuralized XML files) and being updated during calibration process.

Thank you