Participant document: Particle Tracking and the TrackML Challenge

v1.0

April 30, 2018

Abstract

This document supplements the information of the Kaggle web site[1]. It is meant to give enough details for a good participation. No background in physics is required.

1 Introduction

1.1 Tracking particles

In the LHC, proton bunches (beams) circulates and collide at high energy. Each collision produces a firework of new *particles* To identify the types and measure the kinematic properties of these particles, a complex apparatus, the *detector* bends them in an externally applied magnetic field and records the small energy deposited by the particles when they impact well-defined locations in the detector.

The tracking problem refers to reconstructing the trajectories from the information recorded by the detector. Given the coordinates of the impact of particles on the detector (3D points), the problem is to "connect the dots" or rather the points, i.e. return all sets of points belonging to alleged particles trajectories.

1.2 The challenge dataset

The data created for the challenge are representative of the real HEP experimental scenarios. The dataset have been simulated with the realistic simulator ACTS[1] anticipating the new LHC architecture to be deployed by 2025. The task has been made almost as difficult as for real events.

1.3 Methods

The methods provided in the kernels submitted by the organisers (`DBScan` and `Hough`), as those used in HEP, are fully unsupervised (clustering-like). As the ground truth is provided, the participant can use any kind of **supervised or semi-supervised** method; of course, unsupervised ones are perfectly legitimate.

[1]https://www.kaggle.com/c/trackml-particle-identification

1.4 Organization of the document

The rest of this paper is organized as follows. Section 2 presents the essentials: the challenge task and metrics. Section 3 elaborates on the detector and measurement process. Reading it is not necessary to get started with the challenge, although its material might be very useful to get a good ranking. For the interested reader, a companion document available on the web site dwells on the physics of the simulation. Section 4 is devoted to the complete description of the data set and format.

Useful links:

- the Accuracy phase competition site on Kaggle[2], with most up to date info in the Discussion, and useful Kernels
- the overall TrackML challenge site [3], with a brief description of the challenge Throughput phase with a big incentive on the speed
- the helper library : [4]
- follow us on twitter : [5]

2 Overview and vocabulary

2.1 The task

An event is a set of particle measurements in the detector. The collisions are not recorded, only the particles they generate. From an abstract point of view, the detector is simply an apparatus that records the impacts, called the hits, of the particles traversing the detector in an event. The detector is formed of discrete layers.

The basic setting is as follows.

- Each particle is created close to, but not exactly, at the center of the detector (details in section 3.2).
- Each hit is a 3D measurement in euclidean coordinates (x, y, z). For each particle, the number of hits is on average 12, but as low as 1 and up to 20.
- The participants should associate the hits created by each particle together, to form tracks. Typically, at least 90% of the true tracks should be recovered.
- The tracks are slightly distorted arc of helices with axes parallel to the z-axis, and pointing approximately to the interaction center (see figure 1).

In an ideal world:

[2]https://www.kaggle.com/c/trackml-particle-identification
[3]https://sites.google.com/site/trackmlparticle/home
[4]https://github.com/LAL/trackml-library
[5]https://twitter.com/trackmllhc

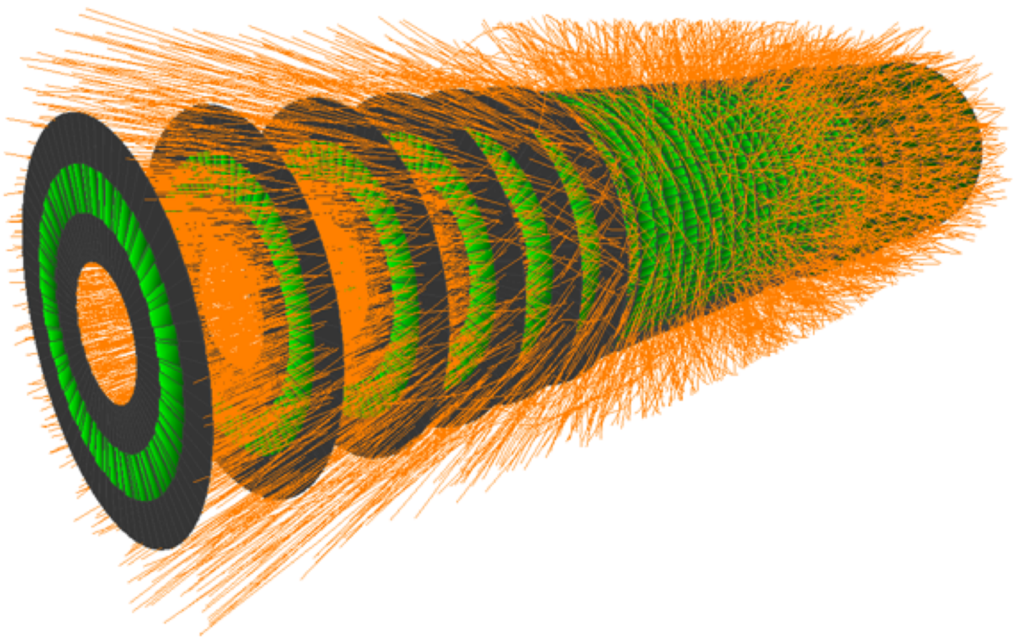


Figure 1: Detector and particle trajectories.

(a) each particle would leave one and only one hit on each layer of the detector;

(b) the trajectories would be exact arcs of helices;

(c) the (x, y, z) coordinates would be exact.

In this ideal world, the `Hough` method suffices to solve the problem. However, the devil (and the challenge) is in the details.

(a) Each particle may leave multiple hits in a layer, but may not record anything as well. However, the multiplicity is constrained by the organization of the detector (section 3.3).

(b) The arcs are often slightly distorted (section 3.4).

(c) The measurements have some non isotropic error (section 3.5).

The gist of the challenge is to show robustness with respect to all these perturbations. It is enforced by the metric defined by the score.

2.2 The score

Participants' submission are judged on a single score: it is the intersection between the reconstructed tracks and the ground truth particles, normalized to one for each event, and averaged on the events of the test set. It is implemented in the helper library[6].

By convention, we note tracks the proposed solutions and particles the ground truth. A perfect algorithm will uniquely and correctly associate each point to the particles it belongs to. An imperfect algorithm will miss some particles altogether, miss one or more points for an otherwise valid track, associate wrong points to an otherwise valid track, find tracks from random association of points, find multiple variants of the same track.

Because the data come from simulation, we know which particle created each hit (point), in other words the ground truth. A hit must belong to at most one track. All hits should be assigned to a track (it is adviseable to have a catch-all track for all initially unassigned hits). The score is defined as follows.

- tracks with 3 points or less have a zero score, as they do not allow to compute any meaningful physical quantity in further analysis.
- tracks are uniquely matched to particles by the following rules:
  - For each track, the matching particle is the one to which the absolute majority of the tracks points belong; if there is no such particle (in particular if there is a 50/50 tie), the score for this track is zero.
  - The track should have the absolute majority of the points of the matching particle, otherwise the score of this track is zero

[6]https://github.com/LAL/trackml-library

These two requirements guaranty a one to one relation M between all remaining tracks and particles.

- The score of a track r is the sum of the weights of the hits in the intersection $r \cap M(r)$.
- The score of the event is the sum of the scores of the tracks. The total sum of the weights of all points being 1, this actually normalizes the score to the $[0, 1]$ range, with 1 being the score of a perfect reconstruction.

The overall ranking score is the average over the validation events. Participants are required to build submission on the 125 event validation sample, which are split internally by Kaggle to yield the public leaderboard score (as it appears online) and the private leaderboard score (which will be revealed at the end of the competition). We have evaluated the statistical uncertainty to be a few 10^{-3}.

The weights are incentives to get physically meaningful reconstructions, along two directions: the weight of a point is the product of two independent quantities *weight_order* and *weight_pt*.

- *weight_order* The points at the beginning of the track, close to the collision, and at the end, are more important than the ones in the middle. The weights reflect this hierarchy and are normalized to sum to 1.
- *weight_pt* The high energy particles (large transverse momentum p_T) are the most interesting ones. As the bulk of the tracks have low p_T, we have to explicitly favor high p_T. *weight_pt* is 0.2 if $p_T < 0.5$GeV and 1. for $p_T > 3$GeV, with a linear interpolation in between. Note that the lower the p_T, the larger the geometrical curvature; at large p_T tracks appear as straight lines.
- Particles which generates 3 hits (points) or less are considered spurious, the weights of the associated points are set to zero. There are also random hits, which weight is also set to zero. The assignment of random hits to different tracks do not change the overall score, unless the matching particles lose the majority count.

As the weights disclose important information, they are provided along with the ground truth in the training data, but are kept hidden for the validation data.

3 How the detector works

3.1 What you should know

This section is intended to walk the reader through some intricacies of the particle production and hits measurement process. In summary, we explain where hits can happen, and how the measured trajectory of a particle may (and generally will) depart from an exact arc of helix.

- Section 3.2: the origin of particles is close to, but not exactly at coordinate $(0, 0, 0)$.

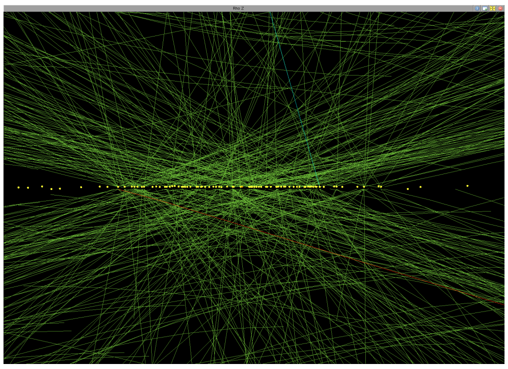


Figure 2: Pile-up: an event with 78 collisions

- Section 3.3: how the detector organization constraints (or not) the measurement.
- Section 3.4: the physical (true) trajectory is approximately an helix, with parameters determined by the particle momentum, and how it is perturbed.
- Section 3.5: the measurement errors, ie (x, y, z) is only an approximation of the real hit coordinates.

3.2 The collisions

An event is the set of particle measurements in the detector associated to single beam-beam crossing. Multiple (in average 200, with Poisson distribution) proton-proton collisions (*pile-up*) occur during a single beam-beam crossing (figure 2). In this context, some particles are grouped in dense "jets", increasing the possibility of confusion, and some particles stop early.

The individual collisions are randomly placed in small region, the so-called *luminous region* around the nominal center position of the detector $(0, 0, 0)$. The luminous region follows a three dimensional gaussian distribution with transverse width $(\sigma_x, \sigma_y) = (15\mu m, 15\mu m)$ and a longitudinal width of $z = 55mm$. Most particles come from The production point, or vertex of the particle is the origin of the track)

In most cases, the particles come from the production points which are in the luminous region, however a small fraction of the particles come from short-lived unstable heavy particles flying a few millimeter.

3.3 The detector

Structure

The detector that has been simulated is called the TML detector. It is formed of three concentric devices, each of them composed of a cylindric structure (the barrel) and multiple discs (end-caps) on each side of the (figure 3).

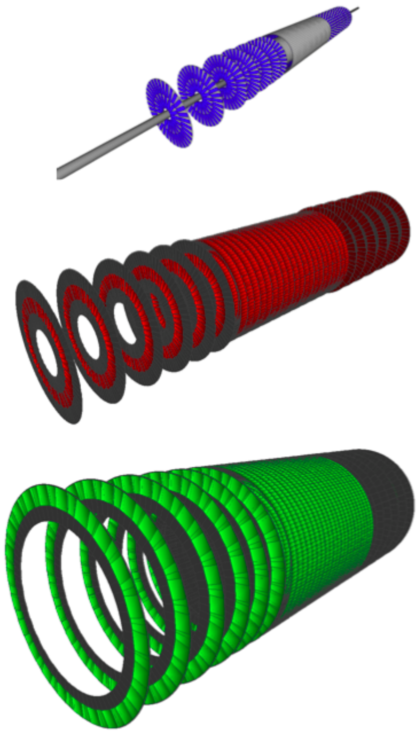


Figure 3: 3D model of the TML detector: the different detectors are shown separately with different colors, with the innermost pixel detector in blue, followed by a short (red) and long strip (green) detector. In reality, there are nested in one another.

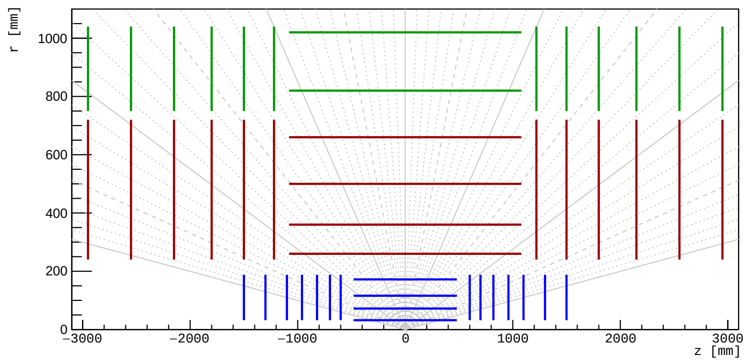


Figure 4: Longitudinal midsection plane of the TML detector, same color code as in figure 3. The 3D detector layout is obtained by rotating

Figure 4 shows the internal organization. The figure represents the mid section of the barrels + discs system, viewed from the beam direction (imagine you cut a can along its length and you look at one of the halves from above). The innermost detector (blue) is built from four cylinders and seven discs (end-caps) on each side. It is surrounded by a short strip detector (red) with four barrel layers and six endcap discs on each side. The long strip detector is the outermost detector device with two barrel layers and six end-cap disks.

In the dataset, the barrels and end-caps are uniquely identified with volume identifiers (`volume_id`); within a single volume, the individual layers (the blue, red and green lines in figure 4) are uniquely identified with a layer identifier (`layer_id`).

The TML detector is built from planar modules (mostly rectangular, or of trapezoidal shape in the end-caps), composed of sensitive devices, the cells (further described in section 3.5). Modules within a layer are uniquely labeled with a module identifier (`module_id`).

Measurement

A charged particle, when traversing a silicon module, induces ionization charge that is read out by a segmented readout chip, called a *channel*, finally delivering a measurement of the particle position.

Modules somewhat overlap in order to ensure hermetic coverage. The overlap of modules within a layer, combined with the concentric geometry of the layers, and the pooling of measurement at the module level (see section 3.5), result in the following simple rule.

A given particle can indeed produce several measurements on a given layer, but only one measurement on a given module.

Illustration

Assume you consider two hits, with the same overall structure identifier (`volume_id`,`layer_id`, `module_id`), and two different positions (x_1, y_1, z_1) and (x_2, y_2, z_2). They necessarily belong to different track.

3.4 Geometry of the trajectories

Coordinates system

The coordinate system is a right-handed cartesian coordinate system (x, y, z) with the global z axis defined along the beam direction, i.e. the ML detector builds a cylindrically shaped object around the global z axis. The $(x - y)$ plane is referred to as the transverse plane, and the azimuthal angle $\phi \in [-\pi, \pi]$ is defined in the transverse plane with $\phi = 0$ denoting the x-axis. The polar angle θ is measured from the z-axis and is defined to be within $[0, \pi]$.

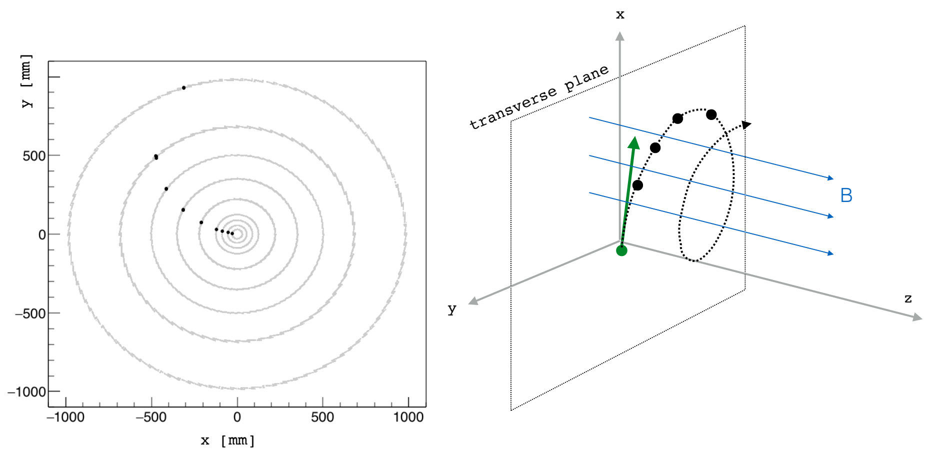


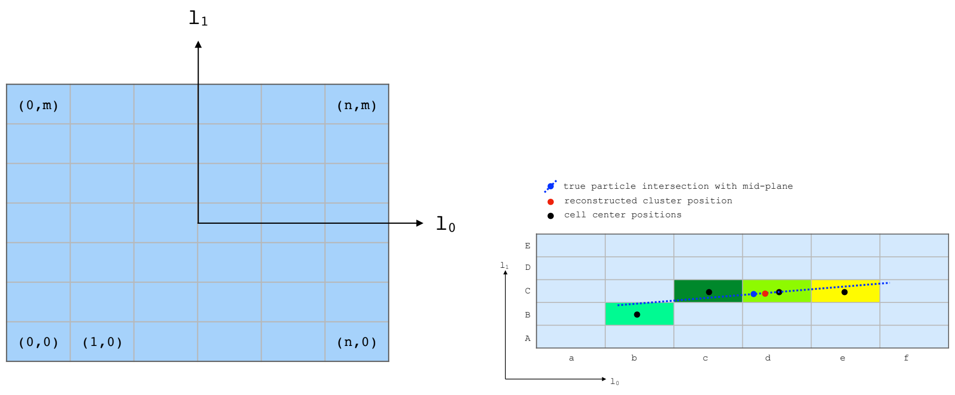
Figure 5: Illustration of a particle moving through a constant magnetic field. In the transverse projection, this corresponds to the particle moving along a circle.

Perturbed helices

In order to measure the particle momentum, tracking detectors are embedded in a strong magnetic field. A charged particle, when moving with constant velocity through a constant magnetic field follows a helicoidal trajectory (figures 5). The relation between the parameters of the helix and the momentum are explained in appendix **??**.

For inner tracking detectors, the magnetic field is usually (as much as possible) aligned with the beam direction, such that the particle is bent in the transverse plane. The intrinsic (as opposed to measurement errors) departing from a true helix has two causes. The departure from a perfect helix measured at the exit of the 1 meter radius TML detector would be a few millimeters .

- A perfect constant magnetic field is not possible to generate.
- Interactions with the detector material (Multiple Coulomb Scattering in physics terminology), result in a stochastic deflection following a normal (gaussian) distribution with mean 0 and a certain width. The width is given by the amount of the traversed material and the momentum of the particle. The dependency is rather



l1
(0,m)
(n,m)
l0
(0,0)
(1,0)
(n,0)
true particle intersection with mid-plane
reconstructed cluster position
cell center positions
E
D
C
B
A
l1
a
b
c
d
e
f
l0

Figure 6: Pixel-level measurement. (a) organization of a module (b) Measurement when a particle traverses multiple pixels.

intuitive: the more material is traversed the broader is the resulting spread, the *faster* the particle moves, the smaller the spread.

3.5 Measurement accuracy

The elementary sensitive device is the cell, which is called pixel for the innermost detector, and pitch in the strip detectors. Cells are assembled to form modules (fig. 6 (a)).

There are two types of measurement errors.

- A hit may not be recorded at all, due to thresholding in the recording device. This is rare (less that $\sim 1\%$).
- For all recorded hits, the so-called reconstructed positions, are only an approximation of the real (x, y, z) value of the hit.

There are two possible cases .

- Only one cell of the module is traversed by the particle; then the center of the cell is taken as the measurement position. The distribution of the discrepancy between the real and measured position depends on the situation of the module in the detector.
- More than one cell of the module is traversed. This happens when a particle hits a module tangentially enough. The measured position is reconstructed from all involved cells (fig. 6 (b)).

The (x, y, z) positions that are given in the training and test files are reconstructed positions. The positions in the truth are the exact ones, more precisely, the position that correspond to the intersection of the trajectory with the bisection plane of the module depth-wise.

See appendix A for a summary of the reconstruction methods, and section 2.3 of A. Salzburger paper for a more detailed description and quantitative information. .

3.6 Illustration: the Hough transform

The methods currently in use are essentially combinatorial, typically highly optimized Kalman filters. Thus, the application of the Hough transform presented below is not representative of the production methods. However, it exemplifies in a simple way the impact of the perturbations described in the previous sections.

Warning: please note that this is definitely not intended as a suggested approach. The only goal is to illustrate the effects of some of the above-mentioned issues with measurement.

The general method in the Hough transform maps the points into the parameter space of helices and looks for set of parameters which satisfies the maximum number of image space points. It is linear in the number of points (hits), thus fast. A kernel illustrating the Hough transform will be submitted to Kaggle on the first days of the challenge.

4 The dataset

A detailed description of the data provided is given below. It is recommended to use the provided helper library[7] to read the data.

4.1 Dataset structure

The dataset is structured in per event files. All files are in csv comma separated value format.

- `event<eventnumber>-hits.csv`: per event hits file, with the 3D coordinates of the hits and other features.
- `event<eventnumber>-truth.csv`: the ground truth for `event<eventnumber>-hits.csv`, with the connection to the generated particles.
- `event<eventnumber>-particles.csv`: per event particles file, with the origin and monmentum of the particles.
- `event<eventnumber>-cells.csv`: detector-related additional features of the hits.

The training dataset includes all the files. For the test dataset, only the hits and cells files are provided.

[7]https://github.com/LAL/trackml-library

4.2 Data description

Warning: for completeness, we provided most of the features that could be available in reality. In particular, the hit cells might be useful for fine tuning, but can safely be ignored initially.

Event hits

- `hit_id`: a numerical hit identifier, unique inside the event.
- `x`, `y`, `z` : the reconstructed position (x, y, z), with the accuracy limitations described in section 3.5.
- `volume_id`,`layer_id`, `module_id`: the module where the hit was read out by using the identification scheme from section. 3.3. For technical reasons the layer identifier takes only even values. The other values are contiguous.

Note that the volume, layer, and module identifiers could be in principle be deduced from (x, y, z).

Event Particles

- `particle_id`: a unique identifier of a particle inside the event.
- `vx,vy,vz`: the coordinates (in millimeters) of the particle vertex , ie the origin of the trajectory (section 3.2).
- `px,py,pz`: initial momentum (in GeV/c) at particle vertex
- `q` : the charge as multiple of the absolute electron charge
- `nhits`: number of hits generated by this particle

Event truth file

The truth file contains the mapping between hits and generating particles, together with the true particle state at each measured hit. Each entry maps one hit to one particle.

- `hit_id`: numerical identifier of the hit as defined in the hits file.
- `particle_id`: numerical identifier of the particle generating this hit as defined in the particles file.
- `tx, ty, tz`: true hit position (in mm) at the intersection with the module. Remember (section 3.5) that the true hit position is the intersection of the trajectory with the module mid-plane.
- `tpx, tpy, tpz` true particle momentum (in GeV/c) in the global coordinate system at the intersection point. The corresponding unit vector is tangent to the particle trajectory.
- `weight`: per-hit weight used by the scoring function; note that the total sum of weights within one event equals to one.

Event hit cells

- `hit_id`: numerical identifier of the hit as defined in the hits file.
- `ch0, ch1`: cell identifier/coordinates unique with one module, encoded as in figure 6. Depending on the detector type only one of the channel identifiers may be valid, see Appendix A.
- `value`: Signal value information, e.g. how much charge a particle has deposited. See Appendix A for a discussion of the resolution.

4.3 Submission format

A submission is a list of integer triples (event_id,hit_id,track_id).

A Kaggle submission is a single csv file, with the list of triples for all the events. The notebooks provide examples for building submissions. The csv file must have the header `event_id,hit_id,track_id`; it the file must be readable with a standard csv reader expecting 3 integers. Participants are advised to compress the file (with zip, bzip2, gzip) before submission.

The submission file must associate each hit in each event to one and only one reconstructed track. The reconstructed tracks must be uniquely identified only within each event. More precisely, what is expected is a list of:

- `event_id`: numerical identifier of the event; corresponds to the number found in the per-event file name prefix;
- `hit_id`: numerical identifier (non negative integer) of the hit inside the event as defined in the per-event hits file;
- `track_id`: user defined numerical identifier (non negative integer) of the track; they do not have to be contiguous.

The values are not checked at submission time, and might be spurious . However, submissions are screened along the following rules:

1 event_id must be in the expected range for the concerned sample;

2 hit_id must be in [0,1E6];

3 track_id must be in [0,1E9];

4 any tuple (event_id, hit_id) must appear at most once.

A submission either unreadable or not complying with the screening rules is invalid. No score is computed. The participant is warned by an error message. In the validation phase, the submission is deduced from the daily submission rights. To help preventing invalid submissions, a validation script implementing these rules is provided.

Note that the validation script does not check elementary errors. In particular, it accepts spurious values of event_id and hit_id that are absent in the dataset, although the corresponding entries will obviously be discarded for scoring.

References

[1] C. Gumpert, A. Salzburger, M. Kiehn, J. Hrdinka, and N. Calace, "ACTS: from ATLAS software towards a common track reconstruction software," *J. Phys. Conf. Ser.*, vol. 898, no. 4, p. 042011, 2017.

A Resolution

In the following, (m_0, m_1) denotes a reconstructed position, and (t_0, t_1) the true position, of a hit, in the 2D coordinate system of the module. The two dimensional residual is defined as $(r_0, r_1) = (m_0 - t_0, m_1 - t_1)$.

In all cases, if only one cell is traversed, the reconstructed position is the center of the cell (c_0, c_1).

Innermost pixel detector

The cells are square pixels of $p_0 \times p_1 = 50\mu m \times 50\mu m$.

If only one cell is traversed, the residual distribution is basically a box distribution between $[\frac{-p_i}{2}, \frac{p_i}{2}]$

In the innermost pixel barrel, the very shallow incident angle in the forward (z) direction creates very large cluster sizes in the l_1 direction. As the readout electronics records how much charge was induced by that particle, and the deposited charge is roughly linear to the path length in the pixel, the reconstructed cluster position is built as a weighted mean position (*analog clustering*):

$$(m_0, m_1) = \frac{1}{Q_i} \sum_i q_i (c_0, c_1)_i, \tag{1}$$

where $Q_i = \sum_i q_i$ and q_i is the individual read out charge value of pixel i.

Strip detectors

The particle density per detection area decreases rapidly with the distance to the interaction point. The particle localisation at outer radii is mainly needed for the curvature measurement (contrary to the innermost measurements that are needed for the particle origin measurement). This allows outer tracking detectors to have a reduced granularity, which directly translates in bigger measurement structures. Since the highest possible precision is not required, the readout is limited to binary (i.e. a strip either receives a signal or not) and no charge information is present. This simplifies equation Eq. 1 to

$$(m_0, m_1) = \frac{1}{N} \sum_{i=1}^{N} (c_0, c_1)_i, \tag{2}$$

the arithmetic mean of the individual cell centers, and is consequently called *digital clustering.*

For the Short Strip detector, the cell (strip) geometry is trapezoidal: $80\mu mm \times 1.2$mm, yielding in strongly reduced resolution in the longitudinal direction. Due to the larger segmentation, cluster sizes are basically limited to 1 or 2. For the Long Strip detector, the cell (strip) geometry is also trapezoidal: $120\mu m \times 10.8mm$.

Missing hits

A minimum path length of the particle in the silicon is required for the strip cell to be above threshold in order to be read out. The actual path length is smeared to emulate a readout uncertainty, by 10 % following a normal distribution.