Study of Vertex Finding and Multiplicity Determination in Au-Au Collisions using the sPHENIX MVTX

Fanxin Sun

University College London, London WC1E 6BT, United Kingdom

fanxin.sun.20@ucl.ac.uk

Abstract. This project investigated a simulated Au-Au collision data sheet under sPHENIX environment, with 10 events in total. The sPHENIX's MVTX will be assembled in three cylindrical layers with the main geometrical parameters listed in Table 1. Finally, we can conclude that for this Au-Au collision simulation events, nearly all the detected hits are due to primary charged particles, which made abandoning "remote" layers cannot improve the found Z coordinates resolution.

Keywords: sPHENIX MVTX; Au-Au Collisions; High Energy Physics.

1. Introduction

1.1 Heavy Ion Collisions & QGP

Ion collision under a relativistic level of velocity is used to confirm the Standard Model as well as looking for particles beyond the SM.

Basically, two relativistic ions smash and form a tiny volume of Quark–gluon plasma (QGP) [2]. According to initial conditions of those ions, the plasma will reform & reconstruct undergoing some nuclear reactions into many high speeds charged particles. By detecting these charged particles, physicists are able to know what happened at the collision point, called a "vertex", while one time of collision is called an "event".

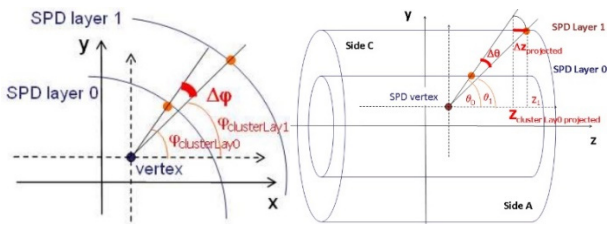
Charged particles which directly propagated from the QGP are called primary ones, some of them may further nuclear react and produce secondary charged particles.

1.2 Tracklet Method

After a collision event occurred, generated charged particles will hit onto the sensors mounted on the three MVTX layers of sPHENIX, forming a cluster of information respectively for each layer. Briefly, a "tracklet" is defined as taking out one hit from a layer's cluster, then looking for a companion hit from another layer in alignment with the reconstructed primary vertex, within a fiducial window [3].

1.3 Physics Parameters

As shown below in Figure 1, two main parameters to describe a hit is the azimuthal angle φ and the pseudo-rapidity η



(a)Showing φ and Δφ in x-y plane (b) Showing θ in 3D, which determines η

Figure 1. Taking illustrating examples from ALICE [4], which has two layers of cluster

Taking a hit's global coordinate as (x_g, y_g, z_g), its corresponded vertex's coordinate as (x_v, y_v, z_v), then in the vertex system, the hit's coordinate becomes:

$$x = x_g - x_v \tag{1}$$

$$y = y_g - y_v \tag{2}$$

$$z = z_g - z_v \tag{3}$$

then we can get the azimuthal angle φ by (in C++):

$$\varphi = \text{atan2}(y, x); \tag{4}$$

and the polar angle θ by (in C++):

$$\theta = \text{atan2}(r, z); \tag{5}$$

where r is given by:

$$r = \sqrt{x^2 + y^2} \tag{6}$$

Finally, the pseudo-rapidity η is given by (in C++):

$$\eta = -\log(\tan(\theta/2)) \tag{7}$$

for z > 0 and

$$\eta = \log(\tan((\pi - \theta)/2)); \tag{8}$$

for z <= 0.

1.4 Multiplicity Density

The multiplicity of charged particles is considered a key observable to characterize the properties of the particles created in collisions [5]. For this report, we take the value $dN_{ch}/d\eta$ as the multiplicity density of the event, where the subscript "ch" represents "charged particles", the range of η is from -2 to 2.

2. Data Sheet Check

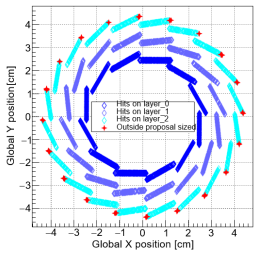
A simulated Au-Au collisions data sheet with 10 events is provided, with 5735 hits in total.

2.1 Hardware Check

From the proposal document of sPHENIX [1], the dimensions of detecting panels are known as:

Table 1. A table without vertical lines.

	Layer 0	Layer 1	Layer 2
Radial position(min.) (cm)	2.37	3.14	3.91+
Radial position(max.) (cm)	2.80	3.59	4.34+
Length (cm)	27.1	27.1	27.1+
# of hits	2133	1892	15894

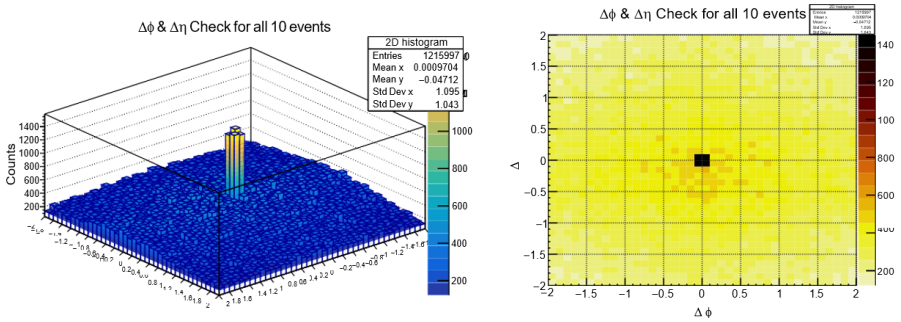


Layers Check of Au-Au Events
Global X position [cm]
Figure 2. Checking all hits distributions in 3 layers

which means there are 121 “ghost” hits in the data sheet, shown in Figure 2 in red stars.

2.2 Event’s “Parameters” Check

To check the distributions of Δη and Δφ among different layers, we fill (Δη, Δφ) pairs into a 2D histogram with bin range of (-2, 2) for both Δη and Δφ, and show it in Figure 3.



(a)3D distribution of Δη and Δφ (b) 2D distribution of Δη and Δφ

Figure 3. Using 2D histogram to check distribution of (Δη, Δφ) pair.

From Figure 3, we can confirm that the “signal” information is concentrated in an area of $|\Delta\eta| < 0.1$, $|\Delta\varphi| < 0.1$, which looks like a skyscraper.

3. Z Vertex Finding

3.1 Methodology

We use tracklet method to find Z vertex coordinate of an event, whose algorithm is explained in Figure 3 below:



Figure 4. Basic flowchart showing how to scan out a Z coordinate of vertex.

3.2 Parameters for Scanning

As mentioned in Figure 3.1, there are two parameters to be set in this algorithm: Δη and scan step, which means if the scan step is set as 0.03 cm, with the scan beginning with z candidate as -15 cm,

then in the next time of scan the z candidate becomes -14.97cm, and so on until it ends with z candidate is equal to 15 cm.

To be noticed that in a non-simulation data sheet, we may not know the true x and y coordinates of the vertex, so we approximately take global (0, 0) as "true" (x, y) value since the incident beams are both constrained within an extreme small area around the origin in x-y plane.

In Figure 4, we tried many sets of threshold and scan step values, determining the found Z value as the bin with maximum entries in the histogram. From the plots, we can confirm that a smaller threshold of $\Delta\eta$ will bring the found Z coordinate closer to the true value. Also, decreasing the scan step can also improve the resolution, but it will increase the scanning running time significantly, especially for those events with larger number of hits.

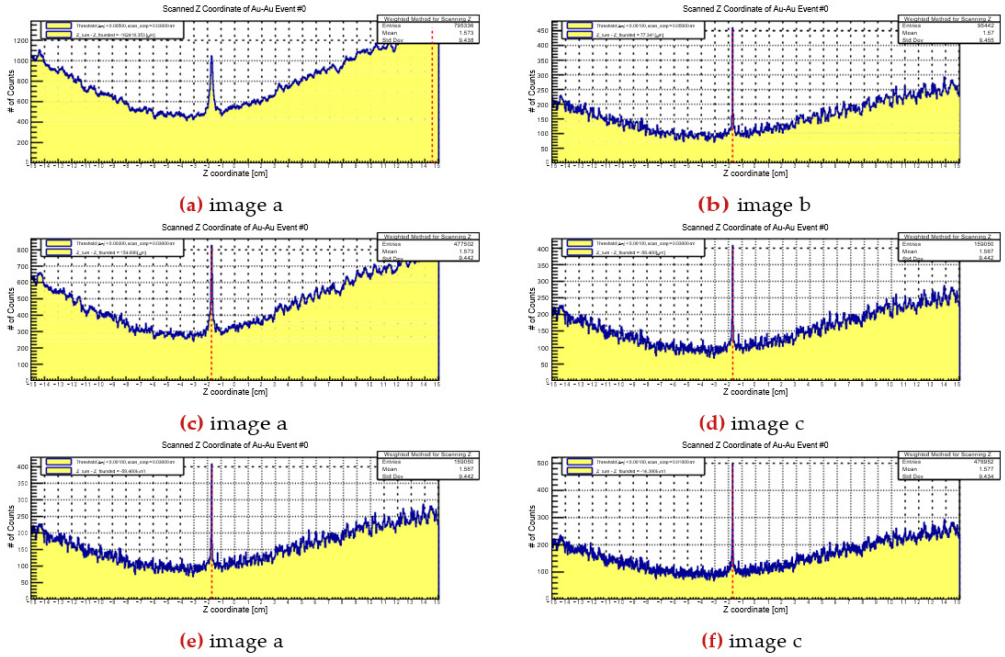


Figure 5. Searching for better parameter values set

3.3 Weighted Method

You may have noticed that in the last subsection the scan step only has a precision of 2 after the decimal point. To further increase the resolution, we adopt an alternative method which is called "weighted method", whose algorithm is simple:

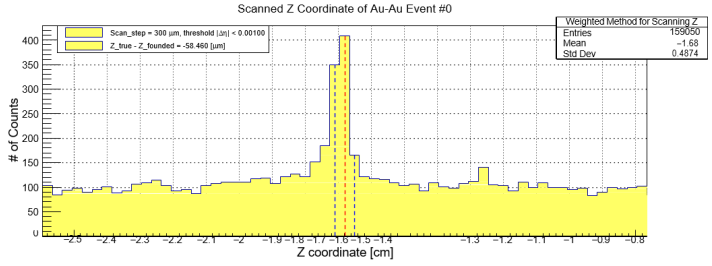
(i). find out the bin b_0 with maximum entries as well as its number of entries n_0;

(ii). find out the bin b_{-1} next to b_0 on the left-hand side as well as its number of entries n_{-1};

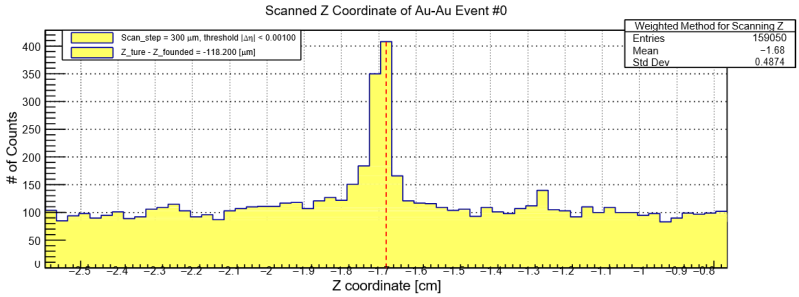
(iii). find out the bin b_{+1} next to b_0 on the right-hand side as well as its number of entries n_{+1};

(iv). set the found z as $(b_0*n_0 + b_{-1}*n_{-1} + b_{+1}*n_{+1})/ (n_0 + n_{-1} + n_{+1})$.

By using this method, we can check from Figure 6 below that the resolution is greatly improved with the same set of scanning parameters.



(a)



(b)

Figure 6. Found Z with weighted method compared with unweighted method.

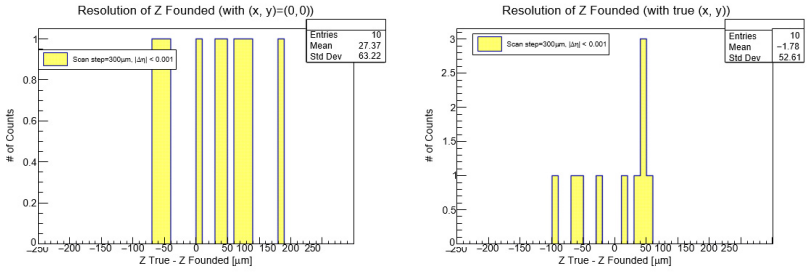
3.4 Found Z with True (x, y)

As investigated above, to obtain a good resolution of found Z with fast scanning speed, we further investigate our method with applying scan parameters as $\Delta\eta = 0.001$ and scan step = 300μm to all the experiments listed in Table 2 below.

Table 2. A table showing found Z vertex with different (x, y).

Z [cm]	true Z	w/ (0, 0)	w/ true (x, y)
Event #0	-1.69182	-1.68597	-1.68584
Event #1	0.329866	0.329688	0.328636
Event #2	-6.53345	-6.54154	-6.53903
Event #3	-4.16317	-4.16686	-4.16683
Event #4	-5.53056	-5.52395	-5.52394
Event #5	-2.84232	-2.84671	-2.84664
Event #6	-2.76559	-2.76138	-2.76328
Event #7	3.73688	3.72358	3.74662
Event #8	5.34607	5.33913	5.34206
Event #9	7.5074	7.49995	7.50333

We also compare the resolution of the found Z set using origin (x, y) with the set using true (x, y), shown in Figure 3.6:



(a) image a (b) image b

Figure 7. Comparing found Z using different (x, y)

From Figure 7, the resolution of Z vertex using true (x, y) is much better than that using only (0, 0).

4. Multiplicity Density

4.1 Methodology

We also use tracklet method with a 1D histogram to collect "Signal" and "Background" information of one collision event.

As checked in Figure 7, the information representing tracklets is concentrated in a range of $\Delta\eta < 0.1$ and $\Delta\varphi < 0.1$, called "signal", while all the information outside this range could be called as "background". The flowchart below explains the algorithm to obtain a multiplicity density of an event:

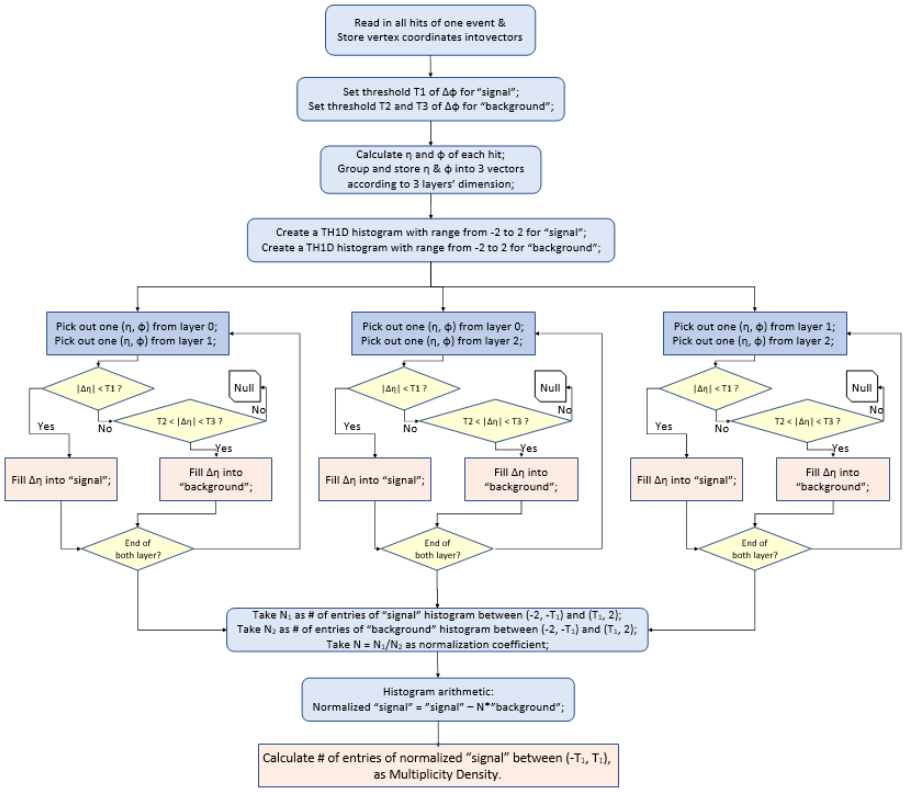


Figure 8. Basic flowchart showing how to obtain multiplicity density using histogram normalization.

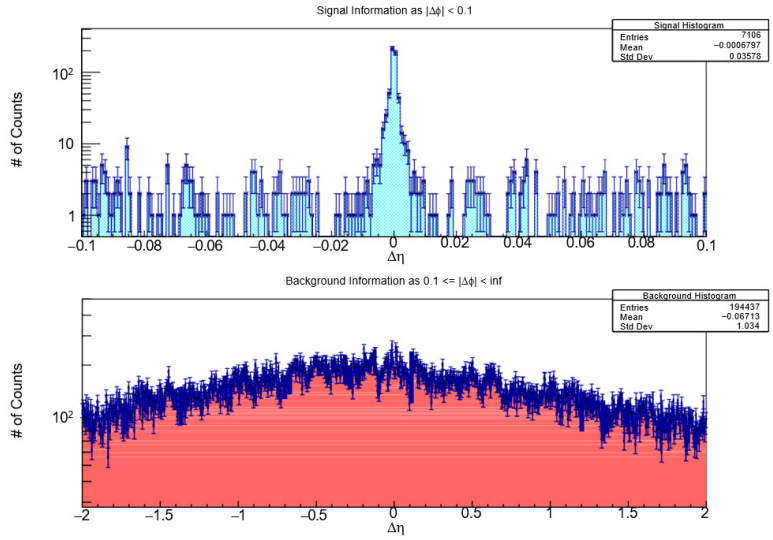


Figure 9. The "signal" histogram (above) and "background" histogram (below) of event #0.

4.2 Signal Histogram Normalization

As we can see in Figure 8, even if we have refined the "signal" information through trickle method, there are still some noises deviated from a "natural" Gaussian shape positioned in the center representing the "true trickle". To eliminate these noises, we need further to normalize the signal histogram with the help of the background histogram.

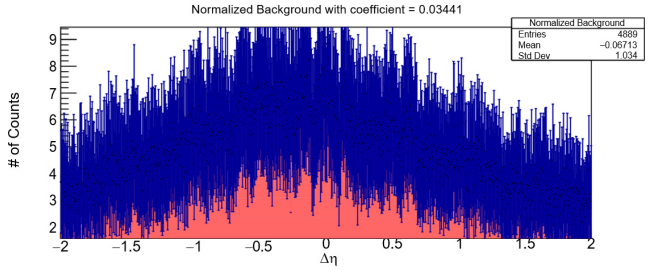


Figure 10. Normalized background histogram of event #0.

4.3 Histogram Subtraction

By using histogram subtraction, we can check from Figure 11 that nearly all the "noise" outside the central information area is cancelled out, the entry of central information is taken as the multiplicity density of this event.

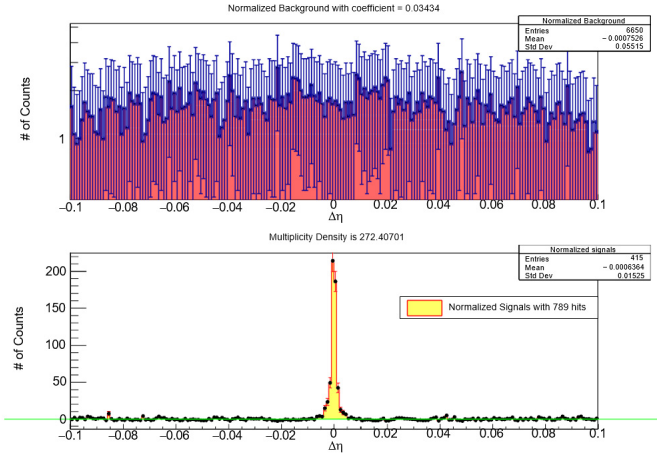


Figure 11. Normalized "signal" histogram of event #0

5. Results

5.1 Multiplicity Density with Different Vertices

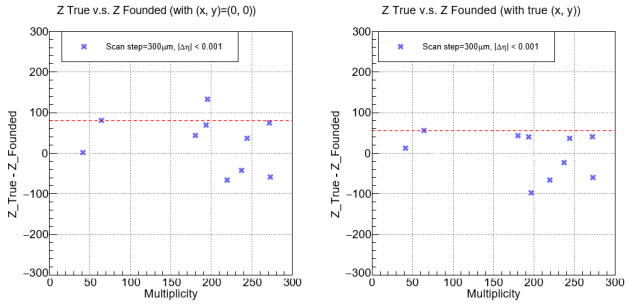
Table 3. A table showing multiplicity density of all 10 events

MD	w/ true Z	w/ found Z_1	w/ found Z_2	# of hits
Event #0	272.407	272.648	272.67	789
Event #1	41.3077	41.3077	41.3077	99
Event #2	64.0621	64.0621	64.0621	152
Event #3	243.8191	243.968	243.968	677
Event #4	219.188	219.371	219.371	638
Event #5	180.159	180.084	180.086	669
Event #6	237.042	237.055	237.005	557
Event #7	196.056	195.119	196.418	704
Event #8	193.256	193.456	193.355	779
Event #9	272.17	271.518	272.003	671

To be noticed that found Z1 is denoted for those being found through global origin of (x, y) while being scanned, and found Z2 stands for that being found through true vertices of (x, y) while being scanned.

From the table 5.1 we can confirm that for the events with very small number of hits, the multiplicity density is not affected too much by hundreds of μms difference of z coordinate.

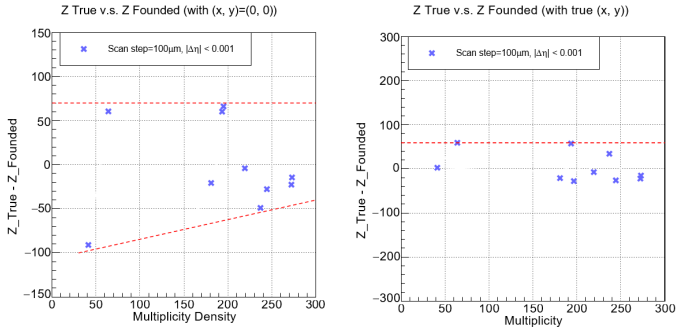
5.2 Resolution v.s. Multiplicity



(a) image a (b) image b

Figure 12. Resolution v.s. Multiplicity Density with different (x, y)

What we have expected is, the resolution will become better with the multiplicity density increasing. But from Figure 12, it's hard to confirm this effect from those using origin (x, y). Although it can be confirmed from those using "true" (x, y), but for most cases of real collision events the true values of (x, y) coordinates are hard to obtain, so we need to refine the method of using origin (x, y) only.

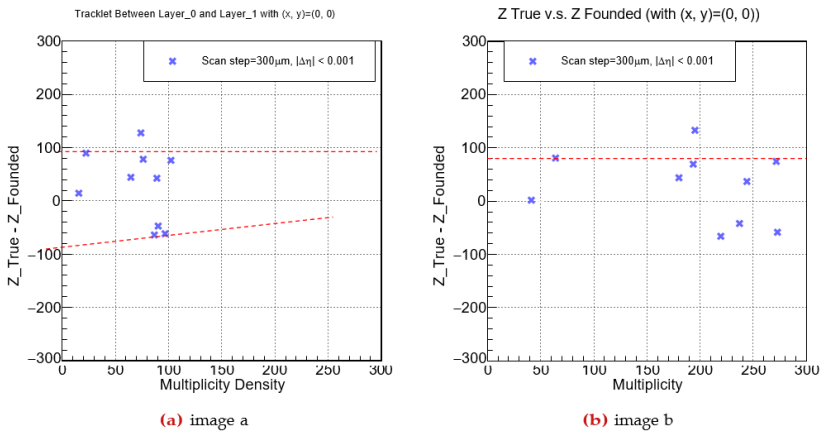


(a) image a (b) image b

Figure13. Decrease scan step from 0.03 cm to 0.01 cm.

In Figure 13, we sacrificed the scanning time duration, decreased the scan step from 0.03 cm to 0.01 cm. This sacrifice worked, not only the expected effect has been confirmed from Figure 13a, but also the overall resolution is compressed. But it's for only 10 events, with this scan step set, it took nearly 5 seconds to process one event. When there would be thousands of events each with thousands of hits... Is there another way to solve the problem?

5.3 Abandon "Remote" Layers



(a) image a (b) image b

Figure 14. On the left plot, the tracklets between layer 1 and layer 2 as well as between layer 0 and layer 2 are abandoned in both Z scanning procedure and multiplicity density determining procedure

As we discussed in Section 1.1, the primary charged particles may further nuclear react and form secondary or higher charged particles, which are "noises" for determining the Z coordinate of vertex. As the farther from the vertex, the more secondary or higher charged particles may form, we filtered the tracklets between layer 0 and layer 2, as well as the tracklets between layer 1 and layer 2, to check what may happen:

Unfortunately, with the same aspect ratio, Figure 5.3 didn't show a distinct improvement of resolution, disappointingly.

6. Conclusion

Finally, we can conclude that for this Au-Au collision simulation events, nearly all the detected hits are due to primary charged particles, which made abandoning "remote" layers cannot improve the found Z coordinates resolution.

Not want to be hated by people who work for determining true x and y coordinates, but scanning with true (x, y) coordinates greatly improves the resolution of found Z coordinates.

The finer resolution you want, the more time you need to spend on scanning for it, equivalent exchange principle always works.

References

[1] Dr. Jehanne Gillo, Ming X. Liu, and et al. A Monolithic-Active-Pixel-Sensor-based Vertex Detector (MVTX) for the sPHENIX Experiment at RHIC. DOE Office, February 14, 2018.

[2] Wikipedia contributors. Quark–gluon plasma, 2022. [Online; accessed 2-May-2022].

[3] Marco Monteno (for the ALICE Collaboration). Particle multiplicity in proton–proton collisions with ALICE. Journal of Physics: Conference Series, (5):209–218, January 2005.

[4] D. Elia, J.F. Grosse-Oetringhaus, M. Nicassio, and T. Virgili. The pixel detector based tracklet reconstruction algorithm in ALICE. ALICE reference number, ALICE-INT-2009-021(version 1.0), November 2009.

[5] ALICE Collaboration. Centrality dependence of the charged-particle multiplicity density at mid-rapidity in Pb–Pb collisions at √SNN = 2.76 TeV. CERN-PH-EP, 2010-071, Dec 06, 2010.