Study of Retina Algorithm on FPGA for Fast Tracking

Wendi Deng and Zixuan Song
Université libre de Bruxelles (ULB)

TWEPP conference - 2018

Introduction

Real-time track reconstruction in high energy physics experiments at colliders running at high luminosity is very challenging for trigger systems. To perform pattern-recognition and track fitting, artificial Retina or Hough transformation algorithms have been introduced in the field which have usually to be implemented in the state of the art FPGA devices.

We study two possible FPGA implementations of retina algorithm : one using online Floating-Point core and one using Look-up Table and fixed-point representation. Detailed measurements of the performance on hardware designs are investigated. So far the Retina has mainly be used in a detector configuration made of parallel planes, without magnetic field. Moreover we report on the simulated performance in a detector configuration made of concentric detection layers with high magnetic field (4Tesla).

Retina algorithm & Track reconstruction without magnetic field

Retina algorithm is inspired from the processing of visual images by the brain where each neuron is sensitive to a small region of the retina. The strength of each neuron is proportional to how close the actual image projected on the retina region is to the particular shape that particular neuron is tuned to [1][2].

In a real HEP detector, the geometry of the detector and the topology of the events are quite complicated. To validate retina algorithm, we have used a simple tracker detector model made of 8 parallel tracking planes in the space without or with small magnetic field. We assume that every 3D trajectory of a charged particle is a straight line from the primary vertex (0,0,0) and identified by a set of 2 parameters (x,y) in the plane. The (x,y) is the spatial coordinates of the intersection point of the track from the last layer (8th) . We discretize the last layer into a number of cells (patterns) 100* 100 , considering it as parameter space (u,v). The vertex (0,0,0) and the center of each cell (u_i,v_j) could identify an ideal track in the detector space uniquely, which means a set of straight lines with an array of the intersection coordinates over all layers are mapped in the space. The distance $\mathbf{s_{i,j}^k}$ of the intersections of the coordinates of the track $(\mathbf{x_{i,j}^k}, \mathbf{y_{i,j}^k})$ from the measured hits $(\mathbf{x_k}, \mathbf{y_k})$ is computed (Figure 1). Then we are able to calculate the excitations **R** of each cell (u_i,v_j) following the function below :

$$R = \sum_k \exp\left(-\frac{{s_{i,j}^k}^2}{2\sigma^2}\right) \qquad (1)$$

where σ is a adjusted parameter for optimal response. The total response of the retina is obtained by calculating the excitations R of all cells. Finally tracks are identified by looking for a local maximum in the response array.

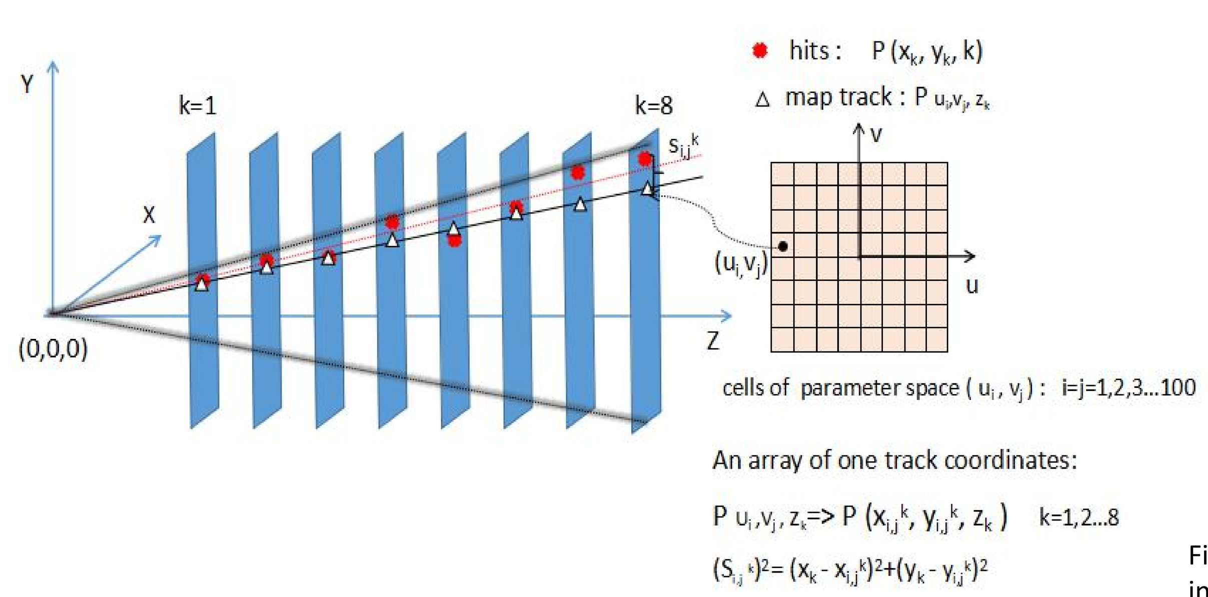

Fig.1 3D detector model without M.F（magnetic field）

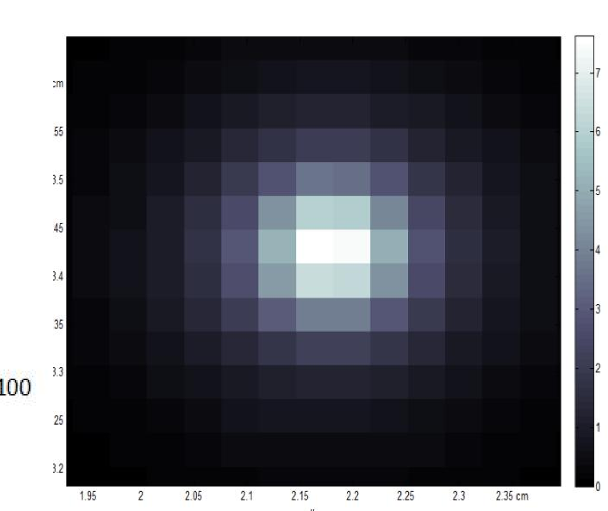

Fig.2 Sub cell excitation levels of Retina to one track in the (u, v) parameter space

Fig 2 shows the simulation with MATLAB of the reconstruction of the straight track in a 8 layer detector with a granularity of 100x100 cells in the (u,v) plane.

Hardware Design:

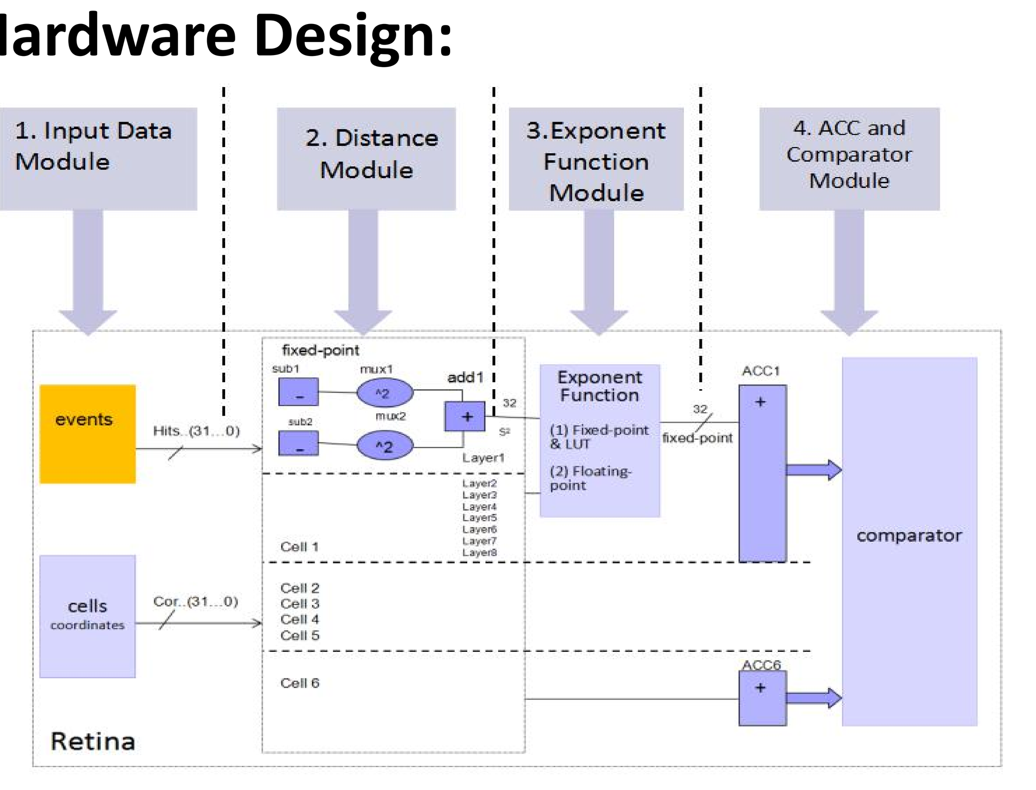

Fig.3 Firmware design architecture of retina algorithm

To validate the algorithm and measure its performances, hardware prototypes have been designed. From Fig 3 you can find our first Retina algorithm: online processor embeds a Floating-Point Operator IP core and a bus standard for on-chip communication AXI [3], providing rapidly and easily floating-point operators that can be targeted to any of the latest Xilinx FPGA platforms. After a system size evaluation for a matrix of 6 cells on one KC705, retina algorithm processes at a fixed latency of 80µs with the cost of 70% FPGA resource.In order to implement retina on FPGA more efficiently and economically,we also use the fixed-point LUT approach to instead the Floating-point approach in Exponent Function Module in which function values are pre-calculated at certain sample points and stored in memory.

Table.1 FPGA resource usage and latency for two firmware designs

Clock (Hz)	Firmware Design	DSP (%)	LUT (%)	LUTRAM (%)	BRAM (%)	FF (%)	Latency (Cycles) /µs
100M	Floating-point	17.14	70.72	15.24	3.03	42.32	156/1.56
100M	Fixed-point and LUT	11.43	7.25	8.46	33.48	9.8	68/0.68

In Table.1, we compare the performance for two firmware designs, under the same test conditions, our fixed-point and LUT firmware typically reduce the latency by a factor 2 and resource usage.

Conclusion and Outlook

In this note, we study FPGA-based implementations of the artificial retina algorithm for fast track reconstruction in trigger system.

In the early stage, we apply retina to a simple detector ignoring magnetic field effects and present on KC705 using both Floating-Point IP and fixed-point &LUT approaches. The performance of implementation including latency, resource, algorithm precision performance have been compared as well, which can be estimated to a complete prototype of hardware system scale.

So far, our research is targeted to adapt retina algorithm to a more realistic tracker detector with cylindrical geometry. Due to the magnetic field effects, charged particle trajectories are bent and treated as partial arc. A first retina modelling of track reconstruction under the situation of particles in a magnetic field with six barrel layers tracker has been setup. Our purpose is to find out the optimal configuration parameter to balance the size of parameter space and reconstruction resolution of Retina. Then change the software design into firmware design, the firmware design about this more complex barrel shape will be started from fixed-point LUT approach witch shows lower calculation latency and much less FPGA resource usage in the case of parallel plane shape.

Track reconstruction with magnetic field

Reconstruction of curved tracks in ideal case:

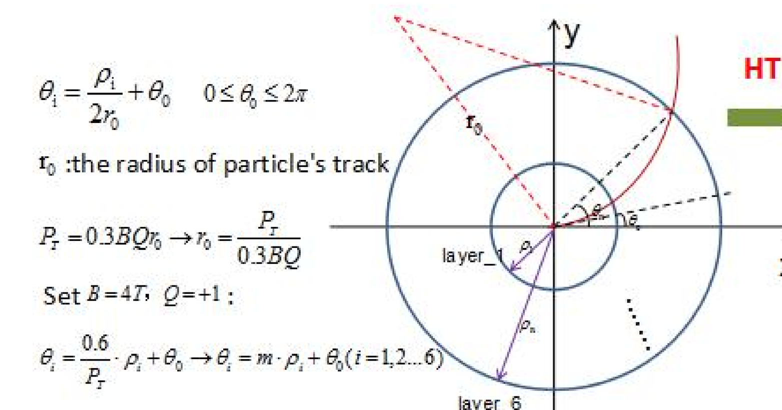

Fig.4 Track modeling with M.F（magnetic field）

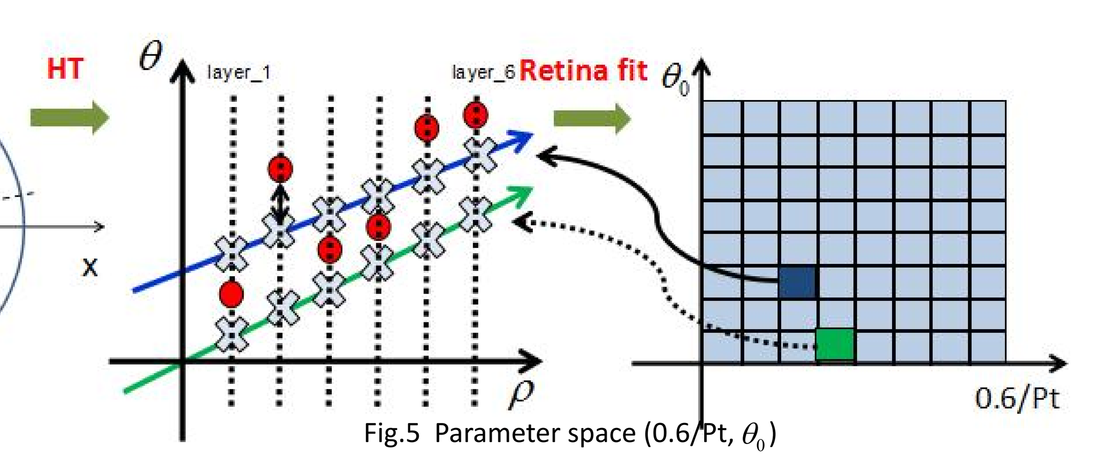

Fig.5 Parameter space (0.6/Pt, θ_0)

A more complex use case is the reconstruction of high Pt tracks in a barrel-like multi-layer detector in presence of a strong magnetic field. In the magnetic field the charge particle trajectory will bend (Fig 4). The geometry is described as six (n=6) concentric circle layers with equal distance between them. The range of radius of those concentric circle layers is from 0.2 ($\rho_1 = 0.2$) meters (innermost) to 1.15 ($\rho_6 = 1.15$) meters (outermost). All the particles will start at the center of the circles with a given initial angle θ_0 , and go across six barrel layers from inner to outer. Due to the magnetic field effect, the shape of the track in the detector area is arc. So we use Hough Transformation change the parameter space into (0.6/Pt, θ_0) where Pt is the momentum of each charged particle and θ_0 is the initial direction angle of each track (Fig 5). In the end we use Retina to find optimal Pt and θ_0 for each individual track by scan the whole parameter space(cells) one by one. The Parameter range of Pt we set to scan by Retina is from 1 GeV to 50 GeV and for θ_0 is from 0 to 2 PI (in radian). Whole parameter map is divided into 400*200 cells, while 400 bins for Pt and 200 bins for θ_0 .

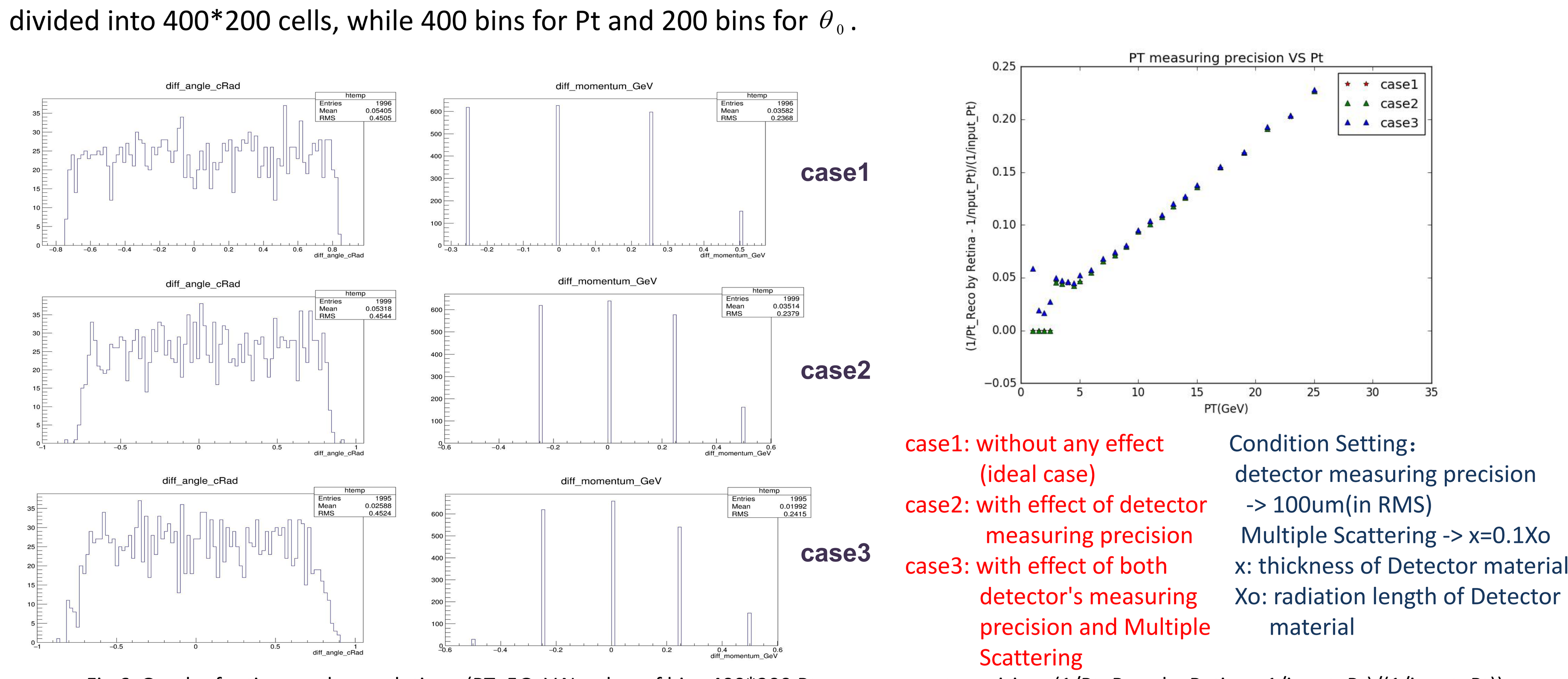

Fig.6 Graph of retina results resolution : (PT=5GeV,Number of bins 400*200,Pt measurement precision=(1/Pt_Reco by Retina - 1/input_Pt)/(1/input_Pt))

Fig 6 shows the Pt reconstruction performance of the first run about the Retina simulations with magnetic field. We select fixed Pt of particles and random θ_0 $(0 \le \theta_0 \le 2\pi)$ as the input track data. Result in left shows the reconstruction Pt (output of Retina) distribution for each input track event with fixed momentum of particles, and the plot in right shows the Pt measurement precision against input_Pt with three different cases.

Reconstruction of Tracks generated by GEANT4:

In order to make track reconstruction more closer to reality, we start to use GEANT4 simulation tools to setup the detector geometry and generate tracks.Then use these tracks to do track RECO.

In this case, the detector geometry is similar to the above ideal case (see Fig. 7). The configuration of track generation is shown in Fig 7.

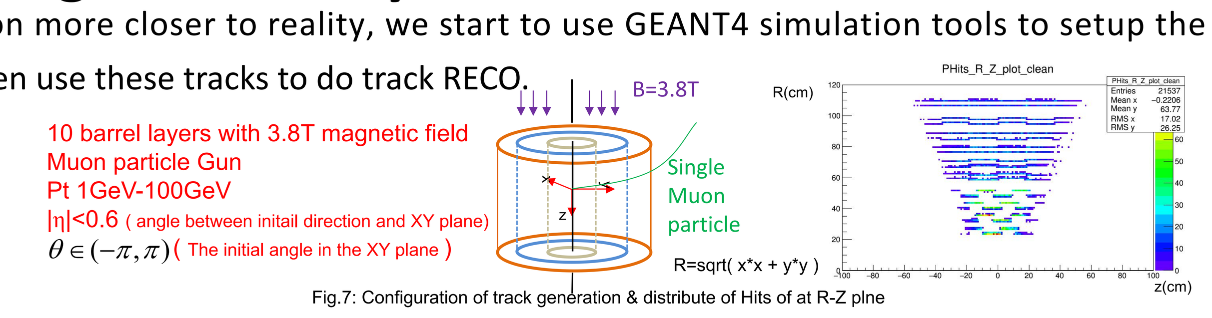

Fig.7: Configuration of track generation & distribute of Hits of at R-Z plne

Single positive Muon(u+) particle is chosen to generate the tracks. All particles will start at zero point and go across all 10 layers from inner to outer and leave there position information (Hits data) on these barrel layers.

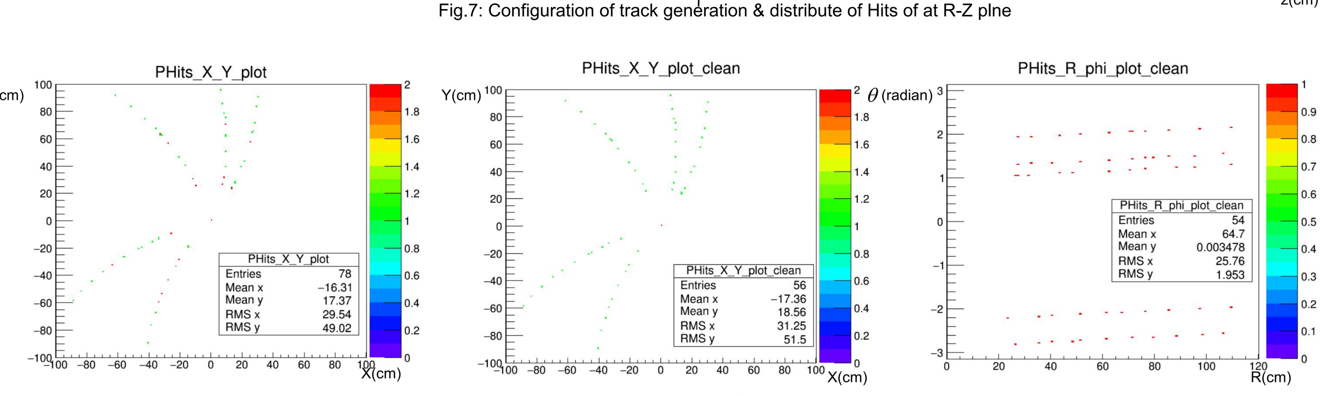

Fig.8: tracks with Pt of 2GeV show in XY plane and transfer to θ -R plane(Hough transform)

The right side of Fig 7 shows the distribution of hits accumulated over 20000 tracks we generated at R-Z plane, from here we can clearly see the collection structure of each barrel layer and their radius. In Fig 8 we also given 5 tracks we generated with GEANT4 in X-Y plane, here we set the Pt of Muon particle equal to 2 GeV. The trajectory of the particle is bend as an arc and turns to straight line when one converts the cartesian coordinate system to polar coordinate system (θ -R plane).

Diff_angle=initial_angle_Reco - initial_angle_input
Diff_Pt=Pt_Reco -Pt_input
Diff_Pt_inverse=(1/Pt_Reco by Retina - 1/input_Pt)/(1/input_Pt)

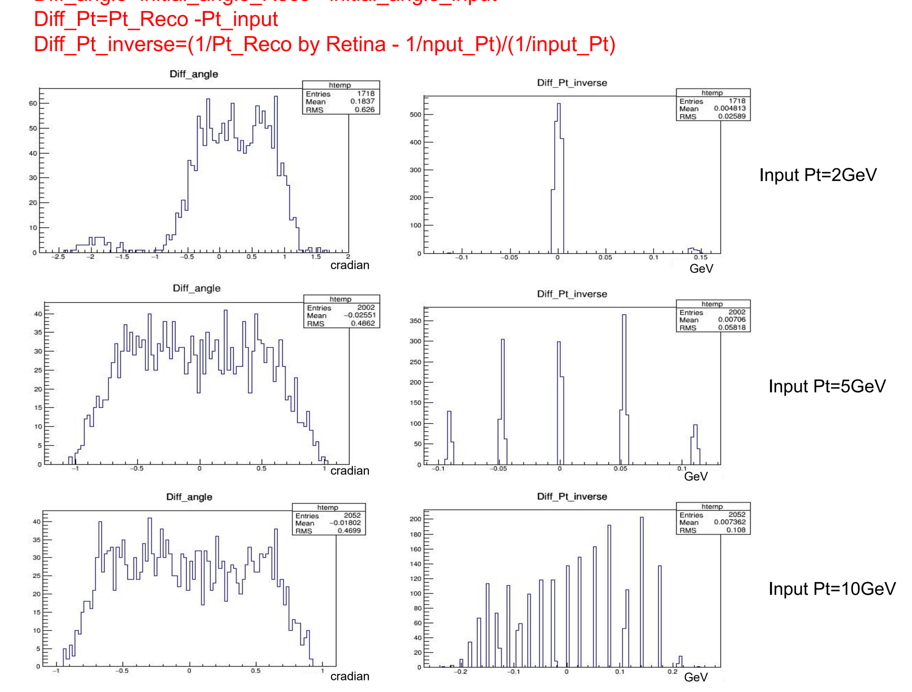

Fig.8:Distribution of Diff_angle and Diff_Pt_inverse in three different particle Pt

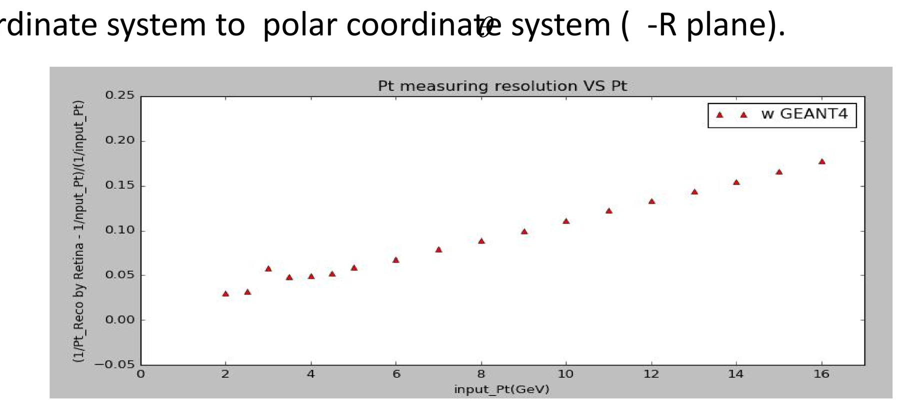

Fig.10:Pt reconstruction resolution VS Pt input

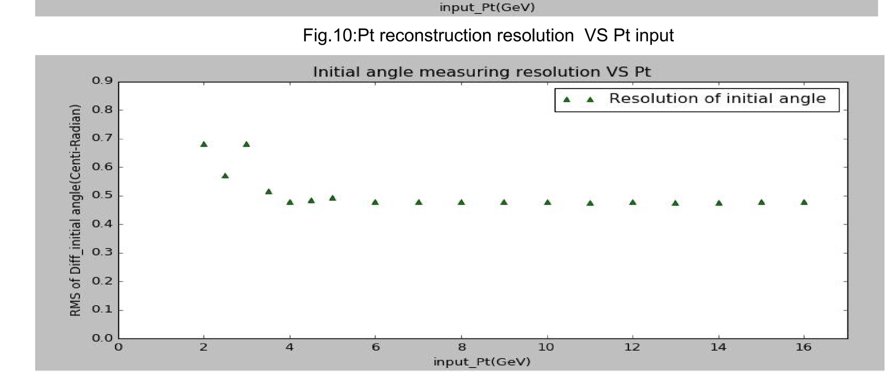

Fig.11:Initial angle reconstruction resolution VS Pt input

The pictures above show the result of track reconstruction by Retina algorithm. Fig 9 is the Diff_angle and Diff_Pt_inverse distribution under three different Pt of particles. Fig 10 and Fig 11 show the Pt and initial angle resolution against Pt of Muon particles. The definition of Diff_angle, Diff_Pt and Diff_Pt_inverse are also given at the top of Fig 8 in red texts.

References

[1] L.Ristori, “An artificial retina for fast track finding,” Nucl. Instrum. Meth. A 453 (2000) 425.
[2] A.abba et al., “The artificial retina processor for track reconstruction at the LHC crossing rate,” JINST 10, C03018 (2014), [arXiv: 1409.1565].
[3] Z.Song et al., Study of hardware implementation of fast tracking algorithms, 2017 JINST 12 C02068.

1. Zixuan.Song@ulb.ac.be 2. dengdandan@mails.ccnu.edu.cn