Central Pattern Recognition

Gianluigi Boca, INFN Pavia

Panda Tracking Workshop Sep. 19. 2018

A central tracker pattern recognition

This Pattern Recognition code was written for the Central part of the panda detector and uses the MicroVertex Detector, the Straw Tube Tracker and the SciTil detector.

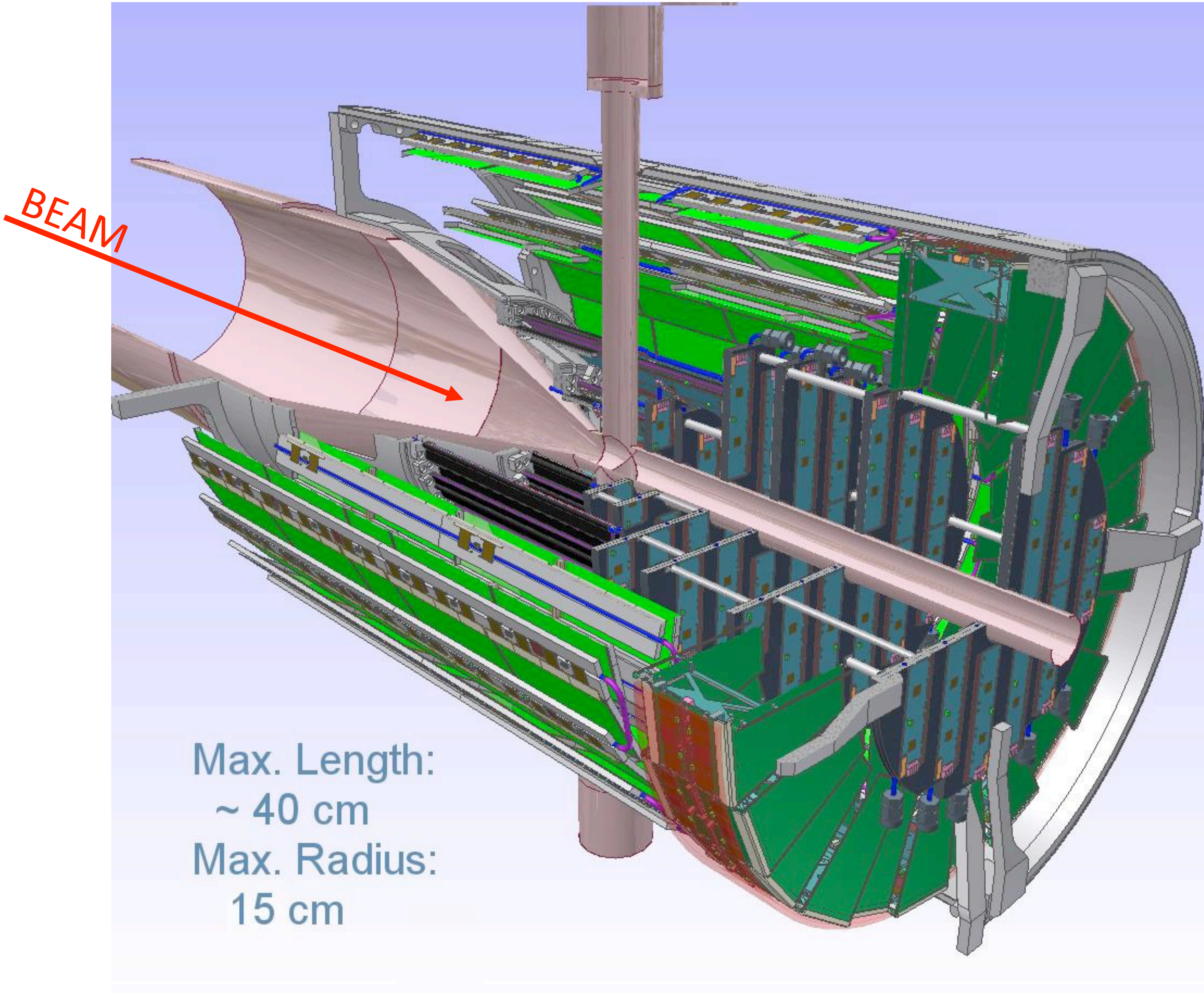
Basic assumptions used :

1) a t_0 is available somehow and consequently the drift time of the each Straw Detector hit is known;
2) tracks come from the interaction vertex at (0,0,0)

ALSO : tracks with less than 2 Straw axial hits and 1 MVD hit are NOT reconstructed.

PANDA
MicroVertex Detector

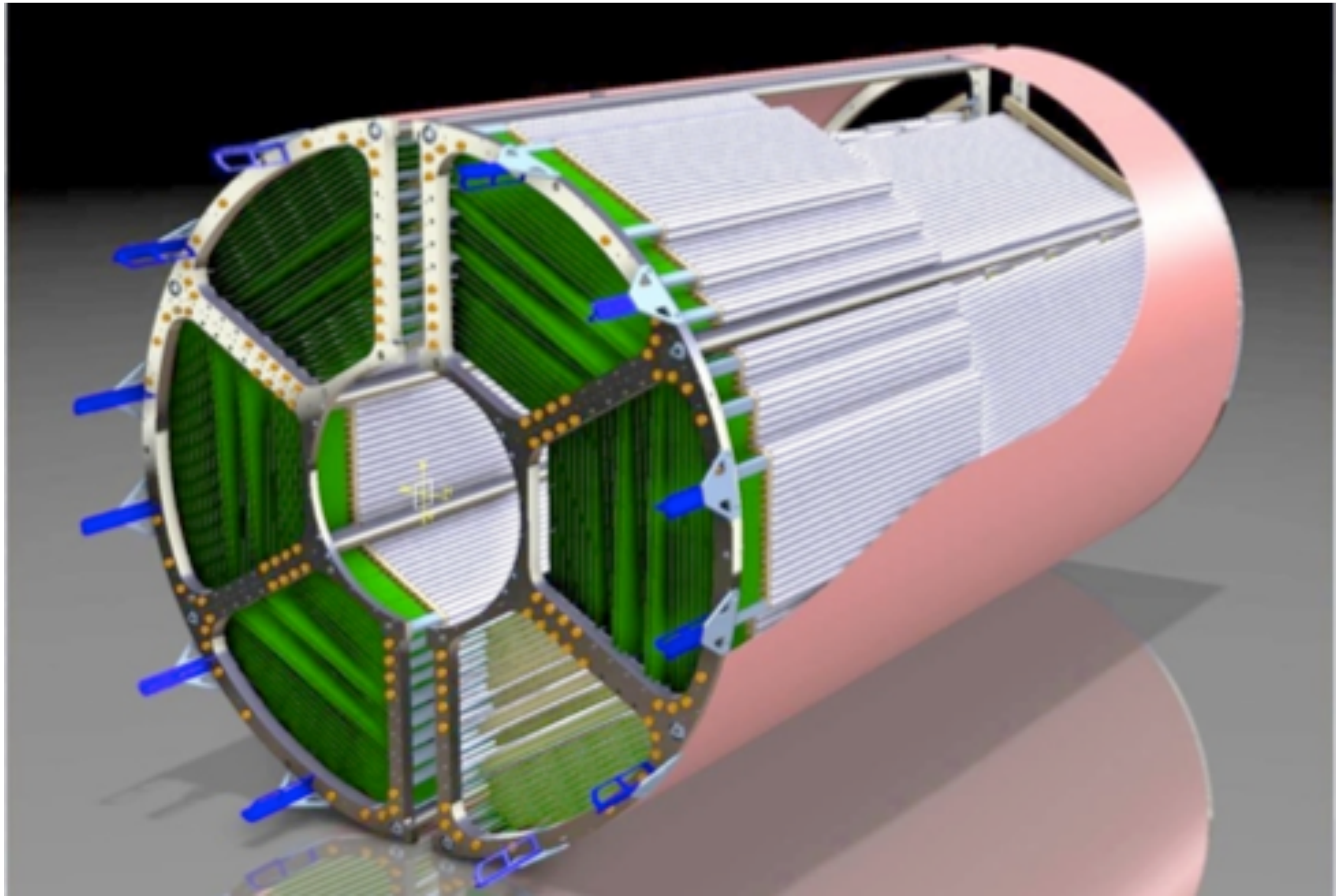
MicroVertex Detector (MVD)

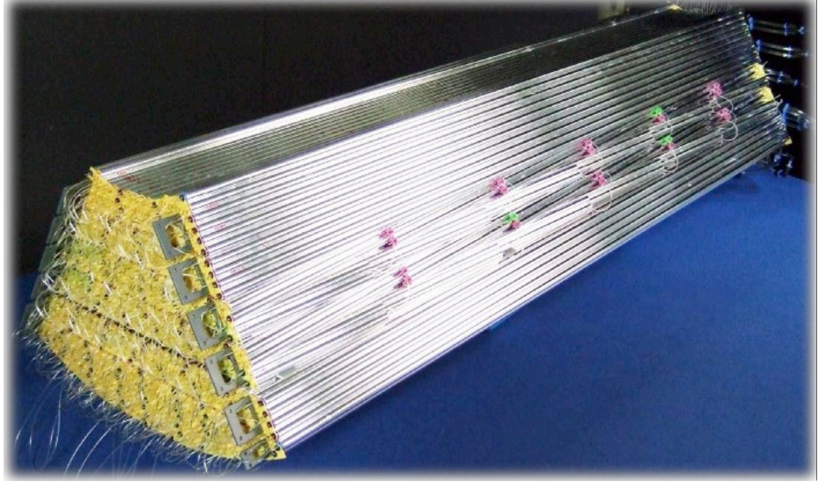


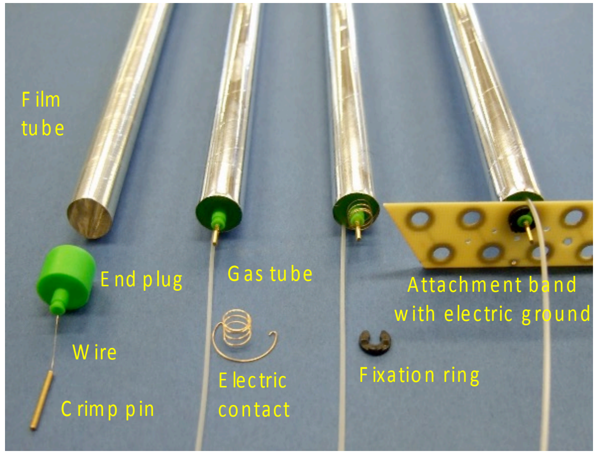
Pixels : 10.3 M Channels
Strips : 200 K Channels

PANDA
Straw Tube Tracker

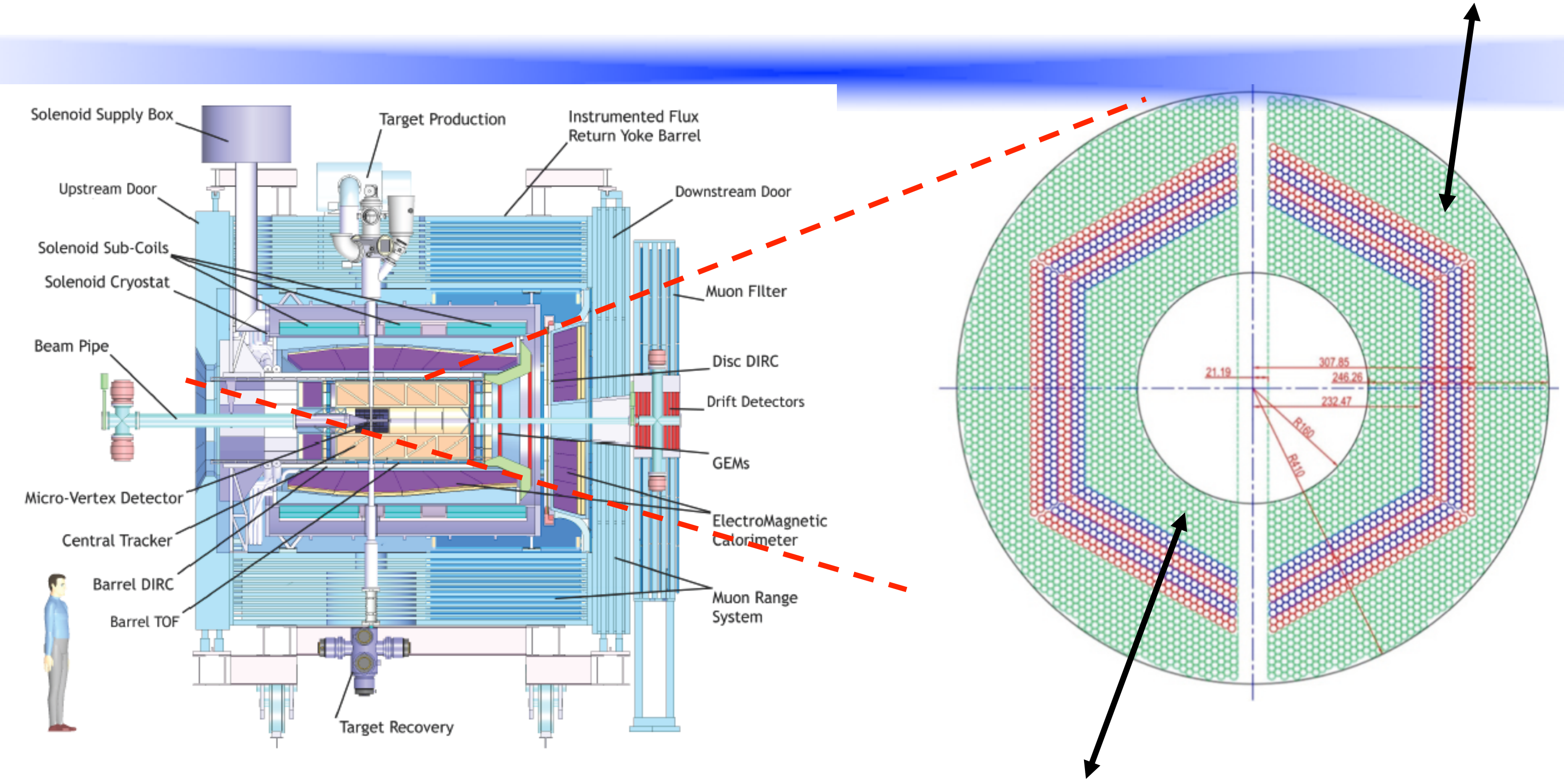
Straw Tube Tracker system (STT)







Straw Tube Tracker system (STT)



- ➢4636 Straw tubes
- ➢23-27 planar layers
- 15-19 **axial layers** (green) in beam direction
- 4 **stereo double-layers** for 3D reconstruction, with ±2.89 skew angle (blue/red)

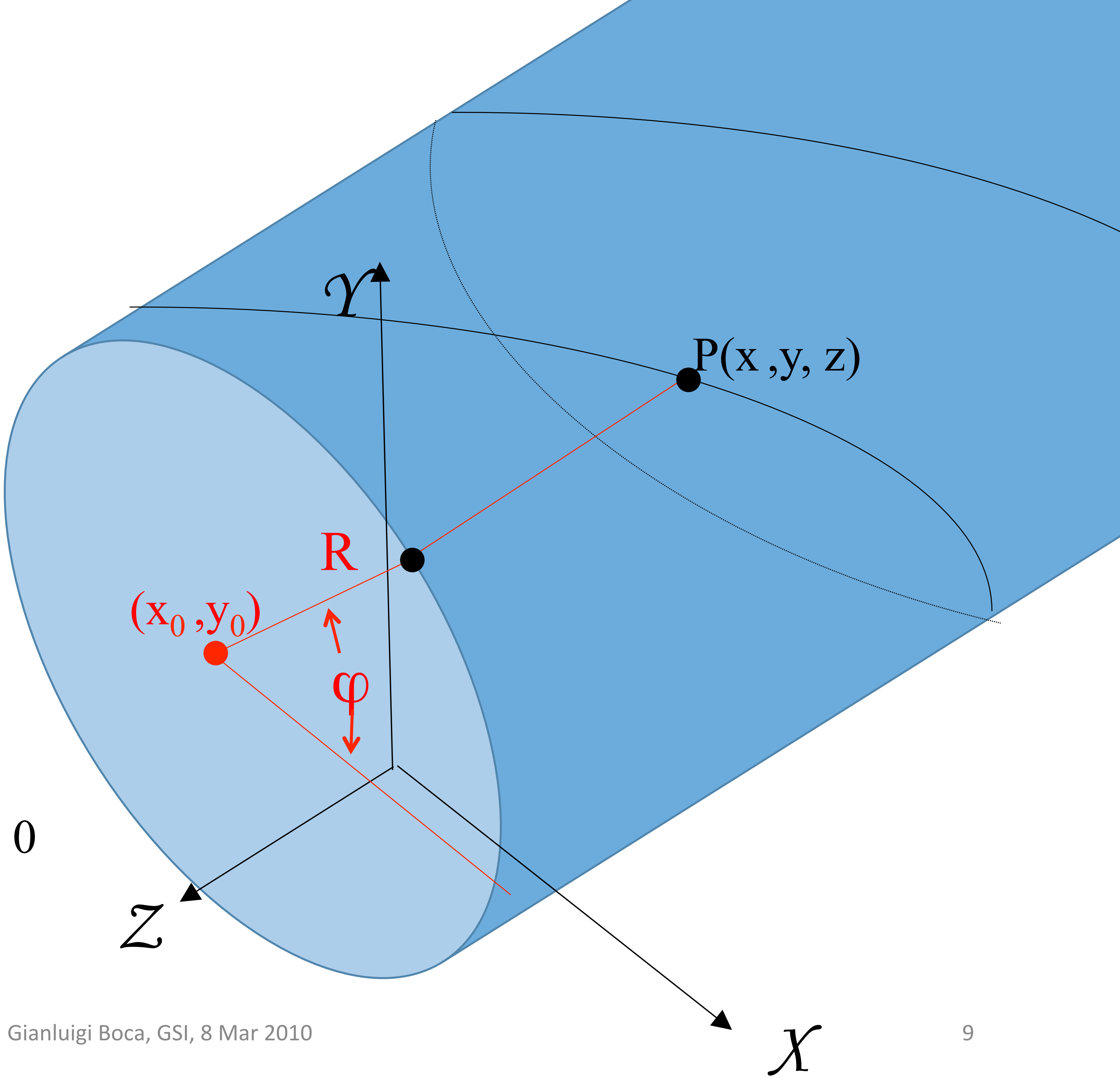
The trajectory parametrization used in this PR

Helix trajectory

$x - x_0 = R\cos(Kz + \varphi_0)$

$y - y_0 = R\sin(Kz + \varphi_0)$

$\varphi = Kz + \varphi_0$



5 parameters :

x_0 ≡ abscissa of center of cylinder

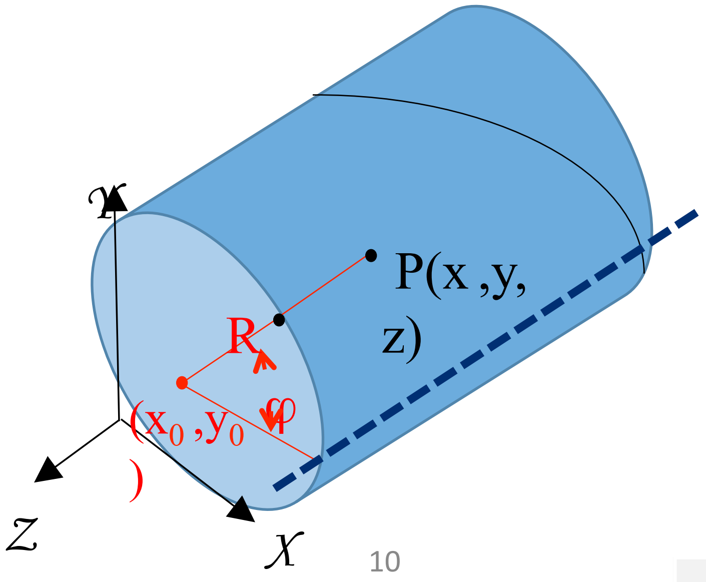
y_0 ≡ ordinate of center of cylinder

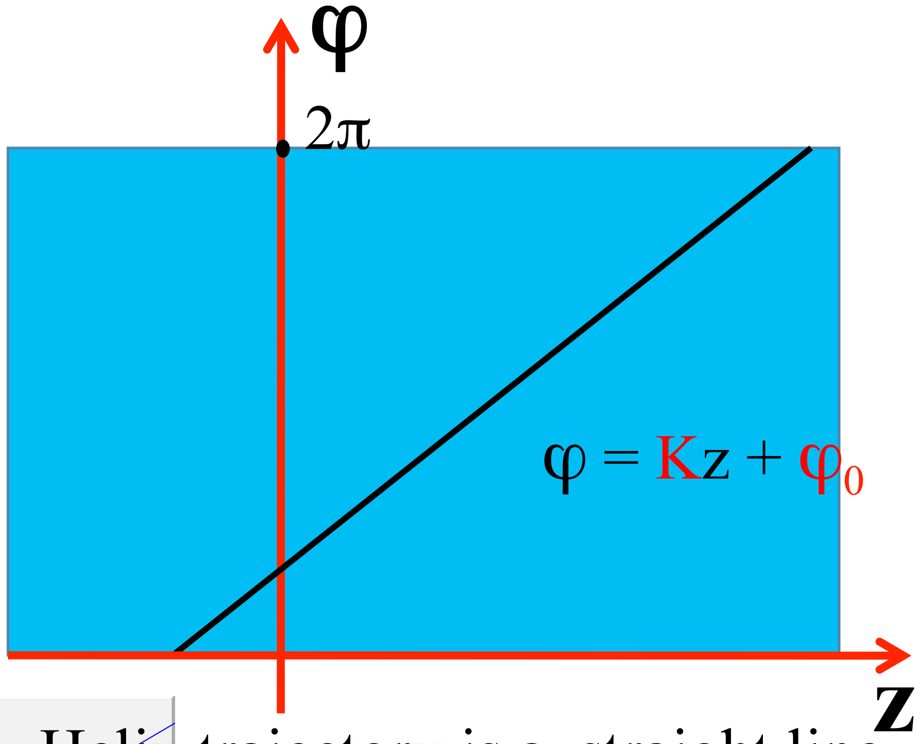
φ_0 ≡ azimuthal angle at $z = 0$

R ≡ radius of cylinder

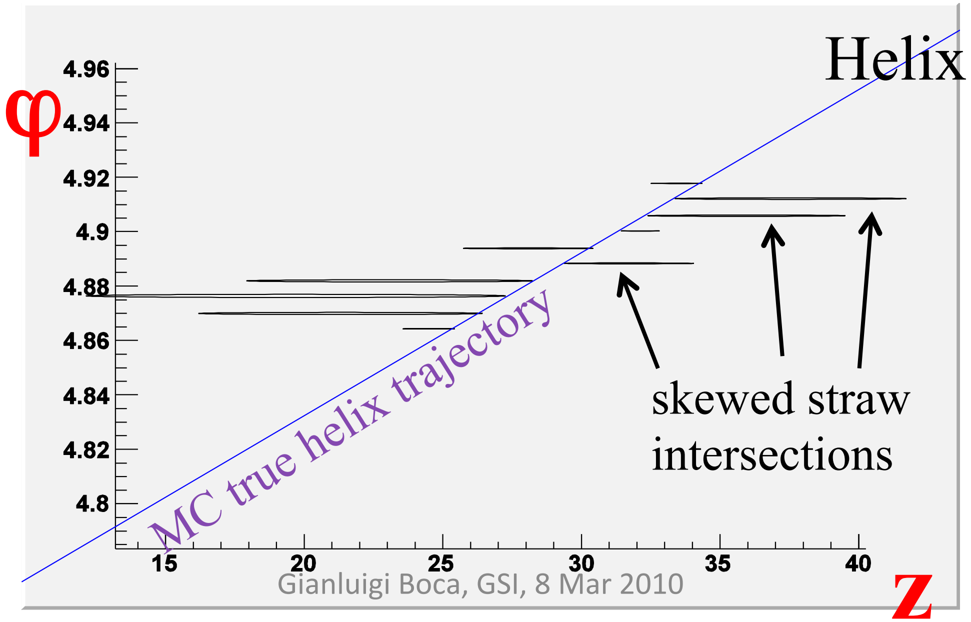
K ≡ rate of increase of φ

The φ z projection





Helix trajectory is a straight line



The pileup problem at 20 MHz Interaction Rate

The problem arises because the STT drift time can be as long as 200 ns.
Hits from previous events or subsequent events in STT can mix to the track true hits generating fake tracks.

2 MHz interaction rate at the beginning → essentially no overlap of different event hits

(in perspective at 20 MHz interaction rate) → up to 8 different event hits can pileup

At the end PANDA willl run at 20 MHz IR, the PR has to deal with many ghost tracks !

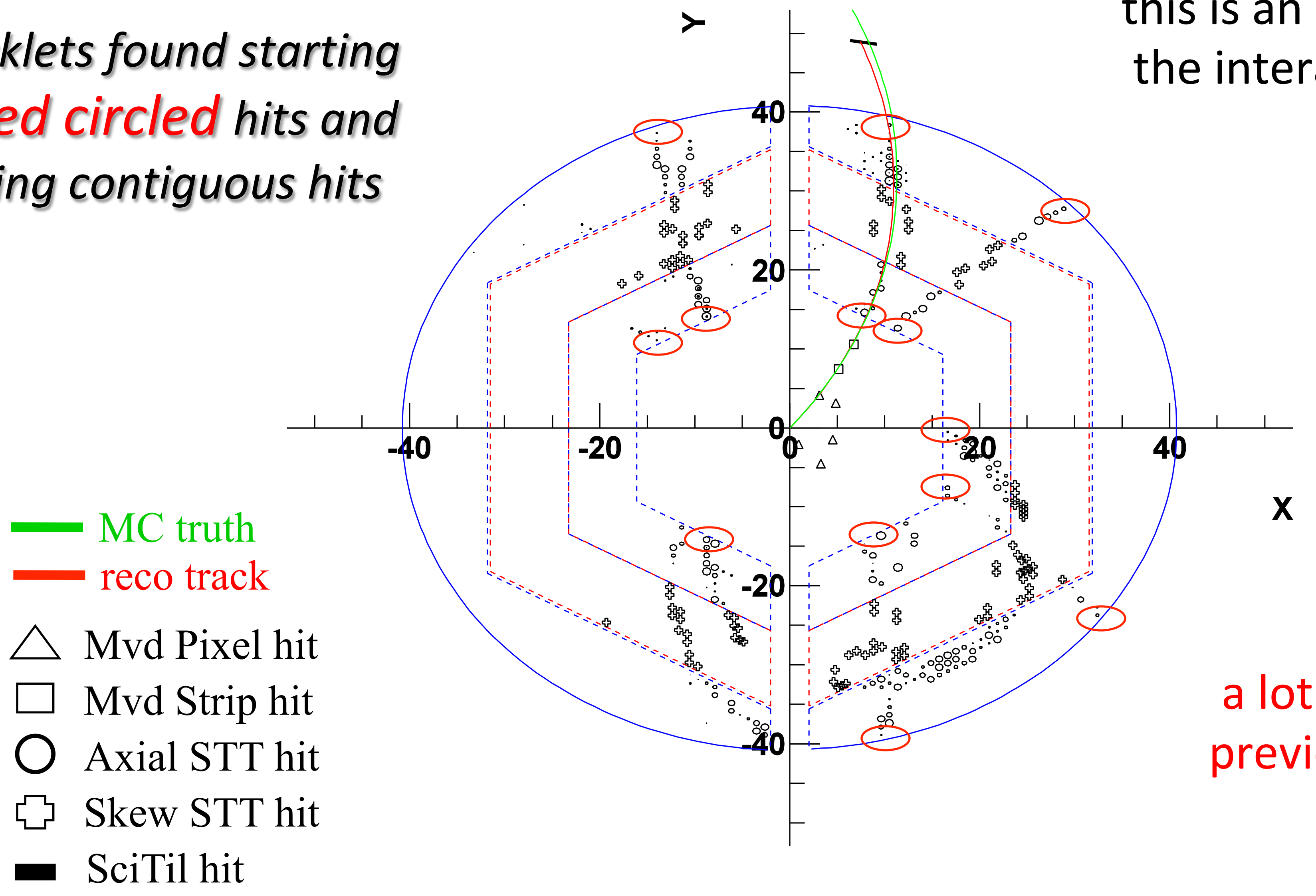
The brief description of the PR algorithm

The Pattern Recognition algorithm

- *Road finding method using initial hit cluster to find tracklets first and track candidates later*
- *The parameter fit of the candidate tracks are calculated with a Hough Transform method first (it is fast) and a* χ^2 *minimization later (it is more precise)*
- *This Pattern Recognition uses Mvd and Stt hits in central region (an extension in the forward region including Gem as well is also available in a separate package);*
- *track candidates found first in XY view and then in* φZ *view;*
- *tracklets found starting from* seed hits *at the boundary of axial Stt layers*
- *many fake tracks remain in the 20 MHz interaction rate. A final Cleanup procedure that eliminates the fakes is necessary.*

Road Finding : gathering STT hits in tracklets (in XY view)

- *tracklets found starting from* ***red circled*** *hits and collecting contiguous hits*

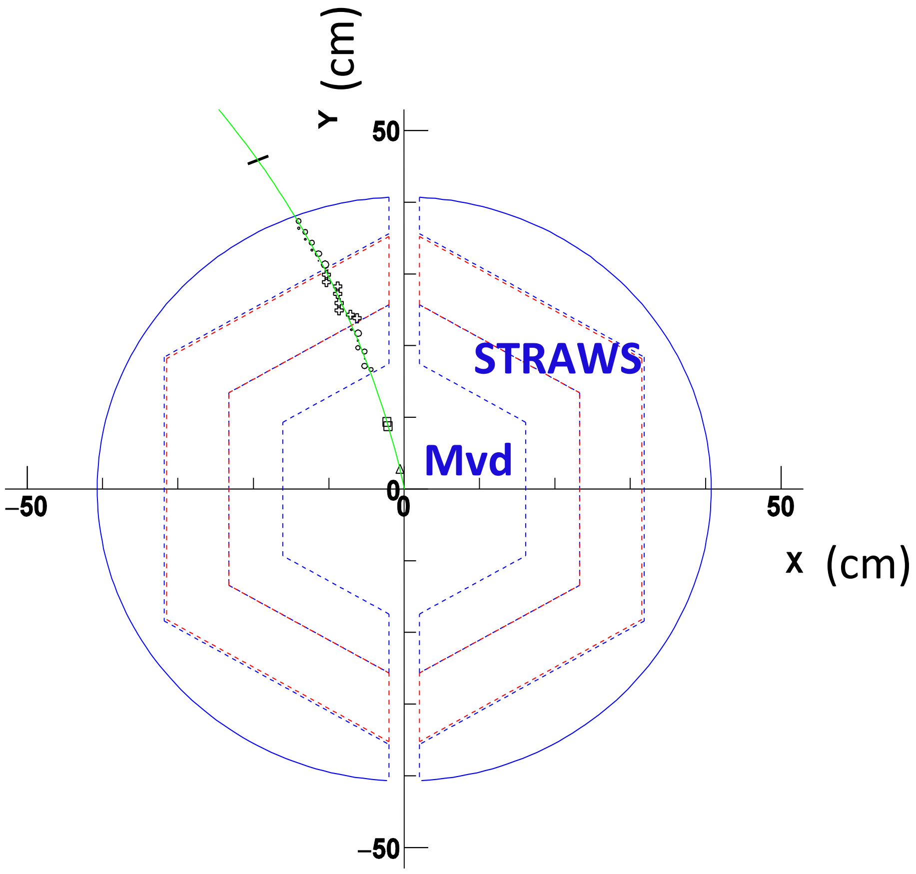


this is an event in the case when the interaction rate is 20 MHz.

a lot of pile up hits from previous and subsequent events !

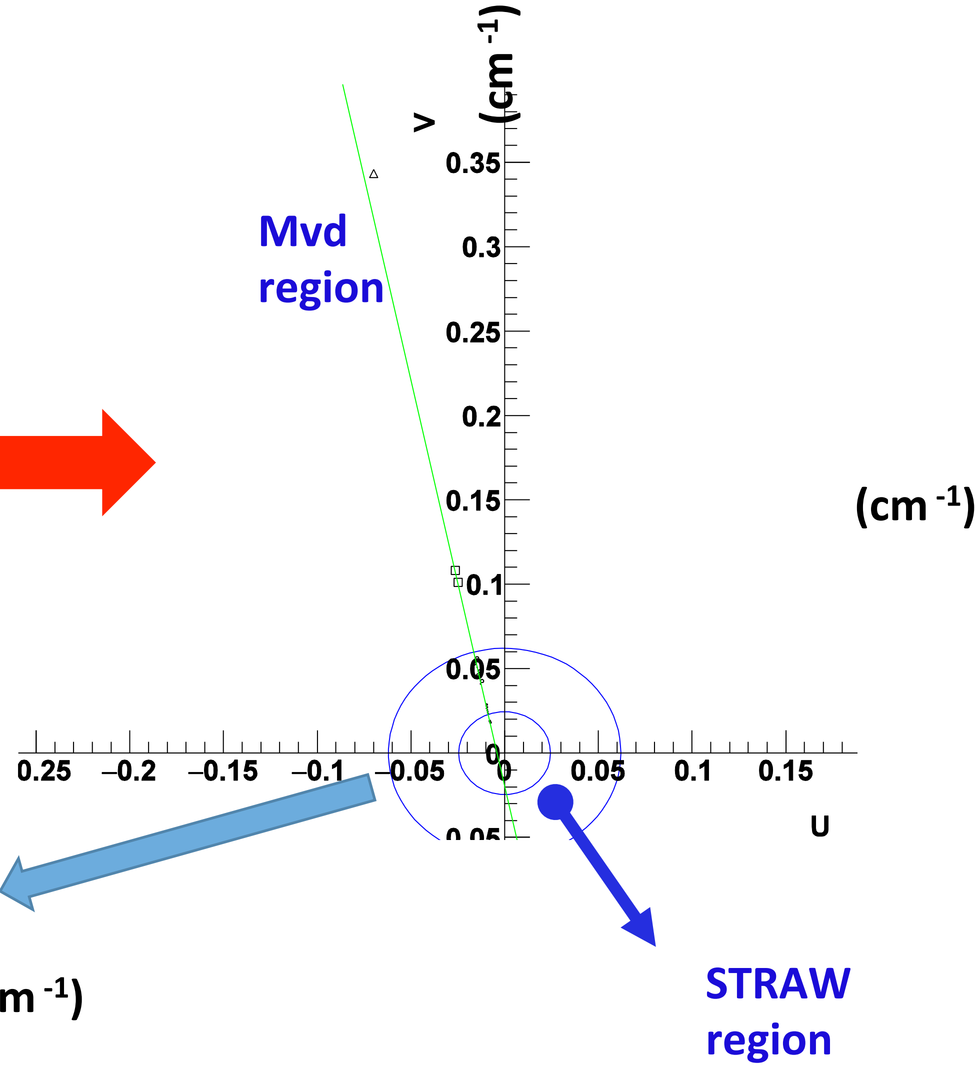
Fitting center and radius of trajectory in XY plane ($\equiv P_t$ determination)

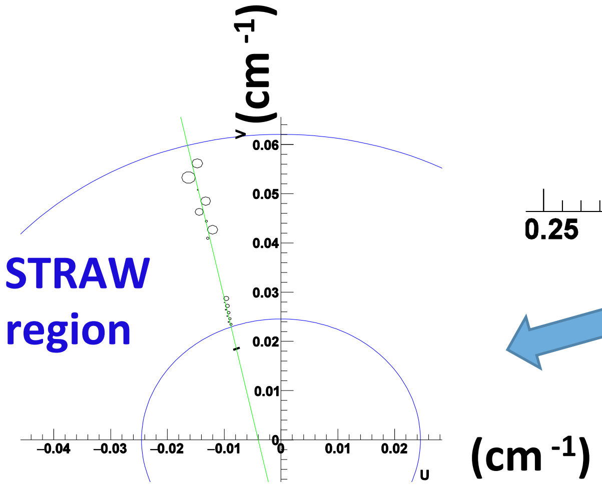
- *use Conformal transformation …………*



$$u \equiv \frac{x}{x^2+y^2}$$

$$v \equiv \frac{y}{x^2+y^2}$$



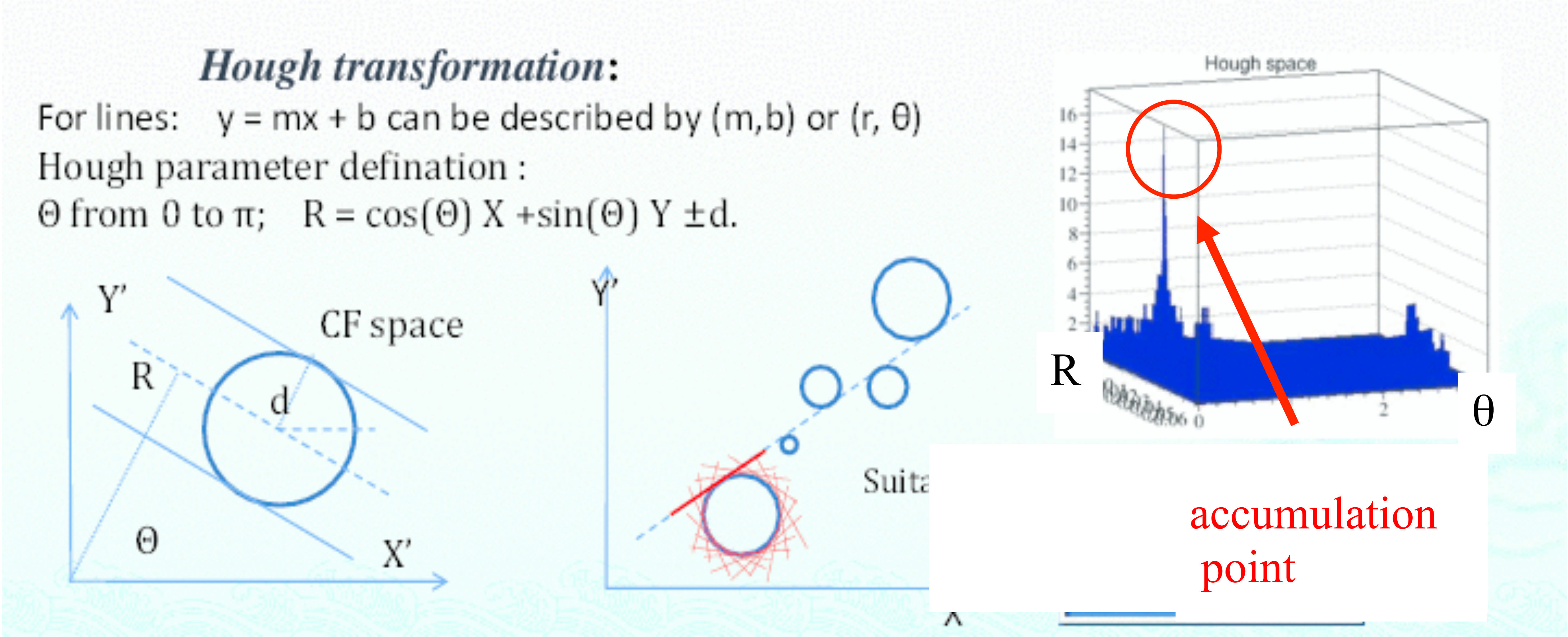


Green line = MC truth Track

Fitting center and radius of trajectory in XY plane

……… + *Hough Transform for first* *fast calculation* *of the Radius and center of the Helix trajectory using only the STT hits belonging to a tracklet.*

Hough transform in this case is performed only with STT hits belonging to the tracklet → Hough plot cleaner, peak stends out more clearly !



Fitting better center and radius of trajectory in XY plane

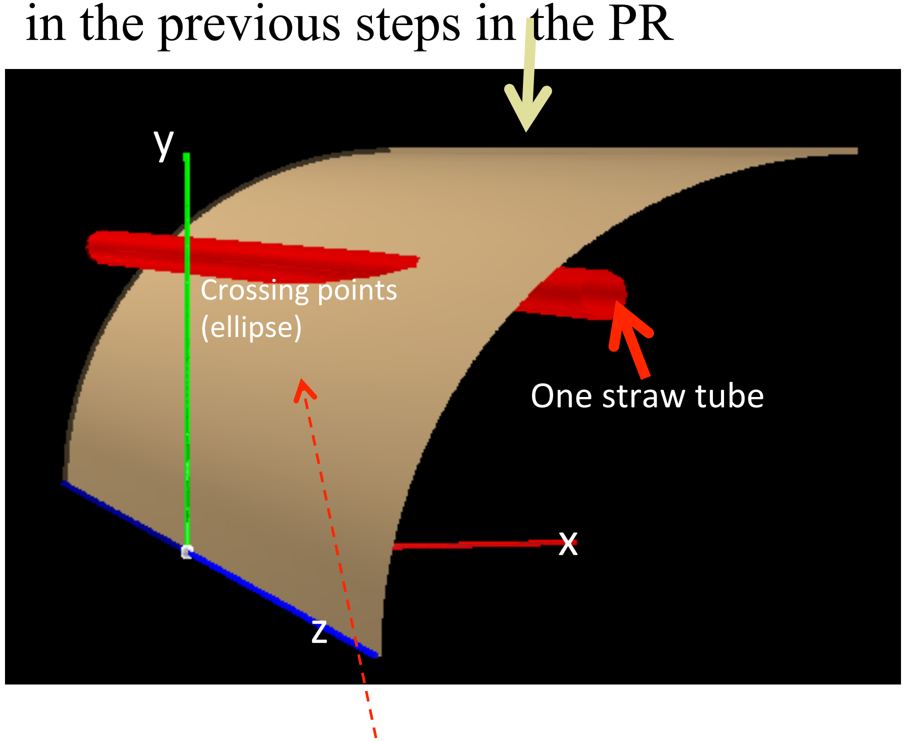
- *discard obviously unphysical tracks (radius too big or too small)*
- *work again in XY plane and using the radius and center of the trajector collect all possible hits close to the track, including MVD hits , and eliminate those far away.*
- *in XY plane determine again R and center of Helix of track trajectory but this time using also the MVD hits and with a χ^2 minimization. This leads to a more precise determination of R and the center. The MVD hits lying too far from the found trajectory are eliminated.*
- *use newly found trajectory to eliminate possible spurious hits still remaining in the track and associate SciTil hits to the track.*
- *determine the charge of the track.*
- *use found trajectory in XY plane to associate Skew STT hits to the track.*

Finding K (≡P_z determination) using the φz space

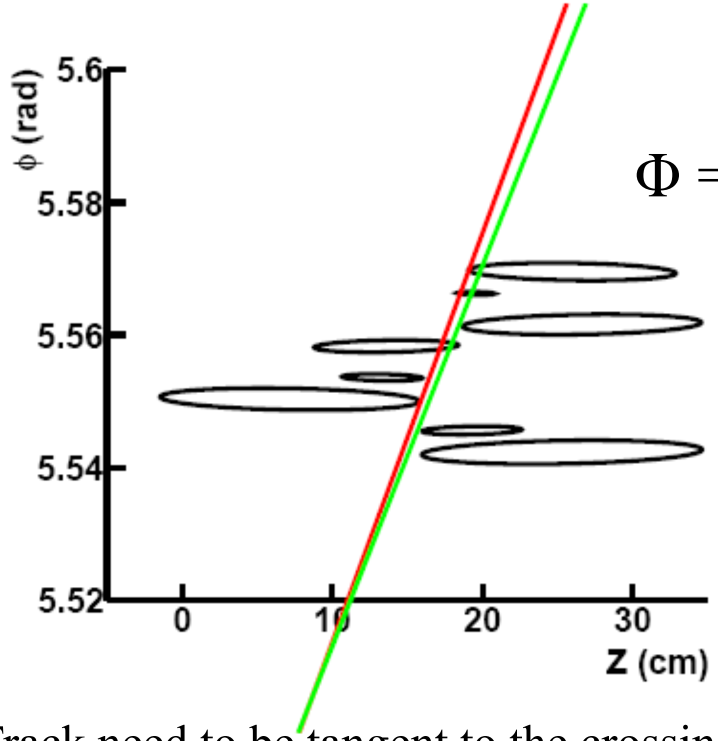
P_z determination : in the φ Z plane the Helix trajectory is a STRAIGHT LINE

1: The radius, the position of the helix center in the XY plane are determined.

Cylinder on which the helix lies; determined in the previous steps in the PR



Skew straw hits as they look in the ΦZ plane



$$\Phi = kZ + \Phi_0$$

Track need to be tangent to the crossing ellipse.

Finding K ($\equiv P_z$ determination) using the φz space

- *for each track candidate found in the previous steps, associate all hits of those Skew STT straws that cross the trajectory cylinder .*

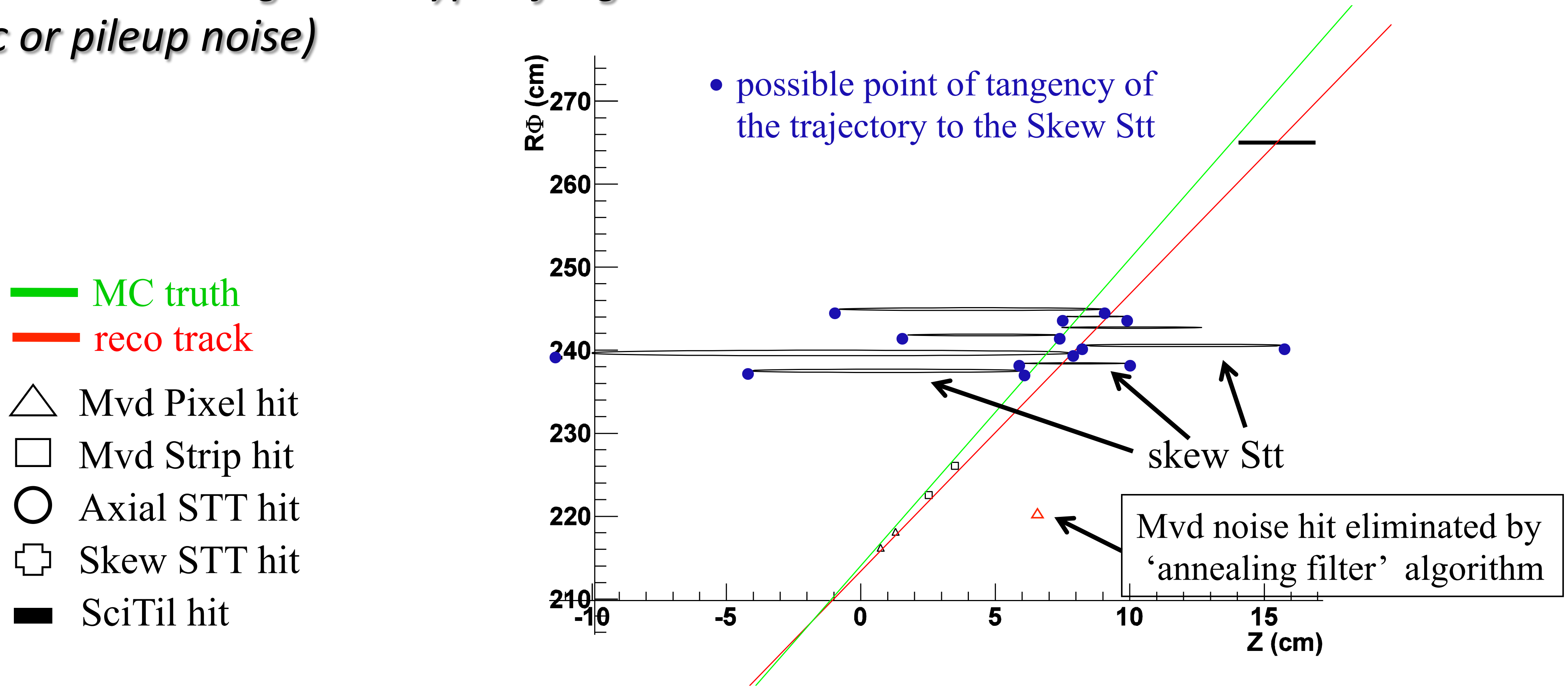
- *in the ϕz plane fit K the first time using only the MVD hits and the vertex constraint.*

- *Eliminate possible MVD hits too far away in the the ϕz plane from fitted trajectory.*

- *With the survived MVD hits and the Skew STT hits perform the final fit in the ϕz plane and determine K of the track.*

Finding K ($\equiv P_z$ determination) using the φz space

- *perform a χ^2 fit of track in ΦZ space allowing all possible point of tangency (blue points) and using an 'Annealing Filter' type of algorithm to esclude at most 1 noise MVD hit (electronic or pileup noise)*



- *further reject spurious using the result of fit and a proximity cut*

The Cleanup procedure

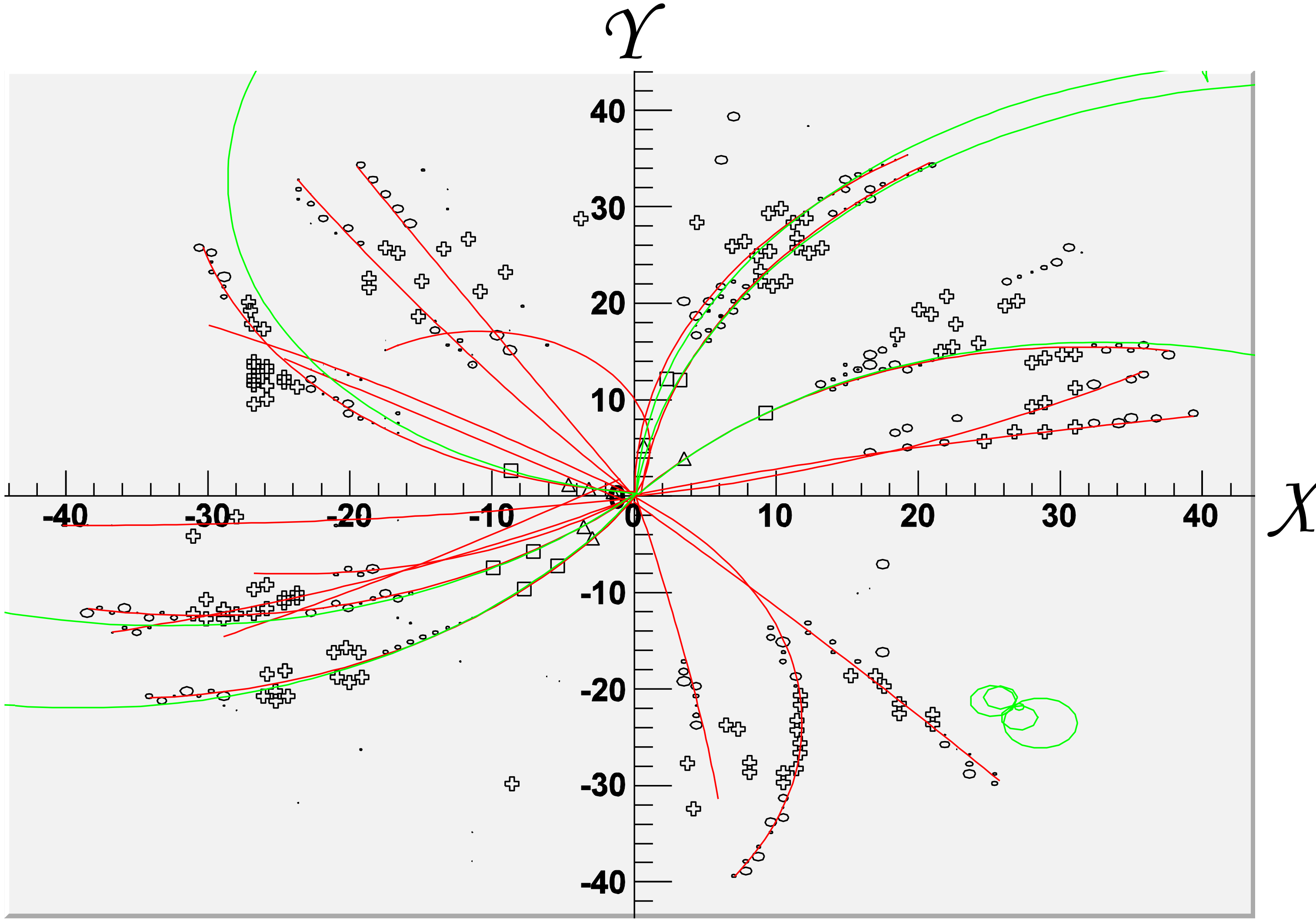
- *A 'Cleanup' procedure is necessary to reject fake tracks caused by pileup of events at 20 MHz interaction rate; the rejection is based on continuity of hits requirement*

Example in the case
of 20 MHz interaction rate
in the XY view

an event BEFORE
cleanup

Green line = MC truth

Red line = tracks found by PR



The Cleanup procedure

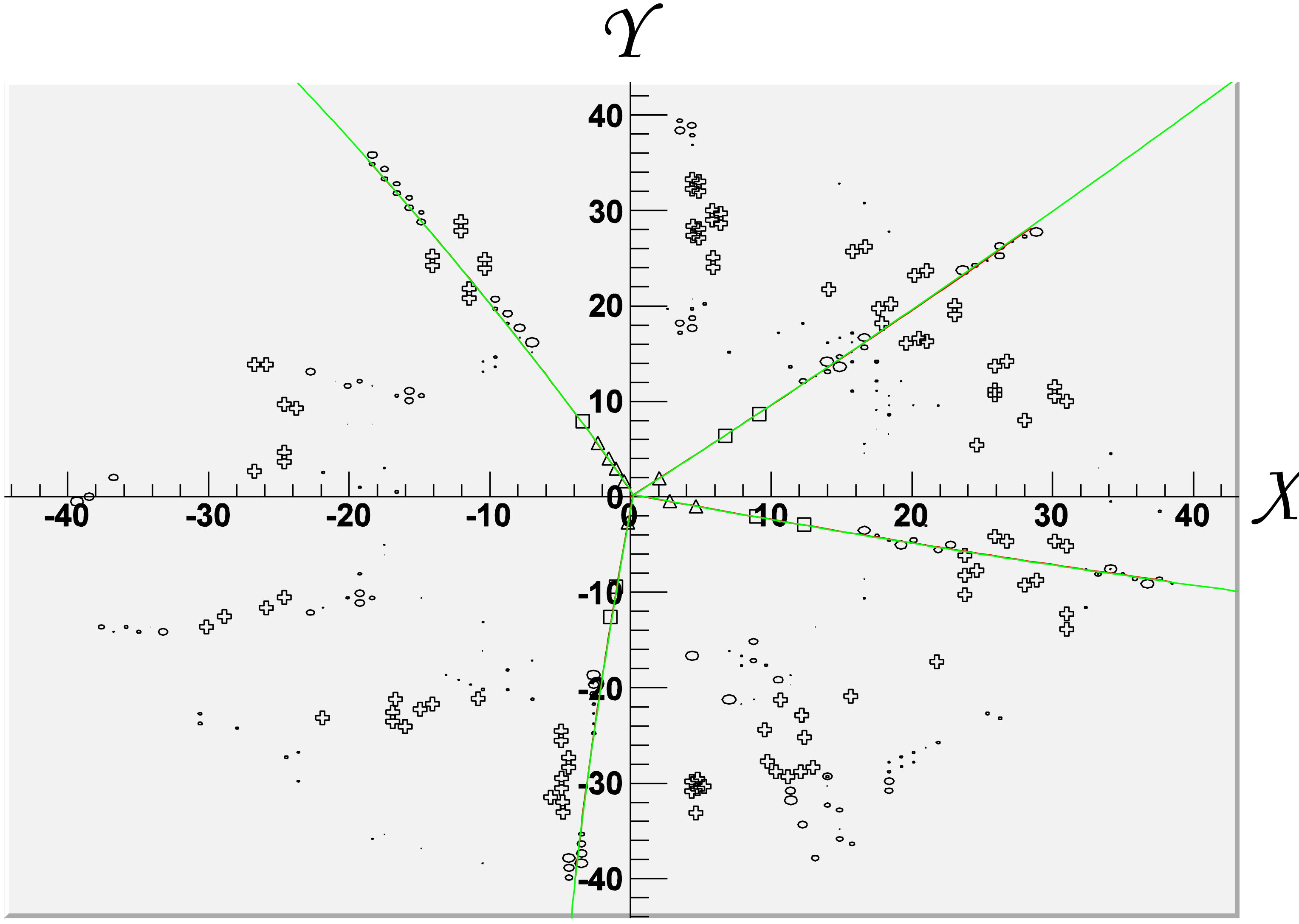
- *A 'Cleanup' procedure is necessary to reject fake tracks caused by pileup of events at 20 MHz interaction rate; the rejection is based on continuity of hits requirement*

Example in the case
of 20 MHz interaction rate
in the XY view

an event AFTER
cleanup

Green line = MC truth

Red line = tracks found by PR



The performance of the PR code

Performance of the code

- *The efficiency, purity, fake tracks rate and CPU time consumption are presented for an initial 2 MHz scenario (= essentially no pileup) and a 20 MHz interaction rate (strong pileup).*

- *Efficiency ≡ number of true (according to MC truth) tracks reconstructed over the total generated (the criterion used to declare a reconstructed track 'true' has been discussed several times in PANDA and here I don't go into details on that).*

- *Purity ≡ number of true (according to MC truth) hits in a true reconstructed track over the total number of hits in the reconstructed track.*

- *Fake track ≡ reconstructed track NOT associated to any true (according to MC) track.*

Performance of the code, case with no pileup (2 MHz IR)

MonteCarlo, Box generator, μ tracks

P (GeV/c)	Tracks/evt	Total tracks generated	% rec. tracks	# ghost tracks	# ghost/evt
0.3	1	3776	98	0	0.0
0.3	4	3797	94	0	0.0
0.3	8	3762	90	0	0.0
1.0	1	3762	98	0	0.0
1.0	4	3724	95	2	0.0
1.0	8	3723	92	6	0.0
2.0	1	3766	98	1	0.0
2.0	4	3730	95	2	0.0
2.0	8	3736	91	3	0.0
5.0	1	3750	98	0	0.0
5.0	4	3732	94	0	0.0
10.0	1	3735	98	0	0.0

generated tracks per evt

total generated tracks (ie summed over all events)

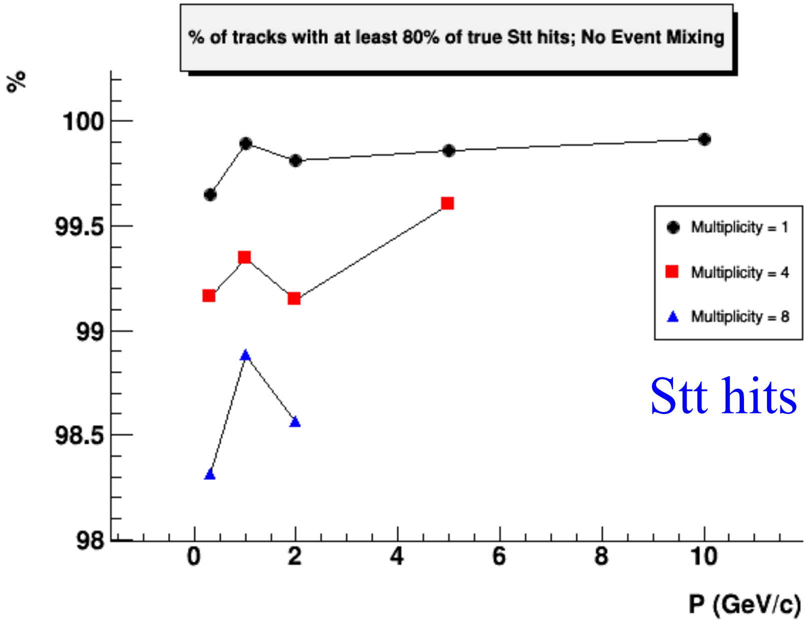
total fake tracks (ie summed over all events)

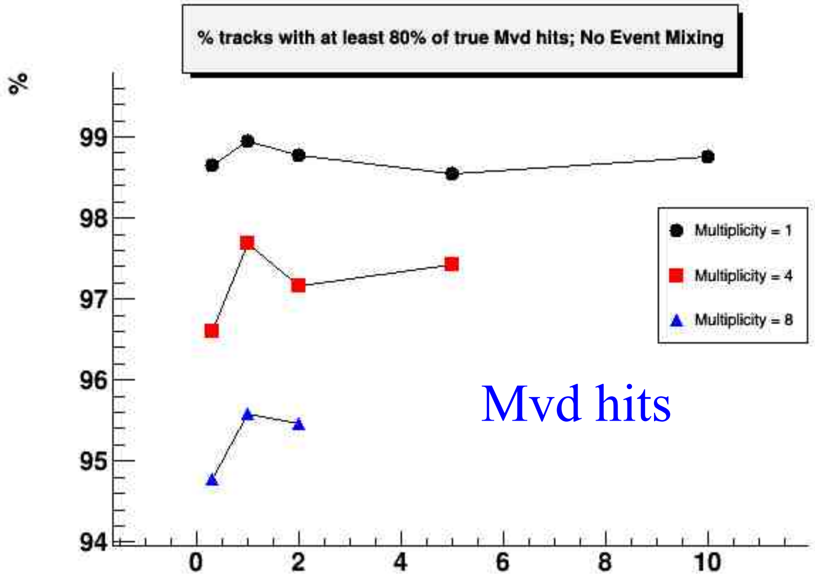
average fake tracks per event

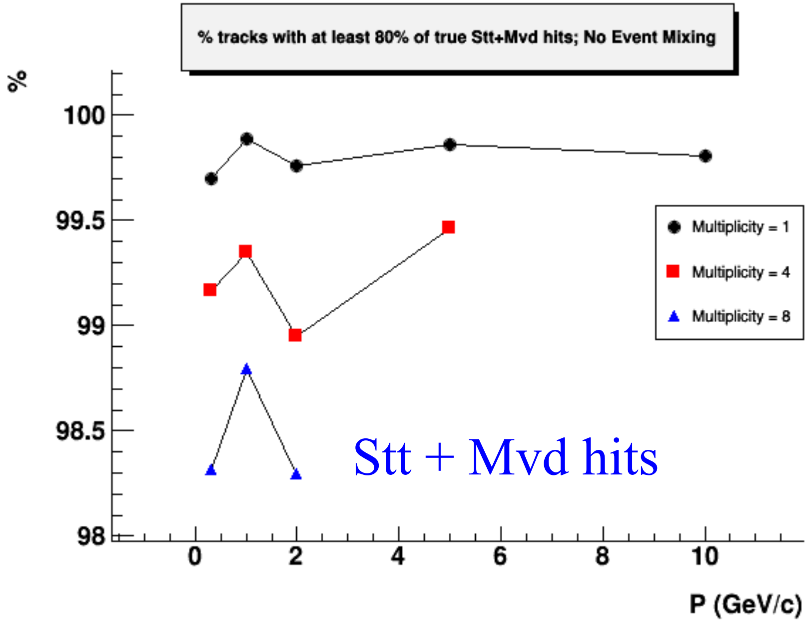
P of each track

P (GeV/c)	Tracks/evt	Total tracks generated	% rec. tracks	# ghost tracks	# ghost/evt
0.3	1	3776	98	0	0.0
0.3	4	3797	94	0	0.0
0.3	8	3762	90	0	0.0
1.0	1	3762	98	0	0.0
1.0	4	3724	95	2	0.0
1.0	8	3723	92	6	0.0
2.0	1	3766	98	1	0.0
2.0	4	3730	95	2	0.0
2.0	8	3736	91	3	0.0
5.0	1	3750	98	0	0.0
5.0	4	3732	94	0	0.0
10.0	1	3735	98	0	0.0

Performance of the code, case with no pileup (2 MHz IR)







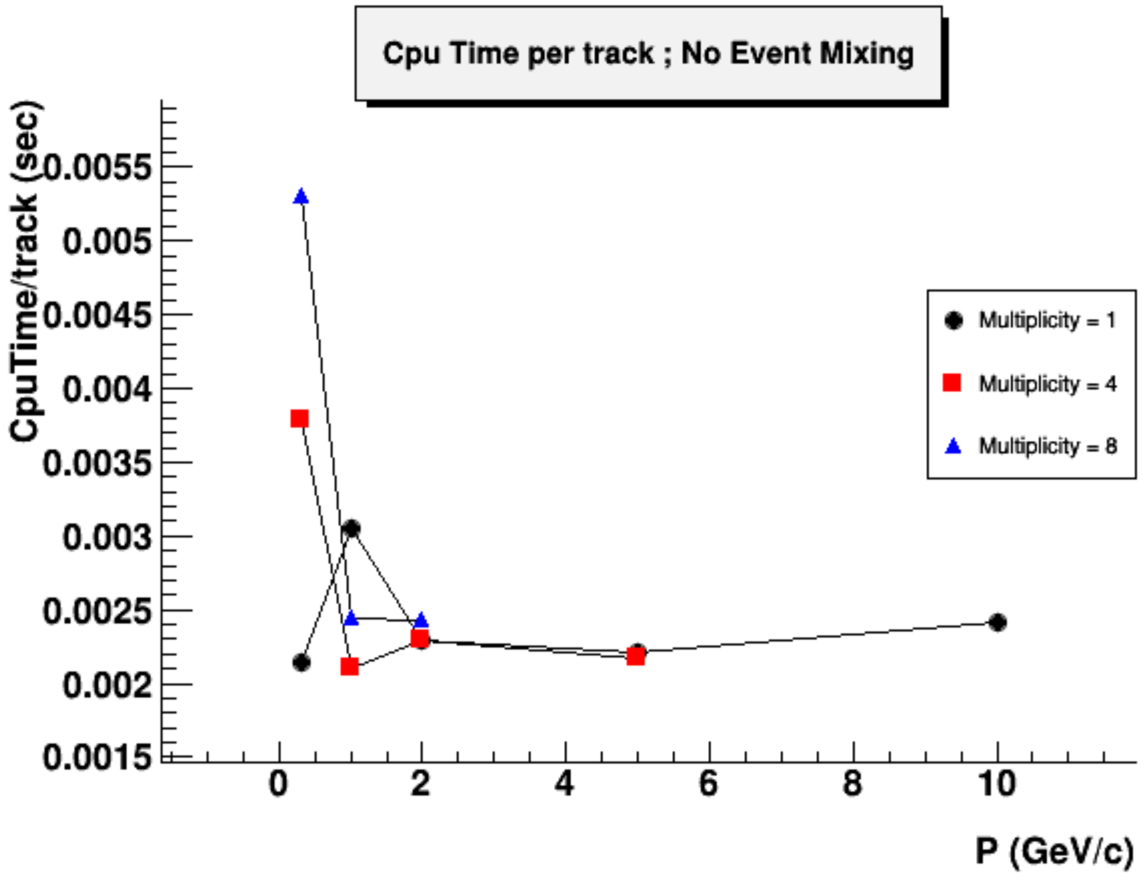
Here '% of tracks with at least 80% of the true ...' means : this is the percentage of TRUE reconstructed tracks having at least 80% of the hits truely belonging to the track (according to montecarlo).

G.Boca U. Pavia & INFN, Italy

Performance of the code, case with no pileup (2 MHz IR)

Cpu time consumption/track

Processor : Intel Core i7-2600K CPU @ 3.40 GHz

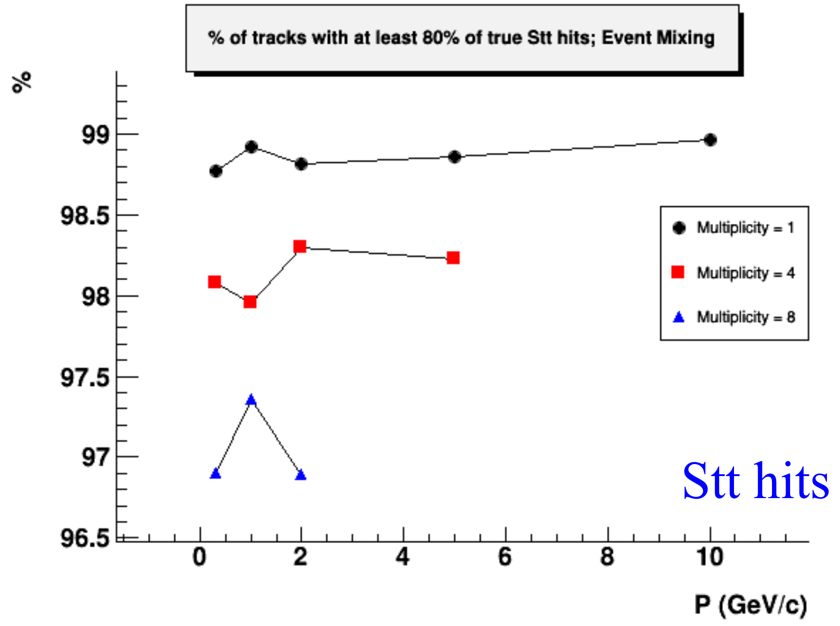


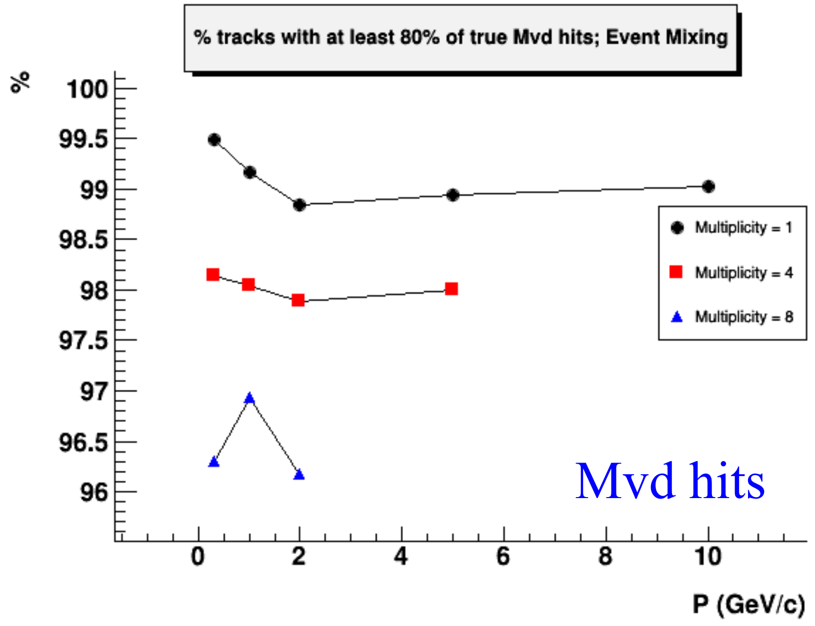
Performance of the code, case with strong pileup (20 MHz IR)

MC, Box generator, μ tracks

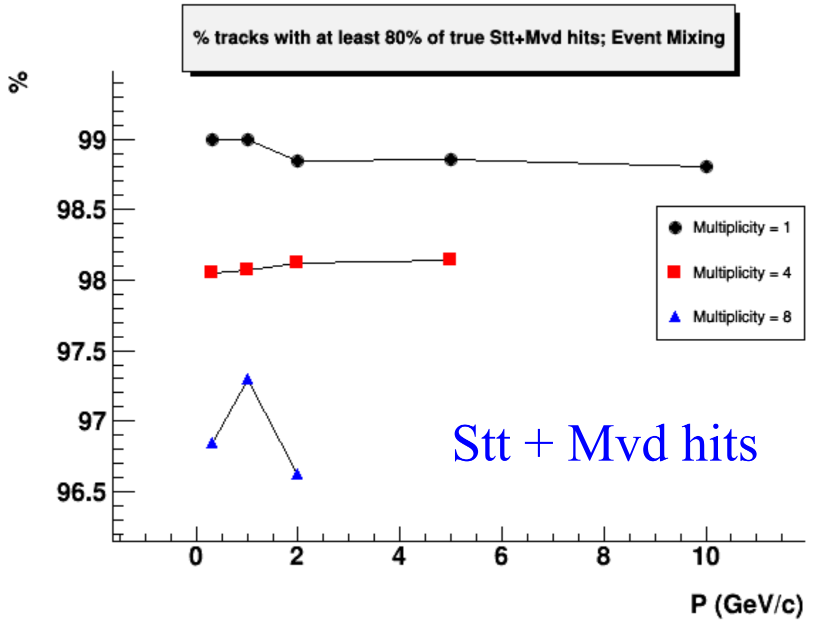
P (GeV/c)	Tracks/evt	Total tracks generated	% rec. tracks	# ghost tracks	# ghost/evt
0.3	1	3776	95	491	0.1
0.3	4	3797	92	361	0.3
0.3	8	3762	88	196	0.4
1.0	1	3762	96	499	0.1
1.0	4	3724	93	287	0.3
1.0	8	3723	89	208	0.4
2.0	1	3766	96	463	0.1
2.0	4	3730	93	279	0.3
2.0	8	3736	90	195	0.4
5.0	1	3750	96	497	0.1
5.0	4	3732	92	273	0.3
10.0	1	3735	96	484	0.1

Performance of the code, case with strong pileup (20 MHz IR)





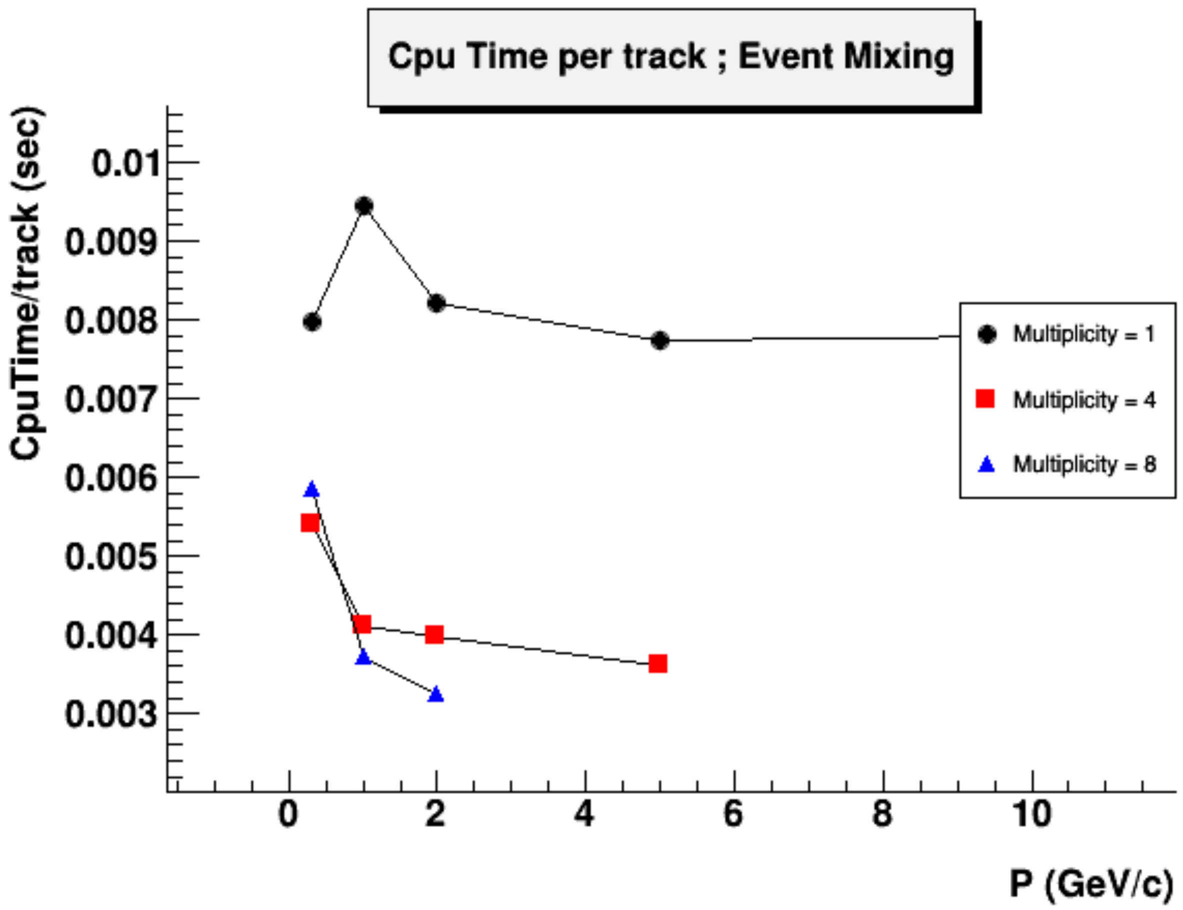
Track Quality Check



Performance of the code, case with strong pileup (20 MHz IR)

Cpu time consumption/track

Processor : Intel Core i7-2600K CPU @ 3.40 GHz



THANK YOU VERY MUCH
FOR
YOUR ATTENTION !