Subject: PndLhePidTrack
Posted by donghee on Wed, 31 Mar 2010 07:59:57 GMT
View Forum Message <> Reply to Message

Dear colleagues,

I have generated the electron in the 90 < theta < 100 (degree) range
with 2.5 GeV to 3.5 GeV momentum.
Total 945 events are reconstruced with using PndLhePidTrack class from 1000 events. TPC and MVD are simply working. and GEM is not
interested in this kinematic range.
And then Pid Candidate has been compared with MC true.
Pandaroot version is slightly old as the v.7878, that's mean some
bug, which are reported during last month(March), are still within there. However the usage of PndLhePidTrack seems to be ok at reco task!

But in the final check, resolutions are too bad and simply wrong with PndLhePidTrack approach. Momentum resolution is about 20~30%, that is not usual because it must be below 5%.

I'm wondering that PndLhePidTrack is not meaningful in this time.
or is the problem related with sets of bugs with MVDcluster and etc..., that was posted in last few weeks?

So you can see resolutions for momentum, theta and phi.
Thanks.

File Attachments
1) res_all_2.eps, downloaded 580 times

Subject: Re: PndLhePidTrack
Posted by StefanoSpataro on Wed, 31 Mar 2010 08:13:15 GMT
View Forum Message <> Reply to Message

Hi,
please try the macros in macro/pid.
PndLhePidTrack is not used anymore since last year and is obsolete, the correlation is done inside PndPidCorrelator. In macro/pid you can find also the macros to check momentum resolution.

Subject: Re: PndLhePidTrack
Posted by donghee on Wed, 31 Mar 2010 09:47:47 GMT

View Forum Message <> Reply to Message

Thanks Stepano,

So, first of all I have to update pandaroot.
Then, PndPidCorrelator have to be used to correct my specta.
tutorial/charmonium/* and macro/pid/* should be helpful.
In addition, nice implementation of PndRecoKalman can also be used.
That is now quite clear.

But why do PndLhePidTrack show such kind of strange information, if it is already obsolute in the pandaroot?
I guess that some hits from both mvd and tpc were not used anymore in this class, or even the coordinate or some name of hit accessor have been changed internally.

The understanding for the reason is also important to me.
Do you have a rough explanation or simple diagnostic for that?

Thanks.

Subject: Re: PndLhePidTrack
Posted by StefanoSpataro on Wed, 31 Mar 2010 11:41:04 GMT
View Forum Message <> Reply to Message

PndLhePidTrack takes momentum from helix assumption, before kalman, and it had bad resolution for high momentum values.

In macro/pid the pidcorrelator takes momentum after kalman, then the resolution should be much better. Moreover, LhePidTrack was not developed anymore and probably it was not syncronised with the rest of the code.

Subject: Re: PndLhePidTrack
Posted by donghee on Wed, 14 Apr 2010 22:14:31 GMT
View Forum Message <> Reply to Message

Dear Stepano,

I have made small test with the electron in the 90 < theta < 100 (degree) range and 0.5 GeV to 1.5 GeV momentum.
PndPidCorrelator have been used and compared with generated one.

Different interaction region z=0 and z=5cm are tested and the results looks reasonable.
If interaction point is moved to z=5cm, 92-94 degree could not recontructed well, becuase of positioning of taget pipe and absent of barrel layer of MVD.

But I do not understand clearly on the general concept of tracking procedure with shifted interaction region.

As far as I understand, hits of every detectors are only important for the tracking.
If final vector components are pointed back to vertex position and the tracking is surely correct, then origin of tracks can correctly find, even though real vertex postion is moved to few cm. So, in some sense the tracking is independent from vertexing.

Is this statement also valid in LHEtracking task?
or do I have to consider some assumption of vertex position to z(0,0,0) in the tracking, specially in LHE?

Thanks,
Donghee

File Attachments
1) LHE_tracking_z_00.eps, downloaded 518 times
2) LHE_tracking_z_05.eps, downloaded 517 times

Subject: Re: PndLhePidTrack
Posted by StefanoSpataro on Thu, 15 Apr 2010 05:42:40 GMT
View Forum Message <> Reply to Message

Hi,
the statement should be correct, lhe should not depend on the vertex position. But efficiency/resolution studies shifting the interaction point were never performed.

Subject: Re: PndLhePidTrack
Posted by Tobias Stockmanns on Thu, 15 Apr 2010 06:24:26 GMT
View Forum Message <> Reply to Message

Hi Stefano and Donghee,

what about track finding? I thought that conformal mapping relies on tracks coming from the primary vertex.

Cheers,

Tobias

Subject: Re: PndLhePidTrack
Posted by StefanoSpataro on Thu, 15 Apr 2010 06:57:43 GMT
View Forum Message <> Reply to Message

In theory it should not, but this has to be checked by somebody with data.

Subject: Re: PndLhePidTrack
Posted by Gianluigi Boca on Thu, 15 Apr 2010 11:03:57 GMT
View Forum Message <> Reply to Message

Tobias Stockmanns wrote on Thu, 15 April 2010 08:24Hi Stefano and Donghee,

what about track finding? I thought that conformal mapping relies on tracks coming from the primary vertex.

Cheers,

Tobias

In the STT pattern recognition indeed there is the assumption that tracks come from 0,0,0 . This assumption is crucial for the algorithm to work. Indeed it works well as long as the interaction vertex is far from 0,0,0 a few mm. When the vertex is farther than 1 cm then problems arise (loss of efficiency and spurious hits inclusion).

Consequently. based on my experience, I would be (very) surprised if any pattern recognition algorithm using the conformal mapping in the XY plane, worked when the primary vertex is far 5 cm from the expected position.

That is also the reason why I have always claimed that as far as the Lambda is concerned we need to write a special pattern recognition package.

Gianluigi

Subject: Re: PndLhePidTrack
Posted by donghee on Sat, 17 Apr 2010 12:52:18 GMT
View Forum Message <> Reply to Message

Dear Gianluigi Boca,

If your statement is correct, then every secondary vertex like K0, lambda, which have some decay length more than few cm, cannot reconstructed any more. That is quite pity, but I

assume that is not implemented right now because of some ordering of priority in global tracking.
So, current tracking package is now only optimized at primary vertex.

You mentioned only about STT.
I am wondering how about rest of trackers.
What is the situation for stand alone track finding with TPC, MVD, MDT, and GEM?
If one of those do not find any track due to strong assumption of z=0, the efficiency of lhetrack would be automatically drop down, whatever rest of part are correctly contributed to build a true trajectory.

Best regards,
Donghee

Subject: Re: PndLhePidTrack
Posted by Gianluigi Boca on Sun, 18 Apr 2010 15:51:09 GMT
View Forum Message <> Reply to Message

Donghee Kang wrote on Sat, 17 April 2010 14:52Dear Gianluigi Boca,

If your statement is correct, then every secondary vertex like K0, lambda, which have some decay length more than few cm, cannot reconstructed any more. That is quite pity, but I assume that is not implemented right now because of some ordering of priority in global tracking.
So, current tracking package is now only optimized at primary vertex.

Best regards,
Donghee

Yes, in my opinion, for the Vees it is necessary to write a package that come after the 'normal' pattern recognition. That is true
for the STT, and my feeling is that could be true also for other detectors.
Only those hits not used by the 'normal' pattern recognition would need to be taken into account at this point. These hits should be searched for finding tracks not coming directly from 0,0,0.

Gianluigi

Subject: Re: PndLhePidTrack
Posted by StefanoSpataro on Tue, 20 Apr 2010 08:58:46 GMT
View Forum Message <> Reply to Message

Hi,
I have done a check with lhetrack tpc+mvd, firing 1 GeV muons produced at (0,0,0) RED and

at (4,4,4) [cm] BLUE.

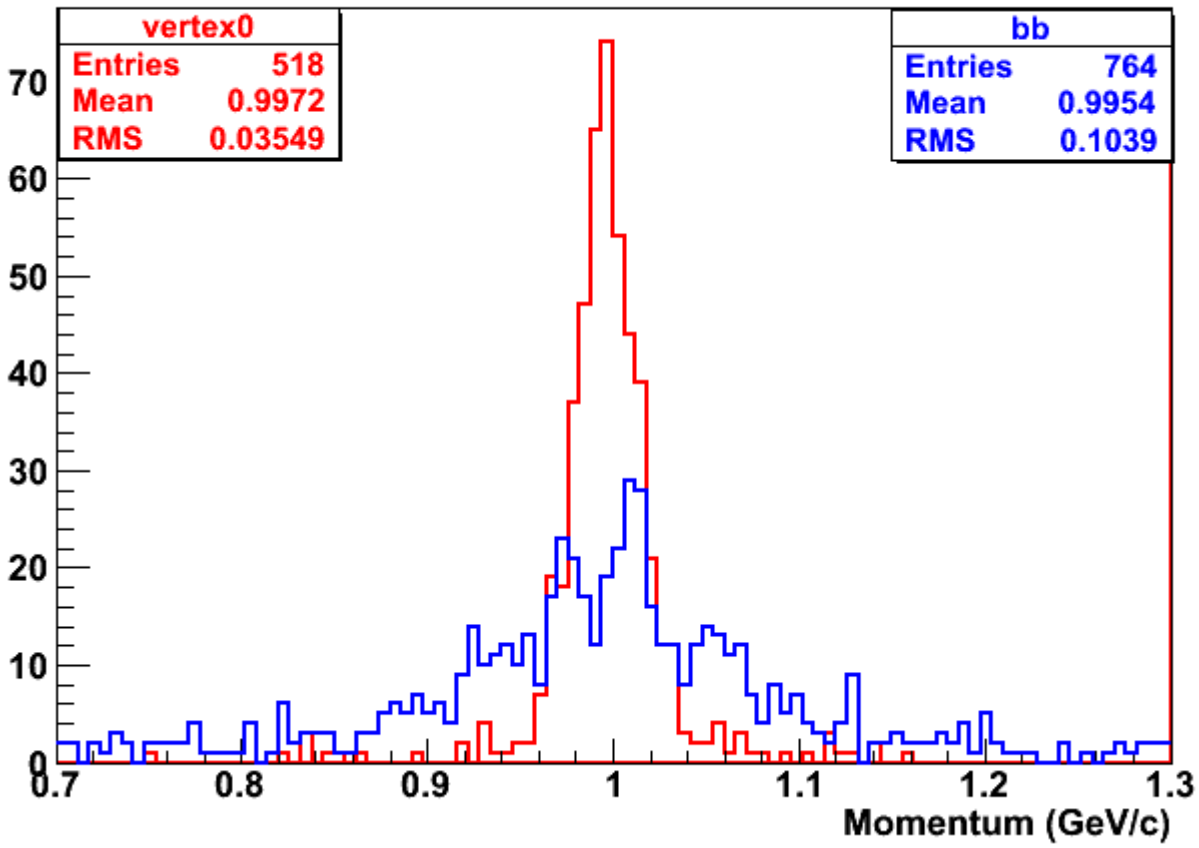
This is the momentum plot for LHE tracks:

and after genfit:

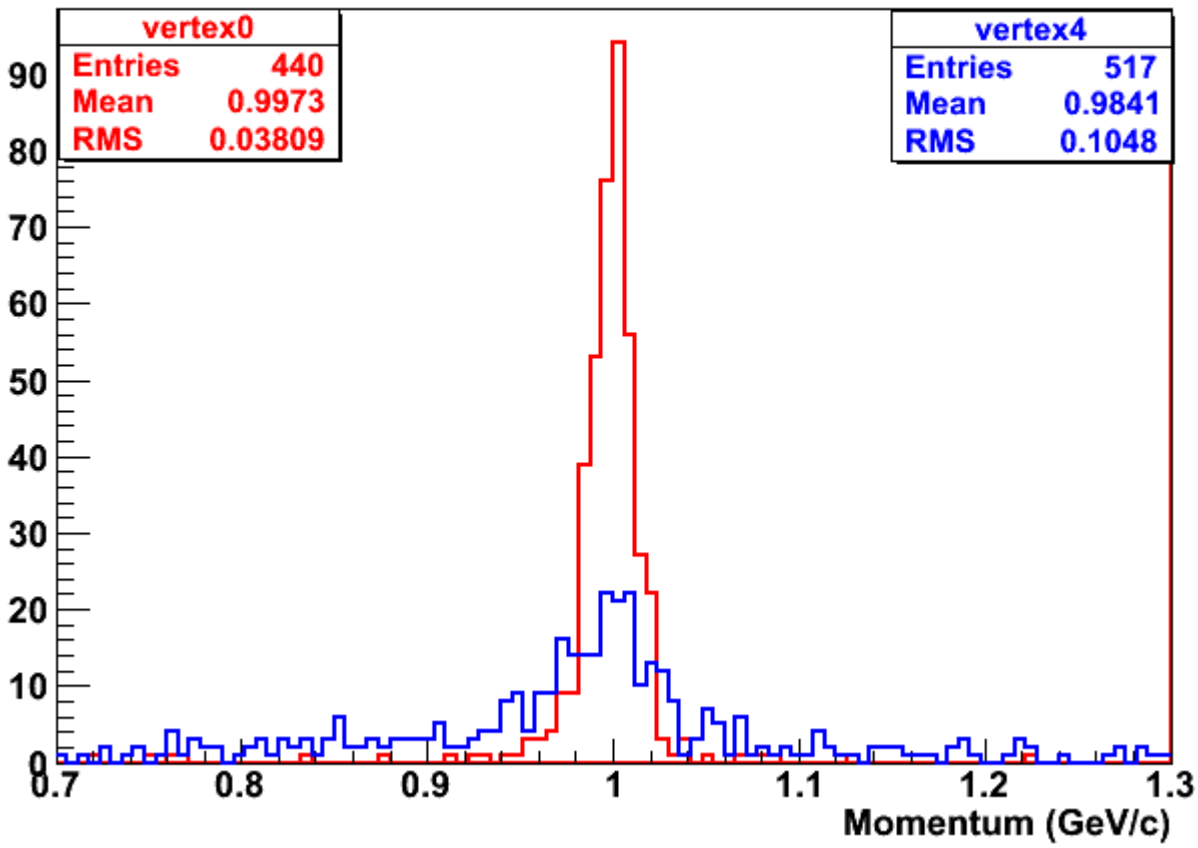
The loss in resolution is quite evident.

File Attachments

1) lhetrack_vertex_comp.gif, downloaded 1231 times



2) lhegentrack_vertex_comp.gif, downloaded 1382 times



Subject: Re: PndLhePidTrack
Posted by Anonymous Poster on Fri, 23 Apr 2010 09:59:21 GMT
View Forum Message <> Reply to Message

Hi Stefano,

the resolution of the fit can not depend on the difference in start vertex (0,0,0) vs (4,4,4). Could you please investigate and/or explain the following points:

- why are there 764 tracks in the distribution for 4,4,4 and much less in the 0,0,0 distro?
- what is the theta range you used?
- Could you please check whether the number of hits in the track is reduced in the 4,4,4 case? It could explain the worse resolution in the fit. Maybe your pattern reco can not use all hits in the case?

Cheers, Christian

Subject: Re: PndLhePidTrack
Posted by StefanoSpataro on Fri, 23 Apr 2010 10:33:53 GMT
View Forum Message <> Reply to Message

The problem should not be on the fitter, but on the finder and of the initial parameter guess.

Subject: Re: PndLhePidTrack
Posted by donghee on Fri, 23 Apr 2010 11:55:14 GMT
View Forum Message <> Reply to Message

Dear Stepano and Christian,

Many of us is now pointing and understanding that the pattern recognition of track finding is strongly depends on the initial vertex position. If the primary vertex is moved to some way around from 0,0,0, then each detector part can not effectively find one of track.
Consequently, resolution would be drop down in the pid of lhetracking.
I cannot understand also, why are there 764 tracks in the distribution for 4,4,4 and much less in the 0,0,0 distro.
In my point of view tracking efficiency of (4,4,4) must be worser than (0,0,0).

I guess that it is not only related LHE tracking, but also the tracking of each single piece of every detector are strongly connected. One cannot simply introduce and modify something like an option for different position of interaction point for track finding in the lhetracking.
A single track from each detector must to be found with moved vertex at first, then global hits can be defined in the lhetracking efficiently. That's mean that one have to modify every detector part before building one track in lhetracking.
I think that is quite huge stuff.

My question is whether one can handle the interaction point directly in lhetracking or have to have some improvement of track finidng from every detector piece with moved point at first.

Best wishes,
Donghee

Subject: Re: PndLhePidTrack
Posted by donghee on Mon, 26 Apr 2010 15:05:49 GMT
View Forum Message <> Reply to Message

Dear Panda User,

I'm trying to understand how the LHE tracking works in very general and naive(?) view.

Hits of every detectors (MVD, TPC, GEM etc...) have been produced in MC+Digitization stage.

In LHE tracking, a hit maker use every hits from tpc, mvd, and gem to construct a global hit.
Upto this procedure, there is no assumption for interaction point 0,0,0 in my understanding.

Then the track finder try to build a track candidate with introducing track cuts, and a pattern recognition has been done in this stage.
And finally fitter perform track fitting for this candidate.

In principle, one can modify track finder to find shifted interaction point, if I correctly understand.

You don't need to care about full stuffs from all detector.

That's mean that one can directly extend tracking to secondary vertex and shifted vertex at lhetracking in very simple view.

I know that the real modification should be really really difficult.
If I'm wrongly understanding the lhetracking package, please correct me!

Thank you for your reading!
Donghee

Subject: Re: PndLhePidTrack
Posted by StefanoSpataro on Mon, 26 Apr 2010 16:09:25 GMT
View Forum Message <> Reply to Message

Donghee Kang wrote on Mon, 26 April 2010 17:05Then the track finder try to build a track candidate with introducing track cuts, and a pattern recognition has been done in this stage. And finally fitter perform track fitting for this candidate.

In principle, one can modify track finder to find shifted interaction point, if I correctly understand.
You don't need to care about full stuffs from all detector.

That's mean that one can directly extend tracking to secondary vertex and shifted vertex at lhetracking in very simple view.

In theory what you say is correct: by applying some modifications, it is possible to use conformal mapping also for secondary verteces. One should improve the LheTrackFinding, and also the LheTrackFitting.
But, in order to do this, we should dig inside the code and try to find all the "guilty" points.
Feel free to check the code and to suggest some implementations that could help us on this side. Unfortunately, inside the "lhe developer group" (myself) there it not so much manpower to perform this kind of study and improvement.

Subject: Re: PndLhePidTrack
Posted by donghee on Tue, 27 Apr 2010 08:29:26 GMT
View Forum Message <> Reply to Message

Dear Stepano,

I'm looking for the lhetrack.

If I correctly understand your suggestion, I need to start from track finder PndLheTrackFinder.cxx with conformal mapping point.

There is PndLheCMPoint.cxx which is a representation of Conforaml Mapping points, and PndLheCMCandidate.cxx is a class for the track with those CM points.
Probably, that is something like transformation from hit in lab frame to new conventional coordinate, which takes into account the shifted vertex.
3 functions (SetIntPoint(); -> SetShiftedCoord(); -> SetConfCoord() play mainly role in this stage.
If this process works properly, then the tracking should work also for secondary verteces.

I think that I don't need to modify any part of PndLheCMPoint.
Main modification have to be done only in PndLheTrackFinder.cxx.

Could you correct me, if I'm now going to wrong way!

Shall we move our discussion in tracking forum?
Best wishes,
Donghee

Subject: Re: PndLhePidTrack
Posted by StefanoSpataro on Tue, 27 Apr 2010 16:01:55 GMT
View Forum Message <> Reply to Message

I have not understood how you determine the vertex of the reaction. In theory you should know this position in order to do a transformation of the coordinates. But this depends on the particle (in the same event you can have particles coming from different vertices).
The point is that one should take out the dependence of the track origin in the finder code. I think also the fitting should be changed, because the point 0,0,0 is also used there, for the initial momentum value of the circular fitting routine.
