
Subject: plane id in track candidate

Posted by [Anonymous Poster](#) on Tue, 23 Feb 2010 09:23:17 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi,

I have stumbled upon an interesting problem in track fitting:

If the pattern recognition determines that two hits in the same plane (e.g. MVD) belong to the same track, because a clear distinction is impossible, they are both input to the track fit. Then maybe the fit can determine which is the outlier or simply it has to use both. The question is how to handle that situation, i.e. how does the Kalman filter (or another algorithm) know that two hits are on the same plane. We could compare the complete plane parameters of both planes, but that is a big computational effort and hardly seems necessary. That is why I propose the following:

We should add an (unsigned) integer is called planeId to all hits in the track candidate. This way any tracking algorithm can easily determine if the next hit is in the same plane or in another one that requires an extrapolation.

If there are no objections, I will add the corresponding things to the PndTrackCand. The long term id behind this is that we will in the near future work on a tracking algorithm called

noise: a study of elastic tracking and nonlinear filters, Computer Physics Communications 120 (1999) 197–214) which has a dynamic weighting of hits in a fit, especially if there are several of them in the same plane. This will be cool for doing realistic pattern reco track fitting including the MVD & GEMs.

Cheers, Christian

Subject: Re: plane id in track candidate

Posted by [Stefano Spataro](#) on Tue, 23 Feb 2010 09:36:44 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi,

I think the "two planes" problem is a problem of the finding, and not of the fitting.

Maybe the problem is another: if the kalman filter should be able to reject hits or not. I think the kalman should do it, regardless if two hits belong to the same plane or if one hit is just noise or coming from another track.

I would not support the hypothesis to add an integer to each PndTrackCandHit, it takes disk spaces (it means adding more than 20 integers per each track), and I think its use is too much "detector" related.

Once the kalman code will be ready to exclude hits from the candidates, maybe one could think about giving back a new PndTrackCand without noise hits, or adding a boolean flag such as fGood (at least only one bit).

My personal opinion, of course.

Subject: Re: plane id in track candidate
Posted by [Tobias Stockmanns](#) on Tue, 23 Feb 2010 09:47:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Christian,

I think the planeId is a good idea to have it in every hit. So this should be part of FairHit/PndHit and not of PndTrackCand.

The problem is how to generate this unique ID. For the MVD we store a string which looks like this: 1_1/202_1/205_4/1024_2/1025_3. This string is an encoding of the full path to a sensor within the geoManager. /`[volumeId]_[copyNr]/[next stage in hierarchy]/...` . With this information we are able to retrieve the full volume information and the positioning matrix of all our sensors.

If you want to match this information to a unique number you would need two maps: one from stringId to intId and one vis versa.

In addition you need to store this information in the parameter database to ensure that you do not loose the links.

This would mean some work for us but I think could be done.

Cheers,
Tobias

Subject: Re: plane id in track candidate
Posted by [Anonymous Poster](#) on Tue, 23 Feb 2010 09:53:32 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Stefano,

I described the problem of how the fitter is supposed to know whether the next hit is in the same plane as the last one. That is not reflected in your reply. What do you propose for this issue instead of what I just proposed? If I dont have this info, I might not be able to make the DAF work for PANDA, I guess...

Cheers, Christian

Subject: Re: plane id in track candidate
Posted by [Anonymous Poster](#) on Tue, 23 Feb 2010 09:57:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Tobias,

I understand your point of putting the planeId into PndHit. However, I do not see, how that info is available when I do the fit. When I do the fit, I just base it on what is inside PndTrackCand. Do you see a way out of this.

About assigning the ID: The ids would not even have to be consistent from track to track. As long as the same plane carries the same id in the same TrackCand, it would be fine.

BTW, for hits which are not even in planar detectors (STT, TPC, ...) we could just use `planeId==0`, which will be compressed away on the disk by ROOT. So, this will not be a major issue.

Cheers, Christian

Subject: Re: plane id in track candidate
Posted by [Anonymous Poster](#) on Tue, 23 Feb 2010 10:07:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

I just had another idea: How about putting the plane ID into the detector ID unsigned int which we have anyway. We could reserve the least significant 8 bits for detector id, the next 8 bits for `planeId`, and the rest maybe for the future. That would make the least overhead.

Christian

Subject: Re: plane id in track candidate
Posted by [Tobias Stockmanns](#) on Tue, 23 Feb 2010 10:29:37 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

I am a bit surprised, because `PndTrackCand` only contains a vector of links (`PndTrackCandHit`) to the underlying subdetector data. So you have to retrieve the hits from the detectors anyhow.

I am not in favor for having different ids from track to track. This means that you have to calculate the ids every time you do a track fitting.

To misuse the `detectorId` is no a good idea because this information is not used in the same way in the different sub detectors.

Cheers, Tobias

Subject: Re: plane id in track candidate
Posted by [Anonymous Poster](#) on Tue, 23 Feb 2010 10:45:19 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

you are right that I still need to access the hit info before fitting of course. But I just create the reco hits from the `TClonesArrays`, I dont look at more info. It is difficult to organize.

I would exactly call what I proposed a misuse of the detector ID....

Well then if there are only objection and no ideas on how to do it otherwise for the time being, I will not do it.

Christian

Subject: Re: plane id in track candidate
Posted by [Lia Lavezzi](#) on Tue, 23 Feb 2010 10:51:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Christian,

Quote:If the pattern recognition determines that two hits in the same plane (e.g. MVD) belong to the same track, because a clear distinction is impossible, they are both input to the track fit. Then maybe the fit can determine which is the outlier or simply it has to use both. The question is how to handle that situation, i.e. how does the Kalman filter (or another algorithm) know that two hits are on the same plane.

if you want to know that two hits are on the same plane in Kalman isn' t it enough to use a check like the one in GeaneTrackRep::getPos or getMom functions

if(pl!=fRefPlane)extrapolate(pl,statePred) to determine whether the extrapolation is needed or not?

Ciao,
Lia.

Subject: Re: plane id in track candidate
Posted by [Anonymous Poster](#) on Tue, 23 Feb 2010 10:58:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Lia,

that is how it was done so far, but I wanted to avoid all those operator== calls. And for the DAF I would need this information of which hits are in the same plane before i even call extrapolate.

Christian

Subject: Re: plane id in track candidate
Posted by [StefanoSpataro](#) on Tue, 23 Feb 2010 11:04:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

Christian Hoepfner wrote on Tue, 23 February 2010 10:53Hi Stefano,

I described the problem of how the fitter is supposed to know whether the next hit is in the same plane as the last one. That is not reflected in your reply. What do you propose for this issue instead of what I just proposed?

I supposed my answer was clear, but maybe I have not stressed it enough. I try to explain it again:

Stefano Spataro wrote on Tue, 23 February 2010 10:36Hi,
I think the "two planes" problem is a problem of the finding, and not of the fitting.

The pattern recognition algorithm should take care about this issue, if it is problematic. How much is the number of tracks affected by this problem?. How much are the performances reduced for this kind of problem? Do you have a quantitative estimation? Just to understand if it is a second order problem or more urgent.

Christian Hoeppe wrote
If I dont have this info, I might not be able to make the DAF work for PANDA, I guess...

What do you mean by "DAF"?

Subject: Re: plane id in track candidate
Posted by [Lia Lavezzi](#) on Tue, 23 Feb 2010 11:12:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

Ok, understood

Subject: Re: plane id in track candidate
Posted by [Anonymous Poster](#) on Tue, 23 Feb 2010 11:28:45 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

the DAF is the Deterministic Annealing Filter I was quoting in my first message in this thread.

I dont know what the order of the problem is. It is a principle problem that pattern reco can never 100% decide only the right hit in a detector plane. I am trying to find a consistent way to treat this problem. It might or might not affect many tracks, however the problem is interesting and real. It will be very interesting to study what happens with realistic MVD pattern reco for instance with different pile-up or noise levels. I think especially pile-up will be an issue as the MVD does not have a zero interation time, I assume. How do different fitting algorithms (such as the DAF) make the situation better/worse, ..?...?

Christian

Subject: Re: plane id in track candidate
Posted by [Gianluigi Boca](#) on Tue, 23 Feb 2010 14:31:33 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi all,

I believe that having also the subdetector plane, and/or , more generally, the info on what piece of the detector fired (e.g. MVD pixel or strip number; for the STT : tube number; for the TPC : the cell number if any; and so on) is very useful at fitting time.

I never investigated wether it is possible to have it easily based on the info stored in PndTrackCand.

If it is not (as Christian suggests) I believe we should provide one, either by using part of the Detector ID integer or with some other variable.

(By the way, also the STT can have that problem i principle. In fact it happens sometimes (in MC at least) that there are two hits in the same straw. Depending on the STT electronics that will be used, we could distinguish the two hits).

I also STRONGLY suggest that the Munich people please sooner or later add to GENFIT the capability of excluding outlier hits. In other words, the Kalman filter should have the functionality of eliminating a hit when it is too far away from the extrapolated track.

I understand their manpower problems.

However I think that should be put at the top of their priority list.

Cheers Gianluigi

Subject: Re: plane id in track candidate

Posted by [Anonymous Poster](#) on Tue, 23 Feb 2010 14:41:57 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Gianluigi,

what exactly do you propose as a strategy to reject outliers? Please remember that so far we do not have a smoother for our Kalman filter.

A request such as this seems reasonable, but I am not sure, it should be made to our group. There are two coordinators for tracking. We are not the only ones who can contribute to track fitting.

However, as I said we have such plans. But it is just a technical student on the task. So, there is no time estimate of guarantee that it will work. That is all the manpower we have for this in the moment.

Cheers, Christian

Subject: Re: plane id in track candidate

Posted by [Gianluigi Boca](#) on Tue, 23 Feb 2010 15:03:36 GMT

[View Forum Message](#) <> [Reply to Message](#)

The strategy to reject the outliers could simply consist in the exclusion of a hit when it is too

distant from the extrapolated trajectory to it.

I understand and respect your group's manpower problems; I agree that somebody else could help implementing that.

Gianluigi

Subject: Re: plane id in track candidate

Posted by [Anonymous Poster](#) on Tue, 23 Feb 2010 15:15:16 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi,

-- The strategy to reject the outliers could simply consist in the exclusion of a hit when it is too distant from the extrapolated trajectory to it.

That requires a good knowledge of the track parameters at any point. It requires a smoother which we currently don't have. Currently we only know the best estimates of the track parameters, including all hit information, at the first and last point. So currently it can't be done in a nice way.

Cheers, Christian

Subject: Re: plane id in track candidate

Posted by [Radoslaw Karabowicz](#) on Tue, 23 Feb 2010 15:45:50 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi,

I would like to focus in this post on the detectorID issue, as it is so close to the discussion I started ~2 weeks ago about the fDetectorType.

Assuming that we will all agree that we need a fast information on the plane id then the fDetectorId member of the FairHit class is the most intuitive place to get this information from. We could reserve there several bits for detector id, plane id, straw tube id for STT/DCH, side and strip number for GEM digi, and so on.

In order to have also information about the data type, the last 4 bits could be reserved for it. This could be set to either MC track, MC point, digi, cluster, hit, local track, global track or whatever you like. Of course, when the fDetectorId for track is created, then plane or even detectorId are useless, but the spared bits could be used for something else.

To summarize, we would have a 32 bit int, which could be divided into:

5 bits for the detectorId (31 possibilities, excluding 0, should be enough, 14 defined in PndDetectorList)

5 bits for the planeId (any detector needs more?)

18 bits for any detector specific stuff

4 bits for the data type (MC track, MC point, digi, cluster, hit, local track, global track or track

candidate, track)

I would not like to introduce another integer into existing data structure. I think we should try to use what is already available.

yours,
radek

Subject: Re: plane id in track candidate
Posted by [Anonymous Poster](#) on Tue, 23 Feb 2010 15:48:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Radek,

if the STT wants to use the 'planeId' to set the tube number, they would need more bits (I guess 16 bits).

I like your idea. Do you see anything speaking against transporting this info into the track candidate?

Cheers, Christian

Subject: Re: plane id in track candidate
Posted by [Radoslaw Karabowicz](#) on Tue, 23 Feb 2010 16:06:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

Well, it is very simple. One just add hits not with kGemHit, kSttHit, kMvdPixelHit, but with the hit->GetDetectorID(), and basing on this very number, or rather on the detector part of it, corresponding RecoHitFactory should be choosed. Therefore each added PndTrackCandHit will have different fHitId (equal to fDetectorId of the FairHit). But the decision on which RecoHitFactory to choose will base on part of this integer, a combination of the detectorId and dataType parts.

Is it possible to implement it like this?

yours,
radek

Subject: Re: plane id in track candidate
Posted by [Anonymous Poster](#) on Tue, 23 Feb 2010 16:11:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

yes this could be implemented. So to summarize, the detector ID in the trackCand would consist of a bit mask. Some part of this bit mask would be the kGemHit, kSttHit, kMvdPixelHit, ... which is used to produce hits in the RecoHitFactory. Could we do it a way that this kGemHit,

kSttHit, kMvdPixelHit, ... info is in the least significant bits? That way it will be easiest for me.

Regards, Christian

Subject: Re: plane id in track candidate
Posted by [Lia Lavezzi](#) on Tue, 23 Feb 2010 16:56:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Quote:if the STT wants to use the 'planeId' to set the tube number, they would need more bits (I guess 16 bits).

Hi,

yes, 5 bits are too few to store the tubeID: we have around 5000 straws in the STT, so 13 bits should be the number.

Ciao,
Lia.

Subject: Re: plane id in track candidate
Posted by [Stefano Spataro](#) on Tue, 23 Feb 2010 16:59:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

Excuse me,
maybe I have missed some part of the discussion,
but, if I have understood well, this fPlaneId is already contained inside the data object (FairHit->GetDetectorId()), and the trackcand is just the link between the track and each hit of the track.

Then, why you want to propagate an information already existing in the hit inside the track cand? From the trackcand one can retrieve the detector id and then compute plane id and so on. It is not clear to me why we should copy this number another time, and invent a new mask scheme, inside the track candidate.

Maybe the answer is in some message of the thread, but I was not able to find exactly the reason. Could you please short comment about, just to have a better idea?

Subject: Re: plane id in track candidate
Posted by [Radoslaw Karabowicz](#) on Tue, 23 Feb 2010 17:00:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

In fact, I would like to get rid of this kGemHit, kSttHit, kMvdPixelHit, and to replace it with something else. I would prefer PndDetectorList.h to look like:

```
enum DetectorId {  
    kDCH,kDRC,kDSK,kEMC,kGEM,kLUMI,kMDT,kMVD,kRPC,kSTT,kTPC,kTOF,kHYPG,kHYP}  
;
```

```
enum DataType {
```

kUnknown, kMCTrack, kMCPoint, kDigi, kCluster, kHit, kTrackCand, kTrack};

enum SensorSide { kTOP, kBOTTOM };

Now, if one creates for example a point in GEM station3, front side, I would set fDetectorId to:

fDetectorId = kMCPoint << 27 | kGEM << 22 | 3 << 17 | kTOP << 16;
or something like this depending on the number of bits reserved.

Digi created from this point would have fDetectorId:

fDetectorId = kDigi << 27 | kGEM << 22 | 3 << 17 | kTOP << 16 | digiNr;

Hit created from this point would have fDetectorId:

fDetectorId = kHit << 27 | kGEM << 22 | 3 << 17 | kTOP << 16;

If I create hit factory for GEMhits, I would give there the number kGEM<<5 | kHit, and choose for the factory only hits having correct bits in detector and data places.

As for which side of the fDetectorId to use, I would nevertheless prefer to use the most significant bits for more general information, so first datatype, then detectortype, planeid, side, and whatever.

BTW, I just talked to Gianluigi about the STT, and for STT I would record information about the planeID (defined as the straw layer number, starting from the closest to the beam pipe, looking at the STT picture provided by Gianluigi, there are 28 layers at most, and so 5 bits is just enough), sectorID (which is simply the sextant number, 3 bits), strawID (id of the straw in a layer 6 bits).

So a MC point in STT:

fDetectorId = kMCPoint << 27 | kSTT << 22 | layerID << 17 | sectorID << 14 | strawID << 8;
I hope it satisfies STT:)

For MVD it would mean, there should be two data types: kMVDStrip and kMVDPixel.

And so on also for different detectors. This way the information is highly structurized, and from this one integer one can get information about the data type, detector and plane ids. Exactly this integer fDetectorId would be stored in PndTrackCandHit.

yours,
radek

Subject: Re: plane id in track candidate

Posted by [Radoslaw Karabowicz](#) on Tue, 23 Feb 2010 17:06:38 GMT

[View Forum Message](#) <> [Reply to Message](#)

Dear Stefano, I think it was the idea long time ago, but please show me how one can retrieve information from a hit, digi or cluster member fDetectorId?

We are putting there: kMVDHitsStrip or kGEMHit and no any plane information is coded inside there.

Subject: Re: plane id in track candidate
Posted by [Stefano Spataro](#) on Tue, 23 Feb 2010 17:20:07 GMT
[View Forum Message](#) <> [Reply to Message](#)

From detID you know the name of your TCA and the data object.
From the hitID you know the position of your object inside the TCA.

With a switch/case you are able to retrieve all the informations you need. Example for MVD:

```
if (candHit.GetDetId()==kMVDHitsPixel) mvdHit =  
(PndMvdHit*)fMvdHitsPixel->At(candHit.GetHitId());  
if (candHit.GetDetId()==kMVDHitsStrip) mvdHit =  
(PndMvdHit*)fMvdHitsStrip->At(candHit.GetHitId());
```

and from the MvdHit you get the detector name, you use the MvdGeoHandling to have all the kinds of plane you need.

Or maybe I have not understood the question,

Subject: Re: plane id in track candidate
Posted by [Tobias Stockmanns](#) on Tue, 23 Feb 2010 19:48:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Radek,

I am not in favor for your suggestion to separate the detectorId and the type of data which is stored. The data is structured in different ways in the subdetectors.

In my point of view all branches stored in a root file should have a unique ID to retrieve the data in an automatic way. This is mandatory to use the MC information propagation as I tried to explain in my presentation during the last EVO meeting.

If you replace now the single enum fDetectorType by two you can have combinations that are not reasonable (for example MVD plus bump) and for me it is not clear how to match non-detector data in this scheme.

I fear we are mixing up two different things. The structure of the detector enum and the request to have a common planeId for each hit. In the MVD such a method to get the planeID already exists. The STT and the TPC do not need such a thing, so only the GEMs are missing. We could think about a common interface which should clearly go into the FairHit but to change the enum has nothing to do with it.

I thought that we fixed the enum last EVO meeting but we can discuss this again during the PANDA meeting in two weeks.

Just as a remark. The Riemann track finder in the MVD already excludes hits on the same

detector for one track.

Cheers,

Tobias

Subject: Re: plane id in track candidate

Posted by [Radoslaw Karabowicz](#) on Tue, 23 Feb 2010 20:28:53 GMT

[View Forum Message](#) <> [Reply to Message](#)

Dear Stefano,

Then I do not understand what Christian would like to have. Maybe a faster, easier, more intuitive and consistent way to get information about planeId? You have to agree that:

```
if (candHit.GetDetId()==kMVDHitsPixel) mvdHit =  
(PndMvdHit*)fMvdHitsPixel->At(candHit.GetHitId());  
if (candHit.GetDetId()==kMVDHitsStrip) mvdHit =  
(PndMvdHit*)fMvdHitsStrip->At(candHit.GetHitId());
```

and then retrieving plane information from PndMvdHit fDetName string is not really easy. Is for sure not fast. And will not work for GEM, STT or DCH. And intuition says one could do it better.

I've just seen a response by Tobias, so I guess I have to step down so we can forget about my proposition.

Only one comment for Tobias. IMHO "the common interface" will never go into FairHit, because f.e. CBM is dealing with the problem completely different. For sure they won't be happy with trashing FairHit with another variable and for sure will never agree to store TString inside...

yours,
radek

Subject: Re: plane id in track candidate

Posted by [Tobias Stockmanns](#) on Wed, 24 Feb 2010 07:12:33 GMT

[View Forum Message](#) <> [Reply to Message](#)

Dear Radek,

I never wanted to have a TString in FairHit. But one could think about an integer or even use the detectorID, but one should have it in a common way for all subdetectors. If CBM does not like our suggestions we could derive a PndHit from FairHit and they are not affected at all.

The two things I do not like is the storage of the plane information in the PndTrackCand because this is surely a hit information and your suggested modification of the detector enum which is in conflict with the MC propagation.

Cheers,

Tobias

Subject: Re: plane id in track candidate
Posted by [Radoslaw Karabowicz](#) on Wed, 24 Feb 2010 08:59:01 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Tobias,

I never said I want to store information about planeId in the PndTrackCand. I meant that for this we should use the fDetId member of the PndTrackCandHit class. There would be no changes to the PndTrack{,Cand,CandHit}.{h,cxx} classes AT ALL.

Another issue is the MC propagation. The information that used to be coded in the fDetectorId is still there, only coded at some bits of the Int_t. To retrieve this information is trivial - as Christian agrees - and the bit shift operations are virtually costless.

You expressed concern also about the misuse of the scheme proposed by me and possibility of creating MVD Bumps. Well, it is always possible and you even now you cannot avoid users using integers out of the the enum fDetectorType of PndDetectorList.h.

Anyways, as I see my idea was generally disliked, so we still have to think a bit more about this problem.

Yours,
radek

Subject: Re: plane id in track candidate
Posted by [Lia Lavezzi](#) on Wed, 24 Feb 2010 11:47:37 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

sorry, maybe the discussion has already moved to a different topic (I just read quickly the last messages), and this remark is not usefull anymore, but let me just specify a thing on the tubeID of the STT.

Quote:BTW, I just talked to Gianluigi about the STT, and for STT I would record information about the planeID (defined as the straw layer number, starting from the closest to the beam pipe, looking at the STT picture provided by Gianluigi, there are 28 layers at most, and so 5 bits is just enough), sectorID (which is simply the sextant number, 3 bits), strawID (id of the straw in a layer 6 bits).

So a MC point in STT:

```
fDetectorId = kMCPoint << 27 | kSTT << 22 | layerID << 17 | sectorID << 14 | strawID << 8;
```

I hope it satisfies STT:)

I am implementing a map of the tubes (I don't remember if I already told about this, but it is one of the workpackages) and in it I assigned the tubeID with the same numbering scheme of the stt geo files: each tube has a different number, from 0 to about 5000.

I would really prefer (if possible) to keep this numbering scheme and use only the strawID (or

tubeID), so 13 bits, unless there is some particular reason to change it, because in this way the macro for tube positioning and the geo files will all have the same numbering scheme.

Thank you,
Lia.
