Subject: ILC vertexing
Posted by cdamerell on Thu, 29 Sep 2005 16:33:25 GMT
View Forum Message <> Reply to Message

Dear SiD tracking colleagues,

After last Friday's discussion, Sonja and I thought we should send round an e-mail to clarify some points from Snowmass where we evidently did not communicate our studies adequately, and also to provide updates on some recent developments, in the areas of:

luminosity factors associated with vertex charge

maximum hit density in layer-1

radial separation between layer-1 and layer-2

None of this is worked out in any detail, and it will be great if others look into these questions, as Bruce suggested he might do.

1. Luminosity factors

We generated b-bbar events at a series of sqrt(s) values (50, 100 and 500 GeV), purely in order to obtain a sample of b jets of energies 25, 50 and 250 GeV. Then the quality of the vertex charge measurement is evaluated for various detector configurations. The 'luminosity factor' or L-factor is defined to be the factor in integrated luminosity needed in order to make a measurement that relies on vertex charge QVTX, with the same statistical precision as for the 'standard' LDC vertex detector. This factor obviously depends on the jet energy, the sensitivity to the detector configuration being greatest for low energy jets. We call this factor the 1-jet L-factor, f_L,1 in Sonja's talks.

In a physics process (eg e+e- --> t tbar H at sqrt(s) = 1 TeV, one may need to tag QVTX for more than one jet (eg one of the b jets in the t or tbar decay chain, to know which was t and which tbar, and one of the b jets in the H decay, in order to perform the top-polarisation and top-Higgs angular correlation analyses. Physics examples were given in my LCWS 2005 talk, based on advice from Peskin, Zerwas, Bartl, Moortgat-Pick, Riemann, etc.

The point is that there will be many physics analyses where one needs to determine QVTX for two jets, and these will have an associated L-factor of [f_L,1]^2, which we call the 2-jet L-factor f_L,2. For some processes, one will need to measure QVTX for 4 jets in the event (say an 8-jet process such as AH production), for which the 4-jet L-factor [F_L,1]^4 would be appropriate. So we plotted these factors for three values of jet energy (25, 50 and 250 GeV) as examples. Of course, in multi-jet processes, the jets have broad energy distributions, so these are only an indication. But

the average jet energy in a 6-jet process at sqrt(s) = 500 GeV is 83 GeV, so these plots give a good general idea of the sensitivities.

We haven't yet got to the point of full simulations of any physics processes. When we do, we will encounter further complications such as track confusion between overlapping jets, but there is a plausible procedure for dealing quite effectively with that (as regards tracks in the decay chain) so we are optimistic that these naive results give a reasonable indication of what to expect.

As Sonja emphasised in her talks, these L-factors are only a first indication, but they should alert the machine people to the necessity of restricting the beampipe radius to the range 12-15 mm if possible. We have lost a lot of physics at previous e+e- colliders through excessively large beampipes (B_s mixing at SLD and a conclusive Higgs result at LEP) and should if possible avoid history repeating itself at ILC.

We hope that this clarifies our Snowmass study, and will of course be delighted to discuss this at a future SiD meeting, if it isn't yet clear. We are the first to agree that this is only the tip of the iceberg, in terms of work to be done on this subject.

2. Maximum hit density in layer-1

Witold took an upper limit for the background hit density of 6 hits/mm^2 per SW (sensitivity window) as representing 1% occupancy, and queried whether this could be pushed higher. The first point to emphasise is that a SW of 50 microsec is the best on offer from all the VXD options currently conceived, and is applicable to the CPCCDs of LCFI, CMOS pixels of Strasbourg, and ISIS of LCFI. In one sense, the micropixel/macropixel concept with its fast time stamping does better, but in another sense it is worse, as will become apparent from the following discussion.

Let us first agree on the purpose of layer-1. We have for years said that the standalone track finding should be based on layers 2-5 where backgrounds are much lower. If that doesn't work, one should add a 6th layer and/or endcaps, etc. So the purpose of layer-1 is primarily to provide another point on the already-found track, in order to improve the impact parameter precision at the IP. For high momentum tracks, this is not needed. That innermost point is only really important for tracks in the mult-scatt dominated p_t range where one still needs performance for QVTX (NOT for b or charm tagging) namely 0.1-0.5 GeV/c. In this range, the extrapolation of the found track to layer 1 results in a search ellipse of significant area. For example, at 0.1 GeV/c, the material in layer-2 (0.1% X_0) for a 90 degr polar angle track, results in a 2-sigma search circle on layer-1 of radius 76 microns, so the probability of a background hit in this region is 11%. Not a disaster, but it would be highly undesirable to push much higher with the background.

For the micropixel/macropixel it is much worse, since the time stamping is ineffective for the 50 micron square macropixel in which the signal is located. For that region, background hits THROUGHOUT THE TRAIN will cause confusion. To my knowledge, this was first pointed out by

Gerhard Lutz, at Snowmass.

So the conclusion is that unless there is a dramatic new idea, the specified max hit density in layer-1 should be respected, or this layer would rapidly become useless for refining the measurement of low momentum tracks for which it is really needed. Again, this is rather handwaving, and we look forward to ourselves or others being able to improve on this calculation. It requires a lot of simulations (QVTX to physics processes) to be done properly.

3. Separation layer-2 to layer-1

As John J pointed out, one could in principle improve the impact precision for low momentum tracks by extending the radial difference between layer-1 and layer-2. However, if the layer-1 hit is lost for some reason (inefficiency, or ambiguity with a background hit as mentioned in point 2), then one pays a heavy price if the nearest layer-2 hit is further away. Fine tuning this aspect of the geometry needs a detailed simulation of backgrounds, plus some realistic estimate of the layer efficiency - which will hopefully be very close to 100%.

We hope this note is useful, despite being so handwaving at this stage. What do others think?

Best regards,

Chris and Sonja

Subject: Re: ILC vertexing
Posted by sinev on Fri, 30 Sep 2005 20:26:19 GMT
View Forum Message <> Reply to Message

Hi Chris, Sonya and all SID trackers,

Let me add few comments to Chris' and Sonya's note.

1.I think that use of occupancy figure to understand limitation on
the tolerable background level for the tracking performance is a little bit
misleading. It looks to me, that hit density per sensitive window is more
informative. It is certainly better for low momentum tracks, where search
area for assigning hits to track is dominated by multiple scattering, but it
is also a good parameter for high energy tracks. Here our search area is
defined by layer spatial resolution, which does not depend on the occupancy,
but is the characteristics of detector parameters. So, for given resolution
our performance is a function of hit density again. As soon as clusters from

different hits do not merge, we don't care if cluster contain 1 pixel, or 2, or 3...
So, occupancy is misleading in that sense also. It only relevant to the case
then clusters start merging, but it occurs at such high occupancies, that
we will not be able to reconstruct tracks anyway for other reasons.

2. I agree with Chris, that pushing hit density higher than ~6 hits/mm**2/SW
will cause troubles for any existing VXD design, including Macro/micro pixels.

3.Comparison of Macro/micro pixels performance with traditional pixelated
device with limited time resolution requires more study. At the first look,
Chris is right that Macro/micro pixel device will have worse performance at
 low momentum, when search area exceeds the size of macro pixel.
The size of the search area I would rather assume as 3 sigma instead of
Chris's 2 sigma, to keep efficiency high. The Macro/micro pixel concept
will start showing advantage only when search area diameter became smaller
than Macro pixel size (50 microns), i.e. at more than 0.4 GeV. At very low
momentum (less than ~0.06 GeV) Macro/micro pixels may be again better
than traditional pixels with 50 micron resolution, because at such momentum
search area will exceed 20 times area of Macro pixel. But such low momentum
tracks unlikely can be reconstructed at all.

4.Concerning weather to include first layer in the pattern recognition - it, again, requires
more simulation studies, but at the first look, we can exclude it from pattern
recognition even without increasing number of layers. And it will not cause noticeable
drop in efficiency. The trick is to keep it for hit counting .
(For number of hits assigned to track cut)
Dropping it from pattern recognition will increase number of layer combinations
needed to be considered, but this probably will be compensated by much smaller
number of hit combinations (as most populated layer will be excluded). So reconstruction
may even became faster. And number of fake tracks would probably go down.

Best regards,
Nick

Subject: How to estimate track finding efficiencies?
Posted by wpark on Thu, 12 Jan 2006 06:03:54 GMT
View Forum Message <> Reply to Message

Dear Tim Nelson or anyone else,

I read the following links
http://nicadd.niu.edu/cdsagenda//askArchive.php?base=agenda& amp;
amp;categ=a0567&id=a0567s1t0/moreinfo

and tried to do track finding by myself using hit information.

I found several tracks from my track finding code but I would like to know if my tracks are from one of MCtruth matched particle. Then, I can estimate track finding efficiency. As I guess, according to the presentation, Tim knows which hit belongs to what particle exactly. But, I don't know how.

If I found one track and I knew the id's of hits, then how
can I be sure all the hits are from single track and from which track?

In page4, I don't understand the first two conditions.
"Take all combinations of three hits that have no nearby (<0.5mm) hits"::: why? Is it because we don't want track
in jets? If there are nearby hits, what does it mean?
Then, we don't want such a track which can be confused by
nearby hit?

"Tried both 10 cm and +/- z modularity"
Actually, I don't understand this at all.

Isn't there any update about this topic recently?

I appreciate any comments on this.
Thanks, Woochun

Subject: Re: How to estimate track finding efficiencies?
Posted by wpark on Tue, 17 Jan 2006 18:47:03 GMT
View Forum Message <> Reply to Message

Hi, all,

Some of questions are solved by myself thanks to org.lcsim
document. Now, I have new questions.

(1) I did check hit information to see which MCtruth particle deposit hits in the tracker and vertex detector. Sometimes, hits are associated with MCtruth particle which is in intermediate status. Should I consider those hits to do track finding? Or, is it okay to ignore those one?

Cheers, Woochun

Subject: Re: How to estimate track finding efficiencies?
Posted by wpark on Tue, 17 Jan 2006 23:11:03 GMT
View Forum Message <> Reply to Message

Hi Woochun,

To be clear, the cuts described in that talk are the default cuts
that are applied by my (very rudimentary) standalone tracking code.
You are welcome to play with those cuts (they are easy to change),
but be aware that I have tuned them to work well with the current
detector configuration...

On Jan 17, 2006, at 1:34 PM, Woochun Park wrote:

> Hi, Tim,
>
> I reviewed your presentation in Sep 2, 2005 and would like to ask
> questions.
> http://nicadd.niu.edu/cdsagenda//askArchive.php?
> base=agenda&categ=a0567&id=a0567s1t0/moreinfo
>
> In page 4, you showed several cuts to apply.
>
> (1) Take all combinations of three hits that have no nearby
> (<0.5mm) hits:::
> I checked hit information carefully and there are many hits which are
> nearby one another. The following lines are from singlePi MC
> costheta = 90
> degree and 5 GeV energy::
> As you see, whenever GeneratorStatus==0 (i.e. intermediate status),
> I can
> see a lot of nearby hits. Why do we see those hits? Can it be real
> effect
> in real data taking? If so, what does make such a chunk of hits?
>
>

Yes, I have seen this also. It is my understanding that this is a

real effect due to the way that GEANT4 works: every time GEANT4 simulates a step in the detector, it creates a new SimTrackerHit so that a given track may create more than one SimTrackerHit in a given piece of silicon. In particular, every time that GEANT4 produces a secondary (e.g. a delta ray) it begins a new step at the production point so every interaction of this type results in two SimTrackerHits where there would otherwise be one, plus at least one SimTrackerHit for the delta ray itself. Also, GEANT4 produces very short range secondaries in neighboring volumes that enter the front face of the silicon but deposit very little energy.

That said, I do agree that the number of SimTrackerHits seems excessive and am still trying to convince myself that this is the correct behavior.

Now, the real intent of the cut you have cited is to try to seed tracks with 3-hit combinations that will point unambiguously to the primary and to the other layers. Remember that when all 5 layers are hit, one has 10 chances to seed a track with some combination of 3 hits. However, a side effect of the cuts I have chosen is that it ensures that each track is found only once even when there are many SimTrackerHits nearby.

When the silicon is simulated properly, all of these nearby SimTrackerHits will be clustered together into a single TrackerHit, and all of this strange behavior will disappear. This should result in performance that is both better (higher efficiency) and faster than you will find when running this code on SimTrackerHits.

> x_position of hit y_position z_position ||
> ParticleGeneratorStatus PDGID energyofTrack
> 10.325094209070244 9.587519468669045 -5.7604608714084556E-5 ||
> 1 -211
> 5.001947320271581
> 18.357463504736167 17.102971485874463 -5.271918297547584E-4 ||
> 1 -211
> 5.001947320271581
> 26.363896753699848 24.64615685921717 -9.335380191350699E-4 || 1
> -211
> 5.001947320271581
> 35.06972721500355 32.90535416965908 -9.919138593817624E-4 || 1
> -211

> 5.001947320271581
> 43.746383901883 41.19541231110294 -4.2774421353624855E-4 || 1 -211
> 5.001947320271581
> 144.14176991269727 141.63861539488215 0.004670921243307884 || 1
> -211
> 5.001947320271581
> 318.6226784930068 338.79735221209717 -0.00803043956901691 || 1
> -211
> 5.001947320271581
> 476.88030924541226 548.8513164856695 -0.08596810148078912 || 1
> -211
> 5.001947320271581
> 619.1385483757203 772.616181964388 -0.1716982835022584 || 1 -211
> 5.001947320271581
> 726.9318291273557 973.6537743229092 -0.20095058126166754 || 1 -211
> 5.001947320271581
> 237.29841906985402 961.1461682864208 -364.1049486349469 || 0 11
> 5.316918498016788E-4
> 195.32140763750402 1199.3667096615004 431.7639547556472 || 0 0
> 25.126853059385308
> 195.30026629760823 1199.3866221579983 431.7919142341049 || 0 2212
> 0.9410298755058041
> 88.71994838723148 -986.2321309561347 -467.98763607655155 || 0 11
> 7.771632249124437E-4
> 88.67581315828947 -986.2303270571076 -467.9829323893658 || 0 11
> 7.771632249124437E-4
> 88.65021169857584 -986.2530643343255 -467.96565887017164 || 0 11
> 7.771632249124437E-4
> 819.5135966176931 -897.1165831721878 154.79314004220382 || 0 11
> 6.178083948069108E-4
> 949.3555406313811 281.57950988360074 -16.169725220417895 || 0 11
> 9.5359907797218E-4
> 949.3196584860093 281.6214941421723 -21.823229150203517 || 0 11
> 9.5359907797218E-4
> 949.3255627902593 281.5954197213556 -21.849517776381255 || 0 11
> 9.5359907797218E-4
> 949.3532580949012 281.56258443005277 -21.844634022102774 || 0 11
> 9.5359907797218E-4
> 949.3581714113019 281.53191364343036 -21.815833997257776 || 0 11
> 9.5359907797218E-4
> 949.3607147104608 281.52550338336385 -21.796455308290522 || 0 11
> 9.5359907797218E-4

>
> (2) Tried both 10 cm and +/- z modularity::
> I don't understand this part. How can I require this thing and what is
> this for?
>

My code uses a rough approximation to require this by default.
However, you can easily change the default if you desire.

> I appreciate your help in advance.
>
> Thanks, Woochun

No problem. I will get the code working again (changes to org.lcsim
broke it recently) and let you know where to find it... probably
later today or tomorrow.

Best,
Tim

Subject: Vertex Class for LCIO?
Posted by NormanGraf on Tue, 16 May 2006 16:54:34 GMT
View Forum Message <> Reply to Message

Dear Colleagues,
Please see this thread in the LCIO forum for a discussion on whether a new Vertex class is needed for LCIO.
Norman Graf

Subject: TPC Subdetector Description in GEAR
Posted by jabernathy on Tue, 13 Jun 2006 19:35:26 GMT
View Forum Message <> Reply to Message

Hello everyone,

I was wondering what everyone thought about adding some more tags to the GEAR description of the TPC.

I propose something that looks like this in the GEAR file:

```
<!-- build the TPC inward from the readout electronics -->
 <gap name="induction_gap" width="5mm" eField="0 0 1000V"/>
 <gem name="gem2" gain="40" width="0.5mm" eField="0 0 1000V"/>
 <gap name="transfer_gap" width="0.5mm" eField="0 0 1000V"/>
 <gem name="gem1" gain="40" width="0.5mm" eField="0 0 1000V"/>
 <gap name="drift_gap" width="*" eField="0 0 1000V"/>
```

and then the GEAR interface could have methods for retrieving information about the geometry of the TPC as so:

```
gear::TPCParameters.getGemParameters("gem1");
```

or

```
gear::TPCParameters.getGapParameters("drift_gap");
```

This would allow a common interface for describing and querying the geometry inside of the TPC which is useful for the simulation and event display.

Subject: Re: TPC Subdetector Description in GEAR
Posted by mjanssen on Wed, 14 Jun 2006 08:09:07 GMT
View Forum Message <> Reply to Message

I"m not convinced, that we should place fields, and gain (which are values changing very often in real data) in the gear file.
I know that also drift velocity is in the gear file, and for simulation an the first look it is the right place, but we should use LCCD for these values, because that will make the reconstruction of MC generated data easier.
Last year, we (DESY and Aachen TPC groups) agreed on a package to store conditions data for TPCs. This may be a good starting point. And I say starting point, of cause now for the new Project changes may be needed. I attached the files in a tar-archive.

File Attachments
1) tpcconddata_20060614.tar.gz, downloaded 45095 times

Subject: Re: TPC Subdetector Description in GEAR
Posted by killenberg on Wed, 14 Jun 2006 08:56:52 GMT

View Forum Message <> Reply to Message

The dimensions of the setup look somewhat strange to me: 0.5 mm for a GEM? The transfer and induction gap should have the real size (1 mm for instance) and the GEM should be either 60 microns or, if this is too detailed, 0.

As the gain and the field settings should be in the conditions data, as Matthias proposed, the GEM does not have much functionality, but never the less it should be there to indicate that there is a GEM and then the reconstruction or simulation can read the conditions data for this object.

But I think the drift gap is not needed, it is the active TPC volume itself. Again, the field is in the conditions data and the length is the TPC length (at least in the definition we are using in Aachen).

Greetings

Martin

Subject: Re: TPC Subdetector Description in GEAR
Posted by jabernathy on Thu, 15 Jun 2006 00:05:12 GMT
View Forum Message <> Reply to Message

Yes, those dimensions were just examples. They have no meaning whatsoever.

If it is not suggested to keep the gem / gap parameters inside of the GEAR file then how about the physical parameters such as the location of the gem / micromegas?

The reasoning behind the gaps inside of the description would only have meaning if non-physical parameters were also there (such as diffusion and electric field parameters). If those parameters are not there then perhaps it would make sense to keep the gaps in the file for aiding with the physical placement of the gems?

In our simulation the width of the "induction gap" is a needed parameter for one of the implementations of a shaping amp. Should this geometric parameter be put inside of the GEAR file?

Perhaps the file could look something like this?

<gap name="induction_gap" width="1mm"/>
<gem name="gem2" gain="40" width="0mm"/>
<gap name="transfer_gap" width="0.5mm"/>

<gem name="gem1" width="0mm"/>

Subject: Re: TPC Subdetector Description in GEAR
Posted by killenberg on Thu, 15 Jun 2006 06:19:10 GMT
View Forum Message <> Reply to Message

Yes, I think that's exactly the way it should be. (Except of the `gain = "40"` in the second GEM, which you forgot to remove)

For the micromegas the amplification gap should be part of the micromegas itself and not a separate gap, as the amplification takes place in the gap and not in the mesh. Perhaps the 50 microns is not negligible here, as it is the only width in this structure.

Subject: LCIO classes for Timepix readouts
Posted by wiene on Wed, 21 Jun 2006 18:03:57 GMT
View Forum Message <> Reply to Message

Hi,

I have a couple of questions concerning the usage of the LCIO tracker classes for Timepix based readouts.

The Timepix chip can be run in the following modes
(the mode can be chosen on a pixel by pixel basis):

1. Time-over-threshold (TOT) mode:
Pixel provides total time (in clock counts) over threshold for all hits in a readout cycle.

2. Medipix mode:
Pixel provides number of hits over threshold during readout cycle

3. Timepix mode:
Pixel provides time (in clock counts) between first hit and end of the readout cycle.

4. Forth mode:
Probably not needed for TPC applications.

How shall we store these data in LCIO?

TrackerData (containing the raw data for this particular type of readout technology (and for TDC based readouts) -> see previous discussion):

Mode 1: How shall we store the time information? The time variable expects a drift time, not a TOT (which is more charge information than time information). Do not use the time variable and use the first entry of the getChargeValues() vector to save the TOT - or is there a better solution? What convention shall be use to indicate that the time variable is not used? Set it to a value < 0?

Mode 2: How shall we store the number of hits (n) over threshold?
Fill n TrackerData objects for one channel in the corresponding collection? What charge should they contain in this case?

Mode 3: Here the time provided by the pixels is T_cycle - t_drift
with T_cycle = duration of readout cycle and t_drift = drift time. Storing this information in the time variable (expected to contain a drift time) would be misleading.

TrackerPulse:

Here we should use some of the quality bits to indicate the operation mode of the pixel. Should we use official bits for that or some of those reserved for private usage?
From these bits we can then decide whether the time and charge variables contain useful information (depending on the operation mode).

TrackerHit:

Here again we should use some of the quality bits to indicate that there might be hits with incomplete information.

Any suggestions, comments, ...?

Cheers, Peter

Subject: MarlinTPC Repository
Posted by wiene on Thu, 29 Jun 2006 08:41:09 GMT
View Forum Message <> Reply to Message

The MarlinTPC repository is set up.

The details on how to check out and check in code is described on the web site

http://www-zeuthen.desy.de/linear_collider/

The MarlinTPC CVSROOT path (for anonymous checkout) is

export CVSROOT=:ext:anonymous@cvssrv.ifh.de:/marlintpc

The module name is "MarlinTPC"

Cheers, Peter

Subject: importing GDML file
Posted by danieleINFN on Sat, 01 Jul 2006 10:36:16 GMT
View Forum Message <> Reply to Message

ILCRoot import GDML file for Vertex detector

Subject: Re: importing GDML file
Posted by danieleINFN on Sat, 01 Jul 2006 10:55:45 GMT
View Forum Message <> Reply to Message

Hi Sidders,

I work to ILCRoot in INFN Lecce, I implemented into framework the possibility to import a GDML file. Now only for VXD. I imported the sid00.gdml. This file contains tracking volumes and support volumes. Each layer is composed by one barrel on 6,2... degrees angle. (Deltaphi, gdml value). To obtain a full geometry of VXD, it needs clone barrel volumes foreach layer or there are others newer gdml file versions where geometry is full?

There are real and complete GDML geometry images on the web?

Thank you so much.

Daniele Barbareschi

Subject: LCIO Track incomplete for Likelihood Track Fitting

Posted by jabernathy on Sat, 01 Jul 2006 18:38:37 GMT
View Forum Message <> Reply to Message

Hello!

While working on the LikelyhoodTrackFitter in Marlin we realized we didn't have spaces in the Track class for storing (as an example) the sigma (diffusion) of the Track.

Does anyone have some suggestions for extending the Track to store extra parameters?

I could think of a couple quick ways for us to store the extra parameters.

1. Adding extra Parameters to the collection in which the fitted Track is stored. That way sigma (or any other extra parameter) could be accessed using

FittedTrack.getParameters().getFloatVal("sigma");
or for multiple tracks in the collection
FittedTrack.getParameters().getFloatVals("sigma").at(0);

2. Just throwing the things like sigma in already existing variables that we aren't using (such as chi^2).

3. Perhaps rewrite the Track class with an LCParameters member variable. That way when a new algorithm is developed arbitrary parameters can be added.

Does anyone have any suggestions on how to extend Track without _actually_ inheriting from it and needing to write new input/output implementations?

Subject: Re: MarlinTPC Repository
Posted by mjanssen on Mon, 03 Jul 2006 07:59:20 GMT
View Forum Message <> Reply to Message

before someone check in code, we should agree on the setup of directories:
I propose the following

in the main folder
- src
- include
- bin
- lib

all empty. In the src and include you softlink the code you need from the subpackage to compose you personal MatlinTPC, but in the repository this should be empty.

Additional we need subfolders for the different packages
- reco
  - src
  - incluce
  - bin
  - lib
- simulation
  - src
  - include
  - bin
  - lib
- analyzer (e.g. pedestal calculating processor)
  - src
  - include
  - bin
  - lib
- tools
  - tpcconddata
    - inlucde
    - src
    - lib

please add folders to this list, if you thing it will be needed.
We should discuss the final setup on Thursday.

cheers Matthias

Subject: Re: LCIO Track incomplete for Likelihood Track Fitting
Posted by mjanssen on Mon, 03 Jul 2006 08:21:21 GMT
View Forum Message <> Reply to Message

Hi Jason,

the proposal of Frank for this extra information is to use LCFixedObjekts (special LCGenericObjects) to to store this informations.
Every time you store a track in the track collection you store the extra inforation object in another collection.
So you know that the i th track object in the track collection is related to the i th extra object in the

other collection.
I know that this is not the best object oriented approach, but in out program you can combine these two information again.

Unfortunatly I have no time to provide you some code, which will explain this in more detail.

cheers Matthias

Subject: Re: MarlinTPC Repository
Posted by wiene on Mon, 03 Jul 2006 08:22:02 GMT
View Forum Message <> Reply to Message

If you check

http://www-zeuthen.desy.de/lc-cgi-bin/cvsweb.cgi/MarlinTPC/? cvsroot=marlintpc

you see that I have already added some directories to the repository. During the DESY meeting we decided to have the following dirs:

- simulation
- reconstruction
- analysis
- (a still missing directory for generic object wrapper classes and other stuff)

No further decisions were made so far.

I would suggest to split up each dir into include, src, etc.

If the general feeling is that this structure is not wanted anymore, we should put it on the agenda on Thursday and decide
on it once and for all.

Cheers, Peter

Subject: Re: LCIO Track incomplete for Likelihood Track Fitting
Posted by jabernathy on Mon, 03 Jul 2006 17:40:04 GMT
View Forum Message <> Reply to Message

I understand. It would be similar to the way the TPCConditions data works (without the LCCD).

Subject: INSTALL of MarlinTPC
Posted by mjanssen on Fri, 07 Jul 2006 10:30:34 GMT
View Forum Message <> Reply to Message

Hi all,

for compiling the Processor of Marlin TPC it is recomended to set:
export LCIO="Where ever it is"/lcio/v01-07
export PATH=$LCIO/tools:$LCIO/bin:$PATH

export MARLIN="Where ever it is"/marlin/v00-09-04

export TPCCondData="where ever it is"/tpcconddata

the submitted exaples work without LCCD and GEAR but you should set:
export LCCD="Where ever is is"/lccd/v00-03
export GEAR="Where ever it is"/gear/v00-03beta
export PATH=$GEAR/tools:$GEAR/bin:$PATH

if you what to use the CondDBDatabase you must set:
export CondDBMySSQL="Where ever it is"/CondDBMySQL
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$CondDBMySQL/lib

export MYSQL_PATH="Where ever it is"/mysql
export PATH=$MYSQL_PATH/bin:$PATH
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$MYSQL_PATH/lib/mysql

I Hope this will help you.

Matthias

Subject: TrackFinder
Posted by jabernathy on Sat, 29 Jul 2006 03:37:15 GMT
View Forum Message <> Reply to Message

Hello

I had a question about the data stored in the TrackerPulses which are used to create TrackerHits

coming from the TrackFinder section of reconstruction.

Can it be assumed that the charge information in the TrackerPulses only contain information from the one track that the TrackerHit came from? Or does the TrackerPulse contain the entire spectrum?

- Jason

Subject: Re: TrackFinder
Posted by killenberg on Mon, 31 Jul 2006 07:43:45 GMT
View Forum Message <> Reply to Message

Hello Jason,

the TrackerPulse contains all the charge the ADC counted at a particular time. If there are two crossing tracks, at the intersection point the charge from both tracks is in the TrackerPulse and even in the TrackerHit. They are reconstructed hits and not comparable with hits created by Monte Carlo generators like Geant.

In case the pulse shape is strange (too long of containing a dip), the pulse can be flagged as double pulse candidate. The TrackerHit has some quality flags, too.

Greetings

Martin

P.S. There even might be noise pulses or hits which do not belong to a track at all.

Subject: Re: TrackFinder
Posted by jabernathy on Tue, 01 Aug 2006 20:24:52 GMT
View Forum Message <> Reply to Message

Ok, then does this mean that two crossing tracks will share TrackerHits and Pulses or is there a separate copy for each track?

The reason I am asking is that in the TrackFitterLikelihood we need to know the time of the maximum peak contributing to a track. If there are multiple tracks (aka peaks) in a TrackerPulse/Data I'm not sure how to deduce that.

Subject: Re: TrackFinder
Posted by killenberg on Wed, 02 Aug 2006 07:57:20 GMT
View Forum Message <> Reply to Message

There will only be one copy of a pulse or hit. Else each copy would have the full charge of the double hit.

I think the idea was to separate a double pulse into two, and then two hits are reconstructed from the separated double pulses. But I don't know how to disentangle how much charge was in which pulse/hit. As far as I know, none of the existing algorithms do this separation.

Would it be possible to have something like a track finder which produces a double track candidate, containing all hits from both tracks, and then do one likelihood fit for both tracks?

Subject: Re: TrackFinder
Posted by wiene on Wed, 02 Aug 2006 08:38:18 GMT
View Forum Message <> Reply to Message

Hi Jason,

as Martin already wrote, one will try one's best to separate double pulse candidates. But even the best separation algorithm will have its limits. We have to find a way to make the algorithm robust enough so that inseparable pulses have only a weak influence on the performance.

Why do you need the time of the maximum pulse contributing to a track for the likelihood fit?

Cheers, Peter

Subject: Re: TrackFinder
Posted by jabernathy on Wed, 02 Aug 2006 16:52:11 GMT
View Forum Message <> Reply to Message

wiene wrote on Wed, 02 August 2006 01:38Hi Jason,

Why do you need the time of the maximum pulse contributing to a track for the likelihood fit?

When we have a row of signals on pads we determine which pad the track most likely went through by finding the highest peak in the row which contributed to this Track. Then, knowing the

time of the highest peak we integrate the charge in each pad in the row over a small interval around that time.

It reduces the affect multiple signals and noise has on the collected charge by localizing the data sample.

Unfortunately it only works if you can either separate the signal into multiple pulses (one for each Track) or you know the estimated time the Track crossed the pad.

killenberg wrote on Wed, 02 August 2006 00:57

I think the idea was to separate a double pulse into two, and then two hits are reconstructed from the separated double pulses. But I don't know how to disentangle how much charge was in which pulse/hit. As far as I know, none of the existing algorithms do this separation.

That solution would make the TrackFitter easier for sure (I think Dean Karlen might do something like that in JTPC - I'll have to check).

If it's not possible to create two TrackerPulses because the signals are too hard to separate could it be possible to use two TrackerHits and then set the time for each hit to be the estimated peak time of each Track in the signal?

continued...

Would it be possible to have something like a track finder which produces a double track candidate, containing all hits from both tracks, and then do one likelihood fit for both tracks?

Theoretically I don't see anything wrong with that. Double tracks could just be passed as a single Track with 2 tracks returned in getTracks().

To sum it up, perhaps if the signals are separated enough in the TrackerData they can be split into two TrackerPulses / TrackerHits and the TrackFitter could just do a single Track fit to each. However if it is too difficult to determine how much charge to assign to each TrackerPulse the TrackFitter could use the double-fit technique.