Subject: riemann track finding
Posted by Sebastian Neubert on Wed, 23 Jan 2008 17:07:43 GMT
View Forum Message <> Reply to Message

Here are some references to the riemann tracking.
he root file contains a 3D diagram of the hits on the riemann sphere. MVD and TPC included.
Play around to see the planes!

For code look at the tpc/tpcreco/PndTpcRiemannTrackingTask and classes used by this.

Cheers! Sebastian.

File Attachments
1) riemann1.root, downloaded 480 times
2) extended_riemann.fruehwETstrandl.NIM.pdf, downloaded 663 times
3) error_riemann.fruehwETstrandl.NIM.pdf, downloaded 618 times

Subject: Re: riemann track finding
Posted by Tobias Stockmanns on Thu, 24 Jan 2008 16:15:18 GMT
View Forum Message <> Reply to Message

Hallo Sebastian,

hast Du etwas dagegen, wenn ich den Riemann-Trackfinder auf eine etwas allgemeinere Basis stelle und eine Kopie in PndTools erstelle? Oder hast Du Lust (und Zeit) das selbst zu machen? (Was mir natürlich viel lieber wäre.)

Es wäre auch sehr schön, wenn Du mir den Code schicken könntest, den Du geschrieben hast, um den MVD einzubauen.

Viele Grüße aus Groningen

Tobias

Subject: Re: riemann track finding
Posted by Sebastian Neubert on Thu, 24 Jan 2008 16:23:20 GMT
View Forum Message <> Reply to Message

Hi Tobias!

Since this trackfinder is one major part of my thesis I want to be involved!
I would be happy to put it onto more general feet together with you! Let's do it together - what do you think?

I will put some safety switches into the code with the MVD and submit it. Then you can have a

look and we can discuss the code.

There is much room for improvement still! Next thing I wanted to tackle was error treatment, which I omitted for the time being. But also the general structure is a bit cumbersome at some points. I would be happy to discuss this!!!

Cheers! Sebastian.

Subject: Re: riemann track finding
Posted by Tobias Stockmanns on Thu, 24 Jan 2008 16:29:34 GMT
View Forum Message <> Reply to Message

Hi Sebastian,

you are right! In the forum I should write in English.

I would be happy to do it together with you.

I will have a look into your code and think a bit about the structure of it. We should discuss it maybe next week.

Cheers,

Tobias

Subject: Re: riemann track finding
Posted by Sebastian Neubert on Mon, 28 Jan 2008 10:51:16 GMT
View Forum Message <> Reply to Message

Hi!

I have protected the RiemannTrackingTask against missing MVD hits. So we can start porting and generalizing.

We should start with the PndTpcRiemannHits and PndTpcRiemannTracks. Can you have a look and give me some feddback on these classes. Do you understand what's going on?

Tobias, can you copy them to the PndTools and start renaming?

There is a tpcrecp/test directory with a testing class. We should have something like this also for the general version!

cheers! Sebastian.

Subject: Re: riemann track finding

Posted by Tobias Stockmanns on Mon, 28 Jan 2008 11:37:58 GMT
View Forum Message <> Reply to Message

Hi Sebastian,

I can start porting the PndTpcRiemannHits and Tracks to PndTools. I think that I understand what is going on at least in these classes.

I just have a question concerning the projection. According to the paper you have posted, they suggest to project to a paraboloid but (I thin) in the Hit class you projection goes to a sphere. Shall we change this?

Cheers!

Tobias

Subject: Re: riemann track finding
Posted by Sebastian Neubert on Mon, 28 Jan 2008 12:19:16 GMT
View Forum Message <> Reply to Message

Hi!

In their first publication the authors use the rieman sphere. They later switch to the paraboloid because then the trafo becomes simpler and also the error claculation is a bit simplified.

Maybe it is a good idea to follow this! And when we are modifiying things in that direction we should also include the errors. Maybe this is a good starting point. Lets go to the paraboloid form and add the errors.

Can we somehow subdivide the work?

Sebastian.

Subject: Re: riemann track finding
Posted by Tobias Stockmanns on Mon, 28 Jan 2008 13:04:50 GMT
View Forum Message <> Reply to Message

Hi Sebastian,

at the moment I am still studying, how the Riemann mechanism works in detail and how errors are handled.

Do you have any suggestions how to split the work?

Cheers,

Tobias

Subject: Re: riemann track finding
Posted by Tobias Stockmanns on Mon, 28 Jan 2008 14:51:07 GMT

Hi Sebastian,

just for my understanding:

- You use two methods to determine if a hit belongs to a track:

ProximityHTCorrelator
RiemannHTCorrelator

The first uses only the distance between the last point in a track and the new point and is only used for less than 5 points in a track. The second checks if the new hit is close to the last hit in the track (as the first), and in addition checks the distance to the fitted plane and if the z-Position is reasonable.

After a new hit is added to a track the track is refitted.

The biggest problem I see at the moment for the MVD is the determination of the start plane, because the argument that hits belonging to a track are close together (like in the TPC) is not true for the MVD.
Either one tests all combination of three points to determine the plane and searches for at least one additional point or one has to use additional geometry information or use the planes determined by the TPC.

How would this look like for the STT?

Cheers,

Tobias

Subject: Re: riemann track finding
Posted by Tobias Stockmanns on Mon, 28 Jan 2008 15:36:30 GMT

Hi Sebastian,

I have moved PndTpcRiemannHit and ...Track to PndTools in a new folder riemannfit (not in svn yet). In addition I removed Tpc out of the names.

The class PndTpcRiemannHit contains a member _cluster of type PndTpcCluster, which has to be removed. I suggest to replace it by a CbmHit, which should be the base class for all data containing 3D hit information.

Cheers,

Tobias

Subject: Re: riemann track finding
Posted by Sebastian Neubert on Tue, 05 Feb 2008 09:32:00 GMT
View Forum Message <> Reply to Message

Hi Tobias!

The _cluster member is for bookkeeping only. So an index would be ok too I guess.

I do not agree that we need a base class for space points!
If so then this should be build around TVector3. I am not happy about the CbmHit construction.
Anyhow we should restrict the use of inheritance wherever possible and I do not see a reason
why to use it here.

Concerning the problem of the starting plane I probably have a solution for this but I need
some more time. As you said: You can always define a plane through 3 points! I will call this
the TripletRiemannTrackFinder Of course you will get ghost tracklets in the MVD!

Cheers! Sebastian.

Subject: Re: riemann track finding
Posted by StefanoSpataro on Tue, 05 Feb 2008 10:04:04 GMT
View Forum Message <> Reply to Message

Well,
CbmHit has, as private members:

  Int_t   fDetectorID;   // Detector unique identifier
  Double32_t fX, fY, fZ;   // Position of hit [cm]
  Double32_t fDx, fDy, fDz;   // Errors of position [cm]
  Int_t   fRefIndex;   // Index of CbmMCPoint for this hit

so a sort of TVector3 plus errors and two indexes.
I think it should be the best candidate as standard for 3d points. What do you have against it?

Subject: Re: riemann track finding
Posted by Sebastian Neubert on Tue, 05 Feb 2008 11:08:19 GMT
View Forum Message <> Reply to Message

Hi!

1) I simply do not understand why we need a standard hit. Maybe you can explain.

2) If we talk about errors, this makes only sense if we have an error MATRIX. And we agreed
that the object that is to be used for such kinds of applications is the RecoHit of genfit!

3) That CbmHit is A KIND OF TVector3 is exactly what I do not like. Would it use the TVector3
as a member directly one would have access to all the nice features of TVector3 for example in
a Draw-Command. Anyway CbmHit can calculate a TVector3 with the Position method, so why

not make a member variable of type TVector3 instead of three doubles??

Cheers! Sebastian.

Subject: Re: riemann track finding
Posted by StefanoSpataro on Tue, 05 Feb 2008 11:25:21 GMT
View Forum Message <> Reply to Message

Quote:1) I simply do not understand why we need a standard hit. Maybe you can explain.

If you have a base class for the points, you do not have to care about which kind of detector is giving you a point (and use its Get functions). You take an object of (CbmHit*) kind whatever detector is providing you this information.

Quote:2) If we talk about errors, this makes only sense if we have an error MATRIX. And we agreed that the object that is to be used for such kinds of applications is the RecoHit of genfit!

I can agree, but one should provide a standard for all kinds of possible algoryths that should be implemented for pattern recognition. Maybe in the future one could need even the error (size of the cluster? size of the mvd strip? size of the straw tube?) for pattern recognition.

Quote:3) That CbmHit is A KIND OF TVector3 is exactly what I do not like. Would it use the TVector3 as a member directly one would have access to all the nice features of TVector3 for example in a Draw-Command. Anyway CbmHit can calculate a TVector3 with the Position method, so why not make a member variable of type TVector3 instead of three doubles??

The TVector3 has nothing that links the points to the track (or detector hit) that has produced them. So when you want to evaluate efficiency/purity a simple TVector3 is maybe too much simple, and does not help you. The CbmHit has more space inside, and it contains even the TVector3, so you have already all its features by a simple Get function.

But this is just my opinion.

Subject: Re: riemann track finding
Posted by Sebastian Neubert on Tue, 05 Feb 2008 11:34:47 GMT
View Forum Message <> Reply to Message

Quote:If you have a base class for the points, you do not have to care about which kind of detector is giving you a point (and use its Get functions). You take an object of (CbmHit*) kind whatever detector is providing you this information.

Especially for the Pattern Recognition it matters which detector gave you the point. I would guess that for most detectors the pattern recognition will rely heavily on the specific features of that detector which might even include a special coordinate system. There are few detectors which really measure space points!

Quote:I can agree, but one should provide a standard for all kinds of possible algorythms that should be implemented for pattern recognition. Maybe in the future one could need even the error (size of the cluster? size of the mvd strip? size of the straw tube?) for pattern recognition.

I perfectly agree! Error treatment is one of the things we want to add next. So a simple TVector3 is not sufficient. Concerning the connection with the McTruth this is more complicated as we havee already discussed on a few occasions.
In a full treatment you cannot make a bijective mapping of reconstructed hits to Monte Carlo hits.

Cheers! Sebastian.

Subject: Re: riemann track finding
Posted by Sebastian Neubert on Tue, 05 Feb 2008 11:37:58 GMT
View Forum Message <> Reply to Message

But let's get back to the RiemannTracker.

I have some stomach-ache with the CbmHit - but maybe I am just suffering from NIHS.

But there is another issue - The RiemannHit is a very special one! It does not live in our 3D-Space but on the Riemann-Sphere. I am not sure that we should mix these two coordinate systems!

Cheers! Sebastian.

Subject: Re: riemann track finding
Posted by Tobias Stockmanns on Wed, 06 Feb 2008 08:05:44 GMT
View Forum Message <> Reply to Message

I am sorry that I did not take part in the discussion about CbmHit but I was busy in the last couple of days with celebrating Karneval .

I see two different points for the PndRiemannHit:

 Do we need a constructor and a Set function taking a pointer to CbmHit as parameter?
 Do we need a data member CbmHit* inside PndRiemannHit which points back to the point the parameters for the RiemannHit were comming from?

For the first point I would clearly say yes. Here I have the same opinion as Stefano. CbmHit should be the base class for all data classes which give you 3D hit information. So a set function could be used by various subdetectors. I am not really happy with the implementation of CbmHit but this is a different topic.

For the second point I am not sure if it is really needed.

Maybe we can arrange an EVO meeting about Riemann-Tracking? I think it is easier to discuss these things directly instead of using a forum.

Alaaf (already over again   ),

Tobias

Subject: Re: riemann track finding
Posted by Sebastian Neubert on Wed, 06 Feb 2008 09:01:16 GMT
View Forum Message <> Reply to Message

Hi Tobias!

We need such a constructor I agree. Let's take the CbmHit as an interface.

I also agree with your second poin. We do not need the member pointer. This is just for book-keeping. Some kind of index-system should also work!

An EVO meeting would be a good idea. This week I am on vacation though. What about next wednesday?

Cheers! Sebastian.

Subject: Re: riemann track finding
Posted by Tobias Stockmanns on Wed, 06 Feb 2008 09:42:46 GMT
View Forum Message <> Reply to Message

Wednesday in the morning (9:00 - 12:00) is fine with me, the rest of the day I have other meetings.

Monday or Tuesday would be fine with me, too.

Tobias

Subject: Re: riemann track finding
Posted by Sebastian Neubert on Tue, 12 Feb 2008 13:30:09 GMT
View Forum Message <> Reply to Message

Ok, let's meet tomorrow 10.00h on EVO!

Subject: Re: riemann track finding
Posted by Tobias Stockmanns on Tue, 12 Feb 2008 13:36:07 GMT
View Forum Message <> Reply to Message

Ok, I will book a room!

Subject: Re: riemann track finding
Posted by Tobias Stockmanns on Tue, 12 Feb 2008 13:44:47 GMT

View Forum Message <> Reply to Message

I have booked the room "Riemann Tracking" from 9:30 - 11:30 for our round table. The password is "TPC"

The official start of the meeting is 10:00 o'clock.

Until tomorrow

Tobias