

EXPERT ROOT - Developing #65

BeamDet поиск трека

07/21/2017 08:07 AM - Vitaliy Schetinin

Status:	Закрыта	Start date:	07/21/2017
Priority:	Низкий	Due date:	
Assignee:	Mikhail Kozlov	% Done:	0%
Category:	BeamDet	Estimated time:	0.00 hour
Target version:	v-0.4		
Description			
<p>Необходимо разработать класс ERBeamDetTrackFinder, унаследованный от FairTask. Задача класса - создание объекта ERBeamDetTrack по данным на MWPC станциях.</p> <p>Предполагаем, что события с множественностью больше единицы мы убрали на этапе диджитализации. То есть в коллекции MWPCDigi должно быть 4 элемента. digi должны принадлежать разным станциям. Это надо проверить и в противном случае кинуть сообщение.</p> <p>ERBeamDetTrack будет состоять из трех точек в пространстве. две из них получены из MWPC, третья из предположения, что трек прямолинеен и знания координаты мишени. (fX1,fY1, ... fZ2, fXt, fYt, fZt t - target)</p> <p>Так как мы знаем, что ERBeamDetTrack точно один в событии, его не надо хранить как один элемент в TClonesArray. Интерфейс ioman->Register("BeamDetTrack.", "BeamDet track", fBeamTrack, kTRUE); позволяет хранить одиночные объекты, наследованные от TNamed. Обрати внимание на точку в конце названия ветки. Она необходима для корректной работы TBrowser.</p> <p>Знания о геометрии детектора необходимо инкапсулировать в отдельный класс - ERBeamDetSetup(см ERNeuRadSetup). Туда же нужно убрать процедуру получения по номеру проволоочки координаты в пространстве. Одной из ключевых фишек FairRunTimeDB является то, что она записывает в par.root геометрию текущей симуляции. Оттуда мы ее достаем к примеру в eventDisplay. Во время оиджитализации и любой другой задачи геометрия также доступна через глобальный указатель gGeoManager. В NeuRad видно как по нему можно определить всю текущую геометрию исходя только из преполодения о том как объемы вложены друг в друга и как называются. Таким образом мы избавляемся от зависимости кода от текущей конфигурации детектора.</p> <p>Туда же потом будут инкапсулированы знания о электронике на каждом канале считывания. Как это сейчас сделано в NeuRad. Но это потом.</p>			

History

#1 - 07/21/2017 09:37 PM - Vitaliy Schetinin

Посмотрел файлы от которых ты решил отталкиваться. Обрати внимание, что в данном случае мы пропускаем такую сущность как хит. Которая была актуальна в muSi.

#2 - 07/21/2017 09:40 PM - Vitaliy Schetinin

Ржавый алгоритм, который раньше делал эту логику описывается так:

1) Он просто по номеру проволоочки восстанавливает четыре отдельные координаты: xmw1, ymw1, xmw2, ymw2.

2) Дальше собирает из них две точки в пространстве:

VmwFa(xmw1,ymw1,0)

VmwCl(xmw2,ymw2,расстояние между MWPC)

3) Формирует вектор пучка

Vbeam = VmwCl - VmwFa

4) Есть координата Zdist

Zdist = MWclosDist - header->UpMat.MWXYdist/2.;

Если я правильно понял это середина отрезка между X и Y плоскостями ближайшей к мишени MWPC

5) Координаты на мишени:

xbt = xmw2 - Zdist*tan(Vbeam.Theta())*cos(Vbeam.Phi());

ybt = ymw2 - Zdist*tan(Vbeam.Theta())*sin(Vbeam.Phi());

То есть он вообще никак не учитывает то, что у нас есть смещение пучка между станциями X и Y. Ок. Это вроде также как обсуждали в последний раз.

6) Отбор событий, попавших в мишень. Я думаю нам его тоже стоит сделать и не писать события не попавшие.

Как не писать события в выходной файл можно посмотреть в BeamDetCalibratorNew. FairRun->MarkFill()

#3 - 08/09/2017 08:08 AM - Vitaliy Schetinin

- Status changed from *Открыта* to *Закрыта*