Overview of software activities

Ch. Finck & Y. Dong

XVII FOOT Collaboration Meeting

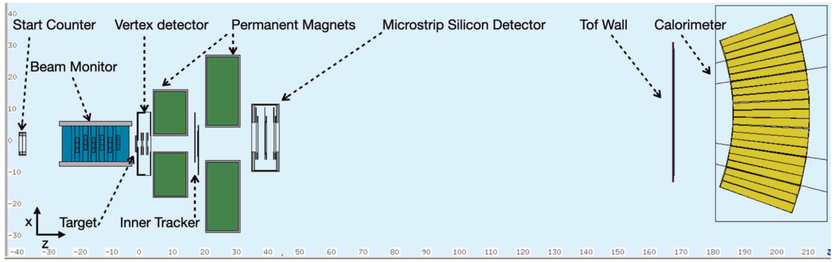
The last months..

➡.. have been “still” busy …

– Many bug fixes and code improvements:

- **Creating run file for CNAO2024**
- **New object reference framework**
- **Changing name of Position Getter (Roberto) avoid confusions**
- **Correct intersection method (Marco & Yun)**
- Add histograms update calibration/alignment parameters

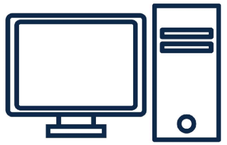
The last months..



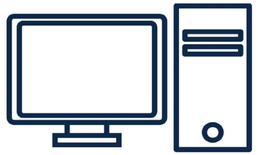
➡Analysis during data takings

– Reconstruction and analysis during data takings:

- DAQ FootBol1: take raw data
- FootBol4: copy data from FootBol1to Tier1
- GNAM plot from DAQ: online monitoring of raw measurements (no reconstruction)
- Tier1: data available for the whole collaboration + data processed with condor system, dedicated for offline analysis



DAQ
FootBol1



FootBol4

Tier1
(data storage
+Condor)

➡Since CNAO2024 FastDecode:

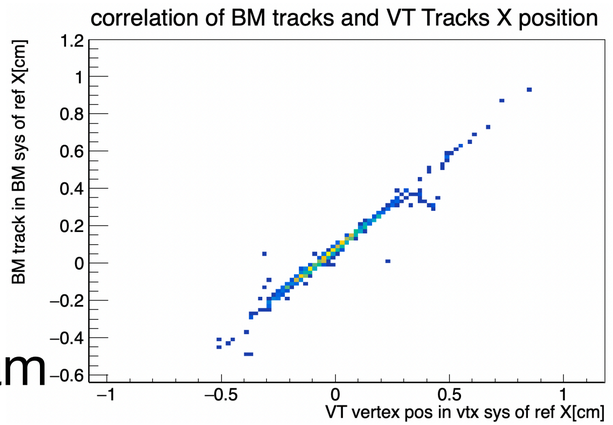
Online monitoring with SHOE reconstructed quantities and analysis

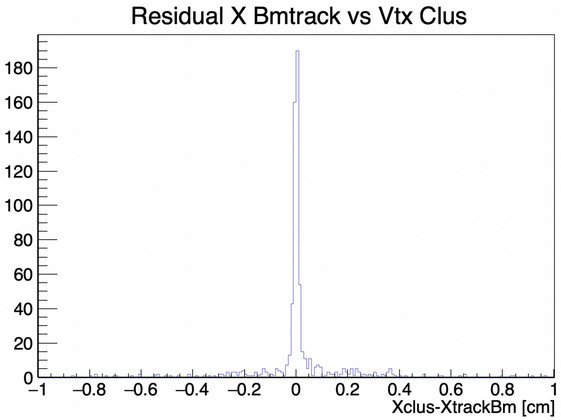
The last months..

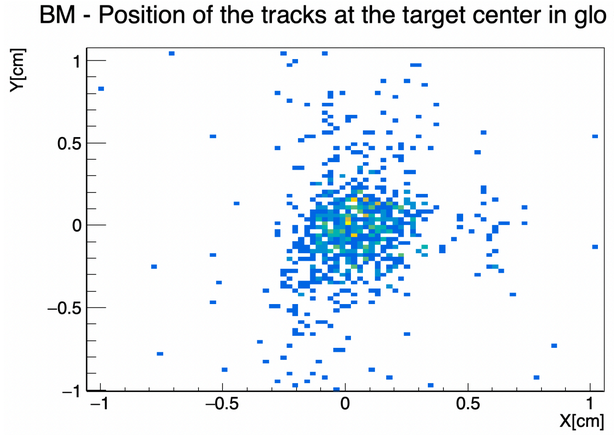
➡Fast Decoding

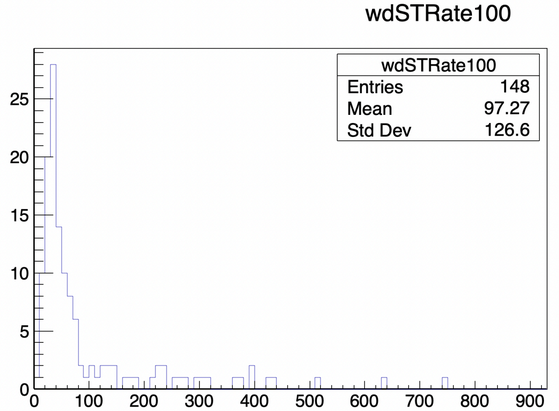
– Main properties of FastDecode:

- Perform analysis in the fastest way possible (exploit subfiles, produced ctrl plots)
- Make reconstruction plots, not provided by the gnam
  -Number of VTX vertex/tracks
  -Number of BM tracks
- Monitor inter-detectors quantities
  -check synchronisation
  -check correlations between detectors
  -VTX pile up (using BM or other detector)
- Used in CNAO2024 to
  -Check the VTX performances during data taking
  -Check the beam position both for the physics and Emulsion runs









The last months..

➡Towards next data taking

– In CNAO2024 the FastDecode monitoring worked, with few difficulties:

- The script developed to execute FastDecode on FootBol4 didn’t worked, processed the data with personal laptop
- People were forced to mount the NAS (if possible) or copy the data and process them manually —> if FootBol4 (or another PC) could be used.
- For the VTX analysis the synchronisation is required —> maybe we can develop a script that synchronise automatically all the data files as soon as they are copied on the NAS

– For the next data taking:

- Fully exploit fastDecode and onlineMonitor root gui to:
  -Process all the raw data file using FastDecode
  -Show the plots with a onlineMonitor gui (Thank to Zarrella)
- Improve the speed of FastDecode (skip some events, e.g.: events with too large hits)

Status (i)

- Data taking of CNAO2023/2024:
  - Calibration on-going BM, VT, IT, etc…
  - Geometry survey/debugging
  - Updated simulation parameters
  - Alignment of BM, VT, MSD and TW (ongoing)

Status (ii)

- Doxygen documentation ongoing, server hosted in Roma (ST, MSD, CA not done)
- Update doc automatically from master branch to baltig site
- Possibility to add the documentation directly in git under study

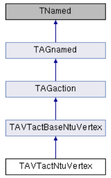
SHOE documentation

Main Page | Related Pages | Modules | Namespaces | Classes | Files

TAVTactNtuVertex Class Reference

NTuplizer for VTX vertices. More...

Inheritance diagram for TAVTactNtuVertex:



Public Member Functions

	TAVTactNtuVertex (const char *name=0, **TAGdataDsc** *p_ntutrack=0, **TAGdataDsc** *p_ntuvertex=0, **TAGparaDsc** *p_config=0, **TAGparaDsc** *p_geomap=0, **TAGparaDsc** *p_geomapG=0, **TAGdataDsc** *p_bmtrack=0)
virtual	**~TAVTactNtuVertex** () Destructor.
void	**SetEps** (Double_t q) Set tolerance.
Double_t	**GetEps** () const Get tolerance.
void	**SetMinimumZ** (Double_t min) Set miminum Z for vertices search.
Double_t	**GetMinimumZ** () const Get miminum Z for vertices search.
void	**SetMaximumZ** (Double_t max) Get maxinum Z for vertices search.
Double_t	**GetMaximumZ** () const Set maxinum Z for vertices search.

Public Member Functions inherited from **TAVTactBaseNtuVertex**

Public Member Functions inherited from **TAGaction**

Public Member Functions inherited from **TAGnamed**

Detailed Description

NTuplizer for VTX vertices.

Constructor & Destructor Documentation

TAVTactNtuVertex()

```
TAVTactNtuVertex::TAVTactNtuVertex ( const char *     name = 0,
                                     TAGdataDsc *     pNtuTrack = 0,
                                     TAGdataDsc *     pNtuVertex = 0,
                                     TAGparaDsc *     pConfig = 0,
                                     TAGparaDsc *     pGeoMap = 0,
                                     TAGparaDsc *     pGeoMapG = 0,
                                     TAGdataDsc *     pBmTrack = 0
                                   )
```

Default constructor.

Parameters

[in]	**name**	action name
[in]	**pNtuTrack**	track container descriptor
[out]	**pNtuVertex**	vertex container descriptor
[in]	**pConfig**	configuration parameter descriptor
[in]	**pGeoMap**	geometry parameter descriptor
[in]	**pGeoMapG**	target geometry parameter descriptor
[in]	**pBmTrack**	input BM track container descriptor

Ongoing

➡ New commit policy:

– Branches:

- Development branch: *newgeom_v1.0* and production branch: *master*
- No restriction on commits, can lead to some issues (shoe crashing after some commits)

➡ Having only one development branch: *main*

– Commits:

- All developments done in dedicated branches
- Merge with the *main* branch only by “librarian” after checking the code
- Open a merge request with pipeline (testing code, e.g.: compilation, macros…)

– Production branch:

- Tagged branch will be done in a given frequency, date stamped (to keep track)

‘final’ considerations

➡ Please start to use dedicated branch for your developments

➡ Please do not push your changes in the *main* branch, ask (still we have no pipeline yet) the “librarians” to check you code before merging

Pb of recurrent man power, need at least one guy per detector

Calibration/alignment done by the core team !

➡ We’d like also to add one more deputy in the SW coordination:

- Roberto Z., for all the work he has done in the last years

What's ahead of us..

- Improve documentation:
  - Update the twiki page, now all the shoe developers can do it
  - We still need to update the Doxygen documentation

- Analysis of GSI2021 - HIT2022 - CNAO2022/23/24 data

Thanks for your attention