Introduction in TBmon – A Testbeam Data Analysis Software

Reconstruction and Analysis Workshop

André Rummler

GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

Outline

- Installation and basic configuration
- Available Documentation
- General Tbmon usage
  - Workflow
  - New data blocks
- Output of reconstruction / Tbtrack file format
- Functionality
- A look under the hood
  - Main routines
  - UML
  - Modularity
- Outlook / Current Development / Alternatives

Installation and basic configuration

- Requirements:
  - CERN root – version does not really matter
  - C++ compiler, make system, doxygen if documentation
- Download it from CERN svn:
  svn co svn+ssh://svn.cern.ch/reps/atlasibltbsw/tbmon/ tbmon
- cd tbmon/trunk/ (usually trunk is used, read changelog!)
- cp siteconfig.h.example siteconfig.h
- Edit at least three lines in siteconfig.h:
  line 15: trySet(config.outPath, (char*) "/path/to/tbAnalysis/output");
  line 20: trySet(config.tbslot, (char*) "eudetfeb2011");
  and e.g. trySet(config.dataPath, (char*) "/path/to/tbAnalysis/data/cern_2011_sep_ibl/tbtrack");
- make

Available Documentation

- Unfortunately a bit lacking
- “doc” subdirectory: doxygen base, old “mainpage”
- Doxygen documentation effort ongoing
- New wiki, bug tracker, etc.:
  https://svnweb.cern.ch/trac/atlasibltbsw/wiki
- Some effort to document basic plots on private PPS twiki page:
  https://twiki.cern.ch/twiki/bin/view/Atlas/PhysicsAnalysis
  will be moved (hopefully) soon to the trac wiki
- “the code”

TBmon usage / workflow

- ./tbmon -help
- ./tbmon -s 61527 -a hotpixelfinder -c eudetIBLsep2011
- Typical order:
  - hotpixelfinder
  - checkalign
  - getetacorr
  - checkalign
  - all other analyses
- The first are analysis which produce output file which have to be set up in the driver.cc afterwards and tbmon needs to be re-build before the next step. Please take care of the location of the output files
- e.g. bool masknoisyanddeadpixels = false;
- Useful switches: -f, -v debug

New data blocks

- Siteconfig.h
  //default datapath for September 2011 PPS EUDET TB

  #define SITEEUDETPPSSEP2011_SET

  void siteEudetPPSSep2011(TbConfig &config){
  trySet(config.dataPath, (char*)
  "/path/to/tbAnalysis/data/cern_2011_sep_apix/tbtrack");
  }
- Driver.cc
  void EudetIBLSeptember2011(TbConfig &config) {…}

  } else if (strcmp(config.tbslot, "eudetIBLsep2011") == 0) {
  EudetIBLSeptember2011(config);

Output of reconstruction / tbtrack file format

- Standard root file with trees and native data types
- 2d track point in local coordinates
- Hit information
- Cluster information
- Clustering is redone

Tree/Branch	Data type	Description
euhits		
nHits	Int	Number of hits in this event
xPos	std::vector<double>	Global x coordinate [mm]
yPos	std::vector<double>	Global y coordinate [mm]
zPos	std::vector<double>	Global z coordinate [mm]
clusterId	std::vector<int>	ID of the corresponding cluster
sensorId	std::vector<int>	ID of the corresponding sensor
zspix		
nPixHits	int	Number of raw hits in this event
euEvt	int	Current event number
col	std::vector<int>	column of raw data hit
row	std::vector<int>	row of the raw data hit
tot	std::vector<int>	TOT of the raw data hit
lv1	std::vector<int>	LVL1 value of the raw data hit
iden	std::vector<int>	ID of the sensor
chip	std::vector<int>	ID of the sensor in the MCC board
clusterId	std::vector<int>	ID of corresponding cluster
eutracks		
nTrackParams	int	Number of parameters for estimation
euEvt	Int	Event number
xPos	std::vector<double>	The fitted x position [mm]
yPos	std::vector<double>	The fitted y position [mm]
dxdz	std::vector<double>	The fitted derivate $\partial x/\partial z$
dydz	std::vector<double>	The fitted derivate $\partial y/\partial z$
trackNum	std::vector<int>	The track ID
iden	std::vector<int>	ID of the corresponding sensor
chi2	std::vector<double>	χ^2 of the track
ndof	std::vector<double>	tracks' degrees of freedom
euclusters		
euEvt	Int	Event number
size	std::vector<int>	Number of pixels in a cluster
sizeX	std::vector<int>	Cluster width in x [pixels]
sizeY	std::vector<int>	Cluster width in y [pixels]
posX	std::vector<int>	Position of the cluster in x [pixels]
posY	std::vector<int>	Position of the cluster in y [pixels]
charge	std::vector<int>	Sum charge of the cluster [TOT]
iden	std::vector<int>	ID of the corresponding sensor
ID	std::vector<int>	ID of the cluster

Functionality

- Read file and create objects
- Referencing
- Matching (for criterium see code): 1.5 * pitch
- Hotpixelfinder: out of time criterium
- Mostly modular, easy to add additional analysis:
  init, event, finalize
-

A look under the hood

- Most important objects:
  - Looper.cc
  - TbConfig.cc
- Those two are calling each other several times
- Own event loop, not the root mechanism
- Certain limitations and inefficient code

Outlook / Current Developments / Alternatives

- Lots of work is currently done:
  - update often and read the change log
  - If something stops working please tell us immediately
- Current goals:
  - High eta analysis (remove cuts, different cluster center algorithms, extended input format, optimization routine etc.)
  - Improve output (everything into one root file)
  - General clean-up / Refactoring
  - Move constants out of source into configuration files
  - Better workflow (make the constant re-building superfluous)
  - Better processing speed: optimization and probably parallel execution
- Alternative tool developed in Dortmund for comparison purposes: TbTupleAna