Subject: PndTrack array available
Posted by Gianluigi Boca on Wed, 24 Nov 2010 00:24:11 GMT
View Forum Message <> Reply to Message

hallo,
I just updated the svn repository.
Now the PndTrack's are available out of the pattern recognition
with the STT alone and with the STT+MVD.
Just update the stt and the sttmvdtracking directories.
Cheers Gianluigi

Subject: Re: PndTrack array available
Posted by Gianluigi Boca on Wed, 24 Nov 2010 14:30:12 GMT
View Forum Message <> Reply to Message

Gianluigi Boca wrote on Wed, 24 November 2010 01:24hallo,
I just updated the svn repository.
Now the PndTrack's are available out of the pattern recognition
with the STT alone and with the STT+MVD.
Just update the stt and the sttmvdtracking directories.
Cheers Gianluigi

I forgot to mention that in order to use the Stt or Stt+Mvd real Pattern Recognition code you could have a Macro more or less like the following :

```
{
  // Verbosity level (0=quiet, 1=event level, 2=track level, 3=debug)
  Int_t iVerbose = 0;
  TString MCFile = "Mvd_Test.root";
  TString parFile = "Mvd_Params.root";
  Int_t nEvents = 0;
  // ----  Load libraries   -------------------------------------------------
  gROOT->Macro("$VMCWORKDIR/gconfig/rootlogon.C");

//  gSystem->("libSttMvdTracking");

  // -----------------------------------------------
  // Output file
  PndFileNameCreator creator(MCFile.Data());
  TString DigiFile = creator.GetDigiFileName(false).c_str();
  TString outFile = creator.GetRecoFileName(false).c_str();

  std::cout << "MCFile  : " << MCFile.Data()<< std::endl;
  std::cout << "DigiFile: " << DigiFile.Data()<< std::endl;
```

```
  std::cout << "RecoFile: " << outFile.Data()<< std::endl;

  // -----   Timer   --------------------------------------------------------
  TStopwatch timer;
  timer.Start();

  // -----   Reconstruction run   -------------------------------------------
  FairRunAna *fRun= new FairRunAna();
  fRun->SetInputFile(MCFile);
  fRun->AddFriend(DigiFile);
  fRun->SetOutputFile(outFile);

  // -----  Parameter database   --------------------------------------------
  FairRuntimeDb* rtdb = fRun->GetRuntimeDb();
  FairParRootFileIo* parInput1 = new FairParRootFileIo(kTRUE);
  parInput1->open(parFile.Data(),"UPDATE");
  rtdb->setFirstInput(parInput1);

  Bool_t kParameterMerged=kTRUE;

  // -----    Default MVD hit producer   ------------------------------------------

  PndMvdClusterTask* mvdmccls = new PndMvdClusterTask();
  mvdmccls->SetVerbose(iVerbose);
  fRun->AddTask(mvdmccls);

  //=============================================
  // ======   Riemann finder MVD             ======
  // =============================================

  // -----    MVD hit producer
  PndMvdRiemannTrackFinderTask* mvdTrackFinder = new
PndMvdRiemannTrackFinderTask();
  mvdTrackFinder->SetVerbose(iVerbose);
  mvdTrackFinder->SetMaxDist(0.05);
  fRun->AddTask(mvdTrackFinder);

 // -----  end  MVD  Rieman finder

  // TRACK FINDING ====================
  // OUTPUT: PndTrackCand                  -> STTTrackCand
  // trackfinding ....

PndSttTrackFinderReal* sttTrackFinder = new PndSttTrackFinderReal(iVerbose);
  PndSttFindTracks* sttFindTracks = new PndSttFindTracks("Track Finder", "FairTask",
sttTrackFinder, iVerbose);
  sttFindTracks->AddHitCollectionName("STTHit", "STTPoint");
  fRun->AddTask(sttFindTracks);


// ================================================
//------------------------ stt-mvd   task ---------
```

```
  PndSttMvdTracking *  SttMvdTracking = new PndSttMvdTracking(iVerbose);
  fRun->AddTask(SttMvdTracking);


  rtdb->setOutput(parInput1);
  // =====    End of HitProducers        =====
  // ========================================

  // -----   Intialise and run --------------------------------------
  fRun->Init();

  fRun->Run(0,nEvents);
  //  ------------------------------------------------------------------------

//  SttMvdTracking->WriteHistograms();

 rtdb->saveOutput();
 rtdb->print();
  // -----   Finish   ---------------------------------------------------
  timer.Stop();
  Double_t rtime = timer.RealTime();
  Double_t ctime = timer.CpuTime();
  cout << endl << endl;
  cout << "Macro finished successfully." << endl;
  cout << "Output file is "    << outFile << endl;
  cout << "Parameter file is " << parFile << endl;
  cout << "Real time " << rtime << " s, CPU time " << ctime << " s" << endl;
  cout << endl;
  //  ------------------------------------------------------------------------


}
```

Also, the name of TClonesArray containing the PndTrack's is :

PndSttMvdPndTrackArray   for the array of PndTrack from the Stt+Mvd PR

Also a PndTrackCand array is available from the Stt+Mvd PR with name :

PndSttMvdPndTrackCandArray

Gianluigi