Table of Contents

Trajectory Filtering Algorithm....1
Goal of the page....1
Contacts....1
Introduction....1
Usage of the algorithm....1
Software architecture....1
Testing....1

SWGuideTrajectoryFiltering from Combinatorial Track Finder....2

Available....3

How to in 18X-on....5
Putting a in the....5
Using the....5
Using dedicated ESProducer....5
Accessing a....5
Configuring the....6
For "simple""....6
For more complex....6

HOW TO FIX YOUR PACKAGE DUE TO RECENT FILTERING CHANGES....7
List of package with a critical need to adapt their configuration files....7
How to adapt your configuration file....7
List of package with a critical need to adapt their code....8
How to adapt in the code....8
Review status....8

Trajectory Filtering Algorithm

Complete: 

Goal of the page

State what the reader is expected to learn from this page, e.g., This page is intended to familiarize you with ...

Contacts

Introduction

Usage of the algorithm

Explain what a user needs to know when using the algorithm:

- when to use this algorithm
- configuration sequences
- default parameters and how (and why) to change them
- products

Software architecture

Explain the design of the algorithm to ensure that the software can be maintained once the existing authors have left:

- design arguments - why this is done as it is done
- description of algorithm.

Testing

SWGuideTrajectoryFiltering from Combinatorial Track Finder

The CkfTrajectoryBuilder (and the GroupedCkfTrajectoryBuilder) used to use TrajectoryFilter and hard coded selection to control the Trajectory that needs to be continued during pattern recognition.

TrajectoryFilter has been made configurable in CkfTrajectoryBuilderBase so that the user can specify different filter and write its own filters. The TrajectoryFilter are accessible using a PluginFactory of the Event Setup.

TrajectoryFilter has two principal member functions

- toBeContinued(...) which decide if a given trajectory should be used at the next step of the combinatorial Kalman trajectory builder.
- qualityFilter(...) which decide where a stop trajectory should be kept as a good trajectory.

Available

- CompositeTrajectoryFilter:
  Is a fully configurable TrajectoryFilter which *toBeContinued* decision will be false if any of the component filter *toBeContinued* function returns false. *qualityFilter* follows the same rule.

- CompositeLogicalTrajectoryFilter: **under developement**
  Inherits from CompositeTrajectoryFilter, but the logic between component filters is configurable as well.

- CkfBaseTrajectoryFilter:
  The CkfBaseTrajectoryFilter has a concrete implementation of each of the required filters (seed later on): MaxHitsTrajectoryFilter, MaxLostHitsTrajectoryFilter, MaxConsecLostHitsTrajectoryFilter, MinHitsTrajectoryFilter, MinPtTrajectoryFilter and ChargeSignificanceTrajectoryFilter.

- MaxHitsTrajectoryFilter:
  Stops the trajectory building if the number of found hits has reach a given number.
  Does not contribute to *qualityFilter*.

- MaxLostHitsTrajectoryFilter:
  Stops the trajectory building if the number of lost hits on the trajectory has reach a given number.
  Does not contribute to *qualityFilter*.

- MaxConsecLostHitsTrajectoryFilter:
  Stops the trajectory building if the number of consecutive lost hits on the trajectory has reach a given number.
  Does not contribute to *qualityFilter*.

- MinPtTrajectoryFilter:
  Stops the trajectory building if the estimation of pT on the last measurement of the trajectory is below a given pT cut, at a given number of sigma times the error on pT.
  1/pT - nsigma*inversepT_error > 1/pTmin
  Does not contribute to *qualityFilter*, which means trajectory with pT less than the cut could end up it the final step of trajectories if no other filter rejects it.

- MinHitsTrajectoryFilter:
  Does not contribute to *toBeContinued*.
  Keeps the trajectory if it has a minimun of number of hits attached.

- ChargeSignificanceTrajectoryFilter:
  Filters on the internal consistency of a candidate concerning sign(q/p). If among the TrajectoryMeasurements of a candidate there are significantly negative **and** positive q/p estimates the building will be stopped and and the candidate will not be added to the result.

- RegionalTrajectoryFilter:
  a MinPtTrajectoryFilter based on the pt limit stored in a region.

- ThresholdPtTrajectoryFilter:
  Stops the trajectory building if the number of hits is above a given cut **AND** the estimation of pT on the last measurement of the trajectory is above a given pT threshold, at a given number of sigma times the error on pT:
  1/pT + nsigma*inversepT_error < 1/pTthreshold
  This filter **does** contribute to the *qualityFilter*: trajectory will be "kept" (however might be rejected by other filters) if it was stopped by this filter.

- ImpactParameterSignificanceTrajectoryFilter: **under developement**
  would stop the trajectory if a certain level of precision is reached on the estimation of the transverse impact point parameter.

- ClusterShapeTrajectoryFilter:
  stops the trajectory building if the geometrical shape of one of the pixel hits or the width of one of the strip hits is incompatible with the local direction of the trajectory.

How to in 18X-on

disclaimer The following instructions are indicators of what to do and might not work out of the box in the future due to version synchronization and code development.

If you need a dedicated TrajectoryFilter you should write a class which inherits from TrajectoryFilter and which overload the virtual member functions of TrajectoryFilter.

Putting a in the

Using the

If the constructor of your TrajectoryFilter as only a edm::ParameterSet in argument, you should declare your filter to the TrajectoryFilterFactory with

```
#include "FWCore/PluginManager/interface/ModuleDef.h"
#include "FWCore/Framework/interface/MakerMacros.h"
#include "FWCore/Framework/interface/ModuleFactory.h"
#include "FWCore/Framework/interface/ESProducer.h"

#include "TrackingTools/TrajectoryFitering/interface/TrajectoryFilterFactory.h"
#include "myCleaner.h"

DEFINE_EDM_PLUGIN(TrajectoryFilterFactory, myFilter, "myFilter");
```

You can then use the generic TrajectoryFilterESProducer with

```
es_module myFilterESproducer = trajectoryFilterESProducer from "TrackingTools/TrajectoryFiltering
replace myFilterESproducer.ComponentName = "myFilterName"
replace myFilterESproducer.filterPSet = {
      string ComponentType = "myFilter"
      string dummy = "myFilterParameter"
 }
```

Using dedicated ESProducer

If your TrajectoryFilter has more elaborate constructor, then you have to write a dedicated ESProducer to put your filter in **TrajectoryFilter::Record**, which the user is not supposed to be aware the concrete type of (TrackingComponentsRecord for the time being).

Accessing a

TrajectoryFilter object are put in the Event Setup in the record **TrajectoryFilter::Record** TrajectoryFilter can be retrieved from the EventSetup using

```
    std::string trajectoryFilterName = conf_.getParameter<std::string>("TrajectoryFilter");
    edm::ESHandle<TrajectoryFilter> trajectoryFilterH;
    es.get<TrajectoryFilter::Record>().get(trajectoryFilterName, trajectoryFilterH);
```

Or if your are using the CkfTrajectoryBuilder (or GroupedCkfTrajectoryBuilder)

```
     replace CkfTrajectoryBuilder.trajectoryFilterName = "myFilterName"
```

Configuring the

For "simple""

The configuration of the CompositeTrajectoryFilter is as follows

```
es_module myCompositeTrajectoryFilterESProd = TrajectoryFilterESProducer
{
     string ComponentName = "myCompositeFilterName"
     PSet filterPset =
     {
            string ComponentType = "CompositeTrajectoryFilter"
            VPSet filters =
            {
                  {
                 string ComponentType = "MinPtTrajectoryFilter"
                 double minPt = 1.0
                 double nSigmaMinPt = 5.0
                 int32 minHitsMinPt = 3
                  },
                  {
                      string ComponentType = "ThresholdPtTrajectoryFilter"
                 double thresholdPt = 10.0
                 double nSigmaThresholdPt = 5.0
                 int32 minHitsThresholdPt = 4
                  }
            }
     }
}
```

To configure a TrajectoryFilter that will stop trajectories below 1.0 GeV and above 10.0 GeV. Only trajectories above 10.0 GeV (at 5 sigma) will be kept in the final collection of trajectory, which might be interesting for HLT oriented algorithms.

For more complex

You can also configure a CompositeTrajectoryFilter as follows, in case the filters you want to combine have constructor different than the base constructor (with a parameterSet).

```
es_module myCompositeTrajectoryFilterESProd = CompositeTrajectoryFilterESProducer
{
     string ComponentName = "myCompositeFilterName"
     vstring filterNames = { "ckfBaseTrajecotyFilter", "myOtherFilter" }
}
```

To combine the CkfBaseTrajectoryFilter and another filter. **NOTE**: ESProducer must be configured for **both** of "ckfBaseTrajectoryFilter" and "myOtherFilter" in addition, since CompositeTrajectoryFilterESProducer retrieves them from the TrajectoryFilter::Record itself.

HOW TO FIX YOUR PACKAGE DUE TO RECENT FILTERING CHANGES

The changes made to the tracking code is very likely to have broken your configuration files and/or your code. Here is a recipe to adapt to those unavoidable backward incompatibilities. In case of any problem with the implementation of these changes, please contact me (jean-roch vlimant) with Boris Mangano and Steve Wagner in Cc. I will help you as much as I can to make the transition in the smoothest way.

List of package with a critical need to adapt their configuration files

- Alignment/LaserAlignment
- AnalysisExamples/SiStripDetectorPerformance
- DQM/TrackerMonitorTrack
- HLTrigger/btau
- CosmicMuonGenerator
- RecoEgamma/EgammaElectronProducers
- RecoEgamma/EgammaPhotonProducers
- RecoLocalTracker/SubCollectionProducers
- RecoParticleFlow/PFTracking
- RecoPixelVertexing/PixelLowPtUtilities
- RecoTracker/CkfPattern
- RecoTracker/NuclearSeedGenerator
- VisReco/VisCustomTracker

How to adapt your configuration file

Your configuration file usually looks like

```
es_module myCkfTrajectoryBuilder = CkfTrajectoryBuilder from RecoTracker/CkfPattern/data/CkfTraje
replace myCkfTrajectoryBuilder.ComponentName = "myCkfTrajectoryBuilder"
replace myCkfTrajectoryBuilder.ptCut = 2.0
replace myCkfTrajectoryBuilder.maxLostHit = 4
replace myCkfTrajectoryBuilder.alwaysUseInvalidHits = false

module myckfTrackCandidates = ckfTrackCandidates from "RecoTracker/CkfPattern/data/CkfTrackCandid
replace myckfTrackCandidates.TrajectoryBuilder = "myCkfTrajectoryBuilder"
```

and should look like

```
es_module myCkfTrajectoryFilter = trajectoryFilterESProducer from "TrackingTools/TrajectoryFilter
replace myCkfTrajectoryFilter.ComponentName = "myCkfTrajectoryFilterName"
replace myCkfTrajectoryFilter.filterPset.minPt = 2.0
replace myCkfTrajectoryFilter.filterPset.maxLostHits = 4

es_module myCkfTrajectoryBuilder = CkfTrajectoryBuilder from "RecoTracker/CkfPattern/data/CkfTraj
replace myCkfTrajectoryBuilder.ComponentName = "myCkfTrajectoryBuilder"
replace myCkfTrajectoryBuilder.trajectoryFilterName = "myCkfTrajectoryFilterName"
replace myCkfTrajectoryBuilder.alwaysUseInvalidHits = false

module myckfTrackCandidates = ckfTrackCandidates from "RecoTracker/CkfPattern/data/CkfTrackCandid
replace myckfTrackCandidates.TrajectoryBuilder = "myCkfTrajectoryBuilder"
```

- *ptCut* (renamed *minPt*) and *maxLostHits* are parameters of the default TrajectoryFilter defined by TrackingTools/SWGuideTrajectoryFiltering/data/TrajectoryFilterESProducer.cfi

(CkfBaseTrajectoryFilter)
- _alwaysUseInvalidHits_is a parameter of the CkfTrajectoryBuilder itself and not part of the filtering

For your convenience here is listed the parameters of the trajectorybuilder, **for which no change has to be made**

- CkfTrajectoryBuilder

```
int32  maxCand
double lostHitPenalty
bool intermediateCleaning
bool   alwaysUseInvalidHits
```

- GroupedCkfTrajectoryBuilder (in addition to those of CkfTrajectoryBuilder)

```
double foundHitBonus
bool   lockHits
bool   bestHitOnly
bool   requireSeedHitsInRebuild
int32  minNrOfHitsForRebuild
```

For your convenience, here is the list of parameters that define the filtering, **which are no longer to be defined to the TrajectoryBuilder** but to the TrajectoryFilter instead

```
int32  maxLostHit (to be replaced by maxLostHits)
int32  maxConsecLostHit (to be replaced by maxConsecLostHits)
int32  minimumNumberOfHits
double ptCut (to be replaced by minPt)
int32  maxNumberOfHits
double chargeSignificance
```

List of package with a critical need to adapt their code

- RecoMuon/L3MuonTrackFinder
- RecoTracker/DebugTools

How to adapt in the code

The only change is in the constructor of the CkfTrajectoryBuilder, GroupedCkfTrajectorBuilder and BaseCkfTrajectoryBuilder, which now takes one more argument "const TrajectoryFilter *"

Review status

Reviewer/Editor and Date (copy from screen)	Comments
KatiLassilaPerini - 29 Jan 2008	created template page

Responsible: ResponsibleIndividual
Last reviewed by: Most recent reviewer

This topic: CMSPublic > SWGuideTrajectoryFiltering
Topic revision: r10 - 2009-04-22 - JeanrochVlimant