

Table of Contents

Vertex Fitting	1
Goal of the page.....	1
The VertexFitter.....	1
Usage example.....	1
Implementation details.....	1
Vertex fitting algorithms.....	1
Track refitting after Vertex fit.....	2
The SimpleVertexTree.....	2
Usage of a TransientTracks.....	2
Review status.....	2

Vertex Fitting

Complete: 

Goal of the page

The task of Vertex Fitting is, given a set of tracks, to compute the best estimate of the vertex parameters (position, covariance matrix, constrained track parameters and their covariances) as well as indicators of the success of the fit (total χ^2 , number of degrees of freedom, track weights).

Contact persons: T.Speer, W.Waltenberger

The VertexFitter

The task of the *VertexFitter* is to control all the steps of the vertex fit from the input of the initial information to the output of the estimated quantities. Depending on the implementation of the concrete *VertexFitter*, the different objects which perform the different steps are either hard-coded or have to be given at construction time. This page describes all available vertex fitting algorithms.

The fitters can be used easily via the *ConfigurableVertexFitter*.

Usage example

A simple example of the use of a *VertexFitter* is given on the *WorkBook* page. A more detailed description is given here.

Implementation details

Implementation details, are given in the following pages:

- The main steps of the fit
- The main components of the fit
- Internal reconstruction objects used

Vertex fitting algorithms

Vertex fitting algorithms can be divided into least-squares algorithms and robust algorithms. In least-squares algorithms all tracks are used, with a weight 1. The *KalmanVertexFitter* is such a fitter. Robust fitters are able to downweight tracks, and are thus less sensitive to outliers (mismeasured tracks (type 1 outliers) or tracks from another vertex (type 2 outliers)). A common measure of the of the robustness is the break-down point, i.e. the fraction of outliers below which the vertex fit is not affected.

The fitters now available are:

- The *KalmanVertexFitter*: the simple least-squares algorithm
- The *AdaptiveVertexFitter*: iterative re-weighted *KalmanFitter* which down-weights tracks according to their distance to the vertex
- The *TrimmedVertexFitter*: conventional robust version of the *Kalman* fitter, which removes tracks incompatible with the vertex
- The *GaussianSumFitter*: fitter using the non-Gaussian distributions of measurement errors
- The *AdaptiveGsfVertexFitter*: a combination of the adaptive fitter and the Gaussian-sum fitter

Track refitting after Vertex fit

After a vertex fit, the track parameters can optionally be re-estimated at the fitted vertex position, using this position as a constraint. Usage and examples are given on this page.

The SimpleVertexTree

The classes SimpleVertexTree is provided to produce a simple TTree to analyse the results of a fitter

Usage of a TransientTracks

The most frequent use case of a VertexFitter is the computation of a vertex from an STL vector of TransientTracks.

Review status

Reviewer/Editor and Date (copy from screen)	Comments
Main.speer -17 Jul 2006	new page (Thomas Speer)
JennyWilliams - 28 Mar 2007	edited for reorganisation of swguide vertexing pages

Responsible: Main.speer

Last reviewed by: Most recent reviewer

This topic: CMSPublic > SWGuideVertexFitting

Topic revision: r19 - 2009-02-27 - ThomasSpeer



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback