RAVE - an Open, Extensible, Detector-Independent Toolkit for Reconstruction of Interaction Vertices

Wolfgang Waltenberger, Fabian Moser

Institute for High Energy Physics of the Austrian Academy of Sciences

USA, October 2006

Synopsis

A Toolkit is presented that reconstructs interaction vertices given a set of reconstructed tracks. The toolkit is generic enough to be embeddable in various environments. The code has already been run in LHC's CMS environment, as well as in the (European) ILC setup.
The toolkit features very modern "adaptive" reconstruction methods. It is written in C++, with a very simple API, but comes with Python and Java interfaces, as well.

The toolkit is complemented by a simple "sandbox" framework, that can simulate various experimental setups.

Origin

All algorithmic code comes from the CMS vertexing community,

Rave algorithms have remained and will continue to remain 100 % source code compatible with the CMS framework.

An as simple as possible API

Easiest use case:

User must implement interfaces

```
rave::Factory factory ( MyMagneticField(), MyPropagator() );

std::vector < rave::Vertex > vertices = factory.create( tracks,
"method" );
```

Factory creates vertices from tracks, using method "method"

... robust algorithms ...

"Adaptive" methods introduce the notion of track-to-vertex assignment probabilities in the Kalman filter formalism.
The AdaptiveVertexFitter (avf) and the MultiVertexFitter are two such adaptive methods which are able to deal with contamination without any prior information of the type of contamination.

algorithms

The AdaptiveVertexFitter is an iterative, weighted Kalman filter. An annealing schedule is introduced to avoid falling into local minima (see next slide).

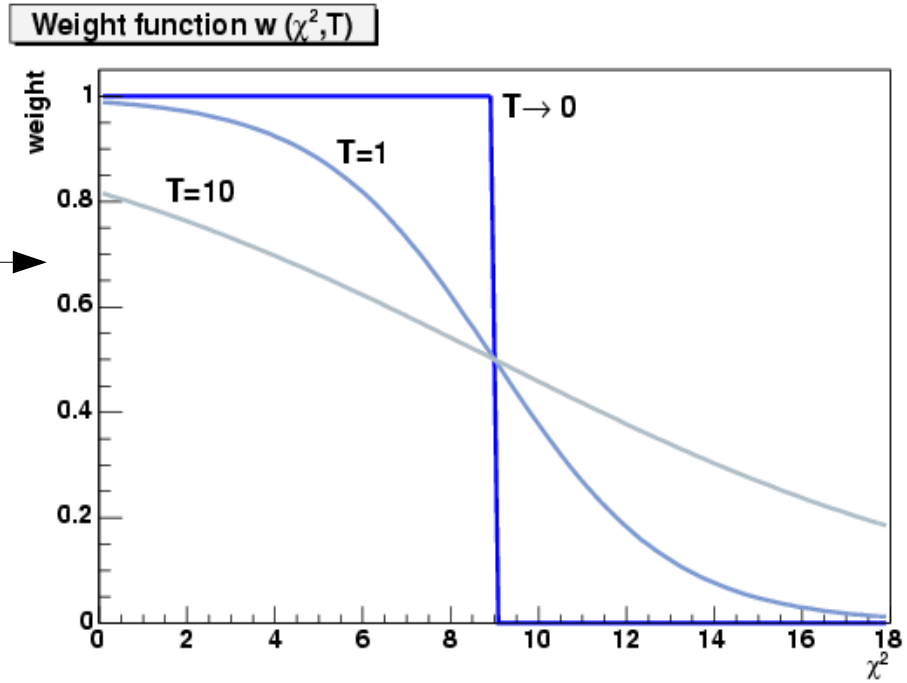
The MultiVertexFitter fits several vertices at once, introducing competition: the vertices have to "compete" for the tracks. It solves both the statistical problem (vertex position estimation) as well as the pattern recognition problem (vertex finding) at once.

... robust algorithms ...

Objective function of the adaptive method:

$$\hat{\beta}_{Adaptive} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \left(w_i \cdot r_i^2(\beta) \right)$$

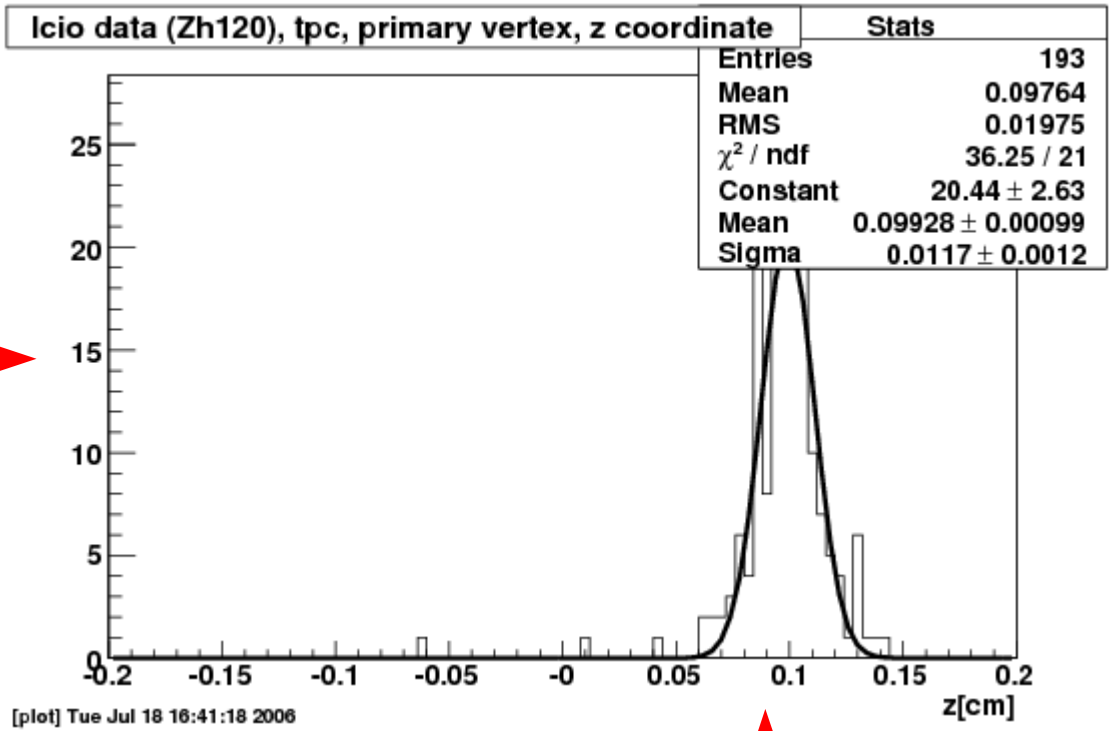
assignment probability as a function of a x^2 and the "annealing temperature" T, which is lowered geometrically in the iterative fitting process.



... easily embedabble ...

Rave as an ILC "MarlinProcessor", fitting primary vertices from TPC Tracks.

z coordinate of primary vertices



1 mm shift in z – a bug in the ILC track reconstruction found by rave!

... and a sandbox to play with ...

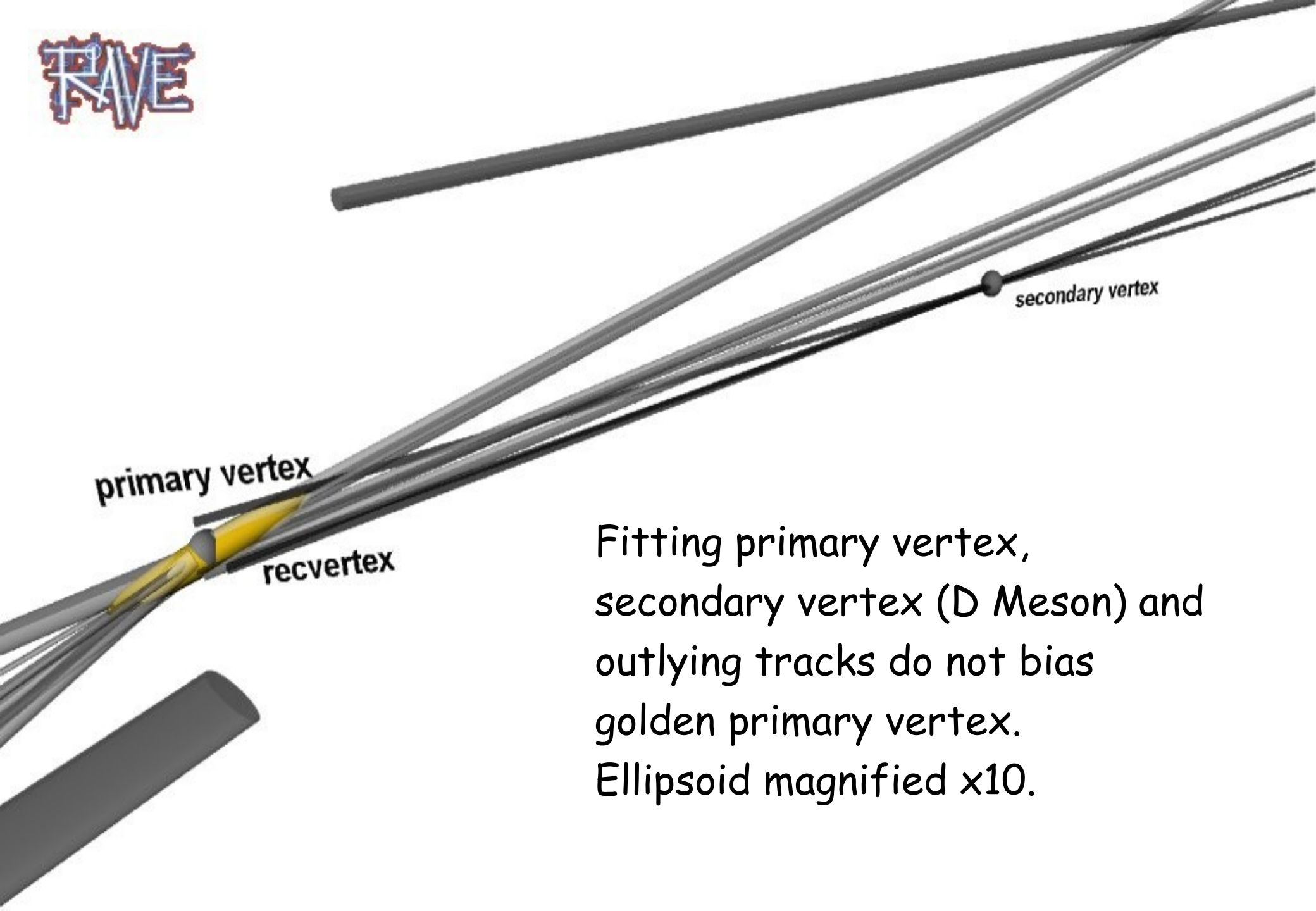
"Vertigo" implements a simple standalone framework that can be used to test rave algorithms.

It can simulate different experiment setups for vertexing purposes and read from various data sources.

Vertigo serves as a very fast development tool for vertexing algorithms!

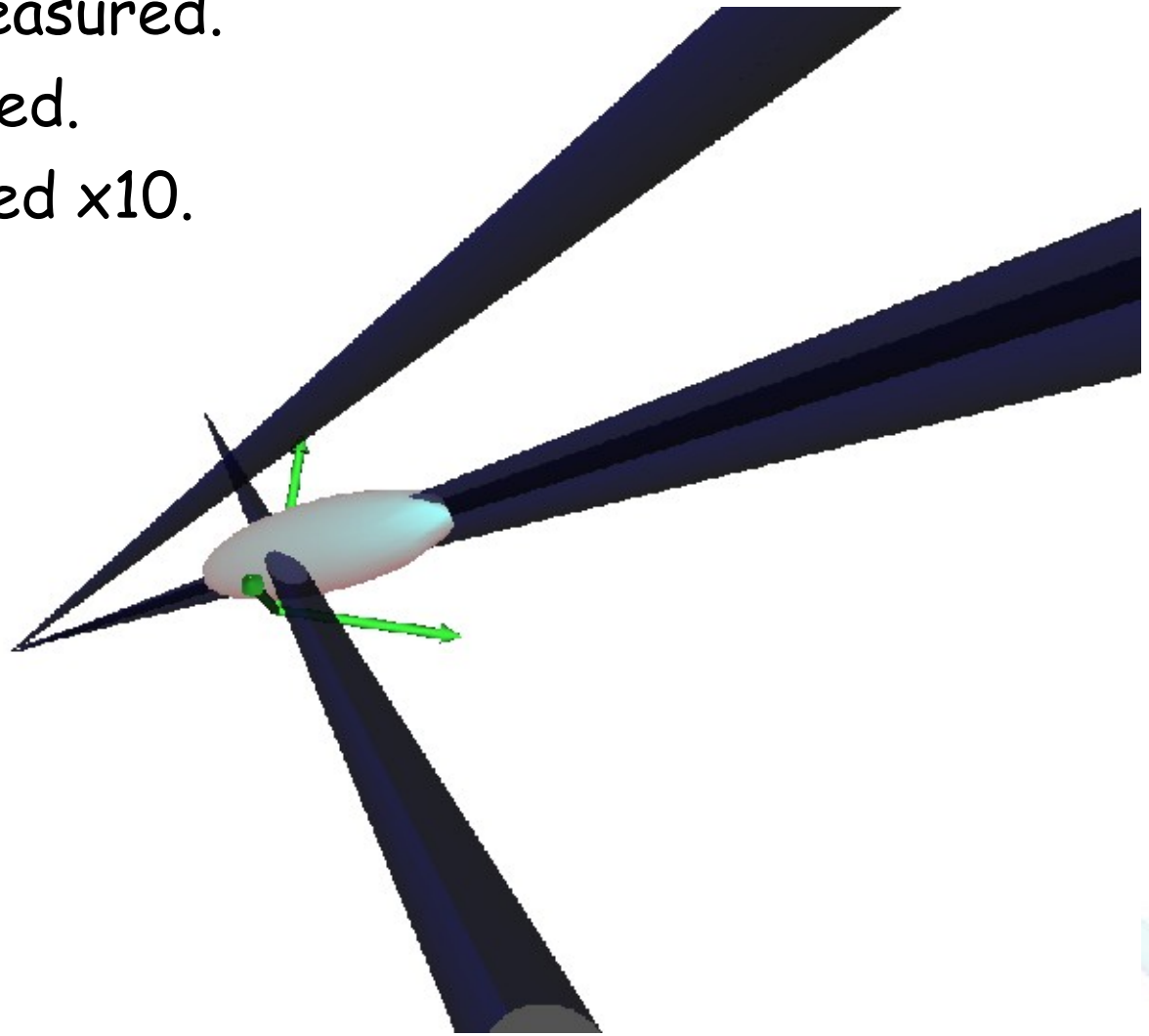
Fitting CMS tracks without the need for CMS software.

```
centurion ~/svn/vertigo> vertigo -v0 -n1 -sgun:simple -oEventPrinter -mavf

            Rave Event  1 (dvg)  [13 ms]
          ?-------------------------------------?

SimVertex #11 at (1.27551e-05, 1.06171e-05, 0.525283)  (rectible)
    `-- SimTrack #0: (1.27551e-05, 1.06171e-05, 0.525283  , 1.00652, -0.174495, -0.073179)
          Track #12: (4.25446, -0.625808, 0.221045  , 1.01796, -0.12402, -0.0737569) q=-1
    `-- SimTrack #1: (1.27551e-05, 1.06171e-05, 0.525283  , 3.54454, -0.795207, 1.7637)
          Track #13: (4.18822, -0.972657, 2.61364  , 3.51428, -0.839079, 1.7585) q=1
    `-- SimTrack #2: (1.27551e-05, 1.06171e-05, 0.525283  , 2.14129, 0.19788, 0.993521)
          Track #14: (4.27526, 0.447036, 2.51509  , 2.12266, 0.248297, 0.989563) q=-1
    `-- SimTrack #4: (1.27551e-05, 1.06171e-05, 0.525283  , 0.981928, -0.0909635, 0.0495309)
          Track #15: (4.26952, -0.511339, 0.73918  , 0.974378, -0.142527, 0.049657) q=1
    `-- SimTrack #5: (1.27551e-05, 1.06171e-05, 0.525283  , 2.54354, -0.571272, -0.661853)
          Track #16: (4.20282, -0.90309, -0.571476  , 2.55027, -0.518887, -0.664557) q=-1
    `-- SimTrack #6: (1.27551e-05, 1.06171e-05, 0.525283  , 1.66227, 0.0899377, -0.432759)
          Track #17: (4.29594, 0.165186, -0.595952  , 1.66162, 0.0386202, -0.430936) q=1
    `-- SimTrack #7: (1.27551e-05, 1.06171e-05, 0.525283  , 0.897776, -0.131544, -0.285651)
          Track #18: (4.23735, -0.742095, -0.822961  , 0.885463, -0.180531, -0.284264) q=1
    `-- SimTrack #8: (1.27551e-05, 1.06171e-05, 0.525283  , 3.92705, 0.0446647, -1.57284)
          Track #19: (4.29593, 0.023694, -1.20689  , 3.97945, -0.00293343, -1.59191) q=1
    `-- SimTrack #9: (1.27551e-05, 1.06171e-05, 0.525283  , 1.20667, -0.113276, 0.00159821)
          Track #20: (4.2884, -0.31761, 0.53437  , 1.21349, -0.0619361, 0.0017907) q=-1
    `-- SimTrack #10: (1.27551e-05, 1.06171e-05, 0.525283  , 0.860757, -0.110031, 0.258953)
          Track #21: (4.28074, -0.420268, 1.80443  , 0.865886, -0.0590022, 0.259206) q=-1

Unassociated RecTracks:

Reconstructed vertices:
-----------------------

Vertex #22 at (-0.00109195, -0.0022394, 0.523577) [30.31760m]
    `-- associated with SimVertex #11 [30.31760m,1.9154s off]
    `-- Track #12 w=0.953831
    `-- Track #13 w=0.983217
    `-- Track #14 w=0.985284
    `-- Track #15 w=0.984351
    `-- Track #16 w=0.982273
    `-- Track #17 w=0.979565
    `-- Track #18 w=0.96781
    `-- Track #19 w=0.907144
    `-- Track #20 w=0.965554
    `-- Track #21 w=0.979714
     -- misassociated tracks 0.311257 (hard: 0)


[SimpleScore] eff=1 +/- 0, fake=0 +/- 0
centurion ~/svn/vertigo>
1$* 0:  2-$ 0:  4$ 0:
```



Fitting primary vertex, secondary vertex (D Meson) and outlying tracks do not bias golden primary vertex. Ellipsoid magnified x10.

- Fitting J/ψφ→ KKμμ (CMS ORCA event),
- one track mis-measured.
- Vertex unaffected.
- Ellipsoid magnified x10.



It all comes with Python interfaces


```
import vertigo

eventfactory=vertigo.EventFactory("lcio:tracks.slcio" )
ravefactory=vertigo.RaveFactory()
visualiser=vertigo.Visualiser()

for event in eventfactory:
  vertices=ravefactory.create ( event.tracks(), "avf" )
  event.add ( vertices )
  visualiser ( event )
```

instantiate, data source is an lcio file

fit tracks with adaptive "avf" method, add vertices to event, and visualise.

Downloads

http://stop.itp.tuwien.ac.at/publish/

contains:

- RAVE
- VERTIGO
- DataHarvesting plugin for Vertigo (optional) (can write/read Root/Hdf/Xml/Text files)
- Visualization plugin for Vertigo (optional) (simple visualisation – see plots in this talk)

http://stop.itp.tuwien.ac.at/websvn/ Web Subversion Server

Documentation:

http://stop.itp.tuwien.ac.at/docs/vertigo/

http://stop.itp.tuwien.ac.at/docs/rave/

RAVE User Guide

Wolfgang Waltenberger

<walten@hephy.oeaw.ac.at>

Fabian Moser

This document is the official User Guide for RAVE (Reconstruction in an Abstract Vertices Environment). This is version $Rev: 447 $.

Table of Contents

List of Figures

Next
Foreword

Vertigo

Name

vertigo -- a command line tool to test rave algorithms in various environments

Synopsis

vertigo [-h | -l | -v*VERBOSITY* | -s*SOURCE* | -m*METHOD* | -S*SKIN* | -n*NUM* | -o*OBSERVER* | -c*CONFIGURABLE* | -C]

DESCRIPTION

vertigo is conceived as a command line tool to test rave algorithms with both artifical ("vertex gun") and realistic data. It can currently read in data generated by a data harvester, as well as LCIO data. On the analysis side, **vertigo** has a few Observer classes that make it easy to analyse the performance of the vertex reconstruction algorithms. This man page is for $Rev: 191 $.

OPTIONS

vertigo accepts the following options:

-h,--help

Shows the help page

-l,--list

List all methods and observers (then quit)

-v,--verbosity

Outlook

Future developments for Rave:

- A special-purpose vertex finder for "B-jettish" event topologies.
- A vertex fitter that deals with non-Gaussian errors (already implemented in CMS)
- Make (better) use of beamspot constraints for fitting primary vertices.
- Write org.lcsim "Driver" in Java

References

- "Adaptive Vertex Fitting" W. Waltenberger, R. Frühwirth and P. Vanlaer. CMS AN-2006/104.

- "Adaptive Multi-Vertex Fitting", W. Waltenberger, R. Frühwirth, CMS CR-2004/062, CHEP proceedings Interlaken, Switzerland, http://cmsdoc.cern.ch/documents/04/cr04_062.pdf.

- "Vertex Fitting in the CMS Tracker" T. Speer, K. Prokofiev, R. Frühwirth, W. Waltenberger and P. Vanlaer. CMS-NOTE-2006-032.

- "A Vertex Reconstruction Toolkit and Interface to Generic Objects (VERTIGO)", W. Mitaroff and W. Waltenberger, Proc. 7th Int. Conf. on Linear Colliders (LCWS 04), Paris, 2004, LCnote LC-TOOL-2004-017.

- "LiC Detector Toy" (MatLab based mini simulation and track fitting tool for fast and flexible detector optimization studies) Ref. http://forum.linearcollider.org/ => Fast Simulations => LiC Toy.