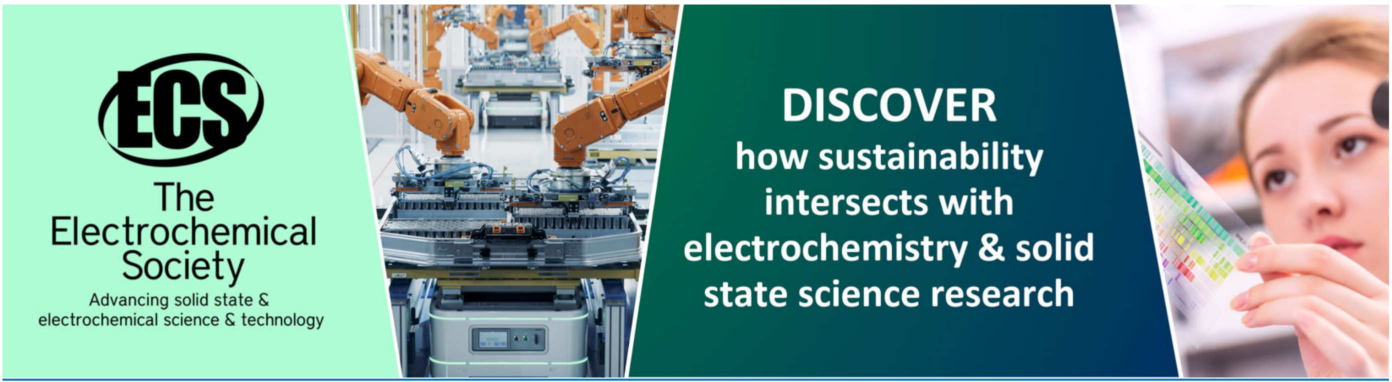
Journal of Physics: Conference Series

The RAVE/VERTIGO vertex reconstruction toolkit and framework

To cite this article: W Waltenberger *et al* 2008 *J. Phys.: Conf. Ser.* **119** 032037



International Conference on Computing in High Energy and Nuclear Physics (CHEP'07) IOP Publishing
Journal of Physics: Conference Series **119** (2008) 032037 doi:10.1088/1742-6596/119/3/032037

The RAVE/VERTIGO vertex reconstruction toolkit and framework

W. Waltenberger, W. Mitaroff, F. Moser, B. Pflugfelder and H. V. Riedel

Austrian Academy of Sciences, Institute of High Energy Physics, A-1050 Vienna, Austria, EU

E-mail: `walten@hephy.oeaw.ac.at`

Abstract. A detector-independent toolkit for vertex reconstruction (RAVE[1]) is being developed, along with a standalone framework (VERTIGO[2]) for testing, analyzing and debugging. The core algorithms represent state-of-the-art for geometric vertex finding and fitting by both linear (Kalman filter) and robust estimation methods. Main design goals are ease of use, flexibility for embedding into existing software frameworks, extensibility, and openness. The implementation is based on modern object-oriented techniques, is coded in C++ with interfaces for Java and Python, and follows an open-source approach. A beta release is available.

1. Motivation and goals

Experiments at high-energy particle colliders rely on precise track and vertex reconstruction, which must not compromise the high spatial resolution achieved by modern tracking detectors. Data analysis therefore requires new, sophisticated methods beyond the traditional least squares (Kalman filter) estimators, using robust, non-linear, mostly adaptive algorithms.

In the data reduction chain, the early stages of local pattern recognition, track search and track fitting are highly detector-dependent, whereas the next stage – geometric vertex reconstruction (finding and fitting) – is almost fully detector-independent[3]. Thus, it should neither be necessary nor desirable for each new detector collaboration to re-code vertex reconstruction from scratch, provided an adequate, reliable and easy-to-use toolkit would be available. Note, however, that vertex fitting with kinematic constraints may be subject to the requirements of a subsequent physics analysis.

2. Design and functionality

The idea of a detector-independent library for "vertex evaluation by robust algorithms" dates back to 1995 [1], but its realization as a toolkit with high level of abstraction became feasible only after the advent of object-oriented software frameworks. This project was initiated in 2003, and first parts of VERTIGO (the data harvester/seeder and visualiser tools) were developed and presented in 2004 [2, 3].

---

[1] RAVE = "reconstruction (of vertices) in abstract versatile environments",

[2] VERTIGO = "vertex reconstruction toolkit and interface to generic objects".

[3] at LHC, the overlaid events will have an impact on the frequency of outliers.

2.1. The RAVE toolkit

The RAVE core is intended to collect the world's best algorithms available for all aspects of vertex reconstruction – finding, fitting and kinematics. At present, it includes both vertex finding (a pattern recognition task a.k.a. track bundling), and vertex fitting (i.e. parameter estimation of the vertex position and associated "smoothed" track momenta, together with their covariance matrices). The algorithms had originally been developed for the CMS reconstruction software ORCA [4], and have recently been re-factored for being ported to a new framework (CMSSW). Source code compatibility between RAVE and CMSSW is desirable and will continue.

Principal assets of the RAVE toolkit are robust reconstruction algorithms with estimators based on adaptive filters [5, 6], thus down-weighting the influence of outlier tracks (large residuals w.r.t. the errors, or not belonging to the vertex being fitted); it also provides an adaptive multi-vertex fitter (MVF) [7], the performance of which is demonstrated below. A discussion of the algorithms supported by RAVE is given in [8].

Thanks to its simple API, the RAVE toolkit may be embedded into the software environment of any non-CMS collider experiment with minimal effort: all that is needed is some glue code (adaptor classes) for interfacing to the framework's data objects, e.g. its class representing a reconstructed track. So far, RAVE has been successfully tested with the ILC software frameworks "MarlinReco" [9] (C++ based) and "org.lcsim" [9] (Java based, using a SWIG wrapper). More information about the API can be found also in [8].

2.2. Performance of RAVE

The "AdaptiveVertexFitter" (AVF) is a robust generalization of the Kalman filter, iteratively downweighting the contribution of outlier tracks to the objective function ("soft assignment"). The extra weights w_i on the reduced residuals r_i are calculated by a Fermi function with cutoff parameter r_{cut} (Fig. 1). In addition, a deterministic annealing schedule with decreasing "temperature" T is introduced in order to avoid falling into local minima:

$$w_i(r_i, T) = \frac{e^{-r_i^2/2T}}{e^{-r_i^2/2T} + e^{-r_{cut}^2/2T}} = \frac{1}{1 + e^{(r_i^2 - r_{cut}^2)/2T}}$$

Iterations (index k, omitted above) should start with an a-priori guess of $w_{i,1}$. For $k > 1$, the $w_{i,k} = w_i(r_{i,k-1}, T_k)$, with $T_k \leq T_{k-1}$ defined by the annealing schedule. For $T \rightarrow 0$, the Fermi function approximates the Heaviside function, and the assignment turns into a "hard" one ($w_i = 1$ or 0).

The "MultiVertexFitter" (MVF) is a generalized AVF, simultaneously fitting n vertices by "soft assignment" of each track to more than one vertex. The extra weights w_{ij} on the reduced residuals r_{ij} w.r.t. vertex j are

$$w_{ij}(r_{ij}, T) = \frac{e^{-r_{ij}^2/2T}}{\sum_{\ell=1}^{n} e^{-r_{i\ell}^2/2T} + e^{-r_{cut}^2/2T}}$$

As an example, events were generated with a primary vertex (PV) at [0,0,0] cm and one secondary vertex (SV) at [0,0,0.2] cm; with the PV (SV) emitting a jet of 5 (3) tracks of total momentum [0,25,25] ([-15,0,20]) GeV and openening angle 0.5 (0.5) rad, respectively. Gaussian track errors were simulated by the tracks' covariance matrices. Thereafter, one PV track and one SV track were chosen to be swapped, i.e. assigned to the wrong vertex (Fig. 2 top). The two track bundles, each containing one wrongly assigned track, were submitted to a Kalman filter (KF), the AVF and the MVF.

Resolutions of the reconstructed decay lengths are shown in Fig. 2 bottom, demonstrating the superior performance of the AVF w.r.t. the KF, and also improved performance of the MVF over the AVF.

2.3. The VERTIGO framework

The vertex reconstruction toolkit is complemented by a simple standalone framework, VERTIGO, for testing, analyzing and debugging core algorithms. Thanks to powerful persistency solutions, VERTIGO may also serve as an alternative for running RAVE without embedding it into another framework; for this purpose, emulation of a detector setup is supported by the "skin" concept, introducing an intermediate layer between RAVE and VERTIGO proper.

The VERTIGO framework provides interfaces and tools, either natively or as plugins, for input ("event generators") and output ("observers"), see Fig 3. The most prominent ones are:

- Vertex Gun – an event generator that creates "artificial" events, including *b*-jet like secondary vertices;
- LCIO Event Generator – interface for the ILC common data format LCIO [9];
- Data Harvester – an abstraction layer for persistency solutions , based on STL mappings of heterogeneous objects, and supporting a wide range of data formats including simple CSV text, XML, HDF5, and ROOT.
- Vertex Histogrammer – an observer based on the Data Harvester for analyzing and quality checking results of the RAVE fitters;
- Visualiser – an observer that interfaces to the "RaveVis" event display tool, deliberatly kept simple for the sake of detector independence. It is based on Coin3D [10]; a snapshot is shown in Fig. 4.

More information about VERTIGO's features can be found in [8].

VERTIGO/RAVE has so far been tested with input data from CMS (CMS skin), ILC detectors (LCIO, LDC and SiD skins), and a simple detector simulation by the LiC Tool [11] ("dummy skin"). Fig. 5 shows the resolutions of the z coordinates of reconstructed ILC collision vertices.

3. Conclusions and outlook

The RAVE vertex reconstruction toolkit and its companion standalone framework VERTIGO have successfully been tested with input data from the CMS and ILC detectors, and on various platforms (Linux, Mac OS X, Windows via MinGW). It has been used on both intel and ppc CPU architectures. The project is now hosted at HepForge [13]. Beta releases as well as finalised packages for Debian and SLC4 can be downloaded from there.

The RAVE toolkit has recently been augmented by the CMSSW "combined *b*-tagger", based on a likelihood-ratio method. This may eventually evolve into a general-purpose "flavour tagger".

Work is under way for a link between RAVE and the topological vertex search algorithm ZvTop [14]. A completely re-coded C++ version has recently been released by the Oxford LCFI group, and a Java version exists at SLAC. Combining ZvTop with RAVE (e.g. as a seeder for the MVF) could improve its performance for complex topologies.

A project to be started in October 2007 aims at augmenting the RAVE toolkit with a "kinematic fitter", using existing code which had originally been developed in 2005 for ORCA. There is no intention to push RAVE beyond vertex finding and fitting, flavour tagging, and kinematics.

For VERTIGO, we plan the addition of an interface (similar to LCIO EvtGen) for the BELLE data format "Panther" [15].

Acknowledgements

Thanks are due to the "CMS vertexing circle" for code shared with RAVE. Special thanks go to Meinhard Regler and to Rudi Frühwirth for formulating a great part of the algorithms used.

[1] R. Frühwirth et al.: Comp.Phys.Comm. 96 (1996) 189-208.

[2] W. Mitaroff, W. Waltenberger: Proc. Int. Conf. on Linear Colliders (LCWS 04), Paris, 19 - 23 April 2004, Ed. de l'École Polytechnique, Palaiseau 2005 – vol. 1, p. 413, ISBN 2-7302-1281-7.
[3] W. Mitaroff et al.: Proc. Int. Conf. for Computing in High Energy and Nuclear Physics (CHEP 04), Interlaken (CH), 27 Sep. - 1 Oct. 2004, CERN, Geneva 2005 – vol. 1, p. 272, CERN-2005-002-V1.
[4] T. Boccali et al., Proc. Int. Conf. for Computing in High Energy and Nuclear Physics (CHEP 03), La Jolla (USA), 24 - 28 March 2003, SLAC, Menlo Park 2003 – ref. TULT012, SLAC-R-636 / eConf C0303241.
[5] R. Frühwirth et al., Proc. Int. Conf. for Computing in High Energy and Nuclear Physics (CHEP 03), La Jolla (USA), 24 - 28 March 2003, SLAC, Menlo Park 2003 – ref. TULT013, SLAC-R-636 / eConf C0303241.
[6] R. Frühwirth et al.: *Adaptive vertex fitting*, Report CMS-NOTE-2007-008, CERN, Geneva 2007 (submitted to J.Phys.G).
[7] R. Frühwirth, W. Waltenberger: Proc. Int. Conf. for Computing in High Energy and Nuclear Physics (CHEP 04), Interlaken (CH), 27 Sep. - 1 Oct. 2004, CERN, Geneva 2005 – vol. 1, p. 280, CERN-2005-002-V1.
[8] W. Waltenberger, F. Moser: Proc. IEEE Nuclear Science Symposium (NSS 06), San Diego (USA), 29 Oct. - 4 Nov. 2006, Report IEEE-NSS-Note-06-104 (also available on CD-ROM), IEEE 2006.
[9] `http://www-flc.desy.de/ilcsoft/`
`http://lcsim.org/software/`
`http://lcio.desy.de/`
[10] `http://www.coin3d.org/`
[11] M. Regler et al.: *The LiC Detector Toy Program*, Proc. 11^{th} Vienna Conf. on Instrumentation (VCI 07), 19 - 24 Feb. 2007 – Nucl.Instr.Meth. A (in print).
[12] `http://stop.itp.tuwin.ac.at/websvn/`
[13] `http://www.hepforge.org/`
[14] D.J. Jackson: Nucl.Instr.Meth. A 388 (1997) 247-253.
`http://www.physics.ox.ac.uk/LCFI/Physics.html`.
[15] `http:/belle.kek.jp/group/software/panther/`

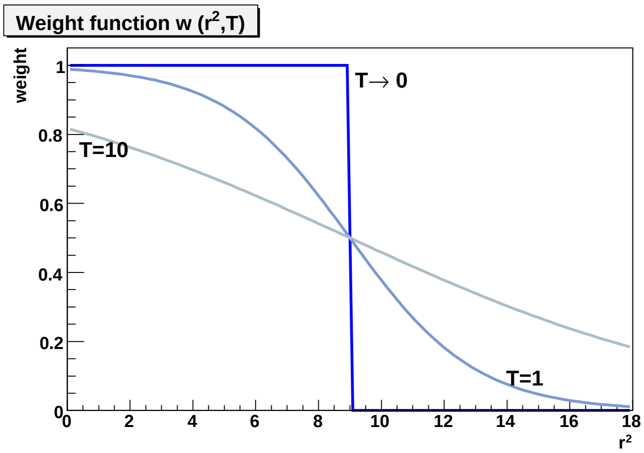


Figure 1. Example of a Fermi function $w_i(r_i^2, T)$ with $r_{cut} = 3$, and for $T = \{10, 1, 0\}$, as used by the AVF. The objective function of the parameters $\vec{\beta}$ to be fitted is $\mathcal{M}(\vec{\beta}) = \sum_{i=1}^{n} w_i \cdot r_i^2(\vec{\beta}) \rightarrow \min(\vec{\beta})$.

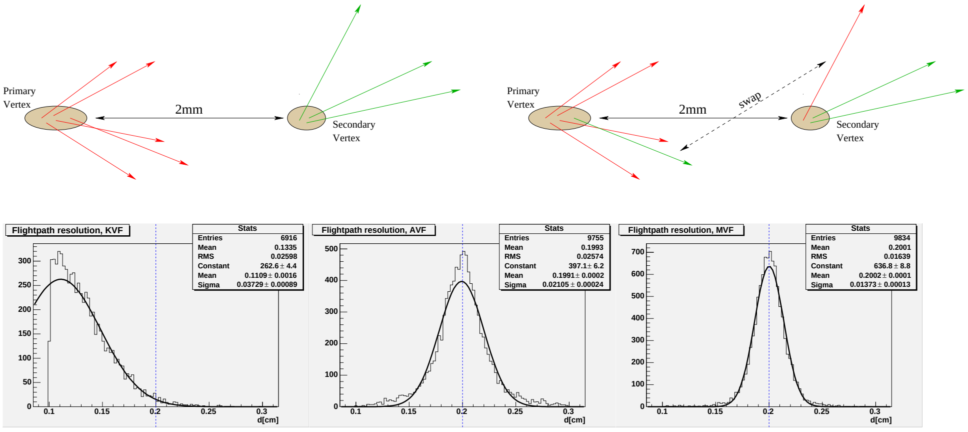


Figure 2. Performance of the multi-vertex fitter (MVF).

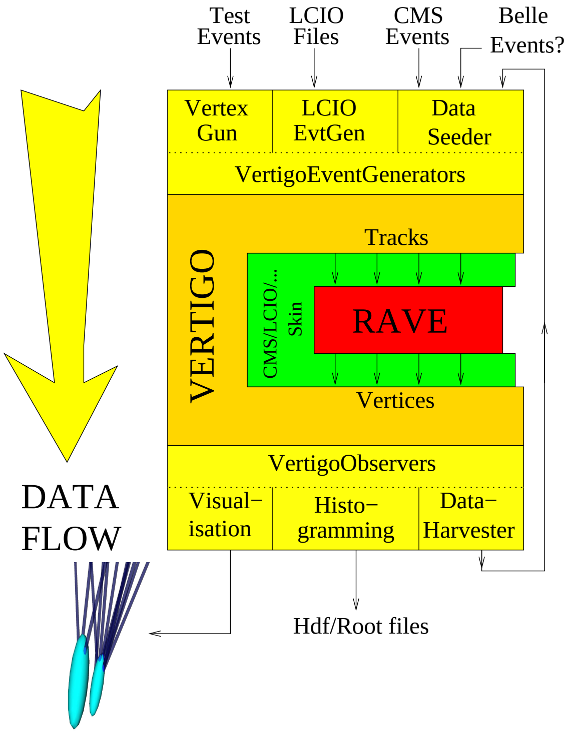
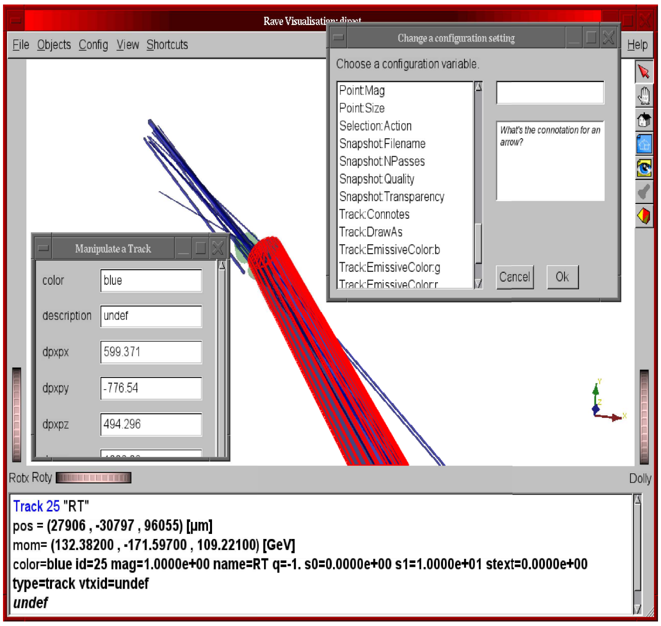


Figure 3. Functionality of the standalone framework VERTIGO.



color: blue
description: undef
dpxpx: 599.371
dpxpy: -776.54
dpxpz: 494.296

```
Track 25 "RT"
pos = (27906 , -30797 , 96055) [µm]
mom= (132.38200 , -171.59700 , 109.22100) [GeV]
color=blue id=25 mag=1.0000e+00 name=RT q=-1. s0=0.0000e+00 s1=1.0000e+01 stext=0.0000e+00
type=track vtxid=undef
undef
```

Figure 4. The VERTIGO visualisation plug-in in action.

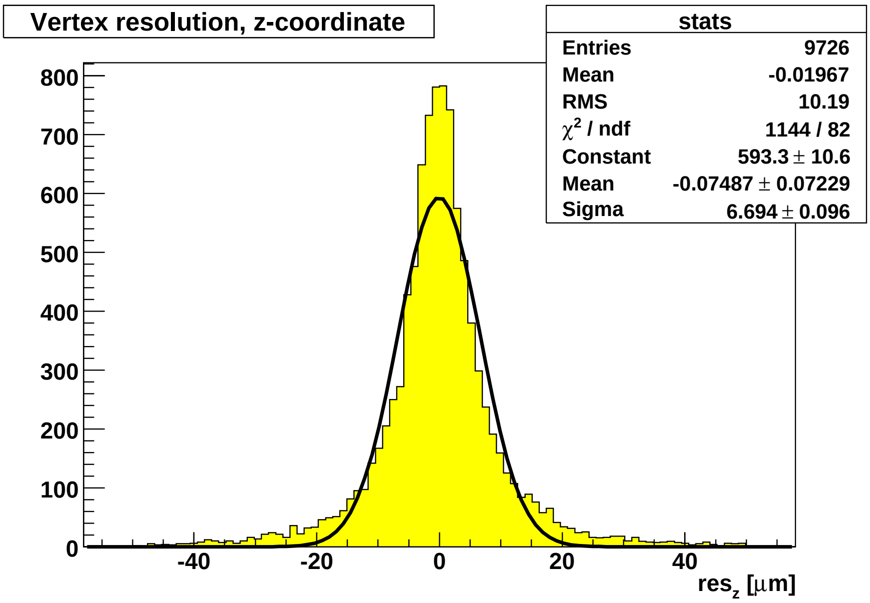


Figure 5. z coordinates of reconstructed (collision) vertices, ILC data. No prior information is used in this context.