SLAC-PUB-11718

Java Vertexing Tools

Jan Strube
University of Oregon, Eugene, OR 97403, USA
Norman Graf
Stanford Linear Accelerator Center, Menlo Park, CA 94025, USA

This document describes the implementation of the topological vertex finding algorithm ZVTOP within the org.lcsim reconstruction and analysis framework.

1. INTRODUCTION

Information about vertices in an essential component of any physics analysis at a linear collider. The SLD experiment spurred the development of an algorithm for topological vertex finding, named "ZVTOP". This algorithm was successfully used in many analyses. Because of its success, it is desirable to have an implementation of this algorithm in toolkits designed for studies at the ILC. However, the original algorithm was coded in a Fortran dialect, which is not suited for modern toolkits. The North American Study Group developed a toolkit in the Java programming language, which enables a modern and collaborative design and allows for easier deployment across platforms.

2. AN IMPLEMENTATION IN JAVA

The first implementation of the ZVTOP algorithm in the Java programming language was done in the hep.lcd reconstruction framework. The implementation was based on the original source code that was used in the SLD experiment.

In the spirit of the design of the new framework it was decided to start the implementation afresh, based on a publication[1] rather than source code. The fact that the author of the first implementation[2], was no longer available for the task gave further justification for this decision.

2.1. DESIGN CONSIDERATIONS

One of the main design goals for the new implementation was close integration into the org.lcsim framework, taking advantage of existing classes and interfaces for tracks, vectors, fitters and swimmers rather than providing implementations of its own. This significantly reduces the maintenance cost of the package and allows focussing on the algorithm.

Another important aspect of the design is its abstraction, meaning that the core of the package should depend on interfaces rather than implementations. This allows for an easy change of, for example, the vertex fitter to be used by the package. One of the main reasons why a port of the old Java code to the new framework was deemed unfeasible is a deep-rooted dependency on concrete implementations like the class of particles.

We are confident that this is the right approach to ensure longevity of the package. Minimizing the number of external as well as internal dependencies and taking advantage of more recent language features reduces the learning curve which encourages more people to help maintain the package, thereby providing higher stability and reliability of the code.

2.2. INTEGRATION INTO ORG.LCSIM

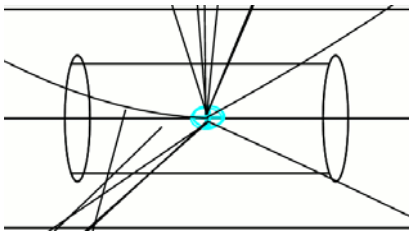

Figure 1: Demonstration of the integration into WIRED 4. The ellipsoids in the center (aqua) denote the location of vertices found by ZVTOP. The cylinder in the center is the inner layer of the vertex detector. The event is $e^+e^- \to \tilde{t}\tilde{t}$

In the process of integrating the package into the org.lcsim framework, we were able to take advantage of the excellent application programming interface (API) of the WIRED4[3] event display. Thanks to the well thought-out design, it was very easy to get ZvTop to show its vertices in the event display. As well as being a great tool for presentation purposes, this feature will aid in debugging, tuning and improving the code, allowing for a quick visual check of the performance. Furthermore it will give the user quick feedback on the performance of the algorithm.

3. CURRENT STATUS

At the time of this writing, the ZVTOP implementation in org.lcsim is not yet ready for use in production. It was decided that a more advanced vertex fitter be used than was provided by the original source code. This has not yet been implemented. Furthermore parts of the algorithm that were supplied in the original implementation but are not included in the write-up, such as the "Ghost Track" algorithm, are missing.

After achieving feature-completeness, validating the package is essential to (im)prove its reliability and give users confidence in its performance. It is planned to directly compare the Java implementation with the one used at the SLD experiment by feeding the same events to both packages.

4. SUMMARY

At the present date, Java vertexing tools allow users to perform topological vertexing on tracks that have been obtained from a Fast MC simulation. An implementation that will be able to handle fully reconstructed events is being designed from the ground up for longevity and maintainability.

Acknowledgments

The authors wish to acknowledge the hard work by the members of the org.lcsim development team. Furthermore we wish to thank David Jackson for taking the time to explain the ZVTOP algorithm.

References

[1] Jackson, David J. *Nucl. Instrum. Meth.* **A388** (247-253)
[2] http://www-lc.fnal.gov/proceedings/p4_eweak/walkowiak_ewsb.ps
[3] http://wired.freehep.org/
[4] http://lcsim.org