Vertexing at BaBar

J.M. LoSecco
for the BaBar Computing Group

Physics Department, University of Notre Dame, Notre Dame, IN 46556 USA
John.M.LoSecco.1@nd.edu

An system for vertexing and kinematic fitting developed for use with the BaBar detector, is described.

Key words: Object-oriented; vertexing; kinematic fitting; BaBar.

1 Vertexing

One significant consumer of tracking information is vertexing. Vertexing attempts to find points common to subsets of the tracks in an event. In BaBar vertexing plays a vital role since a signature of CP violation is a difference in the decay position distribution of a particle and its antiparticle.

A number of algorithms have been designed to solve the vertexing problem. It has been traditional to vertex by looking at the intersection of some subset of tracks and varying the tracks included in the subset until a suitable fit is obtained. The concept of an incremental fit based on the ideas of Kalman filtering has some features that make it preferable. In particular, tracks are added to a vertex one at a time. This leads to simpler mathematics, involving only five-dimensional matrices at each step rather than larger and larger matrices, with associated mathematical instabilities and long compute times, as the number of tracks increases. In addition, an unsuitable track can be removed from the vertex if it is later found that it does not belong there.

The objects used in this approach to vertexing are tracks and vertices. Vertices can be formed from a vertex and a track. In reality a vertex can only be formed by two or more tracks but the incremental nature of the Kalman fit permits one to define a vertex object with zero, one or more tracks. Only those with more than one track are meaningful. For vertexing the "+" operator has been overloaded, to permit adding tracks to a vertex. The "+" operator refines the position of the vertex based on a weighted average of the existing vertex position and the additional information introduced by the track. The covariance matrix associated with the vertex is updated to be consistent with having the additional track added. So if a well determined track is added to a well determined vertex where it doesn't belong, a very poor fit will result. In such cases the track can be removed.

Most vertices of interest to the experiment come from the decay of short lived particles, which may themselves come from other decays. Tracks that come from a single vertex can be combined to produce "pseudotracks" that can themselves be vertexed. The name "pseudotracks" implies that these do not have a corresponding track object. There can be no vertexing of primary neutrals since they lack position information. Vertexing is possible with both charged and neutral pseudotracks since the spatial information about the pseudotracks is available in the form of the vertex where the tracks meet. The model for vertexing pseudotracks is the same as for tracks, and pseudotracks themselves can be constructed from tracks and pseudotracks.

Vertexing provides additional information about the tracks at that vertex. This information can be used to improve knowledge about the track or pseudotrack. Member function to perform this "smoothing" function exist and give a refined estimate of a track's momentum and position.

2 C++ Implementation of Kalman Filter Vertexing

While the Kalman Filter algorithm is a procedure that does not necessarily need to be implemented in an OOP environment some features of C++ make it very easy to generalize it. In particular, the vertexing algorithm can be coded in terms of matrix operations that do not have any explicit dependence on the track representation. All that is needed from the track class is that the tracks have a finite dimensional representation. The representation, its covariance and a linearized expansion of the representation about an assumed vertex and momenta are all that are needed for the algorithm. Such simplicity eases code management as new track classes are developed. Such new classes can be accomodated via a templated friend function of the vertexing class. As templates they are only instantiated when the need arises so code is only generated for those tracking classes actually used in vertexing.

3 Kinematic Fits

At modest to high multiplicity the spatial resolution of the detector is not capable of relating all tracks to the correct vertex on the basis of spatial information alone. For example, for a 12 track event about 50% of the time the lowest χ^2 does not correspond to the correct relation of tracks to vertices when only spatial coincidence is used. Since the tracks are known to come from the decays of unstable particles of definite mass. a kinematic fitter can be used to greatly reduce the ambiguity associated with the spatial information.

Kinematic fits take advantage of energy and momentum conservation to impose the requirement that the secondary tracks be consistent with the decay hypothesis of a known particle. This utilizes the momentum and particle identification information, or can be used to test whether a particular particle identification hypothesis is viable. Kinematic fitting introduces an additional complication since most particles of interest decay to a combination of charged and neutral tracks. A fit based on only the charged tracks is often unlikely to yield an interesting result. On the other hand the correct assignment of momentum to a neutral track requires, to some extent, a reasonable estimate of its origin, or at least two points on its trajectory. But in general kinematic fits give much better rejection of random combinations than spatial vertexing alone. The solution to this problem is to assume an origin for the neutral tracks, perform the kinematic fit, compute the vertex associated with the kinematic fit from the charged tracks in the fit and recompute the neutral from the reconstructed vertex. The errors on the neutrals are usually so large that this procedure does not particularly hinder the success of the method.

Output of the kinematic fit is a list of possible hypotheses that are consistent with the reconstructed tracks in the event. Vertices and tracks on vertices are provided along with the improved value of the track momenta used to satisfy the energy and momentum constraint. In the case of ambiguous particle identification the mass hypothesis used for the particular fit is assigned to the track. Since a fully reconstructed hypothesis may have multiple vertices the output is a tree in which internal branches are tracks of definite mass. Since multiple solutions can exist, this possibility must be accommodated in the output. An example with multiple hypotheses would be finding all of the neutral pion decays to two photons in an event. In general there will be more than one pion and sometimes a single photon may satisfy the pion hypothesis with more than one other photon.

Kinematic fitting, combined with vertexing, takes advantage of both the momentum resolution of the tracker and the spatial resolution of the vertex detector to provide optimal rejection of incorrect hypotheses and a refinement of the measurements and hence better resolution for the correct hypothesis.

References

[1] Joseph Boudreau, *The SLT Tracking Toolkit version 4* February, 1996.
Joseph Boudreau, *http://cactus.phyast.pitt.edu/~joe/slt.html.*

[2] R. Fruhwirth, N.I.M. **A262**, 444-450 (1987).

[3] S.M. Bozic, "Digital and Kalman Filtering", (Edward Arnold, London, 1979).

[4] R. Luchsinger and C. Grab, Comp. Phys. Comm. **76**, 263-280 (1993).

[5] Michael M. Smyth, "A Tracking Library for a Silicon Vertex Detector", Masters Thesis in Physics, Cornell University, August 1993.

[6] G.H. Golub and C.F. Van Loan, "Matrix Computations", (Johns Hopkins, Baltimore, 1983).

[7] O.I.Dahl, T.B.Day, F.T.Solmitz and N.L.Gould, "SQUAW Kinematic Fitting Program", Group A Programming Note P-126, University of California, Lawrence Radiation Laboratory, Berkeley, California (July, 1968).

[8] J.M.LoSecco, "Vertexing Tools for BaBar", BaBar note 308 (June, 1996).