2005 ALCPG & ILC Workshops - Snowmass, U.S.A.

Vertex Detector Based Tracking for SiD

N.B.Sinev
University of Oregon, Eugene , OR 97403-1274 USA

In the Silicon Detector (SiD) concept, main task of track finding is performed with the help of a vertex detector. Because main tracker consisting of 5 layers of microstrip detectors provides only one dimensional measurements, the pattern recognition in such a detector is difficult, especially in the case of large density of background hits. The performance of the tracking algorithm using Vertex Detector as pattern recognition device is studied in this report.

1. OVERVIEW OF EXISTING TRACKING PACKAGES FOR JAS

Most of the tracking algorithms used in Java Analysis Studio were developed in the hep.lcd package. Now migration of some of these programs into org.lcsim is in progress.

The algorithm studied in this work was created in year 2003 (see [1]) specifically for SiD. Before there was a tracking package called "combined tracker" (since 1999). However this code used Z coordinate measurements from central tracker while assigning hits to tracks. This could be adjusted to SiD conditions by setting very poor Z resolution. However, we decided to rewrite code and make it more SiD specific. This code can be called "tracking from inside to outside".

About the same time the code for tracking from "outside to inside" was developed in Kansas State University (Dmitry Onoprienko,Eckhard von Toerne). They used tracking capability of electromagnetic calorimeter to associate hits in central tracker with calorimeter "track stub". This method allows for reconstruction of tracks originated outside vertex detector.

Also, the code based on Kalman filter was adapted by Norman Graf for use in JAS. It is not completely optimized yet, as I could tell from testing it in hep.lcd implementation, but it is already migrated into org.lcsim.

For the case of very high density of background hits the Hugh Transformation based algorithm was developed a few years ago. It gives advantage in the reconstruction speed at very high ($> 10^4$ hits/layer) hit density. Such algorithm is suitable for specialized hardware based on FPGA Digital Signal Processing systems.

2. DESCRIPTION OF VERTEX DETECTOR BASED TRACKING ALGORITHM

The basic algorithm I am using here was developed a long time ago by Henry Videau for use in PEP TPC track reconstruction and was later adapted for BaBar by Henry Lynch and Orin Dahl. The pattern recognition starts from selecting 3 hits in 3 different layers of the tracking detector. If a helix can be traced through these hits, an attempt is made to find more hits close enough to this helix. If the number of such hits (+3 original) exceeds predefined threshold (we usually use 5 as such a threshold), we consider this helix as a track candidate.

Before accepting candidates more cuts are applied. First, the set of hits, assigned to a given track should be unique. Because we are trying triplets of hits from different layer combinations, we will see the same track in more than one try. To better eliminate such "duplicates" we have very strict cuts on hits sharing - no more than 1 hit can be common for 2 different tracks. In case two track candidates have more than 1 shared hit, one of such candidates is discarded. A candidate which has more hits associated or better chi squire is accepted. Another cut removes multiple turns of the same helix in the tracking detector. If we see new track candidate close in curvature and dip angle to one of those found before, we assume that this may be another turn of the same helix. Only the closest to IP turn is accepted as a track. Additional cut is designed to reduce number of "fake" tracks. In the case of dense backgrounds,

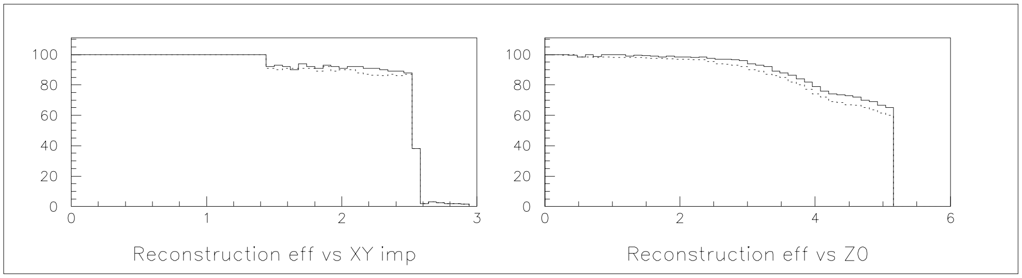

Figure 1: Reconstruction efficiency as a function of track impact parameters. Reconstruction cuts are set as 3.0 cm for XY impact and 5.0 cm for Z. Solid lines correspond to high Pt (> 1 GeV), dashed - low Pt (< 0.5 GeV).

track candidates, composed from only 3 hits in vertex detector, can easily find close hits in the central tracker, which do not really belong to it. This may create "fake" tracks. To reduce the probability of this we are requiring that not only threshold on the total number of assigned hits was satisfied, but also that at least four assigned hits were in Vertex Detector. Because of the high spatial resolution of the Vertex Detector the probability of assigning wrong hits to the track is much smaller here than in the outer tracker.

To have maximum reconstruction efficiency one would like to use all possible combinations of 3 vertex detector layers for the original track "seed". In SiD Vertex Detector geometry (version of February 2005 and later) with 5 barrel and 4 endcap layers the number of such combinations is 84. To speed up reconstruction we can use the fact that at least 4 layers of the Vertex Detector are required to have hits from the track. In that case the number of layer combinations used in track seed finding can be reduced to 13.

Because of large number of background hits in the Vertex Detector, the number of hit combinations (from 3 layers) can be huge if we combine any hit in one layer with any combination of hits in two other layers. To reduce processing time we can use the fact that we are interested only in tracks originated close to IP and that we need to limit curvature of the tracks to get rid of low momentum backgrounds. These two assumptions allow us, after selecting 1 hit in the outer most layer of a triplet, to set limits on regions in two other layers in the triplet where hits should be considered for construction of track seed. The size of such regions depends on how close to IP we want to limit track origin and on minimum track curvature.

3. TRACKING PERFORMANCE

3.1. Reconstruction efficiency

If we are not concerned about reconstruction speed, and want to see tracks from B decays (which may be centimeters away from IP) we can set algorithm parameters to allow reconstruction of tracks originated far from IP. Obviously, we can't reconstruct tracks originated outside the Vertex Detector. The Vertex Detector layout is determined by requirement that tracks from IP traverse at least 5 layers of the Vertex Detector. If tracks originate far from IP either in Z or in R, there will be gaps in the 5 layers traversing conditions for some values of the track dip angle. This leads to reduced average efficiency of track reconstruction which can be seen on figure 1. For this test I was not using any physics Monte Carlo events as they do not have large enough statistics of such tracks. Instead, single track hits were generated by my own program, and then the reconstruction algorithm was applied. Hit smearing and layer inefficiency was applied before reconstruction. However, there was no simulation of multiple scattering in this simple test. You can see from the plots that even tracks originated outside the first VXD layer are reconstructed with good efficiency though about 8% smaller than for tracks from IP.

Reconstruction efficiency as function of Pt and deep angle can be seen from figure 2. Now I was using real physics Monte Carlo TTbar events. I think some of the efficiency drop at high values of dip angle can be attributed to a bug in the geometry used (February 05), where there were small gaps in 5 layer VXD coverage in VXD endcaps.

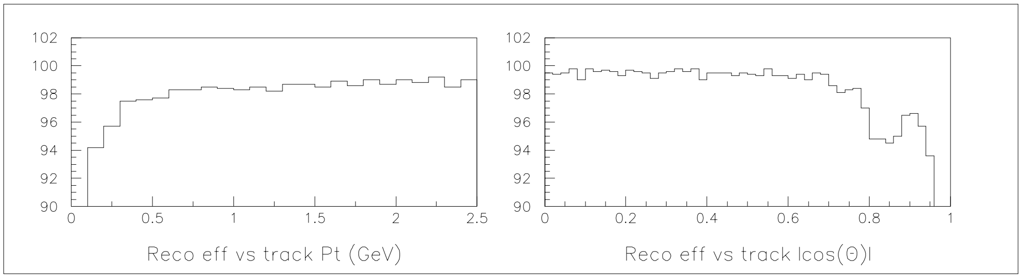

Figure 2: Reconstruction efficiency as a function of tracks Pt and dip angle.

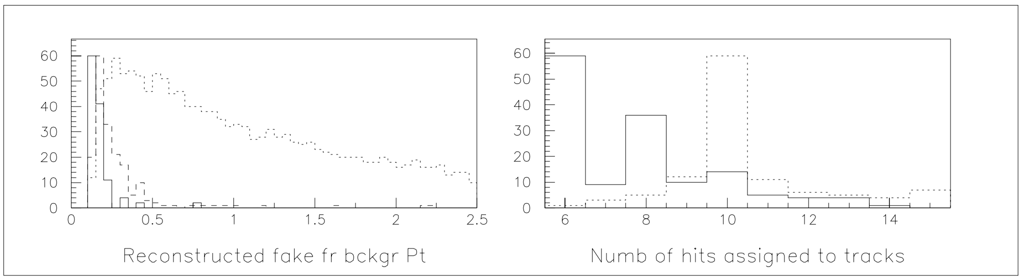

Figure 3: Fake tracks Pt distribution and distribution of the number of assigned to track hits. Solid curves are for fake tracks in the presence of background hits, long dashed - for fake tracks in the pure physics events (without background hits), and short dashed - for good physics tracks. Histograms are normalized at maximum

3.2. Fake tracks and effect of backgrounds

Sometimes we can see that more than one track has the same Monte Carlo particle designated as its parent. This may happened when two tracks are close to each other and hits assigned to such tracks may be confused. In that case track "purity" (defined as fraction of hits generated by the same Monte Carlo particle among all hits assigned to the track) is low. If some track "steals" the majority of assigned hits from another track, we call such track a "fake". In pure physics (TTbar) events the fraction of such fake tracks is about 0.8%. If we superimpose backgrounds, the number of fake tracks goes up. We assume a timing resolution of our tracking detectors such, that 10 beam crossings are piled up together. In that case hits from background particles can increase the number of fake tracks to 15% of all reconstructed. This is not acceptable. So we need either to use detectors with better timing resolution, or find cuts to suppress fake tracks reconstruction. Looking at parameters of fake tracks (figure 3), it is obvious that increasing minimum Pt cuts to 0.2 GeV and requiring not 6 but 7 hits for accepting a track will almost completely eliminate fakes (of course at a price of slightly reduced efficiency for reconstruction of good tracks).

Acknowledgments

The author wishes to thank Mike Ronan, who created the original reconstruction framework for use in the Java Analysis Studio.

References

[1] N.Sinev http://www.linearcollider.ca:8080/lc/vic04/abstracts/detector/simul/nick_sinev.ppt