A Fast Track Finder

Pablo Yepes
Bonner Nuclear Laboratory, MS 315
Rice University
6100 Main Street, Houston TX 77005

An algorithm for fast track pattern recognition is presented. Its main features are an optimized data organization and a conformal mapping to speed up fitting procedures. The code is applied to a generic collider detector. Reconstruction time, track finding efficiencies, and momentum resolution are presented.

Key words: Tracking ; Fast ; Recognition ; Trigger ;

1 Introduction

The complexity of events in particle and nuclear physics has increased greatly in recent years. When analyzing complex events, track information could be essential to deciding whether an event is interesting. Therefore, a fast way to reconstruct tracks could be a fundamental ingredient of a high-level trigger or off-line data filtering.

A fast track finding algorithm developed for a high level trigger is presented. The algorithm has been implemented in a generic program which can be applied to different tracking scenarios.

2 Algorithm

The program assumes particles follow helical trajectories. Therefore their motion in the plane perpendicular to the magnetic field can be described by a circle. Since fitting straight lines is easier and faster than fitting circles, it is desirable to transform the circular trajectories into straight lines. This is achieved by applying a conformal mapping to the coordinates in the plane perpendicular to the magnetic field.

A space point (x,y) produced by a track crossing the space point (x_t,y_t) is transformed: $x' = (x - x_t)/r^2$, $y' = (y_t - y)/r^2$, where: $r^2 = (x - x_t)^2 + (y - y_t)^2$. This transformation requires the knowledge of a point included in the track trajectory. If tracks are assumed to have been produced at the interaction point, IP, (x_t,y_t) is replaced by the IP coordinates. This case is referred to as *vertex constraint.* If no assumptions are made about the production point of a track, (x_t,y_t) is replaced by the coordinates of the first space point associated with the track. This option is referred to as *no vertex constraint.*

An essential component in a fast algorithm is efficient handling of the data. This is achieved by organizing space points in sub-volumes. Sub-volumes are obtained by segmenting the tracking volume into cubes in the r-ϕ-η space, where r, ϕ and η stand for the radius, azimuthal angle, and pseudorapidity. Each volume is identified by three indexes, i_r,i_ϕ,i_η, corresponding to the three coordinates used: r,ϕ and η. Each space point is uniquely associated with a sub-volume. The index in the radial direction plays a special role in the track finding and is referred to as *the layer.*

Linked chains of space points are established to allow fast loops over all the space points in sub-volumes and layers. A more detailed description of the algorithm can be found in [1].

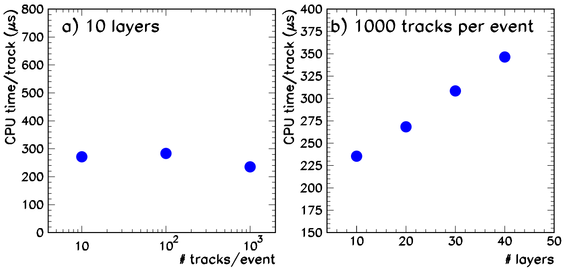

Fig. 1. a) CPU time per track as a function of the number of tracks per event for a detector with 10 layers. b) CPU time per track as a function of the number layers in the detector for events with 1000 tracks. Times are measured on an IBM RS/6000 43P-133 machine.

3 Results

A generic detector consisting of 10 concentric sensitive cylinders with radii ranging from 40 to 100 cm and centered around the beam axis is simulated with a simple model. Tracks are generated with a flat distribution in pseudorapidity in the interval (-2,2) and with an exponential transverse momentum (p_t) distribution $e^{-p_t/2}$. Tracks originate at the center (0,0,0) of the detector and propagate through a constant magnetic field of 0.5 Tesla. In the present study a spatial resolution of 300 microns is used. For simplicity, no physics effects, like multiple scattering, energy loss, etc., are considered.

Figure 1.a shows the CPU time needed to reconstruct one track as a function of the number of tracks per event in the case of a detector with 10 layers. CPU times are obtained running the code on an IBM RS/6000 43P-133 [1].

The CPU time required per track as a function of the number of layers in presented in Figure 1.b for events with 1000 tracks. As can be observed, the time required to reconstruct a track with 40 layers is less than a factor of 2 larger than the time needed for a 10 layer track.

Each reconstructed track is associated with the generated track with which it has the most space points in common. A generated track is considered well reconstructed if either of the following conditions are fulfilled:

(i) the associated reconstructed track contains at least 80% of the space points of the generated track.
(ii) the difference between the momentum of the generated track and that of the reconstructed track is less than 10%.

The reconstruction efficiency versus p_t obtained following the criteria above, is shown in Figure 2 for three sets of events with 10, 100 and 1000 tracks respectively for a detector with 10 layers.

The previous results are obtained with the track search performed in two passes. In the first pass a vertex constraint is imposed. The second pass looks to form tracks with the remaining hits without assumptions about the vertex. The main difference between both methods is the momentum resolution. When a primary track is reconstructed with the vertex constraint, the momentum obtained is much more accurate than when this piece of information is ignored. This is illustrated in Figure 3 where the momentum resolution is shown versus p_t for tracks reconstructed with and without vertex constraint in events with ten layers. The results do not depend on the number of tracks per event.

[1] SPECint92 176.4, SPECfp92 156.5

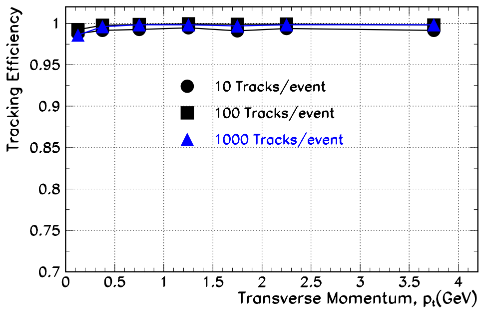

Fig. 2. Track finding efficiency as a function of transverse momentum for a detector with 10 layers and events with 10, 100 and 1000 tracks. Statistical errors per bin are 2%.

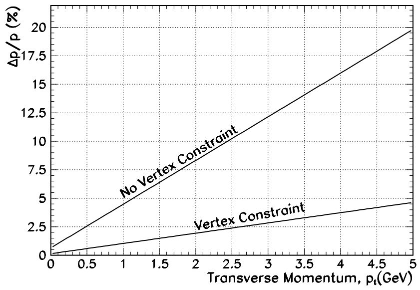

Fig. 3. Transverse momentum resolution with and without vertex constraint as a function of transverse momentum for a detector with 10 sensitive layers.

4 Conclusions

A very fast algorithm has been developed to reconstruct tracks from a set of space points. The time required to reconstruct tracks with 10(40) space points is 225(350) microseconds in events with 100-1000 tracks on a 176 SPECint92 machine. This very fast time is achieved maintaining a track finding efficiency above 98% and a good momentum resolution.

5 Acknowledgments

I would like to thank Dirk Schmischke and Michael Levine for their comments and support in developing the code and Geary Eppley for reading this manuscript. I am indebted to Helio DaMotta for his numerous suggestions and testing of the code. This work was supported by DOE under grant DE-FG03-92ER40772.

References

[1] P. Yepes, *A Fast Track Pattern Recognition*, Nuclear Instruments and Methods in Physics Research **A 380** (1996) 582 and references therein.