A Silicon Tracking Algorithm for Level3

Justin B. Kinney

Department of Physics, Cornell University, Ithaca, NY, 14853

August 1999

Abstract

The Level3 algorithm for the new CLEO III detector filters out uninteresting events, such as beam-wall events, before they are written to tape. It consists of two parts. One algorithm analyses output from the CsI crystals; the other uses information from the drift chamber trigger and the silicon detector to identify tracks left by charged particles. This summer I helped create the tracking algorithm as well as tools to aid in its upkeep and versatility. I then tested this algorithm against different Monte Carlo simulations. Results show a 97% acceptance rate for $e^+e^- \to \mu^+\mu^-$ events and a 90% acceptance rate for $e^+e^- \to \tau^+\tau^-$. With the current noise estimates, bad events, such as beam-wall events, are rejected over 98% of the time. However, when twice the expected noise is used this statistic drops to only 86%, demonstrating Level3's susceptibility to noise. The Level3 algorithm is still acceptable, but the current program requires too much memory and must be restructured before it goes online in November.

Introduction

The Level3 algorithm is our way of making split-second decision regarding which events are "interesting." In any given run, as few as one-tenth of all events may be worth even looking at. The job of Level3 is to filter out the rest of these events, before they are sent on to the Event Builder and written to tape. Because of time constraints the Level3 algorithm is only able to look at a portion of the data generated at each event. Level3 must be fast; each event must be processed in about one millisecond. In addition, the algorithm must be flexible and able to handle unforeseen problems such as unexpected noise, defects in silicon, and detector misalignment.

The first part of Level3 accepts events which leave more than about 2 GeV in the calorimeter. This is sufficient to accept events such as ordinary hadronic events. However, good events such as $e^+e^- \to \mu^+\mu^-$ and $e^+e^- \to \tau^+\tau^-$ typically will not leave enough energy in the calorimeter to pass the first algorithm's criteria. This is where silicon tracking comes in. All events not accepted by the first algorithm are passed on to the tracking algorithm for a second evaluation. The tracking algorithm's job is to filter out the rest of these good events from the bad ones, such as beam-wall events.[1] Our goal for the tracking algorithm was an acceptance rate of at least 90% for good events and a rejection rate of over 80% for the bad ones.

Tracking Strategy and Algorithm Overview

To differentiate between good and bad events we require at least two tracks originate

---

[1] Beam-wall events occur when a stray beam electron or positron hits the side of the beam pipe, usually causing a bunch of positively charged particles to spew fourth.

from the center of the beam pipe. Since beam-wall events typically do not produce more than two tracks pointing towards the center, this requirement is very effective.

First we make a map of the silicon detector, as shown in Fig. 1. We divide each layer of the detector into equal numbers of equally sized ϕ-bins. A given ϕ-bin is said to be hit if a substantial amount of charge is detected on any one of the r-ϕ strips in that bin. This provides Level3 with a convenient geometrical interpretation of silicon hit information. In addition, grouping multiple r-ϕ strips into single bins drastically reduces the amount of information Level3 must process at each event.

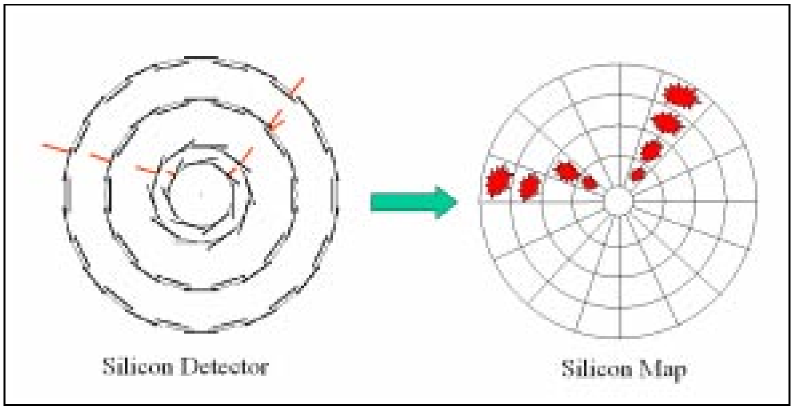

FIGURE 1. The silicon map is a simplified geometrical representation of the silicon detector. The radial lines in the diagram on the left represent hits in individual silicon strips. There are 511 such strips per wafer. These silicon strips are grouped into ϕ-bins, shown in the diagram on the right. A phi bin is "hit" if any strips in that ϕ-bin are lit from the passing of a charged particle.

The tracking algorithm first looks at the trigger information. If the drift chamber trigger detects the presence of two or more tracks, the silicon map is filled. The algorithm then determines, from the trigger information, which regions of the silicon map may contain evidence of good tracks. If two or more good charged particle tracks are found in the region identified by the trigger, the event is considered worthy.

After the algorithm is finished, a 32-bit word is sent to the event builder. This word contains a tag–a single bit which tells the event builder whether or not the event data should be stored. This word also contains other information, such as event type, number of tracks found, total energy, etc.

Algorithm Specifics

The silicon map consists of four layers (one for each silicon layer), each divided into 672 ϕ-bins. The trigger gathers information from the twelve innermost layers of the drift chamber, and projects the tracks onto the wires in the ninth drift chamber layer. Level3 requires, for geometrical convenience, that the number of ϕ-bins in each layer be an even multiple of the 112, the number of wires in ninth layer of the drift chamber.

Since one track may project onto as many as three different wires in layer nine of the drift chamber, Level3 clusters adjacent trigger signals together. It then looks for tracks in the regions of silicon, called track seeds, that lie directly below the middle of each cluster. Currently Level3 uses 224 track seeds, half positioned directly below the wires in ninth layer of the drift chamber, half positioned between these wires. Two examples illustrating how this works are diagrammed in Fig. 2 and Fig. 3.

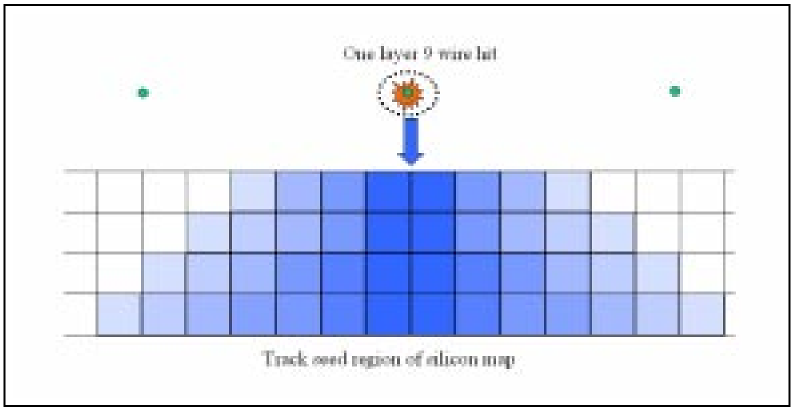

FIGURE 2. In this figure three wires in layer nine of the drift chamber are shown. The trigger tells Level3 that a track has gone past the center wire, and thus Level3 examines the track seed directly below it.

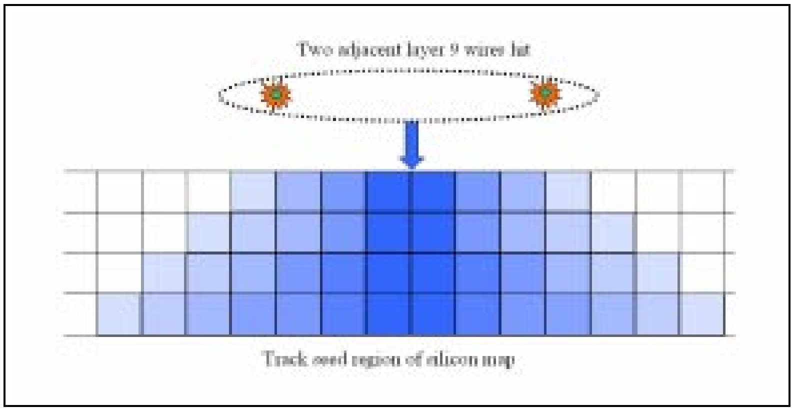

FIGURE 3. Here the trigger has detected tracks near two of the wires in drift chamber layer nine. Level3 clusters these wires together, assuming both trigger signals came from the same charged particle track, and examines the track seed positioned directly between them.

Each track seed contains a list of paths that charged particles, traveling though that region of silicon, can take. Paths consist of sets of ϕ-bins in either three or four of the silicon layers, as shown in Fig. 4. These paths are created by so-called path builders. During initialization, each track seed is passed to a set of path builders. Each path builder, in turn, attaches a set of paths to each track seed, that is, it adds paths to the list of paths inside each track seed. This process is represented in Fig. 5. This method of using path builders allows the user to easily create paths of different kinds and different shapes, and place them in different regions of each track seed.

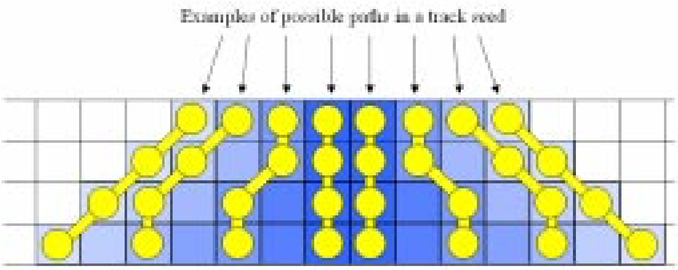

FIGURE 4. This figure shows a track seed containing eight 4-layer paths. Paths consist of multiple ϕ-bins, all of which must be hit in order for the path to be validated.

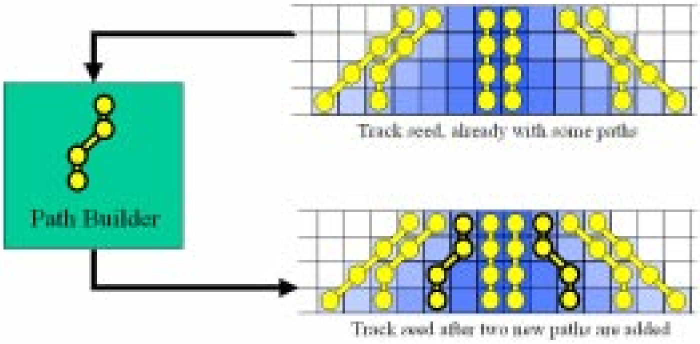

FIGURE 5. This figure illustrates how a Path builder adds paths to a track seed. The track seed in the upper part of the diagram is passed to the path builder. The path builder then attaches paths of a certain shape to different positions in the track seed.

When searching for a track, a track seed will check all of its paths to see if any one is validated. A path is considered validated if every ϕ-bin making up that path is hit. If any path in that track seed is validated, the track seed lets the algorithm know a track has been found.

Difficulties

In creating the Level3 algorithm we ran into a number of difficulties, the most serious being the expected inefficiency and noise in the silicon.

Because of the gaps between silicon wafers and the likelihood that some of the silicon channels won't work, each layer of silicon is expected to be about 97% efficient in detecting the presence of a tracks. This means a track will be picked up by all four layers only about 88% of the time. In order to increase the algorithm's efficiency in detecting tracks, we not only have to look for tracks detected in all four layers, but ones detected in only three of the four layers, so-called 3-layer tracks.

This leads, however, to the second problem: that of noise. Silicon noise, produced mostly by stray particles, is a grave concern for Level3. With the current noise estimates given by Dr. David Cinabro[2], about 0.8% of all silicon strips will be lit at any given moment. However, grouping them together into ϕ-bins causes a drastic increase in effective noise; at any given moment about 9% of the ϕ-bins will be hit. Noise is unlikely to randomly form valid 4-layer tracks, but will form valid 3-layer tracks a substantial fraction of the time. In order to help counteract the effect of noise, Level3 differentiates between hits on the East side of the detector and hits on the West side. This reduces the number of tracks produced by noise by about half. In addition, we require that at least one of the tracks from a good event leave traces in all four layers of the silicon. This substantially reduces the number of bad events that would be accepted due to noise.

Choosing Paths

The most important decision to make in calibrating the Level3 tracking algorithm is in deciding which paths to allow. Each 3-layer path has approximately the same chance of being validated by noise. Thus, the chance a track will be detected from pure noise is directly related to the number of paths each track seed uses. This produces a complex optimization problem: How do we maximize the number of good events accepted, while minimizing the number of accepted bad events due to noise? Obviously, the solution to this problem requires a careful, discriminating choice of paths.

In order to single out which paths are best used by the tracking algorithm I looked at Monte Carlo events. To get an idea of which paths should be chosen, I wrote a path-extraction program. This program goes though single pion events created by the Monte Carlo simulator and extracts at each event the "simplest" path that would, if it were being used by a track seed, be validated by the single pion. It does not pick the "best" set of paths–that is a complex computational problem which I did not have time to pursue. The program, however, gives very useful hints at which paths would be most useful and where they should be placed in the track seed.

By extracting single pion paths from over 34,000 events with momenta ranging from 5.0 GeV/c to 200 MeV/c, and then going though the results by hand, I was able to come up with a set 15 differently shaped paths. These paths detect single particle tracks, of any momentum greater than 275 MeV/c, with an efficiency of over 96%. These paths, however, would be very susceptible to noise. To avoid this I cut the number of differently shaped paths down to 8. Even with half of the paths gone, the algorithm still detected single particle tracks down to 400 MeV/c with an efficiency of over 96% for each momentum. We then decided that 400 MeV/c was a reasonable cut off for momentum, and proceeded to test these paths in the actual algorithm. Depending on the noise levels observed when the detector turns on

---

[2] Estimates for layer 1: 2.9%; layer 2: 1.7%; layer 3: 0.3%; layer 4: 0.2%

in November, we might be able to reduce this cutoff to as little as 250 Mev/c.

Results

Using the Level3 tracking algorithm on Monte Carlo simulations we obtained results shown in Table 1 and Table 2. $e^+e^- \to \mu^+\mu^-$ events were accepted 97.2% ± 1.1% of the time, and $e^+e^- \to \tau^+\tau^-$ 89.7% ± 2.0% of the time. Real beam-wall events, taken from CLEO II and translated to CLEO III with Dr. David Cinabro's noise estimates taken into account, were rejected over 98%[3] of the time. However, a rejection rate of only 85.8% ± 2.4% was obtained when twice the expected noise was used.

TABLE 1. Acceptance rates for good events

Event Type	Acceptance Rate (250 Events)
$e^+e^- \to \mu^+\mu^-$	97.2% ± 1.1%
$e^+e^- \to \tau^+\tau^-$	89.7% ± 2.0%

TABLE 2. Rejection rates for bad events

Event Type	Rejection Rate (250 Events)
beam-wall (expected noise)	> 98%
beam-wall (2×expected noise)	85.8% ± 2.4%

Conclusions

Level3's ability to accept $e^+e^- \to \mu^+\mu^-$ events is much better than we had expected. However, Level3's performance on $e^+e^- \to \tau^+\tau^-$ events is surprisingly mediocre. The reason for this discrepancy not known as of now.

Using Dr. David Cinabro's noise estimates we found that Level3 rejects virtually all beam-wall events. However, when these estimates are doubled the acceptance rate for beam-wall events shoots up dramatically, from less than 2% to 14%! Level3's performance with twice the expected noise is still acceptable, but these statistics demonstrate just how susceptible the algorithm is. So despite its trouble detecting $e^+e^- \to \tau^+\tau^-$ events and its unfortunate dependence on noise, the current algorithm has been deemed acceptable for use in November.

The current program, though, is not; Level3 presently uses over 525 MB of RAM at run-time. Level3's need for memory should be able to be reduced by at least half, but this

---

[3] The measured rejection rate of 99.5% should be taken with a grain of salt. During this run only one bad event was accepted. Because of statistical variation I might just as easily have observed two or three events. The true statistic may very well be as low as 98%, but due to time constraints I was not able to pinpoint this figure any more precisely. Thus only a value of > 98% is mentioned throughout this paper.

will require a significant restructuring of the program. I plan to undertake this revision when I return to Cornell in the fall.

The speed of Level3 has not yet been tested. This is unfortunate, but as of now the proper delivery systems required to test this are not working. However, I believe the current algorithm will be fast enough. With the revisions I plan to make in the fall the most time consuming part of the algorithm should be, by far, the reading of silicon and trigger information. The design of the algorithm performs very few calculations while checking for tracks, and provides for the possibility of using awkward but fast boolean path-validation to avoid calls to memory, if needed. However, the speed of this algorithm, for safety sake, should be checked as soon as possible.

Afterthoughts on Data Types

I was surprised to learn while writing this algorithm that there are virtually no standard data types, such as ring-like arrays, to use in working with the cylindrical geometry of the CLEO III detector. In order to deal with this in the Level3 algorithm, I created what I call a “WrapVector.” This is essentially an array that wraps around on itself so that the last element is adjacent to the first. I found this data type extremely useful, not only in representing detector geometry, but also in the algorithm itself. In the revision of the Level3 algorithm I plan to use these data types on a much more involved level and it seems to me that having a standard library of ring-like arrays and modulated integers (integers that only take on values from 0 to some number n) would be extremely useful in the development of future CLEO software.

Acknowledgments

I would like to thank Dr. David L. Kreinick of Cornell University’s Laboratory of Nuclear Sciences, and Dr. Tomasz Wlodek of Southern Methodist University for their part in writing the Level3 algorithm and the freedom they gave me to explore my own ideas on the matter. Thanks also to Dr. Chris Jones of LNS for his help and guidance in developing the algorithm’s structure, and to Dr. David G. Cassel of Cornell University for allowing me, at the last minute, to take part in this REU program. I would also like to take this opportunity to thank Professors Joe Rogers and Lawrence Gibbons of Cornell University for their guidance in helping me obtain this position at Wilison Lab. This work was supported by the National Science Foundation REU grant PHY-9731882 and research grant PHY-9809799.