SLAC-PUB-3442
CALT-68-1171
UI-HEPG-84-18
September 1984
(E/I)

A New Approach to Track Finding and Fitting in Vector Drift Chambers*

J.J. Becker,[a] J.S. Brown, D. Coffman, S. Dado,[b] J. Hauser
S. A. Plaetzer, J.J. Russell,[c] R.H. Schindler, J.J. Thaler

California Institute of Technology, Pasadena, CA 91125
University of Illinois at Urbana-Champaign, Urbana, IL 61801
Stanford Linear Accelerator Center, Stanford, CA 94305
University of Washington, Seattle, WA 98195

Abstract

We describe the algorithm that is used to find and fit charged particle trajectories in the Mark III detector at SPEAR. The computer program uses a novel non-numerical pattern recognition technique analogous to that used by the digital hardware in the experiment's track finding trigger processor. The technique is both fast and efficient. The complete reconstruction of events is performed at a rate of 37 ms per track on an IBM 3081K, compared with 91 ms per track with a more conventional technique. A preliminary fit of all tracks, suitable for online monitoring, is available after 15 ms per track. Similar techniques are also applicable to future experiments operating in high multiplicity environments. The organization of the algorithms is such as to lead to simple implementation on vector processors.

(Submitted to Nuclear Instruments and Methods)

⋆ Work supported in part by the Department of Energy, contracts, DE-AC02-76ER01195, DE-AC03-81ER40050, DE-AM03-76SF0034, DE-AC03-76SF00515, and by the National Science Foundation.

1. Introduction

Most current and planned detectors at colliding beam machines employ central tracking chambers composed of concentric cylinders of drift cells. In many of the chambers, these cylinders are grouped into clusters of sense wires, rather than being distributed uniformly along particle trajectories. Such a distribution allows one to make local vectors from each cluster of wires and in principle improves the pattern recognition capabilities of the chamber. The Mark III central drift chamber was designed in this manner, and two generations of software have been developed to utilize these features of the drift chamber geometry. In designing our current code, we were motivated not only by the desire to speed up the event reconstruction, but also by the desire to implement logic which performs global pattern recognition using bit manipulation rather than doing local point-to-point searches or global trial-and-error fits. The former is subject to confusion, and the latter is too slow.

The first stage of pattern recognition utilizes only simple cell hit information. The technique was guided by the structure of the Mark III trigger [1] which identifies circular tracks using 80 programmable array logic ICs operating in parallel, each loaded with the drift chamber cell patterns for all tracks passing through a specific portion of the chamber. While it is not possible to implement the parallel processing in IBM software, it is possible to emulate this style of pattern recognition. A dictionary is written which contains all 12832 cell hit patterns which correspond to real tracks. Then, for each event, the drift chamber hit pattern is compared with the dictionary entries. This method of track finding is fast, because the pattern recognition requires no numerical calculation. All the real tracks will be found, plus some spurious ones due to confusion. This approach enjoys several advantages over more conventional ones:

1. Bit manipulation is much faster than arithmetic and is more suited to the

pattern recognition stage of computation. Also, a nonnumerical track finder will be more efficient than one which manipulates the data numerically. There are no cuts that need to be fine tuned.

2. Because of the algorithms employed and more importantly the code's organization and structure, both the pattern recognition and track fitting are suitable for implementation on vector machines. This is the first reconstruction program for a solenoidal detector to have this property and, thus, that can utilize the highest speed computers.

3. The pattern recognition does not have to be limited to two dimensions or to information solely from the drift chamber. Similarly, no restriction needs to be made that the detector be cylindrical or symmetric. This means that our approach can be used in other experimental configurations as well.

4. Pattern requirements are easily changed, and the effects of such changes are easily understood, because they are digital.

5. The pattern recognition does not need an accurate set of detector constants and is not sensitive to unusual patterns of missing points or noise in the chamber. This makes it more useful in the setup and checkout phase of an experiment, when the apparatus may be malfunctioning or not well understood. Also, the drift chamber performance can be monitored with cosmic ray data, for which detailed calibration constants may be inconvenient or impossible to obtain.

6. Unreliable and spurious data is recognized and discarded before a great deal of time is spent on fruitless numerical procedures. Tracks which are eventually submitted to a detailed fitting program usually contain a largely valid set of data points. The fitting program is not employed to separate track candidates, resolve ambiguities on points, or to reject erroneous data.

7. The code is simple to debug and maintain. This is partly due to its non-numerical logic, which is easy to follow, and partly to its modular structure. The various stages of the track finding process naturally separate into simple procedures. Errors can usually be traced to problems in specific parts of the logic, not to an obscure numerical value or calculation.

In the second stage of pattern recognition remaining ambiguities among the candidate tracks are resolved by fast numerical tests using local cell quantities, and occasionally by a more detailed non-iterative circle fit approximating the exact track trajectory. These tests reduce the set of candidate tracks to the final set of real tracks, identify poorly measured points, and provide accurate starting values for the final iterative fit.

The iterative fit, which incorporates the details of detector geometry and response, calculates the momentum and trajectory of each track. Its execution time is significantly reduced by taking advantage of the chamber's clustered wire geometry.

2. The Detector

The Mark III detector and drift chamber have been described in detail elsewhere. [2] We give here a brief summary of the features which are relevant to the track finding. Figure 1 shows the main detector components. The drift chamber consists of 8 concentric cylindrical layers of cells, 6 axial with the beam, and 2 stereo. The radius of the chamber is 107 cm and its length is 239 cm. The cell structure is shown in Fig. 2. The cells in each layer contain several sense wires (3, 4, or 12). A particle which gets to the outermost layer, covering 75% of the solid angle, will be measured by 34 sense wires. A particle going out the end will be measured by fewer wires, however 95% of the solid angle is covered by at least 19 sense wires. Tracks having large dip angles may leave the chamber before passing through any stereo cells. The reconstruction of these tracks is aided by the use of information from the electromagnetic calorimeter, which covers 96% of the solid angle and has about 1 cm resolution, adequate for determination of the dip angle.

3. Pattern Recognition

The search for tracks is first made in the transverse plane (r-ϕ in cylindrical coordinates). The Mark III detector provides more accurate information in this projection than along the longitudinal (z) direction, making pattern recognition in r-ϕ more reliable. Z information is associated with tracks only after all are found in the r-ϕ plane.

The track dictionary is generated by a "geometry" program which draws circles from the beam line through the detector and notes which sets of drift chamber cells lie on each. For the purposes of pattern recognition, a cell is considered hit if 2 of 3, 2 of 4, or 8 of 12 sense wires were struck. This cell definition reduces sensitivity to drift chamber inefficiencies, and the occasional random hits due to noise. The method is illustrated in Fig. 3. Each dictionary entry is one distinct set of these cells. To keep the dictionary small we only draw circles which correspond to transverse momenta greater than 50 MeV (a particle needs 70 MeV to reach the outer radius of the drift chamber). With this restriction, the dictionary contains 12832 entries. The program scans through the dictionary faster if the entries are ordered first by cells in the middle layers, and then in the outer and inner layers.

The dictionary also contains a list for use in suppressing spurious tracks. This auxiliary list is needed because inefficiencies in the drift chamber can render cell hit patterns ambiguous; a pattern with a missing cell might be a piece of any of several tracks. The auxiliary list is a cross reference between tracks which will become indistinguishable if cells are missing from them. When a particular pattern is found, all other tracks which are formed from the same pattern or its subsets are suppressed.

Because data from the detector is unpacked cell by cell, it is natural to structure the dictionary not only as a list of cells on each track, but also inversely as a list of the tracks that pass through each cell in the chamber. During the data analysis, as each hit cell is identified, the program sets bits in a two dimensional bit array called PATARY (one row for each layer in the drift chamber and one column for each track in the dictionary). For each hit cell one bit is set for each track that might have caused the hit. These bits then indicate which of the drift chamber layers on any given track are actually hit. To be specific, in the Mark III detector there are eight concentric layers of cells, so PATARY is an 8 row by 12832 column array. (Each column is thus one byte of computer memory.) A perfect track (no drift chamber inefficiency) will generate a column of 8 bits in this array and also some other columns with missing bits, corresponding to other tracks that pass through some of the same cells (see Fig. 4). Drift chamber inefficiency will also generate columns with some missing bits. The auxiliary track list helps to distinguish between imperfect tracks which are real and those which are merely subsets of a more perfect pattern of hits (see Fig. 5a).

With PATARY filled, the problem that remains is to identify columns (bytes) which have patterns of cell hits satisfying the criteria demanded of good tracks (see Fig. 4): enough cells hit to give a good fit and some measure of continuity. While the criteria chosen are digital rather than numeric, even a digital test performed separately on every PATARY byte takes too long. To reduce the time, another array is filled at the same time as PATARY. This array, called OVRSEE, has 12832/32 bytes (one byte per 32 tracks in PATARY, see Fig. 4), and can be thought of as a small extension of PATARY. At the same time that a bit in PATARY is set, a corresponding bit in the proper byte of OVRSEE is set (the bits to set in OVRSEE are also part of the dictionary), so that OVRSEE ends up with each byte containing the OR'ed patterns of 32 contiguous bytes in PATARY. Thus, we can test for the possibility of a track in any set of 32 (in PATARY) by testing the correct byte in OVRSEE. This saves time, because the bits set in PATARY tend to be sparse, so most regions of PATARY do not have valid patterns even when 32 tracks are OR'ed together. If OVRSEE has a valid pattern, then the 32 tracks are individually tested.

As was mentioned, allowing for drift chamber inefficiency may cause a particle's trajectory to appear as several valid track patterns. To avoid this, we search for tracks in several passes through PATARY. In the first pass only tracks with no missing cells are found, and when a track is found all tracks in its auxiliary list are inhibited. The second pass then finds all single-miss tracks that were not previously inhibited; their auxiliary tracks are then inhibited. This continues up to tracks with two missing cells, and finds all the real tracks together with a class of spurious tracks, namely those which take some cell hits from one real track and some from another (see Fig. 5b). In the Mark III detector the average charged multiplicity is 4, so this is not a serious problem, because the amount of confusion depends on the ratio of the cell size to the typical distance between tracks. The technique can be expanded to a higher multiplicity environment by subdividing the cells in software. Finer segmentation would tend to increase the size of the dictionary, offsetting the decreased size due to the higher average particle momentum at higher beam energy.

Up to this point, the drift time information has not been exploited. Tracks consist merely of sets of cells with hits. Before proceeding to detailed track fitting, we take advantage of the fact that the multiple sense wires in a cell both resolve the left-right ambiguity and determine the slope of the track at the measured point (see Fig. 2). Resolution of the left-right ambiguity is a notorious absorber of time in many track fitting algorithms; we separate this problem from the track fitting by resolving most ambiguities locally. Because each cell has its sense wires staggered by 200-400 microns, local ambiguity resolution is achieved by a simple set of linear constraints among the drift times. These constraints also allow us to identify and discard bad data. If a cell, or two adjacent cells, satisfy the constraints (true for about 95% of hit cells) the drift times are grouped into what we call an unambiguous OBJECT. If local resolution is not possible, all reasonable choices are retained in what we call an unresolved OBJECT. OBJECTs are data structures which contain the ambiguity information, track direction and curvature (calculated assuming a track coming from the beam line), pointers to the raw data, and a quality code indicating the internal consistency of the data. Track direction is determined with an accuracy of 10 milliradians and curvature with an accuracy of between 0.02 and 0.2 meter^{-1}, depending on distance from the beam line. Unresolved OBJECTs contain the information for each possible ambiguity resolution. The use of this information will be discussed below.

There are two principle reasons why we have abstracted to the concept of an OBJECT. First, all local information is evaluated once and retained through the analysis, thus avoiding the need to repeat arithmetic calculations. Second, information is compressed into a few numbers corresponding to the properties of the OBJECTs rather than the wires that make them up. In the Mark III chamber there can be 8 OBJECTs on a track containing 34 wires. Both of these reasons lead to simplicity in the flow of information and to considerable savings in execution time.

The next step is to replace each cell on each track with its corresponding OBJECT and sort the tracks into BUNDLEs, which are sets of tracks which share one or more OBJECTs. The purpose of BUNDLEs is to simplify the process by which the initial set of candidate tracks is reduced to the correct final set of tracks. Tracks in different BUNDLEs cannot share hits and thus are completely independent. Subsequent pattern recognition is confined to individual BUNDLES, allowing a reduction in execution time.

Before fitting the tracks in r-ϕ, OBJECT ambiguities are resolved, the correct tracks are selected from the candidates in each BUNDLE, and inconsistent time measurements on these tracks are eliminated. The general procedure for resolving ambiguities is to form a discriminator using the curvature and direction information associated with each OBJECT. For each ambiguity choice we evaluate a χ^2 measuring its consistency with the direction of the candidate track at the OBJECT. The procedure is fast, as it only involves the calculation of two sums for each choice. We continue to examine only one possible choice if χ^2 discrimination is good, or several if the it does not provide sufficient discriminating power. In BUNDLEs with only *one* track there is a unique choice most of the time, allowing us to proceed immediately to fitting. In the Mark III data, a large fraction of the events have only isolated tracks and can be fit immediately after the ambiguity resolution. Even in higher multiplicity events some tracks are isolated and can be quickly identified. Only when there are *two or more* real tracks near each other do the BUNDLEs become complicated.

In BUNDLEs containing more than one candidate track, the correct choices are made by assigning each shared OBJECT to one particular track. We first order the tracks in a BUNDLE by track length, track continuity, and OBJECT χ^2. Then, starting with the most reliable track, shared OBJECTs are tested and assigned to specific tracks. In a large fraction of the cases, the assignment of an overlapping OBJECT to one track or the other can be done by comparison of the OBJECT's attributes with the average properties of both tracks. For example, if two tracks cross within a cell, the local tangent determined by the OBJECT in that cell is likely to align much more accurately with one track than the other. The comparison involves precomputed quantities and takes little time. If an assignment is unambiguous, the OBJECT is removed from one track. If the OBJECT attributes do not discriminate well enough, if the track appears to have erroneous information on it, or if any OBJECT ambiguities remain, we resort to the more powerful discrimination of a fast non-iterative circle fitting program. [3] This fit minimizes an estimator which is equivalent to a true chisquared in the limit of small residuals, but which can be minimized analytically, unlike the true chisquared. Points which make an anomalously large contribution to the estimator are deleted from the track. Inclusion of these erroneous points in the final iterative fit would seriously degrade the speed of the program.

Upon completion of a pass through the list, the tracks which no longer overlap other tracks are extracted and passed to a final track list. The fast circle fitter is applied to all tracks in this final list to generate accurate starting parameters for the slower iterative helix fit. The remaining tracks in the BUNDLE are then tested to be sure that they still satisfy minimal requirements for valid track patterns. Those tracks that have had OBJECTs deleted in favor of other candidate tracks often vanish, so the BUNDLE rapidly shrinks. If the BUNDLE still contains more than one track, it is resorted and another pass is made through it. The procedure continues until no remaining tracks share any OBJECTs. The ordered scan through the BUNDLE quickly eliminates the spurious tracks, because the track at the top of the list is the one most likely to be real and thus to be assigned most of the OBJECTs. In our method most tracks candidates are not subjected to any fitting at all, because their points become used up by higher quality tracks. This ability to postpone numerical processing of tracks as long as possible is an important component in the speed of the algorithm.

The algorithm just described is very fast because it does not employ complicated numerical computation, but rather uses simple analytic quantities for discrimination. Implementation of the method is straightforward, if appropriate data structures are utilized. For example, all of the information resides in doubly linked lists, eliminating edge effects and other complications in scanning through the data.

4. Z Reconstruction

When every BUNDLE has been decomposed into its valid tracks, and these have been successfully prefit in the x-y projection, the z reconstruction starts. The object of the z pattern recognition is to associate the most reliable and accurate z information possible with each track. It is assumed that the track has been found in the axial projection, and that the parameters in that projection are fairly well measured. Delaying z reconstruction until after the preliminary x-y fit allows us to use the fit information to help reduce the ambiguities usually associated with stereo track information. The search for z information proceeds in decreasing order of data reliability.

The best z information is obtained when both stereo layers can be used. The procedure of associating tracks with full stereo information has three steps. First, a list of all possible associations of tracks with stereo hits is formed including any stereo OBJECT ambiguity. Next, for each track which has some unshared stereo hits the best combination is selected by forming χ^2 discriminators from the axial track parameters and the information associated with each stereo OBJECT. If there is clear discrimination of one choice, it is selected. Finally, for those cases where stereo information may be shared between two or more tracks, every possible combination is tried, deciding on z association only in those cases where the χ^2 from one choice is clearly favored. The OBJECT is then assigned to one track and deleted from the others. This procedure finds full stereo information for 75% of all tracks, somewhat less than the 84% geometrical coverage of the stereo layers.

After the assignment of full stereo to tracks, those tracks are passed to the final helix fit. The event vertex is found using the fitted tracks, and it is used to aid in the Z reconstruction of tracks which do not have full stereo information.

With tracks having one stereo layer (11% of all tracks), one can form a χ^2 using that layer and the measured event vertex position. All tracks which have one or more possible single stereo assignments are tested, looking for clear choices consistent with the vertex. In cases where such a choice is made and there is a hit cell in the other stereo layer consistent with the track, an extrapolation to the other stereo layer is performed, allowing either ambiguity choice. This solves, for instance, cases where the left/right ambiguity resolution was erroneous due to a bad time measurement. If the extrapolation is successful, one then assigns both stereo layers to the track. If there is no possible extrapolation, as is the case when the track left the chamber before the second stereo layer, then the vertex position (with appropriate errors) and the single stereo measurement are used in the final fit.

If no stereo association is possible for a track, then one attempts to find useful charge division information from layers 1, 3, 5, or 7. The vertex position may be used as described above if the information is inadequate or confused. 1% of all tracks are fit in this manner.

If the charge division association cannot be made (this is the case with tracks which quickly leave the chamber out the end), we look for a shower in the end cap calorimeter which is consistent in position with the axial projection and is near the end of the visible track in the drift chamber. If such a shower is found, its position is used along with the vertex to determine the z parameters of the track. 4% of all tracks are fit in this manner.

If no z association is possible at all (4% of all tracks), the track cannot receive a helix fit. Information from the circle fit is entered into the track list, together with flags indicating the lack of z information.

5. Short Track Finding

The shortest track found by the technique described above must extend through layer 3 of the drift chamber. Thus, tracks exiting in layer 2 will not be found. After all dictionary tracks have been found and fit, the OBJECTs utilized are deleted from the OBJECT list, and we then search for remaining OBJECTs in layers 1 and 2 that align radially. Aligned pairs of OBJECTs are circle fit, and the fit is extended into the outer layers to pick up any other objects that might exist on the track. These tracks generally have only the endcap and the event vertex for dip angle determination. They are passed to the final fit along with the other tracks. The forward direction is usually the most confused region of the chamber. Consequently, postponement of pattern recognition on short tracks until after the removal of longer ones reduces confusion considerably and allows the use of the slower trial and error technique employed.

6. Final Fit

After the pattern recognition and prefitting, a final combined fit must be made to properly account for the inhomogeneous magnetic field, multiple scattering in the beam pipe and chamber gas, and details of the detector geometry and drift chamber response. The fitting technique is a modification of the iterative piecewise helix fit developed for the Mark II detector. [4] This is a least squares fit in 3-space which takes advantage of the local field uniformity to propagate an orbit as a set of linked helices.

Linearization of the minimization procedure leads to an iterative fit for the 5 helix parameters characterizing the trajectory. Iteration of the fit is time consuming, so some care is worthwhile. Three improvements to the code have proved particularly useful. First, use of the noniterative circle fit discussed above removes faulty information and provides accurate trajectory parameters as starting values for the helix fit. This eliminates the need for extra iterations which would result from poor starting values or from the presence of spurious data points. Second, the orbit propagation is organized to take maximum advantage of the clustered wire geometry. The fit must calculate a residual and its derivative with respect to the helix parameters at each wire. Since wires are clustered into layers that are about 13 cm apart, but have wires within each layer only about 1 cm apart, the natural organization is to propagate the orbit to the center of each layer using the full 3-space calculation and then to make fast local orbit approximations to derive the residuals and derivatives for the nearby wires in the layer. The use of a local analytic solution where possible reduces round off and truncation error. This can be done with no loss in precision and a 25% savings in execution time. Third, subroutine and function calls are minimized by replacement with in-line code where possible. These three improvements have resulted in a reduction in execution time by about 55% compared with the first generation fitter. The code will perform the helix fit in an average of 17 ms on an IBM 3081K for a track with 34 measured hits.

7. Conclusions

We have described a novel technique of pattern recognition which relies on a non-numerical algorithm to find tracks in the Mark III central drift chamber. This technique, when combined with an improved fitting package results in a threefold increase in speed and an improvement in tracking reliability over a conventional approach to pattern recognition and fitting. While the program is implemented for a serial processor, the code's organization would naturally adapt to implementation on a vector machine. These techniques are applicable to most of the next generation of tracking detectors which must cope with denser track environments.

References

(a) Present address: ESL, Sunnyvale, CA 94088.

(b) On leave from: Technion-Israel Institute of Technology, Haifa, Israel.

(c) Present address: International Imaging Systems, Milpitas, CA 95035.

1. J.J. Thaler, *etal*, IEEE Trans. Nucl. Sci. **NS-30**, 236(1983); J.J. Becker, Univ. of Illinois PhD Thesis, (1983), unpublished.

2. D. Bernstein, *etal*, SLAC Pub 3222, 1983, submitted to NIM; J.R. Roehrig, *etal*, SLAC Pub 3199, 1983, submitted to NIM.

3. J. Crawford, Nucl. Inst. and Meth. **211**, 223(1983).

4. G.H. Trilling and A.D. Johnson, LBL internal memo TG-301 (1978); R.H. Schindler, SLAC Report 219, (1979).

Figure Captions

1. The Mark III Detector.

2. Drift Chamber Cell Structure, with Ambiguity Resolution and Track Vectors.

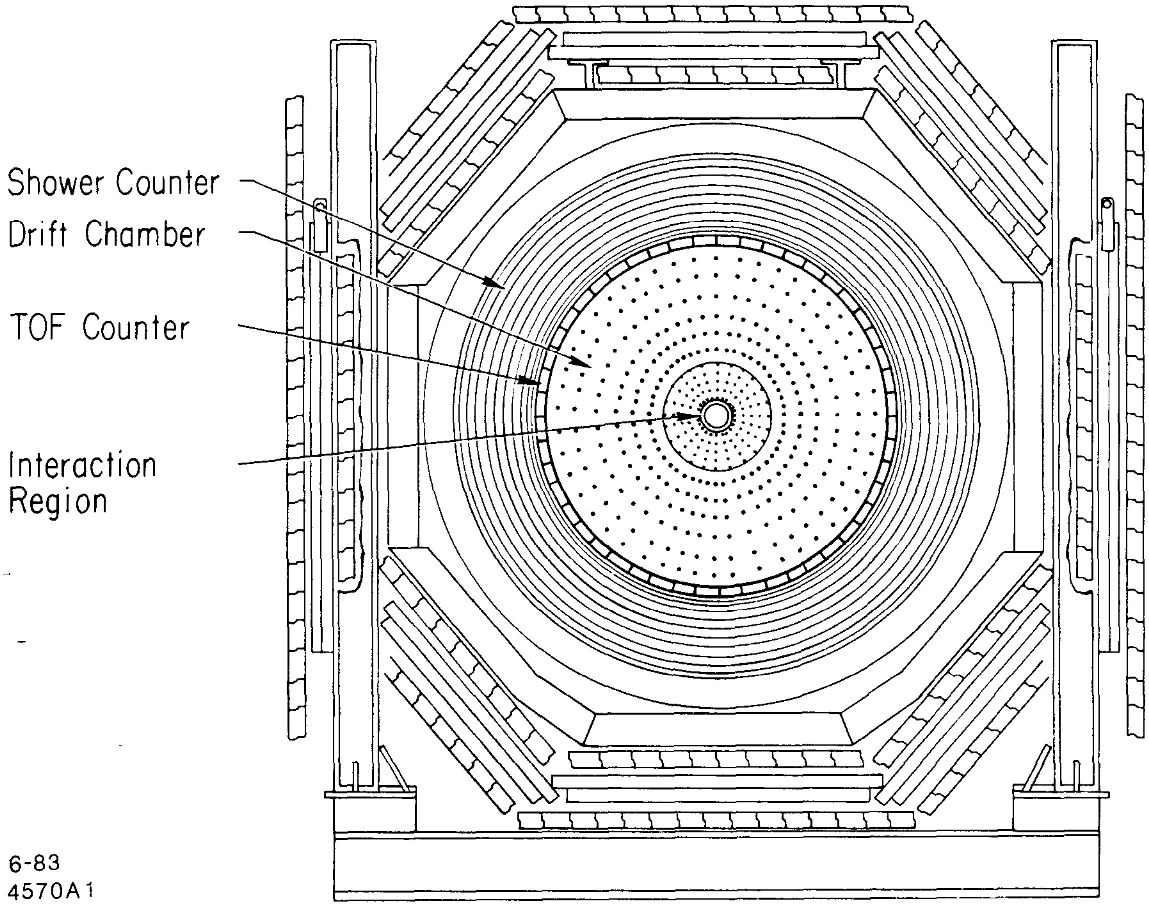
3. Schematic Representation of Dictionary Generation.

4. Bit Pattern Generated by a Track.

5. Sources of Confusion:

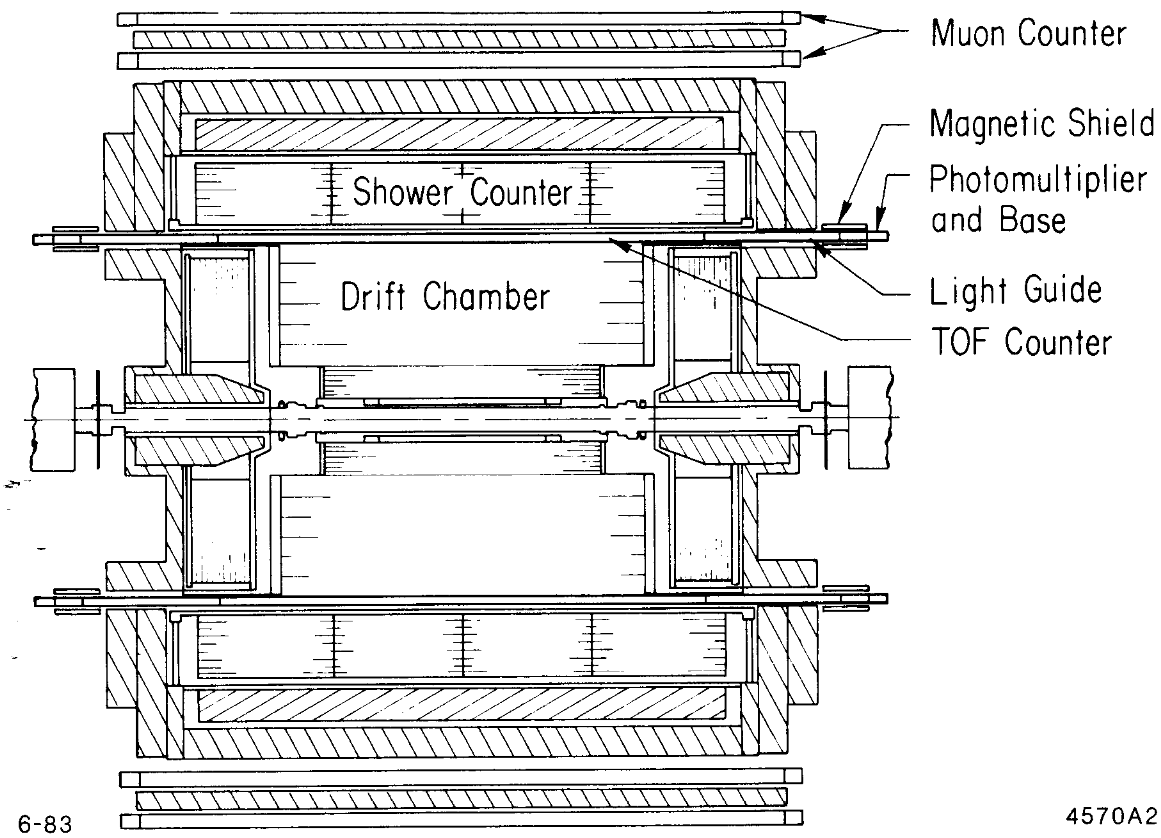
   a) With layer 8 missing, tracks i, j, and k become indistinguishable;

   b) Track i shares 5 cells with track j and 2 with track k. Its pattern passes the test.



6-83
4570A1

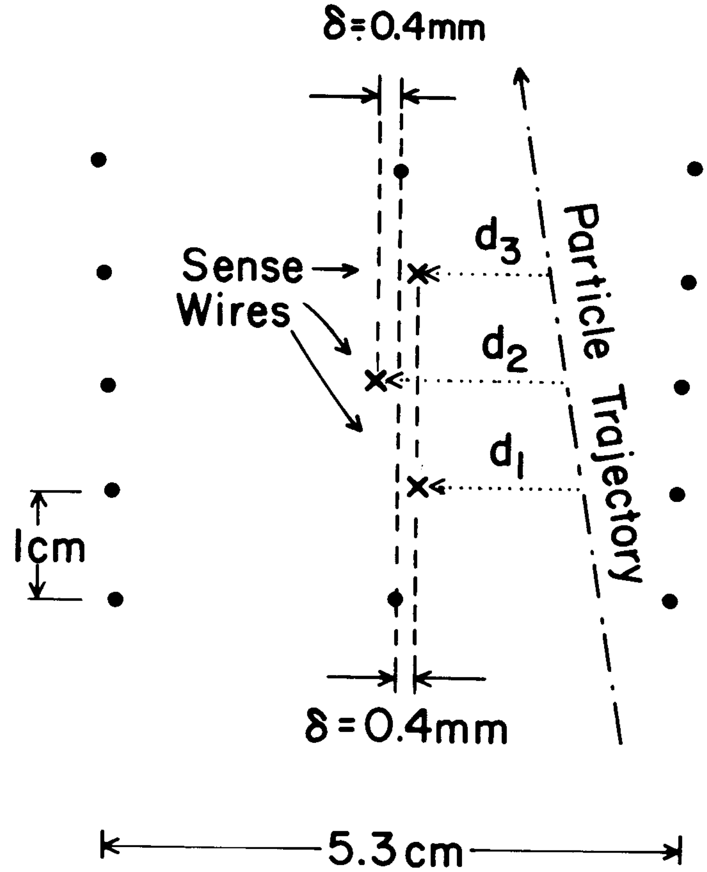
Fig. 1(a)



Muon Counter
Magnetic Shield
Photomultiplier and Base
Shower Counter
Drift Chamber
Light Guide
TOF Counter
6-83
4570A2

Fig. 1(b)

3-wire cell:



Ambiguity resolution: $\Delta = \frac{d_3 + d_1}{2} - d_2$

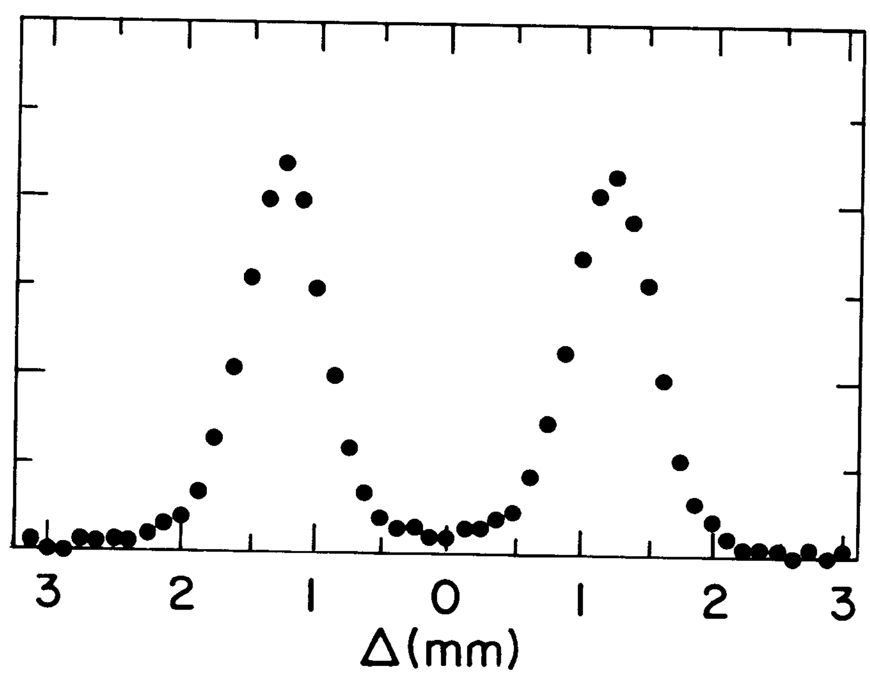
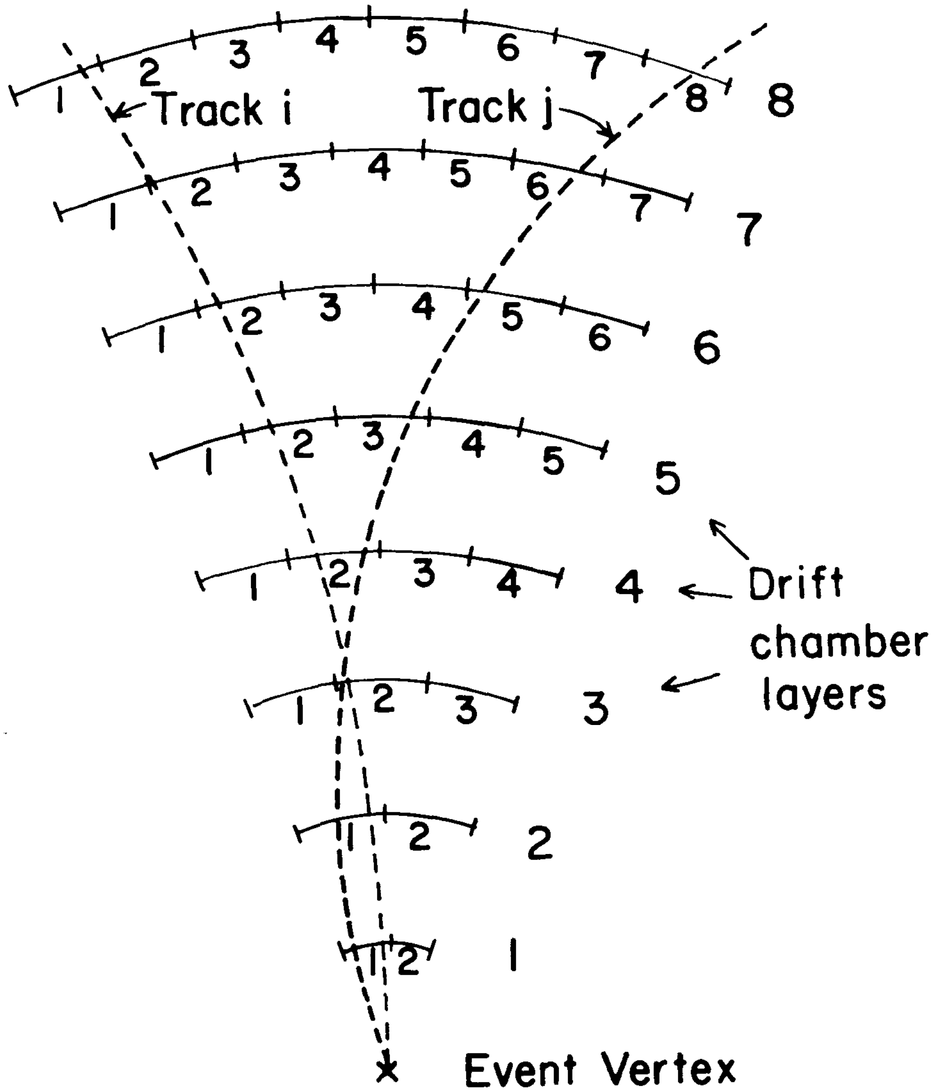


Fig. 2

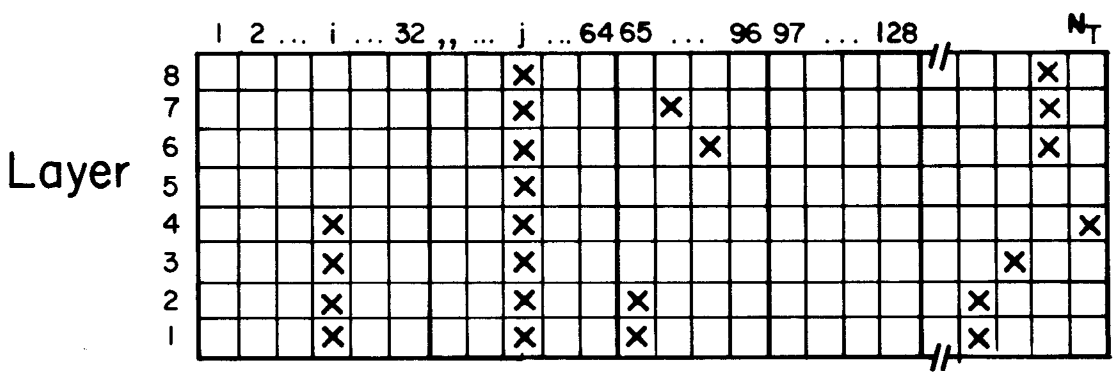


Track i = (1, 1, 2, 2, 2, 2, 2, 1)

Track j = (1, 1, 2, 2, 3, 5, 6, 8)

Fig. 3

Patary:



× denotes a bit that is set.

Ovrsee:

Byte	Contents	Result of Pattern Test
1	00001111 = ϕF	Fails
2	11111111 = FF	Passes
3	01100011 = 6F	Fails
⋮	⋮	⋮
N	11101111 = EF	Passes

$N = N_T/32$

Fig. 4

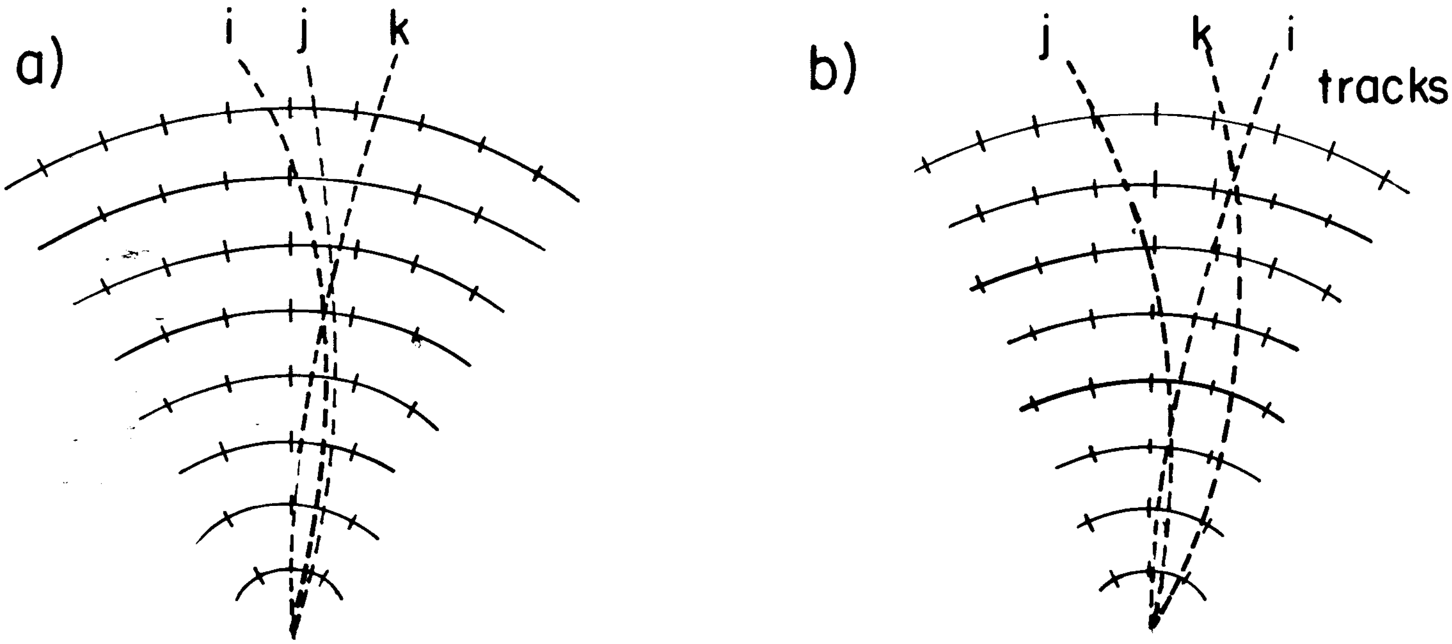


Fig. 5