A NON-NUMERICAL METHOD FOR TRACK FINDING IN EXPERIMENTAL HIGH ENERGY PHYSICS USING VECTOR COMPUTERS

C. H. Georgiopoulos *

Supercomputer Computations Research Institute

Florida State University, Tallahassee, FL 32306-4052

December 6, 1985

A non-numerical vector algorithm is used to reconstruct charged particle trajectories in the E -711 detector at Fermilab.The program is written for a CYBER 205.The vectorized version of the code is approximately 10 times faster than that of the previous scalar codes. The techniques used are applicable to other spectrometers relying on wire chambers for tracking information. The average event takes 7.7 msec to process in vector mode on a CYBER 205 compared to 1.6 sec in scalar mode on a VAX-11/780.

1. Introduction

Current detectors of fixed target as well as colliding beam experiments rely heavily on wire chambers for track finding. It has become almost folklore among high energy physicists[1] that track reconstruction is not a suitable problem for vector computers. We present here an existence proof that such is not the case, for a fixed target experiment recently run at Fermilab.

We have developed two generations of track reconstruction software. The first generation consists of two separate algorithms, implemented, on a VAX-11/780 which are numerical in nature, that is they do local point-to-point searches and require algebraic calculations. These two algorithms are completely different and serve to cross check the efficiency of the track finding. Both algorithms give the same results, at a rate of 1.6 CPU sec per event. The second generation of software was developed for a CYBER 205, is non-numerical in nature, and uses the vector features of the CYBER 205 to improve the speed of data processing. We were led to develop an algorithm that performs global track finding using bit patterns rather than local point searches. All the numerical calculations are done just once, prior to the analysis, which then finds all the possible tracks in one sweep. We have found that for this particular algorithm a fully reconstructed event takes approximately 7.7 msec, 200 times faster than the code developed for the VAX-11/780. Since the scalar mode of the CYBER 205 is no more than 20 times faster than that of the VAX-11/780, we have gained approximately a factor of 10 in speed from vectorization.

Section 3 describes the algorithm used to do the track finding on the CYBER 205. This particular method of non-numerical calculations enjoys several advantages over numerical ones:

(1) The CPU time consumption of this kind of vector algorithm does not depend on chamber multiplicities. For scalar numerical methods the time consumption increases dramatically with higher multiplicities. This type of algorithm is attractive for present and future collider experiments, where much higher multiplicities are expected.

(2) Code development and maintenance is simplified, since vectorized logic is easier to debug due to the non-numerical nature of the calculations.

(3) The track finding does not really need an exact set of alignment constants and moreover it is mostly insensitive to noise and spurious hits as far as CPU time consumption is concerned.

(4) Unusual patterns of hits and noise can be easily recognized and removed before time is wasted in the next stages of analysis.

2. The E-711 Detector

The tracking system of E-711 at Fermilab consists of four drift chamber stations. The first two are mini-drift Proportional Wire Chambers (PWC's) (they are PWC's but with time digitization to interpolate between wires). Each station has four views (x,y,u,v) with a projective cell size geometry focusing on the event production target. The cell sizes are 2.03, 2.53, 4.17 and 5.56 mm for the x and y views in stations 1,2,3,4 respectively, while for the u and v views the cell sizes are 2.03, 2.53, 4.11 and 5.47 mm respectively. Only the x and y views are truly projective, while the u and v are approximately so. The whole system consists of about 5700 wires. There is a magnetized volume between the target and the tracking stations which bends charged tracks in the horizontal plane (x-z). Fig. 1 shows the main components of the detector, while Fig. 2 shows a scaled down projection of one of the bend views.

The data is recorded on magnetic tape in coded bit pattern format (16 bit words written on a PDP-11). We found that the data unpacking vectorized rather easily, using the special features of the CYBER 205 to split the 64 bit words into 16 bit words and then decode the data further. Thus, the relative speed of the program should increase since the packing of the reconstructed data into an output format (not yet implemented) will be done in the same fashion.

3. Track Finding

In trying to use vector techniques to speed up the processing we found that straight or syntactic vectorization of an existing scalar code does not work. Timing the scalar code and then trying to vectorize those parts of the code that take most of the time does not improve the speed of the track finder. Basically this is because the problem is not how to "translate" the code into vector code, but rather how to develop new algorithms amenable to vectorization. The approach developed uses the idea of projective geometry in a non-numerical fashion. We have also written a scalar version of this algorithm but as expected it runs far slower (about ten times) than the original scalar versions.

The E-711 apparatus was originally conceived to be a projective geometry apparatus, that is the cells of the drift chambers were to focus on the production target. However, due to hardware construction problems, only the x and y views are actually projective. The two slanted views do not focus back on the target. Thus, in order to use the projective geometry algorithm, we map the wires into a set of fictitious chamber wires with cell size chosen to project back to the target. This kind of mapping is very fast since it utilizes the vector nature of the CYBER 205. Consequently, other spectrometers using wire chambers for track finding can also use this method.

The search for tracks is first made in each of the four projections (x-z, y-z, u-z, v-z) independently. Note that it is essential to our algorithm that truly independent projections exist, i.e that a collection of planes all be aligned at the same angle. This search is the most time consuming part of the analysis. The 2-dimensional track projections thus found are later matched up to form 3-dimensional tracks, since any two projections will predict the other two.

In the track reconstruction phase the chambers are treated as PWC's, that is the drift distance information is ignored. This coarsening of the spatial resolution leads to the finding of many spurious track candidates, which are rejected later. The drift distance to the wire is used later in the analysis to improve the momentum resolution during the track fitting phase and to reject spurious tracks and spurious hits on real tracks.

For each projection we proceed by first creating "roads" through the four drift stations. The roads are three wires wide to take care of edge effects and to allow for slight misalignments of the different drift stations. These misalignments are actually necessary later in the analysis to resolve the left/right ambiguity in the drift distance direction. The three-wire-wide clusters are stored in array **JAND(N,M)** (half precision floating point), where **N** is the central wire number (runs from 1 to the no. of wires in the plane) and **M** is the drift station number (runs from 1 to 4). For example, for the road consisting of wires 15,16 and 17 of station no. 3 with wires 15 and 17 on, the value of **JAND(16,3)** will be non-zero (it will actually be a bit pattern which preserves the wire pattern for later use). One can create roads of any size depending on the resolution or the specific requirements of the detector.

The next step in the analysis is to try to form tracks out of these roads. A previously run program is used to calculate all possible wire combinations that form a track. At this stage all of the alignment information is put in and all of the algebraic calculations are dispensed with once and for all. The results are preloaded into an array called **ISHIFT(M,L)**, where **M** is the station identifier (runs from 1 to 4) and **L** is such that the value of **ISHIFT** provides an index to the **JAND** array and thus uniquely identifies that particular track combination. For example, if wires 3, 4, 6 and 6 for stations 1, 2, 3 and 4 respectively, form a track(see Fig. 2 track j), then the values of **ISHIFT(M,L)** will be 3, 4, 6 and 6 for **M** = 1, 2, 3 and 4 respectively, while **L** will uniquely identify this set of wires.

The vector features of the CYBER 205 are now used over the variable **L** of the **ISHIFT** array. Four vectorized loops are created, one for each value of the station index **M**, and an array called **ISUM(K)** (half precision floating point) is incremented by one count whenever **JAND(ISHIFT(M,L),M)** is non-zero. Here **K** is a counter that also uniquely points back to this particular track. For example, for the case of track j in Fig. 2, **JAND(3,1)**, **JAND(4,2)**, **JAND(6,3)** and **JAND(6,4)** will be tested for non-zero values and **ISUM(K)** will be set to four counts. We now make use of the fact that the CYBER 205 is bit addressable in vector mode. Bit vector **BISUM(K)** is set if **ISUM(K)** is greater than or equal to three, i.e. a "track" is an alignment of at least three clusters. At this point all the possible tracks have been found. They are the non-zero bits of **BISUM(K)**. The dimensions of **BISUM** are 47180, 10268, 16045 and 16291 for the y, x, u and v views respectively. These dimensions are the number of different tracks that can be found in each view.

Next one can proceed in either of two ways: either go back and look at the tracks (which are now all contained in the BISUM array) and try to remove any hits which are spurious due to the coarseness of the three-wire-wide roads, or else proceed with linking up the 3-dimensional tracks and leave all track cleanup to the next stage of the analysis, which is the track fitting. We have chosen to remove some of the most obvious spurious hits first. The track pointers are gathered in vector mode (using the **GATHER** function on the **BISUM** arrays) and the bit patterns of the tracks found are compared to preloaded patterns. Thus, one tries to remove extra noise, see if a track can be split into two tracks, etc.

Upon completion of this task we assemble the tracks found in the four 2-dimensional projections into 3-dimensional tracks. For the case of E-711 only about 3.5 tracks per event survive from the 15 tracks per view per event originally found. The hit multiplicity in the individual chambers is high due to photon conversions and stray radiation downstream of the magnets. The number of 2-dimensional tracks reconstructed is high due to the coarse nature of the "roads" employed, which leads to a high probability for assembling unassociated hits into a track candidate. After the 3-dimensional matching, the final step is to replace the track list of projective wire numbers with a list of the actual physical wire numbers. We then can proceed with further stages of the analysis, e.g. track fitting, vertex finding, etc. This will be the subject of a later publication.

4. Conclusions

We have described here a non-numerical track finding technique implementing the use of a vector computer that effectively gains a factor of ten in speed compared to scalar mode algorithms. We expect this technique to be even more valuable for higher multiplicity experiments, such as exist at various collider detectors. Other algorithms which appear to lend themselves to vectorization are also under consideration at the moment. For example, an approach developed by the MARK III group at SLAC [2] appears very promising. In any case, development of vector algorithms is under way for the more complicated problems of track finding in collider experiments with charged tracks inside a magnetic volume.

References

* Work supported in part by the Department of Energy, contracts DE-FC05-85ER25000 and DE-AS05-76ER03509.

1. V. Zacharov, "Invariant Coding and Parallelism in Data Processing", CERN Preprint, **DD/82/1**.

   R. H. Schindler, "Comments on Vectorization of Track Finding and Fitting Algorithms for High Energy Physics Applications", prepared for the HEPAP Subpanel on Computer Needs for the Next Decade, **DOE/ER-0234**, August 1985.

2. J.J Becker et al., *Nucl. Instr. and Methods* **A235** (1985) 502.

Figure Captions

1. Schematic view of the E-711 detector, vertical (non-bend) view.
2. Schematic representation of the generation of the **ISHIFT** array for one of the bend views. For track i, **ISHIFT(M,L)** = 1,1,1,1 for **M** = 1,2,3,4, and **L** will uniquely identify this set. For track j, **ISHIFT(M,L)** = 3,4,6,6 for the same values of **M** but different **L**.

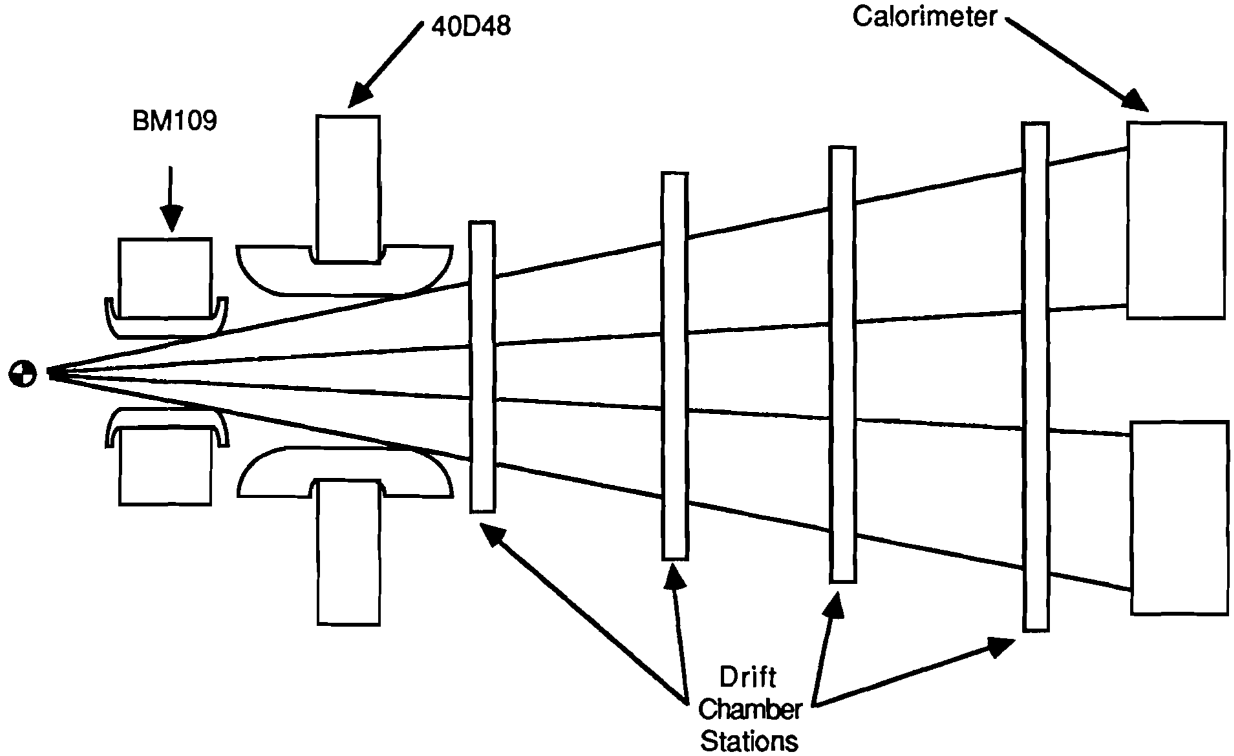

Figure 1

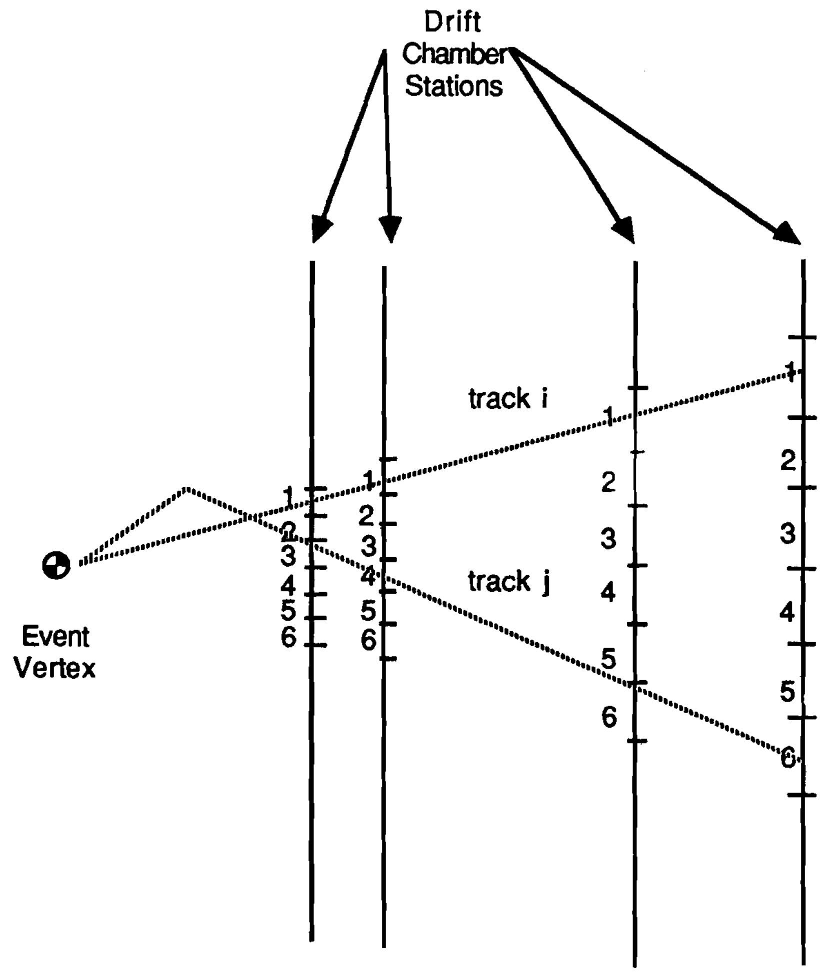

Figure 2