81-1-279

CERN-Data Handling Division

DD/80/11

H. Grote

P. Zanella

March 1980

APPLIED SOFTWARE FOR WIRE CHAMBERS

DD-jt

APPLIED SOFTWARE FOR WIRE CHAMBERS

H. Grote, P. Zanella
Data Handling Division, CERN
Geneva, Switzerland

Invited paper at the International Wire Chamber Conference,
Vienna, Austria (27-29 February, 1980)

ABSTRACT

During the past decade, large amounts of software have been produced for the processing of wire chamber data. This paper attempts to review the present situation and the current trends. Methods and algorithms for pattern recognition and track reconstruction are presented, together with examples of their application in experiments.

The data processing aspects are discussed, particularly in view of the recent developments in microprocessing techniques which suggest new ways of reducing and analysing experimental data.

1. INTRODUCTION

The last decade has seen a steady evolution of wire chamber detectors. Proportional and drift chambers are now widely used in experimental sciences to detect ionizing particles. Thanks to their good energy-, space- and time resolution, continuous sensitivity, and high detection efficiency, wire chambers dominate the field in high-energy physics, where they appear in some form or other in most experiments.

These devices produce accurate but basic information which needs to undergo a complicated processing sequence before it can represent something meaningful. Hence the evolution of the hardware has been accompanied by a parallel and equally intensive effort on the associated software. The number of people involved in the design and production of software has been steadily growing; in large contemporary experiments the software makers are almost as numerous as their hardware counterparts.

The extremely high repetition rates of wire chambers make it possible to collect enormous amounts of data. Handling such high volumes of data is a serious problem. However, the impact of new microcircuit technology is beginning to be felt and many believe that appropriate configurations of microprocessors will provide alternative solutions to the data processing problem.

Under the pressure of thousands of reels of magnetic tape the data handling people try to devise fast and efficient algorithms. In turn, analysis and computer related considerations tend to influence the overall design of experiments and to determine the detector's layout. The lessons of the early 1970's have been learned and the tendency today is to simplify the analysis by providing more precise, more reliable and less ambiguous information about the particle trajectories and by limiting the complexity of track topologies, e.g. reducing the non-uniformity of magnetic fields, or placing some detectors outside the magnets. The past few years have been successful in that nowadays one can start analysing the experimental data more or less straight away, provided the apparatus and the analysis have been carefully designed.

While the struggle for fast and efficient software continues, the new physics and the higher energies of new accelerators and colliders contribute to shift the interest towards events of increasing complexity - jets are a new challenge for pattern recognition experts. At PETRA energies, for example, the detailed

.../

study of jets calls for tracking and identification of some 15 particles, on average. At LEP energies the average charged particle multiplicity should go up to ~30 while the jet width will shrink considerably, thus increasing substantially (by some factor of five over PETRA) the density of tracks per unit solid angle. Even higher track multiplicities (40-50) are to be expected at the $p\bar{p}$ collider, under construction at CERN. As the space gets more and more crowded with particles, techniques for tracking in projections become less and less adequate if high recognition efficiencies have to be maintained. Three dimensional information as provided by some detectors today and promised for tomorrow maybe of great help in disentangling the individual trajectories in jets and in any similarly complex situation, provided one can rely on a reasonably accurate measurement of the third coordinate.

The purpose of this paper is to review the present state of the software for wire chambers, concentrating on the most relevant recent developments rather than describing in detail well established techniques.

2. DATA PROCESSING ASPECTS

The analysis of experimental data in high-energy physics is a non-trivial problem because of the complexity of the computations involved and of the volume of the raw data recorded. To date, the events of interest may involve as many as fifty particles. The number of digitizings per track is now in the range 10-100 and continues to increase. The amount of information recorded per event, all detectors included, is of the order of 10^5 bits.

Pre-trigger event rates up to several MHz can be dramatically reduced by applying a variety of triggering processes in successive stages. The final recording rates being typically 10-20 Hz, the performance of standard tape units available on data acquisition mini-computers is well matched to the recording requirements of most experiments. (Rates of the order of a Mbit/sec are quite typical and can be achieved with standard 75 ips, 6250 bpi tape units.)

The strong motivation to reduce event rates comes rather from the high demand on computing resources made by reconstruction and analysis programs. The computations required to reconstruct an event from its digital image vary considerably depending on the topology of the tracks and on the quantity and quality of the measurements. On the fastest computers (e.g. CDC 7600) it may take of the order of one second CP time to construct fully an event of moderate complexity. This CP time range extends over two orders of magnitude (~50 msec to 5 sec.).

.../

The number of events recorded during an experiment tends to be of the order of 10^7 . Thus the data relative to one experiment amount to some 10^{12} bits of information, and the analysis of such a volume of data can easily saturate a big computer for quite some time. Experimentalists are, however, accustomed to make do with whatever computer time they can collect among the collaborating institutions. In the case of large spectrometers, one or two thousand hours of CDC 7600 CP time can be regarded as a minimal yearly allocation to extract a reasonable amount of physics from the data.

Only a minority of the recorded triggers ends up on the Data Summary Tapes, the remainder being background, accidental triggers or events of little interest. This is why the whole recognition and reconstruction process is normally organized as a multistage decision sequence aimed at rejecting the bad events as early as possible and performing the heaviest computations only at the end on the good events. Rejections may occur during the on-line filter-in phase before recording on tape, or off-line, afterwards.

On-line the timing of the computations at the various triggering levels is critical since the storage medium at each level has to be cleared as soon as possible in order to minimise the detector's dead time. The time available to take decisions increases at each level (tens of nsec. at the first level, tens of msec. at the last) so one can apply trigger criteria of growing refinement.¹⁾

The impact of the new micro-electronics technology is beginning to be felt in the triggering and filtering area. Microprocessors are bound to play an important role in the on-line selection of events and are gradually replacing the faster but inflexible hardwired logic.

Complete micro-computers with eight and 16-bit memories are being produced in large quantities by the leading "chip" manufacturers. However, the relatively slow cycle-time (~ 1 μ sec) is a limitation to their generalised utilization as trigger processors. The very low cost of such devices suggests their operation in the form of multi-micro clusters.

Much more powerful microprocessors can, however, be built in the laboratories from commercially available micro-components, e.g. bit slices. Several devices of this type have been recently used on-line to experiments, e.g. ESOP²⁾, CAB³⁾, MONIKA⁴⁾. They are characterized by short cycle-times

.../

(~100 nsec) and small memories for the data (a few Kbytes) and the micro-code. They are normally microprogrammed and can execute simple algorithms of the type required at intermediate trigger processing levels extremely fast (hundreds of μ secs.).

Substantial progress has been made in providing tools to assist in the development of software for microcomputers. The early difficulties with coding, debugging and modifying the programs have been overcome by providing network access to sophisticated software implementation facilities made available on large computer systems, e.g. cross-assemblers, compilers, translators, simulators, linkers, loaders.

Because of their rejection power, software filters are a critical component of the experiment. The trend away from hardwired processors and towards more flexible devices is reappearing in the design of micro-computers as a generalised trend towards easier programmability. While high speed of execution remains a fundamental aim, current developments tend to emphasize ease of software development. The "emulator" concept, which is raising at present considerable interest among physicists, is an important manifestation of the need to simplify the production and assessments of software. The idea is to build microprocessors emulating well-known computers such as the IBM 370/168, DEC PDP-11 or CDC 6600. These micro-computers should be able to execute the instruction codes of the emulated machine and produce identical results. To the users it should make no difference whether the program has been run on the emulator or on the original computer. The interest of such emulators rests on this strict compatibility with widely used systems whose facilities can be exploited during the development and checking out of the algorithms.

The difference is in the cost-performance ratio when running the working programs for production. In fact, during this phase only a relatively small part of the highly complex hardware and system software of the big computers is utilised. Starting from this premise P. Kunz⁵⁾ has designed and constructed at SLAC the 168/E which contains a minimum set of hardware and software components capable of running codes developed and compiled on the IBM 370/168. The list of expensive components unnecessary for a specific use of the system may indeed be quite long, including the console, power distribution units, some of the instruction codes and error detecting and correcting hardware. Also, one can go back to mono-programming so that all the multi-programming

.../

features and a lot of memory can be dispensed with. After all, idle time should not be an obsession, if such devices can be produced at very low cost. Low cost can indeed be achieved by a "do-it-yourself" approach to system integration. In fact, while the price-performance ratio of basic components (chips) improves by a factor of two every year, fully integrated systems tend to reach the same factor over a four to five year period. This is due to the cost of the manpower needed to integrate components of different technologies into a complex final system, both hardware and software.

Kunz has demonstrated that the LASS reconstruction codes run on his 168/E at a speed about half that of the IBM 370/168, obtaining the same results at a much lower cost.⁶⁾ One problem is that the application software must be made to fit these basic versions of familiar computers. Thus, in an era of plenty (of memory, facilities and conveniences) programmers are asked to rediscover the joys of tiny memories and of overlays. It might be wiser to retreat slightly from this extreme position and make some concessions towards the modern style of programming, retaining the essential economic attractiveness of the proposal. Other laboratories are following up this idea; besides some variations on the 168/E theme, suffice it to mention the developments at CERN of a DEC PDP-11 emulator (MICE)⁷⁾ and at Brookhaven of a CDC 6600 emulator (PUMA)⁸⁾; the first built with ECL technology and equipped with powerful microprogramming capabilities is meant for on-line decision-making processes, while the second is intended for off-line applications. Present scores in the emulator mode: three times the PDP 11/70 for MICE and 0.6 times the CDC 6600 speed for PUMA.

Whether this approach makes sense at all depends not only on the relative availability of qualified technical staff as opposed to financial resources, but also on the ease of access to sizeable computing capacity, or on the need to go beyond currently available performance for on-line filtering applications, for example.

Several questions remain to be answered, such as the size of the real savings and how long it will be before the market-place catches up. Experience will teach us the real cost and the effective performance, the inconvenience of use and the problems with operation, utilization and maintenance. Micro-processor emulators have a fair chance of succeeding in high-energy experiments, owing to their low incremental cost, and their potential performance being well matched to present demand both on- and off-line.

Time and effort are still required to bring these devices from the prototype stage to their practical, large scale exploitation. Whatever the success of these attempts, essential know-how is being acquired in the laboratories that may result in very important long-term benefits to the high-energy physics community.

3. APPLIED SOFTWARE

The list of programming tasks in a high-energy experiment is long and keeps growing. From the simulation of the experiment itself to the analysis of the data, an impressive amount of code has to be implemented. The types of program vary considerably whether one is concerned with fast filtering algorithms or graphical presentation of events, sample event analysis or histogramming of some statistical variables, calibration or kinematical fits.

The widespread use of networks and the availability of high capacity mini-computers has made the distinction between on-line and off-line execution far less sharp than it used to be. However, the analysis of experimental data is still largely performed off-line on powerful computer systems.

In recent years, one has noticed the first attempts to build codes of general use in wire chamber experiments. Notable examples of self-contained packages are GEANT⁹⁾, HBOOK¹⁰⁾, LINTRA¹¹⁾ and MUDIFI¹²⁾. They perform simulation, histogramming or determination of functions and coefficients in parametrization procedures, respectively. In general one has made available a number of building-blocks of moderate generality, which are very useful in setting up any new analysis chain. However, the amount of work which has to be done to adapt, integrate, complement and put to work an efficient and reliable analysis chain is quite considerable. The importance of having the essential expertise available while designing and carrying out the experiments should not be underestimated.

How can one keep under control the growing amount of code to be produced for each experiment? The answer has first to be found at the level of the experiment by applying some sort of control on software development, imposing internal standards on the collaboration, encouraging professional software design, such as transportability, consistent updating, use of data and code

.../

management rules, and dynamic memory techniques. Steps can be taken at laboratory and inter-laboratory levels to avoid as far as possible duplication of efforts, making use of the existing software. The exchange of programs would be enhanced by the adoption of common standards for all high-energy physics software. Recently a working group has been set up in the framework of ECFA* to study this problem.

Of all the applied software required for the analysis of experimental data, perhaps the most critical and most demanding in terms of computer resources is the one concerned with the recognition and reconstruction of particles' trajectories. For large spectrometer setups this task alone may require of the order of a Megabyte of code.

In the following sections we shall concentrate on this particular topic.

4. REVIEW OF TRACK RECOGNITION APPROACHES

A particle's trajectory in a magnetic field is determined when the values of five parameters are known. A typical choice of these parameters is: position (x,y) and fixed z near the target, direction cosines $(\frac{dx}{ds}, \frac{dy}{ds})$ and some function of the momentum (e.g. $\frac{1}{p}$) at this point.

To obtain values for these variables, computations have to be performed on the track measurements, which have been selected and associated into trajectories by the pattern recognition programs.

The problem is to calculate these quantities with the highest possible precision and resolution in the shortest possible time.

Several methods are available, all based on the use of the Lorentz equation of motion for a charged relativistic particle in a magnetic field:

$$\dot{\underline{p}} = e \underline{v} \wedge \underline{B} \equiv \frac{e}{m} \underline{p} \wedge \underline{B}$$

(electromagnetic units, m relativistic mass, p momentum, B magnetic induction, e elementary charge).

The methods currently used can be grouped into four broad lines of approach, namely:

- a) explicit track model
- b) variational methods
- c) spline fits
- d) multidimensional function parametrization.

*ECFA: European Committee for Future Accelerators

In order to squeeze as much resolution as possible out of the data using a minimum of computer time, the best strategy may be to mix carefully a number of such techniques.

In principle the geometry problem can be said to have been solved. In practice however the successful construction of an adequate production program may have to be achieved through sweat and tears. This is because besides the basic mathematical aspects one has to worry about a number of physics considerations and numerical problems such as multiple scattering, energy losses, field maps and the error matrix. More often than not, the data display some unexpected behaviour and it takes a considerable amount of time before all the problems can be correctly solved.

An expert today has at his disposal an impressive number of tools which enable him to resolve most of the problems that may be encountered.

No method is a priori clearly superior to the others: the task consists in finding a combination of algorithms and heuristic techniques which can satisfy the particular needs of the experiment, taking into account all the peculiarities and idiosyncracies of the detectors.

4a) Explicit Track Model

In some cases, one can find a simple analytic expression which represents accurately the track over a certain length, perhaps a circle, or a cubic polynomial.

The numerical integration of the equations of motion can then proceed quickly using one of the simple approximation techniques, like Simpson's rule. Of course, one needs a rough estimate of the five parameters to start with and a few iterations to converge to the desired accuracy. This method is very fast and flexible (not so sensitive as others to changes in the detector layout). It is however, not straightforward to find an acceptable track model. This method has been applied for example in the NA4 experiment at CERN¹³).

4b) Variational Method

This method is in some sense the inverse of the tracking with known parameters as used in event simulation. Here the intersections with the wire planes are known and one has to determine the parameters.

.../

The technique consists of tracking a particle through the known points; from a starting point and a good estimate at the other parameters the track is followed through the field, together with five "variation" tracks corresponding each to a small variation in one parameter. The best values of the parameters are obtained through iteration performing a least squares fit of the track to the actual measurements, and assuming the dependance on the parameters to be linear. The tracking is normally a step-by-step integration (e.g. Runge-Kutta)¹⁴⁾ but other approaches exist¹⁵⁾.

This method is probably the slowest because the magnetic field is explicitly invoked at each step, but has the big advantage of working always safely and accurately. Because of this it is used where no other methods work, and to test the accuracy and the reliability of faster approaches. Multiple scattering and energy loss calculations can be integrated naturally with this method.¹⁶⁾

Recent progress has been made to improve its speed by the work of Myrheim¹⁷⁾). It is a solid, stable approach which has been used in many complex experiments. However, for experiments with very high statistics people normally attempt to devise faster algorithms.

4c) Spline Fits

Introduced more recently into experimental physics by H. Wind¹⁸⁾, this method has been used in many experiments and is now well established.

A cubic spline curve is constructed which approximates the second derivative of the trajectory to be found. This implies a knowledge of the magnetic field at the measured points and a good estimate of the first derivative. Double integration using this cubic spline approximation corresponds to fitting a quintic spline curve to the original measurements. The double integral being linear in the five parameters, a linear fit to the data minimizing the χ^2 gives the values of the track's parameters, which are then improved through iteration.

.../

This method is fast and works well if the field is homogeneous. Normally, the magnetic field values have to be calculated only once for all iterations.

However, in case of field inhomogeneity, one has to be very careful since the method may give wrong results. It is particularly sensitive to the estimate of the first derivatives. In summary, this method requires a very detailed knowledge of track behaviour in every part of the detector; because of its sensitivity it has to be handled with care; if applicable and correctly set up it can execute rather fast. Among others, OMEGA¹⁹⁾ and SFM²⁰⁾ spectrometer analysis programs have used it at CERN.

4d) Multidimensional Function Parametrization

The idea is to find explicit functions relating track parameters directly to measured coordinates. A two-step approach was suggested by H. Wind²¹⁾. In the first step new coordinates are defined as linear combinations of the measurements. These new coordinates are the most significant for track representation and are obtained via Principal Components Analysis (Karhunen-Loeve dimensionality reduction)²²⁾. The coefficients of the linear transformation are found using a "training sample" of tracks. The general program LINTRA can be of help at this stage.

In the second step, the five track parameters are expressed as polynomial functions of the most significant coordinates. A program exists, MUDIFI, to assist with this multidimensional fit.

This method is the fastest, but the detailed implementation is not trivial. In particular the choice of the training sample and the sub-division of the feature space are particularly delicate. The development time is long and the whole process has to be repeated in case of changes to the detectors.

This approach has been used successfully in several experiments of moderate complexity. It can be used to obtain estimates of the track parameters to be used as input to one of the iterative procedures described in the preceding sections.

5. TRACK FINDING

Whereas the fitting of the track parameters as described in the previous section is a well understood procedure, this is much less so for the task of track finding. Already the definition of the task is difficult. If we first try:

Definition 1: split the set of all position measurements in a detector for one trigger into sub-sets such that:

- each sub-set contains measurements originating from one and the same particle;
- one sub-set contains all hits not caused by particles,

we have not achieved very much, since it is normally impossible to prove that two signals were really caused by the same particle, and even less so, can one show that a signal was not caused by any particle. Therefore, what is actually used as a workable definition is:

Definition 2: split the set of all position measurements in a detector for one trigger into sub-sets such that:

- each sub-set contains measurements which could be caused by the same particle;
- one sub-set contains all hits that cannot be associated with particles with sufficient certainty.

This requires basically a two step procedure. In the first step, a sub-set of measurements is selected, forming a "track candidate". In the second step, a check is made as to whether this sub-set is a possible track or not. This procedure is often split further in order to make it faster, namely: select a certain number of measurements (initialisation) - check and reject or continue - add a certain number of measurements - check and reject or continue - ... - through to the final check, performed on the fully assembled track. This step-wise procedure can be varied in different ways, but the principle remains. Its important feature lies in the fact that after each check the wrong candidates are rejected, and in this way practically only the good candidates undergo the final check. Accordingly, the check algorithms grow from simple and fast ones at the beginning to powerful and slower ones at the end; there is a simultaneous increase in the degree of confidence that the track candidate is really a good track.

.../

For completeness, it has to be mentioned that to the simple picture of definition 2 above, several variations are possible. In some cases, only track segments are found in the way described, the segments then being joined in a similar procedure afterwards. In other cases, the position measurements are pre-assembled into "space points" or their homologues, this being a pattern recognition task in itself. Additional information may be taken into account and thus provide further constraints, such as direction measurements, ionisation levels, and other information provided by the wire chamber detector. Furthermore, the procedure need not be self-contained as defined above, but may use other information provided by devices such as hodoscopes and calorimeters, or may use the known position of the interaction region, or simply extract constraints from the physics one wants to study.

In the following we attempt to list and classify the methods which have been, or are actually being, used for track finding in counter physics experiments, without claiming the list to be exhaustive. In brackets, we add names of experiments or experimental facilities where the corresponding method has been applied. A few figures are intended to give an idea of the problem in some cases.

5a) Methods to define Track-candidates.

A. Combinatorial methods

A sub-set of chambers in the detector is selected from the full detector, and one point from each chamber chosen by a combinatorial procedure which makes sure that not point is forgotten. This may be done to:

- 1) select the full track candidate: this is obviously only applicable in cases of low multiplicity and a small number of chambers (example: WA18, CERN²³);
- 2) select the initial track candidate: this is done in most applications.

B. Track following methods

They are typically applied to tracks of the "perceptual" type, where the track can be more or less easily recognized by the human eye from the displayed coordinates. After having found an initial track segment, normally as far away from the interaction region as possible, the point in the next innermost chamber is found by extrapolation, or sometimes by interpolation when the interaction regions is sufficiently well known to be able to serve as an extra point, which is to be preferred since

.../

it is more precise. One uses extrapolation of "zero" order by simply choosing the "nearest neighbour" (JADE, DESY²⁴), see fig. 1), first order (certainly in all cases where the track is straight), second order (OMEGA, CERN²⁵), and possibly higher orders, or other curves like circles or ellipses (OMEGA, CERN). Clearly the simplest possible method will be chosen in any specific case, since it is the fastest. Here, the so-called "parametrized tracking" has to be mentioned in particular. In this case, the position of the next point is evaluated in functions of the points already found, and the parameters are found from a "track sample". This allows a very fast and precise prediction even over long distances (OMEGA Prime, CERN²⁶), see fig. 2).

C. Track roads

In this case, a "road" is defined over the full length of the track, usually by using initial points at both ends, and one in the centre in the case of curved tracks. To these, the approximate vertex position may be added. A simple, not necessarily the correct, track model is then used, be it a straight line, a parabola or a spline curve (SFM, CERN²⁷) to define a road, inside which additional points are looked for. The width of the road may be known from theoretical considerations (detector resolution), but more often from a track sample. This width may vary as a function of the curvature, the number of points, or other parameters.

D. Global methods

In this case, all points (at least in principle) are considered together, and all tracks are found in one and the same procedure:

- 1) Histogramming method: n functions of the point coordinates are entered in an n -dimensional histogram, and the entries belonging to each track then form a "cluster" or "peak", well separated from those of the other tracks. The next task is then to find these "peaks". The method works in practice only in the one- or two-dimensional case. A simple example may illustrate this method: assume a star-like event with straight tracks, vertex at zero, and enter y/x into a histogram for each point.
- 2) Minimum Spanning Tree: this is a graph-theoretical approach²⁸⁾, in which a "distance" between all point pairs is used to find all tracks by linking points of nearest "distance". This "distance" may be almost any function of the point pair coordinates and allows for a considerable variation of the method (TASSO, DESY²⁹), see fig. 3).

.../

5b) Methods to Check Track Candidates

These range from very simple to very complex mathematical algorithms, and we can only give a few typical examples.

- A. Local checks such as the local curvature, the sign of curvature, or other constraints. This is typically applied when short track segments ("chain elements") have to be checked (JADE, DESY).
- B. Check on the direction, or the nearest distance of the extrapolated track from the vertex (SFM, CERN).
- C. Fit to a simple analytical model, such as straight line, parabola, cubic curve or spline, which approximates the correct track curve. The confidence limits for this test are normally found from a track sample.
- D. Feature selection: this makes use of the fact that for n space points of a track, there exist $2n-5$ constraints, which define the "feature space" of the track. These constraints can be found from a "training sample" of tracks in the way outlined before. If one assumes a linear sub-space (which is normally the case), the constraint equations represent the $2n-5$ vectors which are orthogonal to this sub-space.
- E. Least squares fit to the correct track model, either analytically or numerically defined. This ultimate check is performed in most cases in the geometry fit program. In general, the fit defining the track parameters will in most cases as well provide a good test of the track hypothesis.

Limitations of space do not permit us to expand upon any of the methods above. It is, however, the detail which matters in this type of application, since many methods may a priori appear to be equally well suited for solving a particular problem, and the final choice of a method will depend on careful studies, in the course of which normally much code is developed, but little remains. Furthermore, it is safe to assume that no existing program follows exactly one of the methods described. Several of them are normally combined, and all of them modified in order to make best use of the specific detector and event type in question.

.../

In conclusion, it can be said that in contrast to the track parameter fitting algorithms, the track finding software is in most cases very much tailored to a certain experiment or experimental facility, and although it may make use of some existing packages, it will in almost all cases not be portable to another experiment. It is therefore in this field that much programming effort is invested continuously, effort which can only be minimized by designing the detector and hardware very carefully.

REFERENCES

1. C. Verkerk: On-line Filtering. Proceedings of the 1978 CERN School of Computing, Jadwisin, Poland, 28 May-10 June 1978. CERN (Geneva) report 78-13.
2. T. Lingjaerde: A Fast Microprogrammable Processor. CERN (Geneva) report DD/75/17.
3. E. Barrelet, R. Marbot, P. Matricon: CAMAC Booster ("CAB" system), a Versatile Microcomputer for High Rate CAMAC Data Acquisition. Ecole Polytechnique, Palaiseau (France) report LPNHE/X/79.
4. P. Schildt: Entwicklung eines mikroprogrammierbaren Rechners zur on-line Rekonstruktion von Spuren für den Detektor TASSO am Speicherring PETRA. Diplomarbeit im Fachbereich Physik, Universität Hamburg, 1979.
5. P.F. Kunz: The LASS Hardware Processor. Nucl. Inst. and Meth. 135 (1976) p. 435.
6. P. Kunz, R. Fall, M. Gravina, J. Halperin, L. Levinson, G. Oxoby, Q. Trang: Experience using the 168/E Microprocessor for Off-line Data Analysis. SLAC-PUB-2418, Oct. 1979.
7. C. Halatsis, J. Joosten, M.F. Letheren, A. van Dam: Architectural Considerations for a Microprogrammable Emulating Engine using Bit-slices. CERN (Geneva) report DD/79/7.
8. J. Niederer: A 6600 Microcomputing System; the NYU-PUMA Project, with a BNL Perspective. Brookhaven National Laboratory internal report, 1979.
9. R. Brun, R. Hagelberg, M. Hansroul, J.C. Lassalle: Simulation Program for Particle Physics Experiments GEANT. CERN (Geneva) report DD/78/2.
10. R. Brun, I. Ivanchenko, P. Palazzi: HBOOK - a Histogramming, Fitting and Data Presentation Package. CERN (Geneva) report DD/77/9.
11. R. Brun, M. Hansroul, J. Kubler: LINTRA - a Principal Components Analysis Program. CERN (Geneva) report DD/US/70 (1980).
12. R. Brun, M. Hansroul, J. Kubler, H. Wind: MUDIFI - a Multi-dimensional Fit Program. CERN (Geneva) report DD/75/23.

13. R. Brun: PATRAC - the NA4 Analysis Program. CERN (Geneva) report DD/EE/79-3.
14. M.G. Shiel: A Comparison of Routines used to Follow Particle Trajectories through Magnetic Fields. CERN (Geneva) report DD/75/22.
15. M. Metcalf, M. Regler: Solution of the Equations of Motion of a Charged Particle in a Stationary Magnetic Field. J. Comp. Phys. 11 (1973), pp. 240-249.
16. M. Regler: Vielfachstreuung in der Ausgleichsrechnung. Acta Physica Austriaca 49 (1978) pp. 37-45.
17. J. Myrheim: A Fast Runge-Kutta Method for Fitting Tracks in a Magnetic Field. Nucl. Inst. and Meth. 160 (1979), pp. 43-48.
18. H. Wind: Momentum Analysis by using a Quintic Spline Model for the Track. Nucl. Inst. and Meth. 115 (1974) pp. 431-434.
19. J.C. Lassalle, E. Riipinen: Implementation of the Quintic Spline Model for Momentum Determination using Non-orthogonal Projections. CERN (Geneva) report DD/76/14.
20. A. Fröhlich: Implementation of the Quintic Spline Model in the SFM Geometry Program. CERN (Geneva) report DD/75/21.
21. H. Wind: Function Parametrization. Proc. 1972 CERN Computing and Data Processing School (CERN 72-21) p. 53.
22. K. Fukunaga: Introduction to Statistical Pattern Recognition. Academic Press, New York and London 1972, p. 226.
23. K.H. Mess, M. Metcalf, R.S. Orr, B. Kröger, A. Rosanov: Track Finding in a Fine-grained Calorimeter. CERN (Geneva) report DD/80/7.

.../

24. M.C. Goddard, T. Nozaki, J. Olsson, G.F. Pearce, P. Steffen: A Pattern Recognition Program for JADE's Jet Chambers. These proceedings.
25. H. Grote, M. Hansroul, J.C. Lassalle, P. Zanella: Identification of Digitized Particle Trajectories. Proceedings of International Computing Symposium, Davos 1973, North-Holland Publ. Co., 1974, p. 413.
26. J.C. Lassalle, F. Carena, S. Pensotti: TRIDENT: A Track and Vertex Identification Program for the OMEGA Particle Detector System. These proceedings, or CERN (Geneva) report DD/80/8.
27. A. Fröhlich, H. Grote, C. Onions, F. Ranjard: MARC - Track Finding in the Split Field Magnet Facility. CERN (Geneva) report DD/76/5.
28. C.T. Zahn: Using the Minimum Spanning Tree to Recognize Dotted and Dashed Curves. CERN (Geneva) report DD/73/24, or Proceedings International Computing Symposium, Davos 1973, North-Holland Publ. Co., 1974, p. 381.
29. H. Boerner, H.M. Fischer, R. Rohrman, H. Hartmann, U. Kötz, H. Kowalski, B. Löhr, P. Schmüser, B. Wiik, M. Wollstadt: The Large Cylindrical Drift Chamber of TASSO. These proceedings.

Fig.1 - JADE event

Fig. 2 - OMEGA' event reconstructed by TRIDENT

Fig. 3 - TASSO event