Rep. Prog. Phys. 50 (1987) 473-500. Printed in the UK

Pattern recognition in high-energy physics

H Grote

CERN, Geneve 23, CH-1211, Switzerland

Abstract

Pattern recognition is of crucial importance to many high-energy physics experiments during their analysis phase. This review gives a short introduction to those aspects of experiments that require the application of pattern recognition methods, which are discussed in detail and illustrated with examples from high-energy physics experiments of the last 15 years. At the end, a number of papers are recommended dealing with track and vertex fitting.

This review was received in its present form in June 1986.

Contents

1. Introduction

1.1. The scope of pattern recognition in high-energy physics

Since the days of Wilson's cloud chamber, tracks of sub-atomic particles have had to be recognised and measured. Initially, the recording medium was film, and film is still in use today for 'optical' detectors, such as bubble chambers, optical spark chambers and optical steamer chambers, where charged particles leave visible tracks behind which can be photographed.

Starting in about 1960, computers, which have the useful ability of being able to perform boring and tedious calculations rapidly, accurately and without complaint, were used to assist in the recognition and measurement processes. This use of computers implied that the input data had to be in digital form and there were three approaches.

(i) Photographs were digitised, that is to say, the picture was divided into very many tiny areas and the grey level of each area was measured.

(ii) Rather than recording the optical image on film it was captured by a television camera and the resulting analogue signal (an electric voltage) was converted into digital form.

(iii) So-called 'electronic' detectors were developed, which rely on physical effects other than light emission to detect the passage of a particle, and which deliver either digital information, such as the number of a wire, directly, or which produce analogue signals which are then converted to digital form.

If required, the digital representation of an event can be converted into a picture on a graphics terminal (figure (1*a*)). Nowadays, the electronic detectors have almost completely replaced bubble chambers and other optical detectors, mainly because they allow the selection of specific event types to be observed and because they can be made sensitive for very short periods of time, giving them the ability to record single events even in the presence of very many interactions.

This review is based on a lecture given at a CERN Summer School for Computing (Grote 1981) and consists of three parts. In the first, the field is introduced to the reader who is not too familiar with high-energy physics. The second part, after having recalled the basic principles of pattern recognition and some methods which are used in high-energy physics, deals with the different methods of track finding and some aspects of pattern recognition for event classification, giving examples from real experiments wherever they exist. The emphasis lies on the automatic pattern recognition for electronic detectors, although many of the methods described apply to bubble chambers and other optical devices as well.

The third part is very short and contains mainly references to relevant papers on single track and vertex fitting, a field that has already been dealt with in previous review papers (Eichinger and Regler 1981) and that is normally well separated from pattern recognition.

Nowadays, automatic processing of the bulk of the data taken in a high-energy physics experiment is the rule, and the software to perform this task has become an integral part of the experiment. Within this software, the track finding is certainly of crucial importance for most experiments, to the extent that modern detectors have to be designed to take into account the track finding methods that will be employed.

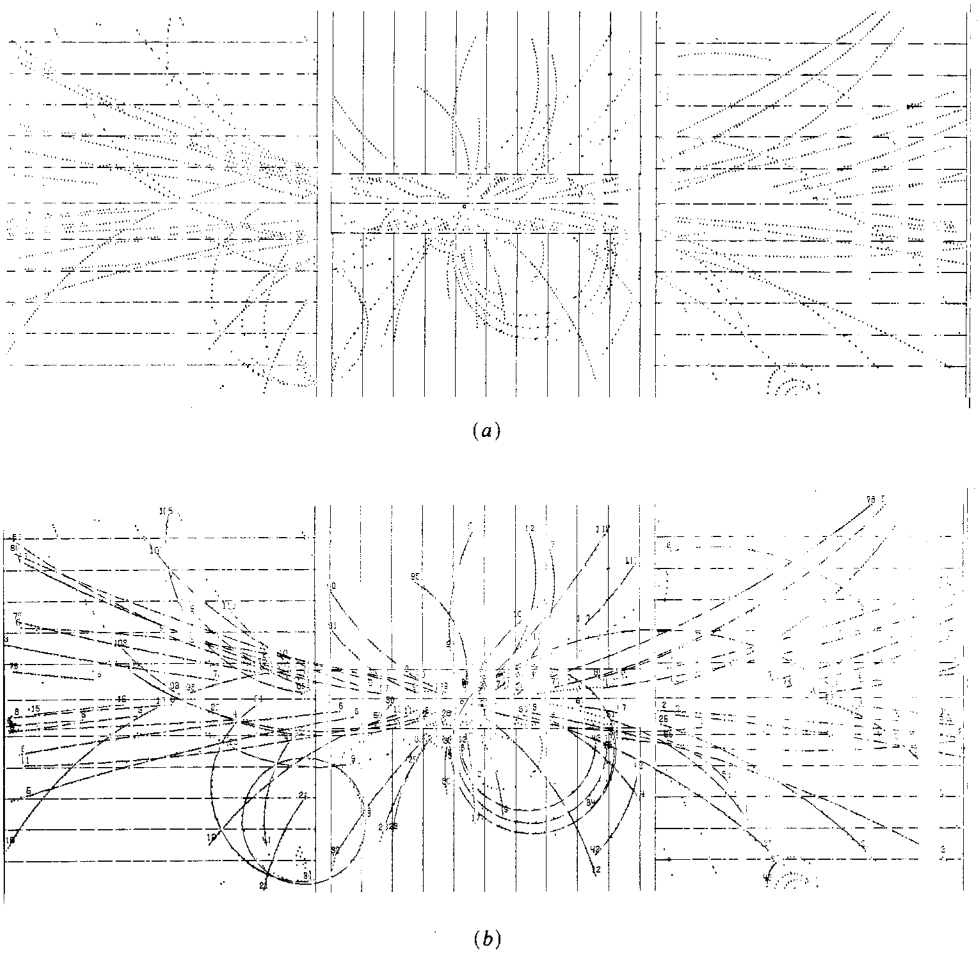

Figure 1. (*a*) Typical event recorded in the central detector of experiment UA1 at the CERN SPS collider; only every second signal plotted. (*b*) The same UA1 event as before, with reconstructed tracks.

Readers wishing a more general introduction into the usage of computers in high-energy physics are referred to a comprehensive paper by Metcalf (1986).

1.2. The physics

As the name implies, high-energy physics deals with elementary particles (or light nuclei) at very high kinetic energy. Since the speed of light gives an absolute upper limit to their velocity, particles can acquire more kinetic energy only by becoming more massive. Accordingly, 'high-energy' here means that the energy in a two-particle collision is comparable to or high than the mass of a typical particle, e.g., the proton with its rest mass of 0.938 GeV/c^2.

Electrically charged particles such as protons or electrons can acquire this high energy in accelerators, where they are accelerated by a suitable electromagnetic field. The highest man-made energies (1 TeV = 10^{12} eV) available today are achieved with circular accelerators, where the particles do many turns before they reach the energy required.

Without giving more details on the types and energies of acceleratores, one important point must, however, be stressed here. There are two distinct ways in which particle collisions can take place. In the first case, one particle (called the 'beam' particle) is accelerated, and the other one (called the 'target' particle) remains at rest. In the collision, only the energy in the centre of mass, the 'invariant mass'

$$E_{\mathrm{cm}} = (m_a^2 + m_b^2 + 2E_a m_b)^{1/2}$$

(where a = beam particle, b = target particle) is available for physics processes such as the creation of new particles, since the rest of the energy of the beam particle is needed to accelerate the interaction products in order to conserve momentum. As one can see, the centre of mass energy increases with the square root of the beam energy, and amounts to 13.7 GeV when a proton with a momentum of 100 GeV/c hits a proton at rest. In addition, because of momentum conservation, most of the products of the interaction are confined to a relatively narrow angular region around the beam direction, where the detector has to be placed.

In a colliding beam machine, where particles and their antiparticles circulate with the same energy in opposite directions and collide inside small intersection regions where the two beams cross, the collision takes place in the centre of mass frame if the small crossing angle is neglected. In this case, the centre of mass energy is just twice the individual beam energy and therefore increases linearly with it. Accordingly, in the case of a proton and an antiproton beam of 100 GeV energy each, the centre of mass energy theoretically available for physics processes is 200 GeV.

One important aspect of high-energy physics research is the confirmation of theories about particle interactions, and another is the exploration of ever higher energy ranges in the search for completely new phenomena. Since modern theories tend to predict massive particles, such as the recently detected intermediate vector bosons Z_0 (mass = 93 GeV/c^2) and W (mass = 83 GeV/c^2), both these aspects call for ever higher energy ranges. This is why colliders have become very important tools in high-energy physics research, and several of them are being exploited or under construction world-wide.

As far as pattern recognition is concerned, collider experiments differ in two aspects from fixed target ones: firstly, since the interaction takes place in the centre of mass system, particles can fly away in any direction and the detector has to surround the interaction region if all particles are to be seen. Secondly, the number of particles produced in the interaction (the multiplicity) increases with higher energy, and new phenomena such as 'jets' (several particles within a very narrow space angle) appear. Current collider experiments record events with tracks from over 100 particles, and sometimes the physics analysis requires that each individual track must be reconstructed accurately (figure (1*b*)).

1.3. The data

Four types of data are normally needed to perform the pattern recognition in an experiment.

The first type are those which define the exact position of the different active and passive elements of a detector, normally called 'survey' data. They are obtained by geometers measuring the positions of detector parts of known dimensions. For active parts, which detect particles, these measurements may have to be refined later by using measured particle trajectories.

The second category of data is formed by the various constants needed to convert the 'raw' data delivered by the detecting elements into physics units, taking into account possible non-linear behaviour of a device, and are called 'calibration constants'. As an example, consider the mainly ultraviolet light coming from a scintillator, the amount of which is in some way proportional to the energy deposited there. This light will be 'wavelength shifted' to the visible range and will be transported via light guides to a photomultiplier, which will eventually produce a measurable current. It is clear that the normal 'end-user' physicist wants to know the energy deposited in the scintillator, and not the photomultiplier current.

The third category describes the static magnetic field which is very often used in connection with particle detectors since it allows the momentum of a charged particle to be measured by means of the curvature of its trajectory. The magnetic field is normally given either in tabular form on a grid, from which the value at a given point can be derived through linear or quadratic interpolation, or in the form of Laplace polynomials which have the property of satisfying Maxwell's equation (Metcalf and Regler 1973). In the most desirable case (from the viewpoint of the pattern recognition) the magnetic field is constant or almost constant and can therefore be represented in a very simple functional form.

Whereas the three types of data described so far are normally obtained before the experiment starts, the fourth type, the 'raw data' of an event, form the basis for the measurement of an interaction between particles at very high energy.

The energy which is set free in the centre of mass system of two colliding particles can only be 'dissipated' through the creation of elementary particles, including high-energy photons. Consequently, in order to describe fully such an interaction, it is necessary and sufficient to identify all of the created ('secondary') particles and list their kinematic properties. Of course, some of the outgoing particles may already be decay products of secondary particles which lived for too short a time to be detected directly.

1.4. Detectors

Detector systems are normally designed to provide three independent pieces of information about a particle. The first is a measurement of the energy of the particle at a well defined point. The second is the identification of the nature of the particle, e.g., K_0, μ^- or e^+. The third is the determination either of the coordinates of many points on the track or of the coordinate of one point on the track, together with the direction of the track at that point. By using these three elements it is possible to calculate the momentum of the particle at any point on its trajectory and, in particular, at the point of interaction, the 'vertex'.

If the tracks pass through a suitable static magnetic field then information about the energy (or, more precisely, the momentum) can be extracted from the shape of the particle trajectory. Alternatively, the energy of charged particles, and neutral ones as well, can be measured by letting the particle lose all of its energy in a solid block of matter and measuring the energy dissipated. This type of apparatus is called a 'calorimeter' (Fabjan and Ludlam 1982).

The great majority of signals for track finding in high-energy physics today are delivered by wire chambers, which have shown a considerable evolution over the last twenty years. In this review, there is space to outline only the most basic mechanisms and the reader interested in more details is referred to the paper by Charpak (1978).

The basic unit of a wire chamber is a thin metallic wire of typically 50 μm diameter. In its simplest form, a detector consists of one such anode wire surrounded by a cylindrical cathode, with a voltage of several thousand volts between them. The chamber is normally filled with a noble gas such as argon, with the addition of a gas such as isobutane at atmospheric pressure.

When ionising radiation passes through this cylinder, the light electrons liberated will drift towards the anode wire, whereas the much heavier positive ions will stay behind, moving comparatively very slowly towards the cathode. Since the electric field grows inversely with $1/r$, where r is the distance from the wire centre, the electrons will gain more and more energy over their mean free path between inelastic collisions, until they are able to initiate an electron avalanche through further ionisation, at a distance of the order of the wire diameter away from the wire centre. The negative charge thus collected on the anode wire leads to a short negative pulse that is detected by the electronics connected to the wire, which is said to have 'fired'.

1.4.1. Proportional chambers This mechanism remains practically unchanged when many parallel anode wires are arranged in one plane between two cathode planes. The electric field (figure 2) is then almost homogeneous everywhere except for the region near to the wires, where it resembles that in the tube. This arrangement therefore works like many independent tubes: an electron cloud created somewhere in this 'chamber' will drift towards a specific wire and lead to a pulse there. A particle passing through this chamber and creating regions of ionisation along its path will lead to signals on one or several adjacent wires, depending on its orientation with respect to the chamber plane.

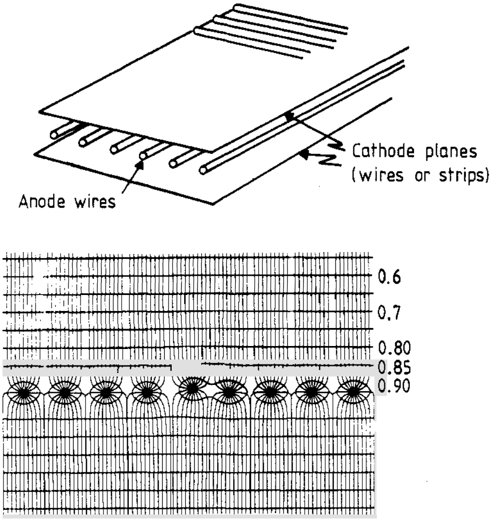

Figure 2. A multiwire proportional chamber is a grid of uniformly spaced thin anode wires, sandwiched between two cathode planes. The two main regions of the electric field are a region of roughly constant field and a region of rapidly increasing field around the wires where avalanche multiplication occurs.

By spacing the wires at a suitable distance *d*, one can in this way measure the position of the impact point on the plane of the wires with a precision of about $d/3$. Detectors of over 50 000 wires (Bouclier *et al* 1974) have been constructed with chambers operating in this 'proportional' mode. Proportional wire planes achieve a detection efficiency for charged particles of over 99%, the small inefficiencies arising from too low ionisation or failures in the electronics.

Since a plane of wires measures only one coordinate of the avalanche position, one has developed the following method to measure the space position: the cathode planes are divided into parallel strips of about 1 cm width which run perpendicularly to the wires, or at a suitable angle (figure 3). The avalanche induces a positive pulse on these strips, which can be detected and can deliver a second coordinate (the third coordinate of the space point is of course the wire plane position). Another method consists of measuring the amount of charge flowing into amplifiers at both ends of the wire. If the wire has a reasonable resistance, the ratio of the charges will depend on the position of the avalanche along the wire. With this 'charge division read-out', a precision of about 1% of the wire length can be reached.

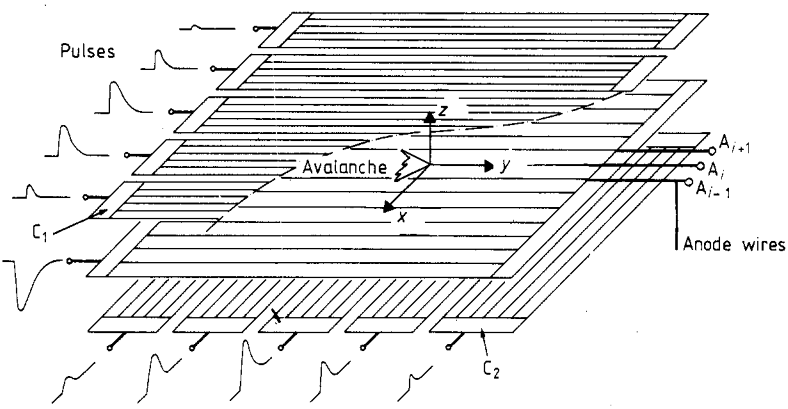

Figure 3. Localisation by centre of gravity of the induced pulses. The motion of ions leaving the vicinity of the anode wires in a multiwire proportional counter induces positive pulses on all surrounding electrodes.

1.4.2. Drift chambers To achieve good accuracy, large proportional detectors need many wires, each of which needs its own amplifier. In order to reduce this cost but also in order to improve the accuracy, detectors have been constructed which exploit the same arrangement of anode wires between cathode planes in a completely different mode. To this end, the chamber is rotated by 90 degrees, such that the particle path runs now parallel to the wire plane but still orthogonal to the wires. Since the electron cloud drifts initially at constant speed of some $20\,\mu\mathrm{m\,ns}^{-1}$ towards the anode wire, the time lapse between the impact time of the particle and the avalanche is a direct measure of the distance from the wire. The impact time is normally determined by separate scintillation counters. With a time resolution of the order of 1 ns, one can therefore reach position measurements of better than 50 μm precision. However, the apparent resolution can be considerably worse because of correlation effects (Drijard *et al* 1980). A drift space of many tens of centimetres can be obtained, which leads to a considerable reduction in the number of wires. However, since the chambers only measure the drift distance, there is no information as to which side of the wires the particle has passed. This 'left-right ambiguity' represents an additional difficulty for track finding. The remedies employed, e.g., 'staggering' the wires slightly and staggering drift chamber cells, cannot normally be used during the initial part of the track finding to reject the 'image' or 'ghost' tracks.

1.5. Event simulation

Before an experiment really starts taking data, the optimal detector lay-out has to be studied and then the reconstruction programs have to be written and tested (this work normally takes several years), so that as soon as the first real data become available, they can be analysed almost immediately and the results published as quickly as possible. During and after data taking, the detector acceptance has to be calculated for certain classes of tracks or events.

For all these purposes, events with all of their expected raw data have to be simulated as exactly as possible. With the help of a theoretical model predicting the type of events likely to be observed, 'particles' are created in the computer program and then followed through the detector, taking all the important effects of interaction with matter into account, in order to make the simulation as realistic as possible. The detector response is then simulated with great care and 'raw data' are produced as if they came from a real detector.

As a consequence of the stochastic nature of most simulated processes, variables such as the lifetime of a particle, its total scattering angle after having passed through a piece of matter or the number of ionised atoms in a drift chamber cell, have to be chosen at random from a given probability distribution. These random variables are chosen using Monte Carlo techniques (James 1983), which is the reason why the whole process is often called Monte Carlo simulation (Brun *et al* 1985).

1.6. Particle trajectories

The trajectory of a charged particle in a static magnetic field is given by the Lorentz equation

$$\mathrm{d}P/\mathrm{d}t = eV \times B = (e/mP) \times B \tag{1.1}$$

(where upper case letters indicate vectors, lower case letters indicate scalars, and '×' stands for cross product) with e = charge, V = speed, P = momentum, m = (relativistic) mass of the particle and B = magnetic induction at the position X of the particle.

Multiplying (1.1) by P yields

$$\mathrm{d}(P^2)/\mathrm{d}t = 0$$

since $P*(P \times B) = 0$ ('*' for scalar product), leading to

$$P^2 = \text{constant}$$

and

$$m = (m_0^2 + P^2)^{1/2} = \text{constant}.$$

With

$$V = \mathrm{d}X/\mathrm{d}t \qquad P = mV \qquad p = \|P\| \qquad v = \|V\|$$

and replacing time derivatives by space derivatives with respect to s, the curvilinear track length (for any function F, $\mathrm{d}F/\mathrm{d}t = v\mathrm{d}F/\mathrm{d}s$) equation (1.1) becomes

$$\mathrm{d}^2X/\mathrm{d}s^2 = e/p\mathrm{d}X/\mathrm{d}s \times B \tag{1.2}$$

which can be rewritten as

$$\mathrm{d}X/\mathrm{d}s = N \qquad \mathrm{d}N/\mathrm{d}s = aN \times B \tag{1.3}$$

where N is the unit vector tangent to the trajectory at point X and a is a constant.

For the specific units GeV/c, tesla and metre, and measuring the charge in multiples q of the elementary charge, a becomes

$$a = 0.2998q/p$$

where the numerical constant is the speed of light in vacuum.

This system of first-order differential equations has five integration constants, e.g., x_0, y_0, n_x, n_y at a given z_0, and a, or rather q/p.

For a homogeneous magnetic field (B constant in space), the solution of (1.3) is a helix winding around the direction of B, or a straight line precisely in the direction of B. For an arbitrary B, the solution of (1.3) can be written in closed form with the help of integrals (Eichinger and Regler 1981). In simulation programs, the tracking is normally done by stepwise integration of the Runge–Kutta type (Bugge and Myrheim 1981).

The six-dimensional space formed by the three space coordinates and the three momentum coordinates is called 'phase space'. Each particle trajectory can be represented by a point in phase space, according to equation (1.3).

For any primary interaction of particles the total phase-space region which may contain tracks of secondaries is known *a priori*, given by the target size or interaction region on one hand, and by the beam momentum and the detector geometry on the other hand.

2. Pattern recognition

2.1. Principles

Pattern recognition is a field of its own, with dedicated journals and many books and other publications written on the subject. Reviews have appeared in 1974 (Kanal) and 1980 (Fu), from both of which the wide range of applications becomes apparent, including character recognition, speech recognition, electrocardiogram analysis, chromosome classification and scene analysis, to name but the most important. However, reports of pattern recognition in high-energy physics are almost entirely missing from this specialised literature, the reason for which will possibly become clear later.

Of the many text books in the field, I give here six in their order of relevance for high-energy physicists: Andrews (1972), Young and Calvert (1974), Reingold *et al* (1977), Tou and Gonzalez (1974), Patrick (1972) and Ullmann (1973).

Pattern recognition in general can be defined as follows.

Given n measurements of an object (such as all the raw data of a high-energy physics event, the grey levels of mesh squares in a picture, the Fourier spectrum of an acoustic signal, etc), one wants to associate this specific object with one out of several *classes* of objects. If X is the vector consisting of all measured values, one is therefore looking for a *decision function d* such that

$$c = d(X)$$

is the class to which X belongs. In the general case, classes may share objects, but in high-energy physics we are mainly interested in distinct classes, which do not share objects.

The definitions used to separate the classes are sometimes known *a priori*, but they may well be unknown. For example, a British person reading an English text can be considered to have prior knowledge both of the classification of the letters, and of the words in the text. The same person attempting to read an Arabian text would normally have to work hard to perform both of these classifications, finding out where the letters start and end, and which words have the same root. Clearly, the task of classifying *a priori* unknown classes is much more difficult than dealing with known classes.

All possible objects and their associated vectors X span the *pattern space P* of dimension *n*. The task of pattern recognition is then to find a set of hyperplanes which divide P into disjoint regions of classes.

The specific task of a computer program used for automatic or assisted pattern recognition is nicely described in the following quotation (Andrews 1972).

'Mathematically, pattern recognition is a classification problem.... The major goal in designing a pattern recognition machine is to have a low probability of misclassification'.

A simple example may illustrate this. In figure 4, the printed numerals 0 to 9 are entered in a measuring grid of 4 × 6 lines each. We want to count the number of times each grid line crosses a dark area of the numeral. One can easily imagine a simple device with a laser beam and a photodiode performing this count.

Taking the complete pattern of vertical and horizontal crossings, we have the following distinct classes for the numerals 0 to 9:

0220222222, 0110111111, 0230221111, 0240111112, 0210122211, 0330112112, 0330213222, 0220111111, 0440221222, 0330222312.

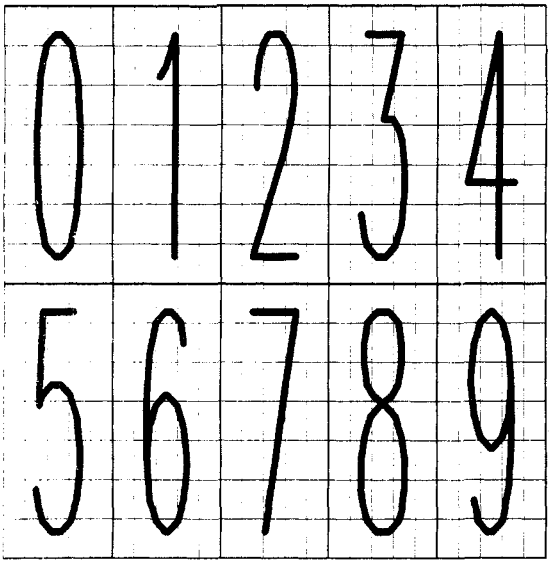

Figure 4. Printed numerals in boxes with scan lines.

A pattern recognition machine would now compare the scanning pattern of a character with all of these classes, in order to attempt to find a match, possibly allowing for slight deviations from the ideal pattern as long as this can be done without risk. If a match cannot be found, the unknown object would be entered in the class of 'unidentified objects', which in practice has always to be added to the classes of known objects. The method outlined here, called 'template matching', is also used in high-energy physics pattern recognition.

2.2. Pattern space and feature space

In many cases, the classification is not performed directly in P, but in the *feature space* F which is spanned by vectors Z of dimension m with

$$Z = R(X)$$

where R is a linear or non-linear transformation. The major purposes of this transformation are

(1) to reduce the dimension of the vectors to be studied, without losing significant information, and

(2) to obtain vector components which are better suited for pattern recognition than the original ones. This transformation is called 'feature extraction'.

As an example, consider a vector X consisting of 100 measurements along a straight line. An obvious feature extraction algorithm would perform a least squares fit and represent the track in space by its four parameters plus the chi-square. These five variables would contain enough information for many applications, but of course not the complete information contained in X (the information on systematic errors in a given measurement plane is lost).

Rather widely used because of its relative simplicity is linear feature extraction which, after a shift of the origin, consists of a matrix multiplication:

$$Z = T(X - X_0).$$

This transformation matrix is found as follows.

Suppose that for a given phase-space region, a 'training sample' of several hundreds or thousands of trajectories are given by n coordinates (or less) along each trajectory (e.g., x or y values for given z values). Then each trajectory defines one point in the n-dimensional pattern space. However, since there are only five free parameters per trajectory, these points will occupy a five-dimensional subspace of the pattern space. This subspace is defined by the n-5 constraint equations between the n coordinates. It is normally curved and the constraint equations are normally not known explicitly since they cannot be expressed in analytical form in the general case.

One can, however, for relatively small phase-space regions, find an m-dimensional hyperplane which contains all tracks, where m is equal to or greater than five. By defining a new system of coordinates in this hyperplane, one can now describe all tracks with this small number of m coordinates. This 'feature space' is constructed following the approach of Karhunen–Loeve (Young and Calvert 1974).

The eigenvectors of the matrix

$$F = \langle (X - \langle X \rangle) * (X - \langle X \rangle)^{\dagger} \rangle$$

(† = transpose) form the rows of the transformation matrix T, such that the new coordinates become

$$Y = T * X.$$

In the above, $\langle \ldots \rangle$ stands for 'mean over the training sample'.

The reasoning behind this approach can be outlined as follows. The matrix F, which is actually the unnormalised covariance matrix of the track sample, can be expressed as

$$F = \langle A * A^\dagger \rangle \qquad \text{with} \qquad A = (X - \langle X \rangle).$$

If E_i is a normalised eigenvector of F corresponding to the eigenvalue e_i, then

$$E_i^\dagger * F * E_i = E_i^\dagger * \langle A * A^\dagger \rangle * E_i = e_i$$

holds. From this follows

$$\langle E_i^\dagger * A * A^\dagger * E_i \rangle = e_i$$

or

$$\langle (E_i^\dagger * A)^2 \rangle = e_i.$$

Since $\langle \ldots \rangle$ stands for 'mean over training sample', e_i is equal to a sum of squares and therefore not negative. If the vectors A, i.e. the original track vectors, span the full pattern space, then the eigenvalues are all positive, since

$$e_i = 0 \qquad \text{means} \qquad \langle (E_i^\dagger * A)^2 \rangle = 0$$

and therefore $E_i = 0$ because of the condition above, which is a contradiction to the assumption of E_i being normalised. In the case where the tracks span the full original pattern space F is therefore positive-definite, otherwise it is semi-definite with its rank equal to the dimension of the hyperspace spanned by the track sample.

The aim of feature extraction is to find a basis in pattern space such that the hyperspace spanned by the first m vectors of this basis 'contains' all tracks in the sample, which is to say that if a track vector is decomposed into two parts, one entirely contained in the hyperspace of dimension m, and a second part orthogonal to it, then the length of this second part, which is the distance of the true track point from the hyperspace, is smaller (absolutely or relatively) than a given error bound. This can be written in mathematical terms as follows. Let Q_i $(i = 1, \ldots, n)$ be an arbitrary orthonormal basis in the n-dimensional pattern space. Then the object vectors A can be written as

$$\begin{aligned} A_j &= \sum_{i=1}^{n} a_{ji} Q_i \qquad (2.1) \\ &= \sum_{i=1}^{m} a_{ji} Q_i + \sum_{i=m+1}^{n} a_{ji} Q_i. \end{aligned}$$

The 'best' linear feature extraction consists therefore obviously in finding the basis Q_i which minimises the mean quadratic error if the expansion above is truncated at m, i.e. which minimises

$$S(m) = \left\langle \left[A_j - \sum_{i=1}^{m} a_{ji} Q_i \right]^2 \right\rangle = \left\langle \left[\sum_{i=m+1}^{n} a_{ji} Q_i \right]^2 \right\rangle.$$

Because the basis Q_i is orthonormal, multiplication of (2.1) above with $Q_i^\dagger$ gives

$$a_{ji} = Q_i^\dagger * A_j$$

leading to

$$\begin{aligned} S(m) &= \langle [\sum a_{ji} Q_i]^2 \rangle \\ &= \langle [\sum (Q_i^\dagger * A_j) Q_i]^2 \rangle \\ &= \left\langle \sum_{i=m+1}^{n} \sum_{k=m+1}^{n} (Q_i^\dagger * A_j) Q_i^\dagger * Q_k (A_j^\dagger * Q_k) \right\rangle \\ &= \left\langle \sum_{i=m+1}^{n} (Q_i^\dagger * A_j)(A_j^\dagger * Q_i) \right\rangle \qquad \text{since } Q \text{ orthonormal} \\ &= \sum Q_i^\dagger * \langle A_j * A_j^\dagger \rangle * Q_i \\ &= \sum_{i=m+1}^{n} Q_i^\dagger * F * Q_i. \end{aligned}$$

Since F is at least semi-definite, as has been shown before, minimising this expression means minimising each term separately. Introducing Lagrange multipliers λ_i means therefore that

$$Q_i^\dagger * F * Q_i - \lambda_i Q_i^\dagger * Q_i + \lambda_i$$

has to have a minimum.

Setting the derivatives of this expression with respect to the components of Q_i to zero gives the minimum condition for $S(m)$:

$$F * Q_i = \lambda_i Q_i$$

i.e. Q_i is an eigenvector of F, with eigenvalue λ_i, which we called e_i before. The minimum of $S(m)$ is then simply

$$S(m) = \sum_{i=m+1}^{n} \lambda_i.$$

By taking the eigenvectors of the largest m eigenvalues as the new basis, one has then found the 'best' hyperspace obtainable with linear feature extraction. The average distance of any track in the sample to this hyperspace is given by $S(m)$.

In the general case a certain number, m (greater than or equal to five), of the components of Y will be significant, and the remaining $(n-m)$ components will have small absolute values (Brun *et al* 1980).

This 'feature extraction', where the most significant eigenvectors form the coordinate system in 'feature space', leads potentially to the following three improvements.

(i) The dimension m is often much smaller than n, thus making the manipulation of the tracks (=vectors) simpler.

(ii) The $n-m$ insignificant components of Y can serve as constraints in order to decide whether a given vector X represents a trajectory or not.

(iii) Even in the case that no reduction of the dimensionality is possible (i.e. $m=n$), the new coordinates are now given in the order of their importance, which means that in an analytical approximation, such as a multi-dimensional polynomial, fewer terms will be needed in Y than in X components (Brun *et al* 1975).

To illustrate feature extraction with the help of a simple example, I have constructed a track sample of 2000 tracks in an idealised detector for fixed target physics, and then applied the Karhunen–Loeve feature extraction algorithm.

The detector consists of eight planes equally spaced along the x axis, between $x = 1$ m and $x = 5$ m, orthogonal to the x axis, and measuring both y and z simultaneously. A homogeneous magnetic field of 0.5 T exists in the z direction. The tracks all originate at (0, 0, 0) and correspond to particles of both charges. The 1000 different track parameter values are simply obtained by taking ten equidistant values in each of the three variables $1/p$ (where p is the momentum), ϕ (the angle in the xy plane between the track and the x axis, at the origin), and θ, in this case chosen to be the angle of the track with respect to the xy plane. The respective intervals were 5 GeV/c to 50 GeV/c for p, and -100 mrad to $+100$ mrad for both ϕ and θ. To each hit in the eight detector planes, a 'measurement error' was added, being a Gaussian distributed error around the track with $\sigma = 0.2$ mm.

This track sample was entered into the program LINTRA (Brun *et al* 1980) which gave the following result.

The eigenvalues were 0.500, 0.487, 0.013, 0.7×10^{-6}, 0.4×10^{-6}, etc, down to the lowest eigenvalue of 0.2×10^{-7}.

The sharp drop after the third eigenvalue means that the track points of the sample actually occupy almost precisely a three-dimensional hyperspace of the original 16-dimensional pattern space (eight y and eight z coordinates for each track). This outcome is not surprising. Since the tracks in the sample are not very much curved, they can be approximated by parabolas in the xy projection, and straight lines in the xz projection. Since a parabola is a linear expression in three of its y coordinates, and one y value is always zero (at the track origin), and since a straight line is of course a linear expression in two z values, in this case one of them always zero as well, all error-free tracks that can be precisely described by a parabola and a straight line respectively in the xy and xz projection, and passing through the origin, will be contained entirely in a three-dimensional hyperspace.

The basis consisting of the eigenvectors can now be used to distinguish correct and wrong points along a trajectory. To this end, the sum of the squares of components four to eight of the transformed coordinates was entered into a histogram (figure 5(a)) for all tracks of the original track sample. Then an error of 4 mm in y and z was added to all tracks only in plane no 3, and the corresponding sum of squares from the transformed coordinates was entered into another histogram (figure 5(b)). As can be seen, the linear feature extraction provides in this case a powerful means of separating correct and wrong track candidates.

2.3. Parametrisation

This is the name given to an algorithm which allows values such as track coordinates to be expressed as functions of a variable such as the track length or a detector plane position. These functions depend of course on parameters, such as the five kinematic constants of a trajectory.

In high-energy physics, parametrisation has been applied in three different forms.

(1) To calculate the coordinates of a track in a given set of detector planes as functions of the plane positions, with the five kinematic quantities as parameters. This is useful both for Monte Carlo simulation and for track fitting, and is of course very

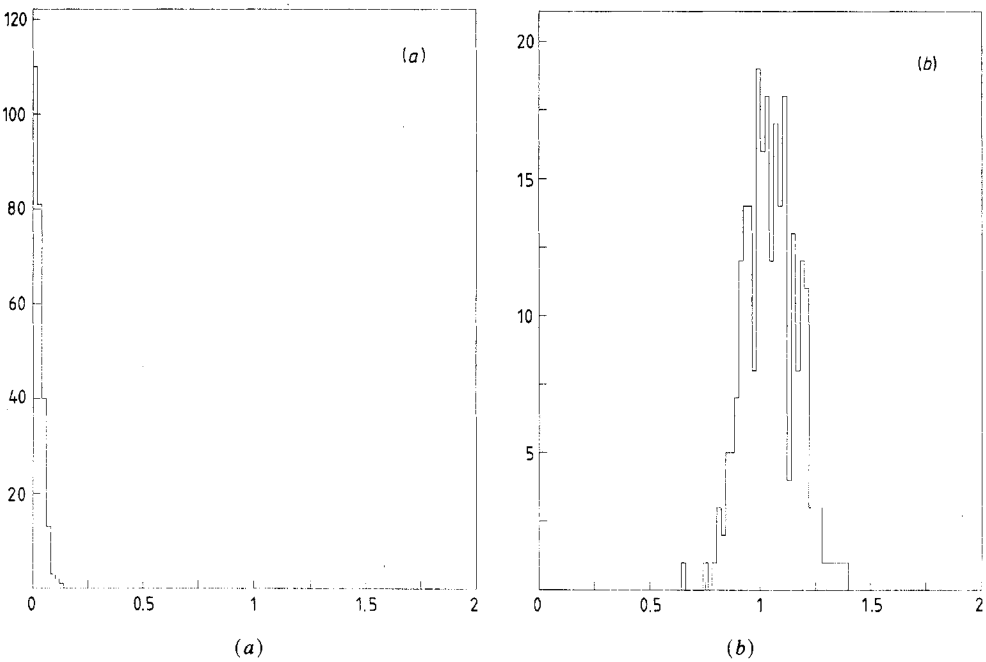

Figure 5. (*a*) Histogram of *d* (in arbitrary units), where *d* is the sum of squares of the feature space coordinate 4 to 8. (*b*) The same histogram as in (*a*), but with an error of 4 mm added to both *y* and *z* in plane no 3.

much faster than tracking each particle individually through the magnetic field and then calculating the plane hits.

(2) To parametrise some of the track hits in planes as functions of preceding track hits. This is used in track finding to predict coordinates once a part of the track has been found (Lassalle *et al* 1980).

(3) Do the inverse of (1) above, namely to express the five kinematics quantities as functions of the hits (=coordinates) in a given detector. These values are normally not good enough to be final, but may represent good starting values for an iterative fit procedure.

In all three cases, the most obvious merit of the method is its speed. Indeed, if the magnetic field is rather well behaved, and, more importantly, if the phase-space region in question is small (Aubert and Broll 1974), one can hope to express the desired quantities by polynominals of relatively low order. This is a corollary of the fact that the magnetic field need no longer be used explicitly once the parametrisation has been performed.

However, the method is difficult to apply and normally requires much study and optimisation effort. The choice of the phase-space region and the selection of the 'training sample' are particularly delicate operations. For example, it is important to choose more points in pattern space near to the borders of the trajectory cluster than near to the centre.

Worst of all, however, is the fact that any given parametrisation is only valid for a fixed detector arrangement and each change in the detector set-up and different subset of planes that must be treated (owing to detector inefficiencies) require a new parametrisation.

2.4. *Pattern recognition and high-energy physics*

If we apply the definition of pattern recognition given in §2.1 to high-energy physics events as objects, it could comprise the analysis of an event from signal decoding to the final classification (e.g. classifying the event as a Z_0 decay plus some extra particles), thus encompassing the full operation of track finding, cluster finding in calorimeters, track fitting and kinematics. This definition is not very practical, because there is almost an infinity of target classes. This is why we are concerned with track reconstruction and event reconstruction separately.

This leads to the more practical approach of applying the principles of pattern recognition to track finding alone. In this case, the objects are space points or points in projections, track segments or the like, and the classes are track elements or complete tracks.

It is useful to compare track recognition with the example of classification of numerals. On the one hand, in high-energy physics we have simpler shapes (straight lines, arcs of circles or helices in space in most cases). On the other hand, these objects occur almost anywhere and not just in precisely defined boxes, they can have almost any orientation and they can overlap. This last problem is the worst and corresponds to typing several numerals into one and the same box and then finding them all without knowing how many there are.

The most important problem in general pattern recognition as it is treated in text books is normally to find the decision function d which selects the correct class. The vector of measurements X is normally assumed as given. In track recognition for high-energy physics, d is known: it is actually a procedure such as a least squares fit for comparing X with a solution of equation (1.3) and deciding whether X is a track or not, based on the goodness of the fit. The problem in high-energy physics is rather to define X, i.e. find a suitable way of selecting good 'track candidates' for the test d, since trying all possible combinations of measured signals is prohibitive in most cases. For this reason, rather little of relevance to our needs can be found in text books and articles on pattern recognition in general, with the exception of those methods which apply to a large number of problems in cluster analysis, such as the minimum spanning tree.

Track finding in high-energy physics is mostly based on a mixture of simple basic principles and detector-specific *ad hoc* algorithms. There are, however, a few mathematically well founded methods available. These, in turn, normally will only perform correctly under certain well defined conditions. For this reason, it is essential to choose the track finding and fitting algorithms at the time of the detector design and to possibly even modify the detector accordingly. A detector which is badly suited for track reconstruction will be a continuous source of troubles during the subsequent analysis.

3. Track finding applications

3.1. *Points in projections or in space*

It has been pointed out before that wire chambers normally provide only one coordinate, whereas two are needed to reconstruct the impact point of a particle in the plane of the wires. If several planes with parallel wires are placed behind each other, they provide one projection of the track onto a plane orthogonal to the wires. For a reconstruction of the track in space, at least two different projections are necessary and extra projections provide further information which can be invaluable if the chambers are less than 100% efficient, or if the wires in different chambers are more or less orthogonal, in which case it is needed to correlate the two projections of a track. The different projections can be provided either by high voltage strip read-out, charge division or wire planes with different wire orientations.

To reconstruct the trajectory in space, as is necessary in the great majority of cases, there are basically the following choices.

(i) Combine several local projections into space points and perform the track finding on those space points.

(ii) Find the tracks in the different projections independently, and subsequently match the tracks in the different projections.

(iii) Find the tracks only in some of the projections and match them with the help of the other projection(s). In this case, one needs fewer points in the projections used only for matching, since the tracks are not reconstructed here.

Space points can be reconstructed from a wire and one or two high voltage stip read-out coordinates on the corresponding cathode plane(s). This point will give the avalanche position at the wire hit, regardless of the particle's impact angle with the wire plane, since the induction spot is always opposite the avalanche. This means that the particle does not necessarily cross the strips which give a signal. The space point reconstruction is normally preferred to track finding in the wire and high voltage projections independently. Since the induction spot size is typically one cm or more, signals from nearby tracks frequently overlap in the cathode plane and, as a result, the track finding algorithms may find fewer tracks here than among the more precise anode wire signals.

When the different projections are all provided by similar wire planes, it is possible to proceed either by reconstructing 'space points' and then associating these into tracks, or by finding the tracks in the projections independently and then matching them. In the first case four wire planes are necessary to define each 'space point', no three of which may have identical wire directions, since one needs four straight lines to define a unique line in space (which cuts all four). This approach not only provides a space point, e.g., the centre of the track segment thus constructed, but also the trajectory direction at that point, which is certainly an advantage (Eichinger 1980).

3.2. Efficiency

When trying to optimise an algorithm, it is desirable to have a quantitative measure of its performance. This measure is called 'efficiency' in the case of pattern recognition in high-energy physics. It is surprising, at least at first sight, that the papers on this subject almost never discuss the 'efficiency' obtained. It is certain, on the other hand, that all of the authors must have used such a measure to evaluate their methods.

The reason for this absence is almost certainly to be found in the lack of a universal definition of such a number, and therefore in the fear of unjustified comparison and misinterpretation. To illustrate this, let us consider a sample of m events with an average number of n tracks in each event. There are several possibilities for defining an efficiency.

(1) The average probability of finding one single track is the number of correct tracks found divided by the total number of tracks (mn). This number will normally decrease fairly slowly with increasing n, as the events become more confused.

(2) The fraction of events with *all* tracks found correctly. This represents a very severe measure and normally decreases rapidly with *n*.

(3) The fraction of events in which at least a minimum fraction, say 90%, of the tracks has been found correctly.

(4) Any of the above metrics, but not counting certain tracks which are notoriously difficult to find, such as tracks below a certain momentum, or in particularly tricky detector regions.

All of these approaches, however, are far from providing a satisfactory definition of efficiency. To be of any use, such a definition would have to include in some way the number of incorrect tracks constructed, *i.e.* those tracks which are made from points belonging to different tracks in reality, or containing spurious points which cannot be reasonably associated with any track: the higher the number of these incorrect tracks, the lower the efficiency must become. Otherwise the best method with a guaranteed efficiency of 100% would simply consist of listing all possible point combinations, among which the correct tracks must be present.

In view of all this, it should be much less surprising that such figures are not quoted. They can only be understood in a specific context and can normally not be compared with each other between different experiments. On the other hand, everybody undertaking to write and test a track finding method will have to define such a number for himself.

It should finally be pointed out that the computing time used by a given method is also important, and sometimes forces compromises to be made between a good, but awfully slow, and a fast, but less efficient method. In most cases, however, the efficiency is the more important consideration and one will normally try to speed up a good method by extending the algorithm which gives the desired results rather than to use a less efficient method, or, as Weinberg (1972) has put it, 'Any program that works is better than any program that doesn't.'

3.3. *The task of track finding*

Given a set of position measurements in a detector, the task of track finding is to split this set into sub-sets (=classes) such that

(i) each class contains measurements which could be caused by the same particle and

(ii) one class contains all measurements that cannot be associated with particles with sufficient certainty.

This definition is modest enough to represent a realistic goal. It reduces track finding to a cluster analysis problem, where a cluster is described as follows (Andrews 1972): 'A cluster is loosely defined as a collection of vectors or points that are close together'.

As a consequence, cluster analysis is mainly concerned with defining suitable 'distances' between pairs of objects and to provide algorithms for clustering based on these distances. The minimum spanning tree algorithm belongs there and will be described in more detail later, since it is one of the few examples of applying a non-heuristic method to real-life track finding.

3.4. *Methods of track recognition*

As should have become clear in the preceding outline of the principles of pattern recognition, all methods basically require a two-step procedure. In the first step, a subset of measurements is selected, forming a 'track candidate'. In the second step, a decision function is used to check whether or not the subset is an acceptable track.

This two-step procedure is often split further in order to make it faster. In an initialisation phase, a certain number of measurements are selected and they are either rejected or accepted. If they are accepted then additional measurements are processed, in several steps, until a final check is performed on the fully assembled track. The gain in speed over the basic two-step procedure, which first selects a full track candidate and then applies a decision function, is twofold. Firstly, each reduction in the number of measured values ('points') per track candidate brings a considerable reduction in the number of such candidates, since the number of combinations grows with the power of this value. Secondly, the application of the final decision function can be very time consuming, for example, if an iterative fit to a track model has to be performed. Therefore, a sequence of decision functions from simple and fast to precise but slow improves the speed if really at each intermediate test a good fraction of the wrong candidates is rejected. Of course, great care is needed to avoid the rejection of good track candidates by any of the approximate and simple tests.

If, for a given method, all possible tracks have been found, a certain number of points will normally remain unassigned to any track. They form the 'background'. If several different methods are applied, or the same method in different ways, a good measure of the efficiency is essential. Such a measure, as pointed out before, is specific to each experiment.

3.4.1. The complete combinatorial method. This works in the following way:

(i) split the set of all position measurements into all possible subsets and

(ii) fit a track model to each sub-set (=track candidate) and call it a track if it fits the model.

Although this method can be directly applied in some very simple cases, where there are very few coordinates in total, for most practical problems it is too time consuming. Imagine five tracks in ten planes, which produce 50 measurements, and assume that one ms is required to fit the track model to each subset. Even if we ignore possible multiple hits of tracks in the same plane, the processing will take about 3 h of computing time, which is normally prohibitive.

One should, however, bear in mind that the major objective of all track finding methods is to reduce this excessive computing time while trying to keep the efficiency as high as possible. This is not always easy.

Besides the complete combinatorial method, the different track finding methods can be classified as 'global' or 'local'. I call a method 'global' if all objects (points) enter into an algorithm in the same way. This algorithm produces a table of tracks, or at least a table in which the tracks can be found more easily than among the original data. The algorithm can therefore be considered as a general transformation of the totality of the event coordinates. The computing time of a global method should in principle be proportional to n, the number of points in the event.

A 'local' method, on the contrary, is one that selects one track candidate at a time, typically by starting with a few points only (track candidate initialisation) and then making predictions as to further points belonging to this track candidate, e.g., by interpolation or extrapolation of the current track model based on the track candidate found so far. If additional points are found, they are added to the candidate, otherwise the candidate is dropped after a certain number of attempts, depending on the degree of detector inefficiency the algorithm wants to allow for. Since local methods invariably have to make fruitless attempts in order to find track candidates, and thus use the same point in different combinations, the computing time grows faster than linearly with the number of points.

Pure global methods are independent of the order in which points enter the algorithm, local methods are not, since the treatment of each point depends on the initialisation, and the 'track finding history' so far in general. A good track finding algorithm gives the same tracks even if the order of the raw data from each chamber is randomised, and therefore the measurements enter the track finding algorithm in a different order.

3.5. Local methods

3.5.1. Track following method. This is typically applied to tracks of the 'perceptual' type, where the track can be more or less easily recognised by the human eye from the displayed coordinates. An initial track segment is first selected, consisting of a few points (one up to three or four), and this segment is normally chosen as far away from the interaction region as possible, since there the tracks are at least on the average more separated than anywhere nearer to it.

In the next step, a point is predicted by extrapolation in the next chamber towards the vertex. This extrapolation may be of 'zero' order simply by choosing the nearest-neighbour, first-order (straight line), second-order (parabola) and possibly higher ones, or other track forms such as circles or helices. The aim is in all cases to have a fast algorithm of point prediction, representing the track locally by the simplest model possible. For chambers which are sufficiently close, the parabola extrapolation will be sufficient in many cases with magnetic fields, since it preserves the sign of curvature and therefore behaves like a real particle track. In addition, it is very fast, since a parabola through three points can be written as

$$y = a_1(x)y_1 + a_2(x)y_2 + a_3(x)y_3$$

where the coefficients depend only on the plane positions and can therefore be calculated once and for all for the prediction of hits in any of the chambers.

Here particular mention should be made of the track parameterisation which has been applied successfully in the Omega detector (Lassalle *et al* 1980), and which in this specific case works as follows. From a track sample generated at the target and tracked through the detector, the Karhunen–Loeve feature extraction calculates the significant coordinates as well as the insignificant ones which can be used as constraint equations. During track finding, the tracks are followed in space, although each detector plane measures only one coordinate. The linear constraint equations are used to express a coordinate as a function of other coordinates which have been found previously. It should be noted that in reaility this is not an extrapolation but an interpolation since, owing to the choice of the track sample for the parametrisation, the target region is implicitly used as constraint for the track coordinates.

Another example of track following can be found in Mess *et al* (1980), where muon tracks are followed through a calorimeter, this task being made more difficult by multiple scattering and the presence of hadron showers.

In summary, the track following method is in one way or another concerned only with the local track model, since it always looks only at the next few points, using the most recently found ones to extrapolate the track. This therefore allows a simple (i.e. fast) track model. On the other hand, once the distances become too big, the approximate model will not be precise enough and because of the measurement errors, even an absolutely correct tracking based on a few recently found points is problematic, since most detectors deliver sufficiently precise track parameters only when the full track is used in a fit.

The track following method uses combinatorial initialisation of candidates. Once a few points have been added to a candidate, this is very likely a good track, thus keeping the overhead rather small. Accordingly, the computing time is normally proportional to a number between n and n^2 ($n =$ number of points).

3.5.2. Track road method. In this case, there is no extrapolation as in the track following method, but the much more precise interpolation between points is used to predict extra points on the track. Therefore, by using initial points at both ends and one point in the centre in the case of curved tracks, a simple model of the track is now used to predict the positions of further points on the track, by defining a 'road' around the track model. This track model may be (almost) precise, such as a circle in the case of a homogeneous solenoidal field, or a straight line in a field-free region; or it may be approximate, in which case the width of the road has to be established by Monte Carlo tracks. In principle, the better the model the narrower the road can be, but one can rarely use the theoretical road width of, say, three standard deviations of the detector resolution. This may be due to systematic errors in the position, to signal clusters, to signals being hidden by background signals, etc. The method of track roads is slower then the track following method; but sometimes it has to be chosen when it is the only workable method available, particularly in the case of widely spaced detector planes (Froehlich *et al* 1976). Most modern detectors provide a density of measurements that is high enough to permit the use of faster methods.

Since (in a magnetic field) a road has to be initialised by three points, combinations in different planes have to be chosen because of detector inefficiencies, and most initial combinations of three points will be wrong, the computing time of this method is typically proportional to a factor between n^2 and n^3 ($n =$ number of points).

3.5.3. Track element method. Here, a track candidate is constructed in two steps: firstly, short track elements are made up of points, normally inside 'natural' sub-divisions of the detector such as drift chamber cells. From this track candidate, zero-order (nearest-neighbour), first-order (straight line), or second-order (parabola) extrapolation or interpolation are used to define track elements, each of which is then condensed into a 'master point' (the weighted average of the cluster) plus a direction. In the second step, these master points are then combined through track following or other track finding methods.

The great advantage of this method is its speed, compared to using all (up to several hundred) points per track directly. It is therefore appropriate for detectors with a very high point density and was, historically, frequently used for bubble chamber analysis. In addition, the left-right ambiguity of drift chambers can be solved at the track element level. The reduced number of points and their wider spacing are compensated by their higher precision and the fact that they have a direction associated with them (Eichinger 1980).

This method has been applied in the JADE detector at DESY (Olsson *et al* 1980). There, the cylindrical drift chamber surrounds the beam tube inside the homogeneous field of a solenoid magnet. The drift chamber sense wires are thus parallel to the beams and magnetic field lines. They are staggered by 200 μm to either side in order to be able to distinguish signals from their mirror images. Three concentric rings of chambers provide up to 48 points for an outgoing track.

Charge division on the wires delivers a z coordinate, but with two orders of magnitude less precision. Therefore, the track finding is performed in the xy plane of the drift time signals only, which is orthogonal to the wires. Track elements are made out of four or more points in each of the 96 detector cells separately. To each segment, a parabola is fitted and the mirror image tracks are rejected at this stage because of their bad chi-square. Track segments are then connected through quadratic extrapolation into full tracks. A similar method, described below, is used in the UA1 detector at CERN.

It is rare that any of the above methods are applied in its pure form to a real detector. A good illustration of this can be found in the track finding for the central chamber of the UA1 detector (Pimiä 1985) which has to deal with probably the most complicated event topologies seen so far (figure 1). For this reason, the track finding procedure will now be described in more detail, and may serve as a demonstration of the different principles that have been discussed.

The UA1 central detector surrounds one of the regions where the proton and antiproton beams collide in the CERN SPS. It consists of a cylindrical arrangement of drift chambers (5.8 m long, 2.3 m diameter) inside a uniform magnetic field of 0.7 T perpendicular to the beams. The drift chamber wires are all parallel to this field, but the drift direction of the electrons is vertical in the central, and horizontal in the forward-backward part of the chamber. There are about 6100 sense wires, providing on the average 70 position measurements per charged particle. Each event shows tracks of between 20 and 200 charged particles. Since the drift space is almost entirely on one side of the sense wires only, there are practically no mirror images of points present.

The track finding is performed in four steps:

(1) chaining of points by local track following;

(2) iterative pairing of local chains to make them longer;

(3) condensation of chains into master points after application of position corrections; and

(4) linking of track segments with a road method.

All of this is done in the projection of the drift coordinates and the wire positions. The third dimensions in only added when the tracks have been found, for reasons which will become clear later. In the first step, the coordinates can be used without correction or even conversion, i.e. the algorithm works with wire numbers and drift times directly. The track following is done using linear extrapolation: a straight line is fitted (least squares method) to the n previous points in the chain. The minimum of n is three (initially) and the maximum eight. The track following stops when no more points are found, when a region with overlapping tracks is reached, or when a quality check fails, based on the chi-square of the fit. In this way, point chains of variable length are constructed. While they are being built up, a search for kinks in the tracks is performed, which shows up in a sudden change in the direction.

These point chains are not well suited for further extrapolation which requires rather precise track parameters. One half of them are less than 5 cm long. Therefore, an iterative pairing of segments follows as a second step, which only stops when the chains can be extended no further. The process is based on a least squares parabola fit to all points in the chain pair candidate.

In step three, the segments are parametrised in two stages. In the first stage, master points are defined for sub-segments containing three to five points from each chain. Before this is done, the coordinates of the points are corrected for two effects: the first is the deflection of the drifting electrons by the magnetic field, which makes them drift towards the sense wires at an angle of 24°; the second simply takes the particle's time of flight from the interaction point to the ionisation region into account. The master points of the sub-segments are then defined through a least squares parabola fit. In the second stage a circular fit, which is the correct track model, is made to all master points on a segment. This now becomes necessary because the segments may be up to two metres long and their parameters are used for extrapolation during the next step. At this stage, segments with radii of less than 50 cm are rejected, which corresponds to a momentum cut of $p_{xy} < 0.1$ GeV/c.

In step four, tracks are finally assembled from segments by always starting with the longest segment still available, because the precision of the extrapolation is proportional to the square of the segment length. In addition, the longest segments exist typically in the cleanest region of the event. A combinatorial search is performed for segments which could be added to the already existing part of the track. Such a segment candidate has first to undergo a preselection procedure checking the position, angle and curvature with respect to the extrapolated road; the gap between the two parts is limited in size also. If the segment candidate passes this test, a circle is fitted to all master points of the extended track part, which is accepted if the chi-square is correct. This process is continued until no more extensions are found and then a new track is started.

Finally, a clean-up procedure removes tracks which are too short (less than 12 consecutive or ten non-consecutive points). Segments which are doubly assigned are also dropped, since they are most likely to contain digitisings from overlapping tracks, in which case the performance of the drift chambers is bad anyway.

Only after the tracks have been found is the z coordinate added. This coordinate is generated by charge division on the wires, so that, at least for wires with single hits, there are true space points available. However, this read-out method is not precise enough for track finding. Therefore, it is only now that a helix is fitted to the complete track in space, on which occasion doubtful z values are dropped.

In this summary of the UA1 track finding, many additional details and complications have been left out, but the level of complexity of such a formidable task should have become apparent. Interested readers are referred to the excellent paper by Pimiä (1985).

3.6. Global methods

A totally different approach is used in the 'global methods', which are also applicable to a wider range of problems in cluster analysis. In this case, all points are considered together and a procedure exists for classifying all tracks simultaneously.

3.6.1. Histogramming method. In this case, one defines a set of n different functions of the point coordinates and enters the function values in a histogram. Then (if the method works correctly) the tracks form 'clusters' or 'peaks' in the histogram; these have then 'only' to be found and the problem is solved.

A simple example may illustrate this method.

Suppose that the interactions always take place at the same point and that there is no magnetic field. In this case, the tracks will form a 'star' around the interaction point (at 0, 0 in a suitable projection). This could, for example, be the case in a colliding beam machine, in the projection orthogonal to the beam direction. If we introduce an x, y coordinate system, and use as function any of

$$y/x \qquad \sin^{-1}[y/\sqrt{(x^2+y^2)}] \qquad \tan^{-1}(y/x)$$

then the tracks will appear as peaks at specific function values.

Another application of the histogram method is possible if the tracks are circles through the origin (interaction point), as in figure 6(a). In this case, the inverse (conformal) transformation

$$u = x/(x^2+y^2)^{1/2} \qquad v = y/(x^2+y^2)^{1/2}$$

will produce straight lines in the uv plane, their nearest distance from the origin being

$$d = 1/(2R)$$

with R = track radius (figure 6(b)).

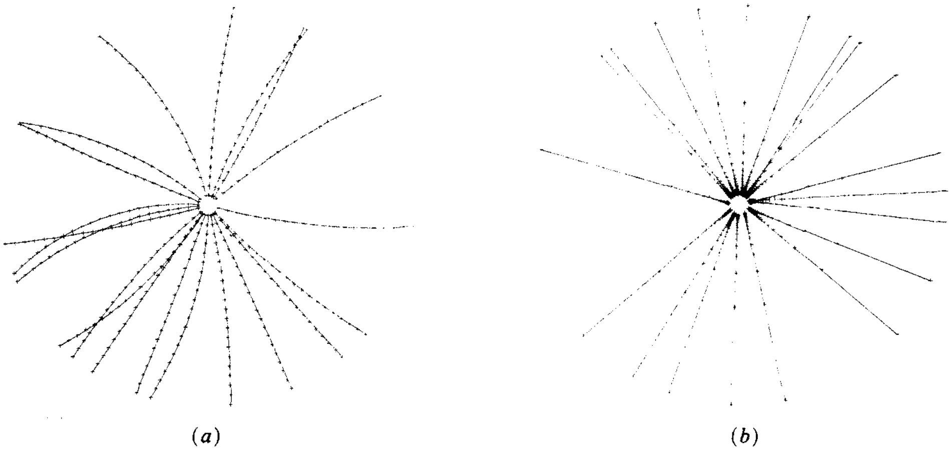

Figure 6. (a) Circles through the origin cutting a set of concentric circles around the origin. (b) The above circles after a conformal transformation.

The histogram method, if applied to this conformal mapping, will tolerate a certain distance from the origin and still find the tracks correctly. Overlaps will of course have to be treated separately, but at least the method will allow us to find well separated tracks quickly. This method has been applied in experiment R807 at CERN (Dahl-Jensen 1979).

In histograms of several dimensions, recognising the track clusters turns out to be more difficult than finding the tracks directly via a track model. This limits this method to one or two projections.

3.6.2. Template matching. This method requires a dictionary of all possible classes (=tracks) and can therefore be applied only in cases where their number can be kept within reasonable limits, somewhere below 10^5, in which case a binary search or a hash algorithm can yield very fast matches.

Used in the track finding for the Mark III detector at SPEAR (Becker *et al* 1984), with a total of 12 832 templates, this method is reported to run three times faster than 'conventional approaches'. Each different combination of cells that can be 'fired' by one track creates a separate template. The sub-division of the detector volume into 'cells' which leads to such a low number of possible combinations means of course that there will be regions of confusion where tracks get very near, or cross. These ambiguous regions are then resolved in a 'classical' way, which in the case of Mark III is a combinatorial, non-iterative circle fit.

This method is particularly well suited for detectors with cylindrical drift chamber arrangements, since the division of such chambers into drift cells provides a natural basis for the algorithm. In the case of Mark III, it works basically as follows.

During raw data conversion ('unpacking'), a cell image matrix of the detector is filled. Combinations of cells having 'fired' are then compared with the dictionary, once all data of the event have been unpacked. The method is therefore global except for the combinatorial search for tracks in regions where they overlap.

Template matching has been implemented in the fast trigger logic of the CELLO detector at DESY (Behrend 1981). In this case, only 57 templates are used.

For the experiment E-711 at Fermilab, a template matching algorithm was implemented on a fast vector processor. This reduced the computer time per event by a factor of ten compared to the same algorithm run in scalar mode on the same computer. However, the scalar code had to be rewritten completely for vectorisation (Georgiopoulos *et al* 1986).

3.6.3. Minimum spanning tree. In order to understand this technique, one has to have a basic notion of graph theory.

A 'graph' consists of 'nodes' and 'edges', where the nodes may be represented by points and the edges by lines connecting these points. These 'points' are of course not geometric entities, but stand for any object on which a classification task has to be performed, be it a space point, a track segment, a complete track or anything else. In an 'edge-weighted' graph, each edge has a positive number assigned to it, which could be the Euclidean distance if the graph points stand for real space points. A 'connected graph' is a graph without isolated nodes. A 'tree' is a graph without loops. A 'spanning tree' is a connected graph without loops. Finally, a 'minimum spanning tree' is a tree in a given graph for which the sum of the edges is the minimum for all spanning trees of that graph.

Algorithms to actually construct a minimum spanning tree are given in Schorr (1976). Representing track points as nodes and pair distances as edge weights, one can construct a minimum spanning tree for a given event (Zahn 1973). Spurious hits will normally appear as nodes with only one neighbour and can easily be eliminated. Tracks can be found piecewise and then combined into full tracks using 'similarity' criteria (common points, similar slope, curvature).

3.6.4. Application of the minimum spanning tree. A variation of this method has been applied to track finding in the TASSO detector at DESY, where it has been shown to be more efficient than the road method (Cassel and Kowalski 1981). The basic element used in this algorithm is no longer a single point, but a point pair in two adjacent drift chamber layers. This pair has an associated pair distance and a direction. Pairs are linked into graphs when they share points and have similar directions. This second condition is particularly efficient in rejecting image points in drift chambers (left-right ambiguity).

In this way, track segments are constructed having a certain minimum length which permits us to calculate their track parameters (circle radius and centre). These track segments are then grouped into full tracks by using 'similarity' based on the segment parameters.

For a fast search for high momentum tracks, a modified minimum spanning tree technique is used, where the curvature of a segment defines the edge weight. In addition, this serves the purpose of rejecting arcs containing mirror points, since those will have much higher curvatures than the arcs made up of correct points only. The method is on the whole rather specific to drift chambers with left-right ambiguity, where it works very well.

4. Treatment after track finding

4.1. Incompatible tracks

In the case of events with many tracks in a narrow angular region, it happens frequently that more tracks are found than can reasonably coexist in the area in question. Typically, some of these tracks share almost all points, which is extremely unlikely to represent reality. These tracks are called 'incompatible'. They are normally due to filter criteria (in most cases the ultimate filter is the fit to a track model) which are too weak. This may be because the track model, the alignment, or the calibration is poor. In this case, the remedy consists clearly in a better model and better detector description. However, strict fit criteria may as well be prohibited by physics effects, such as multiple scattering or secondary ionisation by gamma rays, which tends to fire extra wires near to an electron avalanche, thus increasing the error on the point measured. In this case, where no further improvement can be achieved on the level of a single track, another graph theoretical approach can and has been used with success (Froehlich *et al* 1976), which is based on the assumption that the most likely event underlying the observed data is the one which consists of the maximum of compatible tracks, which may only share none or very few numbers of points. Therefore, the 'incompatibility graph' for the tracks found has to be solved, which is a standard procedure (Das 1973).

4.2. Single track and vertex fit

At some stage, the points on a track have to be fitted to the correct track model, taking all known effects into account: Gaussian errors in the position (Gluckstern 1963), non-Gaussian errors in drift chambers (Drijard *et al* 1980, James 1983), multiple scattering (Gluckstern 1963, Regler 1978, Highland 1975), magnetic fields (Metcalf and Regler 1973, Wind 1974, Mecking 1982) and energy loss (Bugge and Myrheim 1981). Whereas this fit may take place already during the track finding phase, the fit to a common vertex will in most cases only be performed when all tracks are found (Patrick and Schorr 1985, Billoir *et al* 1985, Hart *et al* 1984), except in cases where a vertex from a few tracks has to be used as a constraint in finding the remainder.

This whole area of track and vertex fitting is as important as the track finding for the final reconstruction of the event. However, it consists in principle of applying a well understood mathematical algorithm, and therefore lacks the somewhat heuristic nature of track finding in high-energy physics.

5. Conclusion

After years of development, one may now consider the field of track finding in high-energy physics as consolidated. Detectors, algorithms and computers have improved their performance to the extent that for a well planned experiment the track finding no longer plays the role of the great unknown villain it used to have in the past. Consolidation does in this case not mean stagnation, since the use of vector processors in particular, and probably parallel processors in the near future promises to considerably increase the speed of track finding and will possibly bring a radical change in the methods and algorithms that may be applied.

References

Andrews H C 1972 *Introduction to Mathematical Techniques in Pattern Recognition* (New York: Wiley-Interscience)

Aubert J J and Broll C 1974 *Nucl. Instrum. Methods* **120** 137–41

Becker J J, Brown J S, Coffman D, Dado S, Hauser J, Plaetzer S A, Russell J J, Schindler R H and Thaler J J 1984 *Preprint SLAC-PUB-3442*

Behrend H J 1981 *Comput. Phys. Commun.* **22** 365–74

Billoir P, Fruehwirth R and Regler M 1985 *Nucl. Instrum. Methods* A **241** 115–31

Bouclier R *et al* 1974 *Nucl. Instrum. Methods* **115** 235–44

Brun R, Bruyant F, McPherson A C and Zanarini P 1985 *Preprint CERN/DD/EE/84/1*

Brun R, Hansroul M and Kubler J 1980 *Preprint CERN/DD/US70*

Brun R, Hansroul M, Kubler J, Palazzi P and Wind H 1975 *Preprint CERN/DD/75/23*

Bugge L and Myrheim J 1981 *Nucl. Instrum. Methods* **179** 365–81

Cassel D G and Kowalski H 1981 *Nucl. Instrum. Methods* **185** 235–51

Charpak G 1978 *Phys. Today* **31** (10) 23–30

Dahl-Jensen E 1979 *CERN/R807/8 internal note*

Das S R 1973 *IEEE Trans. Comput.* **C-22** 187–93

Drijard D, Ekeloef T and Grote H 1980 *Nucl. Instrum. Methods* **176** 389–95

Eichinger H 1980 *Nucl. Instrum Methods* **176** 417–24

Eichinger H and Regler M 1981 *Preprint CERN 81/06*

Fabjan C W and Ludlam T 1982 *Ann. Rev. Nucl. Part. Sci.* **32** 335–89

Froehlich A, Grote H, Onions C and Ranjard F 1976 *Preprint CERN/DD/76/5*

Fu K 1980 *IEEE Trans. Comput.* **C-29** 845–54

Georgiopoulos C H, Goldman J H, Levinthal D and Hodous M F 1986 *Nucl. Instrum. Methods* A **249** 451–4

Gluckstern R L 1963 *Nucl. Instrum. Methods* **24** 381–9

Grote H 1981 *Preprint CERN 81/03* 136–81

Hart J C and Saxon D H 1984 *Nucl. Instrum. Methods* **220** 309–26

Highland V L 1975 *Nucl. Instrum. Methods* **129** 497–9

James F 1983 *Nucl. Instrum. Methods* **211** 145–52

Kanal L 1974 *IEEE Trans. Inf. Theory* **IT-20** 697–722

Lassalle J C, Carena F and Pensotti S 1980 *Nucl. Instrum. Methods* **176** 371–9

Mecking B 1982 *Nucl. Instrum. Methods* **203** 299–305

Mess K H, Metcalf M and Orr R S 1980 *Nucl. Instrum. Methods* **176** 349–54

Metcalf M 1986 *Adv. Comput.* **25** 277–333

Metcalf M and Regler M 1973 *Preprint CERN 73/2*

Olsson J, Steffen P, Goddard M C, Peace G F and Nozaki T 1980 *Nucl. Instrum. Methods* **176** 403–7

Patrick E A 1972 *Fundamentals of Pattern Recognition* (Englewood Cliffs, NJ: Prentice-Hall)

Patrick G N and Schorr B 1985 *Nucl. Instrum. Methods* A **241** 132–8

Pimiä M 1985 *University of Helsinki preprint HU-P-D45*

Regler M 1978 *Acta Phys. Aus.* **49** 37–45

Reingold M, Nievergelt J and Deo N 1977 *Combinatorial Algorithms, Theory and Practice* (Englewood Cliffs, NJ: Prentice Hall)

Schorr B 1976 *Preprint CERN/76/3*

Tour J T and Gonzalez R C 1974 *Pattern Recognition Principles* (Reading, MA: Addison-Wesley)

Ullmann J R 1973 *Pattern Recognition Techniques* (London: Butterworth)

Weinberg G M 1972 *The Psychology of Computer Programming* (New York: Van Nostrand Reinhold)

Wind H 1974 *Nucl. Instrum. Methods* **115** 431–4

Young T Y and Calvert T W 1974 *Classification, Estimation and Pattern Recognition* (Amsterdam: Elsevier)

Zahn C T 1974 *Proc. Int. Computing Symp., Davos, 1973* (Amsterdam: North-Holland) pp 381–7