Chapter 13
Pattern Recognition and Reconstruction

R. Frühwirth, E. Brondolin, and A. Strandlie

13.1 Track Reconstruction

13.1.1 Introduction

Track reconstruction is the task of finding and estimating the trajectory of a charged particle, usually embedded in a static magnetic field to determine its momentum and charge. It involves pattern recognition algorithms and statistical estimation methods. Depending on the physics goals, not all charged tracks have to be reconstructed. For instance, in many cases there is a physically motivated lower limit on the momentum or transverse momentum of the particles to be found. Other examples are short-range secondary particles, such as δ-electrons, that normally need not be reconstructed. It may also be useful to reconstruct electron-positron pairs from photon conversions in order to check the distribution of material in the detector. Track reconstruction frequently proceeds in several steps:

1. Pattern recognition or Track finding: Finds the detector signals (hits) that are generated by the same charged particle.
2. Track fitting: Estimates for each track candidate the track parameters and the associated covariance matrix.
3. Test of track hypothesis: Tests for each track candidate whether all hits do indeed belong to the track and identifies outliers.

R. Frühwirth (✉) · E. Brondolin
Institute of High Energy Physics of the Austrian Academy of Sciences, Vienna, Austria
e-mail: Rudolf.Fruehwirth@oeaw.ac.at; erica.brondolin@cern.ch

A. Strandlie
NTNU—Norwegian University of Science and Technology, Gjøvik, Norway
e-mail: are.strandlie@ntnu.no

There are many different algorithms for track finding. A selection of them is described in Sects. 13.1.2.1 and 13.1.2.2. For an extended treatment of the subject, containing many examples, see the excellent exposition in [1]. The track fit takes a track candidate and estimates the track parameters (location, direction, momentum or curvature, see Sect. 13.1.3.2) from the detector hits (Sect. 13.1.3.4), taking into account the equation of motion (Sect. 13.1.3.2) in the magnetic field (Sect. 13.1.3.1) and the effects of the detector material on the trajectory (Sect. 13.1.3.3). In the test stage (Sect. 13.1.3.5) outliers are identified, i.e., hits which apparently do not belong to the track. If outliers are expected, the estimation procedure should be robust so that the estimated track is not significantly biased by the outliers. Some robust methods are discussed in Sect. 13.1.3.5. Section 13.1.4 treats track-based alignment, and Sect. 13.1.5 contains many useful formulas for determining the approximate momentum resolution of a tracking detector without extensive simulations.

13.1.2 Pattern Recognition

Pattern recognition or track finding methods can be divided into global and local methods. In a global method, all detector hits are treated on an equal footing, and all track candidates are found in parallel; in a local method, there is a privileged subset of hits which is used to find initial track candidates, which are then completed to full track candidates.

13.1.2.1 Global Methods

Typical global methods of track finding find the tracks in parallel, for instance by identifying peaks in a one- or two-dimensional histogram, or by observing the final state of a recurrent neural network.

Conformal Mapping

A popular method for finding circular particle tracks is the conformal mapping method [2]. It uses the fact that the mapping

$$u = \frac{x}{x^2 + y^2}, \quad v = \frac{y}{x^2 + y^2},$$

transforms circles going through the origin of an $x-y$ coordinate system into straight lines of the form
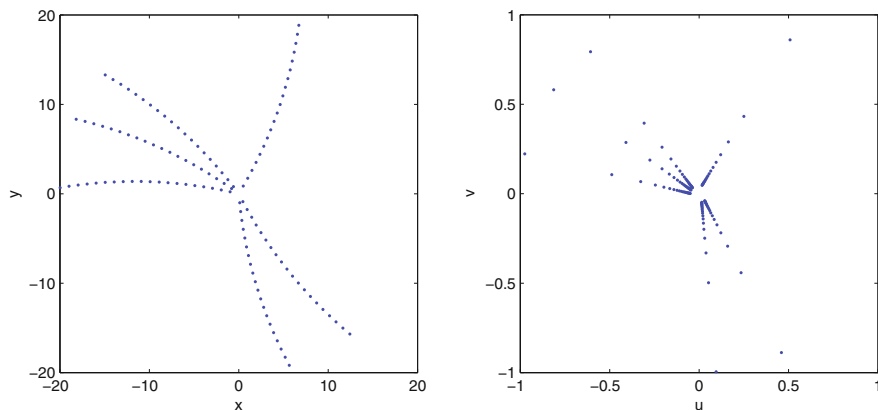
$$v = \frac{1}{2b} - u\frac{a}{b},$$

Fig. 13.1 The original measurements (left) and the transformed measurements (right) of six circular tracks

where the parameters a and b are defined by the circle equation

$$(x - a)^2 + (y - b)^2 = R^2 = a^2 + b^2.$$

The distance of the line to the origin is equal to $1/(2R)$, so that for large radius R it passes very close to the origin. The lines can be found by a histogramming method. After transforming the measurements in the u–v plane to polar coordinates and collecting the polar angle θ in a histogram, measurements belonging to the same particle will tend to create peaks in the histogram.

As an example, the measured points of six circular tracks are shown in the left hand panel of Fig. 13.1, while the transformed measurements are shown in the right hand panel of Fig. 13.1. The resulting histogram of the polar angle θ is shown in Fig. 13.2.

Hough Transform

In the general case of lines not passing close to the origin, a more general approach is needed in order to find the lines. A very popular method for this purpose is the Hough transform [3]. The principle of the Hough transform can be explained by noting that a straight line in an x–y coordinate system, $y = cx + d$, can also be regarded as a straight line in a c–d coordinate system by the transformation $d = -xc + y$. For a fixed point (x, y), the line in c–d space (also denoted parameter space) corresponds to all possible lines going through this point in x–y space (also denoted image space). Measurements lying along a straight line in image space therefore transform into lines in parameter space which cross at the specific value of the parameters of the line under consideration in image space. In practice, parameter space is discretized, and each measurement (x, y) leads to
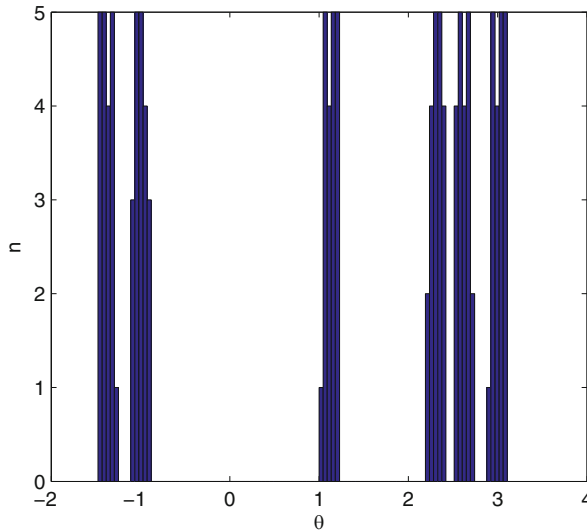
Fig. 13.2 Histogram of $\theta = \arctan(v/u)$

an increment of a set of histogram bins. Measurements lying along straight lines tend to create peaks in the histogram, and the lines can be found by searching for peaks in this histogram. The granularity of the discretization has to be optimized for each specific application, as it depends on the amount of noise present and the actual values of measurement uncertainties. A too fine-grained histogram can split or destroy peaks if the measurement uncertainties are non-negligible. On the other hand, a too course-grained histogram increases the sensitivity to noise, and nearby tracks may merge into a single peak.

The basic formulation of the Hough transform is an example of a divergent transform, i.e., one measurement in image space corresponds to a set of increments of histogram entries in parameter space. The Hough transform can also be made convergent by considering instead a pair of measurements in image space. A unique line passes through any such pair, and only one entry in the parameter space histogram needs to be incremented. A possible disadvantage of such an approach is that the number of pairs grows quadratically with the number of measurements in image space. In order to reduce computational complexity, one may consider only a randomly selected subset of all the pairs. This is the basic feature of probabilistic Hough transforms [4].

The Hough transform has turned out to be successful also for finding circles passing through the origin. With this constraint, two parameters are enough to uniquely describe the circle, and the task again amounts to finding peaks in a two-dimensional histogram. With three or more parameters, one has to search for clusters in multi-dimensional spaces, and in this case the Hough transform is in general less powerful than in the two-dimensional case.

For track finding in drift tubes, with their inherent left-right ambiguity, the drift circles can be transformed to sine curves in the (r, θ) space by applying a Legendre transform [5]. The peaks at the intersections of several sine curves represent the common tangents to a set of several circles.

Neural Networks

Recurrent neural networks of the Hopfield type [6] are used in finding solutions to certain kinds of combinatorial optimization problems, i.e., problems that can be formulated as finding the minimum of an energy function

$$E = -\frac{1}{2} \sum_i \sum_j T_{ij} S_i S_j$$

with respect to the configuration of n binary-valued neurons $S_i, i = 1, \ldots, n$ and fixed connection weights $T_{ij}, i, j = 1, \ldots, n$. It was realized independently in [7] and [8] that the track finding problem can be formulated as a minimization problem of this kind. The neurons are links between measurements which potentially belong to the same track. The connection weights T_{ij} have a structure which favors links sharing a measurement and pointing in a similar direction. The standard network dynamics leads to a solution corresponding to a local minimum of the energy function. A better solution is to apply a mean-field annealing technique [9], which introduces a temperature parameter and thereby allows the neurons to take all values in the interval between the two original binary values. The network is initialized at a high temperature, the mean-field equations are iteratively solved as the network is cooled down, and the low-temperature limit is taken in the end. At a significantly lower computational effort, the approximate solutions found by the mean-field technique have been shown to be very close to the exact solutions [10]. For applications of the Hopfield network in experiments see e.g. [11–15].

The energy function of the Hopfield network can be generalized in order to take into account the track model (see Sect. 13.1.3.2), i.e., the known parametric form of the tracks. The resulting algorithm is called elastic tracking or elastic arms [16–19]. A related generalization is the elastic net, originally used to tackle the traveling salesman problem [20]. Applications to track finding are described in [21] and [22, 23].

13.1.2.2   Local Methods

A local track finding method finds the tracks sequentially, starting from an initial track segment or an initial collection of measured points.

Track Road

The track road method starts out with a set of measurements that potentially belong to the same track, typically one close to the vertex area, one far out in the tracking detector, and one in the middle. The track model can then be used, either exactly or approximately, if speed is an important issue, to interpolate between the measurements and create a road around the hypothesized track. Measurements inside the road are then collected. The number of measurements and the quality of the subsequent track fit are used to determine whether the track candidate should be kept or discarded.

Track Following

A track following procedure takes a track seed as a starting point. A seed is often a short track segment, potentially including a constraint of the position of the vertex region. Seeds can be generated at the inner part of the tracking detector, where the measurements frequently are of very high precision, or at the outer part, where the track density is lower. From the seed, the track is extrapolated to the next detector unit. As for the track roads method, this can be done either with the full track model or with an approximate, simplified model. The measurement closest to the predicted track is included in the track candidate, and the track is extrapolated again.

Kalman Filter

The Kalman filter [24–26] can be regarded as a statistically optimal track following procedure. It works by alternating prediction and update steps. Starting from the seed, the track parameters and their covariance matrix are extrapolated to the next detector unit containing a measurement, using the full track model. If the measurement is compatible with the prediction, it is included in the track candidate, and the track parameters and their covariance matrix are updated with the information from the measurement. The procedure is repeated until too many detector units without compatible measurements are traversed or the end of the tracking detector is reached.

In the original formulation of the method, the measurement closest to the predicted track is included in the track candidate [27]. However, if the density of measurements is high, the closest measurement might originate from another particle or from noise in the detector electronics. Including the wrong measurement could therefore lead to a wrong subsequent prediction and ultimately to the loss of the track. The currently most popular approach, the combinatorial Kalman filter, avoids such losses by splitting the track candidate into several branches when several compatible measurements are found after the prediction [28]. In order to take into account detector inefficiencies an additional branch with a missing hit can be generated.
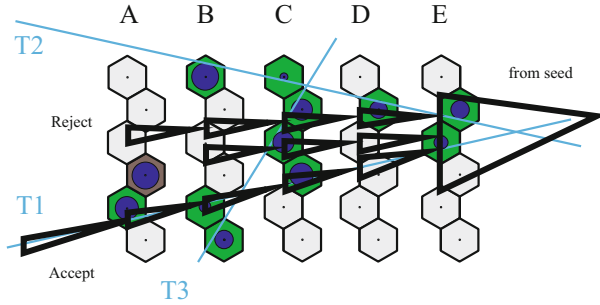
Fig. 13.3 An example of the combinatorial Kalman filter (reprinted from R. Mankel [28], with permission from Elsevier)

All branches are extrapolated to the next detector layer containing compatible measurements. A branch is split again if several measurements are compatible with the branch prediction. Branches are removed if too many detector units without compatible measurements are traversed or if the quality of the track candidate, in terms of the value of a χ^2 statistic, is too low. If there are several surviving candidates after the end of the detector has been reached, the candidate with most measurements and the lowest value of the χ^2 statistic is kept and regarded as the final track candidate. An example is shown in Fig. 13.3.

A similar track finding method has been formulated in the language of cellular automata [23, 29]. The combinatorial problem can also be solved by using generalized, adaptive versions of the Kalman filter [30–32].

13.1.3 Estimation of Track Parameters

13.1.3.1 Magnetic Field Representation

The presence of a magnetic field in a tracking detector causes a bending of the trajectory of a charged particle, and, hence, allows a measurement of the particle momentum. A precise knowledge of the magnetic field is therefore crucial for accurate estimates of the particle momenta.

The magnetic field can be calculated by solving Maxwell's equations, knowing the detailed configuration of the current sources and the magnetic materials in the detector volume. In the general case, a numerical solution of these equations in terms of a finite-element analysis is needed. In special cases, the field can be found by less general approaches. The simplest situation is a solenoidal magnet, providing a homogeneous field in a large volume. Also, it is known that the field inside a volume with no magnetic material can be determined by knowledge of the field on the volume boundary only [33]. Measurements of the field on the volume boundary allows an estimation of coefficients of polynomials obeying Maxwell's

equations. Field measurements inside the volume are used to evaluate the quality of the calculated field. If the measurements inside the volume are precise enough, they can be used to further refine the knowledge of the field by being included in the estimation procedure of the abovementioned coefficients [34].

In a track reconstruction application, fast access to the value of the magnetic field at any point inside the detector volume is crucial. For this purpose, a numerical representation of the field is needed. A frequently used approach is to create a table of the magnetic field values at a grid of points and to determine the field at points between the grid nodes by linear or quadratic interpolation. An alternative approach is to divide the detector volume into several sub-volumes and to fit the coefficients of low-order polynomials to the known field values inside each sub-volume [35, 36]. If the number of sub-volumes is large, potentially many coefficients have to be determined. On the other hand, once the coefficients are determined, the field access is very fast. Also, the derivatives of the field, which are needed by some track reconstruction algorithms, can be computed as fast as the field itself.

13.1.3.2   Track Models

Consider a charged particle with mass m and charge $Q = qe$, e being the elementary charge. Its trajectory $\boldsymbol{x}(t)$ in a magnetic field $\boldsymbol{B}(\boldsymbol{x})$ is determined by the equations of motion given by the Lorentz force $\boldsymbol{F} \propto q\boldsymbol{v} \times \boldsymbol{B}$, where $\boldsymbol{v} = \mathrm{d}\boldsymbol{x}/\mathrm{d}t$ is the velocity of the particle. In vacuum, Newton's second law reads [37]

$$\frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = kq\boldsymbol{v}(t) \times \boldsymbol{B}(\boldsymbol{x}(t)), \tag{13.1}$$

where $\boldsymbol{p} = \gamma m\boldsymbol{v}$ is the momentum of the particle, $\gamma = (1 - \boldsymbol{v}^2/c^2)^{-1/2}$ is the Lorentz factor, and k is a unit-dependent proportionality factor. If $\boldsymbol{p}$ is in GeV/c, $\boldsymbol{x}$ is in meters, and $\boldsymbol{B}$ is in Tesla, $k = 0.29979$ GeV/c T^{-1} m^{-1}. The trajectory is uniquely defined by the initial conditions, the six degrees of freedom specified for instance by the initial position and the initial velocity. If these are tied to a surface, five degrees of freedom are necessary and sufficient. Geometrical quantities other than position and velocity can also be used to specify the initial conditions. The collection $\boldsymbol{q}$ of these quantities is called the initial track parameters or the initial state vector.

Equation (13.1) can be written in terms of the path length $s(t)$ along the trajectory instead of t, giving [37]

$$\frac{\mathrm{d}^2\boldsymbol{x}}{\mathrm{d}s^2} = \frac{kq}{|\boldsymbol{p}|} \cdot \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}s} \times \boldsymbol{B}(\boldsymbol{x}(s)) = F(s, \boldsymbol{x}(s), \dot{\boldsymbol{x}}(s)). \tag{13.2}$$

In simple situations this equation has analytical solutions. In a homogeneous magnetic field the trajectory is a helix; it reduces to a straight line in the limit of a vanishing field. In the general case of an inhomogeneous field, numerical

methods can be used, such as Runge–Kutta integration of the equations of motion or parametrization by polynomials or splines [37]. Among Runge–Kutta methods, the Runge-Kutta-Nyström algorithm is specially designed for second-order equations such as Eq. (13.2). In the fourth-order version a step of length h, starting at $s = s_n$, is computed by [37]

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + h\dot{\boldsymbol{x}}_n + h^2(k_1 + k_2 + k_3)/6, \quad \dot{\boldsymbol{x}}_{n+1} = \dot{\boldsymbol{x}}_n + h(k_1 + 2k_2 + 2k_3 + k_4)/6,$$

with

$$
\begin{aligned}
k_1 &= F(s_n, \boldsymbol{x}_n, \dot{\boldsymbol{x}}_n), \\
k_2 &= F(s_n + h/2, \boldsymbol{x}_n + h\dot{\boldsymbol{x}}_n/2 + h^2 k_1/8, \dot{\boldsymbol{x}}_n + hk_1/2), \\
k_3 &= F(s_n + h/2, \boldsymbol{x}_n + h\dot{\boldsymbol{x}}_n/2 + h^2 k_1/8, \dot{\boldsymbol{x}}_n + hk_2/2), \\
k_4 &= F(s_n + h, \boldsymbol{x}_n + h\dot{\boldsymbol{x}}_n + h^2 k_3/2, \dot{\boldsymbol{x}}_n + hk_3),
\end{aligned}
$$

where $\boldsymbol{x}_n$ is the position of the particle at $s = s_n$ and $\dot{\boldsymbol{x}}_n$ is the unit tangent vector. The magnetic field needs to be looked up for the calculation of k_2, k_3 and k_4, i.e., three times per step. If the field at the final position $\boldsymbol{x}_{n+1}$, which is the starting position of the next step, is approximated by the field used for k_4, only two lookups are required per step. If the field is (almost) homogeneous, as for example in a solenoid, the step size h can be chosen to be constant; otherwise a variable step size is more efficient. The step size can be optimized using an adaptive version of the Runge-Kutta-Nyström algorithm [38]. Note that the error of a step of length h may be larger than $O(h^5)$ if the magnetic field does not have smooth derivatives, as is the case if it is computed by linear interpolation. If the field is represented by low-order polynomials in sub-volumes, Runge–Kutta steps should terminate at volume boundaries.

Different detector geometries often lead to different choices of the parametrization. However, the parametrization of the trajectory should comply to some basic requirements: the parameters should be continuous with respect to small changes of the trajectory; the choice of track parameters should facilitate the local expansion of the track model into a linear function; and the uncertainties of the estimated values of the parameters should follow a Gaussian distribution as closely as possible. For example, curvature should be used rather than radius of curvature, and inverse (transverse) momentum rather than (transverse) momentum.

The track model, given by the solution of the equations of motion, describes how the state vector $\boldsymbol{q}_k$ at a given surface k depends on the state vector at a different surface i:

$$\boldsymbol{q}_k = \boldsymbol{f}_{k|i}(\boldsymbol{q}_i),$$

where $\boldsymbol{f}_{k|i}$ is the track propagator from surface i to surface k. When analytical solutions of the equations of motion exist, the track propagator is also analytical. Even in

a homogeneous magnetic field, the path length can be determined analytically only for propagation to cylinders with symmetry axis parallel to the field direction or to planes orthogonal to the field direction. Otherwise, a Newton iteration or a parabolic approximation has to be used to find the path length.

For track reconstruction purposes, the covariance matrix of the estimated track parameters needs to be propagated along with the track parameters themselves. The track propagator is often a non-linear function of the track parameters at the initial surface, but the covariance matrix has to be transported under the assumption of a linear track model. This procedure, called linear error propagation, is based on a Taylor expansion of the track propagator, keeping only first-order terms. These first-order terms, defining the Jacobians of the track model, are given by

$$F_{k|i} = \left. \frac{\partial q_k}{\partial q_i} \right|_{\breve{q}_i},$$

where $\breve{q}_i$ is the expansion point in surface i. For analytical track models, the Jacobian is also analytical. If, for example, the magnetic field is homogeneous, the general case of propagation to a plane of arbitrary spatial orientation uses a curvilinear coordinate frame moving along with the trajectory as a means of deriving the required Jacobians [39].

In the general case of a non-analytical track model, the Jacobians cannot be computed analytically either. The most straightforward approach is to calculate the relevant derivatives in a purely numerical way. The basis for these calculations is a reference trajectory corresponding to the expansion point. In addition, five other trajectories are created, corresponding to small variations in each of the track parameters. By propagating these five trajectories to the destination surface, numerical derivatives can be obtained. A potential disadvantage of such an approach is its computational complexity, as six trajectories have to be propagated instead of a single one. Much less computational load is introduced by transporting the Jacobian terms in parallel to the track parameters during the Runge–Kutta integration [40, 41], avoiding the need for propagating auxiliary trajectories.

The measurement model describes the functional dependence of the measured quantities on the state vector at a detector surface k:

$$m_k = h_k(q_k).$$

The vector of measurements m_k usually contains the measured coordinates, but may contain also other quantities, e.g. measurements of direction or even momentum. In a pixel detector or in a double-sided silicon strip detector, m_k is two-dimensional; in a one-sided strip detector, it is one-dimensional. In a drift chamber or a multi-wire proportional chamber with several layers, the measurement may be a track segment resulting from an internal track reconstruction. In this case the vector m_k may be four- or five-dimensional, depending on whether the curvature can be estimated or not.

In most cases the function $\boldsymbol{h}_k(\boldsymbol{q}_k)$ includes a transformation of the state vector $\boldsymbol{q}_k$ into the local coordinate system of the detector. For use in track reconstruction, the Jacobian of this transformation is needed:

$$\boldsymbol{H}_k = \left. \frac{\partial \boldsymbol{m}_k}{\partial \boldsymbol{q}_k} \right|_{\breve{\boldsymbol{q}}_k} , \tag{13.3}$$

where $\breve{\boldsymbol{q}}_k$ is the expansion point in surface k. In many cases the Jacobian contains only rotations and projections, and thus can be computed analytically.

The measurement is always smeared by a measurement error:

$$\boldsymbol{m}_k = \boldsymbol{h}_k(\boldsymbol{q}_k) + \boldsymbol{\varepsilon}_k.$$

The mean value and the covariance matrix of $\boldsymbol{\varepsilon}_k$ depend on the detector type and the detector geometry and have therefore in general to be calibrated for each detector unit independently. The measurement error is often assumed to follow a Gaussian distribution, but frequently exhibits tails which are incompatible with this assumption. In this case a Gaussian mixture is a more appropriate model.

13.1.3.3   Material Effects

A charged particle crossing a tracking detector interacts with the material of the detector. The most important types of interactions in track reconstruction are multiple Coulomb scattering, energy loss by ionization, and energy loss by bremsstrahlung. For an in-depth treatment of material effects see Chapter 2.

Multiple Coulomb Scattering

Elastic Coulomb scattering of particles heavier than the electron is dominated by the atomic nucleus. For small angles the differential cross-section is approximately equal to

$$\frac{\mathrm{d}\sigma}{\mathrm{d}\theta} = 2\pi \left(\frac{2Ze^2}{pv} \right)^2 \frac{1}{\theta^3},$$

where θ is the polar angle of the scattering, Z is the charge of the nucleus in units of the elementary charge e, v is the velocity of the scattered particle, and p is its momentum [42]. Because of screening effects and the finite size of the nucleus the differential cross-section is modified to [43]

$$\frac{\mathrm{d}\sigma}{\mathrm{d}\theta} = 2\pi \left(\frac{2Ze^2}{pv} \right)^2 \frac{\theta}{(\theta^2 + \theta_{\min}^2)^2}, \quad 0 \le \theta \le \theta_{\max}.$$

If the momentum p is given in GeV/c, the lower and upper limits are approximately equal to

$$\theta_{\min} \approx \frac{2.66 \cdot 10^{-6} Z^{1/3}}{p}, \quad \theta_{\max} \approx \frac{0.14}{A^{1/3} p}.$$

The average number of scattering processes in a layer of thickness d (in cm) is given by

$$N(d) = d\sigma \frac{N_A \rho}{A},$$

where σ is the integrated elastic cross section, N_A is the Avogadro constant, ρ is the density of the material (in g/cm^3), and A is the atomic mass of the nucleus. In track reconstruction it is convenient to work with the projected scattering angles in two perpendicular planes. The projected multiple scattering angle θ_P is equal to the sum of the projected single scattering angles, and its variance can be obtained by multiplying the variance of the projected single scattering angle by the average number of scatters, the projected single scattering angles being uncorrelated. With increasing thickness d the distribution of the projected scattering angle approaches a normal distribution, and the two projected angles become independent. For thin scatterers, however, the width of the Gaussian core is notably narrower than is indicated by the variance [42]. This is taken into account by Highland's formula for the standard deviation of the projected scattering angle [44]:

$$\sigma_P = E(\theta_P^2)^{1/2} = \frac{0.0136}{\beta p} \sqrt{d/X_0} \, [1 + 0.038 \ln(d/X_0)],$$

where X_0 is the radiation length of the material in cm, $\beta = v/c$ is the particle velocity in units of c, and p is the particle momentum in GeV/c. The logarithmic correction ceases to be applicable above $d \approx X_0$.

If a scatterer is sufficiently thin, the transverse offset of the track due to multiple scattering can be neglected. Only the track direction is affected in this case. If the direction is represented by the polar angle θ and the azimuthal angle φ, their joint covariance matrix is given by

$$\text{var}(\Delta\theta) = \sigma_P^2, \quad \text{var}(\Delta\varphi) = \sigma_P^2/\sin^2\theta, \quad \text{cov}(\Delta\theta, \Delta\varphi) = 0.$$

If the direction is represented by the direction tangents $t_x = dx/dz$ and $t_y = dy/dz$, the covariance matrix is [45]

$$\text{Var}[(\Delta t_x, \Delta t_y)^T] = \sigma_P^2 (1 + t_x^2 + t_y^2) \begin{pmatrix} 1 + t_x^2 & t_x t_y \\ t_x t_y & 1 + t_y^2 \end{pmatrix}.$$

If the direction is represented by the direction cosines $c_x = \mathrm{d}x/\mathrm{d}s$ and $c_y = \mathrm{d}y/\mathrm{d}s$, the covariance matrix is [45]

$$\mathrm{Var}[(\Delta c_x, \Delta c_y)^{\mathsf{T}}] = \sigma_P^2 \begin{pmatrix} (1-c_x)^2 & -c_x c_y \\ -c_x c_y & (1-c_y)^2 \end{pmatrix}.$$

In all cases the projected variance σ_P^2 takes into account the effective amount of material crossed by the track.

If the transverse offset cannot be neglected, its variance and its correlation with the angle have to be taken into account. Assume that the particle passes a scatterer of length d, traveling along the z-axis. Neglecting the curvature of the track, the joint covariance matrix of the offset Δx and the scattering angle θ_x in the x–z projection is

$$\mathrm{var}(\Delta x) = \sigma_0^2 d^3/3, \quad \mathrm{var}(\theta_x) = \sigma_0^2 d, \quad \mathrm{cov}(\Delta x, \theta_x) = \sigma_0^2 d^2/2,$$

where σ_0^2 is the variance of the projected scattering angle per unit length. If the particle enters the scatterer at $z = 0$ with direction (t_x, t_y), the joint covariance matrix of the offsets Δx, Δy and the angles Δt_x, Δt_y at $z = d$ is

$$\mathrm{Var}\left[\begin{pmatrix} \Delta x \\ \Delta y \\ \Delta t_x \\ \Delta t_y \end{pmatrix}\right] = \sigma_0^2(1+t_x^2+t_y^2) \begin{pmatrix} (1+t_x^2)D^3/3 & t_x t_y D^3/3 & (1+t_x^2)D^2/2 & t_x t_y D^2/2 \\ t_x t_y D^3/3 & (1+t_y^2)D^3/3 & t_x t_y D^2/2 & (1+t_y^2)D^2/2 \\ (1+t_x^2)D^2/2 & t_x t_y D^2/2 & (1+t_x^2)D & t_x t_y D \\ t_x t_y D^2/2 & (1+t_y^2)D^2/2 & t_x t_y D & (1+t_y^2)D \end{pmatrix},$$

where $D = (1 + t_x^2 + t_y^2)^{1/2}d$ is the effective thickness crossed. If the direction is represented by θ and φ, the covariance matrix can be computed via the transformation

$$t_x = \tan(\theta)\cos(\varphi), \quad t_y = \tan(\theta)\sin(\varphi),$$

and linear error propagation with the Jacobian

$$\boldsymbol{T} = \frac{\partial(\theta, \varphi)}{\partial(t_x, t_y)} = \begin{pmatrix} \dfrac{t_x}{\sqrt{t_x^2+t_y^2}(1+t_x^2+t_y^2)} & \dfrac{t_y}{\sqrt{t_x^2+t_y^2}(1+t_x^2+t_y^2)} \\ -\dfrac{t_y}{t_x^2+t_y^2} & \dfrac{t_x}{t_x^2+t_y^2} \end{pmatrix} = \begin{pmatrix} \dfrac{\cos(\varphi)}{1+\tan^2(\theta)} & \dfrac{\sin(\varphi)}{1+\tan^2(\theta)} \\ -\dfrac{\sin(\varphi)}{\tan(\theta)} & \dfrac{\cos(\varphi)}{\tan(\theta)} \end{pmatrix}.$$

For analogous formulas in cylindrical coordinates, see [46].

If the curvature of the track cannot be neglected, the simplest approach is a stepwise integration of the equation of motion, assuming the validity of a helical track model within each step and considering each such step as a thin scatterer [39, 47].

Energy Loss

For particles other than electrons the energy loss in material is almost exclusively due to scattering on electrons. The momentum correction Δp in a material layer of thickness d is calculated by integrating the Bethe-Bloch formula [37]:

$$\Delta p = \int_0^d \frac{\mathrm{d}p}{\mathrm{d}x}\mathrm{d}x = \int_0^d \frac{1}{\beta}\frac{\mathrm{d}E}{\mathrm{d}x}\mathrm{d}x = \int_0^d \frac{K}{\beta^3}\left[\ln\frac{2m_e c^2 \beta^2 \gamma^2}{\langle I \rangle} - \beta^2\right]\mathrm{d}x, \quad (13.4)$$

where K is a constant depending on the material, m_e is the electron mass, $\langle I \rangle$ is the average ionization potential of the material, and $\beta = v/c$ and $\gamma = E/mc^2$ are the usual kinematic parameters. The ratio $\langle I \rangle/Z$ is about $20\,\mathrm{eV}$ for hydrogen and helium, between 12 and $16\,\mathrm{eV}$ for light nuclei, and around $10\,\mathrm{eV}$ for heavy nuclei [44]. For practical purposes, the differential energy loss $\mathrm{d}E/\mathrm{d}x$ is a function only of β. For small β, it decreases like $1/\beta^2$. It has a minimum, the position of which drops with increasing Z from $\beta\gamma \approx 3.5$ (carbon) to $\beta\gamma \approx 3$ (lead). In terms of momentum, the minimum is at $p = \beta\gamma mc$ and thus depends on the mass of the particle. This dependency is used for particle identification. The energy loss at the minimum can be parameterized for $Z \geq 6$ by [44]:

$$(\mathrm{d}E/\mathrm{d}x)_{\mathrm{min}} = (2.35 - 0.64 \ln_{10} Z)\ \mathrm{MeV\,g^{-1}cm^2}.$$

From this the constant K in Eq. (13.4) can be calculated. For large $\beta\gamma$ the energy loss increases like $\ln(\beta\gamma)$; this is called the relativistic rise. For momenta in the vicinity of the minimum $\mathrm{d}E/\mathrm{d}x$ can be considered as constant, giving $\Delta p \approx (\mathrm{d}E/\mathrm{d}x)_{\mathrm{min}} \cdot d\rho/\beta$, ρ being the density of the material.

Bremsstrahlung

For an electron (or positron) passing through matter the most significant contribution to energy loss is bremsstrahlung, the emission of photons in the electric field of an atomic nucleus. In the Bethe–Heitler model [48] the relative energy loss is distributed independently of the energy. Let d be the path length in the material in units of radiation length, and z the fraction of energy remaining after the material is traversed. Then the distribution of z is given by the following probability density function:

$$f(z) = \frac{(-\ln z)^{c-1}}{\Gamma(c)}, \quad 0 \leq z \leq 1,$$

where $\Gamma(x)$ is Euler's gamma function and $c = d/\ln 2$. For high energy electrons $p \approx E$, so the momentum correction is $\Delta p \approx p(z - 1)$. The first two moments of Δp are

$$E(\Delta p) = p(2^{-c} - 1), \quad \mathrm{var}(\Delta p) = p^2(3^{-c} - 4^{-c}).$$

The moments can be used for a Gaussian representation of bremsstrahlung as an additional process noise in the Kalman filter (see Sect. 13.1.3.4). As this is a very crude approximation, more sophisticated methods have been developed that take into account the actual shape of the distribution. One of them is the Gaussian-sum filter [49, 50], see Sect. 13.1.3.5. A computationally less intensive approach is described in [51].

13.1.3.4   Estimation Methods

The main task of the track fit is to estimate the values of a set of parameters describing the state of a particle somewhere in the detector, often at a reference surface close to the interaction vertex. The information from the measurements created by the particle while traversing the tracking detector should be processed in an optimal manner. If the track model is truly linear, i.e., if the measurements are strictly linear functions of the track parameters, and all stochastic disturbances entering the estimation procedure are Gaussian, the linear least-squares method is the optimal one [37]. Since track parameter propagation in general is a nonlinear procedure, strict linearity holds very rarely in practice. The relation between the track parameter vector q_0 at a reference surface and the measurement vector m_k at a detector layer k is a function d_k given by

$$m_k = d_k(q_0) + \gamma_k,$$

where γ_k is a noise term containing the measurement error of m_k and all multiple scattering in front of m_k. The function d_k is a composition of the measurement model function h_k and the track propagator functions $f_{i|i-1}$ (see Sect. 13.1.3.2):

$$d_k = h_k \circ f_{k|k-1} \circ \cdots \circ f_{2|1} \circ f_{1|0}.$$

For the linear least-squares method d_k has to be linearized around some expansion point, providing the Jacobian D_k of each d_k:

$$D_k = H_k F_{k|k-1} \cdots F_{2|1} F_{1|0},$$

with H_k from Eq. (13.3). The covariance matrix of γ_k is obtained by linear error propagation:

$$\mathrm{var}(\gamma_k) = V_k + H_k(F_{k|1} Q_1 F_{k|1}{}^\mathrm{T} + \cdots + F_{k|k-1} Q_{k-1} F_{k|k-1}{}^\mathrm{T} + Q_k)H_k{}^\mathrm{T},$$

where V_k is the covariance matrix of the measurement error ε_k of m_k, and Q_j is the covariance matrix of multiple scattering after layer $j-1$ up to and including layer j. The part of Q_j originating from scattering between the layers has to be transported to layer j by the appropriate Jacobian. Because of the cumulative effect of multiple

scattering γ_i and γ_k are correlated. If $i < k$, the covariance is given by

$$\mathrm{cov}(\gamma_i, \gamma_k) = H_i (F_{i|1} Q_1 F_{k|1}{}^\mathrm{T} + \cdots + F_{i|i-1} Q_{i-1} F_{k|i-1}{}^\mathrm{T} + Q_i F_{k|i}{}^\mathrm{T}) H_k{}^\mathrm{T}.$$

The observations m_k, the functions d_k, their Jacobians D_k, and the noise γ_k are now collected in single vectors and a matrix:

$$m = \begin{pmatrix} m_1 \\ \cdot \\ \cdot \\ m_n \end{pmatrix}, \quad d = \begin{pmatrix} d_1 \\ \cdot \\ \cdot \\ d_n \end{pmatrix}, \quad D = \begin{pmatrix} D_1 \\ \cdot \\ \cdot \\ D_n \end{pmatrix}, \quad \gamma = \begin{pmatrix} \gamma_1 \\ \cdot \\ \cdot \\ \gamma_n \end{pmatrix},$$

where n is the total number of measurement layers. This gives the following model:

$$m = d(q_0) + \gamma,$$

which now can be linearized into

$$m = D q_0 + c + \gamma,$$

where c is a constant vector. The global least-squares estimate of q_0 is given by

$$\tilde{q}_0 = (D^\mathrm{T} G D)^{-1} D^\mathrm{T} G (m - c),$$

where $V = G^{-1}$ is the non-diagonal covariance matrix of γ. The quality of the initial expansion point can be monitored by using the obtained estimate as a new expansion point, and the state vector estimate can hence be re-calculated. Such a procedure is repeated until convergence, defined by a suitable stopping criterion.

If the track model is a circle and multiple scattering and energy loss can be neglected, the estimation can be simplified substantially. Explicit estimators are given in [52] for the center and radius of the circle, and in [53] for the curvature, the direction and the distance from a fixed point. Other algorithms are based on conformal mapping in the plane [2] or on a mapping to the Riemann sphere [54–56].

If there is strong multiple scattering, the estimated track can be quite far away from the real track. In order to follow the actual track more closely, two projected scattering angles can be explicitly estimated at each detector layer or at a set of virtual breakpoints inside a continuous scatterer [45, 57]. The breakpoint method, also known as General Broken Lines [58], and the global least-squares method are equivalent, as far as the estimate of the state vector q_0 is concerned [59].

If the number of measurements or the number of breakpoints is substantial, the computational cost of these methods can be high due to the necessity of inverting large matrices during the estimation procedure. The Kalman filter, a recursive formulation of the least-squares method, requires the inversion of only

small matrices and exhibits the same attractive feature as the breakpoint method of following the actual track quite closely [26, 60].

As mentioned earlier, the Kalman filter proceeds by alternating prediction and update steps. The prediction step is the propagation of the track parameter vector from one detector layer containing a measurement to the next,

$$\boldsymbol{q}_{k|k-1} = \boldsymbol{f}_{k|k-1}(\boldsymbol{q}_{k-1|k-1}),$$

and the associated covariance matrix,

$$\boldsymbol{C}_{k|k-1} = \boldsymbol{F}_{k|k-1}\boldsymbol{C}_{k-1|k-1}\boldsymbol{F}_{k|k-1}{}^{\mathrm{T}} + \boldsymbol{Q}_k.$$

The update step is the correction of the predicted state vector due to the information from the measurement in layer k:

$$\boldsymbol{q}_{k|k} = \boldsymbol{q}_{k|k-1} + \boldsymbol{K}_k \left[\boldsymbol{m}_k - \boldsymbol{h}_k(\boldsymbol{q}_{k|k-1}) \right],$$

where the gain matrix $\boldsymbol{K}_k$ is given by

$$\boldsymbol{K}_k = \boldsymbol{C}_{k|k-1}\boldsymbol{H}_k{}^{\mathrm{T}} \left(\boldsymbol{V}_k + \boldsymbol{H}_k\boldsymbol{C}_{k|k-1}\boldsymbol{H}_k{}^{\mathrm{T}} \right)^{-1}.$$

The update of the covariance matrix is given by

$$\boldsymbol{C}_{k|k} = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\,\boldsymbol{C}_{k|k-1}.$$

The information filter is a mathematically equivalent, but numerically more stable formulation of the Kalman filter. In the information filter, the update of the state vector reads

$$\boldsymbol{q}_{k|k} = \boldsymbol{C}_{k|k} \left[\left(\boldsymbol{C}_{k|k-1}\right)^{-1} \boldsymbol{q}_{k|k-1} + \boldsymbol{H}_k{}^{\mathrm{T}}\boldsymbol{V}_k^{-1}\boldsymbol{m}_k \right],$$

whereas the update of the covariance matrix is given by

$$\boldsymbol{C}_{k|k} = \left[\left(\boldsymbol{C}_{k|k-1}\right)^{-1} + \boldsymbol{H}_k{}^{\mathrm{T}}\boldsymbol{V}_k^{-1}\boldsymbol{H}_k \right]^{-1}.$$

The implementation of the Kalman filter requires the computation of the Jacobians $\boldsymbol{F}_{k|k-1}$ and $\boldsymbol{H}_k$. A compilation of analytical formulas for two important cases (fixed-target configuration and solenoidal configuration) is given in [61].

Full information of the track parameters at the end of the track is obtained when all n measurements in the track candidate have been processed by the filter. The full information can be propagated back to all previous estimates by another iterative procedure, the Kalman smoother. A step of the smoother from layer $k + 1$ to layer

k is for the state vector

$$\boldsymbol{q}_{k|n} = \boldsymbol{q}_{k|k} + A_k(\boldsymbol{q}_{k+1|n} - \boldsymbol{q}_{k+1|k}),$$

where the smoother gain matrix is given by

$$A_k = \boldsymbol{C}_{k|k}\boldsymbol{F}_{k+1|k}^{\mathrm{T}}(\boldsymbol{C}_{k+1|k})^{-1}.$$

The smoothed covariance matrix is

$$\boldsymbol{C}_{k|n} = \boldsymbol{C}_{k|k} - A_k(\boldsymbol{C}_{k+1|k} - \boldsymbol{C}_{k+1|n})A_k^{\mathrm{T}}.$$

The smoother can also be realized by combining two filters running in opposite directions: a forward filter from $\boldsymbol{m}_1$ to $\boldsymbol{m}_n$ and a backward filter from $\boldsymbol{m}_n$ to $\boldsymbol{m}_1$. The smoothed states are the weighted mean of the predicted states of one filter and the updated states of the other filter. This approach is numerically more stable than the gain matrix formulation of the smoother.

13.1.3.5 Track Quality and Robust Estimation

Robust estimators are insensitive to outliers, i.e., measurements that are biased or do not originate from the particle creating the majority of the hits in a track candidate. Some estimators are inherently robust by construction; other estimators can be made robust by finding and discarding outliers.

In the Kalman filter, the residual of the measurement in layer k with respect to the updated state vector is

$$\boldsymbol{r}_{k|k} = \boldsymbol{m}_k - \boldsymbol{h}_k(\boldsymbol{q}_{k|k}),$$

and the covariance matrix of this residual is

$$\boldsymbol{R}_{k|k} = \boldsymbol{V}_k - \boldsymbol{H}_k\boldsymbol{C}_{k|k}\boldsymbol{H}^{\mathrm{T}}.$$

The chi-square increment in layer k is

$$\chi^2_{k,+} = \boldsymbol{r}_{k|k}^{\mathrm{T}}\boldsymbol{R}_{k|k}^{-1}\boldsymbol{r}_{k|k},$$

and the total chi-square of the track is found by summing up the chi-square increments for all measurements in the track candidate. The total chi-square is used to evaluate the quality of the track candidate. A too large value of this test statistic indicates that one or more of the measurements of the track candidate do not originate from the particle creating the majority of the measurements. Such measurements are called outliers.

An outlier rejection procedure can make use of the chi-squares of the measurements with respect to the smoothed predictions, i.e., a weighted mean of the predicted states of a forward and a backward Kalman filter. The measurement with the largest value of the chi-square is removed, and the total chi-square is again calculated. This procedure is repeated until the value of the total chi-square falls below a defined threshold.

In the presence of a potentially large fraction of outliers in a track candidate, the sequential outlier rejection procedure outlined above might become unstable, because the smoothed predictions may themselves be biased by outliers. An alternative approach is the Gaussian-sum filter [62]. This algorithm is based on the assumption that the probability distribution of the measurement error can be modeled as a two-component Gaussian mixture, where a narrow component represents the hypothesis that the measurement is real and a wider component represents the hypothesis that the measurement is an outlier. It takes the form of a set of Kalman filters running in parallel, each Kalman filter representing a specific hypothesis of a subset of the measurements that should be classified as outliers. A weight attached to each Kalman filter can be interpreted as the probability of correctness of the hypothesis. In the end, the Kalman filter with the largest weight or a weighted mean of the different filters can be taken as the final estimate.

The Gaussian-sum filter can also be used to deal with a mixture model of the process noise, i.e., the stochastic disturbance of the track because of interactions with the detector material [63]. In the case of bremsstrahlung, a successful application to the reconstruction of electrons is described in [50].

For the treatment of outliers, the Gaussian-sum filter has two disadvantages. First, it may create a large number of Kalman filters running in parallel because of poor knowledge of the track parameters in the early stages of the filter, making the approach expensive in terms of computing time. Second, an explicit outlier model is required. A faster and even more robust alternative is the Deterministic Annealing Filter [64]. This filter is an iterated Kalman filter with annealing, which assigns small weights to measurements far away from the track. A temperature parameter is introduced, facilitating convergence to the globally optimal solution. The iterations start at a high temperature, continue with a gradual lowering of the temperature and converge at the nominal value of the temperature. The procedure is easily generalized to the situation of several measurements being present in the same detector layer. In this case the measurements compete for inclusion in the track. As opposed to a standard outlier rejection approach, the assignment of measurements is soft. This means that several measurements in the same detector layer might contribute to the final estimate of the track parameters, each with a weight equal to the assignment probability. A further generalization is the multi-track filter, where several tracks are allowed to compete for compatible hits in all detector layers [65]. For an experimental application, see [66].

13.1.3.6 Jet Reconstruction

Jets are bundles of collimated hadrons, reflecting hard scattering processes at the parton level. In order to carry out detailed comparisons between parton-level predictions and hadron-level observations a well-defined "jet finder" is required. In the jet finding information from both the tracking devices and the calorimeters is used.

Jet finding can be understood as finding clusters in the set of reconstructed tracks, including neutral tracks. As in the case of vertex finding (see Sect. 13.2.2), various types of clustering methods have been proposed and investigated. The performance strongly depends on the underlying physics, and usually a jet finder is optimized for specific physics requirements. For instance, the widely used $k_\perp$ clustering algorithm comes in several versions, for instance one for e^+e^- collisions [67], and one for hadron-hadron collisions [68].

Hierarchical cluster algorithms offer a large variety of jet finders, differing mainly by the definition of the measure of distance between objects (tracks and jets), but sometimes also by the order in which the objects are combined. Some examples of agglomerative clustering algorithms are described and studied in [69]. Table 13.1 gives a summary of the distance measures used. The names refer to the ones used in [69]. E_i is the energy of cluster i, p_i is its momentum, θ_{ij} is the opening angle between the momentum vectors of the two clusters, and E_{vis} is the visible energy.

A divisive hierarchical clustering algorithm is described in [76]. It is based on the following measure of distance between two tracks:

$$d_{ij} = \frac{\theta_{ij}^2}{p_i p_j},$$

Table 13.1 Some distance measures used for agglomerative jet finding with respective references

Name	Distance d_{ij}	References
Jade	$\dfrac{2\,E_i E_j (1 - \cos\theta_{ij})}{E_{\text{vis}}^2}$	[70]
Durham, $k_\perp$	$\dfrac{2\,\min(E_i^2, E_j^2)(1 - \cos\theta_{ij})}{E_{\text{vis}}^2}$	[67, 68, 71–73]
Luclus	$\dfrac{2\,p_i^2 p_j^2 (1 - \cos\theta_{ij})}{(p_i + p_j)^2 E_{\text{vis}}^2}$	[74]
Geneva	$\dfrac{8}{9}\dfrac{E_i E_j (1 - \cos\theta_{ij})}{(E_i + E_j)^2}$	[75]
Cambridge	$\dfrac{2\,\min(E_i^2, E_j^2)(1 - \cos\theta_{ij})}{E_{\text{vis}}^2}$	[71]

but can be generalized to any other measure of distance. The method first constructs a minimum spanning tree [77] in the edge-weighted graph connecting all particles with each other and then proceeds to cut the tree along its longest edges. The procedure stops when the longest remaining edge is shorter than a fixed multiple of the median of all edge lengths.

Several non-hierarchical cluster algorithms have been proposed as well. Some of them employ general unsupervised learning methods, such as deterministic annealing [78] or k-means [79]. Others are specially designed for jet finding, for instance the cone algorithm described in [72]. It is an iterating procedure which constructs jets out of seeds. In contrast to the hierarchical clustering method the jets may overlap and a unique assignment has to be forced at the end. A modified cone algorithm suitable for the much larger multiplicity of heavy-ion collisions is proposed in [80]. A specialized jet finder for the reconstruction of hadronic τ-decays is described in [81].

13.1.4   Detector Alignment[1]

Alignment is the general term used in experimental high energy physics to refer to the process of obtaining and applying corrections to the nominal setup of a given experiment. These corrections are typically related to geometrical displacements of devices with a spatial resolution, in contrast to calibrations, where the corrections are usually extracted from pedestal or reference measurements to compensate for offsets in scalar measurements. Misalignment compromises tracking and vertex finding [82] and thus directly affects physics measurements such as momentum and invariant mass resolutions, or the efficiency of b-tagging algorithms. There are various possibilities for the treatment of alignment corrections, ranging from simple translations and rotations, equivalent to those of a rigid body, to more complex deformations, like sags or twists.

To this end experiments typically use several independent strategies [83]. For testing the long-term stability or the alignment of sub-detectors with respect to each other, very often so-called hardware alignment is utilized, where special reference markers are measured directly e.g. via optical systems or photogrammetry. However, these techniques reach only a limited precision in the range of several tens to hundreds of microns. If the intrinsic resolution of a tracking device is smaller, an improved resolution can only be obtained with track-based alignment, where the information from recorded particle tracks is used to obtain the alignment parameters [83, 84]. For various examples of the track-based alignment methods used in experiments since the LEP era, see [85–100].

---

[1]The section on detector alignment was contributed by E. Widl (Institute of High Energy Physics, Vienna; now at Austrian Institute of Technology).

13.1.4.1 General Overview

The basis of all track-based alignment algorithms is an extended track model d, where the measurements m depend not only on the true track-parameters q_0, but also on a set of alignment parameters p_0 that describe the effects of sufficiently small deviations from the ideal geometry:

$$m = d(q_0, p_0) + \gamma, \quad \text{cov}(\gamma) = V.$$

The stochastic term γ, which describes the intrinsic resolution of the tracking devices and the effects of multiple scattering, is dealt with via its covariance matrix V. Since typically high momentum particles are used, energy-loss effects can be assumed to be deterministic and hence directly taken care of in the track model d itself.

With an initial guess $\breve{q}$ for the track parameters and $\breve{p}$ for the alignment parameters, this model allows to define residuals that are functions of the unknowns q and p:

$$r(q, p) = m - d(q, p) \approx m - \breve{d} - D_q \Delta q - D_p \Delta p \tag{13.5}$$

with

$$\breve{d} = d(\breve{q}, \breve{p}), \quad \Delta q = q - \breve{q}, \quad \Delta p = p - \breve{p},$$

$$D_q = \left. \frac{\partial d}{\partial q} \right|_{\breve{q}, \breve{p}}, \quad D_p = \left. \frac{\partial d}{\partial p} \right|_{\breve{q}, \breve{p}}.$$

The goal of a track-based alignment algorithm is to determine p from the residuals r, by minimizing the quadratic form $\chi^2 = r^T V^{-1} r$, using a sufficiently large set of recorded tracks. The methods used are quite diverse, but can be grouped into two categories: biased and unbiased algorithms.

Biased algorithms initially ignore the fact that the initial guess of the track parameters q_0 is in general biased by the factual misalignment. In other words, by setting $q = \breve{q}$ for every track, the residuals become a function of p alone, i.e., $r(q, p) \rightarrow r(p)$. In general, the influence of the biased track information has to be compensated by iterating several times over the track sample, where at each iteration step the previously determined parameters are applied to the track reconstruction.

Unbiased algorithms on the other hand, minimize the residuals or the normalized residuals, respectively, estimating at the same time the track parameters. The problem with such an approach is the resulting huge number of parameters. In the presence of N alignment parameters and a sample of M tracks with m track parameters each, a total of $N + m \cdot M$ parameters have to be dealt with. While the value of N depends on the experimental setup, and m usually equals 5, the number of tracks M has always to be of considerable size to acquire reasonable statistics.

On the other hand, unbiased algorithms usually do not require iterations, with the possible exception of problems like non-linearities or rejection of outliers.

Besides the differences between various algorithms it should be noted that the final result of any track-based alignment is always limited by the tracks used. Basic quality cuts, like the selection of high momentum tracks to minimize the influence of multiple scattering or cuts on the minimum number of hits, have a strong influence on the convergence. More subtle is the effect of an unbalanced mixture of tracks or the complete absence of some types of tracks, such as tracks from collisions and cosmic events or tracks taken with and without a magnetic field. This is due to the fact that any kind of tracks has several unconstrained degrees of freedom, usually referred to as weak modes, weakly defined modes or χ^2-invariant modes. As an example, typical weak modes for straight tracks are shears but not bends, and vice versa for curved tracks. Combining the information of both kinds of tracks is therefore a reasonable strategy to avoid these deformations in the final result. The most obvious weak mode is a translation or rotation of the entire tracking device, which can be only fixed with some kind of reference frame, be it an external system or by definition. This, however, is less severe and sometimes even not considered at all, as it does not affect the internal alignment of the tracking device.

Once a set of alignment parameters is calculated, it should always be validated [83, chapter 11]. Apart from checking the improvement of the residuals, several physics measurements can be utilized, especially to probe for remaining weak modes. Known charge, forward-backward or φ-symmetries of distinct physics processes can be used. Distributions of the signed curvature or the signed transverse impact parameter are also sensitive observables.

13.1.4.2   Examples of Alignment Algorithms

Some modern experiments deploy large tracking devices that require a large number of alignment parameters, of the order of 10^5. In such a case the computation of parameters by using straightforward recipes might become unreasonably slow or cause numerical problems. The two algorithms presented in this section are examples of how to cope with such challenging circumstances.

The HIP Algorithm

The HIP algorithm [101] is a straightforward and easy-to-implement biased alignment algorithm. It computes the alignment parameters for each alignable object separately. Only when iterating on the track sample a certain kind of indirect feedback between the alignable objects is established due to the track refit.

Since only individual alignable objects are regarded, Eq. (13.5) can be partitioned. This is simply done by evaluating the corresponding expressions for each alignable object i together with its associated parameters $\boldsymbol{p}_i$:

$$\boldsymbol{r}_i(\boldsymbol{p}_i) = \boldsymbol{m}_i - \boldsymbol{d}_i(\check{\boldsymbol{q}}, \boldsymbol{p}_i) \approx \boldsymbol{m}_i - \check{\boldsymbol{d}}_i - \boldsymbol{D}_{pi}\boldsymbol{\Delta}\boldsymbol{p}_i$$

with

$$\check{\boldsymbol{d}}_i = \boldsymbol{d}_i(\check{\boldsymbol{q}}, \check{\boldsymbol{p}}_i), \quad \Delta \boldsymbol{p}_i = \boldsymbol{p}_i - \check{\boldsymbol{p}}_i, \quad \boldsymbol{D}_{pi} = \left. \frac{\partial \boldsymbol{d}_i}{\partial \boldsymbol{p}_i} \right|_{\check{\boldsymbol{q}}, \check{\boldsymbol{p}}_i}.$$

The result is determined by minimizing the normalized squared residuals from a given set of tracks, again for each alignable object separately. The formal solution is given by

$$\Delta \boldsymbol{p}_i = \left(\sum_{\text{tracks}} \boldsymbol{D}_{pi}{}^{\mathrm{T}} \boldsymbol{V}_i^{-1} \boldsymbol{D}_{pi} \right)^{-1} \left(\sum_{\text{tracks}} \boldsymbol{D}_{pi}{}^{\mathrm{T}} \boldsymbol{V}_i^{-1} \boldsymbol{r}_i(\check{\boldsymbol{p}}_i) \right)$$

The Millepede Algorithm

The Millepede algorithm [102] is an unbiased algorithm that minimizes the sum of the squared residuals of all tracks at once. To this end a system of linear equations, equivalent to the formal solution of an ordinary χ^2-fit, is solved. However, to achieve this within a reasonable amount of time, only the solution for the alignment parameters is computed, while the computation of the improved track parameters is skipped. This is possible because of the special structure of the system: Firstly, the coefficient matrix is symmetric and, mostly due to the independence of the individual tracks, relatively sparse. Secondly, only the alignment parameters are common parameters for all track measurements, while the specific track parameters are only relevant for each corresponding track. Due to the latter, the solutions for the alignment and track parameters are only coupled via coefficient matrices of the form

$$\boldsymbol{G} = \boldsymbol{D}_p{}^{\mathrm{T}} \boldsymbol{V}^{-1} \boldsymbol{D}_q.$$

To set up the reduced system of equations, for each track the following information has to be extracted:

$$\boldsymbol{\Gamma} = \boldsymbol{D}_q{}^{\mathrm{T}} \boldsymbol{V}^{-1} \boldsymbol{D}_q, \quad \boldsymbol{\beta} = \boldsymbol{D}_q{}^{\mathrm{T}} \boldsymbol{V}^{-1} \left(\boldsymbol{m} - \check{\boldsymbol{d}} - \boldsymbol{D}_p \, \Delta \boldsymbol{p}' \right).$$

Here $\Delta \boldsymbol{p}' = \boldsymbol{p}' - \check{\boldsymbol{p}}$ may already include an estimate $\boldsymbol{p}'$ on the actual alignment. Then compute

$$\Delta \boldsymbol{C} = \boldsymbol{D}_p{}^{\mathrm{T}} \boldsymbol{V}^{-1} \boldsymbol{D}_p - \boldsymbol{G} \, \boldsymbol{\Gamma}^{-1} \boldsymbol{G}^{\mathrm{T}}, \quad \Delta \boldsymbol{g} = \boldsymbol{D}_p{}^{\mathrm{T}} \boldsymbol{V}^{-1} \left(\boldsymbol{m} - \check{\boldsymbol{d}} - \boldsymbol{D}_p \, \Delta \boldsymbol{p}' + \boldsymbol{D}_q \, \boldsymbol{\Gamma}^{-1} \boldsymbol{\beta} \right).$$

Note the expression $-\boldsymbol{\Gamma}^{-1}\boldsymbol{\beta}$ instead of $\boldsymbol{\Delta q}$. These are all necessary terms, including implicitly the full information from all track parameters. The complete system of equations to determine the alignment parameters then reads

$$\boldsymbol{C}\,\boldsymbol{\Delta p} = -\boldsymbol{g},$$

with

$$\boldsymbol{C} = \sum_{\text{tracks}} \boldsymbol{\Delta C}, \quad \boldsymbol{g} = \sum_{\text{tracks}} \boldsymbol{\Delta g}.$$

The solution by matrix inversion is only feasible if the number of parameters is fairly small ($N \leq 10^3$). The matrix $\boldsymbol{C}$ is usually relatively sparse, so that less time-consuming and more reliable methods can be used, such as the GMRES algorithm [103].

It is also possible to introduce constraints into the solution, which allows to align on various hierarchical levels at once. When aligning for instance on module- and layer-level at the same time, these constraints can remove redundant degrees of freedom by forcing the average movement of all modules within one layer to zero.

Millepede is a well-tested algorithm. To use it efficiently, some knowledge of its inner workings is of advantage. The HIP algorithm is simpler to implement, but less suitable for very large setups than Millepede. Another unbiased algorithm is the Kalman Alignment Algorithm [104]. It is a sequential method, derived from the Kalman filter (see also [105]).

13.1.5 Momentum Resolution

The momentum resolution that can be achieved by a tracking detector is determined by the magnetic field, the arrangement and precision of the tracking detectors, and the amount of material crossed by the particle. Simple approximate formulas can be obtained for two cases:

(a) A spectrometer consisting of a central bending magnet and two arms of tracking detectors in front of and behind the magnet. This is a typical arrangement for a fixed-target experiment with small track multiplicities.
(b) A set of cylindrical tracking detectors immersed in a homogeneous magnetic field. This is a typical arrangement for the barrel part of a collider experiment, for instance layers of silicon or a TPC.

The units are the same as in Sect. 13.1.3.2: momentum in GeV/c, length in meters, and magnetic field in Tesla.

13.1.5.1 Two-arm Spectrometer

We assume that the trajectory of the particle is parallel to the z axis and that B_y is the only significant component of the magnetic field. The angle of deflection is then given by [37]

$$\alpha \approx -\frac{kq}{p} \int_L B_y \, \mathrm{d}z = -\frac{kq}{p} \bar{B}_y L,$$

where L is the length of the magnet, $\bar{B}_y$ is the average value of the field along the trajectory, p is the momentum, and q, k are as in Eq. (13.2). Assuming that $|q| = 1$, linear error propagation gives

$$\frac{\sigma(p)}{p} = \frac{p\sigma(\alpha)}{k|\bar{B}_y|L}.$$

Assume that each arm consists of m identical position detectors spread over a length l, and that the standard deviation of the measurement error of x is equal to δ. The best angular resolution is obtained if in each arm half of the detectors is placed at each end of the arm. Neglecting all multiple scattering, it is equal to

$$\sigma(\alpha) = \frac{2\delta}{l\sqrt{m/2}}.$$

The relative momentum resolution due to measurement errors is therefore

$$\frac{\sigma_{\mathrm{me}}(p)}{p} = \frac{2p\delta}{l\sqrt{m/2}k|\bar{B}_y|L}.$$

Although this arrangement optimizes the precision in terms of geometry, it offers little redundancy for track finding and should be used only in setups with trivial pattern recognition requirements, for instance in the forward direction of fixed target experiments.

At low energies, multiple scattering can no longer be neglected. Whereas $\sigma(p)/p$ arising from position measurement errors only is proportional to p, the term $\sigma_{\mathrm{ms}}(p)/p$ arising from multiple scattering is proportional to $1/(\beta|\bar{B}_y|L)$, which is large for small β and constant for high momenta ($\beta \approx 1$). Under the same assumptions about the detector positions as above, the following formula is obtained:

$$\frac{\sigma_{\mathrm{ms}}(p)}{p} = \frac{0.0136}{\beta k|\bar{B}_y|L} \left(\frac{md}{X_0}\right)^{1/2},$$

where d/X_0 is the thickness of the detectors in units of radiation length. The total resolution is obtained by adding the corresponding variances and taking the square root,

$$\frac{\sigma(p)}{p} = \frac{\sigma_{\mathrm{me}}(p)}{p} \oplus \frac{\sigma_{\mathrm{ms}}(p)}{p} = ap \oplus b,$$

with a and b depending on the detector and the magnetic field.

13.1.5.2 Cylindrical Spectrometer

Assume that there are m cylindrical detectors immersed in a homogeneous magnetic field B_z parallel to the z axis. The projection of the track on the $x-y$ plane is a circle with curvature κ. For high momentum the circle can be approximated by a parabola, the detector cylinders can be approximated by planes, and multiple scattering can be neglected. For this case closed formulas for the joint covariance matrix of κ and the tangent $t_\varphi = \tan\varphi$ of the initial track direction φ can be given [106, 107]. For equidistant detectors, uniform resolution δ and $t_\varphi = 0$ it is given by

$$\mathrm{Cov}\begin{pmatrix} t_\varphi \\ \kappa \end{pmatrix} = \frac{\delta^2}{m(m+1)(m+2)} \begin{pmatrix} \dfrac{12\,(m-1)(2m-1)(8m-11)}{L^2(m-2)} & -\dfrac{360\,(m-1)^3}{L^3(m-2)} \\[2ex] -\dfrac{360\,(m-1)^3}{L^3(m-2)} & \dfrac{720\,(m-1)^3}{L^4(m-2)} \end{pmatrix},$$

where L is now the track length in the $x-y$ projection. L is approximately equal to the radial distance between the innermost and the outermost detector. As $\kappa = kB_z/p_\mathrm{T}$,

$$\frac{\sigma_{\mathrm{me}}(p_\mathrm{T})}{p_\mathrm{T}} = \frac{p_\mathrm{T}}{k|B_z|L}\frac{\delta}{L}\left[\frac{720\,(m-1)^3}{(m-2)m(m+1)(m+2)}\right]^{1/2}.$$

There is a high negative correlation between $1/p_\mathrm{T}$ and the direction tangent t_φ. For large m, the asymptotic values are

$$\frac{\sigma_{\mathrm{me}}(p_\mathrm{T})}{p_\mathrm{T}} = \frac{p_\mathrm{T}}{k|B_z|L}\frac{\delta}{L}\left[\frac{720}{m+4}\right]^{1/2}, \quad \sigma(t_\varphi) = \frac{\delta}{L}\left[\frac{192}{m+3.875}\right]^{1/2},$$

$$\mathrm{cov}(t_\varphi, 1/p_\mathrm{T}) = -\frac{\sqrt{15}}{4} = -0.968.$$

More general closed formulas for $t_\varphi \neq 0$ are given in [107].

Table 13.2 Values of C_m in Eq. (13.6)

m	3	4	5	6	7	8	9	10	> 10 m
C_m	1.16	1.06	1.04	1.03	1.02	1.02	1.01	1.01	≈ 1 m

If one half of the detectors is placed at the center of the track and one quarter at either end, the variance of the curvature is minimal, and the covariance matrix reads

$$
\mathrm{Cov}\begin{pmatrix} t_\varphi \\ \kappa \end{pmatrix} = \delta^2 \begin{pmatrix} \dfrac{72}{L^2 m} & -\dfrac{128}{L^3 m} \\[2mm] -\dfrac{128}{L^3 m} & \dfrac{256}{L^4 m} \end{pmatrix},
$$

which is considerably smaller than in the equidistant case. This arrangement, however, is not particularly well suited for track finding and moreover difficult to realize.

The contribution of multiple scattering to the transverse momentum resolution can be approximated by

$$
\frac{\sigma_{\mathrm{ms}}(p_{\mathrm{T}})}{p_{\mathrm{T}}} = C_m \cdot \frac{s}{\beta k |B_z| L} \left(\frac{md}{X_0 \cos \lambda} \right)^{1/2}, \tag{13.6}
$$

where d/X_0 is the thickness of the detectors in units of radiation length, $\lambda = \pi/2 - \theta$ is the dip angle of the track, $s = 0.0136(1 + 0.038 \ln(d/X_0))$, k is as in Eq. (13.2), and C_m is a factor depending on m. Values of C_m for small m, obtained by the program described in [108], are given in Table 13.2. Note that the values are different from the ones given in [106]. In a time projection chamber md/X_0 has to be replaced by L/X_0, where X_0 is the radiation length of the gas. The factor $\cos \lambda$ in the denominator accounts for the actual amount of matter traversed by a track with dip angle λ. Approximate formulas for the best possible resolution including multiple scattering can be found in [109].

The total transverse momentum resolution is calculated by quadratic addition,

$$
\frac{\sigma(p_{\mathrm{T}})}{p_{\mathrm{T}}} = \frac{\sigma_{\mathrm{me}}(p_{\mathrm{T}})}{p_{\mathrm{T}}} \oplus \frac{\sigma_{\mathrm{ms}}(p_{\mathrm{T}})}{p_{\mathrm{T}}},
$$

which can be written in the form

$$
\frac{\sigma(p_{\mathrm{T}})}{p_{\mathrm{T}}} = \frac{a\, p_{\mathrm{T}}}{\sqrt{m+4}} \oplus \frac{b\sqrt{m}}{\sqrt{\cos \lambda}}.
$$

This shows that an optimal m exists for every p_{T} and λ if the projected track length L is kept fixed. Overinstrumentation will deteriorate the resolution for low momenta unless additional measurements can be included without increasing the amount of matter to be traversed.

In order to calculate the error of the momentum $p = p_T/\cos\lambda$ the error in λ must be taken into account:

$$\sigma^2(p) = \sigma^2(p_T)/\cos^2\lambda + \sigma^2(\lambda)\,p_T^2\sin^2\lambda/\cos^4\lambda,$$

the correlation between p_T and λ being negligible in practice. Because of $\sigma(p)/p = p\sigma(1/p)$ it follows that:

$$\frac{\sigma(p)}{p} = \frac{\sigma(p_T)}{p_T} \oplus \sigma(\lambda)\tan\lambda.$$

With the exception of very low momenta the track can be approximated by a straight line in the r–z projection, where $r = (x^2 + y^2)^{1/2}$. For m equidistant detectors and uniform resolution δ, the variance of the direction tangent $t_\lambda = \tan\lambda$ due to the measurement errors is given by [106]:

$$\sigma_{me}^2(t_\lambda) = \frac{\delta^2}{L^2}\frac{12(m-1)}{m(m+1)}\frac{1}{\cos^4\lambda}.$$

If the measurement error in z is very small, the variance of t_λ is dominated by multiple scattering. For equidistant layers of uniform thickness d an approximate formula can be given. Remarkably, it does not depend on the number of layers:

$$\sigma_{ms}^2(t_\lambda) \approx \frac{s^2}{p^2}\frac{d}{X_0\cos\lambda}\frac{1}{\cos^4\lambda} = \frac{s^2}{p_T^2}\frac{d}{X_0\cos\lambda}\frac{1}{\cos^2\lambda},$$

with $s = 0.0136(1 + 0.038\ln(d/X_0))$.

In the design and optimization phase of the detector a precise evaluation of the resolution of all track parameters is mandatory. There are several software packages that allow a fast track simulation plus reconstruction in a general detector setup, for instance [110] (in FORTRAN), [111] (in Matlab/Octave), or [108] (in Java).

13.2   Vertex Reconstruction

13.2.1   Introduction

Vertex reconstruction is the task of finding and estimating the production point of a set of particles. The pattern recognition algorithms and statistical estimation methods involved are in many respects similar to the ones used in track reconstruction. For an overview of vertex reconstruction algorithms used in past or active experiments see for instance [112–116].

In practice it is useful to distinguish between several types of vertices:

1. The primary vertex is the point of collision of two beam particles (in a collider experiment) or of a beam particle and a target particle (in a fixed-target experiment).
2. A secondary decay vertex is the point where an unstable particle decays in the detector volume or in the beam pipe. An example is the decay $K_S^0 \to \pi^+\pi^-$.
3. A secondary interaction vertex is the point where a particle interacts with the material of the detector. Examples are bremsstrahlung, pair production, and inelastic hadronic interactions.

Vertex reconstruction frequently proceeds in several steps:

1. Vertex finding: Finds the tracks that belong to a common primary or secondary vertex.
2. Vertex fitting: Estimates for each vertex candidate the location of the common vertex and computes the associated covariance matrix.
3. Test of vertex hypothesis: Tests for each vertex candidate whether all tracks do indeed belong to the vertex and identifies outliers.
4. Update: Uses the vertex constraint to improve the location and momentum estimate of the tracks belonging to the vertex.
5. Kinematic fit: Kinematic constraints such as momentum and energy conservation are imposed on the mother and daughter particles of a vertex, and mass hypotheses are tested. Kinematic fits are most frequently applied to secondary decay vertices.

Vertex finding can be accomplished in many different ways. A few of them will be described in Sect. 13.2.2. The vertex fit takes a vertex candidate and estimates the vertex location from the estimated track parameters of the outgoing particles (Sect. 13.2.3). As a rule, only charged particles are used, but sometimes also neutral particles contribute to the vertex fit. In the test stage (Sect. 13.2.3.2) outliers are identified, i.e., particles that apparently do not belong to the estimated vertex. As this can lead to a different assignment of particles to vertices, it can be considered as a method of vertex finding. If outliers are expected, the estimation procedure should be robust so that the estimated vertex is not significantly biased by the outliers (Sect. 13.2.3.3). Kinematic constraints (Sect. 13.2.4) are usually imposed via Lagrange multipliers. By repeating the kinematic fit under various mass hypotheses of the mother and/or daughter particles the most likely mass assignment can be found out.

13.2.2 Vertex Finding

Vertex finding is the process of dividing the reconstructed tracks in an event into classes such that presumably all tracks in a class are produced at the same vertex. The primary vertex in an event is usually easy to find, especially if prior information

about its location is available (beam profile, target position). On the other hand, secondary decay vertices of short-lived decays are hard to find, as some of the decay products may also be compatible with the primary vertex. Vertex finding methods can be roughly divided in three main types: generic clustering algorithms, topological methods, and iterated estimators. The latter can be considered as a special divisive clustering method.

13.2.2.1 Clustering Methods

As mentioned above in the context of jet finding (see Sect. 13.1.3.6), clustering methods are based on a distance matrix or a similarity matrix of the objects to be classified. A cluster is then a group with small distances (large similarities) inside the group and large distances (small similarities) to objects outside the group. The distance measure reflects only the geometry of the tracks.

Various clustering methods have been evaluated in the context of vertex finding, of both the hierarchical and the non-hierarchical type [117]. Hierarchical clustering can be agglomerative or divisive. In agglomerative clustering each track starts out as a single cluster. Clusters are merged iteratively on the basis of a distance measure. The shortest distance in space between two tracks is peculiar insofar as it does not satisfy the triangle inequality: if tracks a and b are close, and tracks b and c are close, it does not follow that tracks a and c are close as well. The distance between two clusters of tracks should therefore be defined as the maximum of the individual pairwise distances, known as complete linkage in the clustering literature. Alternatively, the distance between two clusters can be the distance between the two vertices fitted from the clusters. Divisive clustering starts out with a single cluster containing all tracks. Further division of this cluster can be based on repeated vertex estimation with outlier identification (see Sect. 13.2.2.3). Examples of non-hierarchical clustering methods used in vertex finding are vector quantization, the k-means algorithm and deterministic annealing [113].

13.2.2.2 Topological Methods

A very general topological vertex finder was proposed in [118]. It is related to the Radon transform, which is a continuous version of the Hough transform used for track finding (Sect. 13.1.2.1). The search for vertices is based on a function $V(\boldsymbol{v})$ which quantifies the probability of a vertex at location $\boldsymbol{v}$. For each track a Gaussian probability tube $f_i(\boldsymbol{v})$ is constructed. The function $V(\boldsymbol{v})$ is defined taking into account that the value of $f_i(\boldsymbol{v})$ must be significant for at least two tracks:

$$V(\boldsymbol{v}) = \sum_{i=0}^{n} f_i(\boldsymbol{v}) - \frac{\sum_{i=0}^{n} f_i^2(\boldsymbol{v})}{\sum_{i=0}^{n} f_i(\boldsymbol{v})}$$

Due to the second term on the right-hand side, $V(v) \approx 0$ in regions where $f_i(v)$ is significant for only one track. The form of $V(v)$ can be modified to fold in known physics information about probable vertex locations. For instance, $V(v)$ can be augmented by a further function $f_0(v)$ describing the location and spread of the interaction point. In addition, $V(v)$ may be modified by a factor dependent on the angular location of the point v.

Vertex finding amounts to finding local maxima of the function $V(v)$. The search starts at the calculated maxima of the products $f_i(v) f_j(v)$ for all track pairs. For each of these points the nearest maximum of $V(v)$ is found. These maxima are clustered together to form candidate vertex regions. The final association of the tracks to the vertex candidates can be done on the basis of the respective χ^2 contributions or by an adaptive fit (see Sect. 13.2.3.3). In [119] the topological vertex finder was augmented by a procedure based on the concept of the minimum spanning tree of a graph.

13.2.2.3 Iterated Estimators

Vertex finding can also be accomplished by iterated vertex fits (see Sect. 13.2.3). The procedure can be summarized in the following way:

1. Fit one vertex with all tracks
2. Discard all incompatible tracks
3. Repeat step 1 with all discarded tracks

The iteration stops when no vertex with at least two tracks can be successfully fitted. Step 2 might itself be iterative, especially if the vertex fit is not robust, so that the incompatible tracks have to be removed sequentially. Iterative vertex finders based on a least-squares fit (Sect. 13.2.3.1) and an adaptive fit (Sect. 13.2.3.3) are implemented in the RAVE toolbox [120, 121].

13.2.3 Vertex Fitting

The input to the vertex fit is a vertex candidate, i.e., a set of estimated track parameters $\{\tilde{q}_1, \ldots, \tilde{q}_n\}$ located at one or more reference surfaces, along with their covariance matrices $\{C_1, \ldots, C_n\}$. For instance, in the primary vertex fit in a collider experiment the reference surface may be the beam tube. If possible, the reference surface(s) should be chosen such that multiple scattering between the vertex and the location of the track parameters is negligible.

The parameters to be fitted are the vertex position v and the track momenta p_i at the vertex. The functional dependence of the track parameters on the vertex parameters requires a track model, which depends on the shape of the magnetic field in the vicinity of the vertex. If the field is homogeneous, the track model is a helix;

if the field is zero, the track model is a straight line. In other cases the track model may have to be computed numerically (see Sect. 13.1.3.2).

13.2.3.1 Least-Squares Methods

The conventional approach to estimating the vertex position is the minimization of some quadratic objective function, yielding a least-squares estimate. There are two main flavors of least-squares estimation in vertex fitting, constrained and unconstrained minimization. In the first case the vertex constraint is introduced into the objective function via a Lagrange multiplier, in the second case the constraint is implicit in the track model.

As an example, consider a vertex fit with n straight tracks. The n straight tracks originating from the common vertex $\boldsymbol{v} = (x_{\mathrm{v}}, y_{\mathrm{v}}, z_{\mathrm{v}})^{\mathrm{T}}$ can be represented by n straight lines with parameters λ_i:

$$x = x_{\mathrm{v}} + \lambda_i a_i, \quad y = y_{\mathrm{v}} + \lambda_i b_i, \quad z = z_{\mathrm{v}} + \lambda_i, \quad i = 1, \ldots, n,$$

where a_i and b_i are the direction tangents at the vertex. At the reference surface $z = z_{\mathrm{ref}}$ track i is specified by its parameter vector $\boldsymbol{q}_i = (x_i, y_i, a_i, b_i)^{\mathrm{T}}$, consisting of the intersection point (x_i, y_i) and the two direction tangents (a_i, b_i). The track fit delivers estimates $\tilde{\boldsymbol{q}}_i$ and information matrices $\boldsymbol{G}_i$ for $i = 1, \ldots, n$. In the constrained problem the sum of the squared residuals

$$M(\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n) = \sum_{i=1}^{n} \boldsymbol{e}_i^{\mathrm{T}} \boldsymbol{G}_i \boldsymbol{e}_i, \quad \boldsymbol{e}_i = \tilde{\boldsymbol{q}}_i - \boldsymbol{q}_i, \tag{13.7}$$

must be minimized under the $2n$ nonlinear constraints

$$x_{\mathrm{v}} = x_i + a_i(z - z_{\mathrm{ref}}), \quad y_{\mathrm{v}} = y_i + b_i(z - z_{\mathrm{ref}}), \quad i = 1, \ldots, n$$

There are $4n + 3$ unknowns, $4n$ observations and $2n$ constraints, giving $4n + 2n - (4n + 3) = 2n - 3$ degrees of freedom. The resulting track parameters $\bar{\boldsymbol{q}}_i$ fit best, in the least-squares sense, to the track fit estimates $\tilde{\boldsymbol{q}}_i$ and at the same time have a common vertex. For the solution of the constrained vertex fit see Sect. 13.2.4.2.

In the example, the constraints can be rewritten as

$$x_i = x_{\mathrm{v}} + (z_{\mathrm{ref}} - z)a_i, \quad y_i = y_{\mathrm{v}} + (z_{\mathrm{ref}} - z)b_i, \quad i = 1, \ldots, n. \tag{13.8}$$

Insertion of Eq. (13.8) into Eq. (13.7) gives the objective function of the unconstrained nonlinear least-squares problem:

$$M(\boldsymbol{v}, a_1, b_1, \ldots, a_n, b_n) = \sum_{i=1}^{n} \boldsymbol{e}_i^{\mathrm{T}} \boldsymbol{G}_i \boldsymbol{e}_i, \quad \boldsymbol{e}_i = \tilde{\boldsymbol{q}}_i - \boldsymbol{q}_i.$$

There are now $4n$ observations and $2n + 3$ unknown parameters, namely the vertex position and the track directions at the vertex, giving again $4n - (2n + 3) = 2n - 3$ degrees of freedom.

A generalization of this simple case to helix tracks can be found in [122–124]. In the general case, the unconstrained problem can be formulated in terms of the unknown vertex position $\boldsymbol{v}$ and the unknown track momentum vectors $\boldsymbol{p}_i$ at the vertex [26, 59]. The measurement equation reads

$$\boldsymbol{q}_i = \boldsymbol{h}_i(\boldsymbol{v}, \boldsymbol{p}_i), i = 1, \ldots, n, \tag{13.9}$$

where the function $\boldsymbol{h}_i$ incorporates the track model in the magnetic field. The objective function is equal to

$$M(\boldsymbol{v}, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_n) = \sum_{i=1}^{n} \boldsymbol{e}_i^{\mathrm{T}} \boldsymbol{G}_i \boldsymbol{e}_i, \quad \boldsymbol{e}_i = \tilde{\boldsymbol{q}}_i - \boldsymbol{q}_i.$$

Minimization of the objective function can proceed in several ways. For a detailed exposition of non-linear least-squares estimation see e.g. [125].

Gauss-Newton Method

Assume that there are approximate values $\check{\boldsymbol{v}}$ and $\check{\boldsymbol{p}}_i$ for all i. Then Eq. (13.9) can be approximated by an affine function:

$$\boldsymbol{q}_i \approx \boldsymbol{h}_i(\check{\boldsymbol{v}}, \check{\boldsymbol{p}}_i) + \boldsymbol{A}_i(\boldsymbol{v} - \check{\boldsymbol{v}}) + \boldsymbol{B}_i(\boldsymbol{p}_i - \check{\boldsymbol{p}}_i) = \boldsymbol{c}_i + \boldsymbol{A}_i \boldsymbol{v} + \boldsymbol{B}_i \boldsymbol{p}_i,$$

with

$$\boldsymbol{A}_i = \left. \frac{\partial \boldsymbol{h}_i(\boldsymbol{v}, \boldsymbol{p}_i)}{\partial \boldsymbol{v}} \right|_{\check{\boldsymbol{v}}, \check{\boldsymbol{p}}_i}, \quad \boldsymbol{B}_i = \left. \frac{\partial \boldsymbol{h}_i(\boldsymbol{v}, \boldsymbol{p}_i)}{\partial \boldsymbol{p}_i} \right|_{\check{\boldsymbol{v}}, \check{\boldsymbol{p}}_i}, \quad \boldsymbol{c}_i = \boldsymbol{h}_i(\check{\boldsymbol{v}}, \check{\boldsymbol{p}}_i) - \boldsymbol{A}_i \check{\boldsymbol{v}} - \boldsymbol{B}_i \check{\boldsymbol{p}}_i.$$

The objective function then reads

$$M(\boldsymbol{v}, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_n) = \sum_{i=1}^{n} (\tilde{\boldsymbol{q}}_i - \boldsymbol{c}_i - \boldsymbol{A}_i \boldsymbol{v} - \boldsymbol{B}_i \boldsymbol{p}_i)^{\mathrm{T}} \boldsymbol{G}_i (\tilde{\boldsymbol{q}}_i - \boldsymbol{c}_i - \boldsymbol{A}_i \boldsymbol{v} - \boldsymbol{B}_i \boldsymbol{p}_i).$$

As M is now quadratic in the unknown parameters, the minimum can be computed explicitly. The estimated vertex position and its covariance matrix are given by

$$\tilde{\boldsymbol{v}}_n = \boldsymbol{C}_n \sum_{i=1}^{n} \boldsymbol{A}_i^{\mathrm{T}} \boldsymbol{G}_i^{B}(\tilde{\boldsymbol{q}}_i - \boldsymbol{c}_i), \quad \mathrm{Var}(\tilde{\boldsymbol{v}}_n) = \boldsymbol{C}_n = \left(\sum_{i=1}^{n} \boldsymbol{A}_i^{\mathrm{T}} \boldsymbol{G}_i^{B} \boldsymbol{A}_i \right)^{-1}, \tag{13.10}$$

with

$$G_i{}^B = G_i - G_i B_i W_i B_i{}^T G_i, \quad W_i = (B_i{}^T G_i B_i)^{-1}.$$

In general, the procedure has to be iterated. The measurement equation is expanded at the new estimate, and the estimate is recomputed until convergence is obtained. The formulas required for the implementation of two important cases, fixed-target configuration and solenoidal configuration, are given in [61].

Once $\tilde{v}_n$ is known, the track momenta and the full covariance matrix can be computed:

$$\tilde{p}_i^n = W_i B_i{}^T G_i (\tilde{q}_i - c_i - A_i \tilde{v}_n),$$

$$\text{Var}(\tilde{p}_i^n) = D_i^n = W_i + W_i B_i{}^T G_i A_i C_n A_i{}^T G_i B_i W_i, \quad (13.11)$$

$$\text{Cov}(\tilde{p}_i^n, \tilde{v}_n) = E_i^n = -W_i B_i{}^T G_i A_i C_n.$$

The estimates can also be computed recursively, resulting in an extended Kalman filter [25, 26, 59]:

$$\tilde{v}_i = C_i[C_{i-1}^{-1}\tilde{v}_{i-1} + A_i{}^T G_i{}^B(\tilde{q}_i - c_i)], \quad C_i = (C_{i-1}^{-1} + A_i{}^T G_i A_i)^{-1}$$

$$\tilde{p}_i = W_i B_i{}^T G_i (\tilde{q}_i - c_i - A_i \tilde{v}_i), \qquad D_i = W_i + W_i B_i{}^T G_i A_i C_i A_i{}^T G_i B_i W_i,$$

$$E_i = -W_i B_i{}^T G_i A_i C_i.$$

The associated smoother is tantamount to recomputing the track momenta using the last vertex estimate $\tilde{v}_n$, i.e., Eq. (13.11).

Newton–Raphson Method

This method uses a local quadratic approximation to the objective function. In order to simplify the notation we introduce $\alpha = (v, p_1, \ldots, p_n)^T$, $\tilde{q} = (\tilde{q}_1, \ldots, \tilde{q}_n)^T$ and $h = (h_1, \ldots, h_n)^T$. Then the objective function can be written as

$$M(\alpha) = [\tilde{q} - h(\alpha)]^T G[\tilde{q} - h(\alpha)], \quad G = \text{diag}(G_1, \ldots, G_n).$$

If $\check{\alpha}$ is an appropriate expansion point, $M(\alpha)$ is approximated by

$$M(\alpha) \approx M(\check{\alpha}) + g^T(\alpha - \check{\alpha}) + \tfrac{1}{2}(\alpha - \check{\alpha})^T \Omega(\alpha - \check{\alpha}),$$

where

$$g = \frac{\partial M}{\partial \alpha} = -2H^T G[\tilde{q} - h(\check{\alpha})], \quad \Omega = \frac{\partial^2 M}{\partial \alpha \partial \alpha^T} = 2H^T G H - 2\frac{\partial H^T}{\partial \alpha^T} G[\tilde{q} - h(\check{\alpha})]$$

are the gradient and the Hessian of M, respectively, evaluated at $\breve{\boldsymbol{\alpha}}$, and $\boldsymbol{H}$ is the Jacobian of the track model $\boldsymbol{h}(\boldsymbol{\alpha})$. If $\boldsymbol{\Omega}$ is positive definite, M has a minimum when its gradient is zero, leading to

$$\tilde{\boldsymbol{\alpha}} = \breve{\boldsymbol{\alpha}} - \boldsymbol{\Omega}^{-1}\boldsymbol{g}.$$

If the second term of the Hessian is set to zero, the Gauss–Newton method is recovered. Clearly, the Newton–Raphson method is more complex, but it gives some additional information about the problem. In particular, a Hessian that is not positive definite indicates that the expansion point is too far from the true global minimum.

Levenberg–Marquardt Method

In this method the matrix $\boldsymbol{H}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{H}$ is inflated by a diagonal matrix $k\boldsymbol{I}$. As a consequence, the direction of the parameter update is intermediate between the direction of the Gauss–Newton step ($k = 0$) and the direction of steepest descent ($k \to \infty$). An example of a vertex fit with the Levenberg–Marquardt method is given in [122].

Fast Vertex Fits

The estimated track parameters $\tilde{\boldsymbol{q}}_i$ are frequently given at the innermost detector surface or at the beam tube. If the $\tilde{\boldsymbol{q}}_i$ are propagated to the vicinity of the presumed vertex, the vertex estimation can be speeded up by applying some approximations.

The "perigee" parametrization for helical tracks was introduced in [123], with a correction in [124]. The track is parameterized around the point of closest approach (the perigee point $\boldsymbol{v}^{\mathrm{P}}$) of the helix to the z-axis. The variation of transverse errors along the track is neglected in the vicinity of the perigee, and the track direction and curvature at the vertex is considered to be constant. The approximate objective function of the vertex fit can then be written entirely in terms of the perigee points:

$$M(\boldsymbol{v}) = \sum_{i=1}^{n}(\boldsymbol{v}_i{}^{\mathrm{P}} - \boldsymbol{v})^{\mathrm{T}}\boldsymbol{T}_i(\boldsymbol{v}_i{}^{\mathrm{P}} - \boldsymbol{v}), \tag{13.12}$$

where $\boldsymbol{T}_i$ is a weight matrix of rank 2. The vertex estimate is then

$$\tilde{\boldsymbol{v}} = \left(\sum_{i=1}^{n}\boldsymbol{T}_i\right)^{-1}\left(\sum_{i=1}^{n}\boldsymbol{T}_i\boldsymbol{v}_i{}^{\mathrm{P}}\right).$$

The Jacobians required to compute the $\boldsymbol{T}_i$ are spelled out in [123, 124].

A further simplification was proposed in [126]. In the vicinity of the vertex the track is approximated by a straight line. The estimated track parameters are transformed to a coordinate system the x-axis of which is parallel to the track. The vertex is then estimated by minimizing the sum of the weighted transverse distances of the tracks to the vertex. The resulting objective function has the same form as in Eq. (13.12), again with weight matrices of rank 2. The estimate is exact for straight tracks.

A different type of a fast vertex fitting algorithm is described in [127]. It is based on approximating the tracks by straight lines both in the $x-y$ plane and in the $x-z$ plane. In either projection, the lines representing the tracks are Hough-transformed to points in the dual plane of line parameters. The vertex coordinates are then obtained by a weighted linear least-squares fit in the dual plane.

Adding Prior Information

If the vertex to be fitted is the primary vertex, there may be prior information about the vertex position from the beam profile in a collider experiment or the target location in a fixed target experiment. The prior information usually comes in the form of a position v_0 plus a covariance matrix C_0. The objective function is then augmented by an additional term

$$(v_0 - v)^{\mathrm{T}} C_0^{-1} (v_0 - v).$$

For instance, the Gauss–Newton estimate Eq. (13.10) is modified in the following way:

$$\tilde{v}_n = C_n \left[C_0^{-1} v_0 + \sum_{i=1}^n A_i{}^{\mathrm{T}} G_i{}^B (\tilde{q}_i - c_i) \right], \ \mathrm{Var}(\tilde{v}_n) = C_n = \left(C_0^{-1} + \sum_{i=1}^n A_i{}^{\mathrm{T}} G_i{}^B A_i \right)^{-1}.$$

Similar modifications apply to the Newton–Raphson estimate and the fast vertex fits.

13.2.3.2 Vertex Quality and Outlier Removal

Some tracks used in the vertex fit may be outliers in the sense that they do not actually belong to the vertex. Also, the estimated track parameters may be distorted by outliers or distorted hits in the track fit. Both types of outliers distort the vertex estimate and need to be identified.

In the case of Gaussian errors and a linear model the contribution of each track to the minimum value of the objective function is distributed according to a χ^2-distribution with two degrees of freedom. The contribution χ_i^2 of track i has to be computed relative to the vertex estimated without track i. For instance, in the

Gauss–Newton algorithm:

$$\chi_i^2 = r_i^{n\mathrm{T}} G_i r_i^n + (\tilde{v}_n - \tilde{v}_n^{-i})^{\mathrm{T}} (C_n^{-i})^{-1} (\tilde{v}_n - \tilde{v}_n^{-i}),$$

where $r_i^n = \tilde{q}_i - c_i - A_i \tilde{v}_n - B_i \tilde{p}_i^n$ is the residual of track i and $\tilde{v}_n^{-i}$ is the vertex estimate with track i removed:

$$\tilde{v}_n^{-i} = C_n^{-i} \left[C_n^{-1} \tilde{v}_n - A_i^{\mathrm{T}} G_i^{\,B} (\tilde{q}_i - c_i) \right], \quad C_n^{-i} = \left(C_n^{-1} - A_i^{\mathrm{T}} G_i^{\,B} A_i \right)^{-1}.$$

Analogous but somewhat simpler formulas hold for the fast vertex fits.

The test statistic χ_i^2 can be computed for all i, and the track with the largest χ_i^2 is a candidate for removal. This procedure can be repeated until all χ_i^2 are below the cut. Even if there is only a single outlier, all χ_i^2 are no longer χ^2-distributed and the power of the test is impaired. This loss of power can be compensated by robust estimation of the vertex.

13.2.3.3   Robust and Adaptive Estimators

Robust estimators are less influenced or not influenced at all by outlying observations. This can be achieved by downweighting outliers or by excluding them from the estimate. For example, in the case of a one-dimensional location estimate, the M-estimator [128] downweights outliers, whereas the LMS (least median of squares) estimator [129] uses only one half of the sample (the one spanning the shortest interval) and ignores the other one.

Robust estimators tend to be statistically less efficient and computationally more expensive than least-squares estimators. On the other hand, estimation and outlier detection are performed in parallel, whereas a least-squares estimator has to be recomputed after an outlier has been identified and removed.

One of the earliest proposals for a robust vertex fit is in [130]. The method is an M-estimator with Huber's ψ-function [131]. It is implemented as a re-weighted least-squares estimator. The initial vertex estimate is a plain least-squares estimate. Then, for each track, the residuals are rotated to the eigensystem of the covariance matrix of the track, and weight factors are computed according to

$$w_i = \frac{\psi(r_i/\sigma_i)}{r_i/\sigma_i} = \begin{cases} 1, & |r_i| \le c\sigma_i, \\ c\sigma_i/|r_i|, & |r_i| > c\sigma_i, \end{cases}$$

where r_i is one of the residuals in the rotated frame, σ_i is the standard deviation in the rotated frame, and c is the robustness constant, usually chosen between 1 and 3. The weight factors are applied and the estimate is recomputed. The entire procedure is iterated until convergence.

A different kind of re-weighted least-squares estimator is proposed in [132]. The weights are computed according to Tukey's bi-square function [128]:

$$
w_i = \begin{cases} \left(1 - \dfrac{r_i^2/\sigma_i^2}{c^2}\right)^2, & |r_i| \le c\sigma_i, \\ 0, & \text{otherwise}, \end{cases}
$$

where r_i^2 is the squared residual of track i with respect to the vertex, σ_i^2 is its variance, and c is again the robustness constant. The estimator is now equivalent to a redescending M-estimator, and consequently less sensitive to outliers than Huber's M-estimator.

The combination of a redescending M-estimator with the concept of deterministic annealing [133] leads to the adaptive method of vertex fitting [113, 117, 134, 135]. The concept of the adaptive vertex fit is derived from the Deterministic Annealing Filter [64] (see Sect. 13.1.3.5). The weights are computed according to

$$
w_i = \frac{\exp\!\left(-\chi_i^2/2T\right)}{\exp\!\left(-\chi_i^2/2T\right) + \exp\!\left(-\chi_{\text{cut}}^2/2T\right)},
$$

where χ_i^2 is the χ^2-contribution of track i, χ_{cut}^2 is a cutoff value, and T is a temperature parameter. The computation of the redescending M-estimator can be interpreted as an EM (expectation–maximization) algorithm [136, 137]. Alternatively it can be viewed as the minimization of the energy function of an elastic arm algorithm [18, 19]. If annealing is employed, the iteration starts at high T. The temperature is then gradually decreased. At low T the weights approach either zero or one. The final weights can be used for classification of the tracks as inliers or outliers. A comparison of the adaptive method with other robust estimators can be found in [138]. The adaptive estimator has been extended to a multi-vertex estimator fitting several vertices simultaneously, including competition of all vertices for all tracks [139].

Iterated re-weighted least-squares estimators require a good starting point, in order to ensure convergence to the global minimum and to minimize the number of iterations required. In many cases a standard least-squares estimate is sufficient. In the presence of a large number of outliers also the starting point should be estimated robustly, preferably by an estimator with a high breakdown point [129]. Several such initial estimators have been proposed and studied in [117].

The M-estimators and the adaptive estimator presented above do not presuppose an explicit outlier model. If it is possible to describe the outliers by a Gaussian mixture model, estimation of the vertex can be carried out by the Gaussian-sum filter [140].

Several of the estimators described here are implemented in RAVE, a detector-independent toolkit for reconstruction of interaction vertices [120, 121].

13.2.4   Kinematic Fitting

Kinematic fitting imposes physical constraints on the particles participating in an interaction and thereby improves the measured track momenta and positions. At the same time hypotheses about the interaction and the participating particles can be tested.

13.2.4.1   Lagrange Multiplier Method

The most commonly used method of imposing constraints on the measured tracks is by way of Lagrange multipliers [141]. Let $\tilde{\boldsymbol{q}} = (\tilde{\boldsymbol{q}}_1, \ldots, \tilde{\boldsymbol{q}}_n)^\mathrm{T}$ be the unconstrained estimated parameters of a set of n tracks, along with their joint information matrix $\boldsymbol{G} = \mathrm{diag}(\boldsymbol{G}_1, \ldots, \boldsymbol{G}_n) = \boldsymbol{V}^{-1}$. The r functions describing the constraints can be written as $\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{0}$. Taylor expansion around a suitable point $\breve{\boldsymbol{q}}$ yields the linearized equation

$$\breve{\boldsymbol{g}} + \boldsymbol{D}(\boldsymbol{q} - \breve{\boldsymbol{q}}) = \boldsymbol{0},$$

where $\boldsymbol{D}$ is the Jacobian of $\boldsymbol{g}$ with respect to $\boldsymbol{q}$, evaluated at $\breve{\boldsymbol{q}}$, and $\breve{\boldsymbol{g}} = \boldsymbol{g}(\breve{\boldsymbol{q}})$. The obvious expansion point is $\breve{\boldsymbol{q}} = \tilde{\boldsymbol{q}}$. The constrained track parameters $\bar{\boldsymbol{q}}_i$ are obtained by minimizing the objective function

$$M(\boldsymbol{q}, \boldsymbol{\lambda}) = (\boldsymbol{q} - \tilde{\boldsymbol{q}})^\mathrm{T} \boldsymbol{G}(\boldsymbol{q} - \tilde{\boldsymbol{q}}) + 2\boldsymbol{\lambda}^\mathrm{T} \left[\breve{\boldsymbol{g}} + \boldsymbol{D}(\boldsymbol{q} - \breve{\boldsymbol{q}}) \right]$$

with respect to $\boldsymbol{q}$ and $\boldsymbol{\lambda}$. $\boldsymbol{\lambda}$ is a vector of r unknowns, the Lagrange multipliers. The solution is

$$\bar{\boldsymbol{q}} = \tilde{\boldsymbol{q}} - \boldsymbol{V} \boldsymbol{D}^\mathrm{T} \bar{\boldsymbol{\lambda}}, \quad \text{with} \quad \bar{\boldsymbol{\lambda}} = \boldsymbol{G}_D \left[\breve{\boldsymbol{g}} + \boldsymbol{D}(\tilde{\boldsymbol{q}} - \breve{\boldsymbol{q}}) \right] \quad \text{and} \quad \boldsymbol{G}_D = (\boldsymbol{D} \boldsymbol{V} \boldsymbol{D}^\mathrm{T})^{-1}.$$

The covariance matrix $\bar{\boldsymbol{V}}$ and the χ^2 statistic are given by

$$\bar{\boldsymbol{V}} = \boldsymbol{V} - \boldsymbol{V} \boldsymbol{D}^\mathrm{T} \boldsymbol{G}_D \boldsymbol{D} \boldsymbol{V}, \quad \chi^2 = \bar{\boldsymbol{\lambda}}^\mathrm{T} \boldsymbol{G}_D^{-1} \bar{\boldsymbol{\lambda}} = \bar{\boldsymbol{\lambda}}^\mathrm{T} \left[\breve{\boldsymbol{g}} + \boldsymbol{D}(\tilde{\boldsymbol{q}} - \breve{\boldsymbol{q}}) \right].$$

If required, the constraint function $\boldsymbol{g}$ can be re-expanded at the new point $\breve{\boldsymbol{q}} = \bar{\boldsymbol{q}}$, and the constrained track parameters can be recomputed.

   The Jacobian $\boldsymbol{D}$ depends both on the parametrization of the tracks and on the type of constraint to be imposed. For kinematic constraints it is often convenient to choose a parametrization that uses physically meaningful quantities. In [142] it is proposed to use the four-momentum and a point in space, i.e., $\boldsymbol{q} = (p_x, p_y, p_z, E, x, y, z)$. With this parametrization the following constraints can be formulated in a straightforward manner (for further examples see [142]).

1. Invariant mass constraint. The equation that constrains a track to have an invariant mass m_c is

$$E^2 - p_x^2 - p_y^2 - p_z^2 - m_c^2 = 0.$$

Expanding at $\check{q} = (\check{p}_x, \check{p}_y, \check{p}_z, \check{E}, \check{x}, \check{y}, \check{z})$ yields

$$\boldsymbol{D} = \begin{pmatrix} -2\check{p}_x & -2\check{p}_y & -2\check{p}_z & 2\check{E} & 0 & 0 & 0 \end{pmatrix}, \quad \check{g} = \check{E}^2 - \check{p}_x^2 - \check{p}_y^2 - \check{p}_z^2 - m_c^2.$$

2. Total energy constraint. The equation that constrains a track to have a total energy E_c is

$$E - E_c = 0.$$

It follows that

$$\boldsymbol{D} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}, \quad \check{g} = \check{E} - E_c.$$

3. Total momentum constraint. The equation that constrains a track to have a total momentum p_c is

$$\sqrt{p_x^2 + p_y^2 + p_z^2} - p_c = 0.$$

Expanding at $\check{q} = (\check{p}_x, \check{p}_y, \check{p}_z, \check{E}, \check{x}, \check{y}, \check{z})$ yields

$$\boldsymbol{D} = \left(\frac{\check{p}_x}{\check{p}} \ \frac{\check{p}_y}{\check{p}} \ \frac{\check{p}_z}{\check{p}} \ 0 \ 0 \ 0 \ 0 \right), \quad \check{g} = \sqrt{\check{p}_x^2 + \check{p}_y^2 + \check{p}_z^2} - p_c.$$

13.2.4.2 Vertex Constraint

If a vertex constraint is added to the kinematic constraints, the constraint functions depend on the unknown vertex position $\boldsymbol{v}$ and are extended to $\boldsymbol{g}(\boldsymbol{q}, \boldsymbol{v}) = \boldsymbol{0}$. Taylor expansion around a suitable point $(\check{q}, \check{v})$ yields the linearized equation

$$\check{g} + \boldsymbol{D}(\boldsymbol{q} - \check{q}) + \boldsymbol{E}(\boldsymbol{v} - \check{v}) = \boldsymbol{0},$$

where $\boldsymbol{E}$ is the Jacobian of $\boldsymbol{g}$ with respect to $\boldsymbol{v}$, evaluated at $\check{v}$, and $\check{g} = \boldsymbol{g}(\check{q}, \check{v})$. It is assumed that there is prior information about the vertex position, represented by the position $\tilde{v}$ and the covariance matrix $\boldsymbol{C}$. The position $\tilde{v}$ can be used as the expansion point $\check{v}$.

The constrained track parameters $\bar{q}_i$ and the estimated vertex position $\bar{v}$ are obtained by minimizing the objective function

$$M(q, v, \lambda) = (q - \tilde{q})^{\mathrm{T}} G(q - \tilde{q}) + (v - \tilde{v})^{\mathrm{T}} C^{-1}(v - \tilde{v}) + 2\lambda^{\mathrm{T}} \left[\breve{g} + D(q - \breve{q}) + E(v - \breve{v}) \right]$$

with respect to q, v, and λ. The solution is

$$\bar{\lambda} = W \left[\breve{g} + D(\tilde{q} - \breve{q}) + E(\tilde{v} - \breve{v}) \right], \quad \bar{v} = \tilde{v} - C E^{\mathrm{T}} \bar{\lambda}, \quad \bar{q} = \tilde{q} - V D^{\mathrm{T}} \bar{\lambda},$$

with $W = \left(D V D^{\mathrm{T}} + E C E^{\mathrm{T}} \right)^{-1}$. The covariance matrices are

$$\mathrm{Var}(\bar{v}) = C - C E^{\mathrm{T}} W E C, \quad \mathrm{Var}(\bar{q}) = V - V D^{\mathrm{T}} W D V, \quad \mathrm{Cov}(\bar{q}, \bar{v}) = -V D^{\mathrm{T}} W E C.$$

The χ^2 statistic is

$$\chi^2 = \bar{\lambda}^{\mathrm{T}} W^{-1} \bar{\lambda} = \bar{\lambda}^{\mathrm{T}} \left[\breve{g} + D(\tilde{q} - \breve{q}) + E(\tilde{v} - \breve{v}) \right],$$

with r degrees of freedom, where r is the number of constraint functions. If the vertex constraint is the only constraint imposed on the tracks, the χ^2 has $2n$ degrees of freedom. If there is no prior information about the vertex, the prior vertex position is assigned an infinitely large covariance matrix, and W is replaced by its limiting value:

$$W = \lim_{C \to \infty} \left(D V D^{\mathrm{T}} + E C E^{\mathrm{T}} \right)^{-1} = G_D - G_D E (E^{\mathrm{T}} G_D E)^{-1} E^{\mathrm{T}} G_D.$$

The number of degrees of freedom is reduced to $2n - 3$.

13.3 Track Reconstruction in the LHC Experiments

13.3.1 ALICE

ALICE [143] is the experiment at the LHC that is devoted to the physics of high energy ion collisions. Its main goal is to investigate the physics of strongly interacting matter and the quark-gluon plasma at extreme values of energy density and temperature in nucleus-nucleus collisions. Among the four experiments at the LHC, ALICE is equipped with the largest number of subdetectors in order to face the reconstruction complexity of ion physics events. In particular, three subdetectors focus on measuring the passage of charged particles using the bending power of the magnetic field. They are assembled in a cylindrical fashion: the Inner Tracking System (ITS) with six planes of high-resolution silicon pixel, drift, and strip detectors, the cylindrical Time-Projection Chamber (TPC) and the Transition Radiation Detector (TRD). The principal functions of the ITS are the

identification and reconstruction of secondary vertices, the track reconstruction of low-p_T particles and the improvement of the impact parameter and momentum resolution. The TPC is the most important tracking sub-detector. Thanks to its time information, it can provide an efficient and robust tracking also in a very high multiplicity environments (in the order of 10,000 charged particles). Finally, the TRD is also used for tracking in the central region and for improving the p_T resolution at high momentum.

The first step in the track reconstruction in ALICE is the clusterization, which is performed separately for each of the three subdetectors [144]. Tracking then proceeds by determining the preliminary interaction vertex using tracklets defined as lines built with pairs of clusters in the first two layers of the ITS. The preliminary interaction vertex is thus found as a space point to which a maximum number of tracklets converge. In the next step, track finding and fitting is performed in three stages using a inward-outward-inward strategy:

- Initially, tracks in the TPC are searched for using the Kalman filter technique and the outermost layers of the TPC for the seed. A preliminary particle identification is also possible at this stage based on the specific energy loss in the TPC gas. Then, the reconstructed TPC tracks are propagated to the outermost ITS layer and become the seeds for finding tracks in the ITS. In Fig. 13.4 the ITS–TPC matching efficiency as a function of the transverse momentum for 2010–2013 data and Monte Carlo for pp and heavy ion collisions is shown. Finally, the last step is performed in order to recover tracks of particle with p_T down to 80 MeV. It performs a standalone ITS reconstruction with those clusters that were not used in the ITS–TPC tracks.

- All reconstructed tracks are then extrapolated to their point of closest approach to the preliminary interaction vertex, and are extrapolated from the innermost layer to the outermost one. Tracks are refitted by the Kalman filter using the clusters found at the previous stage. After the reconstruction in the TRD subdetectors, the track is matched with a possible TRD tracklet in each of the six TRD layers. In a
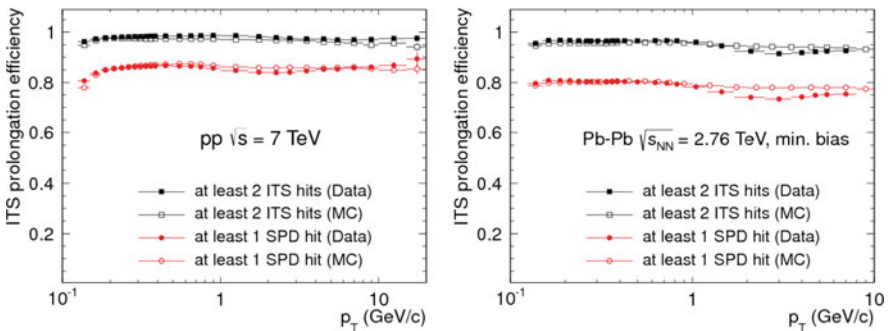

Fig. 13.4 ITS–TPC matching efficiency vs. p_T for data and Monte Carlo for pp (left) for Pb-Pb (right) collisions in the ALICE experiment [144]

similar way, the tracks reaching the time-of-flight (TOF) detector are matched to
TOF clusters.
- At the final stage of the track reconstruction, all tracks in both ITS and TPC
  subdetectors are propagated inwards and refitted one last time to determine the
  final estimate of the track position, direction, inverse curvature, and its associated
  covariance matrix.

The final interaction vertex is then re-determined using the all tracks reconstructed
in TPC and ITS. The precise vertex fit is performed using track weighting to
suppress the contribution of any remaining outliers. For data-taking conditions
where a high pileup rate is expected, a more robust version of vertex finding inspired
by the algorithm described in [132] is used. It is based on iterative vertex finding and
fitting using Tukey bisquare weights to suppress outliers. The algorithm stops when
no more vertices are identified in the scan along the beam direction. Once the tracks
and the interaction vertex have been found, a search for photon conversions and
decays of strange hadrons such as K_S^0 and Λ_0 concludes the central-barrel tracking
procedure.

13.3.2 ATLAS

ATLAS [145] is the largest of the four LHC experiments, measuring 25 m in
diameter and 44 m in length. Its magnet system is composed of a Central Solenoid
Magnet with a 2 T field, a Barrel Toroid and an Endcap Toroids with 4 T each.
The Inner Detector (ID) is very compact and highly sensitive in order to measure
accurately the decay products of each collision. It consists of three different
systems of sensors immersed in the solenoid magnetic field: the Pixel Detector,
the Semiconductor Tracker (SCT), and the Transition Radiation Tracker (TRT).
The Pixel Detector is situated closest to the interaction point and has the highest
granularity with about 80 million readout channels. The intrinsic spatial resolution
of the Pixel Detector sensors is 10 μm in r–ϕ and 115 μm in z. The SCT is a silicon
microstrip detector surrounding the Pixel Detector. It provides eight measurements
per track with an overall resolution of 16 μm in r–ϕ and 580 μm in z. In the
outermost region, the TRT is placed. It is a light-weight detector composed of
proportional gas counters (70% Xe, 27% CO_2 and 3% O_2 straws) embedded in a
radiator material and its operational drift radius accuracy is about 130 μm. The TRT
contributes both to the track pattern recognition stage, featuring typically around 30
hits per track, and to particle identification.

The basic concepts of the ATLAS track reconstruction are described in [146,
147]. The tracking in the ID consists of two principal sequences: an initial inside-out
tracking, and a subsequent outside-in tracking. Inside-out tracking starts with space
point formation in the silicon part of the ID. Using the space points, track seeds
are generated with or without a constraint on the longitudinal vertex position. The
seeds are then followed through the SCT by a combinatorial Kalman filter/smoother.

After ambiguity solving, the remaining track candidates are extended into the TRT. Outside-in tracking first finds track segments in the TRT, using a Hough transform of the straw centers. A Kalman filter/smoother using also the drift times builds the final track segments. These track segments are then extrapolated back into the SCT and the Pixel Detector. Muons are reconstructed in the ID like any other charged particles; for the standalone reconstruction of muons in the muon system and the combined reconstruction, see [148].

Based on the experience gained in Run 1, several improvements to track reconstruction were made for Run 2 [149]. For example, the tracking was adapted to the new insertable B-layer (IBL) [150], and track reconstruction in dense environments (TIDE) was optimized [151]. This included an artificial neural network based approach to identify pixel clusters created by multiple charged particles. The effect of these two developments is shown in Fig. 13.5. In Fig. 13.5a the transverse impact parameter as a function of track momentum resolution is shown for data taken in 2015 at 13 TeV with the inclusion of the IBL information and for data in 2012 at 8 TeV without the IBL. The data in 2015 was collected with a minimum bias trigger. The data in 2012 is derived from a mixture of jet, tau and missing E_T triggers [150]. Figure 13.5b shows the improvement of the track reconstruction efficiency in the jet core due to the TIDE optimization [151].

ATLAS track reconstruction efficiency as a function of pseudorapidity and transverse momentum with simulated data at a center-of-mass energy of 13 TeV is shown in Fig. 13.6 [152].
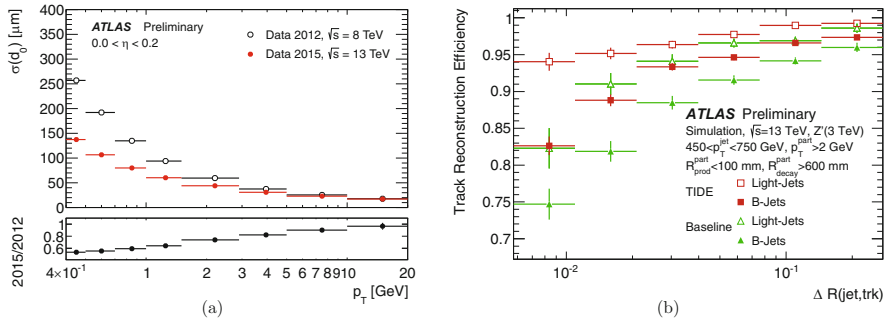

Fig. 13.5 (**a**) Upper panel: unfolded transverse impact parameter resolution measured from data in 2015 at 13 TeV with the Inner Detector including the IBL, as a function of track p_T for values of $0.0 < \eta < 0.2$, compared to that measured from data in 2012 at 8 TeV [150]; lower panel: ratio of the resolution in 2015 over the resolution in 2012. (**b**) Improvement of the track reconstruction efficiency due to the TIDE optimization, as a function of the angular distance of the particle from the jet axis. The track selection is explained in [151]
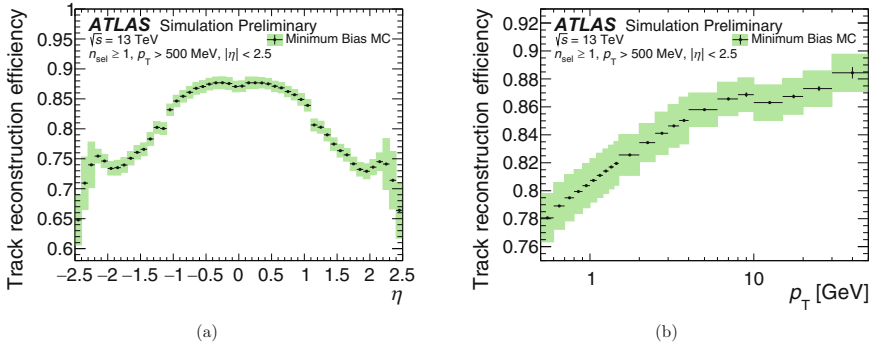
Fig. 13.6 The track reconstruction efficiency (**a**) as a function of pseudorapidity and (**b**) as a function of transverse momentum, as predicted by Pythia 8 A2 simulation. The statistical uncertainties are shown as black lines, the total uncertainties as green shaded areas [152]

13.3.3 CMS

CMS [153], together with ATLAS, is one of the two general-purpose experiments at the LHC. Its main distinguishing feature is a 3.8 T superconducting solenoid. With a length of 13 m and a diameter of 6 m, it provides a high bending power to precisely measure the momentum of charged particles. The solenoid magnetic field lines run parallel to the beam direction in the central region, where the tracking system is placed. The tracking system is designed to provide a precise and efficient measurement of particle trajectories using position-sensitive detectors. The CMS tracker is a silicon-based system [154]. It splits into two parts, the Pixel Tracker and the Strip Tracker and covers a pseudorapidity range up to $|\eta| = 2.5$. The Pixel Tracker is the innermost CMS detector sub-system and is composed of 66 million silicon pixels with dimensions $100 \times 250 \times 250 \, \mu m$, covering a total area of about $1 \, m^2$. In the barrel layers the magnetic field induces a Lorentz angle which increases charge sharing between neighbouring pixels. Charge sharing in conjunction with analog readout allows to achieve $10 \, \mu m$ position resolution for the (r, ϕ) coordinate and $15 \, \mu m$ in the z direction. The pixel detectors in the forward direction are tilted at an angle of $20°$ to induce charge sharing which allows to achieve $15 \, \mu m$ and $20 \, \mu m$ resolution respectively. This resolution is not only necessary for a precise track reconstruction, but also for the determination of both the vertices produced in the primary interaction and the decay vertices of short-lived particles.

The Strip Tracker constitutes the outer part of the tracking system. Its basic building blocks are silicon strip modules. Each module is equipped with one or two silicon sensors and a so-called Front-End hybrid containing readout electronics. In total, the CMS silicon strip tracker has 9.3 million strips and covers $198 \, m^2$ of active silicon area. The resolution in (r, ϕ) is $\simeq 30 \, \mu m$ in all layers. The inner layers of the strip tracker are equipped with double-sided sensors, one side of which is rotated by a stereo angle of 100 mrad, achieving a resolution along the z coordinate

of about 230 µm and allowing the reconstruction of the hit position in 3-D. In the outer layers the sensors are single-sided, and the z resolution can be approximated by the strip length over $\sqrt{12}$, or about 15 mm. In order to maintain excellent tracking performance until the Long Shutdown 3 of LHC, the Pixel Tracker was replaced in the year-end technical stop of 2016/2017 with a new Pixel Tracker composed of four barrel layers and six forward disks providing four-hit pixel coverage up to $|\eta| = 2.5$. After the Long Shutdown 3, the High Luminosity phase of the LHC (HL-LHC) is scheduled where the accelerator will provide an unprecedented instantaneous luminosity of $5 - 7.5 \times 10^{34}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$. In [155] the new CMS silicon tracker and its tracking and vertexing performance for different event types, pileup scenarios and detector geometries are presented.

The CMS track reconstruction algorithm is based on an iterative approach [156]. The main idea is to search for easier-to-find tracks first, to mask the hits associated to the found tracks, and to proceed to the next iteration. In this way the combinatorial problem is reduced, and the search for more difficult classes of tracks is simplified. Moreover, this approach introduces the possibility of developing special iterations that can improve track reconstruction in high-density environments such as jets, or to use the information from other subsystems such as muon chambers and calorimeters. In each iteration, the Combinatorial Track Finder is run. It can be divided into four different steps:

1. **Seed generation**: Using the information of three or four hits, the trajectory parameters and the corresponding uncertainties of the initial track candidates are computed.
2. **Track finding**: Staring from the seed, the current trajectory parameters and their uncertainties are extrapolated to the next layer and compatible hits are found. Each of them is added to a clone of the track candidate. Each of these candidates is again extrapolated to the next layer and compatible hits are found. This procedure is repeated for each candidate until there is more than one missing hit or the extrapolation does not find another tracker layer.
3. **Track fitting**: A Kalman filter or a Gaussian-sum filter/smoother is performed to obtain the final estimate of the track parameters at the interaction point exploiting the full trajectory information.
4. **Track selection**: Tracks are grouped in classes according to different track quality criteria.

As an example, the twelve tracking iterations foreseen for 2017 data taking is listed in Table 13.3 [157]. The main difference between iterations is the configuration of the seed generation and the target tracks.

Figure 13.7 shows the tracking efficiency, using a standard sample of $t\bar{t}$ events simulated with $\sqrt{s} = 13\,\mathrm{TeV}$ with different superimposed pileup conditions. The contribution of different iterations for 2017 track reconstruction is also shown as a function of the p_T of the simulated particle. It can be seen how iterations targeting low-p_T tracks are more efficient in the region between 100 and 500 MeV.

Table 13.3 List of different tracking iterations used after the Pixel Tracker upgrade with the corresponding seeding configuration used and target tracks [157]

| Iteration	Step name	Seeding	Target tracks
0	Initial	Pixel quadruplets	Prompt, high p_T
1	LowPtQuad	Pixel quadruplets	Prompt, low p_T
2	HighPtTriplet	Pixel triplets	Prompt, high p_T recovery
3	LowPtTriplet	Pixel triplets	Prompt, low p_T
4	DetachedQuad	Pixel quadruplets	From b hadron decay, $r \leq 5$ cm
5	DetachedTriplet	Pixel triplets	From b hadron decay, $r \leq 10$ cm
6	MixedTriplet	Pixel+strip triplets	Displaced, $r \leq 7$ cm
7	PixelLess	Inner strip pairs	Displaced, $r \leq 25$ cm
8	TobTec	Outer strip pairs	Displaced, $r \leq 60$ cm
9	JetCore	Pixel pairs in jets	High-p_T jets
10	Muon inside-out	Muon-tagged tracks	Muons
11	Muon outside-in	Standalone muons	Muons
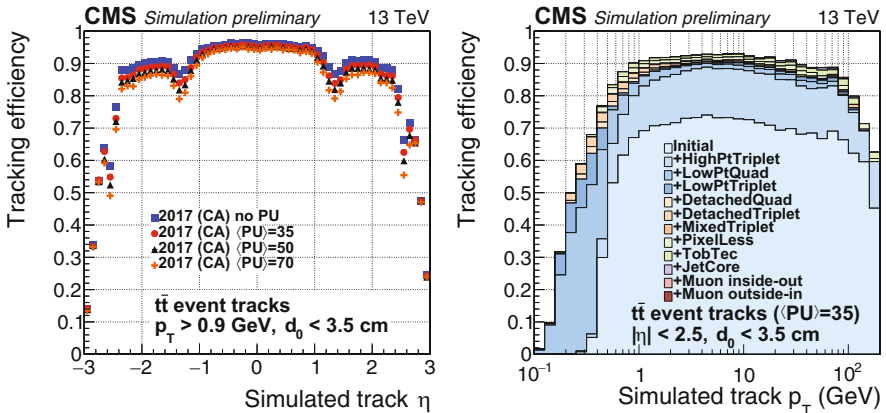
Fig. 13.7 Track reconstruction efficiency as a function of simulated track pseudorapidity for 2017 tracker at different pileup conditions (left) and cumulative contributions to the overall tracking performance from the twelve iterations in 2017 track reconstruction shown as a function of the simulated track p_T (right) [157]. The 2017 tracking reconstruction includes the Cellular Automaton-based Hit Chain-Maker (CA) seeding [158]

Figure 13.8 shows the muon tracking efficiency and the corresponding ratios between real and simulated data for 2016 collisions data coming from the Z resonance using the tag and probe method. The measured track efficiency as a function of |η| is found to be between 99.5% and 100% for the collection including all tracks. It degrades, however, with increasing number of primary vertices.
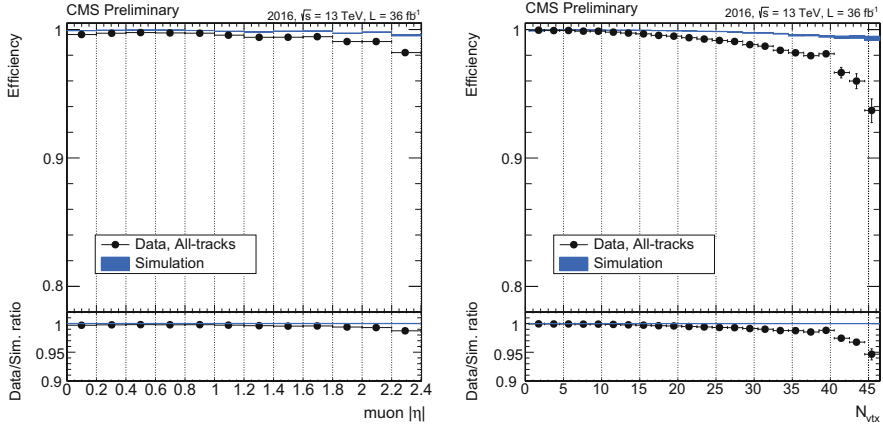
Fig. 13.8 Data (black dots) and simulation (rectangles) tracking efficiency and respective ratio for muons coming from the Z decay as a function of the absolute pseudorapidity of the probe muon (left) and the number of primary vertices (right). The data are based on an integrated luminosity of 36 fb^{-1} [155]

13.3.4 LHCb

As its name indicates, LHCb [159] focuses on physics involving bottom quarks and investigates CP violation phenomena. These studies require the measurement of the rare decays of B_d, B_s, and D mesons which are produced with a large cross-section at the LHC. Given the fact that b hadrons are predominantly produced in the forward or backward cone, the LHCb experiment is a single-arm spectrometer in contrast to the other three experiments. In order to exploit this large number of b hadrons, it requires a robust and flexible trigger and a data acquisition that allows high bandwidth data taking and provides powerful online data processing. Furthermore, superior vertex and momentum resolution are crucial to study the rapidly oscillating $B_s - \overline{B_s}$ meson system. LHCb is thus equipped with the highly sophisticated silicon microstrip detector close to the interaction point, the Vertex Locator (VELO). It can be moved to a distance of only 7 mm from the proton beams and measures the position of the primary vertices and the impact parameters of the track with extremely high precision. A further silicon microstrip detector, the Tracker Turicensis (TT) is placed before the dipole magnet. Its task is to improve the momentum resolution of reconstructed tracks and reject pairs of tracks that in reality belong to the same particle. The magnet is placed behind the TT. It bends the flight path of the particles in the $x - z$ plane and therefore allows the determination of their momenta. The tracking system is completed by the T stations (T1-T2-T3), which, together with the information from the VELO, determine the momentum and flight direction of the particles. The T stations are composed of silicon microstrip sensors close to the beam pipe and by straw tubes in the outer regions.

Track reconstruction uses hits in the VELO, TT and T stations. Depending on which detectors are crossed, different track types are defined [160, 161]:

- **Long tracks** traverse the full tracking system. They have hits in both the VELO and the T stations, and optionally in TT. They are the most important set of tracks for physics analyses.
- **Upstream tracks** pass only through the VELO and TT stations. In general their momentum is too low to traverse the magnet and reach the T stations.
- **Downstream tracks** pass only through the TT and T stations. They are important for the reconstruction of long-lived particles that decay outside the VELO acceptance.
- **VELO tracks** pass only through the VELO. These tracks are particularly important in the primary vertex reconstruction.
- **T tracks** pass only through the T stations. Like the downstream tracks, they are useful for particle identification in the Ring Imaging Cherenkov detectors.

Reconstruction of long tracks starts in the VELO. There are two complementary algorithms to add information from the downstream tracking stations to these VELO tracks. The first one combines the VELO tracks with information from the T stations. The second one combines the VELO tracks with track segments found after the magnet in the T stations, using a standalone track finding algorithm. The candidate tracks found by each algorithm are then combined, removing duplicates, to form the final set of long tracks used for analysis. Finally, hits in the TT consistent with the extrapolated trajectories of each track are added to improve their momentum determination.

Downstream tracks are found starting with T tracks, extrapolating them through the magnetic field and searching for corresponding hits in the TT. Upstream tracks are found by extrapolating VELO tracks to the TT where matching hits are then added in a procedure similar to that used by the downstream tracking. At least three TT hits are required to be present by these algorithms.

The found tracks are fitted using a Kalman filter, taking into account multiple scattering and energy loss due to ionisation. The χ^2-statistic of the fit is used to determine the quality of the reconstructed track. If two or more tracks have many hits in common, only the one with most hits is kept.

The track reconstruction efficiency for the 2012 and the 2015 data as a function of the momentum can be seen in Fig. 13.9 [162]. The results of the two periods are compatible.

Fig. 13.9 LHCb track reconstruction efficiency for the 2012 and the 2015 data as a function of the momentum. The efficiency is computed using the "Long method", described in [161]
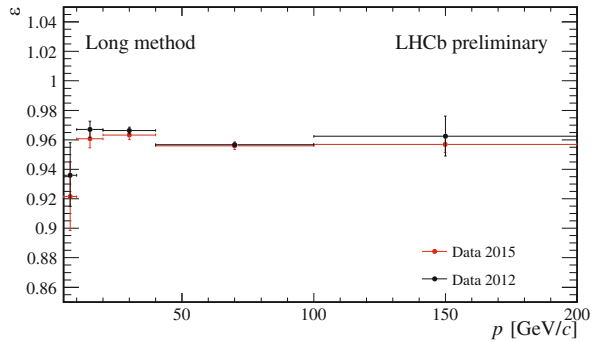
13.4 Conclusion

An overview of current methods in track and vertex reconstruction and alignment has been presented. Many of them have been developed in response to the requirements of the current experimental program at the Large Hadron Collider. The most difficult challenges are:

- Reliable reconstruction of signal events over a large background of non-signal events, pileup events, and low-momentum tracks;
- Reliable reconstruction of secondary vertices with very short distances from the primary vertex;
- Precise alignment of a large number of sensors.

Every experiment has to meet these challenges on its own terms. The outlines of the solutions found by the four major LHC experiments are described in Sect. 13.3 and, in more detail, in the references given there. In addition, the repertory of the methods discussed in this contribution can certainly not lay claim to completeness. We have tried to select widely applicable methods, thereby neglecting by necessity many experiment specific adaptations, improvements and innovations, for which we again refer to the references.

References

1. R. Mankel, "Pattern recognition and event reconstruction in particle physics experiments," *Rept. Prog. Phys.*, vol. 67, p. 553, 2004.
2. M. Hansroul, H. Jeremie, and D. Savard, "Fast circle fit with the conformal mapping method," *Nucl. Instrum. Meth.*, vol. A270, no. 2, p. 498, 1988.
3. P. Hough, "Machine analysis of bubble chamber pictures," in *Proceedings of the International Conference on High Energy Accelerators and Instrumentation*, (CERN, Geneva), p. 554, 1959.
4. H. Kälviäinen, P. Hirvonen, L. Xu, and E. Oja, "Probabilistic and non-probabilistic Hough transforms: overview and comparisons," *Image and Vision Computing*, vol. 13, no. 4, p. 239, 1995.

5. T. Alexopoulos, M. Bachtis, E. Gazis, and G. Tsipolitis, "Implementation of the Legendre Transform for track segment reconstruction in drift tube chambers," *Nucl. Instrum. Meth.*, vol. A592, pp. 456–462, 2008.

6. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci.*, vol. 79, p. 2554, 1982.

7. C. Peterson, "Track Finding With Neural Networks," *Nucl. Instrum. Meth.*, vol. A279, p. 537, 1989.

8. B. Denby, "Neural Networks and Cellular Automata in Experimental High-energy Physics," *Comput. Phys. Commun.*, vol. 49, p. 429, 1988.

9. C. Peterson and J. Anderson, "A Mean Field Theory Learning Algorithm for Neural Networks," *Complex Systems*, vol. 2, p. 995, 1987.

10. M. Diehl, M. Junger, R. Frühwirth, and J. Scherzer, "Global optimization for track finding," *Nucl. Instrum. Meth.*, vol. A389, p. 180, 1997.

11. G. Stimpfl-Abele and L. Garrido, "Fast track finding with neural nets," *Comput. Phys. Commun.*, vol. 64, p. 46, 1991.

12. S. Baginian, A. Glazov, I. Kisel, E. Konotopskaya, V. Neskoromnyi, and G. Ososkov, "Tracking by a modified rotor model of neural network," *Comput. Phys. Commun.*, vol. 79, p. 165, 1994.

13. A. Badalà, R. Barbera, G. Lo Re, A. Palmeri, G. S. Pappalardo, A. Pulvirenti, and F. Riggi, "Neural tracking in ALICE," *Nucl. Instrum. Meth.*, vol. A502, pp. 503–506, 2003.

14. A. Pulvirenti, A. Badala, R. Barbera, G. Lo Re, A. Palmeri, G. S. Pappalardo, and F. Riggi, "Neural tracking in the ALICE Inner Tracking System," *Nucl. Instrum. Meth.*, vol. A533, pp. 543–559, 2004.

15. A. Badalà, R. Barbera, G. Lo Re, A. Palmeri, G. S. Pappalardo, A. Pulvirenti, and F. Riggi, "Combined tracking in the ALICE detector," *Nucl. Instrum. Meth.*, vol. A534, p. 211, 2004.

16. M. Gyulassy and M. Harlander, "Elastic tracking and neural network algorithms for complex pattern recognition," *Comput. Phys. Commun.*, vol. 66, p. 31, 1991.

17. M. Gyulassy and M. Harlander, "High resolution multiparticle tracking without preprocessing via elastic tracking," *Nucl. Instrum. Meth.*, vol. A316, p. 238, 1992.

18. M. Ohlsson, C. Peterson, and A. Yuille, "Track finding with deformable templates: The Elastic arms approach," *Comput. Phys. Commun.*, vol. 71, p. 77, 1992.

19. M. Ohlsson, "Extensions and explorations of the elastic arms algorithm," *Comput. Phys. Commun.*, vol. 77, p. 19, 1993.

20. R. Durbin and D. Willshaw, "An analogue approach to the travelling salesman," *Nature*, vol. 326, no. 16, p. 689, 1987.

21. D. Bui, T. Greenshaw, and G. Schmidt, "A combination of an elastic net and a Hopfield net to solve the segment linking problem in the forward tracker of the H1 detector at HERA," *Nucl. Instrum. Meth.*, vol. A389, p. 184, 1997.

22. I. Kisel and V. Kovalenko, "Elastic net for broken multiple scattered tracks," *Comput. Phys. Commun.*, vol. 98, p. 45, 1996.

23. I. Kisel *et al.*, "Cellular automaton and elastic net for event reconstruction in the NEMO-2 experiment," *Nucl. Instrum. Meth.*, vol. A387, p. 433, 1997.

24. R. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, p. 35, 1960.

25. D. Catlin, *Estimation, Control, and the Discrete Kalman Filter*. NewYork: Springer, 1989.

26. R. Frühwirth, "Application of Kalman filtering to track and vertex fitting," *Nucl. Instrum. Meth.*, vol. A262, p. 444, 1987.

27. P. Billoir, "Progressive track recognition with a Kalman like fitting procedure," *Comput. Phys. Commun.*, vol. 57, p. 390, 1989.

28. R. Mankel, "A Concurrent track evolution algorithm for pattern recognition in the HERA-B main tracking system," *Nucl. Instrum. Meth.*, vol. A395, p. 169, 1997.

29. A. Glazov, I. Kisel, E. Konotopskaya, and G. Ososkov, "Filtering tracks in discrete detectors using a cellular automaton," *Nucl. Instrum. Meth.*, vol. A329, p. 262, 1993.

30. R. Frühwirth and A. Strandlie, "Application of adaptive filters to track finding," *Nucl. Instrum. Meth.*, vol. A559, p. 162, 2006.
31. A. Strandlie and R. Frühwirth, "Reconstruction of charged tracks in the presence of large amounts of background and noise," *Nucl. Instrum. Meth.*, vol. A566, p. 157, 2006.
32. A. Strandlie and R. Frühwirth, "Track and vertex reconstruction: From classical to adaptive methods," *Rev. Mod. Phys.*, vol. 82, pp. 1419–1458, May 2010.
33. H. Wind, "Evaluating a magnetic field component from boundary observations only," *Nucl. Instrum. Meth.*, vol. 84, p. 117, 1970.
34. M. Aleksa *et al.*, "Measurement of the ATLAS solenoid magnetic field," *JINST*, vol. 3, p. P04003, 2008.
35. R. Frühwirth, "Prepating Magnetic Field Measurements," *Comput. Phys. Commun.*, vol. 22, p. 223, 1981.
36. N. Amapane, V. Andreev, V. Drollinger, V. Karimäki, V. Klyukhin, and T. Todorov, "Volume-based representation of the magnetic field," in *Computing in high energy physics and nuclear physics. Proceedings, Conference, CHEP'04, Interlaken, Switzerland, September 27-October 1, 2004*, p. 310, 2004.
37. R. Frühwirth *et al.*, *Data Analysis Techniques for High-Energy Physics*. Cambridge: Cambridge University Press, 2 ed., 2000.
38. E. Lund, L. Bugge, I. Gavrilenko, and A. Strandlie, "Track parameter propagation through the application of a new adaptive Runge-Kutta-Nystroem method in the ATLAS experiment," *JINST*, vol. 4, p. P04001, 2009.
39. A. Strandlie and W. Wittek, "Derivation of Jacobians for the propagation of covariance matrices of track parameters in homogeneous magnetic fields," *Nucl. Instrum. Meth.*, vol. A566, no. 2, p. 687, 2006.
40. J. Myrheim and L. Bugge, "A Fast Runge-Kutta Method for Fitting Tracks in a Magnetic Field," *Nucl. Instrum. Meth.*, vol. 160, p. 43, 1979.
41. L. Bugge and J. Myrheim, "Tracking and Track fitting," *Nucl. Instrum. Meth.*, vol. 179, p. 365, 1981.
42. R. Frühwirth and M. Regler, "On the quantitative modelling of core and tails of multiple scattering by Gaussian mixtures," *Nucl. Instrum. Meth.*, vol. A456, no. 3, p. 369, 2001.
43. J. Jackson, *Classical Electrodynamics*. John Wiley & Sons, 2007.
44. C. Patrignani *et al.*, "Review of Particle Physics," *Chin. Phys.*, vol. C40, no. 10, p. 100001, 2016.
45. H. Eichinger and M. Regler, "Review of track-fitting methods in counter experiments," Tech. Rep. CERN-81-06, CERN, 1981.
46. P. Avery, "Applied fitting theory V: track fitting using the Kalman filter," Tech. Rep. CLEO Note CBX92-39, Cornell University, 1992.
47. V. Innocente and E. Nagy, "Trajectory fit in presence of dense materials," *Nucl. Instrum. Meth.*, vol. A324, p. 297, 1993.
48. H. Bethe and W. Heitler, "On the Stopping of fast particles and on the creation of positive electrons," *Proc. Roy. Soc. Lond.*, vol. A146, p. 83, 1934.
49. R. Frühwirth, "A Gaussian-mixture approximation of the Bethe–Heitler model of electron energy loss by bremsstrahlung," *Comput. Phys. Commun.*, vol. 154, no. 2, p. 131, 2003.
50. W. Adam, R. Frühwirth, A. Strandlie, and T. Todorov, "Reconstruction of electrons with the Gaussian-sum filter in the CMS tracker at the LHC," *J. Phys. G: Nuclear and Particle Physics*, vol. 31, no. 9, p. N9, 2005.
51. V. Kartvelishvili, "Electron bremsstrahlung recovery in ATLAS," *Nucl. Phys. Proc. Suppl.*, vol. 172, p. 208, 2007.
52. N. Chernov and G. Ososkov, "Effective Algorithms of Circle Fitting," *Comput. Phys. Commun.*, vol. 33, p. 329, 1984.
53. V. Karimäki, "Effective circle fitting for particle trajectories," *Nucl. Instrum. Meth.*, vol. A305, p. 187, 1991.
54. A. Strandlie, J. Wroldsen, R. Frühwirth, and B. Lillekjendlie, "Particle tracks fitted on the Riemann sphere," *Comput. Phys. Commun.*, vol. 131, p. 95, 2000.

55. A. Strandlie and R. Frühwirth, "Error analysis of the track fit on the Riemann sphere," *Nucl. Instrum. Meth.*, vol. A480, p. 734, 2002.
56. A. Strandlie, J. Wroldsen, and R. Frühwirth, "Treatment of multiple scattering with the generalized Riemann sphere track fit," *Nucl. Instrum. Meth.*, vol. A488, p. 332, 2002.
57. P. Laurikainen, W. Moorhead, and W. Matt, "Least squares fit of bubble chamber tracks taking into account multiple scattering," *Nucl. Instrum. Meth.*, vol. 98, p. 349, 1972.
58. C. Kleinwort, "General broken lines as advanced track fitting method," *Nucl. Instrum. Meth.*, vol. A673, p. 107, 2012.
59. P. Billoir, R. Frühwirth, and M. Regler, "Track Element Merging Strategy and Vertex Fitting in Complex Modular Detectors," *Nucl. Instrum. Meth.*, vol. A241, p. 115, 1985.
60. P. Billoir, "Track Fitting With Multiple Scattering: A New Method," *Nucl. Instrum. Meth.*, vol. A225, pp. 352–366, 1984.
61. R. Harr, "Calculation of track and vertex errors for detector design studies," *IEEE Trans. Nucl. Sci.*, vol. 42, p. 134, 1995.
62. R. Frühwirth, "Track fitting with non-Gaussian noise," *Comput. Phys. Commun.*, vol. 100, p. 1, 1997.
63. A. Strandlie and R. Frühwirth, "Discrimination between different types of material in track reconstruction with a Gaussian-sum filter," *IEEE Trans. Nucl. Sci.*, vol. 53, p. 3842, 2006.
64. R. Frühwirth and A. Strandlie, "Track fitting with ambiguities and noise: A study of elastic tracking and nonlinear filters," *Comput. Phys. Commun.*, vol. 120, p. 197, 1999.
65. A. Strandlie and R. Frühwirth, "Adaptive multitrack fitting," *Comput. Phys. Commun.*, vol. 133, p. 34, 2000.
66. R. Frühwirth, A. Strandlie, M. Winkler, and T. Todorov, "Recent results on adaptive track and multitrack fitting," *Nucl. Instrum. Meth.*, vol. A502, p. 702, 2003.
67. S. Catani, Y. Dokshitzer, M. Olsson, G. Turnock, and B. Webber, "New clustering algorithm for multi-jet cross-sections in $e^+ e^-$ annihilation," *Phys. Lett.*, vol. B269, p. 432, 1991.
68. S. Catani, Y. Dokshitzer, M. Seymour, and B. Webber, "Longitudinally invariant $k_\perp$ clustering algorithms for hadron-hadron collisions," *Nucl. Phys.*, vol. B406, p. 187, 1993.
69. S. Moretti, L. Lönnblad, and T. Sjöstrand, "New and old jet clustering algorithms for electron-positron events," *J. High Energy Phys.*, vol. 08, p. 001, 1998.
70. W. Bartel *et al.*, "Experimental Studies on Multi-Jet Production in $e^+ e^-$ Annihilation at PETRA Energies," *Z. Phys.*, vol. C33, p. 23, 1986. [,53(1986)].
71. Y. Dokshitzer, G. Leder, S. Moretti, and B. Webber, "Better jet clustering algorithms," *J. High Energy Phys.*, vol. 08, p. 001, 1997.
72. M. Seymour and C. Tevlin, "A Comparison of two different jet algorithms for the top mass reconstruction at the LHC," *J. High Energy Phys.*, vol. 11, p. 052, 2006.
73. M. Cacciari and G. Salam, "Dispelling the N^3 myth for the k_t jet-finder," *Phys. Lett.*, vol. B641, p. 57, 2006.
74. T. Sjöstrand, "The Lund Monte Carlo for e+ e- Jet Physics," *Comput. Phys. Commun.*, vol. 28, p. 229, 1983.
75. S. Bethke, Z. Kunszt, D. Soper, and W. Stirling, "New jet cluster algorithms: Next-to-leading order QCD and hadronization corrections," *Nucl. Phys.*, vol. B370, p. 310, 1992. [Erratum: Nucl. Phys.B523,681(1998)].
76. J. Dorfan, "A Cluster Algorithm for the Study of Jets in High-Energy Physics," *Z. Phys.*, vol. C7, p. 349, 1981.
77. J. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proc. Amer. Math. Soc.*, vol. 7, no. 1, p. 48, 1956.
78. L. Angelini, G. Nardulli, L. Nitti, M. Pellicoro, D. Perrino, and S. Stramaglia, "Deterministic annealing as a jet clustering algorithm in hadronic collisions," *Phys. Lett.*, vol. B601, p. 56, 2004.
79. S. Chekanov, "A New jet algorithm based on the k-means clustering for the reconstruction of heavy states from jets," *Eur. Phys. J.*, vol. C47, p. 611, 2006.
80. S.-L. Blyth *et al.*, "A Cone jet-finding algorithm for heavy-ion collisions at LHC energies," *J. Phys.*, vol. G34, p. 271, 2007.

81. F. Tarrade, "Reconstruction and identification of hadronic tau decays in ATLAS," *Nucl. Phys. Proc. Suppl.*, vol. 169, p. 357, 2007.
82. L. Barbone, N. De Filippis, O. L. Buchmüller, F. P. Schilling, T. Speer, and P. Vanlaer, "Impact of CMS silicon tracker misalignment on track and vertex reconstruction," *Nucl. Instrum. Meth.*, vol. A566, p. 45, 2006.
83. S. Blusk, O. Buchmüller, A. Jacholkowski, T. Ruf, J. Schieck, and S. Viret, eds., *Proceedings of the first LHC Detector Alignment Workshop, CERN, Geneva, Switzerland, 4–6 September 2006*, 2007.
84. V. Blobel, "Software alignment for tracking detectors," *Nucl. Instrum. Meth.*, vol. A566, p. 5, 2006.
85. W. Wiedenmann, "Alignment of the ALEPH tracking devices," *Nucl. Instrum. Meth.*, vol. A323, p. 213, 1992.
86. A. Andreazza and E. Piotto, "The Alignment of the DELPHI Tracking Detectors," Tech. Rep. DELPHI 99–153 TRACK 94, CERN, 1999.
87. A. Sopczak, "Alignment of the central D0 Detector," *Nucl. Instrum. Meth.*, vol. A566, p. 142, 2006.
88. Y. Fisyak *et al.*, "Overview of the inner silicon detector alignment procedure and techniques in the RHIC/STAR experiment," *J. Phys. Conf. Ser.*, vol. 119, p. 032017, 2008.
89. D. Brown, A. Gritsan, Z. Guo, and D. Roberts, "Local Alignment of the BABAR Silicon Vertex Tracking Detector," *Nucl. Instrum. Meth.*, vol. A603, p. 467, 2009.
90. P. Schleper, G. Steinbrück, and M. Stoye, "Alignment of the CMS silicon tracker using Millepede II," *J. Phys. Conf. Ser.*, vol. 119, p. 032040, 2008.
91. M. Gersabeck, "Alignment of the LHCb Vertex Locator," *Nucl. Instrum. Meth.*, vol. A598, p. 71, 2009.
92. S. Gonzalez-Sevilla, "Track-based alignment of the ATLAS inner detector," *J. Phys. Conf. Ser.*, vol. 119, p. 032019, 2008.
93. E. Widl and R. Frühwirth, "A large-scale application of the Kalman alignment algorithm to the CMS tracker," *J. Phys. Conf. Ser.*, vol. 119, p. 032038, 2008.
94. S. Chatrchyan *et al.*, "Alignment of the CMS tracker with LHC and cosmic ray data," *JINST*, vol. 9, p. P06009, 2014.
95. G. Mittag, "Alignment of the CMS Tracker: Latest results from LHC Run-II," *J. Phys. Conf. Ser.*, vol. 898, no. 4, p. 042014, 2017.
96. J. Schieck, "Track-based alignment for the ATLAS Inner Detector Tracking System," *JINST*, vol. 7, p. C01012, 2012.
97. G. Ripellino, "The alignment of the ATLAS Inner Detector in Run-2," *PoS*, vol. LHCP2016, p. 196, 2016.
98. M. Martinelli, "Novel real-time alignment and calibration of the LHCb detector in Run2," *J. Phys. Conf. Ser.*, vol. 898, no. 3, p. 032039, 2017.
99. J. Amoraal, "Alignment of the LHCb detector with Kalman filter fitted tracks," *J. Phys. Conf. Ser.*, vol. 219, p. 032028, 2010.
100. A. Dainese, "Alignment of the ALICE tracking detectors," *PoS*, vol. VERTEX2009, p. 021, 2009.
101. V. Karimäki, T. Lampen, and F. Schilling, "The HIP algorithm for track based alignment and its application to the CMS pixel detector," Tech. Rep. CMS-NOTE-2006-018, CERN, 2006.
102. V. Blobel and C. Kleinwort, "A New method for the high precision alignment of track detectors," in *Advanced Statistical Techniques in Particle Physics. Proceedings, Conference, Durham, UK, March 18–22, 2002*, 2002.
103. Y. Saad and M. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM Journal on scientific and statistical computing*, vol. 7, no. 3, p. 856, 1986.
104. E. Widl, R. Frühwirth, and W. Adam, "A Kalman filter for track-based alignment," tech. rep., CERN, 2006.
105. R. Frühwirth, T. Todorov, and M. Winkler, "Estimation of detector alignment parameters using the Kalman filter with annealing," *J. Phys.*, vol. G29, p. 561, 2003.

106. R. Gluckstern, "Uncertainties in track momentum and direction, due to multiple scattering and measurement errors," *Nucl. Instrum. Meth.*, vol. 24, p. 381, 1963.

107. M. Regler and R. Frühwirth, "Generalization of the Gluckstern formulas. I: Higher orders, alternatives and exact results," *Nucl. Instrum. Meth.*, vol. A589, p. 109, 2008.

108. R. Frühwirth and A. Beringer, "JDOT — a Java detector optimization tool," in *Nuclear Science Symposium Conference Record, 2008. NSS'08. IEEE*, pp. 3483–3487, IEEE, 2008.

109. W. Innes, "Some formulas for estimating tracking errors," *Nucl. Instrum. Meth.*, vol. A329, p. 238, 1992.

110. W. Innes, "TRACKERR: A Program for calculating tracking errors," Tech. Rep. SLAC-BABAR-NOTE-121, SLAC, 1993.

111. M. Regler, M. Valentan, and R. Frühwirth, "The LiC detector toy program," *Nucl. Instrum. Meth.*, vol. A581, p. 553, 2007.

112. S. Masciocchi, "Experience with HERA-B vertexing," *Nucl. Instrum. Meth.*, vol. A462, p. 220, 2001.

113. E. Chabanat, J. D'Hondt, N. Estre, R. Frühwirth, K. Prokofiev, T. Speer, P. Vanlaer, and W. Waltenberger, "Vertex reconstruction in CMS," *Nucl. Instrum. Meth.*, vol. A549, p. 188, 2005.

114. W. Erdmann, "Vertexing in the H1 experiment," *Nucl. Instrum. Meth.*, vol. A560, p. 89, 2006.

115. M. Costa, "Vertex and track reconstruction in ATLAS," *Nucl. Instrum. Meth.*, vol. A582, p. 785, 2007.

116. W. Adam, "Track and vertex reconstruction in CMS," *Nucl. Instrum. Meth.*, vol. A582, p. 781, 2007.

117. W. Waltenberger, *Development of vertex finding and vertex fitting algorithms for CMS*. PhD thesis, TU Wien, 2004.

118. D. Jackson, "A Topological vertex reconstruction algorithm for hadronic jets," *Nucl. Instrum. Meth.*, vol. A388, p. 247, 1997.

119. S. Hillert, "ZVMST: A Minimum spanning tree-based vertex finder," tech. rep., 2008.

120. W. Waltenberger and F. Moser, "RAVE — an Open, Extensible, Detector-Independent Toolkit for Reconstruction of Interaction Vertices," in *IEEE Nuclear Science Symposium Conference Record 2006*, vol. 1, p. 104, IEEE, 2006.

121. W. Waltenberger, W. Mitaroff, and F. Moser, "RAVE — a Detector-independent vertex reconstruction toolkit," *Nucl. Instrum. Meth.*, vol. A581, p. 549, 2007.

122. G. Patrick and B. Schorr, "Vertex Fitting of Several Helices in Space," *Nucl. Instrum. Meth.*, vol. A241, p. 132, 1985.

123. P. Billoir and S. Qian, "Fast vertex fitting with a local parametrization of tracks," *Nucl. Instrum. Meth.*, vol. A311, p. 139, 1992.

124. P. Billoir and S. Qian, "Erratum to Fast vertex fitting with a local parametrization of tracks," *Nucl. Instrum. Meth.*, vol. A350, p. 624, 1994.

125. D. Bates and D. Watts, *Nonlinear regression analysis and its applications*. Wiley & Sons, 1988.

126. V. Karimäki, "Effective Vertex Fitting," Tech. Rep. CMS-NOTE-1997-051, CERN, 1997.

127. E. Calligarich, R. Dolfini, M. Genoni, and A. Rotondi, "A Fast algorithm for vertex estimation," *Nucl. Instrum. Meth.*, vol. A311, p. 151, 1992.

128. F. Hampel, E. Ronchetti, P. Rousseeuw, and W. Stahel, *Robust statistics: the approach based on influence functions*. John Wiley & Sons, 2011.

129. P. Rousseeuw and A. Leroy, *Robust regression and outlier detection*. John Wiley & sons, 2005.

130. R. Frühwirth, P. Kubinec, W. Mitaroff, and M. Regler, "Vertex reconstruction and track bundling at the LEP collider using robust algorithms," *Comput. Phys. Commun.*, vol. 96, p. 189, 1996.

131. P. Huber and E. Ronchetti, *Robust statistics*. Wiley & Sons, 2 ed., 2009.

132. G. Agakichiev *et al.*, "A new robust fitting algorithm for vertex reconstruction in the CERES experiment," *Nucl. Instrum. Meth.*, vol. A394, p. 225, 1997.

133. K. Rose, "Deterministic annealing for clustering, compression, classification, regression, and related optimization problems," *Proceedings of the IEEE*, vol. 86, no. 11, p. 2210, 1998.

134. T. Speer, R. Frühwirth, P. Vanlaer, and W. Waltenberger, "Robust vertex fitters," *Nucl. Instrum. Meth.*, vol. A566, p. 149, 2006.

135. W. Waltenberger, R. Frühwirth, and P. Vanlaer, "Adaptive vertex fitting," *J. Phys.*, vol. G34, p. N343, 2007.

136. A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (methodological)*, vol. 39, no. 1, p. 1, 1977.

137. R. Frühwirth and W. Waltenberger, "Redescending M-estimators and deterministic annealing," *Austrian J. Statistics*, vol. 37, no. 3&4, p. 301, 2008.

138. J. D'Hondt, R. Frühwirth, P. Vanlaer, and W. Waltenberger, "Sensitivity of robust vertex fitting algorithms," *IEEE Transactions on Nuclear Science*, vol. 51, no. 5, p. 2037, 2004.

139. R. Frühwirth and W. Waltenberger, "Adaptive Multi-vertex fitting," Tech. Rep. CMS CR 2004/062, CERN, 2004.

140. T. Speer and R. Frühwirth, "A Gaussian-sum filter for vertex reconstruction," *Comput. Phys. Commun.*, vol. 174, p. 935, 2006.

141. S. Lang, *Calculus of several variables*. Springer Science & Business Media, 2012.

142. P. Avery, "Applied fitting theory VI: Formulas for kinematic fitting," Tech. Rep. CLEO Note CBX98-37, Cornell University, 1999.

143. ALICE Collaboration, "The ALICE experiment at the CERN LHC," *JINST*, vol. 3, no. 08, p. S08002, 2008.

144. B. Abelev *et al.*, "Performance of the ALICE Experiment at the CERN LHC," *Int. J. Mod. Phys.*, vol. A29, p. 1430044, 2014.

145. G. Aad *et al.*, "The ATLAS Experiment at the CERN Large Hadron Collider," *JINST*, vol. 3, p. S08003, 2008.

146. T. Cornelissen *et al.*, "Concepts, Design and Implementation of the ATLAS New Tracking (NEWT)," Tech. Rep. ATL-SOFT-PUB-2007-007, CERN, Geneva, 2007.

147. ATLAS Collaboration, "Performance of the ATLAS Track Reconstruction Algorithms in Dense Environments in LHC Run 2," *Eur. Phys. J.*, vol. C77, no. 10, p. 673, 2017.

148. G. Aad *et al.*, "Muon reconstruction performance of the ATLAS detector in proton–proton collision data at $\sqrt{s}$ =13 TeV," *Eur. Phys. J.*, vol. C76, no. 5, p. 292, 2016.

149. H. Oide, "Improvements to ATLAS track reconstruction for Run-2," *PoS*, vol. EPS-HEP2015, p. 287, 2015.

150. K. Potamianos, "The upgraded Pixel detector and the commissioning of the Inner Detector tracking of the ATLAS experiment for Run-2 at the Large Hadron Collider," *PoS*, vol. EPS-HEP2015, p. 261, 2015.

151. ATLAS Collaboration, "The Optimization of ATLAS Track Reconstruction in Dense Environments," Tech. Rep. ATL-PHYS-PUB-2015-006, CERN, Geneva, Mar 2015.

152. G. Aad *et al.*, "Charged-particle distributions in $\sqrt{s}$ = 13 TeV pp interactions measured with the ATLAS detector at the LHC," *Phys. Lett.*, vol. B758, pp. 67–88, 2016.

153. S. Chatrchyan *et al.*, "The CMS Experiment at the CERN LHC," *JINST*, vol. 3, p. S08004, 2008.

154. CMS Collaboration, "The CMS tracker system project: Technical Design Report," Tech. Rep. CMS-TDR-005, CERN, 1997.

155. E. Brondolin, *Track reconstruction in the CMS experiment for the High Luminosity LHC*. PhD thesis, Technische Universität Wien, 2018.

156. S. Chatrchyan *et al.*, "Description and performance of track and primary-vertex reconstruction with the CMS tracker," *JINST*, vol. 9, no. 10, p. P10009, 2014.

157. CMS Collaboration, "2017 tracking performance plots," Tech. Rep. CMS-DP-2017-015, CERN, 2017.

158. F. Pantaleo, *New Track Seeding Techniques for the CMS Experiment*. PhD thesis, Universität Hamburg, 2017.

159. A. Augusto Alves *et al.*, "The LHCb Detector at the LHC," *JINST*, vol. 3, p. S08005, 2008.

160. R. Aaij *et al.*, "LHCb Detector Performance," *Int. J. Mod. Phys.*, vol. A30, no. 07, p. 1530022, 2015.
161. LHCb Collaboration, "Measurement of the track reconstruction efficiency at LHCb," *JINST*, vol. 10, no. 02, p. P02007, 2015.
162. LHCb Collaboration. Published online at https://twiki.cern.ch/twiki/pub/LHCb/ ConferencePlots/TrackEffPLong2015_2012.pdf, 2017.