A new method for the high-precision alignment of track detectors

Contribution to the Conference

Advanced Statistical Techniques in Particle Physics

Durham, 18 – 22 March 2002

Volker Blobel, Institut für Experimentalphysik, Univ. Hamburg
Claus Kleinwort, Deutsches Elektronensynchrotron DESY, Hamburg

Abstract

Track detectors in high energy physics experiments require an accurate determination of a large number of alignment parameters. A method has been developed, which allows the determination of up to several thousand alignment parameters in a simultaneous linear least squares fit of an arbitrary number of tracks.

The method is general for problems where global parameters (e.g. from alignment) and independent sets of local parameters (e.g track parameters like angles and curvatures) appear simultaneously. Constraints between global parameters can be incorporated. The huge matrix system of the normal equations is reduced to the size corresponding to the global parameters, which allows the exact solution for the global parameters. A program has been developed which allows a certain fraction of outliers in the data.

The sensitivity of the method is demonstrated in an example of the alignment of a 56-plane drift chamber and a 2-plane silicon tracker. In this example about 1400 alignment parameters are determined in a fit of about fifty thousand tracks.

1 Alignment problems of track detectors

Alignment problems for large detectors, especially track detectors, in particle physics often require the determination of a large number of alignment parameters, typically of the order of 100 to 1000. Alignment parameters for example define the accurate space coordinates and orientation of detector components. In addition driftchambers for example require the determination of the drift velocity v_{drift} and of local variations of the drift velocity, and of the Lorentz angle.
In the alignment usually special alignment measurements are combined with data of particle reaction, typically tracks from physics interactions and from cosmics. In this paper the alignment parameters are called *global* parameters; they contribute to all the data. Usually these parameters are *corrections* to default values and are thus relatively small, and the value zero is the initial approximation. Parameter sets valid for a single track are called *local* parameters, examples are track slopes and curvatures.

An approximate alignment method is to perform least squares fits on the data *ignoring* initially the alignment parameters. The *deviations* between the fitted and measured data, the *residuals*, are then used to determine the alignment parameters afterwards. Single fits depend only on the small number of local parameters like slope or curvature of the track, and are easy to perform.
This approximate method however is not a statistically correct method, because the (local) fits depend on a wrong model (namely: ignoring the global parameters), and the result are biased. In addition the results may be unstable due to detector inefficiencies. The adjustment of parameters based on the observed (biased) residuals will then result in biased alignment parameters. If these alignment parameters are applied as corrections in repeated fits, the remaining residuals will be reduced, as desired. However, the fitted parameters will still be biased, even more, if this approximate procedure is applied iteratively. The simple residual method is also unable to determine correlated parameters, for example a geometrical shift of a detector position and a shift in drift velocity in a drift chamber.

A better method is an overall fit, with all the (global) alignment parameters and local parameters, perhaps from thousands or millions of events, determined simultaneously. The most important general method for the determination of parameters from measured data is the linear least squares model. It is usually stable, fast and accurate, requires no initial values of parameters and is able to take into account all the correlations between different parameters, even if there are many parameters. There is of course a practical limit in the number of parameters in simultaneous fits, because of space and time limitations. In an alignment fit the size of the vectors and matrices in the least squares solution will be large, of the order of 10^4 or 10^6. With standard methods such a solution is impossible.
In this paper a special method of solution for this kind of problems is derived. Due to the special structure of the matrices with one set of global parameters and many sets of local parameters the problem can in fact be reduced to a solvable size, without making any approximations. The method is realized in a program called **Millepede** [2].

2 Standard least squares

2.1 Linear models

In a linear model the measured quantity z_k has an expectation value which is a *linear* function of the parameters $\boldsymbol{a}$. The difference between the measured value and the expectation value is the residuum r_k:

$$z_k = \boldsymbol{a}^T \cdot \boldsymbol{d}_k + r_k \tag{1}$$

The measured data are assumed to be *independent*; the result of a measurement z_k does not dependent on any other measurement. Thus the covariance matrix of the measured data is a diagonal matrix, which simplifies the treatment. The accuracy of the measurement z_k is given by the variance σ_k^2 or the standard deviation σ_k. It is assumed, that the precision is at least approximately known.
The solution of the method of least squares is determined by the minimum of the weighted sum of the square of the residuals

$$S(\boldsymbol{a}) = \sum_k \frac{\left(z_k - \boldsymbol{a}^T \cdot \boldsymbol{d}_k\right)^2}{\sigma_k^2} \tag{2}$$

where the sum (Index k) is over all measurements. This sum is a quadratic function of the parameters $\boldsymbol{a}$, and the solution is obtained from a set of linear equations (so-called normal equations of least squares), which can be written in matrix form

$$\boldsymbol{C}\boldsymbol{a} = \boldsymbol{b} \tag{3}$$

with a symmetric n-by-n matrix $\boldsymbol{C}$ and a right-hand-side n-vector $\boldsymbol{b}$, which are sums with (because the measured data are independent) *independent* contributions from each measurement:

$$\boldsymbol{C} = \sum_k w_k \left(\boldsymbol{d}_k \cdot \boldsymbol{d}_k^T\right) \tag{4}$$

and

$$\boldsymbol{b} = \sum_k w_k z_k \boldsymbol{d}_k \; . \tag{5}$$

There is no limitation on the number of measured data; it is not necesary to keep all input data in memory, because the complete information from a single measurement is contained in the contributions to $\boldsymbol{C}$ and $\boldsymbol{b}$, which have a fixed size. The solution vector is written in the form

$$\boldsymbol{a} = \boldsymbol{C}^{-1}\boldsymbol{b} \tag{6}$$

with the inverse of the matrix $\boldsymbol{C}$, which is, by error propagation, equal to the covariance matrix of the parameters $\boldsymbol{a}$. It should be remarked, that different techniques are possible, if the complete input is available in matrix form in memory; this would allow to apply the method of singular value decomposition (SVD), which allows to control the solution in an optimal way.

What are the limits for the number n of parameters, if the matrix equation is solved by standard methods? The symmetry property of the matrix $\boldsymbol{C}$ may be used to store only the $(n^2+n)/2$ elements on and above the diagonal, and also to reduce the computer time to almost half the time for a non-symmetric matrix. Matrix inversion is an explicit process and requires a CPU time proportional to n^3 for a n-by-n matrix. For a 1000-by-1000 symmetric matrix in double precision (about 4 Mbyte) a CPU time of a few minutes (on a Pentium 266 MHz CPU) is required. The time ($\propto n^3$) and space ($\propto n^2$) requirements imply at present a practical limit[1] of the size of the matrix (and number of parameters) between $n = 1\,000$ and $n = 10\,000$.

The accuracy of the computation of the inverse matrix is a critical point, and often computation in double precision is required, especially for large matrices. The accuracy is generally low for a bad condition number of the matrix (the condition number is defined as the ratio of the largest to the smallest eigenvalue of the matrix). From the interpretation of the inverse matrix as covariance matrix it is clear, that the condition number will be large if the correlation between parameters is high. The matrix is singular (one or more eigenvalues are zero) in case of a complete correlation between parameters. In this case and in cases of missing data (e.g. dead channels in an alignment fit) a standard matrix inversion program will fail.
A singular value decomposition would allow to recognize the cases of zero or small eigenvalues, which would *destroy* the full solution. The standard method in case of zero or very small eigenvalues is to set their inverse to zero; this means essentially to remove parameters or linear combinations of parameters from the solution (i.e. the resulting parameter values are zero); if the parameters are in fact *corrections* this is clearly acceptable. In the program **Millepede** [2] a similar, but simpler strategy is used to avoid the effects of a bad condition number. This strategy also works in cases, where due to missing input data[2] one or more parameters are not defined; the parameter will become zero in this case. The strategy allows in addition to remove certain parameters from the least squares fit without the necessity to reorganize the numbering.

2.2 Solution by partitioning

The special structure of the matrix $\boldsymbol{C}$ to be inverted may allow to move the limit of solvable problems further (see also [1]). Below the matrix $\boldsymbol{C}$ is partitioned into submatrices and the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ are partitioned into two subvectors; then the matrix equation becomes

$$\left(\begin{array}{c|c} \boldsymbol{C}_{11} & \boldsymbol{C}_{12} \\ \hline \boldsymbol{C}_{21} & \boldsymbol{C}_{22} \end{array}\right) \left(\begin{array}{c} \boldsymbol{a}_1 \\ \hline \boldsymbol{a}_2 \end{array}\right) = \left(\begin{array}{c} \boldsymbol{b}_1 \\ \hline \boldsymbol{b}_2 \end{array}\right) \tag{7}$$

[1] For $n = 10\,000$ the required space for the matrix is about 400 Mbyte and the CPU time for inversion is a few days.
[2] Large detectors may have dead channels, which means missing data.

where the submatrix $\boldsymbol{C}_{11}$ is a p-by-p square matrix and the submatrix $\boldsymbol{C}_{22}$ is a q-by-q square matrix, with $p+q=n$.

If the sub-vector $\boldsymbol{a}_1$ would not exist, the solution for the sub-vector $\boldsymbol{a}_2$ would be defined by the matrix equation

$$\boldsymbol{C}_{22}\,\boldsymbol{a}_2^* = \boldsymbol{b}_2 \tag{8}$$

where the star indicates the special character of this solution, which is

$$\boldsymbol{a}_2^* = \boldsymbol{C}_{22}^{-1}\,\boldsymbol{b}_2\ . \tag{9}$$

Only the q-by-q sub-matrix $\boldsymbol{C}_{22}$ has to be inverted. The solution $\boldsymbol{a}_2^*$ of course differs from the final solution $\boldsymbol{a}_2$ and is called the *local* solution.
Now, having inverted the sub-matrix $\boldsymbol{C}_{22}$, the submatrix of the complete inverse matrix $\boldsymbol{C}$ correponding to the upper left part $\boldsymbol{C}_{11}$, called $\boldsymbol{B}$, can be obtained by the formula

$$\boxed{\boldsymbol{B} = \left(\boldsymbol{C}_{11} - \boldsymbol{C}_{12}\boldsymbol{C}_{22}^{-1}\boldsymbol{C}_{12}^T\right)^{-1}} \tag{10}$$

which requires in addition to some matrix products the calculation of the inverse of a p-by-p sub-matrix. With this matrix $\boldsymbol{B}$ the solution of the whole matrix equation can written in the form

$$\left(\begin{array}{c} \boldsymbol{a}_1 \\ \hline \boldsymbol{a}_2 \end{array}\right) = \left(\begin{array}{c|c} \boldsymbol{B} & -\boldsymbol{B}\boldsymbol{C}_{12}\boldsymbol{C}_{22}^{-1} \\ \hline -\boldsymbol{C}_{22}^{-1}\boldsymbol{C}_{12}^T\boldsymbol{B} & \boldsymbol{C}_{22}^{-1} - \boldsymbol{C}_{22}^{-1}\boldsymbol{C}_{12}^T\boldsymbol{B}\boldsymbol{C}_{12}\boldsymbol{C}_{22}^{-1} \end{array}\right) \left(\begin{array}{c} \boldsymbol{b}_1 \\ \hline \boldsymbol{b}_2 \end{array}\right) \tag{11}$$

which can be used to obtain the complete result ($\boldsymbol{a}_1$ and $\boldsymbol{a}_2$). Note that to solve the matrix equation two smaller matrices, a symmetric p-by-p and a symmetric q-by-q matrix, have to be inverted.
The sub-vector $\boldsymbol{a}_1$ is obtained by the product

$$\boldsymbol{a}_1 = \boldsymbol{B}\boldsymbol{b}_1 - \boldsymbol{B}\boldsymbol{C}_{12}\underbrace{\boldsymbol{C}_{22}^{-1}\boldsymbol{b}_2}_{\boldsymbol{a}_2^*} \tag{12}$$

which can be simplified by using the special local solution $\boldsymbol{a}_2^*$ to

$$\boxed{\boldsymbol{a}_1 = \boldsymbol{B}\left(\boldsymbol{b}_1 - \boldsymbol{C}_{12}\boldsymbol{a}_2^*\right)}\ . \tag{13}$$

If the interest is the determination of this sub-vector $\boldsymbol{a}_1$ only, while the sub-vector $\boldsymbol{a}_2$ is not needed, then only the boxed equations (10) and (13) have to be solved. Some computer time can be saved by this method, especially if the matrix $\boldsymbol{C}_{22}^{-1}$ is easily calculated or already known before. In special cases the method can be applied repeatedly, and therefore problems with a huge number of parameters can be solved in practice. The method is not an approximation, but is exact and it takes into account all the correlations introduced by the removed parameters. This type of application is discussed in more detail in the next chapter.

It seems that this possibility of removing unnecessary parameters, but still getting the correct solution for the remaining parameters in least squares problems was already known and has been used in the nineteenth century; one example is [3].

2.3 Constraints

If there are explicit relations between the parameters of a least squares fit, then they should be explicitly taken into account in the solution. The case of a single linear constraint between parameters of the form

$$\boldsymbol{a}^T \cdot \boldsymbol{f} = f_0 \tag{14}$$

is assumed here. The Lagrange multiplier method is the standard method to include constraints in a least squares solution; one additional parameter λ is introduced for each constraint. The equation (3) is modified by the additional parameters to the form

$$\left(\begin{array}{c|c} \boldsymbol{C} & \boldsymbol{f} \\ \hline \boldsymbol{f}^T & \boldsymbol{0} \end{array}\right) \left(\begin{array}{c} \boldsymbol{a} \\ \hline \lambda \end{array}\right) = \left(\begin{array}{c} \boldsymbol{b} \\ \hline f_0 \end{array}\right)\ , \tag{15}$$

where the matrix on the left hand side is still symmetric. This shows that *linear* least squares problems with *linear* constraints can be solved direcly, without iterations and without the need for initial values of the parameters.

3 Least squares with global and local parameters

3.1 Local parameters

Local parameters are denoted by $\alpha_j,\ j = 1, 2, \dots \nu$. A set of measurements z is considered. A single value z from the set is ideally given by a linear expression $z = \alpha_1 \cdot \delta_1 + \alpha_2 \cdot \delta_2 + \dots \alpha_\nu \cdot \delta_\nu$, depending on the parameters α_j and on (known) constant factors δ_j. More complete, the dependence can be written as

$$z_k = \alpha_1 \cdot \delta_{1k} + \alpha_2 \cdot \delta_{2k} + \dots \alpha_\nu \cdot \delta_{\nu k} = \sum_{j=1}^{\nu} \alpha_j \cdot \delta_{jk} \ , \tag{16}$$

with an additional index k indicating the k-th measured value.
As an example the measurement of a straight particle track is considered, where each z_k is a space measurement at plane k. The local parameters are intercept (α_1) and slope (α_2) of the particle track. If the distance of the plane is denoted by S_k, the linear track model can be written as

$$z_k = \alpha_1 \cdot 1 + \alpha_2 \cdot S_k \ .$$

From a least squares fit of the model to the set of measured z-values the local parameters intercept and slope can be determined. The formalism of least squares allows to take into account the different precision of the different measurements, by assigning a weight to each measurement

$$\text{weight (determined from the standard deviation } \sigma\text{)} \qquad\qquad w_k = \frac{1}{\sigma_k^2}$$

in the least squares formalism.
The least squares solution, minimizing the weighted sum of squared residuals, is defined by the so-called normal equations, which can be written in terms of matrices

$$\boldsymbol{\Gamma}\,\boldsymbol{\alpha} = \boldsymbol{\beta} \,, \qquad \text{solved by} \quad \boldsymbol{\alpha} = \boldsymbol{\Gamma}^{-1}\boldsymbol{\beta} \,. \tag{17}$$

The elements of the matrix $\boldsymbol{\Gamma}$ and the vector $\boldsymbol{\beta}$ are formed by sums

$$\Gamma_{ij} = \sum_k w_k \cdot \delta_{ik}\delta_{jk} \qquad\qquad \beta_j = \sum_k w_k \cdot z_k \delta_{jk} \ . \tag{18}$$

These formulas are valid in the case of *uncorrelated* z-measurements. Uncorrelated measurement are assumed thoughout this paper. The inverse matrix $\boldsymbol{\Gamma}^{-1}$ is the covariance matrix of the parameter vector $\boldsymbol{\alpha}$.

3.2 Global parameters

Now global parameters are considered, which contribute to all the measurements. They are assumed to be *alignment parameters* here, although the formalism is general and can also be applied to other problems. Alignment parameters are defined to represent *corrections* to ideal (design) values; this parametrization adds further terms to the equations (16)

$$z = \underbrace{a_1 \cdot d_1 + a_2 \cdot d_2 + \dots a_n \cdot d_n}_{\text{global parameters}} + \underbrace{\alpha_1 \cdot \delta_1 + \alpha_2 \cdot \delta_2 + \dots \alpha_\nu \cdot \delta_\nu}_{\text{local parameters}} = \sum_{i=1}^{n} a_j \cdot d_j + \sum_{j=1}^{\nu} \alpha_j \cdot \delta_j \ , \tag{19}$$

Usually only few terms containing global parameters are nonzero for a single z measurement.
In the following it is assumed that, as written in equation (19), the dependence on the global parameters is *linear*. A non-linear relationship for the global parameters would require iterations, starting with reasonable initial values for the parameters, and assuming a linearized expression in each iteration; these are standard methods for nonlinear least squares problems.

3.3 The simultaneous fit of global and local parameters

In the following it is assumed that there is a set of N partial measurements. Each partial measurement, with index i, depends on local parameters $\boldsymbol{\alpha}_i$, and all of them depend on the global parameters. In a simultaneous fit of all global parameters plus local parameters from N subsets of the data there are in total $(n+N\cdot\nu)$ parameters, and the standard solution requires the solution of $(n+N\cdot\nu)$ equations with a computation time proportional to $(n+N\cdot\nu)^3$. Below it is shown, that the problem can be reduced to n equations with a computation time proportional n^3.

Generalizing the formalism of equations (16) to the complete measurement, starting from equations of the type of (19), one obtains a system of normal equations with large dimensions, as is shown in equation (20).

The matrix on the left side of equation (20) has, from each partial measurement, three types of contributions. The first part is a contribution of a symmetric matrix $\boldsymbol{C}_i$, of dimension n (number of global parameters). All the matrices $\boldsymbol{C}_i$ are added up in the upper left corner of the big matrix of the normal equations. The second contribution is the symmetric matrix $\boldsymbol{\Gamma}_i$ (compare equation (18)), which gives a contribution to the big matrix on the diagonal and is depending only on the i-th partial measurement. The third contribution is a rectangular matrix $\boldsymbol{G}_i$, with a row number of n (global) and a column number of ν (local).

There are two contributions to the vector of the normal equations, $\boldsymbol{b}_i$ for the global and $\boldsymbol{\beta}_i$ for the local parameters.

The complete matrix equation is given by

$$\left(\begin{array}{c||c|c|c} \sum \boldsymbol{C}_i & \cdots & \boldsymbol{G}_i & \cdots \\ \hline\hline \vdots & \ddots & 0 & 0 \\ \hline \boldsymbol{G}_i^T & 0 & \boldsymbol{\Gamma}_i & 0 \\ \hline \vdots & 0 & 0 & \ddots \end{array}\right) \cdot \left(\begin{array}{c} \boldsymbol{a} \\ \hline\hline \vdots \\ \hline \boldsymbol{\alpha}_i \\ \hline \vdots \end{array}\right) = \cdot \left(\begin{array}{c} \sum \boldsymbol{b}_i \\ \hline\hline \vdots \\ \hline \boldsymbol{\beta}_i \\ \hline \vdots \end{array}\right) \tag{20}$$

In this matrix equation the matrices $\boldsymbol{C}_i$, $\boldsymbol{\Gamma}_i$, $\boldsymbol{G}_i$ and the vectors $\boldsymbol{b}_i$ and $\boldsymbol{\beta}_i$ contain contributions from the i-th partial measurement. Ignoring the global parameters one could solve the normal equations $\boldsymbol{\Gamma}_i\boldsymbol{\alpha}_i = \boldsymbol{\beta}_i$ for each partial measurement separately by

$$\boldsymbol{\alpha}_i = \boldsymbol{\Gamma}_i^{-1}\boldsymbol{\beta}_i\,. \tag{21}$$

The complete system of normal equations has a special structure, with many vanishing sub-matrices. The only connection between the local parameters of different partial measurements is given by the sub-matrices $\boldsymbol{G}_i$ und $\boldsymbol{C}_i$,

The aim of the fit is solely to determine the global parameters; final best parameters of the local parameters are not needed. The matrix of equation (20) is written in a partitioned form. The general solution can also be written in partitioned form. Many of the sub-matrices of the huge matrix in equation (20) are zero and this has the effect, that the formulas for the sub-matrices of the inverse matrix are very simple.

By this procedure the n normal equations

$$\left(\quad \boldsymbol{C'} \quad\right)\left(\ \boldsymbol{a}\ \right) = \left(\ \boldsymbol{b'}\ \right), \tag{22}$$

are obtained, which only contain the global parameters, with a modified matrix $\boldsymbol{C'}$ and a modified vector $\boldsymbol{b'}$,

$$\boldsymbol{C'} = \sum_i \boldsymbol{C}_i - \sum_i \boldsymbol{G}_i\boldsymbol{\Gamma}_i^{-1}\boldsymbol{G}_i^T \qquad \boldsymbol{b'} = \sum_i \boldsymbol{b}_i - \sum_i \boldsymbol{G}_i\left(\boldsymbol{\Gamma}_i^{-1}\boldsymbol{\beta}_i\right)\,. \tag{23}$$

This set of normal equations contains explicitly only the global parameters; implicitly it contains, through the correction matrices, the complete information from the local parameters, influencing the fit of the global parameters. The parentheses in equation (23) represents the solution for the local parameters, ignoring the global parameters. The equation

$$\boldsymbol{a} = \boldsymbol{C'}^{-1}\,\boldsymbol{b'} \tag{24}$$

represents the solution vector $\boldsymbol{a}$ with covariance matrix $\boldsymbol{C'}^{-1}$. The solution is direct, no iterations are required.

Iterations may be necessary for other reasons, namely

- the equations depend *non-linearly* on the global parameters; the equations have to be linearized;
- the data contain outliers, which have to be removed in a sequence of cuts, becoming narrower during the iteration;
- the accuracy of the data is not known before, and has to be determined from the data (after the alignment).

3.4 Application

A very important point in problems discussed above is the definition of the set of global parameters. It is essential to define them in a way, which is sensitive to the data used to determine them. Computing the inverse of a huge (symmetric) matrix can be a delicate numerical problem. The matrix will be singular or close to a singular matrix, if one or more of the global parameters are not really determined (and constrained) by the data, and this will destroy the whole determination. This will happen in case of a strong correlation between different global parameters.
It is wise to start from few global parameters and to observe the correlations between them, before adding more global parameters. For example if too many global parameters are used in a detector alignment, the whole detector may *shift* and *rotate* freely in the fit. In those cases one could either fix certain global parameters or one has to introduce constraints (e.g. average displacement and rotation equal to zero).

Experience with the method shows that it is essential to make a simultaneous alignment of all track detector components, which are also used for the measurement of tracks, while independent internal alignment of the single track detector components may not lead to a good overall result.

It is also important to use, if possible, different data sets simultaneously. For example in a track detector alignment high-momentum cosmic muons may be useful, which traverse the whole detector and thus relate (and constrain) different parts of the detector. But this should be done simultaneously with tracks from physics events, because these events finally require a good alignment.

The method described here has been realized in the program **Millepede** [2], written in Fortran77. It provides a set of subroutines for the mathematical methods and allows to adapt the method to a particular problem. The program allows to adjust the dimension of vectors and matrices via `PARAMETER`-statements and to introduce linear constraints. Throughout the program use is made of the sparse character of the vectors and matrices in order to reduce the execution time. The program includes a method for outlier rejection, which in practice may be essential in alignment problems especially if some detector components have a bad initial alignment. This feature requires to iterate, realized with an intermediate data file, which is written during the first iteration and read during the other iterations.
Sometimes the model underlying the alignment is *nonlinear* and also constraints may be nonlinear. The standard method to treat these problems is linearization: the nonlinear equation is replaced by a linear equation for the correction of a parameter (Taylor expansion); this requires a good approximate value of the parameter. In principle this method requires an iterative improvement of the parameters, but sometimes even one iteration may be sufficient. The treatment of nonlinear equations is not directly supported by the program package, but it will in general not be too difficult to organize a program application with nonlinear equations.

4 Examples

4.1 A Monte Carlo example

Below an example for the application of the program package is shown. In this example a detector with ten planes is simulated by Monte Carlo methods. Straight tracks hit the ten planes and allow to measure one coordinate z per plane transverse to the axis of the detector, with an efficiency around 90 %. Each plane may be displaced perpendicular to the detector axis. The parametrization of the track in terms of the local parameters is

$$z = \alpha_1 + \alpha_2 \, x \, . \tag{25}$$

The Table 1 shows the x-coordinates of the planes as well as their properties: standard deviation of the measurement, and the bias (or shift) Δ. 1000 tracks are generated and the simulated measurements are used determine the bias or shift Δ with different methods. The problem has in total $9 + 2 \times 1000 = 2009$ parameters, while the interest is in the first 9 parameters.

i	x [cm]	σ [cm]	Δ [cm]	Δ_f	Δ_n	Δ_r
1	5	0.0020	0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	-0.0274
2	9	0.0020	-0.0400	-0.0399 ± 0.0001	-0.0407 ± 0.0012	-0.0636
3	20	0.0300	0.1500	0.1495 ± 0.0009	0.1465 ± 0.0047	0.1269
4	25	0.0300	0.0300	0.0293 ± 0.0009	0.0252 ± 0.0062	0.0099
5	30	0.0300	-0.0750	-0.0755 ± 0.0012	-0.0806 ± 0.0078	-0.0921
6	35	0.0300	0.0450	0.0483 ± 0.0010	0.0423 ± 0.0093	0.0346
7	40	0.0300	0.0350	0.0342 ± 0.0010	0.0271 ± 0.0108	0.0216
8	45	0.0300	-0.0800	-0.0814 ± 0.0010	-0.0895 ± 0.0123	-0.0954
9	50	0.0300	0.0900	0.0902 ± 0.0011	0.0811 ± 0.0139	0.0840
10	55	0.0300	-0.0500	-0.0494 ± 0.0011	-0.0595 ± 0.0154	-0.0513

Table 1: The parameters of the planes, used in the simulation, are shown on the left (x = chamber position, σ = standard deviation of the measurement, Δ shift perpendicular to the track direction). On the right the results of different methods to reconstruct the bias Δ from 1000 tracks are shown. The column Δ_f is the result obtained with the **Millepede** package with constraints, and the column Δ_n is obtained without the constraint of zero mean rotation (see text). The quantity Δ_r in the last column is the result of a method based on the residuals only.

In the application of the **Millepede** package one shift parameter for each of the planes is introduced. It is easy to see, that this ansatz would not give a stable solution (adding a constant term to all planes would not change the quality of the fit). Furthermore one could also rotate the set of all planes without a change of the quality of the fit. Thus two constraints are essential to get stable results. As first parameter the shift of the first plane is fixed to zero, by removing this parameter from the variable parameters (program option). In addition one constraint equation is added, which forces the average rotation to be zero. The result of this fit is shown in the column Δ_f. The data in this column show, that the bias can be reconstructed with high precision, already with 1000 tracks. The deviation between the true and the reconstructed shifts are about 1 μm for the accurate plane 2, and about 10 μm for planes $3 \ldots 10$, where the measurement has a standard deviation of 300 μm.

For comparison another result is shown in the Table 1: the column Δ_n is obtained by small modifications of the program, with the constraint of the average rotation removed. As explained above, the fit can not be stable without the constraint, and the parameter list of the final result would indicate this problem. The matrix inversion would detect a rank defect, and one shift would be returned (wrongly) as zero. The matrix is singular without the constraint. In the code used stability is established again by a special option of the program, which reduces the shifts of planes $3 \ldots 10$. The matrix in made non-singular by this option and the results are meaningful, although the precision is much lower: the standard deviations are a factor of about 10 higher.

The quantity Δ_r in the final column of Table 1 shows the result of the determination of the shift by residuals; this is perhaps a method used very often in practice. Here the local data are fitted without introducing the global shift parameters; the residuals for each plane is added to a histogram and the shifts are determined from the mean value of the histograms. The deviations to the true shifts are up to a factor 100 higher compared to the first method. It should be noted, that a repeated application of this method would make the results even worse, since the procedure can not be stable, due to missing constraints.

parameter	CJC1	CJC2
active length Δz	2200	2200
inner radius $R_{\rm i}$	203	530
outer radius $R_{\rm o}$	451	844
number of cells (rings)	30	60
number of sense wires per cell	24	32
sense wire distance	10.16	10.16
maximum drift distance at $R_{\rm i}$	22.9	28.5
maximum drift distance at $R_{\rm o}$	44.5	43.1
mean wire length	2200	2200

Table 2: Some parameters of the central jet chambers CJC1 and CJC2. All distances are given in mm.

4.2 Alignment of the central track detectors in the H1 experiment

The example described is the alignment in the $r\varphi$-plane perpendicular to the beam of a 56-plane drift chamber and a 2-plane silicon vertex detector in the H1 detector ([4]) at HERA. Both are cylindrical detectors; the drift chamber has a length (z) of about 2 m, and extending from 20.3 cm to 84.4 cm in radius r. The silicon vertex detector ([5]) has two planes around the beampipe made of 0.15 mm Aluminium and 0.9 mm carbon fiber. The drift chamber and the silicon vertex detector have a $r\varphi$ resolution of about 150 μm and 15 μm, respectively. These detectors are interspersed by additional chambers for the z-measurement.
The central jet chambers CJC1 and CJC2 ([4]) have anode sense wires parallel to the beam with two adjacent cathode planes (wires) shaping the drift field. A jet chamber cell extends azimutally from the sense wire plane to both adjacent cathode wire planes, and extends radially over the full radial span of CJC1 and CJC2 with no further subdivision. The jet cells are tilted by about 30° such that in the presence of the magnetic field (1.1 Tesla) the ionization electrons drift approximately perpendicular to high momentum vertex tracks. Some jet chamber parameters are given in Table 2.
The silicon vertex detector CST consists of two cylindrical layers of double sided, double metal silicon sensors read out by a custom designed analog pipeline chip. The two layers of the CST are formed from 12 and 20 faces at radii of 5.75 cm and 9.75 cm, respectively. One face or 'ladder' consists of six silicon sensors and aluminium nitride hybrids at each end. A double layer of carbon fiber strips with a total thickness of 700 μm and a height of 4.4 mm is glued to the edges. The position of the ladders in a layer are shifted tangentially to ensure an overlap in $r\varphi$ of adjacent active areas. The active length in z is 35.6 cm for both layers.

Data taken with the H1 detector are reconstructed online and the reconstruction modules also determine important parameters like average drift velocities, Lorentz-angle and T_0 of the CJC1 and CJC2. In addition beam parameters like the $r\varphi$ position of the primary interaction point are recorded.

In the alignment procedure using the program **Millepede** a large number of parameters is determined; these parameter are in general corrections to detector parameters and are small. The parameters are geometrical shifts, and time or velocity corrections; some have been introduced to correct for distortions of the electrical field due to bad cathode HV.
It turned out that a *common* alignment of the drift chamber and the silicon detector is essential, after an internal detector alignment of each detector, which already determines reasonable starting values for alignment parameters. For each of the drift chambers CJC1 and CJC2 14 global parameters representing an overall shift or tilt are introduced. Local variations of the drift velocity $v_{\rm drift}$ for cells halfs and layers halfs are observed, which are parametrized by 180 + 112 corrections, which change with the HV configuration. For each wire group (8 wires) corrections to T_0 are introduced (330 corrections). For the silicon vertex detector CST an internal alignment has already been done, using the same techniques, with in total 384 local parameters. In the common alignment fit 5 parameters per half ladder representing shifts are introduced (320 parameters). Table 3 contains all parameters. A far as possible the parameters are defined with the dimension *length*. For example the time-zero parameters are multiplied by the mean drift time. Angles are parametrized as shifts relative to a normalized length parameter. For example Δz_r is defined as $\Delta z/z_{max}$ and has a range –1 to +1. The shift Δx with z is parametrized as $\Delta x/\Delta z_r$ and has the dimension length.

row.	number	parameter	σ	unit
1	2	Δx	1	μm
2	2	$\Delta x/\Delta z_r$	2	μm
3	2	Δy	1	μm
4	2	$\Delta y/\Delta z_r$	2	μm
5	2	$\Delta\varphi$	10	μrad
6	2	$\Delta\varphi/\Delta z_r$	10	μrad
7	2	$\Delta\alpha_{\mathrm{Lor}}$	100	μrad
8	2	$\Delta v_{\mathrm{drift}-}/v_{\mathrm{drift}}$	10^{-5}	
9	2	$\Delta v_{\mathrm{drift}+}/v_{\mathrm{drift}}$	10^{-5}	
10	2	$\Delta T_0 \times v_{\mathrm{drift}}$	< 1	μm
11	2	wire staggering in wire plane	few	μm
12	2	wire staggering perp wire plane	few	μm
13	2	sagging in wire plane	few	μm
14	2	sagging perp. wire plane	few	μm
15	180	$\Delta v_{\mathrm{drift}}/v_{\mathrm{drift}}$ per cell half	few	10^{-4}
16	112	$\Delta v_{\mathrm{drift}}/v_{\mathrm{drift}}$ per layer half	few	10^{-4}
17	330	$\Delta T_0 \times v_{\mathrm{drift}}$ per group	10	μm
18	56	wire position in driftdir. per layer	10	μm
19	56	$\Delta T_0 \times v_{\mathrm{drift}}$ per layer	10	μm
20	56	wire pos. perp. driftdir. per layer	few 10	μm
21	112	$\Delta v_{\mathrm{drift}}/v_{\mathrm{drift}}$ for $I_e/50$ mA	few 10^{-4}	
22	90	$\Delta v_{\mathrm{drift}}/v_{\mathrm{drift}}$ per layer	few 10^{-4}	
23	90	Δy_W per layer	few 10	μm
24	90	Δy_W per layer^2	few 10	μm
25	64	Δ in ladder	few	μm
26	64	Δ perp. ladder	few	μm
27	64	rel. Δ in ladder (z_r)	few	μm
28	64	rel. Δ perp. ladder (z_r)	10	μm
29	64	rel. Δ perp. ladder (φ)	few	μm

Table 3: Alignment parameters determined in the **Millepede** fit. The **Millepede** accuray is given by a standard deviation σ. The different parameters are discussed in the text

In addition to physics events (ep interactions) also cosmic tracks with and without B-field were simultaneously used as input to the alignment. With combined fits to both halfs of cosmic tracks parts of the detectors opposite in φ angle are strongly correlated. This allows an accurate interpolation from the drift chamber to the silicon tracker with an accuracy $\sigma = 30$ to 40 μm, which is of the same order as the silicon tracker accuracy. In total about 50 000 tracks are used as input. Constraints require e.g. a zero overall shift of the detector. The CJC2 is fixed (first five parameters) vs. the cryostat on the basis of external survey data as reference.
An overview about the corrections determined in the alignment fit is given in Table 3, which has 29 rows of different corrections. In detail these corrections are as follows.

Rows 1 – 6 These rows contain geometrical parameters; in an external survey the positions are measured with an accuracy of a few 100 μm and angles with an accuracy of a few 10 μrad. This accuracy is increased by a large factor in the alignment fit.

Row 6 This parameter, which is sensitive to a curvature, is determined from $B = 0$ cosmics (straight line track model).

Rows 11 – 14 The parameters are determined only as a check. Their actual values are determined from hit triplets as a function of angle β and z_r (β is the angle between the track and the normal to the drift direction).

Row 15, 16 The parameters change with the HV configuration.

Row 17 Group = 8 wires of same FADC, cable and preamplifier.

Row 18, 20 Corrections for the wire position in and perpendicular to the drift direction. These corrections are determined in a cross check with $B = 0$ cosmic tracks, otherwise the data from a survey of flanges are used ($< 10\mu$m).

Row 21 These correction are used to describe the changes of drift velocity v_{drift} by variations of the electron beam current I_e due to space charge effects; they are determined once.

Row 22 – 24 These parameters change with repair/breakdown of HV cards. They are introduced in order to correct for distortions of the electrical field due to bad cathode HV.

Row 25 – 29 Corrections for the silicon vertex detector, five per half ladder, determined in and perpendicular to ladder, on average and as function of z and of φ.

The alignment is done for run ranges and repeated after e.g. change of the HV configuration. Run-to-run variations of parameters (e.g. the average drift velocity v_{drift}) are taken into account by the data from the online reconstruction as mentioned above.

Result from the alignment are shown in Figures 1, 2 and 3. Figure 1 shows the mean track residuals from the drift chamber fit as a function of the drift length before and after the improved alignment. The alignment procedure allows to reach the intrinsic resolution not only locally but also for tracks. For the drift chambers CJC1 and CJC2 the local hit resolution can be determined from local hit triplets as a function of drift distance; the minimum is about 120 μm for 1 cm drift distance. The minimum value from track residuals is about 125 μm. For the CST the intrinsic point resolution of 15 μm from hits overlaps is also reached globally.
Figure 2 shows the improvement for track parameters, determined from data (comparison of parameters between the two parts of cosmic tracks). A significant improvement is visible for larger track momenta. In Figure 3 the residual distribution in the parameters d_{ca} and $1/p_t$, determined from the difference between the two parts of cosmic tracks, are shown. The two upper figures are from cases, where one of the tracks has been fitted without silicon tracker hits; this represents essentially the drift chamber resolution. In the other cases both tracks are fitted with hits of the silicon tracker (middle: at least 1 hit per track, bottom: at least 2 hits per track) and represent the improved accuracy due to the silicon tracker. The mean value for the d_{ca} resolution is about 210 μm for the drift chamber alone and improves to about 32 μm, if the silicon tracker is used in addition. The relative resolution in $1/p_t$ improves from about 0.56 % to 0.23 % with the silicon tracker. The improvement is visible also in the significant reduction of the width of resonances (e.g. D^*).

Similar techniques are used for the z-measurement of the central track detector which include in addition an inner and an out z chamber with higher accuracy, and the silicon tracker.

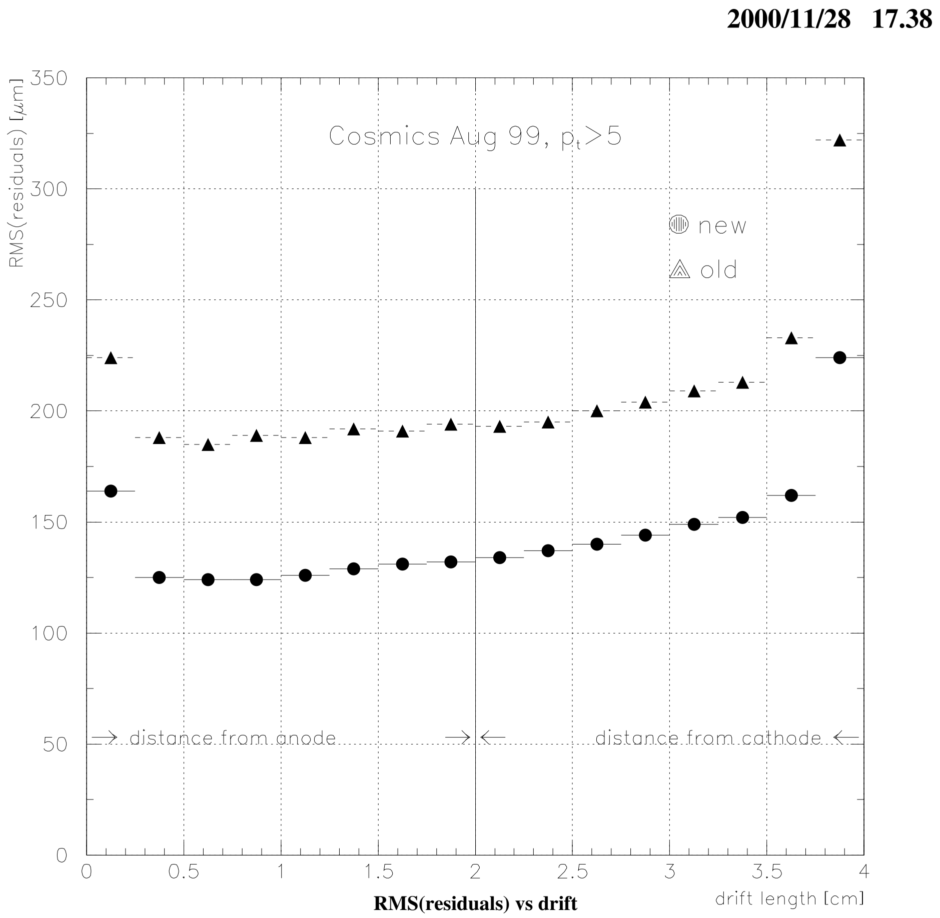

Figure 1: Mean track residuals from the drift chamber fit as a function of the drift length before and after the improved alignment. On the left the drift length is the distance from the anode, and on the right from the cathode.

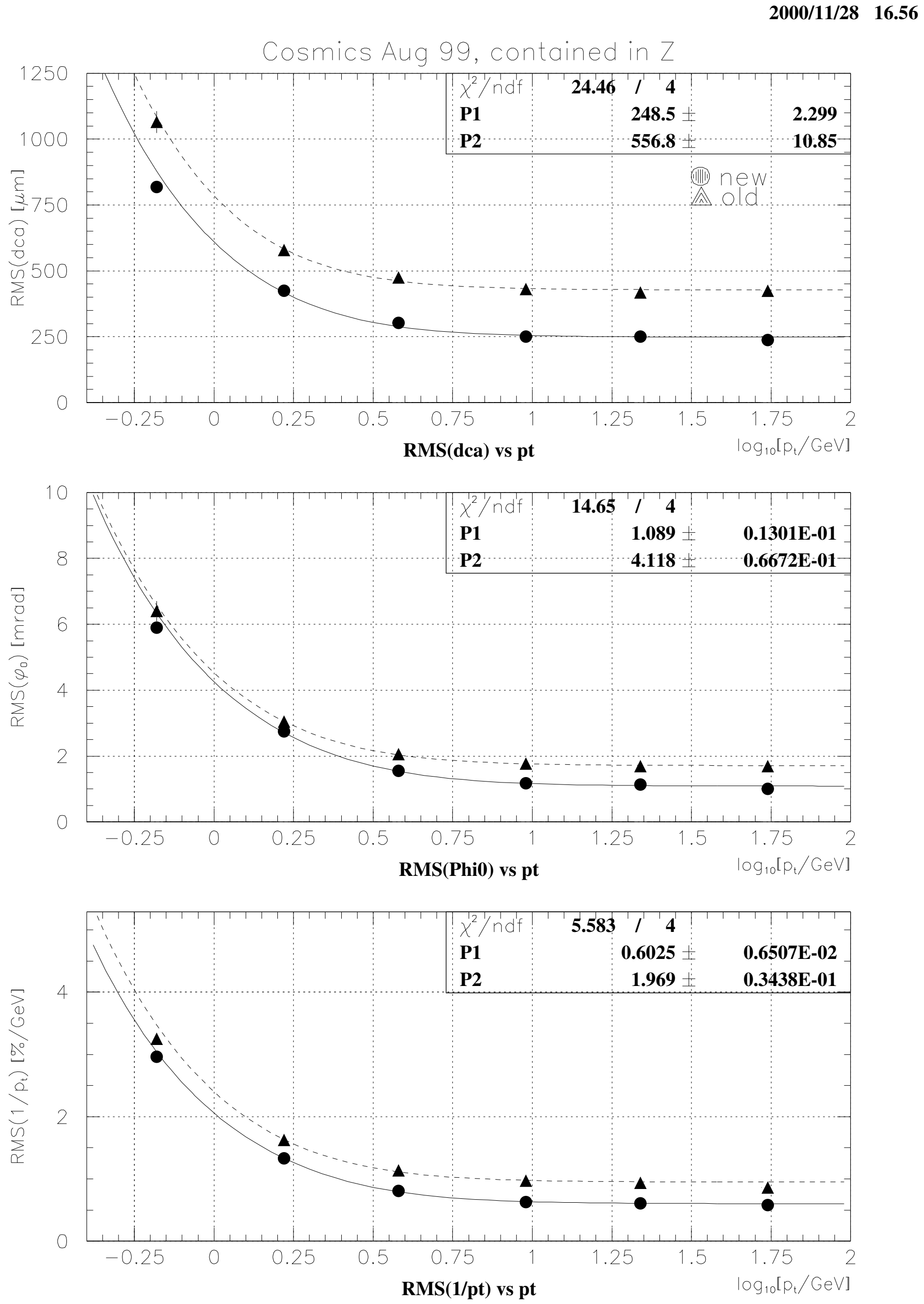

Figure 2: The standard deviations for the three track parameters d_{ca} (distance of closest aproach to the beam axis), the azimutal angle φ and of the inverse transverse momentum $1/p_t$ are shown as a function of $\log_{10} p_t/\mathrm{GeV}$. The standard deviations are determined from the difference between the two parts of cosmic tracks.

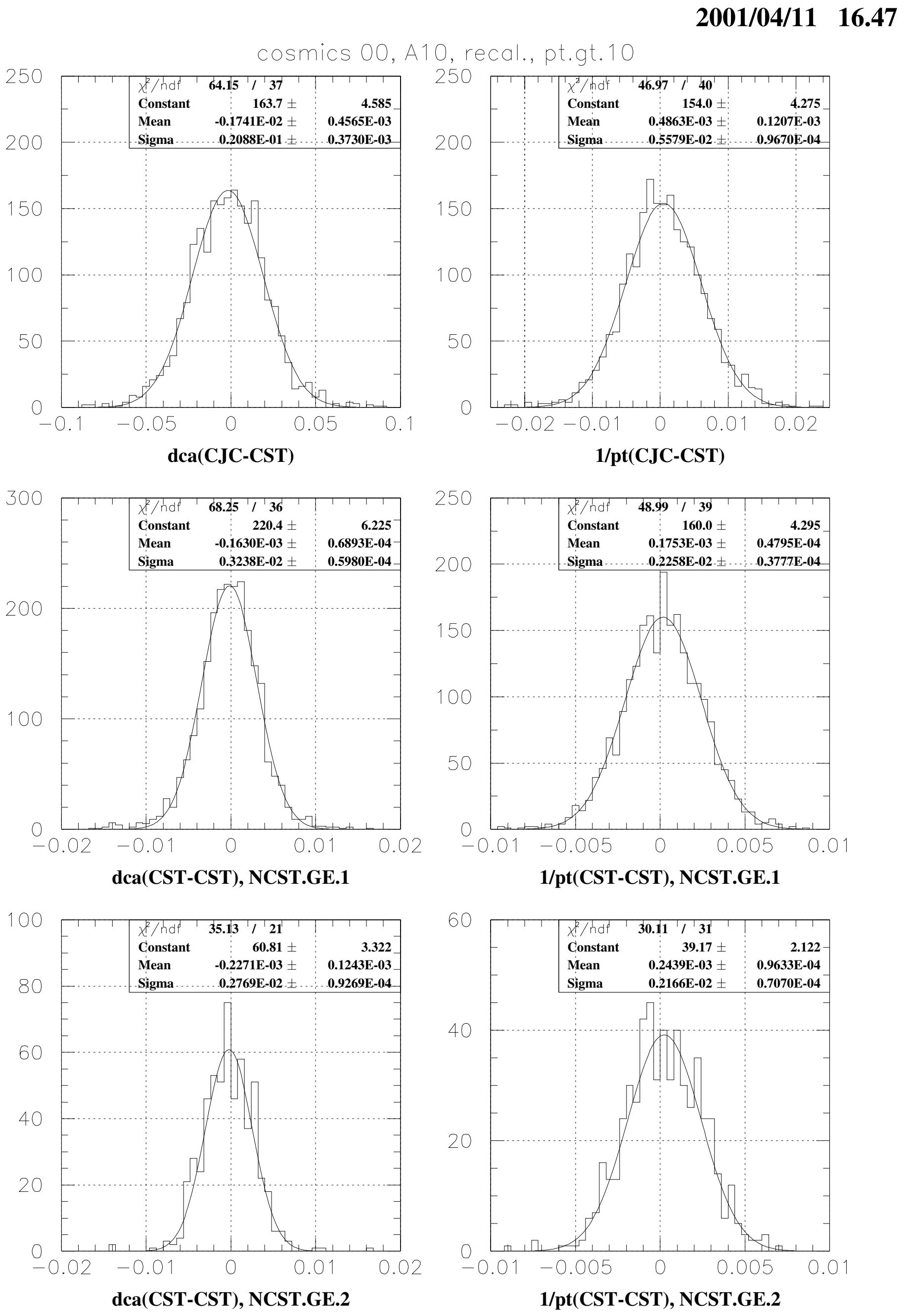

Figure 3: The histogram show the residual distribution in the parameters d_{ca} and $1/p_t$, determined from the difference Δ between the two parts of cosmic tracks. The two upper figures are from cases, where one of the tracks has been fitted without silicon tracker hits; this (difference Δ) represents essentially the drift chamber resolution. In the other cases both tracks are fitted with hits of the silicon tracker (middle: at least 1 hit per track, bottom: at least 2 hits per track) and this (difference $\Delta/\sqrt{2}$) represent the improved accuracy due to the silicon tracker.

References

[1] V. Blobel and E. Lohrmann, Statistische und numerische Methoden der Datenanalyse, Teubner, Stuttgart und Leipzig (1998)

[2] A program description for **Millepede** and the code is available via `http:/www.desy.de/~blobel/`.

[3] Schreiber, O. (1877): Rechnungsvorschriften für die trigonometrische Abteilung der Landesaufnahme, Ausgleichung und Berechnung der Triangulation zweiter Ordnung. Handwritten notes. Mentioned in W. Jordan (1910): Handbuch der Vermessungskunde, Sechste erw. Auflage, Band I, §III: 429-433. J.B.Metzler, Stuttgart.

[4] H1 Collaboration, The H1-detector at HERA, Nucl. Instrum. Meth., **A386**, 310 – 396 (1997)

[5] D. Pitzl et al., The H1 silicon vertex detector, Nucl. Instrum. Meth., **A454**, 334 – 349 (2000)