



## Depfet beam test alignment and resolution analysis

Zdeněk Doležal, Zbyněk Drásal, Peter Kodyš, Peter Kvasnička,† Pavel Řezníček

September 3, 2008

### Abstract

We present results of analysis of DEPFET testbeam data using a simple method for track selection based on the principal components analysis, and a linearized iterative alignment scheme. Detector resolutions are calculated with explicit account of multiple scattering and without the need of infinite energy extrapolation. Uncertainties in alignment parameters and detector resolutions are estimated by bootstrap resampling.

Analysis of simulated data is presented to allow assessment of the reliability of estimates. As an example, we present the results of analysis of a testbeam setup comprising five DEPFET detectors with significantly different resolutions.

---

\*supported by EU I3 Contract 026 126-R II3 (EUDET)

†peter.kvasnicka@ipnp.troja.mff.cuni.cz

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Description and basic processing of raw data</b>	<b>5</b>
2.1	Basic data . . . . .	5
2.2	Track identification in the Prague dataset . . . . .	6
2.3	Comparison of the Prague and Bonn datasets . . . . .	8
2.4	Simulations . . . . .	8
<b>3</b>	<b>The PCA track filter</b>	<b>10</b>
3.1	PCA as a track classifier . . . . .	10
3.2	Formulas for PCA . . . . .	11
3.3	Implementation . . . . .	14
3.4	Conclusions . . . . .	15
<b>4</b>	<b>Alignment</b>	<b>17</b>
4.1	Alignment geometry . . . . .	17
4.2	Linearized alignment/refitting formalism . . . . .	19
4.2.1	Track fitting . . . . .	20
4.2.2	Alignment . . . . .	21
4.3	Implementation . . . . .	23
4.3.1	The Fitter and Aligner classes . . . . .	23
4.3.2	Stability of alignment . . . . .	23
4.3.3	Iteration schedules . . . . .	24
4.3.4	Uncertainties in alignment parameters . . . . .	24
4.4	Results . . . . .	25
4.5	Alignment conclusions . . . . .	25
<b>5</b>	<b>Estimation of detector resolutions</b>	<b>30</b>
5.1	Kinked tracks in misaligned detectors . . . . .	30
5.2	Formulas for track parameters and detector resolutions . . . . .	32
5.2.1	Formulation . . . . .	32
5.2.2	Estimates of track parameters . . . . .	33
5.2.3	Predictions and detector resolutions . . . . .	34
5.2.4	Formulas for a detector not included in the fit . . . . .	35
5.2.5	Notes . . . . .	36
5.3	Implementation . . . . .	36
5.4	Results . . . . .	36
5.5	Conclusions . . . . .	37
<b>6</b>	<b>Conclusions</b>	<b>43</b>
<b>7</b>	<b>Appendix: Some matrix algebra</b>	<b>45</b>

# 1 Introduction

Alignment of beam test detectors and analysis of their resolution is one of the main tasks of beam tests. High resolution of today's detectors, such as DEPFET, requires a careful selection of statistical tools for analysis.

In this note, we present results of analysis of data from the CERN DEPFET testbeam in October 2006 (run 4009). The task was specific in several aspects:

- *Difficult hit reconstruction* due to irregularities in detector response (we have to identify and mask bad pixels/rows on event-by-event basis)
- *Multiplicity of hits* - on some detector planes, we systematically had several clusters per event, so that identification of true tracks was necessary
- *Challenging alignment* due to small active area of detectors, as well as to different quality and resolution of the detectors.
- *The need to estimate detector resolutions* based only on data taken at a single beam energy.

We have therefore used data analysis methods tailored specifically for such situation. These are also presented in some detail.

*Impact point reconstruction* was based on center-of-gravity estimates. We have developed a simple matrix method of computation of estimates and errors, taking into account the lack of noise correlations in the DEPFET. The method is described in the companion note [1].

*Track selection* was based on the principal components analysis (PCA), i.e., on similarity of tracks and a simple assumption about eigenvectors of the track covariance matrix. We used a simple iterative PCA classifier for selection of tracks *prior to* alignment/track fitting.

*Alignment* was based on a linearized alignment/refitting scheme, based on the formalism of Karimäki [2].

*Detector resolutions* were estimated using a direct approach taking explicit account of multiple scattering. This allowed us to obtain fairly reliable estimates of detector resolutions without the need of infinite energy extrapolation.

*Uncertainties in alignment parameters and detector resolutions* were calculated by bootstrap resampling - that is, they were determined from the distribution of results of a large number of analyses on replicas of the original data file; the replicas were obtained by randomization of fitted tracks from the original file.

Along with the set of tracks that we reconstructed from the raw data file for run 4009, we have carried out our analysis on a set of tracks reconstructed from the same data file by the Bonn group and kindly provided to us by Jaap Velthuis.

We have also carried out the analysis on a set of 50,000 tracks obtained by GEANT4 simulation. These results allowed us to assess the reliability of the analysis.

Below we give a description of the methods used in the analysis and present the results for the selected DEPFET testbeam run.

The note is organized as follows:

Chapter 2 contains information about input data (runs) and processing leading to hit reconstruction.

Chapter 3 introduces the PCA track classifier and discusses some of its properties based on the analysis of available data.

Chapter 4 introduces the alignment/refitting scheme and presents alignment results.

In Chapter 5, we introduce our detector resolution estimators and present detector resolutions estimated from the data.

## 2 Description and basic processing of raw data

This chapter covers steps leading from raw data to a set of candidate tracks (prototracks). Therefore, the data processing described here only concerns the Prague data set. The Bonn data set was by itself a collection of tracks. An important part of the chapter, however, is dedicated to comparison of the two datasets. Finally, we also give a short description of the set of simulated tracks.

### 2.1 Basic data

We start with some basic parameters concerning the beam test setup. They are summarized in Fig. 2.1 and in Table 2.1.

Parameter	Unit	Value
Number of detectors		5
Distance between detectors	mm	25
Thickness of detectors	$\mu\text{m}$	450
Number of detector cells		$64 \times 128$
Pitch of detector 1	$\mu\text{m}^2$	$33 \times 24$
Pitch of detectors 2 - 5	$\mu\text{m}^2$	$36 \times 22$

Table 2.1: Basic parameters of detector setup

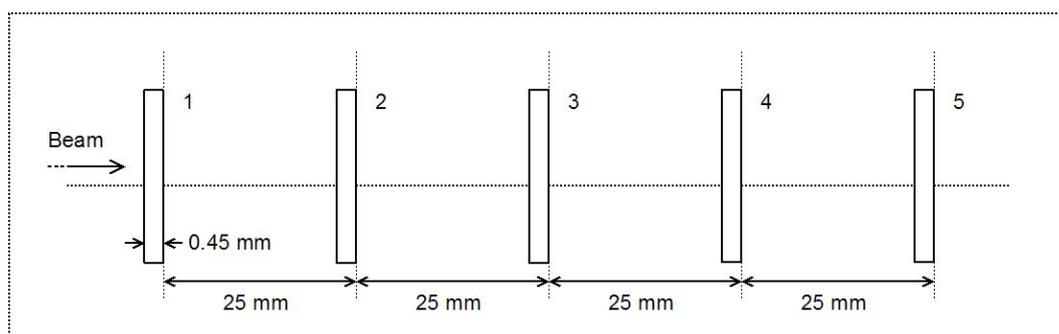


Figure 2.1: Simplified drawing of the detector setup shows basic dimensions and numbering of detectors

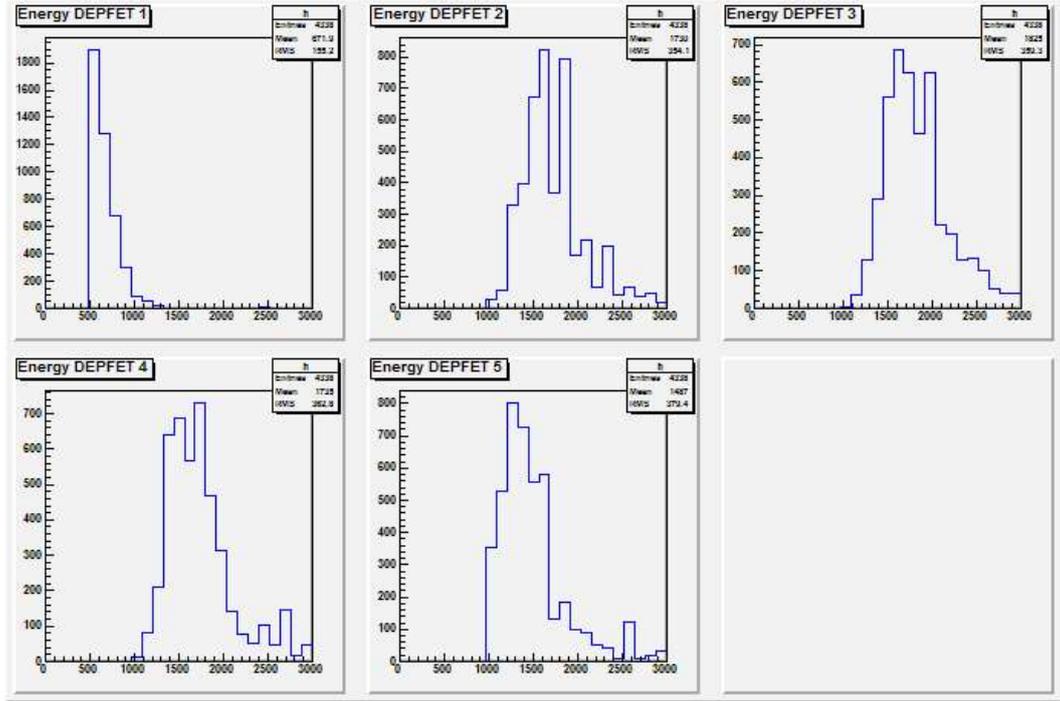


Figure 2.2: Cluster charge distributions of hits in candidate tracks of the Prague dataset.

### Pitch of detector 1

Our analysis of raw data and comparison with the Bonn dataset show that detector 1 differs from the other four detectors in two aspects:

- Its pitch is different (see Table 2.1; note that the actual value of the fine pitch is 23.75 rather than 24  $\mu\text{m}$ ).
- Its signal-to-noise ratio is much lower than in the other detectors and its signal occupancy is much lower, about 10% compared to the other detectors (see the following section).

## 2.2 Track identification in the Prague dataset

Pedestals and common mode were subtracted from the data using standard procedures, the only modification was in the use of robust location and scale parameters (median and median square deviation) in place of Gaussian fits, to achieve higher robustness and reduce computational costs.

Bad rows and pixels were identified and masked iteratively and on event-to-event basis.

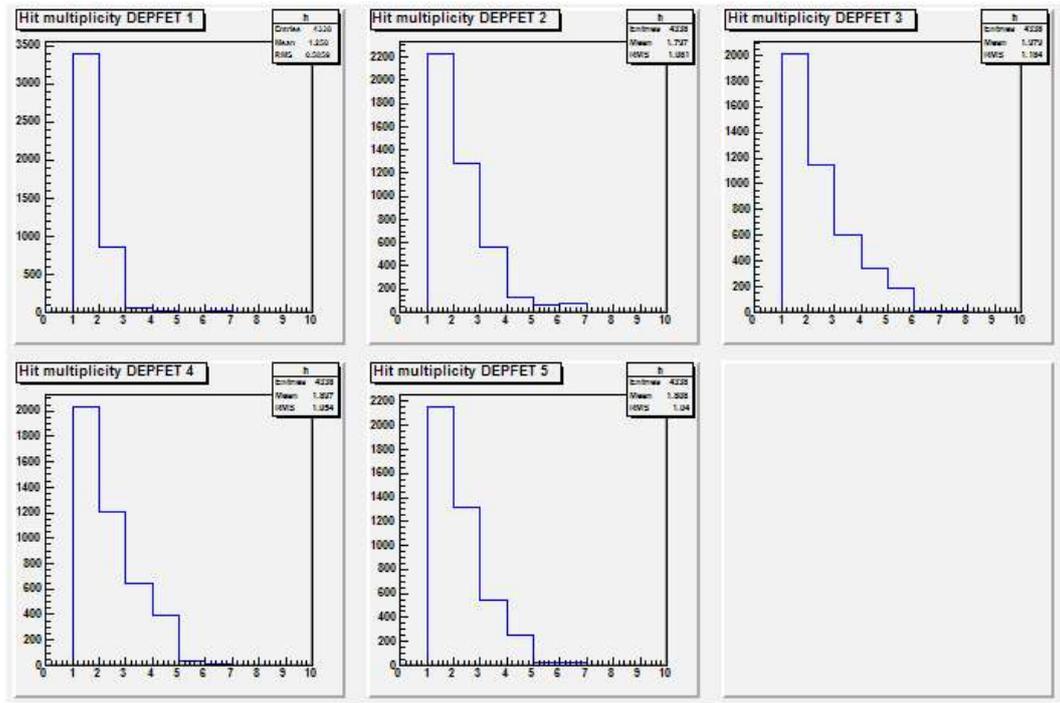


Figure 2.3: Multiplicity of hits in candidate tracks of the Prague dataset.

Clusters were identified using a five-sigma threshold on seeds and three-sigma threshold on other pixels in the cluster. Only clusters outside the masked areas were considered. Impact point coordinates were determined using the matrix CoG method ([1]) based on all signals over threshold in the vicinity of the cluster seed. No eta correction was used.

For track identification, we only considered as hits the clusters with charge exceeding a certain threshold to eliminate false hits. The thresholds were 500 for detector 1 and 1000 for detectors 2 to 5. Even with this filtering, we were often left with 5 or more clusters per detector plane and event.

Candidate tracks were formed from all combinations of hits for a given event. Only complete tracks were considered - that is, events with no hit on one of the detector planes were disregarded. This requirement was quite restrictive due to the low signal occupancy on detector 1. However, further analyses would become very complicated if we accepted a heterogeneous set of tracks with different observations missing.

This left us with 4338 candidate tracks. To avoid selection bias, no filtering based on the geometry of tracks was attempted prior to PCA filtering.

## 2.3 Comparison of the Prague and Bonn datasets

For analysis, we also used a set of pre-filtered tracks kindly provided by Jaap Velthuis. Apparently, this dataset was a result of a more sophisticated track selection, as can be seen in Table 2.2:

Dataset	Total tracks	Tracks after PCA
Prague	4338	308
Bonn	3011	1892

Table 2.2: Numbers of candidate tracks in the Prague and Bonn datasets, and numbers of tracks accepted by the PCA filter (see Chapter 3). Note that, at this stage, the two sets of tracks are substantially different in the stage of processing - the Prague tracks are combinations of identified hits, while the Bonn tracks are already a result of some filtering.

As can be seen in Fig. 2.4, there is a substantial difference in the distribution of hits for the two datasets. Apparently, the acceptance criteria for clusters were significantly different - the Prague procedure masked large areas of detectors, leading to much lower number of useful tracks found.

## 2.4 Simulations

As a third data set, we used 50,000 tracks generated by a GEANT4 simulation for the given beam parameters and detector geometry. The simulated tracks were "exact" in

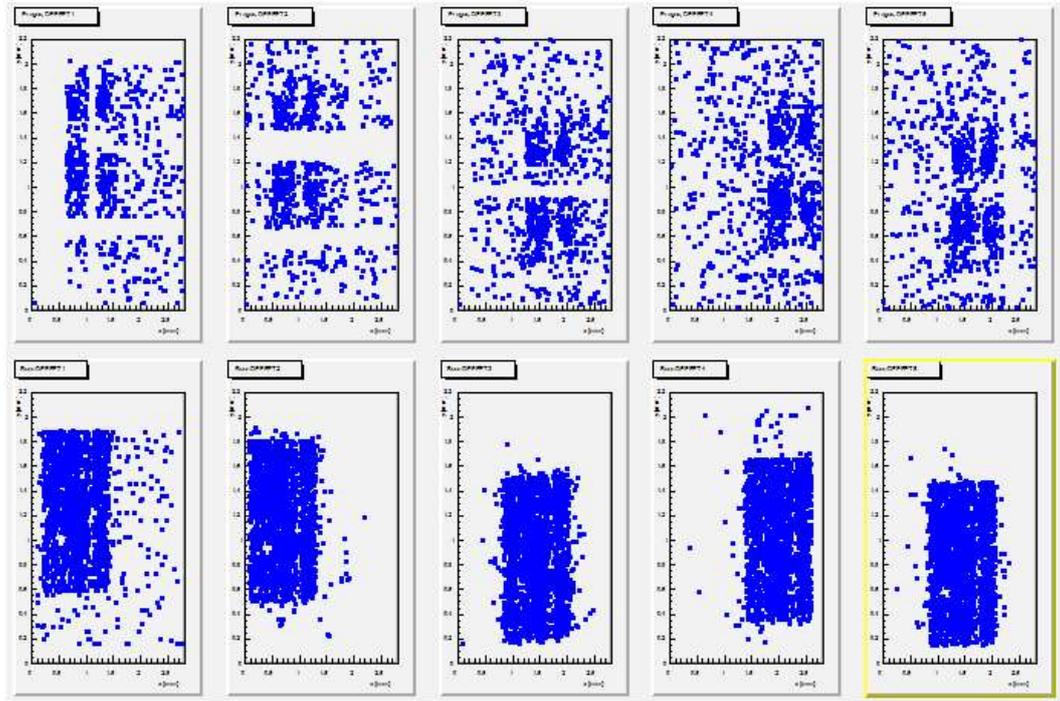


Figure 2.4: Hitmaps of candidate tracks from the Prague (upper row) and Bonn (lower row) datasets. There are substantial differences in the distribution of accepted hits. The more stringent masking used in the Prague data is not seen in the Bonn dataset.

the sense that they did not contain measurement errors.

From this base pool, we generated subsets of the required size, and adjusted the tracks by Gaussian noise to simulate measurement errors. The tracks were also randomly translated in space to reproduce the spatial distribution of tracks in the actual data.

# 3 The PCA track filter

## 3.1 PCA as a track classifier

We use the Principal components analysis to represent each track as a linear combination of eigenvectors of the covariance matrix constructed from all tracks in a data set. The idea is that eigenvectors corresponding to the largest eigenvalues represent (*explain*) dominant, common features in the set of tracks, while eigenvectors corresponding to small eigenvalues represent minor or exotic features pertinent to only a small fraction of tracks.

We can thus efficiently eliminate tracks with unusual features from the set based on their scores (content) of higher principal vectors (that is, of principal vectors with small eigenvalues). The advantage is that this will work independent of whether the set of detector planes has been aligned or not - typical tracks will be kinky in case of misalignment, but remain typical, so the technique can even be used to extract alignment parameters.

The approach may not work well when the set of tracks is highly contaminated - that is, when the typical track is a fake track. To create a PCA classifier suitable also for such situations, we must explore the statistics of "good" particle tracks.

The statistical model describing measured crossing points of a (piecewise) linear particle track (in the simplest setting: global frame, no misalignment) is [3]

$$\begin{aligned}x_k &= x_0 + a^{(x)} z_k + \sum_{j < k} (z_k - z_j) \epsilon_j^{(x)} + d_k^{(x)} \\y_k &= y_0 + a^{(y)} z_k + \sum_{j < k} (z_k - z_j) \epsilon_j^{(y)} + d_k^{(y)} \\k &= 1, 2, \dots, n\end{aligned}\tag{3.1}$$

The differences among tracks are due to several random variables: their position in space ( $x_0, y_0$ ), direction ( $a^{(x)}, a^{(y)}$ ), multiple scattering deflections ( $\epsilon^{(x)}, \epsilon^{(y)}$ ) and measurement errors ( $d^{(x)}, d^{(y)}$ ). Apparently, most important differences (i.e., the largest part of track variability) are due to different positions and directions of particle tracks, while measurement errors and multiple scattering are minor contributions. Thus, we expect that the first two principal vectors will be mostly related to positions and directions of particle tracks, with broad distribution of the corresponding scores (content) among

tracks, while higher principals will correspond to some combinations of measurement errors and multiple scattering deflections, with distribution of scores localized and similar to Gaussians. Thus, we can get rid of bad tracks by excluding tracks

- with scores  $\pi_i, i = 3, \dots, 10$  of the third and higher principals outside of a central (localized) distribution,
- with a high summary score of high principals,

$$s_{high}^2 = \sum_{i=3}^10 p_i^2$$

The first criterion allows to eliminate bad tracks from a moderately contaminated sample. The second criterion is important when central distributions of higher principals are invisible; by a proper cut on the summary score, the central regions can be revealed. Effectively, this is nothing else but a cut on chisquare.

For the present analysis, we have developed and used an iterative PCA classifier that worked as follows:

1. Initially, mark all tracks as good
2. Compute PCA for good tracks and compute scores for all tracks;
3. Select a subset of tracks with summary score of high principals below a small quantile of the summary score distribution (say 5 %);
4. For the subset, fit central parts of distributions of scores  $p_i, i = 3, \dots, 10$  with gaussians and set bounds on scores to  $t_{high}\sigma_i$ , with  $t_{high}$  being 3 or 4, and  $\sigma_i$  being the widths of the gaussians.
5. Mark as good those tracks for which  $|p_i| < t_{high}\sigma_i, i = 3, \dots, 10$ .
6. Iterate starting from item 2.

The idea is to identify the (potentially) small subset of good tracks by a cut on the summary score  $s_{high}^2$ , and classify the tracks based on principals derived from this subset. The iterations then lead to self-consistency of the classifier, i.e., under favorable conditions, the subset of good tracks stops changing after several iterations. To achieve convergence, the cut on summary score must be smaller than the fraction of good tracks in the sample. It is thus safest to start with the strongest cut that still keeps enough tracks to obtain a reliable PCA (that is, keeping initially at least several tens of tracks).

## 3.2 Formulas for PCA

Technically, the PCA is carried out as follows: We start by forming a matrix  $\mathbf{X}$  from track data:

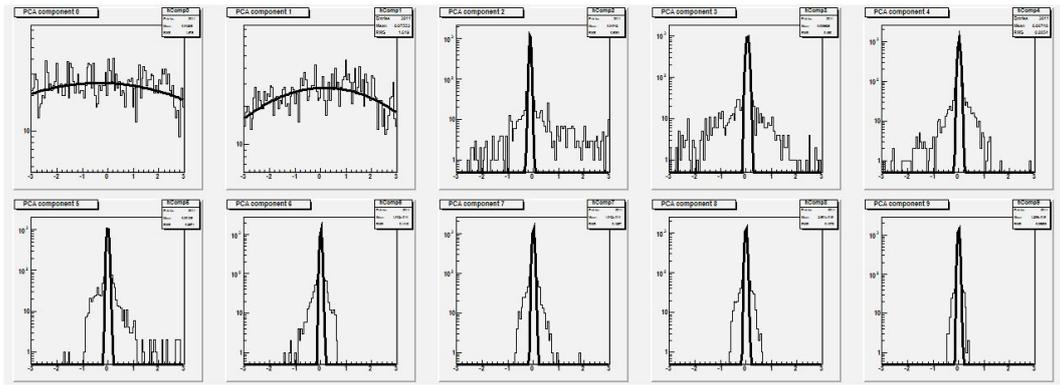


Figure 3.1: PCA analysis of tracks from the Bonn dataset. Distribution of scores of PCA eigenvectors. Track filtering is based on elimination of tracks with scores of higher eigenvectors (3 to 10) outside of the central Gaussian regions. A 3-sigma cut eliminates a large fraction of fake tracks.

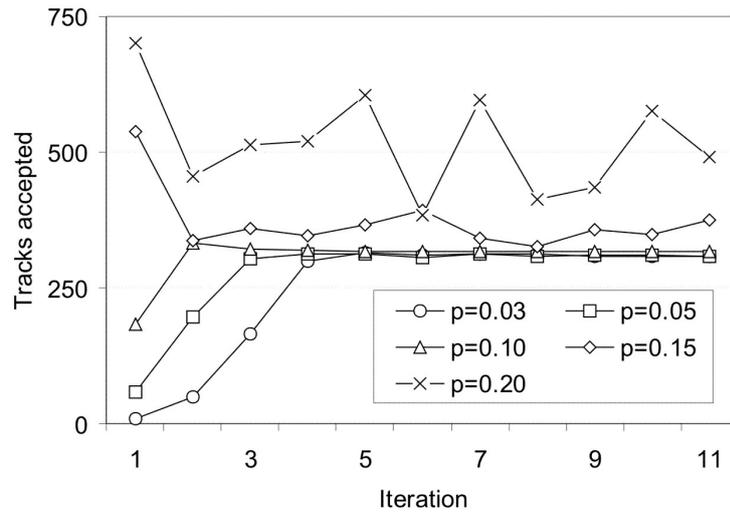


Figure 3.2: Convergence properties of the PCA classifier. The graph shows the number of tracks (Prague dataset) accepted at subsequent iterations at different cuts on the summary score  $p_{high}$ . We see the algorithm converges for sufficiently strong cuts; for weaker cuts, the behavior is oscillatory.

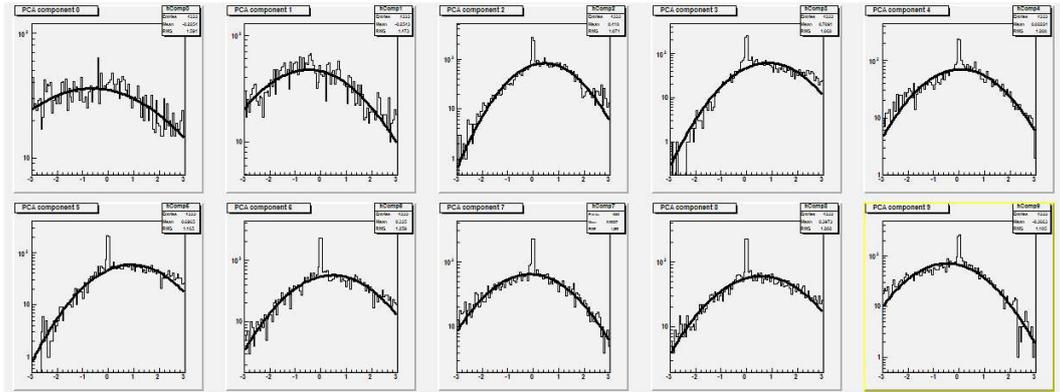


Figure 3.3: PCA analysis of tracks from the Prague dataset. Distribution of scores of PCA eigenvectors without a cut on summary score. We see that the distributions of scores 3-10 are very broad, meaning that there is no group of similar tracks differing (except for position and direction) only by slight deviations.

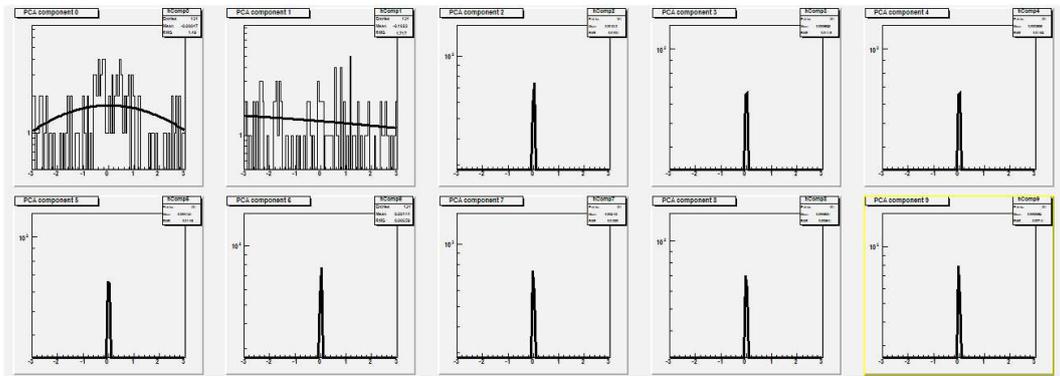


Figure 3.4: PCA analysis of tracks from the Prague dataset. Distribution of scores of PCA eigenvectors after ten iterations of the classifier. Narrow central regions now became visible and we use a 3 sigma cut to eliminate most fake tracks (in this case, we keep 308 out of 4338 tracks).

$$\mathbf{X} = \begin{pmatrix} x_1^{(1)} & \dots & x_n^{(1)} & y_1^{(1)} & \dots & y_n^{(1)} \\ x_1^{(2)} & \dots & x_n^{(2)} & y_1^{(2)} & \dots & y_n^{(2)} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_1^{(N)} & \dots & x_n^{(N)} & y_1^{(N)} & \dots & y_n^{(N)} \end{pmatrix} \quad (3.2)$$

We form the covariance matrix  $\mathbf{C}$ , which is real, positive definite, and symmetric (meaning that all its eigenvalues are positive) and find its eigenvalues and eigenvectors:

$$\begin{aligned} \mathbf{C} &= (\mathbf{X} - \langle \mathbf{X} \rangle)^T (\mathbf{X} - \langle \mathbf{X} \rangle) \\ \mathbf{C} &= \mathbf{U}^T \mathbf{\Lambda} \mathbf{U} \end{aligned} \quad (3.3)$$

The matrix of eigenvectors  $\mathbf{U}$  defines a transformation from the pattern (track) space to track feature space. The vector of scores for a given track is

$$\mathbf{s} \equiv \begin{pmatrix} p_1 \\ \dots \\ p_n \\ p_{n+1} \\ \dots \\ p_{2n} \end{pmatrix} = \mathbf{U}^T \cdot \begin{pmatrix} x_1 \\ \dots \\ x_n \\ y_1 \\ \dots \\ y_n \end{pmatrix} \quad (3.4)$$

and the summary score is defined as

$$s_{high}^2 = \sum_{i>2} p_i^2 \quad (3.5)$$

The iterative PCA classifier is parameterized by two parameters:

$p_{high}$ , which defines the cut  $s_{high}^2(p_{high})$  on summary score as a quantile of the distribution of summary scores, that is,

$$P(s_{high}^2 < s_{high}^2(p_{high})) = p_{high}$$

$t_{high}$ , which defines the cut on scores of higher principals  $|p_i|, i > 2$  as an interval around zero,

$$|p_i| < t_{high} \sigma_i$$

with  $\sigma_i$  being the half-width of the gaussian fit to the distribution of  $p_i$ .

### 3.3 Implementation

The PCA implementation was very simple and efficient thanks to the ROOT class TPrincipal by Christian Holm. The class iteratively builds the covariance matrix as tracks are added one-by-one, so the calculations are very efficient and fast.

For all analyses, we used  $p_{high} = 0.05$  and  $t_{high} = 3$ .

### 3.4 Conclusions

As can be seen in Figs. 3.5 and 3.6, the PCA filter removes tracks outside the dense beam areas, and substantially reduces hit multiplicities.

Some further work is required to better understand and optimize the workings of the filter.

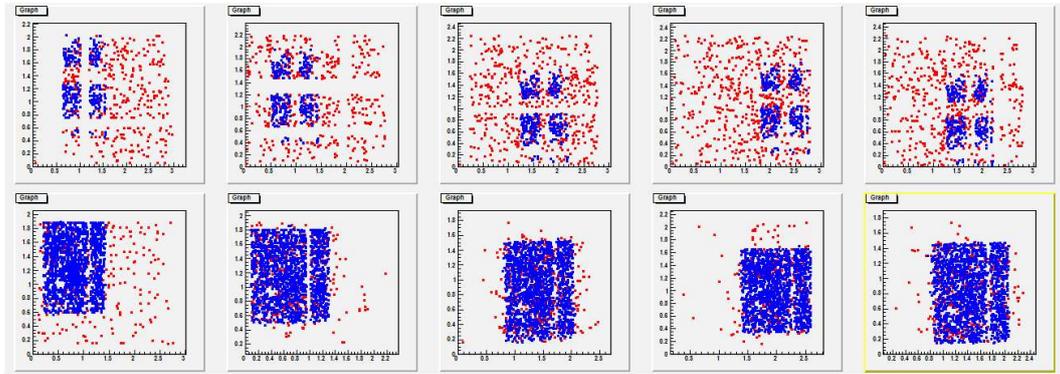


Figure 3.5: Effects of PCA filtering on hitmaps for the Prague (upper row) and Bonn (lower row) datasets. PCA clearly filters out tracks outside the dense beam regions.

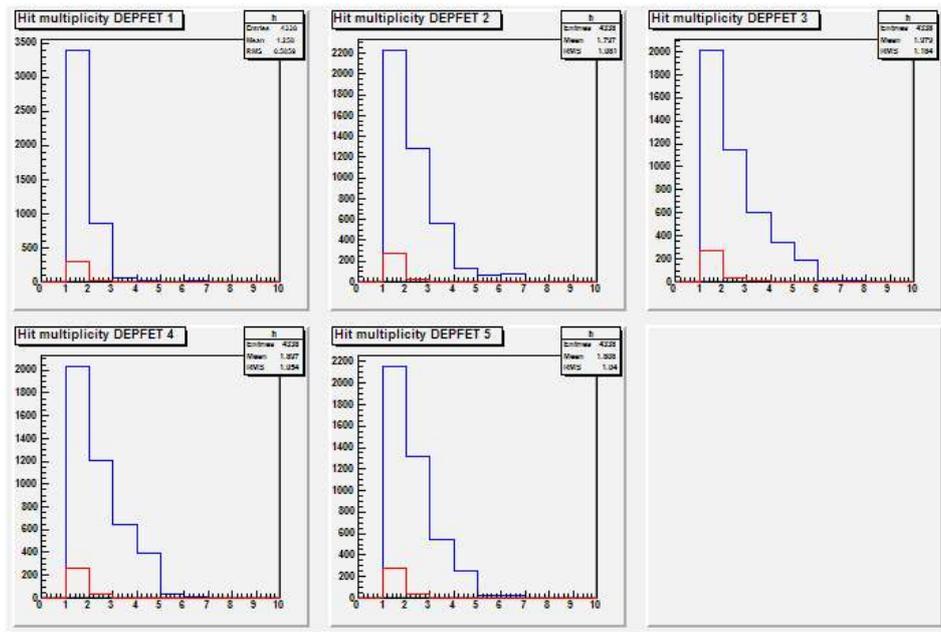


Figure 3.6: Effects of PCA filtering on multiplicity of tracks for the Prague (upper row) and Bonn (lower row) datasets. Apparently, multiplicities are significantly reduced.

## 4 Alignment

For this analysis, we used a simple alignment scheme based on linearization of equations for alignment parameters. Our goal was to have a simple and robust alignment method, keeping in mind that the main problem in alignment are local minima rather than non-linearity.

### 4.1 Alignment geometry

We used the formalism of Karimäki et al. [2].

We define local coordinate system of a planar detector  $(u, v, w)$  as follows: The origin is at the centre of the sensor, the  $w$  axis is perpendicular to detector plane,  $u$  axis is in the direction of fine coordinate, and the  $v$  axis in the direction of the coarse coordinate<sup>1</sup>. Global coordinates are denoted as  $(x, y, z)$ .

The transformation from the global to the local coordinate system is

$$\mathbf{q} = \mathbf{R}(\mathbf{r} - \mathbf{r}_0) \quad (4.1)$$

where  $\mathbf{q} = (u, v, w)$ ,  $\mathbf{r} = (x, y, z)$ ,  $\mathbf{R}$  is a rotation and  $\mathbf{r} = (x, y, z)$  is the position of the detector center in the global coordinate frame.

Due to alignment corrections, the rotation matrix and position of the detector are updated:

$$\begin{aligned} \mathbf{R} &\rightarrow \Delta\mathbf{R}\mathbf{R} \\ \mathbf{r}_0 &\rightarrow \mathbf{r}_0 + \Delta\mathbf{r}_0 \end{aligned} \quad (4.2)$$

The position correction transforms to the local system as

$$\Delta\mathbf{q} = \Delta\mathbf{R}\mathbf{R}\Delta\mathbf{r}_0 \quad (4.3)$$

so the corrected global-to-local transformation becomes

---

<sup>1</sup>For DEPFET, the tradition seems to be the contrary, and thus, in the following,  $u$  is the *coarse* coordinate.

$$\mathbf{q}^{(c)} = \Delta \mathbf{R} \mathbf{R} (\mathbf{r} - \mathbf{r}_0) - \Delta \mathbf{q}_0 \quad (4.4)$$

Next we need an update equation for the crossing point of a linear track with the detector. We consider a linear track in the global coordinate frame

$$\mathbf{r}_s(h) = \mathbf{r}_x + h\mathbf{s} \quad (4.5)$$

Here,  $\mathbf{r}_x$  is the crossing point of the track with the detector in the uncorrected position,  $h$  is a parameter and  $\mathbf{s}$  is a unit vector pointing along the track.

Using Eqn. 4.4, the *corrected* position of the crossing point in the local frame is

$$\mathbf{q}_s(h) = \mathbf{R}_c (\mathbf{r}_x + h\mathbf{s} - \mathbf{r}_0) - \Delta \mathbf{q}_0, \quad \mathbf{R}_c \equiv \Delta \mathbf{R} \mathbf{R} \quad (4.6)$$

The crossing point is in the plane of the detector, so that its  $w$  coordinate should be zero:

$$\mathbf{w} \cdot \mathbf{q}_c(h_x) = 0, \quad h_x = \frac{[\Delta \mathbf{q} - \mathbf{R}_c (\mathbf{r}_x - \mathbf{r}_0)] \cdot \mathbf{w}}{\mathbf{R}_c \mathbf{s} \cdot \mathbf{w}} \quad (4.7)$$

The corrected impact point coordinates in the local frame are then

$$\mathbf{q}_x^c = \mathbf{R}_c (\mathbf{r}_x - \mathbf{r}_0) + \frac{[\Delta \mathbf{q} - \mathbf{R}_c (\mathbf{r}_x - \mathbf{r}_0)] \cdot \mathbf{w}}{\mathbf{R}_c \mathbf{s} \cdot \mathbf{w}} \mathbf{R}_c \mathbf{s} - \Delta \mathbf{q}$$

Introducing

$$\begin{aligned} \mathbf{q}_x &\equiv \mathbf{R} (\mathbf{r}_x - \mathbf{r}_0) \\ &\quad \text{(impact point in uncorrected LF)} \\ \mathbf{t} &\equiv \mathbf{R} \mathbf{s} \\ &\quad \text{(track direction in uncorrected LF)} \\ \Delta w &\equiv \Delta \mathbf{q} \cdot \mathbf{w} \end{aligned} \quad (4.8)$$

we finally arrive at the following expression for impact point update:

$$\mathbf{q}_x^c = \Delta \mathbf{R} \mathbf{q}_x + (\Delta w - [\Delta \mathbf{R} \mathbf{q}_x]_3) \frac{\Delta \mathbf{R} \mathbf{t}}{[\Delta \mathbf{R} \mathbf{t}]_3} - \Delta \mathbf{q} \quad (4.9)$$

## 4.2 Linearized alignment/refitting formalism

The crossing point update formula (Eqn. 4.9) is a fundamental relation for both alignment and track fitting. For the present testbeam setup, we only expect slight misalignment rotations, so we can write the correction matrix as

$$\Delta\mathbf{R} = \begin{pmatrix} 1 & \Delta\gamma & \Delta\beta \\ -\Delta\gamma & 1 & \Delta\alpha \\ -\Delta\beta & -\Delta\alpha & 1 \end{pmatrix} \quad (4.10)$$

Using this form of the matrix, we can linearize Eqn. 4.9 to obtain the following equations for alignment update of impact point coordinates:

$$\begin{aligned} \Delta u_x &= -\Delta u + \delta \tan \phi + \Delta\gamma v_x \\ \Delta v_x &= -\Delta v + \delta \tan \psi - \Delta\gamma u_x \end{aligned} \quad (4.11)$$

with

$$\delta \equiv \Delta w + \Delta\beta u_x + \Delta\alpha v_x \quad \tan \phi \equiv \frac{t_1}{t_3} \quad \tan \psi \equiv \frac{t_2}{t_3} \quad (4.12)$$

This makes the update equations linear in alignment parameters. We can use the same equations also for track fitting. If we start from detectors in reference positions, i.e., with planes properly centered on the  $z$  axis and with local axes  $(u, v, w)$  oriented along  $(x, y, z)$  (so that  $\mathbf{R}$  is unity and  $\mathbf{r}_0^{(k)} \equiv [0, 0, z^{(k)}]$ ), equations for the crossing point in the  $k$ -th detector are, in this detector's local frame,

$$\begin{aligned} u_k &= u_0 + z_k \tan \phi \\ v_k &= v_0 + z_k \tan \psi \end{aligned} \quad (4.13)$$

Using Eqn. 4.11, we can write the equations expressing the positions of crossing points in misaligned detector frames as follows:

$$\begin{aligned} u_k^c &= u_k - \Delta u_k + \Delta\gamma_k v_k + \tan \phi \delta_k \\ v_k^c &= v_k - \Delta v_k - \Delta\gamma_k u_k + \tan \psi \delta_k \end{aligned} \quad (4.14)$$

These equations, together with Eqn. 4.12 and Eqn. 4.13, express predictions of impact points that we expect to observe, and we can fit parameters by minimizing the squared deviation between the observed hit coordinates  $(u_k, v_k)$  and predictions of this set of equations. We will use the equations

- to fit track parameters (on a track-to-track basis) with alignment parameters fixed, or
- to fit alignment parameters (on a detector-to-detector basis) with track parameters fixed.

The alignment problem is linear; the track fitting problem is slightly nonlinear, but can be solved using linear regression iteratively (see below).

For alignment and fitting, we need to rearrange the equations differently. For fitting, we will separate the vector of track parameters, and stack the equations for all points of a track. For alignment, we will separate the vector of alignment parameters, and stack the equations for all hits in a detector.

### 4.2.1 Track fitting

We first note that equations 4.14 are nonlinear in  $\tan \phi_k$  and  $\tan \psi_k$  - indeed, we have

$$\delta_k \equiv \Delta w_k + \Delta \beta_k u_k + \Delta \alpha_k v_k$$

This non-linearity is, however, very slight: the quantities  $\delta_k$  are very small compared to  $z_k$ , so that track slopes are only slightly influenced. We fit the tracks *iteratively*:

In Eqn. 4.14 we use the slopes from a previous fit and start with zero slopes; we only use the slopes in  $u_k$  and  $v_k$  as actual fit parameters. As a rule, one or two iterations of the fit are sufficient to achieve consistency of slopes.

For track fitting, we rearrange Eqn. 4.14 as follows:

$$\begin{pmatrix} u_k^c \\ v_k^c \end{pmatrix} = \mathbf{\Delta}_k + \mathbf{F}_k \mathbf{B} \quad (4.15)$$

with

$$\mathbf{\Delta}_k = \begin{pmatrix} -\Delta u_k + \Delta w_k \tan \phi^{(0)} \\ -\Delta v_k + \Delta w_k \tan \psi^{(0)} \end{pmatrix} \quad (4.16)$$

$$\mathbf{F}_k = \begin{pmatrix} 1 & z_k \end{pmatrix} \otimes \mathbf{R}_k, \quad (4.17)$$

, where  $\otimes$  denotes the Kronecker product (cf. Appendix A) and

$$\mathbf{R}_k = \begin{pmatrix} 1 + \Delta \beta_k \tan \phi^{(0)} & \Delta \gamma_k + \Delta \alpha_k \tan \phi^{(0)} \\ -\Delta \gamma_k + \Delta \beta_k \tan \psi^{(0)} & 1 + \Delta \beta_k \tan \phi^{(0)} \end{pmatrix} \quad (4.18)$$

;

$$\mathbf{B} = \begin{pmatrix} u_0 \\ v_0 \\ \tan \phi \\ \tan \psi \end{pmatrix} \quad (4.19)$$

Here,  $\tan \phi^{(0)}$  and  $\tan \psi^{(0)}$  denote *iterated* parameters, i.e., their value is taken from a previous fit or, initially, set to zero. By stacking the equations for all points of a track, we obtain a regression model for the track:

$$\mathbf{Y} = \mathbf{\Delta} + \mathbf{F}\mathbf{B} + \mathbf{E} \quad (4.20)$$

with

$$\mathbf{Y} \equiv \begin{pmatrix} u_1^{observed} \\ v_1^{observed} \\ \dots \\ u_n^{observed} \\ v_n^{observed} \end{pmatrix}, \quad \mathbf{\Delta} = \begin{pmatrix} \mathbf{\Delta}_1 \\ \dots \\ \mathbf{\Delta}_n \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} (1 \ z_1) \otimes \mathbf{R}_1 \\ \dots \\ (1 \ z_n) \otimes \mathbf{R}_n \end{pmatrix}, \quad \mathbf{E} = \begin{pmatrix} \begin{pmatrix} \epsilon_1 \\ \eta_1 \end{pmatrix} \\ \dots \\ \begin{pmatrix} \epsilon_n \\ \eta_n \end{pmatrix} \end{pmatrix} \quad (4.21)$$

$\mathbf{E}$  is a column vector of measurement errors; in the absence of multiple scattering, the error covariance  $\langle \mathbf{E}\mathbf{E}^T \rangle$  is diagonal with squared detector resolutions on the diagonal. Multiple scattering makes error-covariance matrix non-diagonal.  $n$  runs over all points of a track.

The least-squares solution of the quasi-linear regression problem of Eqn. 4.20 is

$$\hat{\mathbf{B}} = (\mathbf{F}^T \mathbf{F})^{-1} \mathbf{F}^{-1} (\mathbf{Y} - \mathbf{\Delta}) \quad (4.22)$$

All other statistics, such as parameter error covariances, prediction covariances etc. are calculated in the usual manner after the slopes have converged in a couple of iterations:

- The matrix of parameter error covariances is  $(\mathbf{F}^T \mathbf{F})^{-1}$ .
- Model predictions are

$$\hat{\mathbf{Y}} = \mathbf{F}\hat{\mathbf{B}} \quad (4.23)$$

- Model residuals are

$$\mathbf{r} = \mathbf{Y} - \hat{\mathbf{Y}} = \left( \mathbf{I} - \mathbf{F} (\mathbf{F}^T \mathbf{F})^{-1} \mathbf{F}^T \right) \mathbf{Y} = \mathbf{H}\mathbf{Y} \quad (4.24)$$

- Covariance matrix of residuals is

$$\langle \mathbf{r}\mathbf{r}^T \rangle = \mathbf{H} \langle \mathbf{E}\mathbf{E}^T \rangle \mathbf{H} \quad (4.25)$$

### 4.2.2 Alignment

For alignment, we rearrange Eqns. 4.14 to obtain a regression model for alignment parameters of a selected detector plane. As the two steps are never mixed, we can use the same symbols as in the fitting subsection to simplify notation:

$$\mathbf{\Delta}\mathbf{Y}_k = \mathbf{F}_k \mathbf{B}_k \quad (4.26)$$

with

$$\Delta \mathbf{Y}_k = \begin{pmatrix} u_k^{(c)} - u_k \\ v_k^{(c)} - v_k \end{pmatrix} \quad (4.27)$$

$$\mathbf{F}_k = \begin{pmatrix} -1 & 0 & \tan \phi & v_k \tan \phi & u_k \tan \phi & v_k \\ 0 & -1 & \tan \psi & u_k \tan \psi & u_k \tan \psi & -u_k \end{pmatrix} \quad (4.28)$$

$$\mathbf{B}_k = \begin{pmatrix} \Delta u_k \\ \Delta v_k \\ \Delta w_k \\ \Delta \alpha_k \\ \Delta \beta_k \\ \Delta \gamma_k \end{pmatrix} \quad (4.29)$$

Here, we stack all regressions sharing a common vector of alignment parameters, i.e., all regressions for a detector plane:

$$\mathbf{Y} = \mathbf{F}\mathbf{B} + \mathbf{E} \quad (4.30)$$

with

$$\mathbf{Y} = \begin{pmatrix} u_1^{(c)} \\ v_1^{(c)} \\ \dots \\ u_n^{(c)} \\ v_n^{(c)} \end{pmatrix} \quad \mathbf{F} = \begin{pmatrix} \mathbf{F}_1 \\ \dots \\ \mathbf{F}_n \end{pmatrix} \quad \mathbf{E} = \begin{pmatrix} \epsilon_1 \\ \eta_1 \\ \dots \\ \epsilon_n \\ \eta_n \end{pmatrix} \quad (4.31)$$

$n$  runs over all hits in a detector. Therefore, the error covariance matrix  $\mathbf{E}$  is diagonal even if multiple scattering is taken into account.

We again have a linear regression problem, with solution given by

$$\hat{\mathbf{B}} = (\mathbf{F}^T \mathbf{F})^{-1} \mathbf{F}^{-1} \mathbf{Y} \quad (4.32)$$

We don't actually have to build matrices  $\mathbf{F}$  or  $\mathbf{Y}$ ; instead, we can iteratively build matrices  $\mathbf{F}^T \mathbf{F}$  and  $\mathbf{F}^T \mathbf{Y}$  noting that

$$\mathbf{F}^T \mathbf{F} = \sum_{(k)} \mathbf{F}_k^T \mathbf{F}_k \quad \mathbf{F}^T \mathbf{Y} = \sum_{(k)} \mathbf{F}_k^T \mathbf{Y}_k \quad (4.33)$$

;

so in implementation, the dimensionality of the problem does not depend on the number of tracks.

Other statistics, such as parameter error covariances, prediction covariances etc. are calculated in the usual manner.

## 4.3 Implementation

Basically, an alignment session consists in repeating fitting/alignment steps combined with increasingly stringent cuts on chi-square for individual tracks, until alignment stabilizes and there is no improvement in the quality of fits.

### 4.3.1 The Fitter and Aligner classes

With this scenario in mind, our implementation is based on two classes, the fitter and the aligner. The fitter gets track and detector data and produces fit parameters and related statistics. The aligner gets tracks one by one and iteratively recomputes the regression matrices for all detectors. After all tracks are added, the aligner produces, for a required detector, the alignment parameters, their errors and other statistics.

We note that detector planes are aligned independently based on the latest set of track fits. It is thus straightforward to align a detector not included in the fit, or to align only a subset of detectors in a given iteration. The disadvantage is that there is no direct information about correlations of alignment parameters between different detector planes - the equations are independent and the dependencies form in the alignment-refitting-alignment cycle.

### 4.3.2 Stability of alignment

Both the fitter and the aligner use singular value decomposition for inversion of information matrices ( $\mathbf{F}^T \mathbf{F}$ ). This is particularly important for the aligner, as the regression problem for alignment can be seriously underdetermined - in other words, the sample of tracks may not contain enough information to determine each alignment parameter reliably. In such case, the desirable output value would be zero or at least small - and this is what can actually be achieved using the SVD.

SVD itself has a shrinking property, meaning that it will return the shortest vector as a solution to an underdetermined system of linear equations. Moreover, by ignoring eigenvectors belonging to small eigenvalues we can guarantee that eigenvectors representing a minor fraction of overall variation in the data will be disregarded, and will not tell on the values of regression parameters. In alignment, this has twofold effect:

- it provides stability of alignment parameters when tracks contain insufficient information about positions of detector planes
- it seems to also provide *spatial* stability of alignment - we do not need to fix some detector planes to prevent rotations and shifts of the detector system as a whole. Such overall rotations and shifts do not change overall chisquare, or change it only

slightly due to random variations, and thus correspond to combinations of alignment parameters corresponding to zero or small singular eigenvalues. Elimination of the corresponding eigenvectors provides spatial stability of alignment.

Therefore, for the alignment SVD we use tolerance of 0.0001, meaning that eigenvectors with eigenvalues smaller than  $0.0001 \times$  the largest eigenvalue will be disregarded.

### 4.3.3 Iteration schedules

Our alignment-refitting scheme does not provide any safeguard against falling into a local minimum and we have to use cautious alignment schedules similar to those used with other alignment methods.

Starting with detector planes in reference positions, we first (in the first alignment-refitting cycle) align the  $x$  and  $y$  positions of the detector planes (i.e., shifts perpendicular to system axis), then we adjust rotations around the system axis  $z$ . Only then we can proceed to adjustments of other rotations and shifts along the system axis.

The alignment-refitting steps are combined with increasingly stringent cuts on chi-square of tracks, meaning that only tracks with chisquare below a given threshold are used in the calculation of alignment parameters. The cut is defined as a quantile of the actual distribution of chisquare for all tracks.

For our data sets, slow schedules with chi-square cut decreasing from 99% down to about 85% were sufficient to achieve convergence of alignment parameters.

### 4.3.4 Uncertainties in alignment parameters

The alignment-refitting scheme iteratively updates alignment parameters. Due to this incremental updating, the error covariances from the alignment regression don't provide useful estimates of alignment parameter errors: as the updates become statistically insignificant at later iterations, the error covariances inflate and become unreasonably large.

We therefore use another approach to estimate parameter uncertainties - bootstrap re-sampling.

We generate a large number of replicas of the original data files by resampling regression residuals as follows:

1. For each fitted track, generate a new track by combining regression parameters of that track with residuals from another, randomly selected track.
2. With the new set of tracks, calculate alignment parameters and resolutions, and store.
3. Starting from item 1, repeat as many times as necessary to obtain a large enough sample to estimate distributions of estimated parameters.

The logic is that we generate replicas based on actual data without introducing additional assumptions (such as of normality or independence). The replicas preserve the distributions of residuals on individual detectors, and correlations between residuals are preserved along tracks.

Though this is computationally expensive, the advantages are in conceptual simplicity of the method and, in our setting, in that we use the same method to estimate errors in alignment parameters and in detector resolutions.

## 4.4 Results

Alignment was carried out using the same procedure on the Prague and Bonn datasets, as well as on a large series of files containing replicated tracks to assess uncertainties in alignment parameters. The procedure consisted of 7 iterations described in Table 4.1.

Iteration	$\chi_2$ cut (p)	Parameters
1	0.99	$\Delta u, \Delta v$
2	0.97	$\Delta u, \Delta v, \Delta \gamma$
3	0.97	$\Delta u, \Delta v, \Delta \gamma$
4	0.95	all 6
5	0.92	all 6
6	0.89	all 6
7	0.86	all 6

Table 4.1: The alignment scheme used in the analysis of both datasets and bootstrap replicas. The table indicates chi-square cuts used and parameters updated at each step.

Uncertainties in alignment parameters were calculated using bootstrap resampling from the original data files. Results of alignment are shown in Table 4.2. Apparently, the resulting alignment is the same for both datasets in the sense that differences are well within estimated errors.

## 4.5 Alignment conclusions

Using both datasets, we arrived at virtually the same set of alignment parameters. Uncertainties in alignment parameters calculated by bootstrap resampling indicate that the estimates are stable.

Our experience with the linearized alignment/refitting scheme can be summarized as follows:

1. The method is suitable for alignment of simple testbeam setups.

Table 4.2: Detector alignment parameters for the Prague and Bonn datasets. Tracks were selected from the base set using the iterative PCA classifier. In the alignment,  $\chi^2$  cut was decreased in 7 steps from  $p = 0.99$  down to  $p = 0.86$ . Only main alignment parameters are shown; the other two rotations and shift along  $z$  were consistently zero and are not shown. Parameter errors were estimated from distributions obtained by resampling (see 4.3.4) using 1000 replicas of the original file.

Source		Prague set		Bonn set	
Number of tracks		308		1892	
Parameter	Unit	Value	Error	Value	Error
DETECTOR 1					
$u$ shift	$\mu\text{m}$	-29.35	0.55	-28.62	0.25
$v$ shift	$\mu\text{m}$	-39.97	0.58	-40.76	0.28
$z$ rotation	mrad	-0.01	0.42	-0.01	0.21
DETECTOR 2					
$u$ shift	$\mu\text{m}$	-39.72	0.63	-40.20	0.29
$v$ shift	$\mu\text{m}$	320.70	0.63	321.51	0.35
$z$ rotation	mrad	0.00	0.51	0.00	0.27
DETECTOR 3					
$u$ shift	$\mu\text{m}$	168.46	0.78	167.88	0.33
$v$ shift	$\mu\text{m}$	-166.45	0.51	-166.86	0.24
$z$ rotation	mrad	0.01	0.47	0.01	0.23
DETECTOR 4					
$u$ shift	$\mu\text{m}$	-87.51	1.16	-88.49	0.40
$v$ shift	$\mu\text{m}$	-459.47	0.62	-458.15	0.25
$z$ rotation	mrad	0.00	0.57	0.00	0.22
DETECTOR 5					
$u$ shift	$\mu\text{m}$	-9.54	0.71	-8.57	0.29
$v$ shift	$\mu\text{m}$	347.09	0.41	346.21	0.18
$z$ rotation	mrad	-0.01	0.41	-0.01	0.19

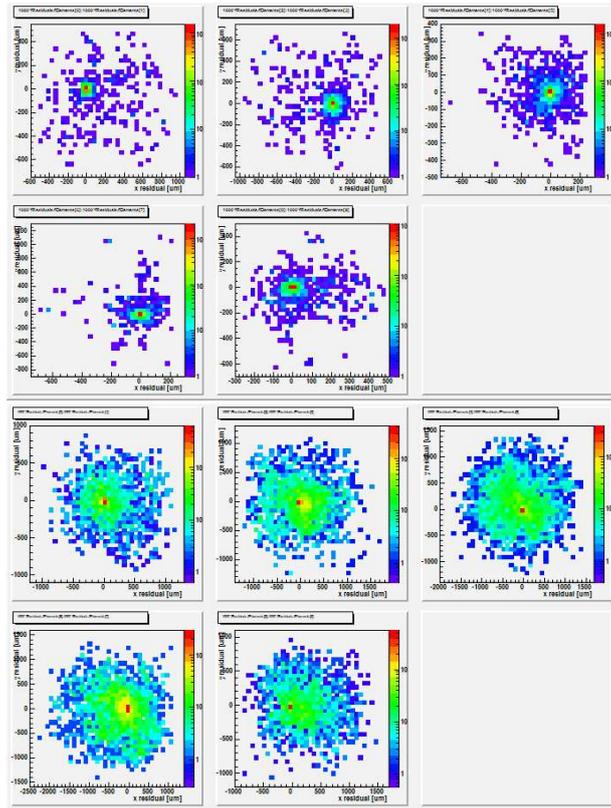


Figure 4.1: 2D histograms showing the distribution of regression residuals for the Bonn (top frame) and Prague (bottom frame) datasets.

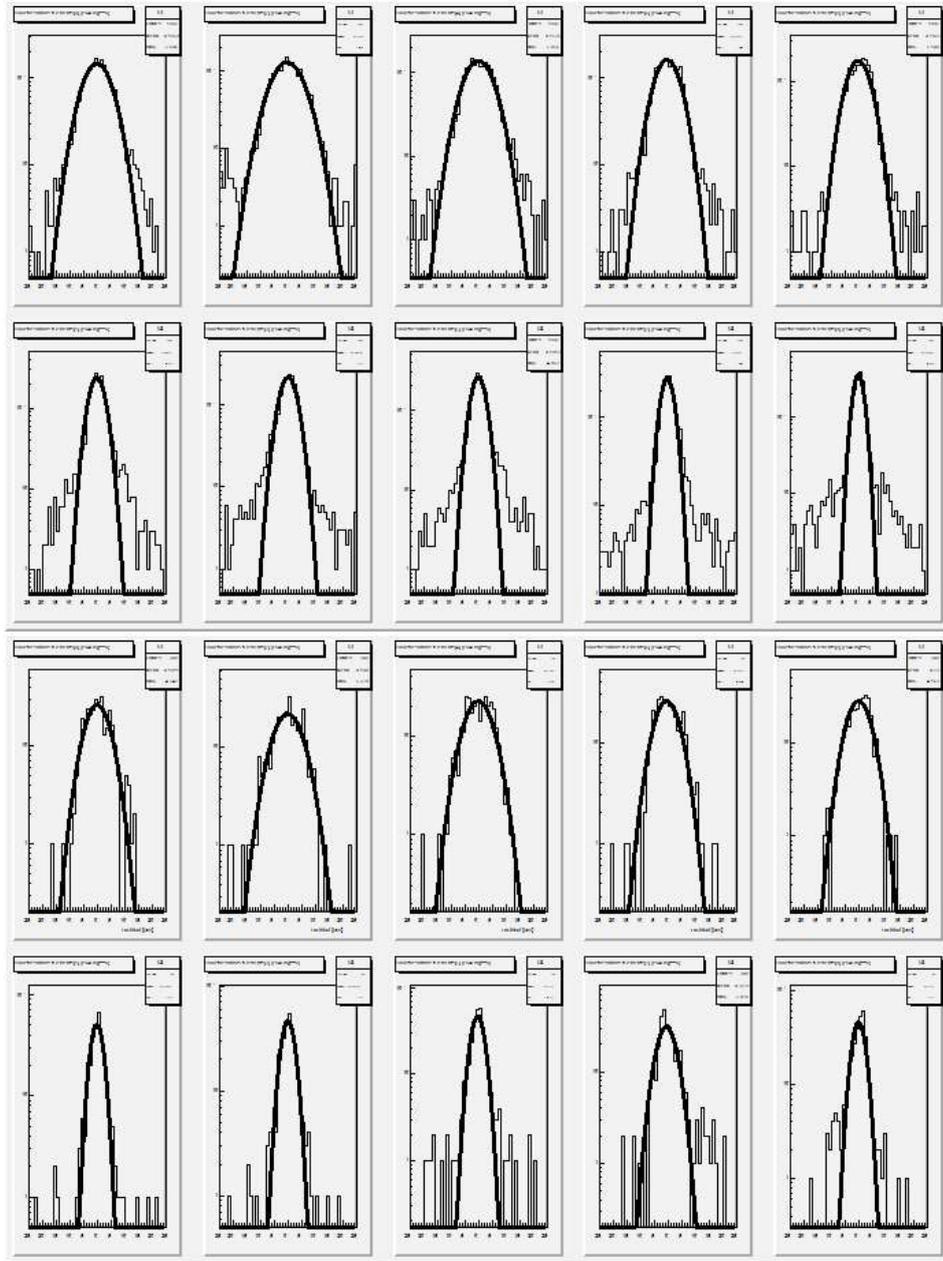


Figure 4.2: Histograms of regression residuals for the Bonn (top two rows) and Prague (bottom two rows) datasets. The distributions are centered around zero and fairly symmetrical.

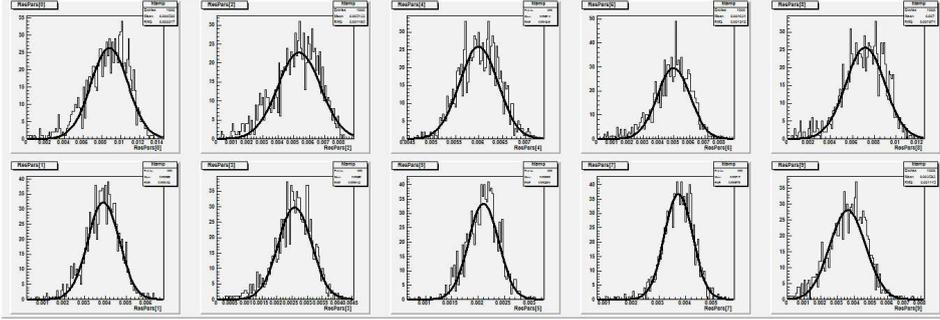


Figure 4.3: Bootstrap distributions of alignment parameters for the Prague dataset. Apparently, the distributions are well localized and the representation of alignment uncertainties as RMS of the distributions is appropriate. Note, however, that the parameters are strongly correlated.

2. In terms of avoiding local minima and alignment overfitting, the method provides only a little advantage over alignment methods based on non-linear optimization. The linearity of estimates combined with iterations provides some control against overfitting.
3. For the simple setup we have used here, we did not need to fix any detector planes as spatial reference. We believe this property is related to the shrinking property of the singular value decomposition used in the calculation of alignment parameter estimates.

## 5 Estimation of detector resolutions

Here we derive formulas for estimation of detector resolution in the presence of multiple scattering. We give formulas for two common cases - when the detector (i) is included, and (ii) is not included in the track fit.

To properly account for multiple scattering, we will use the "kinked" track model [3] for calculation of detector resolutions. To be able to do this for misaligned detectors, we need a formula for alignment update of crossing points coordinates, that is, an analogue of Eqn. 4.14 for kinked tracks.

The corresponding formula is, however, somewhat cumbersome and leads to formalism that we find hardly tractable. Similarly to non-linearities in slopes, we have to cope with products of angular deflections, and thus with non-gaussian noise in the model. Moreover, unlike with linear tracks, misalignments *change kinked tracks*, as scattering events occur at different points in space, introducing explicit correlations between alignment corrections of different detector planes.

Therefore, for the purpose of this note and in view of the results of alignment, we restrict our treatment only to misalignment of detector planes that does not materially move the planes in space - meaning we only consider shifts along the  $x$  and  $y$  axes, and rotations around the  $z$  axis. This is equivalent to only considering misalignment corrections up to the first order in track direction.

### 5.1 Kinked tracks in misaligned detectors

We start with the formula for alignment update of crossing point coordinates (Eqn. 4.14), ignoring terms that are second and higher order in track directions:

$$\begin{aligned}u_k^c &= u_k - \Delta u_k + \Delta \gamma_k v_k \\v_k^c &= v_k - \Delta v_k - \Delta \gamma_k u_k\end{aligned}\tag{5.1}$$

Now we use the track model to express tracks in the reference position of the detectors, which is now, for the purpose of final fitting and calculations of detector resolutions, the "kinked" track model [3]. Note that we only use the model to account for multiple scattering-induced error correlations; the "optimum" track fitting formulas of [3] don't provide measurable improvement in predictions when multiple scattering is low, as is the case with our data.

Thus, the equations of a particle track are

$$\begin{aligned}
u_k &= u_0 + z_k \tan \phi + \sum_{j < k} (z_k - z_j) \epsilon_j^{(x)} + d_k^{(u)} \\
v_k &= v_0 + z_k \tan \psi + \sum_{j < k} (z_k - z_j) \epsilon_j^{(y)} + d_k^{(v)} \\
j &= 1, 2, \dots, m \\
k &= j_1, j_2, \dots, j_n
\end{aligned} \tag{5.2}$$

and hold in the reference position where  $(u, v, w)$  and  $(x, y, z)$  coincide. Here  $j$  runs over all  $m$  scattering planes in the system (such as scintillators, aluminium foils, detectors etc.), and  $k$  only runs over measuring detectors.  $u_0$  and  $v_0$  give the position of the initial/reference point of the track,  $\tan \phi$  and  $\tan \psi$  define initial direction of the track.  $\epsilon_j^{(x)}$  and  $\epsilon_j^{(y)}$  are deflections of the track due to scattering on  $j$ -th scattering plane in directions  $x$  and  $y$ , respectively;  $d_k^{(u)}$  and  $d_k^{(v)}$  are measurement errors in the  $k$ -th detector.

We rewrite equations 5.1 and 5.2 in a more compact form as follows:

$$\begin{pmatrix} u_k^c \\ v_k^c \end{pmatrix} = \begin{pmatrix} -\Delta u_k \\ -\Delta v_k \end{pmatrix} + \mathbf{F}_k \mathbf{b} + \mathbf{G}_k \mathbf{e} + \mathbf{d}_k \tag{5.3}$$

with  $\mathbf{F}_k$  being  $2 \times 4$ ,

$$\mathbf{F}_k = \begin{pmatrix} 1 & z_k \end{pmatrix} \otimes \mathbf{R}_k, \quad \mathbf{R}_k = \begin{pmatrix} 1 & \Delta \gamma_k \\ -\Delta \gamma_k & 1 \end{pmatrix}, \tag{5.4}$$

$\mathbf{G}_k$  being  $2 \times m$  ( $m$  is the number of scattering planes),

$$\mathbf{G}_k = \begin{pmatrix} z_k - z_1 & z_k - z_2 & \dots & 0 & 0 \end{pmatrix} \otimes \mathbf{R}_k, \tag{5.5}$$

and

$$\mathbf{b} = \begin{pmatrix} u_0 \\ v_0 \\ \tan \phi \\ \tan \psi \end{pmatrix}, \quad \mathbf{e} = \begin{pmatrix} \epsilon_1^{(x)} \\ \epsilon_1^{(y)} \\ \dots \\ \epsilon_m^{(x)} \\ \epsilon_m^{(y)} \end{pmatrix}, \quad \mathbf{d} = \begin{pmatrix} d_k^{(u)} \\ d_k^{(v)} \end{pmatrix}. \tag{5.6}$$

The resulting formula for the whole track is obtained by stacking equations 5.3:

$$\mathbf{u}^c \equiv \begin{pmatrix} u_1^c \\ v_1^c \\ \dots \\ u_n^c \\ v_n^c \end{pmatrix} = -\Delta \mathbf{u} + \mathbf{F} \mathbf{b} + \mathbf{G} \mathbf{e} + \mathbf{d} \tag{5.7}$$

with  $\mathbf{b}$  and  $\mathbf{e}$  being the same as in Eqn. 5.3, and

$$\mathbf{F} = \begin{pmatrix} (1 \ z_1) \otimes \mathbf{R}_1 \\ \dots \\ (1 \ z_n) \otimes \mathbf{R}_n \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} (z_2 - z_1 \quad 0 \quad \dots \quad 0) \otimes \mathbf{R}_1 \\ \vdots \\ (z_n - z_1 \quad z_n - z_2 \quad \dots \quad z_n - z_m) \otimes \mathbf{R}_n \end{pmatrix}, \quad (5.8)$$

$$\Delta \mathbf{u} = \begin{pmatrix} \Delta u_1 \\ \Delta v_1 \\ \dots \\ \Delta u_n \\ \Delta v_n \end{pmatrix}, \quad \mathbf{d} = \begin{pmatrix} d_1^{(u)} \\ d_1^{(v)} \\ \dots \\ d_n^{(u)} \\ d_n^{(v)} \end{pmatrix}. \quad (5.9)$$

$\mathbf{d}$  and  $\mathbf{e}$  are random vectors,

$$\begin{aligned} \langle d_k \rangle &= 0 & \langle d_k d_l \rangle &= \delta_{kl} \Delta_k^2 & k, l &= 1, \dots, n \\ \langle e_i \rangle &= 0 & \langle e_i e_j \rangle &= \delta_{ij} \sigma_i^2 & i, j &= 1, \dots, m \end{aligned} \quad (5.10)$$

or, in matrix form,

$$\begin{aligned} \langle \mathbf{d} \rangle &= \mathbf{0} & \langle \mathbf{d} \mathbf{d}^T \rangle &= \mathbf{\Delta}^2 \\ \langle \mathbf{e} \rangle &= \mathbf{0} & \langle \mathbf{e} \mathbf{e}^T \rangle &= \mathbf{\Sigma}^2 \end{aligned} \quad (5.11)$$

with both  $\mathbf{\Delta}^2$  and  $\mathbf{\Sigma}^2$  being diagonal.

## 5.2 Formulas for track parameters and detector resolutions

In the previous section, we formulated our model of tracks, and now we formulate the regression problem and give formulas for estimates of track parameters and detector resolutions.

### 5.2.1 Formulation

Our task is now as follows: Given

- observations  $u_k^c, v_k^c, k = 1, \dots, n$  for a large number  $N$  of tracks,
- detector geometry and alignment parameters  $z_k, \Delta u_k, \Delta v_k$ , and  $\Delta \gamma_k, k = 1, \dots, n$  for all detector planes, and
- distributions of scattering deflections  $\epsilon^{(x)}$  and  $\epsilon^{(y)}$ , that is, mean deflection angles  $\sigma_j^{(x)} = \sigma_j^{(y)} \equiv \sigma_k$  for each scattering plane and assuming Gaussian distribution with possible long tails,

estimate

- $4N$  track parameters  $\mathbf{b}$ , and

- $2n$  detector resolutions  $\Delta_k^{(u)} \equiv \langle (d_k^{(u)})^2 \rangle^{1/2}$  and  $\Delta_k^{(v)} \equiv \langle (d_k^{(v)})^2 \rangle^{1/2}$ .

### Scattering angles

Estimates of deflection angles can be obtained using standard formulas [5] or from simulations (such as GEANT), based on material properties of scatterers. For this work, we obtained the estimates from the formulas of Regler and Fruehwirth [5], and found them to be in excellent agreement with values obtained from GEANT4 simulation data summarized in a companion DEPFET note [7].

### 5.2.2 Estimates of track parameters

We estimate track parameters  $\mathbf{b}$  by fitting a straight line to the data, that is, by minimizing the sum of squared deviations of the data from the line:

$$S^2 = Tr \left[ (\mathbf{u}^c + \Delta \mathbf{u} - \mathbf{F} \mathbf{b}_R)^T (\mathbf{u}^c + \Delta \mathbf{u} - \mathbf{F} \mathbf{b}_R) \right]. \quad (5.12)$$

with respect to track parameters  $\mathbf{b}_R$ . Note that we use simple linear fit to fit tracks; we will plug in our better model (Eqn. 5.7) later to calculate error covariances.

Differentiating with respect to  $\mathbf{b}_R$  and setting the derivative to zero, we obtain the standard formula

$$\hat{\mathbf{b}}_R = (\mathbf{F}^T \mathbf{F})^{-1} \mathbf{F}^T (\mathbf{u}^c + \Delta \mathbf{u}) \quad (5.13)$$

Now it is time to plug in our kinked-track model: we know that in fact  $\mathbf{u}^c = -\Delta \mathbf{u} + \mathbf{F} \mathbf{b} + \mathbf{G} \mathbf{e} + \mathbf{d}$ ; substituting into 5.24 we get

$$\hat{\mathbf{b}}_R = \mathbf{b} + (\mathbf{F}^T \mathbf{F})^{-1} \mathbf{F}^T \mathbf{G} \mathbf{e} + (\mathbf{F}^T \mathbf{F})^{-1} \mathbf{F}^T \mathbf{d} \quad (5.14)$$

From this we see that

$$\langle \hat{\mathbf{b}}_R \rangle = \mathbf{b}, \quad (5.15)$$

and

$$\begin{aligned} cov(\hat{\mathbf{b}}_R) &\equiv \langle (\hat{\mathbf{b}}_R - \mathbf{b})(\hat{\mathbf{b}}_R - \mathbf{b})^T \rangle \\ &= (\mathbf{F}^T \mathbf{F})^{-1} \mathbf{F}^T (\mathbf{G} \Sigma^2 \mathbf{G}^T + \Delta^2) \mathbf{F} (\mathbf{F}^T \mathbf{F})^{-1} \end{aligned} \quad (5.16)$$

The important thing here is that the estimate of track parameters  $\hat{\mathbf{b}}_R$  does not depend on detector resolutions (however, its covariance does).

In the general case when we cannot rely on detector resolutions or scattering angles being equal for all detectors, a somewhat simpler expression for parameter errors can be obtained if we only ask for diagonal elements of the covariance matrix - that is, if we don't care for parameter error correlations. Using Eqn. 7.10 twice we get

$$diag^{-1} [cov(\hat{\mathbf{b}}_R)] = [(\mathbf{F}^T \mathbf{F})^{-1} \mathbf{F}^T]^{\circ 2} (\mathbf{G}^{\circ 2} diag^{-1}(\Sigma^2) + diag^{-1}(\Delta^2)) \quad (5.17)$$

with  $\mathbf{A}^{\circ 2} \equiv \mathbf{A} \circ \mathbf{A}$  and  $\circ$  being the Hadamard (elementwise) matrix product;  $diag^{-1} \mathbf{A}$  is a column vector of diagonal elements of  $\mathbf{A}$ , see Appendix 1.

### 5.2.3 Predictions and detector resolutions

Predictions of the linear fit are

$$\hat{\mathbf{u}}^c = -\Delta\mathbf{u} + \mathbf{F}\hat{\mathbf{b}}_R = -\Delta\mathbf{u} + \mathbf{F}\mathbf{b} + \mathbf{F}(\mathbf{F}^T\mathbf{F})^{-1}\mathbf{F}^T(\mathbf{G}\mathbf{e} + \mathbf{d}) = \mathbf{u}^c - \mathbf{H}(\mathbf{G}\mathbf{e} + \mathbf{d}) \quad (5.18)$$

with

$$\mathbf{H} = \mathbf{1} - \mathbf{F}(\mathbf{F}^T\mathbf{F})^{-1}\mathbf{F}^T \quad (5.19)$$

We see that the predictions are unbiased and the covariance matrix of residuals is

$$\text{cov}(\hat{\mathbf{u}}^c) \equiv \langle (\mathbf{u}^c - \hat{\mathbf{u}}^c)(\mathbf{u}^c - \hat{\mathbf{u}}^c)^T \rangle = \mathbf{H}(\mathbf{G}\Sigma^2\mathbf{G}^T + \Delta^2)\mathbf{H} \quad (5.20)$$

$\mathbf{H}$  is symmetric and a projection matrix,  $\mathbf{H}\mathbf{H} = \mathbf{H}$ . Using a proper estimate of the covariance matrix of residuals, this equation, at least in principle, allows to estimate detector resolutions  $\Delta^2$ . There are, however, two technical difficulties:

1. Covariance estimates typically have bad statistical properties, namely in their inconsistency - in our case, we have to expect that the estimates improve very slowly with increasing number of tracks. The tails of Moliere distributions make this even worse. This can be partially improved by robust estimators, i.e., by taking medians, modes, or Gaussian fits to central part of distributions rather than means of products of residuals as estimates of covariance matrix elements.
2. The other difficulty is in the number of detector resolutions we can effectively obtain by solving Eqn. 5.20. Using the argument of Frühwirth [6],  $\mathbf{H}$  projects from  $2n$  dimensions to  $2n - 4$ , so its rank is  $2n - 4$ , and, in view of Eqn. 5.18, so is the rank of the residual covariance matrix. Both matrices are symmetric in both coordinates, so there exist at most  $2 \times (n - 2)(n - 1)/2$  independent linear combinations of  $\Delta_k^2$ 's and  $\Sigma_j^2$ 's. There is also another limit: in our case of weak multiple scattering, the residual covariance matrix will be strongly diagonally-dominant, leaving us at most with the  $n$  diagonal elements, since (in view of the preceding item) we can't hope to estimate the small non-diagonal elements reliably. So the maximum number of detector resolutions that we can estimate independently is  $n$  or  $(n - 1)(n - 2)$ , whichever is smaller.

We will start with the second complication and defer the first to a later section. As the residual covariance matrix is diagonally dominant and matrices  $\Delta^2$  and  $\Sigma^2$  are diagonal, we can again use Eqn. 7.10 from the Appendix to write

$$\text{diag}^{-1}[\text{cov}(\hat{\mathbf{u}})] = \mathbf{H}^{\circ 2}(\mathbf{G}^{\circ 2}\text{diag}^{-1}(\Sigma^2) + \text{diag}^{-1}(\Delta^2)) \quad (5.21)$$

Using (singular, if necessary) inverse of  $\mathbf{H}^{\circ 2}$ , we can write

$$\text{diag}^{-1}(\Delta^2) = -\mathbf{G}^{\circ 2}\text{diag}^{-1}(\Sigma^2) + [\mathbf{H}^{\circ 2}]^{-1}\text{diag}^{-1}[\text{cov}(\hat{\mathbf{u}})] \quad (5.22)$$

that is,

$$\begin{pmatrix} \Delta_1^{(u)2} \\ \Delta_1^{(v)2} \\ \dots \\ \Delta_n^{(v)2} \end{pmatrix} = -\mathbf{G}^{\circ 2} \begin{pmatrix} \sigma_1^{(x)2} \\ \sigma_1^{(y)2} \\ \dots \\ \sigma_m^{(y)2} \end{pmatrix} + [\mathbf{H}^{\circ 2}]^{-1} \begin{pmatrix} \langle \delta u_1^2 \rangle \\ \langle \delta v_1^2 \rangle \\ \dots \\ \langle \delta v_n^2 \rangle \end{pmatrix}$$

Using this formula, it is possible to calculate detector resolutions without infinite energy extrapolation. For this procedure to work, however, we need at least five points per track.

#### 5.2.4 Formulas for a detector not included in the fit

Let us now consider the case of a detector not included in the fit. The detector, denoted by  $d$ , is "mute" - it does not provide data for regression, but contributes to multiple scattering. From the model (Eqn. 5.7, we have for detector  $d$

$$\mathbf{u}_d^c = -\Delta \mathbf{u} + \mathbf{F}_d \mathbf{b} + \mathbf{G}_d \mathbf{e} + \mathbf{d}_d \quad (5.23)$$

Estimate of track parameters based on data of the other detectors is

$$\hat{\mathbf{b}}_{\mathbf{R}-d} = (\mathbf{F}_{-d}^T \mathbf{F}_{-d})^{-1} \mathbf{F}_{-d}^T (\mathbf{u}_{-d}^c + \Delta \mathbf{u}_{-d}) \quad (5.24)$$

The subscript  $_{-d}$  indicates that the corresponding matrix/vector contains data for all detectors except detector  $d$ , and subscript  $_d$  denotes the missing part. Plugging in the kinked-track model, we get

$$\hat{\mathbf{b}}_{\mathbf{R}-d} = \mathbf{b} + (\mathbf{F}_{-d}^T \mathbf{F}_{-d})^{-1} \mathbf{F}_{-d}^T (\mathbf{G}_{-d} \mathbf{e} + \mathbf{d}_{-d}) \quad (5.25)$$

The prediction of the linear fit for detector  $d$  is

$$\hat{\mathbf{u}}_d^c = -\Delta \mathbf{u} + \mathbf{F}_d \hat{\mathbf{b}}_{\mathbf{R}-d} = \mathbf{u}_d - \mathbf{G}_d \mathbf{e} - \mathbf{d}_d + \mathbf{F}_d (\mathbf{F}_{-d}^T \mathbf{F}_{-d})^{-1} \mathbf{F}_{-d}^T (\mathbf{G}_{-d} \mathbf{e} + \mathbf{d}_{-d}) \quad (5.26)$$

We see that the predictions are unbiased and the covariance matrix of residuals is

$$\begin{aligned} \text{diag}^{-1} [\text{cov}(\hat{\mathbf{u}}_d)] &\equiv \left\langle (\mathbf{u}_d - \hat{\mathbf{u}}_d) (\mathbf{u}_d - \hat{\mathbf{u}}_d)^T \right\rangle = \\ &= \text{diag}^{-1} (\Delta_d^2) + \left[ \mathbf{F}_d (\mathbf{F}_{-d}^T \mathbf{F}_{-d})^{-1} \mathbf{F}_{-d}^T \right]^{\circ 2} \text{diag}^{-1} (\Delta_{-d}^2) \\ &\quad + \left[ \mathbf{F}_d (\mathbf{F}_{-d}^T \mathbf{F}_{-d})^{-1} \mathbf{F}_{-d}^T \mathbf{G}_{-d} - \mathbf{G}_d \right]^{\circ 2} \text{diag}^{-1} (\boldsymbol{\Sigma}^2) \end{aligned} \quad (5.27)$$

with the three terms on the right being the contributions of measurement error on detector  $d$ , of prediction error of the linear fit, and of multiple scattering.

We also introduce *telescope resolutions*, defined for a given detector as uncertainties of impact point predictions based on data of all other detectors. Based on the previous equation, we can write

$$\text{diag}^{-1} [\text{cov}(\hat{\mathbf{u}}_d)] = \text{diag}^{-1} (\Delta_d^2) + \text{diag}^{-1} (\Delta_{TELd}^2) \quad (5.28)$$

and

$$diag^{-1}(\Delta_{TEld}^2) = \left[ \mathbf{F}_d (\mathbf{F}_{-d}^T \mathbf{F}_{-d})^{-1} \mathbf{F}_{-d}^T \right]^{\circ 2} diag^{-1}(\Delta_{-d}^2) \quad (5.29)$$

$$+ \left[ \mathbf{F}_d (\mathbf{F}_{-d}^T \mathbf{F}_{-d})^{-1} \mathbf{F}_{-d}^T \mathbf{G}_{-d} - \mathbf{G}_d \right]^{\circ 2} diag^{-1}(\Sigma^2) \quad (5.30)$$

## 5.2.5 Notes

### A note on implementation

In our case of five detectors playing also the role of telescopes, the task of estimating resolutions of the five detectors is more complicated compared to the classical scenario with a single DUT in the midst of a set of telescopes with known resolutions. We therefore need to use both Eqn. 5.20 and Eqn. 5.27 in estimation of detector resolutions; in particular, we will use the detector resolution estimates from Eqn. 5.20 in the second term of Eqn. 5.27 to obtain more precise estimates.

### Alignment uncertainties

The above expressions for prediction error covariances do not include alignment uncertainties. Such inclusion would turn the simple regression we use into a regression with errors in both variables (alignment uncertainties make the factor matrix  $\mathbf{F}$  uncertain). Moreover, while we could well exclude some alignment parameters for the purpose of track fitting, for an error analysis to be consistent we have to include uncertainties in all alignment parameters, including those we have set to zero because our data did not contain sufficient information to estimate them.

The solution is to calculate errors using bootstrap resampling, as described in section 4.3.4.

## 5.3 Implementation

The formalism used for estimation of detector resolution slightly differs from the alignment/refitting scheme. We implemented the resolution estimator as a C++/ROOT class, which provides the fitting/resolution estimation functionality.

## 5.4 Results

The calculated detector resolutions are listed in Table 5.1, and the differences are graphically illustrated in Fig. 5.1. These values are "exclusive" errors calculated using Eqn. 5.27

The values generally agree in the sense of the differences being well within error; the most apparent difference is in the fine ( $v$ ) resolution on detector 4. Figure 5.2 shows normalized

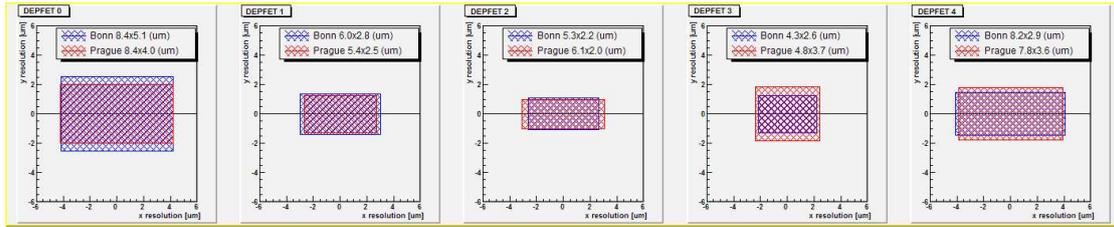


Figure 5.1: Detector resolutions calculated from the Bonn (blue) and Prague (red) datasets. The most apparent difference is on detector 4.

residuals (“pulls”) for track fits. The distributions are expected to be standard normal. We see that there are some irregularities in the fine coordinate on detectors 4 and 5 in the Prague dataset. This may indicate a residual misalignment, persistent due to the small size of the Prague dataset.

Table 5.2 lists the three types of resolutions and is meant to illustrate two points:

- The exclusive and inclusive resolutions are very similar.
- The telescope resolutions are all noticeably larger than 1 micron, the smallest being  $1.7 \mu\text{m}$ .

Table 5.3 and Fig. 5.3 show the results concerning the stability of detector resolution estimates. For each dataset, we generated 1000 files containing the corresponding number of simulated tracks selected at random from the pool of 50,000 tracks and adapted to match the spatial distribution of actual tracks and calculated detector resolutions. These files were analyzed in the same manner as actual data and results stored.

The purpose of these data is twofold:

- They help to assess the stability of estimates - the distributions of simulated data should be located near the actual values.
- They help to assess the precision of resampling estimates of resolution uncertainties.

The simulation data indicate good stability of detector resolution estimates, and we apparently can also rely on the bootstrap estimates of errors.

Together, the bootstrap and simulated distributions also indicate that we do not have seriously biased selections of tracks.

## 5.5 Conclusions

We conclude that we have obtained reliable (as confirmed by simulations) estimates of detector resolutions. The resolutions of individual detectors are substantially different.

Table 5.1: Detector resolutions for the Prague and Bonn datasets. Tracks were selected from the base set using the iterative PCA classifier. For calculation of detector resolutions,  $\chi^2$  cut at  $p = 0.99$  was used. Parameter errors were estimated from distributions obtained by resampling (see 4.3.4) using 1000 replicas of the original file.

Source		Prague set		Bonn set	
Number of tracks		308		1892	
Parameter	Unit	Value	Error	Value	Error
DETECTOR 1					
$u$ resolution	$\mu\text{m}$	8.4	2.0	8.4	0.9
$v$ resolution	$\mu\text{m}$	4.0	0.8	5.1	0.3
DETECTOR 2					
$u$ resolution	$\mu\text{m}$	5.4	1.4	6.0	0.5
$v$ resolution	$\mu\text{m}$	2.5	0.6	2.8	0.3
DETECTOR 3					
$u$ resolution	$\mu\text{m}$	6.1	0.4	5.3	0.2
$v$ resolution	$\mu\text{m}$	2.0	0.3	2.2	0.1
DETECTOR 4					
$u$ resolution	$\mu\text{m}$	4.8	1.1	4.3	0.5
$v$ resolution	$\mu\text{m}$	3.7	0.5	2.6	0.2
DETECTOR 5					
$u$ resolution	$\mu\text{m}$	7.8	1.8	8.2	0.7
$v$ resolution	$\mu\text{m}$	3.6	1.1	2.9	0.6

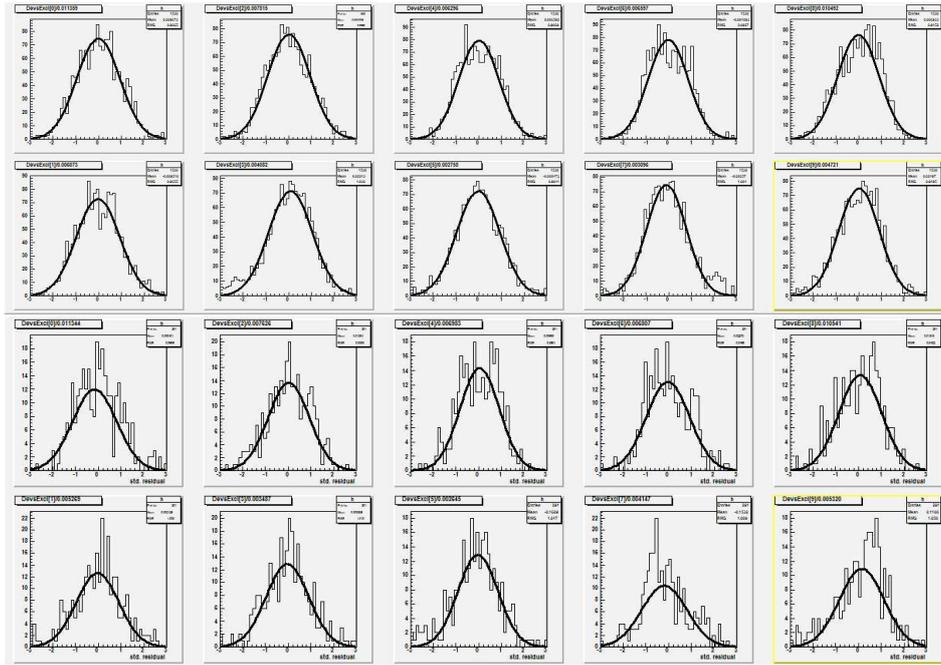


Figure 5.2: Histograms of pulls (normalized residuals) for the Bonn (top two rows) and Prague (bottom two rows) datasets. The Gaussians plotted with the histograms are standard normal distributions  $N[0, 1]$ , with only the norm (height) fitted to the data.

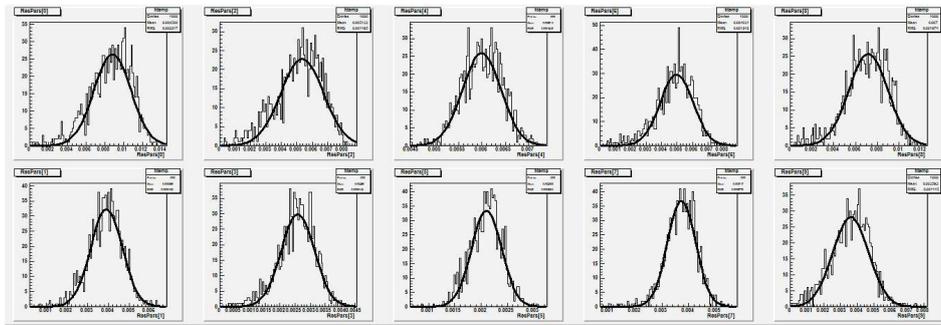


Figure 5.3: Resampling distributions of detector resolutions for the Prague dataset. In spite of moderate asymmetry, the distributions do not show serious anomalies and the representation of resolution uncertainties as RMSs of the distributions seems appropriate.

Table 5.2: Exclusive (Eqn. 5.27), inclusive (Eqn. 5.20) and telescope (Eqn. 5.29) resolutions for the Prague and Bonn data sets. Note the small difference between exclusive and inclusive resolutions. The telescope resolutions are RMS errors of predictions of impact point coordinate on the detector based on measurements on the remaining four detectors.

Source		Prague set			Bonn set		
Number of tracks		309			1823		
Resolution	Unit	Exclusive	Inclusive	Telescope	Exclusive	Inclusive	Telescope
DETECTOR 1							
$u$	$\mu\text{m}$	8.39	8.38	7.30	8.35	8.34	7.70
$v$	$\mu\text{m}$	4.01	3.99	3.23	5.11	5.10	3.28
DETECTOR 2							
$u$	$\mu\text{m}$	5.39	5.39	5.14	5.98	5.97	5.04
$v$	$\mu\text{m}$	2.54	2.54	2.41	2.77	2.77	3.00
DETECTOR 3							
$u$	$\mu\text{m}$	6.13	6.12	3.37	5.27	5.25	3.45
$v$	$\mu\text{m}$	2.00	1.97	1.72	2.15	2.12	1.73
DETECTOR 4							
$u$	$\mu\text{m}$	4.81	4.76	4.81	4.33	4.28	4.97
$v$	$\mu\text{m}$	3.66	3.59	2.08	2.55	2.47	1.75
DETECTOR 5							
$u$	$\mu\text{m}$	7.77	7.72	7.05	8.21	8.16	6.53
$v$	$\mu\text{m}$	3.56	3.44	4.23	2.92	2.78	3.71

Table 5.3: Mean values and errors obtained from data and from simulation. Simulation data were obtained by analysis of files containing tracks from GEANT4 simulations. Track samples were adjusted to match the distribution of impact points on the first detector and detector resolutions found in the data. It was however not possible to match the distribution of chi-square for the tracks, so the effective number of processed tracks was higher in simulation files (typically by about 10 %). This could explain smaller RMS values in simulated data. The results indicate that the estimates are fairly stable in the sense that the mean values coincide and simulation dispersions are never large.

Source	Prague set				Bonn set			
No. of tracks	309				1823			
Resolution	Data		Simulation		Data		Simulation	
	Mean	RMS	Mean	RMS	Mean	RMS	Mean	RMS
	DETECTOR 1							
$u$ ( $\mu\text{m}$ )	8.39	1.98	8.32	1.59	8.35	0.88	8.23	0.60
$v$ ( $\mu\text{m}$ )	4.01	0.76	3.99	0.68	5.11	0.34	5.10	0.29
	DETECTOR 2							
$u$ ( $\mu\text{m}$ )	5.39	1.40	5.31	1.10	5.98	0.52	5.97	0.42
$v$ ( $\mu\text{m}$ )	2.54	0.55	2.53	0.50	2.77	0.34	2.77	0.23
	DETECTOR 3							
$u$ ( $\mu\text{m}$ )	6.13	0.42	6.06	0.55	5.27	0.22	5.23	0.20
$v$ ( $\mu\text{m}$ )	2.00	0.29	1.99	0.27	2.15	0.13	2.16	0.09
	DETECTOR 4							
$u$ ( $\mu\text{m}$ )	4.81	1.07	4.83	1.01	4.33	0.54	4.36	0.45
$v$ ( $\mu\text{m}$ )	3.66	0.52	3.62	0.46	2.55	0.19	2.58	0.18
	DETECTOR 5							
$u$ ( $\mu\text{m}$ )	7.77	1.80	7.67	1.51	8.21	0.71	8.12	0.56
$v$ ( $\mu\text{m}$ )	3.56	1.07	3.61	0.86	2.92	0.60	2.98	0.35

The method described here allows to calculate the detector resolutions by explicitly accounting for multiple scattering, thus avoiding the need of infinite energy extrapolation. Another specific feature is the calculation of uncertainties in estimated parameters by bootstrap resampling, which, though computationally expensive, is an intuitively simple and reliable method of estimating parameter distributions.

## 6 Conclusions

We have presented here a set of analysis methods and used them to calculate resolutions of DEPFET detectors based on October 2006 testbeam data. We have obtained very similar results for two sets of tracks extracted from raw data by substantially different methods, and provided error estimates and simulation data to help assess the quality of our estimates.

Our results show substantial differences in resolutions of individual detectors.

The Bonn group [10] presents estimates of resolution only for detector 3, being  $3.76\ \mu\text{m}$  (coarse coordinate) and  $1.74\ \mu\text{m}$  (fine coordinate). These values are lower than those presented here ( $6.1$  and  $5.3\ \mu\text{m}$  for the coarse coordinate, and  $2.0$  and  $2.2\ \mu\text{m}$  for the fine coordinate).

The telescope resolution in the fine coordinate  $v$  on detector 3 (that is, the RMS error of predictions based on measurements of the other four detectors) is  $1.72\ \mu\text{m}$ . It would be about 1 micron if detectors 1,2,4, and 5 performed equally well as detector 3.

# Acknowledgement

We want to thank Robert Kohrs, Hans Kruger, Jaap Velthuis and Lars Reuen from Physikalisches Institut, University of Bonn, for their support in raw data processing and for kindly providing us with the data set used in this analysis. This project was supported by Grant Agency of the Czech Republic under project Nr. 114-13/201460. Peter Kvasnicka is supported by EU I3 Contract 026 126-R II3 (EUDET).

## 7 Appendix: Some matrix algebra

In this appendix, we explain some matrix algebra used in the text. For more details and an excellent overview of useful matrix algebra, we refer to Tom Minka's note [4].

### Cronecker product and $vec$

We denote by  $\otimes$  the Cronecker product of two matrices:

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \otimes \mathbf{B} = \begin{pmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} \end{pmatrix} \quad (7.1)$$

By  $vec(\mathbf{A})$  we denote a column vector made of stacked columns of  $\mathbf{A}$ :

$$vec(\mathbf{A}) = \begin{pmatrix} a_{11} \\ a_{21} \\ a_{12} \\ a_{22} \end{pmatrix} \quad (7.2)$$

If  $\mathbf{A}$  is symmetric,  $vech(\mathbf{A})$  denotes a column vector made of stacked columns of the lower triangular matrix of  $\mathbf{A}$ :

$$vech(\mathbf{A}) = \begin{pmatrix} a_{11} \\ a_{21} \\ a_{22} \end{pmatrix} \quad (7.3)$$

The important formula is

$$vec(\mathbf{ABC}) = (\mathbf{C}^T \otimes \mathbf{A})vec(B) \quad (7.4)$$

and can be verified directly.

In this note we used another simple identity, which can easily be proved directly: for compatible matrices  $\mathbf{A}$  and  $\mathbf{B}$  and an arbitrary column vector  $\mathbf{v}$

$$\mathbf{A}\mathbf{v}^T \otimes \mathbf{B} = \mathbf{v}^T \otimes (\mathbf{AB}) \quad (7.5)$$

## Hadamard product and diag

The Hadamard product of two matrices of the same size is an elementwise product,

$$\mathbf{A} \circ \mathbf{B} = \begin{pmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{pmatrix} \quad (7.6)$$

By  $diag(\mathbf{a})$  we denote a diagonal matrix with elements of vector  $\mathbf{a}$  on the diagonal:

$$diag \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} a_1 & 0 \\ 0 & a_2 \end{pmatrix} \quad (7.7)$$

$diag^{-1}(\mathbf{A})$  denotes a vector formed from diagonal elements of matrix  $\mathbf{A}$ :

$$diag^{-1} \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} a_{11} \\ a_{22} \end{pmatrix} \quad (7.8)$$

It is easily seen that

$$diag^{-1}(diag(\mathbf{a})) = \mathbf{a} \quad \text{but, for general } \mathbf{A}, \quad diag(diag^{-1}(\mathbf{A})) \neq \mathbf{A} \quad (7.9)$$

The important formula is

$$diag^{-1}(\mathbf{A}diag(\mathbf{b})\mathbf{C}) = (\mathbf{C}^T \circ \mathbf{A})\mathbf{b} \quad (7.10)$$

and can be verified directly.



# Bibliography

- [1] DOLEŽAL Z. ET AL.: *Cluster charge and centroid impact point reconstruction for the DEPFET detector*. DEPFET Internal Note, May 2007
- [2] KARIMÄKI V. ET AL.: *Sensor alignment by tracks*. Computing in High Energy and Nuclear Physics, 24-28 March 2003, La Jolla, California; [arXiv:physics/0306034](https://arxiv.org/abs/physics/0306034)
- [3] LUTZ G.: *Optimum track fitting in the presence of multiple scattering*. Nucl. Instr. Meth. A 273 (1988) 349-361
- [4] MINKA T.: *Old and new matrix algebra useful for statistics*. MIT Media Lab note (1997; revised 12/2000), <http://research.microsoft.com/~minka/papers/matrix/>
- [5] FRÜHWIRTH R. AND REGLER M.: *On the quantitative modelling of core and tails of multiple scattering by Gaussian mixtures*. Nucl. Instr. Meth. A 456 (2001) 369-389
- [6] FRÜHWIRTH R.: *Estimation of variances in a linear model applied to measurements of trajectories*. Nucl. Instr. Meth. A 243 (1986) 173-180
- [7] DOLEŽAL Z. ET AL.: *Geant4 - Depfet testbeam simulations*. Depfet Internal Note, May 2007.
- [8] ŽARNECKI A. F. AND NIEŽURAWSKI P.: *EUDET Telescope Geometry and Resolution Studies*. EUDET-Report-2007-01, <http://www.eudet.org/zms/content/e62/e145/e200/eudet-report-2007-01.pdf>
- [9] YAO W.-M. ET AL.: *The Review of Particle Physics*. Journal of Physics G: Nucl. Part. Phys. 33, 1 (2006) 262+
- [10] VELTHUIS, J.: *Testbeam analysis*. DEPFET collaboration meeting, Aachen, February 2007, [http://h11.mpg.de/twiki/pub/DepfetInternal/MeetingAachen0207/Aachen%2c\\_testbeam\\_status.ppt](http://h11.mpg.de/twiki/pub/DepfetInternal/MeetingAachen0207/Aachen%2c_testbeam_status.ppt)