A Large-scale Application of the Kalman Alignment Algorithm to the CMS Tracker

E. Widl, R. Frühwirth

Institute for High Energy Physics of the Austrian Academy of Sciences, Vienna, Austria

CHEP'07, Victoria, Canada, 2–7 September 2007

Introduction

We describe a method for global alignment with tracks that does not require solving a large system of linear equations.

- The method is iterative, based on the Kalman filter equations.
- The alignment parameters are updated immediately after a track is processed.
- The current knowledge about the alignment can be directly used to improve the tracking.
- The update is not restricted to the modules crossed by the track, but limited to modules with significant correlations to the ones in the current track.
- In order to keep track of the correlations some bookkeeping is required.
- The Kalman filter equations offer the possibility to use prior information abaout the alignment from mechanical or laser alignment, and it is easy to fix the position of reference modules. The method is also highly suitable for alignment relative to another detector.

The algorithm

- The observations $\boldsymbol{m}$ depend on the track parameters $\boldsymbol{x}_\mathrm{t}$ and the alignment parameters $\boldsymbol{d}_\mathrm{t}$ via the track model f:

$$\boldsymbol{m} = \boldsymbol{f}(\boldsymbol{x}_\mathrm{t}, \boldsymbol{d}_\mathrm{t}) + \varepsilon, \quad \mathsf{cov}(\varepsilon) = \boldsymbol{V}$$

  ε contains the effects of the observation error and of multiple scattering. Energy loss is considered as deterministic and is included in the track model. Its variance-covariance matrix $\boldsymbol{V}$ can be assumed to be known.

- For the purpose of the Kalman alignment algorithm, this track model is linearized:

$$\boldsymbol{m} = \boldsymbol{c} + \boldsymbol{A}\boldsymbol{d}_\mathrm{t} + \boldsymbol{B}\boldsymbol{x}_\mathrm{t} + \boldsymbol{\varepsilon}$$

$$\boldsymbol{A} = \partial\boldsymbol{m}/\partial\boldsymbol{d}_\mathrm{t}\big|_{\boldsymbol{d}_0}, \quad \boldsymbol{B} = \partial\boldsymbol{m}/\partial\boldsymbol{x}_\mathrm{t}\big|_{\boldsymbol{x}_0}$$

$$\boldsymbol{c} = f(\boldsymbol{x}_0, \boldsymbol{d}_0) - \boldsymbol{A}\boldsymbol{d}_0 - \boldsymbol{B}\boldsymbol{x}_0$$

- Expansion point $\boldsymbol{d}_0$: the nominal or the currently estimated module alignment.
- Expansion point $\boldsymbol{x}_0$: result of a preliminary track fit.
- The Kalman filter requires a prediction $\boldsymbol{x}$ of the track parameters, along with its variance-covariance matrix $\boldsymbol{C}$, that has to be stochastically independent of the observations in the track.
- The iterative update of the alignment parameters needs some starting values. Mechanical and laser alignment can be used for obtaining suitable starting values. Reference modules can be fixed by giving them very small initial errors.

Case 1: An independent prediction of the track parameters exists. The update equations for the alignment parameters read:

$$\widehat{\boldsymbol{d}} = \boldsymbol{d} + \boldsymbol{K}\left(\boldsymbol{m} - \boldsymbol{c} - \boldsymbol{A}\boldsymbol{d} - \boldsymbol{B}\boldsymbol{x}\right)$$

with the following gain matrix:

$$\boldsymbol{K} = \boldsymbol{D}\boldsymbol{A}^T \underbrace{\left(\boldsymbol{V} + \boldsymbol{A}\boldsymbol{D}\boldsymbol{A}^T + \boldsymbol{B}\boldsymbol{C}\boldsymbol{B}^T\right)^{-1}}_{\boldsymbol{G}}$$

Case 2: No independent prediction of the track parameters exists. In this case, the prediction $\boldsymbol{x}_0$ gets zero weight in order not to bias the estimation. This is accomplished by multiplying $\boldsymbol{C}$ by a scale factor α and letting α tend to infinity:

$$\boldsymbol{G} = \lim_{\alpha\longrightarrow\infty}\left(\boldsymbol{V} + \boldsymbol{A}\boldsymbol{D}\boldsymbol{A}^T + \alpha\boldsymbol{B}\boldsymbol{C}\boldsymbol{B}^T\right)^{-1} = \boldsymbol{V}_D^{-1} - \boldsymbol{V}_D^{-1}\boldsymbol{B}(\boldsymbol{B}^T\boldsymbol{V}_D^{-1}\boldsymbol{B})^{-1}\boldsymbol{B}^T\boldsymbol{V}_D^{-1}$$

with

$$\boldsymbol{V}_D = \boldsymbol{V} + \boldsymbol{A}\boldsymbol{D}\boldsymbol{A}^T$$

Because of $\boldsymbol{G}\boldsymbol{B} = \boldsymbol{0}$ the update equation of the alignment parameters can be simplified to

$$\widehat{\boldsymbol{d}} = \boldsymbol{d} + \boldsymbol{D}\boldsymbol{A}^T\boldsymbol{G}\left(\boldsymbol{m} - \boldsymbol{c} - \boldsymbol{A}\boldsymbol{d}\right)$$

The update of the covariance matrix can be calculated by linear error propagation:

$$\widehat{\boldsymbol{D}} = \left(\boldsymbol{I} - \boldsymbol{D}\boldsymbol{A}^T\boldsymbol{G}\boldsymbol{A}\right)\boldsymbol{D}\left(\boldsymbol{I} - \boldsymbol{A}^T\boldsymbol{G}\boldsymbol{A}\boldsymbol{D}\right) + \boldsymbol{D}\boldsymbol{A}^T\boldsymbol{G}\boldsymbol{V}\boldsymbol{G}\boldsymbol{A}\boldsymbol{D}$$

Implementation and computational complexity

An update of all alignment parameters is too slow for practical purposes. There are two alternatives:

1. Update only the modules in the current track, neglecting all correlations. This gives an unbiased estimate, but is suboptimal because of the missing correlations.
2. Update the modules having significant correlations with the modules in the current track. This method is nearly optimal, but it has to be guaranteed that $\widehat{\boldsymbol{D}}$ is positive definite all the time.

In order to keep track of the necessary updates, a list L_i is attached to each detector module i, containing the detector modules that have significant correlations with i.

1. Update the list L_i for every $i \in I$ (the set of modules crossed by the current track).
2. Form the list L of all detector modules that are correlated with the ones crossed by the current track: $L = \bigcup_{i\in I} L_i$. The size of L should be much smaller than N.
3. For all $j \in L$ compute: $(\boldsymbol{D}\boldsymbol{A}^T)_j = \sum_{i\in I}\boldsymbol{D}_{ji}\boldsymbol{A}_i^T$. Each block $\boldsymbol{D}_{ji}$ is of size $n_j \times n_i$, where $n_i = \dim(\boldsymbol{d}_i)$.
4. Compute: $\boldsymbol{A}\boldsymbol{D}\boldsymbol{A}^T = \sum_{i\in I}\boldsymbol{A}_i(\boldsymbol{D}\boldsymbol{A}^T)_i$.
5. Compute: $\boldsymbol{V}_D = \boldsymbol{V} + \boldsymbol{A}\boldsymbol{D}\boldsymbol{A}^T$ and $\boldsymbol{G}$. All matrices involved are of size $m \times m$, where $m = \dim(\boldsymbol{m})$.
6. Compute: $\boldsymbol{m'} = \boldsymbol{G}\left(\boldsymbol{m} - \boldsymbol{c} - \sum_{i\in I}\boldsymbol{A}_i\boldsymbol{d}_i\right)$.
7. For all $j \in L$ compute and store: $\widehat{\boldsymbol{d}}_j = \boldsymbol{d}_j + (\boldsymbol{D}\boldsymbol{A}^T)_j\boldsymbol{m'}$.

After each track, only the correlations between the modules in the list $L = \bigcup_{i\in I} L_i$ are updated:

$$\widehat{\boldsymbol{D}}_{jl} = \boldsymbol{D}_{jl} + (\boldsymbol{D}\boldsymbol{A}^T)_j(\boldsymbol{G}\boldsymbol{V}_D\boldsymbol{G} - 2\boldsymbol{G})[(\boldsymbol{D}\boldsymbol{A}^T)_l]^T, \forall j, l \in L$$

The computational complexity of the parameter update is of the order $|L|\cdot|I|$, and the computational complexity of the update of the covariance matrix is of the order $|L|^2$. Restricting the size of the lists L_i is therefore of crucial importance.

Restricting the number of updated modules

To determine which alignables have significant correlations and should therefore be included into the list L_i and become updated, the following procedure is used:

- Define a relation "$\sim$" between two different alignables i and j:

  $i \sim j \Longleftrightarrow i$ and j have been crossed by the same track.

- Define the metrical distance $d(i,j)$ on the basis of this relation:

  If $i \sim i_1 \sim i_2 \sim \cdots \sim i_n \sim j$ is the shortest chain connecting i to j, the distance is $d(i,j) = n+1$. In particular, if $i \sim j$, then $d(i,j) = 1$. See Figure 1.

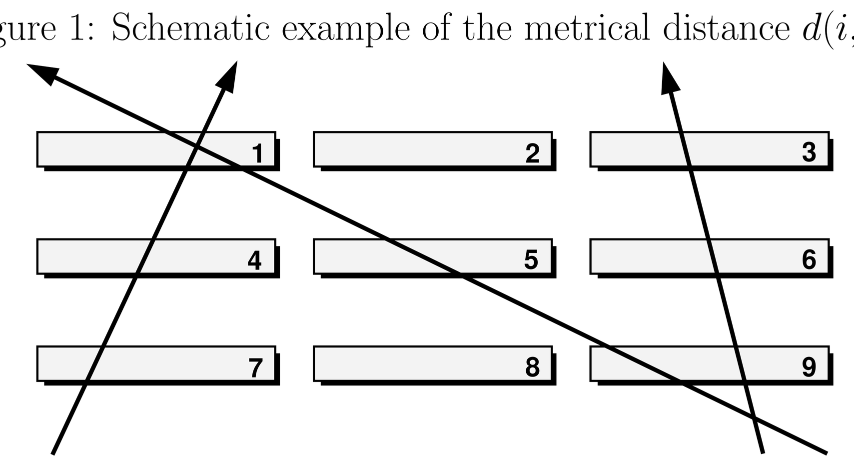
- Include j in the list L_i only if $d(i,j) \le d_\mathrm{max}$.
- Inflate the variance-covariance matrix $\boldsymbol{V}$ to decouple metrically more distant alignables:

$$\boldsymbol{V} \longrightarrow \boldsymbol{V} + \Delta V \cdot \boldsymbol{I}$$

Figures 2–4 demonstrate the effect of an increasing value of ΔV on the correlations $R_{ij} = \sigma_{ij}/\sigma_i \cdot \sigma_j$:

- A small-scale setup of approximately 500 strip-modules was aligned against a fixed reference system. Only the local coordinate x (perpendicular to the strips) was considered.
- After processing 10,000 tracks, the absolute values of the correlations between all alignables were histogrammed in dependence on their metrical distance (shown here only up to $d = 6$). The effect of decreasing correlations with increasing values of d for larger ΔV can be clearly seen.
- In Figure 5, the corresponding results are shown. They are almost identical for all runs in which the correlations were taken into account, also for the case of truncation at $d_\mathrm{max} = 4$. Only when neglecting the correlations entirely, the achieved precision suffers noticeably.

Figure 1: Schematic example of the metrical distance $d(i,j)$.



$d(1,9)=1, \quad d(5,6)=2, \quad d(3,7)=3$

Figure 2: $\sqrt{\Delta V} = 0\,\mu\mathrm{m}$

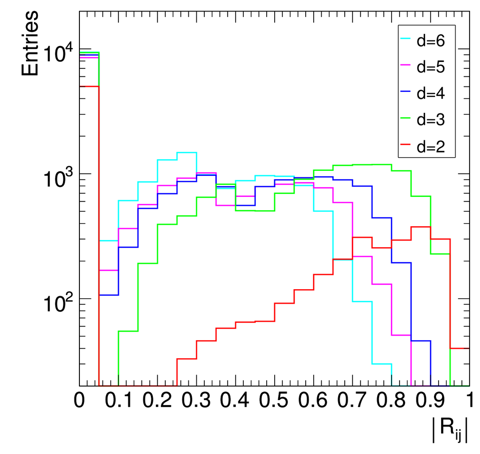


Figure 3: $\sqrt{\Delta V} = 50\,\mu\mathrm{m}$

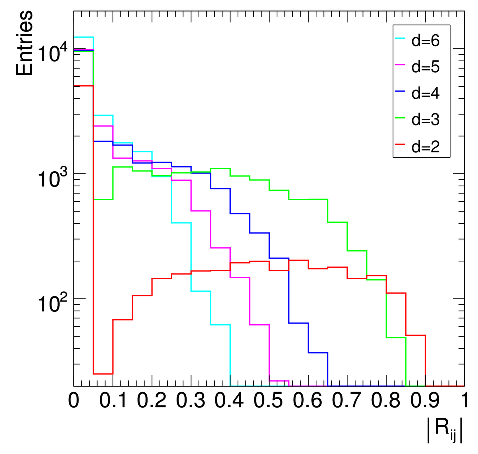


Figure 4: $\sqrt{\Delta V} = 100\,\mu\mathrm{m}$

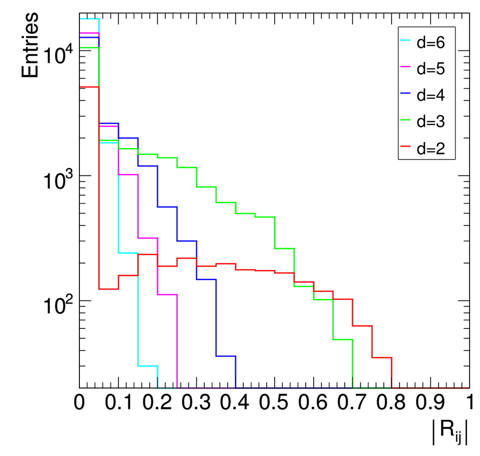
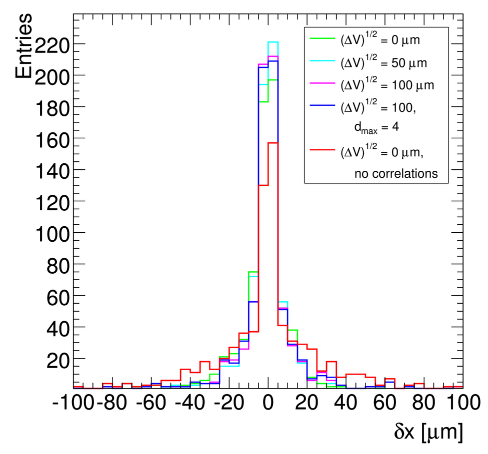


Figure 5: Final resolution.



A large-scale application

The algorithm has been implemented within the CMS software framework. To demonstrate its performance it was applied to a large-scale setup, comprising about a third of all modules of the CMS tracker. A standard PC was used for the calculations (2.2 GHz CPU, AMD Athlon 64 Processor 3500+, 1 GByte RAM).

- All modules in the barrel region of the Tracker, i.e. the Pixel Barrel (TPB), the Inner Barrel (TIB) and the Outer Barrel (TOB), were misaligned by drawing from Gaussians with standard deviations according to the values given in Table 1.
- A sample of 70.000 fully simulated tracks in the region $|\eta| \le 1.0$, stemming from $Z \rightarrow \mu^+\mu^-$-decays, was used for alignment.
- Values of $d_\mathrm{max} = 4$ and $\sqrt{\Delta V} = 100\ \mu m$ were used.
- Each of the subsystems was aligned, using the other systems as an external reference.
- The results for the TPB were applied for the alignment of the TIB and the TOB.
- The results for the TIB were applied for the alignment of the TOB.
- A total of 6125 modules was aligned: 587 TPB, 1654 TIB, 3884 TOB.
- The resulting placement uncertainties can be seen in Table 2.
- The computation times can be seen in Table 3. It shows the total time T_tot, the time spent in refitting the tracks (T_fit), updating the metrics (T_met), computing all derivatives and the gain matrix (T_com), retrieving the parameters and their covariance matrix before the update and storing them back afterwards ($T_\mathrm{i/o}$), and finally the time needed for the algorithmic update itself (T_alg).

Table 1: Initial placement uncertainties.

	x [μm]	y [μm]	z [μm]	α [mrad]	β [mrad]	γ [mrad]
TPB	100	100	100	0.5	0.5	0.5
TIB	200	200	200	2.0	2.0	2.0
TOB	100	100	100	1.0	1.0	1.0

Table 2: Remaining placement uncertainties.

	Δx [μm]	Δy [μm]	Δz [μm]	$\Delta\alpha$ [mrad]	$\Delta\beta$ [mrad]	$\Delta\gamma$ [mrad]
TPB	18.6	26.6	27.9	–	–	0.24
TIB	30.8	109.9 *	149.7 *	–	–	0.56
TOB	23.8	77.9 *	–	–	–	0.30

* Double-sided modules only.

Table 3: Computing times.

	T_tot [s]	T_fit [s]	T_met [s]	T_com [s]	$T_\mathrm{i/o}$ [s]	T_alg [s]
TPB	2837	250	10	350	966	1155
TIB	6314	256	26	785	2353	2670
TOB	5305	260	62	581	2355	1891