Available on CMS information server

**CMS CR 2007/055**

# CMS Conference Report

**9 October 2007**

# 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*

**Abstract**

The Kalman alignment algorithm has been specifically developed to cope with the demands that arise from the specifications of the CMS Tracker. The algorithmic concept is based on the Kalman filter formalism and is designed to avoid the inversion of large matrices. Most notably, the algorithm strikes a balance between conventional global and local track-based alignment algorithms, by restricting the computation of alignment parameters not only to alignable objects hit by the same track, but also to all other alignable objects that are significantly correlated. Nevertheless, this feature also comes with various trade-offs: Mechanisms are needed that affect which alignable objects are significantly correlated and keep track of these correlations. Due to the large amount of alignable objects involved at each update (at least compared to local alignment algorithms), the time spent for retrieving and writing alignment parameters as well as the required user memory becomes a significant factor. The large-scale test presented here applies the Kalman alignment algorithm to the (misaligned) CMS Tracker barrel, and demonstrates the feasability of the algorithm in a realistic scenario. It is shown that both the computation time and the amount of required user memory are within reasonable bounds, given the available computing resources, and that the obtained results are satisfactory.

# 1 Introduction

We describe a method for global alignment with tracks that has been specifically developed [1, 2] to cope with the demands that arise from the specifications of the CMS Tracker [3]. Due to its design it does not require solving a large system of linear equations. The method is iterative, based on the Kalman filter equations [4, 5, 6], such that the alignment parameters for the alignable objects (refered to as alignables) are updated each time after a track is processed. Thus, the current knowledge about the alignment can be directly used to improve the tracking. The update is not restricted to the alignables crossed by the track, but limited to alignables 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 alignables in order to define a reference. The method is also highly suitable for alignment relative to another detector.

# 2 The algorithm

The algorithm is based on a track model $f$ that relates the observations $\boldsymbol{m}$ not only to the track parameters $\boldsymbol{x}_{\mathrm{t}}$ but also to the alignment parameters $\boldsymbol{d}_{\mathrm{t}}$:

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

The stochastic vector $\boldsymbol{\varepsilon}$ contains the observation error and the effects of multiple scattering, whereas energy loss is considered to be deterministic. Its variance-covariance matrix $\boldsymbol{V}$ can be assumed to be known.
For the purpose of the Kalman filter formalism, this track model is linearized around an expansion point $(\boldsymbol{d}_0, \boldsymbol{x}_0)$:

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

where

$$\boldsymbol{A} = \partial \boldsymbol{m}/\partial \boldsymbol{d}_{\mathrm{t}}\big|_{\boldsymbol{d}_0}, \quad \boldsymbol{B} = \partial \boldsymbol{m}/\partial \boldsymbol{x}_{\mathrm{t}}\big|_{\boldsymbol{x}_0},$$
$$\boldsymbol{c} = f(\boldsymbol{x}_0, \boldsymbol{d}_0) - \boldsymbol{A}\boldsymbol{d}_0 - \boldsymbol{B}\boldsymbol{x}_0.$$

The expansion point $\boldsymbol{d}_0$ is either the nominal or the currently estimated alignment, whereas $\boldsymbol{x}_0$ is the 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 is stochastically independent of the observations in the track. In case such an independent prediction of the track parameters exists, the update equations for the alignment parameters read:

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

with the following gain matrix:

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

If no independent prediction of the track parameters exists, the prediction $\boldsymbol{x}_0$ has to be assigned zero weight in order not to bias the estimation. This is accomplished by multiplying $\boldsymbol{C}$ by a scale factor $\alpha$ and letting $\alpha$ tend to infinity:

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

with

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

Here the Sherman-Morrison inversion formula

$$(\boldsymbol{X} + \boldsymbol{H}\boldsymbol{Y}\boldsymbol{H}^T)^{-1} = \boldsymbol{X}^{-1} - \boldsymbol{X}^{-1}\boldsymbol{H}(\boldsymbol{Y}^{-1} + \boldsymbol{H}^T\boldsymbol{X}^{-1}\boldsymbol{H})^{-1}\boldsymbol{H}^T\boldsymbol{X}^{-1}$$

has been used (see e.g. [7]). Because of $\boldsymbol{G}\boldsymbol{B} = \boldsymbol{0}$ the update equation of the alignment parameters can be simplified to

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

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

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

The iterative update of the alignment parameters needs some starting values. For this purpose mechanical and laser alignment can be used to obtain suitable starting values. Alignables defining a reference can be fixed by giving them very small initial errors.

# 3 Implementation and computational complexity

The total number of detector units is denoted by $N$. The current track crosses a certain number of detector units, denoted by $k$. If each of them gives a two-dimensional measurement, the dimension $m = 2\,k$ of the observation vector $\boldsymbol{m}$ is small for high-energy tracks, usually not larger than 30. The matrix $\boldsymbol{B}$ is of size $m \times 5$ and is therefore small. The matrix $\boldsymbol{A}$ is a row of $N$ blocks $\boldsymbol{A}_i$ of size $m \times n$, where $n$ is the number of alignment parameters per detector unit. For each track, only $k$ out of these $N$ blocks are different from zero. The set of detector units crossed by the current track is denoted by $I = \{i_1, \ldots, i_k\}$. Then the matrix $\boldsymbol{A}$ has the following form:

$$\boldsymbol{A} = \left(\boldsymbol{0} \ldots \boldsymbol{0}\ \boldsymbol{A}_{i_1}\ \boldsymbol{0} \ldots \boldsymbol{0}\ \boldsymbol{A}_{i_2}\ \boldsymbol{0} \ldots\ldots \boldsymbol{0}\ \boldsymbol{A}_{i_k}\ \boldsymbol{0} \ldots \boldsymbol{0}\right).$$

The only large matrix in the parameter update is the product $\boldsymbol{D}\boldsymbol{A}^T$. It is a column of $N$ blocks each of which has size $n \times m$. However, only those blocks need to be computed that correspond to the detector units that have significant correlation with the ones in the current track. In order to keep track of the necessary updates, a list $L_i$ is attached to each detector unit $i$, containing the detector units that have significant correlations with $i$. This list may contain only $i$ itself in the beginning and grows as more tracks are processed. This leads to the following procedure for computing the updated alignment parameters:

1. Update the list $L_i$ for every $i \in I$.
2. Form the list $L$ of all alignables 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 the total number of alignables.
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 alignables 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 $|L|^2$.
Especially when aligning at the level of modules, i.e. when dealing with an amount of several thousand alignables, an update of all alignment parameters is too slow for practical purposes. Also, the amount of user memory (RAM) needed to store the correlations exceeds the resources typically available on a PC. There are, however, two alternatives: Either only the alignables in the current track are updated, neglecting all correlations, or all the alignables are updated that have significant correlations with the ones in the current track. While the first approach gives an unbiased estimate, it is suboptimal because of the missing correlations. The latter method is nearly optimal, but it has to be guaranteed that $\widehat{\boldsymbol{D}}$ is positive definite all the time.

# 4 Restricting the number of updated modules

The crucial point is to determine which alignables have significant correlations and should therefore be included into the list $L_i$ and get updated. To do so, a relation "$\sim$" between two different alignables $i$ and $j$ is defined:

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

Using this relation, a metrical distance $d(i, j)$ can be introduced:

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 $d(i, j) = 1$.

For an example see Figure 1, where e.g. $d(1, 9) = 1$, $d(5, 6) = 2$ and $d(3, 7) = 3$.

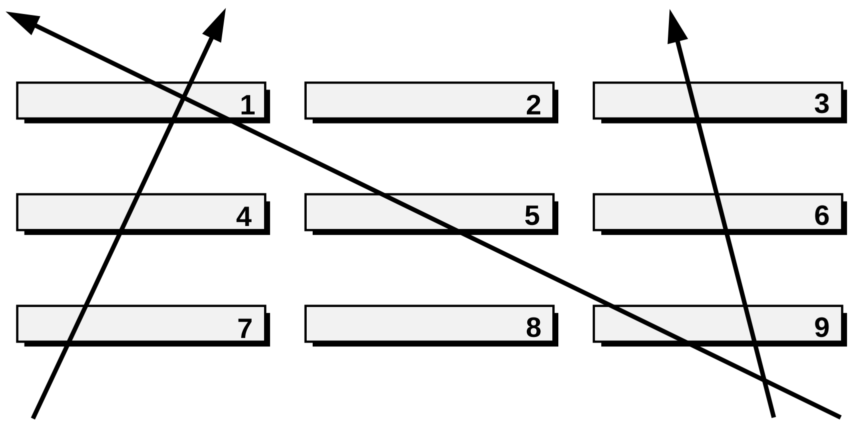


Figure 1: Schematic example of the metrical distance $d(i, j)$.

The definition of this metrics allows to restrict the alignables contained in the lists $L_i$ in a coherent way:

1. Include a alignable $j$ in the list $L_i$ only if $d(i, j) \leq d_{\max}$.
2. Then 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 $\Delta V$ on the correlations $R_{ij} = \sigma_{ij}/\sigma_i \cdot \sigma_j$. For this purpose, 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 $\Delta V$ 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 $d_{\max} = 4$. Only when neglecting the correlations entirely, the achieved precision suffers noticeably.

# 5 A large-scale application

The algorithm has been implemented within the CMS software framework [8]. 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 $|\eta| \leq 1.0$, stemming from $Z \rightarrow \mu^+\mu^-$–decays, was used for alignment. The metrical threshold was set to $d_{\max}$ = 4, the variance-covariance matrix was inflated using a value of $\sqrt{\Delta V}$ = 100 $\mu$m.

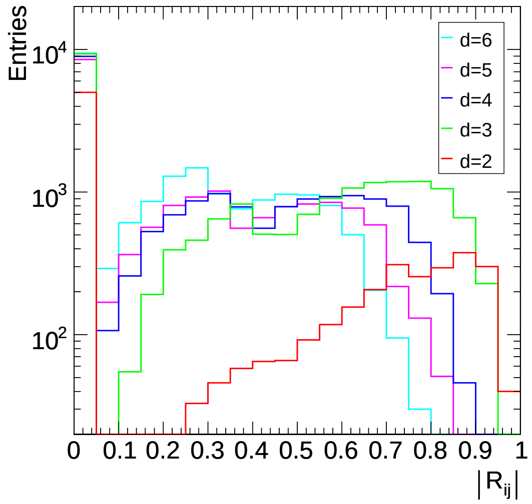


Figure 2: Correlation coefficient for different metrical distances and $\sqrt{\Delta V} = 0\,\mu$m

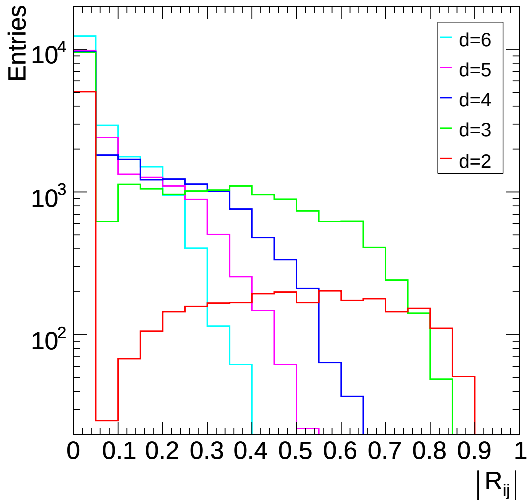


Figure 3: Correlation coefficient for different metrical distances and $\sqrt{\Delta V} = 50\,\mu$m

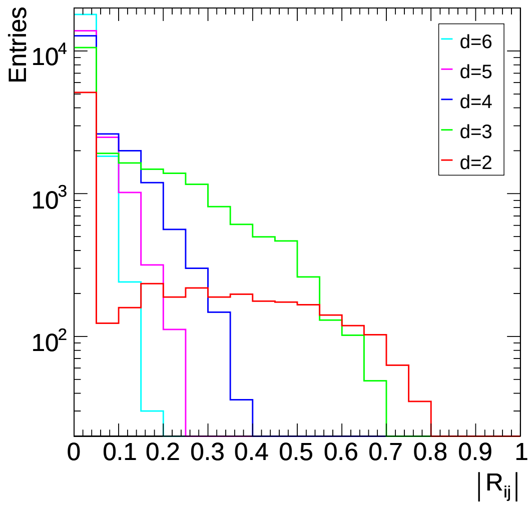


Figure 4: Correlation coefficient for different metrical distances and $\sqrt{\Delta V} = 100\,\mu$m

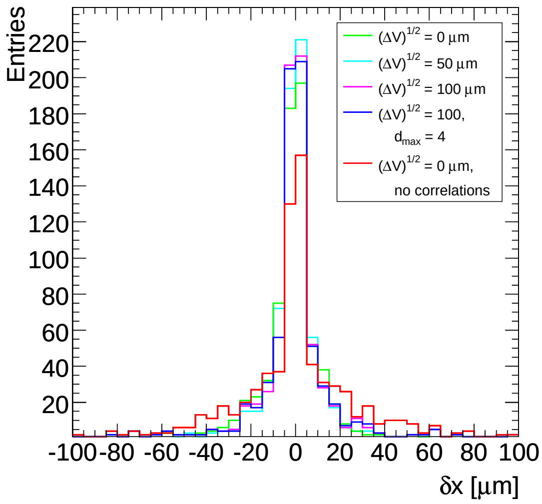


Figure 5: Final resolution for various combinations of $\sqrt{\Delta V}$ and $d_{\mathrm{max}}$.

First of all, the modules of the TPB were aligned using tracks reconstructed [9] in the TIB and TOB as external reference. While the measurements from the TIB modules yield a higher positional precision, the measurements from the TOB modules are important to obtain reasonable momentum estimates due to the larger lever arm. The resulting alignment parameters were then applied for all further calculations. After that, the modules from the TIB were aligned, while the TPB and the TOB were used to calculate external predictions for the track parameters. The already aligned modules in the TPB provide a good external refrence while the measurments from modules from the TOB were again mostly important to obatin proper momentum estimates. Finally, the alignment constants for the modules from the TOB were computed, using the aligned TPB and the aligned TIB as external reference. A total of 6125 modules was aligned (587 TPB, 1654 TIB, 3884 TOB)

The splitting of the computation into three steps has several advanatges. First of all, due to the reduced amount of alignables the computation time is kept at a reasonable level (see below). Also the required user memory does not exceed the given resources. Finally, it allows to introduce a reference system without the need to include the alignables defining it into the update, which of course speeds up the algorithm.

The resulting placement uncertainties can be seen in Table 2. The computation times can be seen in Table 3. The latter shows the total time $T_{\mathrm{tot}}$, the time spent in refitting the tracks ($T_{\mathrm{fit}}$), updating the metrics ($T_{\mathrm{met}}$), computing all derivatives and the gain matrix ($T_{\mathrm{com}}$), 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 ($T_{\mathrm{alg}}$).

Table 1: Initial placement uncertainties.

| | $x$ [$\mu$m] | $y$ [$\mu$m] | $z$ [$\mu$m] | $\alpha$ [mrad] | $\beta$ [mrad] | $\gamma$ [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.

| | $\Delta x$ [$\mu$m] | $\Delta y$ [$\mu$m] | $\Delta z$ [$\mu$m] | $\Delta\alpha$ [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.

| | $T_{\rm tot}$ [s] | $T_{\rm fit}$ [s] | $T_{\rm met}$ [s] | $T_{\rm com}$ [s] | $T_{\rm i/o}$ [s] | $T_{\rm alg}$ [s] |
|---|---|---|---|---|---|---|
| TPB | 2837 | 250 | 10 | 350 | 966 | 1155 |
| TIB | 6314 | 256 | 26 | 785 | 2353 | 2670 |
| TOB | 5305 | 260 | 62 | 581 | 2355 | 1891 |

# 6 Summary

A general algorithm for the track-based global alignment of a complex detector system has been presented. The algorithm is derived from the Kalman filter and is designed to avoid the inversion of large matrices. The alignables included at each update can be restricted, such that the elapsed time and required amount of user memory are kept at a reasonable level. Future work will concentrate on aligning the tracker end-caps relative to the barrel and on finding track samples that are suitable for this purpose.

# References


[1] E. Widl, R. Frühwirth, W. Adam 2006 *A Kalman Filter for Track-based Alignment* CMS NOTE–2006/022

[2] R. Frühwirth, E. Widl 2007 Track-based alignment using a Kalman filter technique *Proc. of the first LHC Detector Alignment Workshop* ed S. Blusk et al. CERN–2007–004

[3] The CMS Collaboration 1998 *The Tracker Project Technical Design Report* CERN/LHCC **98-6**
The CMS Collaboration 2000 *Addendum to the Tracker TDR* CERN/LHCC **2000–016**

[4] D.E. Catlin 1989 *Estimation, control, and the discrete Kalman filter* (New York: Springer)

[5] R. Frühwirth 1984 *Nucl. Instrum. and Meth.* **A 262** 444

[6] R. Frühwirth, T. Todorov, M. Winkler 2003 *J. Phys. G: Nucl. Part. Phys.* **29** 561

[7] R. Zurmühl, S. Falk 1984 *Matrizen und ihre Anwendungen* 5th edition (Berlin–Heidelberg: Springer)

[8] The CMSSW Application Framework
`https://twiki.cern.ch/twiki/bin/view/CMS/SWGuideFrameWork`

[9] W. Adam et al 2006 *Track Reconstruction in the CMS tracker* CMS NOTE–2006/041