Reconstruction software of the silicon tracker of DAMPE mission

A Tykhonov, V Gallo, X Wu and S Zimmer
DPNC, University of Geneva, 24, Quai Ernest-Ansermet, CH-1211 Genève 4, Switzerland
E-mail: andrii.tykhonov@cern.ch

Abstract. DAMPE is a satellite-borne experiment aimed to probe astroparticle physics in the GeV-TeV energy range. The Silicon tracker (STK) is one of the key components of DAMPE, which allows the reconstruction of trajectories (tracks) of detected particles. The non-negligible amount of material in the tracker poses a challenge to its reconstruction and alignment. In this paper we describe methods to address this challenge. We present the track reconstruction algorithm and give insight into the alignment algorithm. We also present our CAD-to-GDML converter, an in-house tool for implementing detector geometry in the software from the CAD drawings of the detector.

1. Introduction
DAMPE (DArk Matter Particle Explorer) is a powerful space telescope launched in December 2015, able to detect cosmic particles in a wide range of energy, 5 GeV to 10 TeV for electrons and photons and up to 100 TeV for protons and heavy ions [1]. It consists of a double layer of plastic scintillator strips detector (PSD) that serves as anti-coincidence detector, followed by the silicon-tungsten tracker-converter (STK), which is made of 6 tracking double layers interleaved with 3 layers of tungsten converters. The STK is followed by an imaging calorimeter of about 31 radiation lengths in thickness, made up of 14 layers of Bismuth Germanium Oxide (BGO) bars in a hodoscopic arrangement. A thick layer of boron-plated scintillator is added to serve as proton veto by detecting delayed neutrons originating from hadronic interactions at high energies in the detector. The total thickness of the BGO calorimeter and the STK makes DAMPE the deepest calorimeter ever used in space.

STK is a crucial component of the detector (figure 1–left), able to determine the direction of detected particles and to trace the origin of incoming gamma rays [2]. It consists of 6 tracking double layers of single-sided silicon strip detectors, interleaved with three layers of tungsten plates, placed after the 1st, 2nd and 3rd layer. The latter promote pair conversion of incoming gamma rays into an electron-positron pair which can be used to reconstruct the direction of the incoming photon. Each tungsten layer is 1 mm thick, which makes a total of about one radiation length of material in the STK. The 12 silicon layers are assembled on 7 supporting trays, as shown in figure 1–right, providing 6 independently measured points for x and 6 for y. In each layer the x and y coordinate of the particle are measured separately. Each silicon layer consists of 16 silicon modules, named ladders. Each ladder is formed by 4 silicon sensors with a strip to strip micro wire-bond connection [3]. There are in total of 768 silicon sensors in the STK.
This proceeding covers the reconstruction software of the STK, including track finding algorithm (section 2), the alignment algorithm (section 3) and a CAD-GDML geometry converter tool-kit (section 4).

![Figure 1. Layout of the DAMPE detector (left) and schematic view of the STK (right).](image)

### 2. STK Reconstruction

The tracker reconstruction consists of two main tasks, hit reconstruction and track finding. The analog read-out of silicon strips in the STK allows the improvement of position resolution with respect to digital read out, since each particle passing through the silicon sensors usually yields signal in more than one strip, with the signal being proportional to the amount of ionization produced in the silicon by the particle. Therefore a clustering of signal strips should be performed, where a cluster corresponds to a particle hit in the silicon. Once the hits are formed in all layers of the tracker, another algorithm combines hits which supposedly belong to the same particle into a track. Because of particle interaction in the detector, there are often more than one track in the tracker, so track finding becomes a pattern recognition problem.

#### 2.1. Hit Reconstruction

The signal of each ladder is represented as an array of 384 channels, each channel corresponding to 12 bit adc value. The clustering algorithm scans the array to find a local signal maximum with a signal-to-noise ratio $S/N > 4$; the seed. Then a hit (cluster) is formed by collecting all the channels with $S/N > 1.5$ neighboring the seed. If the signals in two neighboring channels differ from each other by more than five times with respect to their noise values, the clusters exhibiting multiple peaks are split. The hit coordinate in the ladder is calculated as a center of gravity of the cluster strips weighted by the adc signals. The $x/y$ coordinate of a hit in the global reference frame of the detector is obtained from the hit position in the ladder and the ladder position and orientation in the nominal design.

#### 2.2. Track Reconstruction

Track finding can be grouped into two major steps: seeding and propagation of the track. Track seeds are formed either with the blind-seed algorithm where any combination of three consecutive hits forming a line is considered as a seed, or with a calorimeter-seed approach. In the latter case, the energy shower is first reconstructed in the calorimeter and its direction is projected onto the first layer of the STK. Then the hit closest to the projection is picked up
and the track seed is formed from this hit and shower direction. Once a seed is created, a track is propagated in the STK using a custom implementation of the Kalman filter [4]. The track quality is evaluated based on the number of hits and the $\chi^2$ of the Kalman fit [4] and candidates are rejected if their quality is below a pre-determined threshold. In case no track candidate passes the quality selection, seeding and filtering are repeated, initially with hits in the first layer but if no candidate survives the selection, this procedure is repeated for the 2nd and 3rd layer, respectively. If a good track is found, its first point is removed from the list of ‘seeding’ points. The seeding-filtering is repeated until all seeding points are exhausted. Then the same procedure is done with the three furthermost layers of STK, with the opposite direction of track filtering.

To ensure the photon detection capability of the STK, the presence of dense passive material (in the form of thick tungsten plates) is required, which introduces a major difficulty in the track reconstruction as described above. Passive material in the STK initiates particle pre-showering already at the tracker level. Combined with potential back-splash of particles from the shower in the calorimeter this may yield a high hit multiplicity of up to $O(100)$ hits per layer at high energy, giving rise to very high multiplicity of track seeds in the seeding-filtering stage. Moreover, since $x$ and $y$ projections are measured independently and their combination is ambiguous, the number of possible seeds increases quadratically. To mitigate this problem, the track reconstruction in the STK is done first in $x$ and $y$ projections independently. If the $x$ projection is considered, the complementary $y$ direction from the calorimeter shower is used for evaluating the co-variance, scattering and transport matrices of the Kalman filter. Once the two-dimensional tracks are formed, they are combined in $xy$, and refitted again with the Kalman filter. It has been seen that in case of DAMPE this approach yields more than $10\%$ higher efficiency of photon reconstruction compared to the approach where tracks are reconstructed in $xy$ from the beginning. In the latter case, in order to keep computational time the same as for the baseline approach, the number of seeding-filtering steps per event had to be restricted below ten (if not limited, the computation time in this case would be about two orders of magnitude higher then in the baseline approach).

3. STK Alignment

The construction precision of the mechanical assembly of STK is $O(100) \mu m$. Given the expected position resolution of silicon sensors of $50 \mu m$, a precise alignment of the instrument is required. Aligned positions of hits in the STK are expressed through their ideal positions (obtained from the GDML model, see section 4) with the following conversion

$$x_a = x + \Delta_x - y \cdot \theta_z \quad (1)$$
$$y_a = y + \Delta_y + x \cdot \theta_z \quad (2)$$
$$z_a = z + \Delta_z + x \cdot \theta_y + y \cdot \theta_x \quad (3)$$

where $\Delta_x$, $\Delta_y$, $\Delta_z$, $\theta_x$, $\theta_y$, $\theta_z$ are the alignment parameters for shifts and rotations of each silicon sensor, respectively. There are a total of 3840 alignment parameters, for 768 sensors. The baseline alignment in this paper is performed using a sample of one day of orbit data, and validated with another sample of 5-days, not overlapping with the first one. Both samples were taken in January 2016. The re-alignment is performed using the sample of one day of data collected in May 2016.
3.1. Alignment Algorithm

Alignment is performed through the minimization of the global $\chi^2$

$$\chi^2 = \sum_{t \in \{\text{tracks}\}} \left( \sum_{p \in \{\text{points}\}} \frac{(x_{t,p}^{fit} - x_{t,p}^{hit})^2}{N_x^{tracks,s}} + \sum_{p \in \{\text{points}\}} \frac{(y_{t,p}^{fit} - y_{t,p}^{hit})^2}{N_y^{tracks,s}} \right)$$

(4)

where $x^{hit}$ and $x^{fit}$ are measured and fitted (from a linear fit) coordinates of the track in the $p$-th point; $N_x^{(y)^{tracks,s}}$ is the number of tracks crossing the $s$-th silicon sensor – this factor is introduced in order to take into account the non-uniformity of track statistics for different silicon sensors; Events used for the alignment are required to have exactly one track with $xy$ hits in all six planes (no holes). For the alignment, reconstructed tracks are refitted with a straight line, in $xz$ and $yz$ projections independently. Tracks with bad $\chi^2$ of the linear fit (> $6 \, mm^2$ – far from the bulk of the $\chi^2$-distribution) are removed from the sample.

Figure 2. The deviation of the residual mean (DRM) as a function the of alignment iteration. DRM is defined as a root mean square of the distribution of 768 mean track-hit residuals of all silicon sensors [5]. Two alignment methods are shown: left – fixed-$\chi^2$ approach (we also call it the standard one), middle and right – variable-$\chi^2$ approach (baseline algorithm). The re-alignment is done using the sample of one day of data collected in May 2016 given the alignment algorithm is initialized with the alignment parameters obtained in January 2016.

The minimization of $\chi^2$ is performed in iterative procedure using the derivatives (gradient) of $\chi^2$ with respect to alignment parameters. At each iteration, a step is performed in the phase space of alignment parameters in the direction opposite to the gradient. Then at each iteration derivatives are recalculated and a step is performed again. The alignment precision is limited in part by the internal resolution of the silicon sensors, but more importantly due to the presence of multiple scattering (see section 2). To address this, we apply an even tighter selection criteria to tracks. That is, only tracks with track-hit residue values below some threshold are allowed. We tested two different threshold values of 300 $\mu$m and 100 $\mu$m, respectively and picked 300 $\mu$m for our analysis. Note that while the precise threshold choice does not adversely affect the performance of the algorithm it may introduce a selection bias in the alignment data sample, especially if used on a fixed sample of tracks. To combat this bias, we allow the $\chi^2$ track sample to change from one iteration to another, in other words residue-based track quality selection
is applied at every iteration of alignment, introducing some new tracks in the sample and (or) removing some old ones. Hereinafter we call this approach the \textit{variable-}\(\chi^2\) method and refer to it as a baseline algorithm. The approach with no residue-based cut applied we call the \textit{fixed-}\(\chi^2\) method, the \textit{standard} algorithm. As seen from figure 2 the \textit{variable-}\(\chi^2\) method outperforms the standard one.

![Figure 3](image1.png) **Figure 3.** The \(\chi^2\) and step size of the alignment algorithm as functions of iteration of the algorithm: \textit{left} – initial alignment starting from non-aligned model; \textit{right} – later re-alignment. Both quantities are normalized to their corresponding values at the first iteration.

![Figure 4](image2.png) **Figure 4.** Absolute value of gradient of normalized \(\chi^2\) with respect to alignment parameters: \textit{left} – initial alignment starting from non-aligned model; \textit{right} – later re-alignment.

In the \textit{variable-}\(\chi^2\) method, \(\chi^2\) values can not be used to control step size of the algorithm. Note that a higher number of tracks in the sample with respect to prior iterations may increase the resulting \(\chi^2\) value (see figure 3–\textit{right}). Instead we require the inner product of the gradient between two iterations to be larger than some threshold \(\epsilon\) (0.5 in our case)

\[
\left\langle \frac{\partial(\chi^2)}{\partial(\Delta x, \ldots \theta_z)_{s \in \{\text{sensors}\}}} \right|_{\text{this}} \cdot \frac{\partial(\chi^2)}{\partial(\Delta x, \ldots \theta_z)_{s \in \{\text{sensors}\}}} \right|_{\text{next}} \right\rangle > \epsilon
\]  

(6)

When the inner product is less then \(\epsilon\) the step size is decreased. From figures 3 and 4 it is seen that the total derivative changes substantially and start converging at \(O(1000)\) iterations while \(\chi^2\) changes for about 1% after \(O(100)\) iterations. While around \(O(1000)\) iterations are needed for the very first alignment, \(O(100)\) iterations are enough to re-align the geometry on orbit (figure 4–\textit{right}). In the current offline processing of the orbit data, alignment is performed every two weeks, which is enough to keep position resolution of the instrument stable. A data corresponding to two days on orbit is enough to re-align the STK.

### 3.2. Performance of the Alignment Algorithm

The position resolution of STK after the alignment is used as a measure of performance of the algorithm. To estimate position resolution, a set of tracks with a \textit{validation} selection criterion
Figure 5. Residual distribution of projected minus measured hit position in the track for $x$ layers of the STK, where tracks are selected with the validation criterion; tracks with all inclinations are taken, after the alignment; the projected position is obtained from a 5-point linear fit, excluding the the plane itself where residual is calculated. Statistics corresponds to five subsequent days of orbit data. Distributions are fitted by a double Gaussian function [5].

Figure 6. RMS of the double Gaussian fit of track-hit residual distributions as functions of track inclination. Each curve corresponds to one of the $x/y$ layers of the STK (layers 1–4), excluding the first and last layers (outer layers, 0 and 5) Results are shown for the ideal model obtained from the simulation (left), and for the real data with aligned geometry – using the standard approach (middle) and the baseline variable-$\chi^2$ approach (right).

was used, and residuals were evaluated defined as difference between projected and measured coordinate of track in the $i$-th plane, where the projection is obtained by using the remaining five points of the track not including the point in the $i$-th plane (figure 5). For this validation selection we require the residuals in each point of the 5-point fit to be within 40 microns.
This helps to eliminate events which suffer most from multiple scattering. To ensure that the threshold in this selection does not bias the final result, we have also checked the same selection with a much lower threshold value, 10 microns. The resulting variation in the estimated position resolution is below 10%.

We show the residuals in histograms for different tracker layers (figure 5) and in histograms of equidistant angular bins (figure 6) and find that the variable-$\chi^2$ approach yields better results for the position resolution, compared to the standard one. Note that the improvement in position resolution is the greatest for inclined tracks for which we reach an improvement of up to 15% (figure 7–right).

Instead of using the deviation of the residual mean (DRM) or the $\chi^2$-value itself, we find that the absolute value of the $\chi^2$ gradient provides a better figure of merit to judge the performance of the alignment algorithms. Note that the former two quantities show convergence after $0(10)$ - $O(100)$ iterations (figures 2, 3). The latter may require up to $O(1000)$ iterations (figure 4), especially for inclined tracks (figure 7).

4. CAD-to-GDML Converter

One of the distinctive features of the DAMPE reconstruction and simulation software is the CAD-to-GDML converter tool-kit [6]. GDML (Geometry Description Markup Language) is a geometry description format [7] used in major particle physics simulation software packages, like GEANT4 [8]. Having a tool which converts CAD drawings into the physics model is very useful for a few reasons. First, it simplifies greatly the detector R&D allowing the immediate testing in the physics simulation of any modification of the geometry. Next, it minimizes a possible mismatches between real geometry of the detector and its physics model. Last but not least, it allows the more precise physics simulation of the detector, including all small peculiarities of the detector geometry, which is particularly important for example in the application of multivariate reconstruction algorithms.

The Converter tool-kit is implemented as a python module which allows the creation of a GDML geometry from a CAD drawing, without GEANT4 [8], XERCES [9] or any other software used in conventional approaches [10]. It takes care of material definitions (currently a basic list of materials is implemented) and the relative placement of different parts of a geometry in a detector model. The conversion is done in two steps. First, each part of the geometry corresponding to one single material is saved as an stl file, which can be done in most of CAD developer environments [11]. Second, the resulting set of stl files is passed to the converter which creates a GDML model. A GEANT4 material description for each volume is assigned during the conversion. The final geometry is assembled automatically, without having to convert volumes separately and putting them into the top-level model.

In DAMPE, the GDML geometry is used both by simulation and reconstruction software.
Table 1. Comparison of GEANT4 per-event simulation time for different DAMPE geometries.

| Energy (GeV) | Non-GDML Time (sec) | GDML (Sensitive) Time (sec) | Factor | GDML (Sensitive + Support) Time (sec) | Factor |
|--------------|----------------------|-----------------------------|--------|--------------------------------------|--------|
| Protons      |                      |                             |        |                                      |        |
| 1 - 10       | 0.09                 | 0.23                        | 2.55   | 0.47                                 | 5.22   |
| 10 - 100     | 0.61                 | 1.35                        | 2.21   | 2.57                                 | 4.21   |
| 100 - 1000   | 4.78                 | 9.31                        | 1.95   | 24.9                                 | 5.21   |
| Electrons    |                      |                             |        |                                      |        |
| 1 - 10       | 0.14                 | 0.55                        | 3.93   | 1.21                                 | 8.64   |
| 10 - 100     | 1.16                 | 4.41                        | 3.80   | 8.93                                 | 7.70   |
| 100 - 1000   | 13.22                | 40.56                       | 3.07   | 88.90                                | 6.72   |

Sensitive volumes and their geometrical coordinates are parsed from the geometry. In addition to sensitive volumes, the model includes all support materials. The total size of the GDML model is 50 MBytes, 150'000 vertexes of triangulated solids. As shown in table 1, having triangulated solids instead of simple shapes in the model leads to expected computing overhead in simulation, which depending on the particle type and energy varies from 2 to 4, if one compares GDML model to the same geometry defined via simple shapes (boxes). Adding support structures to the geometry increases computation time by a factor of approximately two. Finally, note that the level of detail of the initial stl files may adversely affect the complexity of the resulting GDML model and thus lead to yet another increase in computations [11].

5. Conclusions
In this work we present a brief overview of the methods and tools we use to reconstruct the direction of incoming tracks in DAMPE and to perform the alignment of its tracker. We also discuss the related CAD-GDML converter which may be used in other experiments to simplify the workflow from engineering CAD models to a readily usable physics-package detector format.

6. Acknowledgment
This work is supported by the Chinese Academy of Sciences, the Swiss National Science Foundation and INFN, Italy.

References
[1] Chang J 2014 Dark Matter Particle Explorer: The First Chinese Cosmic Ray and Hard gamma-ray Detector in Space Chinese Journal of Space Science, 34 (5), 550-557.
[2] Azzarello P et al 2016 The DAMPE silicon–tungsten tracker Nuclear Instruments and Methods A, 831, 378–384.
[3] Gallo V et al 2015 The test results of the Silicon Tungsten Tracker of DAMPE Proceeding of 34–th ICRC 1199.
[4] Fruhwirth R 1987 Application of Kalman filtering to track and vertex fitting Nuclear Instruments and Methods A, 262, 444–450.
[5] Alpat B et al 2010 The internal alignment and position resolution of the AMS-02 silicon tracker determined with cosmic-ray muons Nuclear Instruments and Methods A, 613, 207–217.
[6] Tykhonov A CAD to GEANT4 converter https://github.com/tihonav/cad-to-geant4-converter.
[7] Geometry Description Markup Language (GDML) https://gdml.web.cern.ch/GDML/.
[8] Agostinelli S et al 2003 Geant4—a simulation toolkit Nuclear Instruments and Methods A, 506 (3), 250–303.
[9] The Apache Xerces Project http://xerces.apache.org/.
[10] Poole C M et al A CAD Interface for GEANT4 physics.med-ph/1105.0963.
[11] CERN CAD Support https://edms.cern.ch/ui/file/1519241/Last_released/1519241.pdf