CLICdp-Conf-2018-004
12 July 2018

Conformal tracking for the CLIC detector

E. Leogrande[1)*]

On behalf of the CLICdp Collaboration

[*] *CERN, Switzerland*

Abstract

Conformal tracking is the novel and comprehensive tracking strategy adopted by the CLICdp Collaboration. It merges the two concepts of conformal mapping and cellular automaton, providing an efficient pattern recognition for prompt and displaced tracks, even in busy environments with 3 TeV CLIC beam-induced backgrounds. In this contribution, the effectiveness of the algorithm has been shown by presenting its performances for the CLIC detector, which features a low-mass silicon vertex and tracking system. Moreover, given its geometry-agnostic approach, the algorithm is easily adaptable to other detector designs and interaction regions, resulting in successful performances also for the CLIC detector modified for FCC-ee.

Talk presented at the 4th Connecting the Dots International Workshop,University of Washington, Seattle, USA 20 Mar – 22 Mar 2018

[1] emilia.leogrande@cern.ch

Conformal tracking for the CLIC detector

Emilia Leogrande[1,*] on behalf of the CLICdp Collaboration

[1]CERN

Abstract. Conformal tracking is the novel and comprehensive tracking strategy adopted by the CLICdp Collaboration. It merges the two concepts of conformal mapping and cellular automaton, providing an efficient pattern recognition for prompt and displaced tracks, even in busy environments with 3 TeV CLIC beam-induced backgrounds. In this contribution, the effectiveness of the algorithm has been shown by presenting its performances for the CLIC detector, which features a low-mass silicon vertex and tracking system. Moreover, given its geometry-agnostic approach, the algorithm is easily adaptable to other detector designs and interaction regions, resulting in successful performances also for the CLIC detector modified for FCC-ee.

1 Introduction

In high-energy physics experiments, efficient tracking strategies are required in order to find tracks among the many vertex and tracker hits originating from detector noise or low-energy particles. These strategies should be robust against possible missing hits and should be fast in terms of computing time. As far as possible, the tracking algorithm should be independent of a particular detector geometry. Conformal tracking complies with all these requirements, by making use of a cellular automaton algorithm in a transformed coordinate system, i.e. the conformal space.

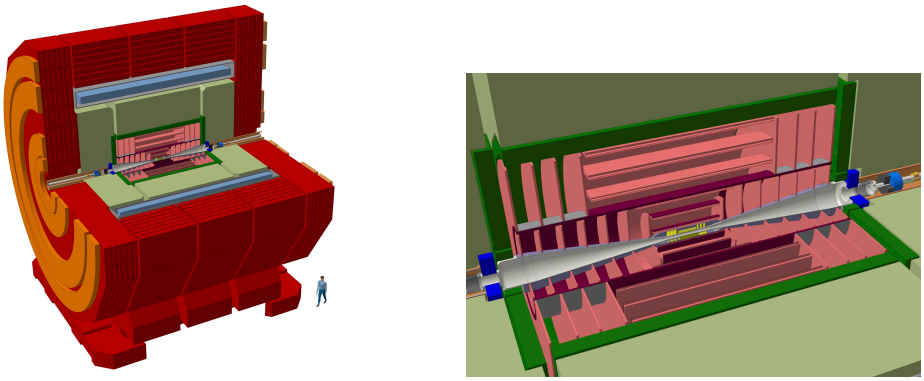
Conformal tracking is currently adopted as the track finding technique for the detector (CLICdet [1]) designed for the CLIC accelerator [2].

This paper is structured as follows. In Sect. 2 the CLICdet performance goals in terms of vertexing and tracking are presented, and the detector model is briefly described. Sect. 3 contains a detailed description of the reconstruction chain, from the digitization of simulated hits, to the pattern recognition in conformal space and to the track fitting. In Sect. 4, examples of tracking performances are shown. Finally, Sect. 5 gives details of recent improvements in the tracking algorithm - this work was in part triggered by this workshop.

2 The CLICdet: performance goals, experimental requirements and detector design

CLIC is a proposed e^+e^- collider, to be operated at three energy stages: 380 GeV, 1.5 TeV and 3 TeV. The CLIC physics program covers precision measurements of the Standard Model, such as the Higgs boson and top quark and their couplings, as well as searches for rare beyond

*e-mail: emilia.leogrande@cern.ch



(a) CLICdet. (b) Tracking region of CLICdet.

Figure 1: The CLICdet detector (left) and the zoom of the tracking region (right).

Standard Model processes. The tracking system has to be able to precisely reconstruct both prompt tracks and tracks from displaced vertices, as well as disappearing tracks, which can be a signature of new physics.

The required performance of the tracking system is:

- transverse momentum resolution $\sigma(\Delta p_{\mathrm{T}}/p_{\mathrm{T}}^2) \leq 2 \times 10^{-5}\,\mathrm{GeV}^{-1}$ for high-energy particles;
- impact parameter resolution with $a \lesssim 5\,\mu\mathrm{m}$ and $b \lesssim 15\,\mu\mathrm{m\,GeV}$ in the parametrization:

$$\sigma_{d_0}^2 = a^2 + \frac{b^2}{p^2 \sin^3\theta}; \tag{1}$$

- best possible angular coverage.

Additional requirements for the detector design arise from the CLIC experimental conditions, in particular the time structure of the beam and the beam-induced backgrounds. A detailed description can be found in [3]. The highest collision energy poses the most challenging conditions. In the following, only examples at 3 TeV are discussed.

In order to achieve high luminosities, extremely small highly-populated bunches are used at CLIC. Such dense bunches, colliding at TeV energies, create very strong electromagnetic fields. As a result, during a bunch crossing, a large number of photons is emitted. Some of these photons create e^+e^- pairs, others result in $\gamma\gamma \rightarrow$ hadrons events. The background from e^+e^- pairs can lead to high occupancy in the vertex and tracker detector. In the CLICdet design, an upper limit of 3% was set on cell occupancy per bunch train (312 bunch crossings). This is achieved by choosing small cell sizes [4].

There are approximately 3.2 $\gamma\gamma \rightarrow$ hadrons events per bunch crossing, with each event resulting in an average of 5 tracks in the detector acceptance. The tracks are mostly forward peaked and their mean momentum is 1.5 GeV. The impact of these $\gamma\gamma \rightarrow$ hadrons events is studied by overlaying 30 bunch crossings of background to each physics event. Details of the overlay procedure are described in [5].

Physics goals and CLIC beam and background conditions result in the detector design shown in figure 1. The beampipe is surrounded by an all-silicon tracking system (vertex detector and tracker). An electromagnetic and a hadronic calorimeter are installed around the tracker. A superconducting solenoid provides a magnetic field of 4 T. The return yoke is equipped with muon chambers. The detector is completed by very forward electromagnetic calorimeters surrounding the beampipe.

The tracking system comprises a vertex detector and a tracker.

The vertex detector is made of three double layers in the barrel and three double layers, arranged in spirals to allow for air cooling, in the forward region. Each layer is pixelated, with $25 \times 25\,\mu m^2$ pixel pitch, which allows for 3 µm single point resolution. The material budget amounts to $0.2\%X_0$ per single layer.

The tracker is divided into an inner and outer part by a light-weight support structure. The inner tracker includes three barrel layers and seven forward disks per side, while the outer tracker includes three barrel layers and four forward disks per side. Besides the first inner tracker disk, which is pixelated like the vertex layers, all other layers are made of silicon microstrips with different dimensions, which allow for 7 µm single point resolution in the plane orthogonal to the field lines. The material budget per layer amounts to about $1\%X_0$, while the support structures and cables sum up to $\sim 2.5\%X_0$. Having such a low-mass tracking system results in low multiple scattering, such that the trajectories of the particles deviate only slightly from the helical path.

3 Simulation and reconstruction

The software framework in which the simulation and reconstruction are implemented, `iLCSoft` [6], is developed by the Linear Collider Community. CLICdet is implemented in full simulation with `DD4hep` [7], as a single source of geometry for simulation, reconstruction and analysis. The basic detector description is a tree-like hierarchy of detector elements (e.g. subdetector, layer, module). For reconstruction purposes, these elements can be equipped with extensions (e.g. cell sizes, single point resolutions, sensitive materials, local-to-global transformation, measurement direction of the hit), providing a high-level detector description [8]. The magnetic field assumed for the studies presented here is a homogeneous 4 T field. Monte Carlo simulations of single particles or complex events are made by using the `Geant4` [9] `FTFP_BERT` physics list. The reconstruction software is implemented in the `Marlin` framework [10]. Large production samples are obtained with the `iLCDirac` grid tool [11].

3.1 Hits simulation and reconstruction

When the Monte Carlo particles traverse sensitive layers of the detector, they leave *simulated* hits, which are stored as sets of information including the hit spatial coordinates, energy deposition, time of arrival, estimated momentum, etc. The combined collection of simulated hits from the physics event and the background particles, overlaid within the integration time window, is then digitized: to approximate the effect of pixels/strips, the positions of the hits are smeared with Gaussian distributions with σ equal to the single point resolution of the subdetector in both directions. The smearing is done in the local coordinate system in each layer. The *reconstructed* (i.e. digitized) hits are used for pattern recognition.

3.2 Tracks in conformal space

A charged particle moving in a magnetic field with a velocity component orthogonal to the field direction travels in a helical path, whose projection in the bending plane x, y of the solenoid is a circle. Conformal mapping [12] is based on a geometry transformation that maps circles in the x, y plane passing through the origin into straight lines in the u, v plane, with

$$u = \frac{x}{x^2 + y^2}, \qquad v = \frac{y}{x^2 + y^2} \tag{2}$$

where the circles in the x, y plane are defined by the equation

$$(x-a)^2 + (y-b)^2 = r^2 = a^2 + b^2 \tag{3}$$

and the straight lines in the u, v plane follow

$$v = \frac{1}{2b} - u\frac{a}{b}. \tag{4}$$

An illustration of reconstructed hits transformed into conformal space according to Eq. 2 is shown in figure 2. Hits in the vertex detector result in larger values of u, v, while hits from the tracker have smaller conformal coordinates.

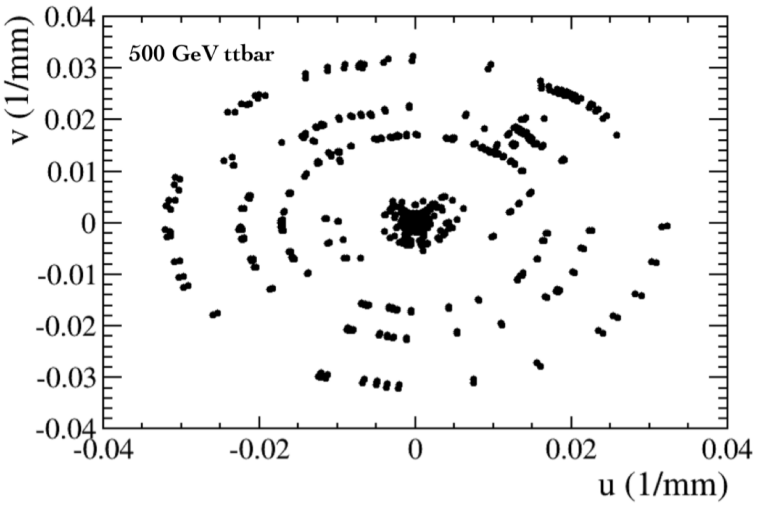


Figure 2: Illustration of hits in conformal space.

Standard conformal mapping methods [13] find track candidates by further transforming the measurements in the u, v plane into polar coordinates and by grouping hits with the same polar angle as they will be aligned along straight lines pointing to the origin. However, tracks with non-zero impact parameter and particles undergoing non-negligible multiple scattering will deviate from the expected polar angle. For this reason, a cellular automaton is preferred to perform a straight line search in the u, v plane.

3.3 Pattern recognition

The two methods of **building** and **extending** tracks are used in the process of pattern recognition, as explained in the following.

Building the tracks starts by considering each hit of the selected collection as a *seed*. From each seed, a search for nearest neighbours is performed, neighbours being defined as hits within a certain angular region and below a certain distance from the seed. Segments connecting the seed to each neighbour hit are so-called *seed cells*, and are then extended to *virtual* hits, i.e. hits sitting on the prolongation of the cell[1], from which the search for nearest neighbours is repeated. New cells connecting the end point of the seed cell and the new neighbours are created. A cell contains information about its start and end point, but also about how many other cells are further connected to it. Each subsequent link increments the

[1]The distance between the end point of the seed cell and the virtual hit is set to 25% of the seed cell length.

cell *weight* by 1 unit, such that the higher the weight, the higher the potential of the cell to make a track. *Cellular tracks* are chains of cells, made by following the paths leading from the highest-weighted cells back to the seed hit. If more than one path is available, the cellular tracks are branched and all possible paths are kept at this level. Once all cellular tracks stemming from the seed hit are found, they are fitted in conformal space with a linear regression in u, v and s, z plane[2]. For each cellular track, the χ^2 per degree of freedom is computed. In this procedure, each hit is progressively removed and the track is refitted without the hit, in order to be able to recover good tracks in which a bad hit was included in the pattern recognition. Per seed hit, the best track is chosen as the one with best χ^2/ndf. A check for clones, i.e. cellular tracks with at least 2 overlapping hits, allows the longest track to be retained by preference, unless the χ^2/ndf is too large and the shortest is preferred. Hits in the cellular track are marked as used and the pattern recognition continues with the other hits in the collection.

Extending the tracks begins with a *seed cell* rather than a seed hit. The seed cell is, in this case, the segment connecting the outermost (in global coordinates) hits of the track obtained with the *building* method. The seed cell is prolonged and the search for nearest neighbours is performed in the same way as for the building stage. Each neighbour is progressively added to the cellular track and the new χ^2/ndf is computed. If the increment of the χ^2 per hit is below a certain threshold, the hit is accepted on the track.

The pattern recognition is performed in steps, in a way to limit the combinatorics:

- tracks are **built** using the collection of hits in the vertex detector barrel
- tracks built in the vertex detector barrel are then **extended** into the vertex detector disks
- with the leftover hits in the combined vertex detector barrel and disks collection, tracks are **built** in this order:
  - with default cuts
  - with looser cuts on the angle between cells in the nearest neighbour search
  - with looser cut on the χ^2 per hit
- tracks built in the vertex detector are then **extended** into the tracker
- a final step is dedicated to **build** displaced tracks, with specific cuts in terms of angle between cells, χ^2 per hit, direction of extrapolation

Particular attention is required when building displaced tracks. As defined in Sect. 3.2, circles in global space *passing through the origin* are mapped into straight lines in conformal space. This is not the case for particles not originating from the interaction point, e.g. particles from a decay, particles produced in the interaction with the material, or displaced tracks. To cope with such cases, a quadratic term is included in Eq. 4, modifying the straight line into a parabola. This makes the reconstruction of displaced tracks possible to some extent. In addition, some modifications with respect to the tracking strategy for prompt tracks have been adopted:

- broaden the angular region in the search for nearest neighbours to take into account deviations from straight trajectories;
- increase the minimum number of hits to 5 to reduce the number of cellular tracks accepted;
- invert the search order: going from tracker to vertex hits allows to start the seeding where the deviations from straight lines are higher.

[2] s is a coordinate along the arc of the helix. Fitting in s, z reduces the combinatorics, by making use of the coordinates discarded by the projection into circles.

This pattern recognition strategy is geometry-agnostic. The algorithm has no dependency on the geometry beyond the concept of barrel and disks, the layer number for the creation of cells, and the resolutions associated to individual hits.

3.4 Implementation

The *ConformalTracking* pattern recognition is implemented as a processor in the C++ Marlin framework [14]. Internally, the pattern recognition uses its own classes to represent hits, cells and tracks. Before the algorithm starts, the hits are converted from the `LCIO::TrackerHits` into `KDClusters`, i.e. objects of a light-weight hit class, containing all information needed for performing the cellular automaton, ordered in a binary tree (`KDTree` [15]). Cells are implemented as objects of the `Cell` class, namely segments connecting two 2-D space points (in this case `KDClusters`) and holding the cell weight. The final cellular tracks (`std::vectors` of `Cells`) are then forwarded to the fitter.

3.5 Track fitting

The track fitting is done in global coordinate space. It consists of:

- a prefit, which makes use of three hits of the track (typically first, middle and last hit) to construct a helix, the parameters of which will be used to initialize the full fit;
- a Kalman filter fit [16], where hits are added one by one, the χ^2/ndf is computed and the track state (track parameters and uncertainties in the form of covariance matrix) is propagated to the next hit.

4 Tracking performance

A detailed study of the performance of the tracking algorithm for the CLICdet is documented in [5]. In the following, a selection of results is being presented.

4.1 Single particles

The simplest way to probe the tracking performance is by simulating single muons passing through the detector at different energies and polar angles. By reconstructing these isolated tracks, the algorithm is validated and low-level physics performance is assessed. The target transverse momentum resolution and impact parameter resolution, as described in Sect. 2, are achieved [5]. Single particle efficiency, defined as the fraction of reconstructable[3] particles which have been reconstructed, is around 100% for all energies down to few MeV and all polar angles, except in the very forward direction (10° tracks) where the efficiency drops by 2% [5].

To study the reconstruction of displaced tracks, the tracking efficiency as a function of production vertex radius of the particles is analyzed. For this study, muons were generated with a displaced vertex within $0 < y < 600$ mm and with an angular distribution in a 10°-wide cone around the y axis, i.e. $80° < \theta, \phi < 100°$. This choice simplifies the analysis of the results, since the number of expected hits per particle is defined by counting the barrel layers with radius larger than the particle vertex. The efficiency as a function of production vertex 2D-radius R, i.e. $\sqrt{x^2 + y^2}$, is shown in figure 3 for muons with momentum of 1, 10 and 100 GeV.

[3] A particle is defined *reconstructable* if it is stable at the generator level (`genStatus = 1`), has a $p_T > 0.1$ GeV, is within the detector acceptance ($|\cos\theta| < 0.99$) and has at least 4 unique hits, i.e. in different subdetector layers.

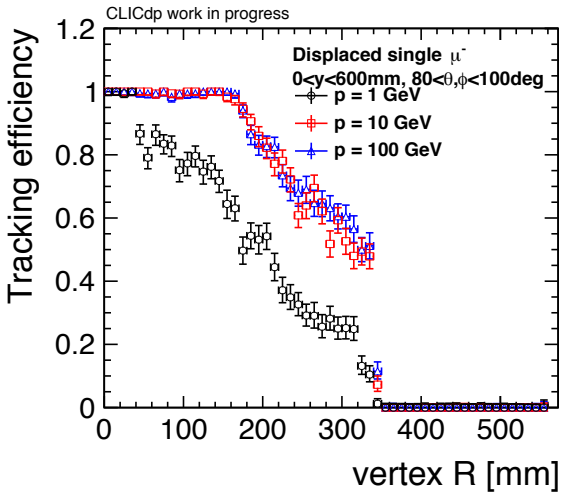


Figure 3: Tracking efficiency as a function of production vertex radius for single muons with momenta of 1, 10 and 100 GeV, uniformly generated with a displaced vertex within $0 < y < 600$ mm and with an angular distribution of $80° < \theta, \phi < 100°$.

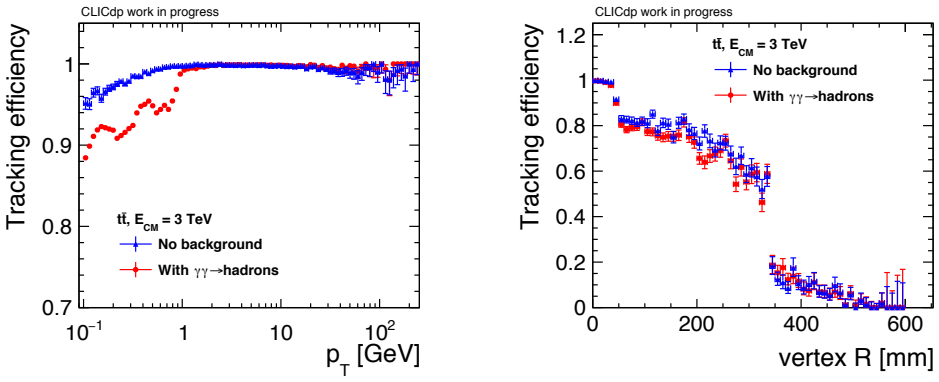
For 1 GeV muons produced within the first two layers of the vertex detector ($R < 40$ mm) the algorithm is fully efficient. For those that are produced right after the second vertex barrel ($R > 40$ mm), the efficiency drops by 20%, due to the fact that there is not enough momentum left to reach the minimum number of detector layers. For higher-energy muons, the efficiency is 100% for $R < 160$ mm. If they are produced further away, the efficiency starts dropping. Regardless of the energy, an abrupt fall-off is observed for all tracks with $R > 350$ mm. This is an effect of the applied cuts, since those tracks will only traverse 4 sensitive layers, while the required minimum number of hits for displaced tracks is 5.

4.2 Complex events without and with background overlay

The performance for complex events has been studied for different event types (Z-like boson events decaying into two light-quark dijets, $t\bar{t}$ and $b\bar{b}$ events) and different center-of-mass energies. To all samples, a 3 TeV CLIC-beam background has been overlaid. The tracking efficiency has been found to be largely independent of the event type and centre-of-mass energy [5]. The definition of tracking efficiency is the same as given for single particles (see Sect. 4.1), with the additional requirement of the reconstructed track to be *pure*, i.e. at least 75% of its hits are associated with the same Monte Carlo particle.

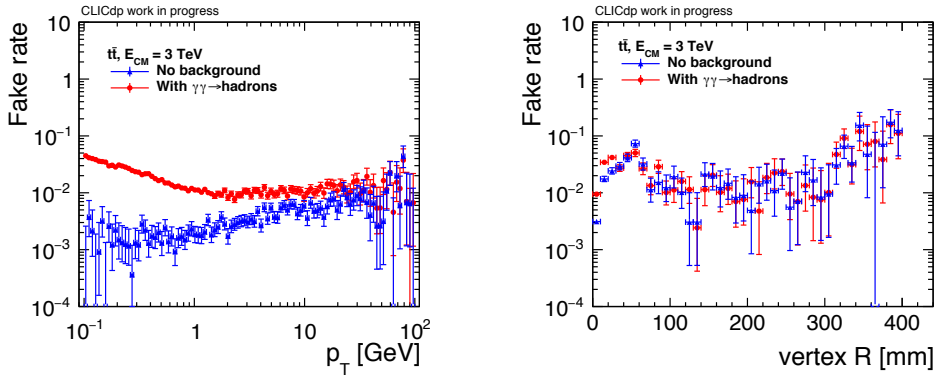
In figure 4, the tracking efficiency in a $t\bar{t}$ sample at 3 TeV is shown as a function of p_T (left) and production vertex radius (right). The following cuts are applied in the left plot: $10° < \theta < 170°$, production vertex radius < 50 mm, distance of closest Monte Carlo particle < 0.02 rad. Similarly, for the right plot: $p_T > 1$ GeV, $10° < \theta < 170°$, distance of closest Monte Carlo particle < 0.02 rad. The effect of the background is visible only for tracks with $p_T < 1$ GeV, as expected given the low-p_T particle spectrum from $\gamma\gamma \rightarrow$ hadrons (see Sect. 2). For tracks with transverse momentum above 1 GeV, the tracking efficiency is 100%. The efficiency dependence on the production vertex radius is similar to the one observed for displaced muons at 1 GeV (see figure 3), since the particle spectrum of $t\bar{t}$ events peaks at low energies and decreases exponentially with energy.

The tracking efficiency is complemented by studying the rate of fake tracks, namely the fraction of *impure* tracks out of the total reconstructed tracks. A track is considered impure



(a) vs p_T (b) vs $R = \sqrt{x^2 + y^2}$

Figure 4: Tracking efficiency for $t\bar{t}$ events at 3 TeV with and without $\gamma\gamma$ →hadrons overlay as a function of p_T (left) and production vertex radius (right).



(a) vs p_T (b) vs $R = \sqrt{x^2 + y^2}$

Figure 5: Fake rate for $t\bar{t}$ events at 3 TeV with and without $\gamma\gamma$ →hadrons overlay as a function of p_T (left) and production vertex radius (right).

if less than 75% of its hits are associated with the same Monte Carlo particle. As shown in figure 5, the background increases the fake rate at low p_T. Moreover, the fake rate shows a peak corresponding to the drop of efficiency for tracks produced after the two first vertex barrel layers (R > 40 mm) and then increases for R > 200 mm.

4.3 Tracking performances of the CLD detector at FCC-ee

The conformal tracking has been successfully applied to the CLIC-Like Detector (CLD) at FCC-ee [17]. CLD is an adapted version of CLICdet to the FCC-ee experimental conditions. It has a similar design for the vertex and tracker detector, but its innermost layer is closer to the interaction point (17 mm instead of 31 mm in CLICdet, given the smaller beampipe radius at FCC-ee) and its outermost layer has a larger radius (2.15 m instead of 1.5 m in CLICdet) to cope with the smaller magnetic field (2 T instead of 4 T in CLICdet). The modified detector configuration and the different beam beackgrounds have required the modification of some parameters of the tracking software, e.g. the angular region and the maximum distance for nearest neighbours search. A detailed performance study is ongoing [17]. In figure 6, the tracking efficiency for CLD is shown in 91 GeV Z-like boson events decaying at rest into two light-quark dijets as a function of p_T with and without 20 FCC-ee bunch crossings of

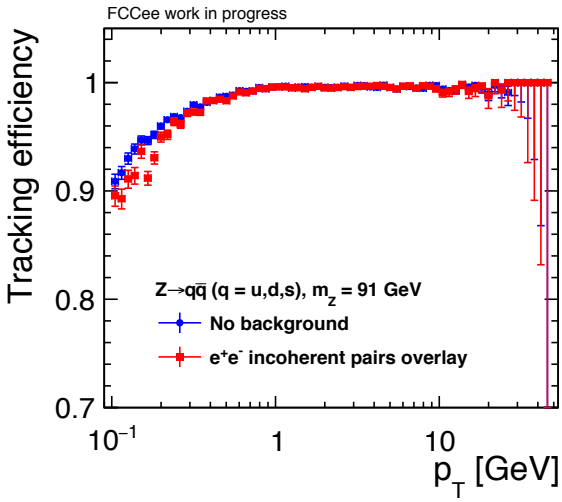


Figure 6: Tracking efficiency for Z-boson like events at 91 GeV with and without incoherent pairs overlay as a function of p_T for the CLD at FCC-ee.

incoherent pairs overlaid. The effect of the background in this study is negligible and the tracking efficiency is 100% for tracks with transverse momentum above 1 GeV.

5 Recent improvements

At the time of this workshop, it was observed that 20% of the particles were being reconstructed multiple times (typically as two tracks instead of one). The two (or more) tracks either were segments of the real track or shared complementary hits in an apparently random way. In the following, both cases are generically dubbed *split* tracks. This problem involved complex events, with and without background overlay, as well as isolated particles.

With the "binary" definition of efficiency as the fraction of reconstructable particles which had been reconstructed, having one or more reconstructed tracks per particle was not noticeable in the performances of fully efficient samples, e.g. single prompt tracks. On the other hand, some cases of inefficiencies, for instance for displaced single particles, turned out to be attributed to this issue. Moreover, this tracking problem was having severe repercussions both on the jet energy resolution, since the track matching with the calorimeter clusters was affected by the splitting of the tracks, and on the flavour tagging performances, as these multiple tracks were used to obtain (fake) secondary vertices.

After a thorough investigation, triggered by discussions at this workshop, it emerged that there were different problems contributing to the issue of split tracks.

The first problem concerned the pattern recognition strategy, in particular the reconstruction of the combined collection of vertex barrel and endcap hits (third step in the chain described in Sect. 3.3). Here, a first reconstruction with tighter cuts in terms of angular region and χ^2/ndf was performed for computation speed reasons. Tracks built at this stage were stored and leftover hits in the collection were examined in a second reconstruction phase with looser cuts. However, this led to reconstructing a first track with the tighter cuts and, with the leftover hits, a second track with the looser cuts. The problem has been solved by adopting for this step broader cuts from the beginning.

The second problem concerned the treatment of clones (i.e. tracks sharing at least two hits). Even in events in which the right hits are assigned to the track during the pattern recognition, the check for clones was implemented such that shorter tracks were preferred

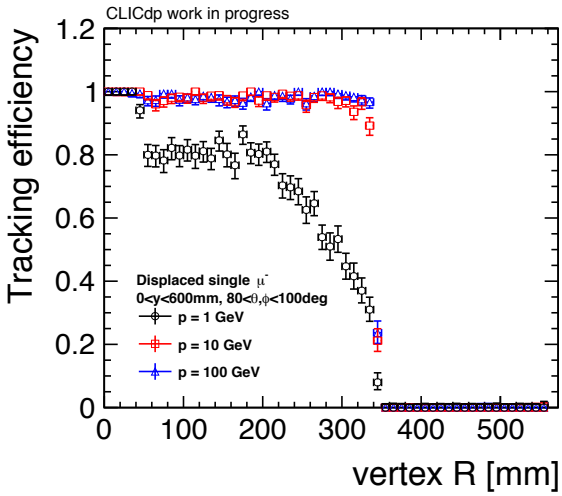


Figure 7: Tracking efficiency of CLICdet for displaced single muons after improvements of the tracking algorithm.

in case of better χ^2/ndf. This led to a shorter-than-expected track being reconstructed and hits being left unused, available for forming a second track. The problem has been solved by always preferring the longest track.

The result of these improvements is visible in the enhanced efficiency for displaced muons shown in figure 7 (compared to the one shown in figure 3), as well as in a much improved jet energy resolution [5]. The flavour tagging studies with the updated tracking software are in progress.

6 Conclusions

Conformal tracking proves to be a powerful technique for track finding. The performance for prompt, as well as displaced tracks, both in the case of the CLIC detector CLICdet and its adaptation for FCC-ee (the CLD detector) is found to be very satisfactory. The impact of beam-induced backgrounds on the tracking performance in complex events is found to be negligible, except for the lowest energy tracks. Using the algorithm in the context of Particle Flow Analysis for the reconstruction of jets, and in physics studies, will trigger further improvements to the conformal tracking for CLIC and FCC-ee.

References

[1] N. Alipour Tehrani et al., *CLICdet: The post-CDR CLIC detector model*, Tech. Rep. CERN (2017), URL https://cds.cern.ch/record/2254048

[2] M. Aicheler et al., *A Multi-TeV Linear Collider Based on CLIC Technology: CLIC Conceptual Design Report*, Tech. Rep. CERN (2012), URL https://cds.cern.ch/record/1500095

[3] L. Linssen et al., *Physics and Detectors at CLIC: CLIC Conceptual Design Report*, Tech. Rep. CERN (2012), URL https://cds.cern.ch/record/1425915

[4] A. M. Nurnberg et al., *Requirements for the CLIC tracker readout*, CLICdp-Note-2017-002 (2017), URL https://cds.cern.ch/record/2261066

[5] D. Arominski et al., *A detector for CLIC: main parameters and performance* (2018), URL https://edms.cern.ch/document/1999007/

[6] *ILCSoft repository*, URL https://github.com/iLCSoft
[7] M. Frank et al., J. Phys. Conf. Ser. **513**, 022010 (2013)
[8] A Sailer et al., J. Phys. Conf. Ser. **898**, 042017 (2017)
[9] S. Agostinelli et al., Nucl. Instrum. Methods A **506**, 250 (2003)
[10] F. Gaede, Nucl. Instrum. Meth. **A559**, 177 (2006)
[11] C. Grefe et al., J. Phys. Conf. Ser. **513**, 032077 (2013)
[12] M. Hansroul et al., Nucl. Instrum. Methods A **270**, 498 (1988)
[13] A. Strandlie et al., Rev. Mod. Phys. **82**, 1419 (2010)
[14] *Conformal Tracking repository*, URL https://github.com/iLCSoft/ConformalTracking
[15] M. B. Kennel, *KDTREE 2: Fortran 95 and C++ software to efficiently search for near neighbors in a multi-dimensional Euclidean space*, arXiv:physics/0408067 (2004)
[16] *DDKalTest repository*, URL https://github.com/iLCSoft/DDKalTest
[17] N. Alipour Tehrani et al., *CLD - a detector concept for FCC-ee* (2018), URL https://edms.cern.ch/document/1999009/