CALICE Analysis Note CAN-054
October 26, 2015

Separation of nearby hadronic showers in the CALICE SDHCAL prototype detector using ArborPFA

The CALICE Collaboration [1]

Abstract

A new reconstruction algorithm, ArborPFA, is being developed to separate nearby hadronic showers in the SDHCAL prototype. This intends to demonstrate the capability of high granularity hadronic calorimeters such as the SDHCAL to efficiently apply Particle Flow Algorithms. The reconstruction algorithm we present here uses the tree-like structure features of hadronic showers, that high granular calorimeters reveal, to associate hits belonging to each hadronic shower and to reduce confusions between two close-by showers. The results of these studies indicate a good single particle efficiency and reconstructed energy. A powerful separation down to distances of 5 cm is obtained.

This note contains preliminary CALICE results, and is for the use of members of the CALICE Collaboration and others to whom permission has been given.

[1] Corresponding authors: Ete Remi; `rete@ipnl.in2p3.fr`

Contents

1 Introduction

To study the Higgs boson properties and to extend the discovery of new particles beyond the scope of LHC, linear (e.g ILC or CLIC) and circular (e.g FCC or CEPC) e^+ e^- colliders are proposed. An important requirement of such a machine is to provide a good jet energy resolution (ΔE/E ~3-4%) in order to distinguish between Z and $W^\pm$ bosons as well as to study the Higgs boson properties.

The Particle Flow concept has been proposed to achieve the ILC benchmarks [1]. This algorithm aims to individually reconstruct particles using the most appropriate sub-detector for the energy and momentum measurement. An implementation of the particle flow algorithm called PandoraPFA has been developed [2] and successfully applied in ILD physics performance studies and to close-by hadronic showers separation in the CALICE SiWECal and AHCAL [3].

To apply efficiently the Particle Flow Algorithms, both good energy resolution and fine transverse and longitudinal segmentation should be provided by the *electromagnetic calorimeter* (ECAL) and the *hadronic calorimeter* (HCAL).

Different calorimeter technologies are currently under study by the CALICE collaboration to fulfill these requirements. In this framework, a *semi digital hadronic calorimeter* prototype (SDHCAL) was built [4] and successfully tested at the CERN H6 test beam line of the SPS (CERN) in 2012. With a transverse readout segmentation of 1 cm^2, 48 sampling layers and good energy resolution [4], this calorimeter statisfies ILC requirements.

In this paper, we present another approach of the particle flow : the ArborPFA approach. The algorithm has been designed for high granularity calorimeters and applied to SDHCAL test beam data. We evaluate the performance of the algorithm on single pion events and to study the ability of the algorithm to separate two overlaid pion showers at different separation distances and energies.

2 The SDHCAL prototype

The SDHCAL prototype is a sampling calorimeter which consists of 48 layers alternating 20 mm steel absorbers and 6 mm gas resistive plate chamber (GRPC) with embedded electronics. The gas gap between the two electrodes of the GRPC is 1.2 mm. 9216 pads (96 x 96) of 1 cm^2 compose the readout of each chamber, leading to a total number of 442368 channels. Signals from particles crossing the gas gap are recorded on those pads in a 2-bits format corresponding to 3 thresholds on the induced charge. A complete description of the calorimeter setup and its features can be found in [4].

The test beam data used in this paper were taken at the CERN H6 beam line in 2012. The pion event selection is also performed according to the selection presented in [4].

The reconstructed energy of a single particle is computed as follows :

$$E_{rec} = \alpha(N_{hit}) \cdot N_1 + \beta(N_{hit}) \cdot N_2 + \gamma(N_{hit}) \cdot N_3 \tag{1}$$

where N$_1$, N$_2$ and N$_3$ are the number of hits between threshold 1 and 2, between 2 and 3, and above 3, N$_{hit}$ = N$_1$ + N$_2$ + N$_3$ and α, β and γ are quadratic functions of the number of hits N$_{hit}$. The nine parameters of these functions are extracted from a χ^2 minimization :

$$\chi^2 = \sum_{evt} \frac{(E_{beam} - E_{rec})^2}{E_{beam}} \tag{2}$$

This minimization was performed over all the hadronic events in the energy range [10, 80] GeV with steps of 10 GeV.

3 The Arbor particle flow algorithm

The Arbor approach has been developed by Henri Videau for the ALEPH collaboration and was recently adapted [5] for the ILD detector design. It is based on the idea that the hadronic shower development follows a tree-like topology.

Figure 1 shows an example of a shower development (left) where all main components of the shower are present : charged particles, neutral particles, electromagnetic and hadronic components. The same figure shows on the right a sampling calorimeter view of a shower interaction as seen in a highly granular calorimeter. The black arrows drawn on this view suggest the tree-like topology development of the shower.

With such an approach, the shower reconstruction follows a principle close to the underlying physics.

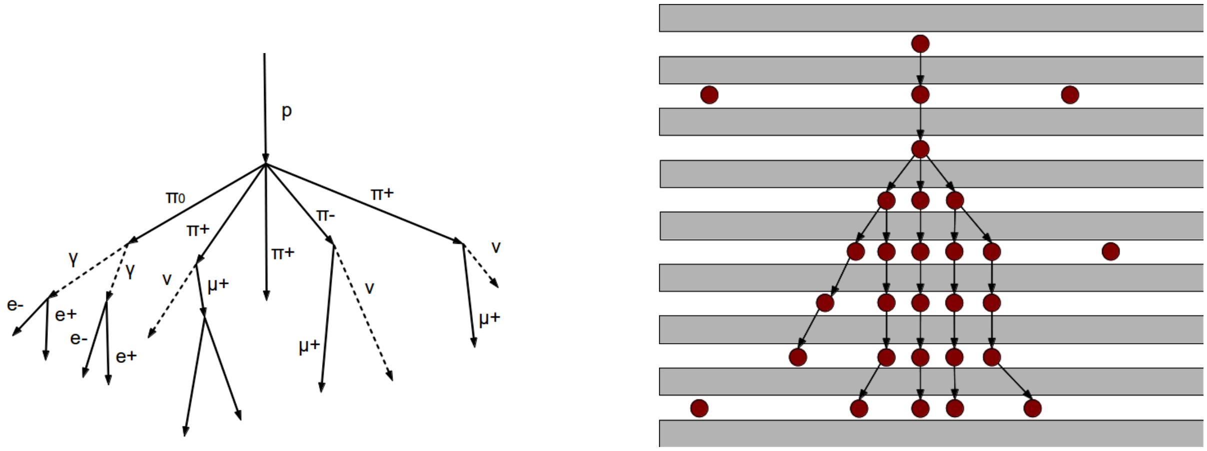

Figure 1: Left : schematic view of an induced proton shower. Right : schematic view of a reconstructed shower in a calorimeter with calorimeter hits (red).

The algorithm we use here is implemented using the PandoraSDK API as a toolkit for generic PFA development [6]. The API is used in a Marlin [7] C++ processor as part of the reconstruction chain in ILCSOFT [8] that produces reconstructed particle flow objects (PFOs). A full description of the algorithm is given in Appendix A.

4 Single particle study

4.1 Setup

To study the single particle performance of the algorithm, we use the SDHCAL charged pion data taken at CERN on the H6 line of SPS in 2012 from 10 GeV up to 80 GeV. In order to select only charged hadrons, an event selection is performed based on topological variables. Muons coming from the beam and cosmic muons are first removed by requiring that the number of hits over the number of layers with more than one hit must be higher than 2.2. Electrons are then removed by requiring that the shower starting layer has to be higher than 5 or the number of touched layers must be higher than 30. Neutral particles are also removed by requiring that a minimum of 4 hits are found in the 5 first layers. A more detailed description of the event selection is described in [4].

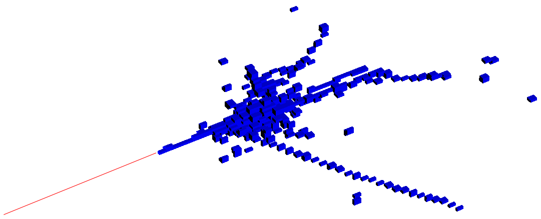

Figure 2: Event display of a 50 GeV pion shower in the SDHCAL detector.

To correctly emulate a charged hadron for the reconstruction program and since there was no tracker in front of the SDHCAL during beam tests, a fake track is created in front of the calorimeter. A global barycentre of all hit positions in the transverse plane (X and Y axes) is calculated. A new barycentre is then determined using only hits in the 4 first layers and within a region of 8x8 cells around the global barycentre in the X and Y directions. This defines the shower entering point in the first layer. From the entering point of the shower, a straight track is created along the beam axis (Z direction) with momentum equal to that of the beam.

The calorimeter hits and the created track are then loaded into the PandoraSDK toolkit [6] within a single hcal endcap geometry (no magnetic field) with the SDHCAL prototype dimensions and processed by the ArborPFA algorithms. An event display of a single pion event is shown on figure 2.

4.2 Single particle analysis

We define the efficiency of the single particle reconstruction, or the hit clustering efficiency ϵ_s, as the fraction of hits recovered by the ArborPFA program and correctly attached to the track in front of the calorimeter. Figure 3 shows the mean efficiency of the single particle reconstruction (left) and the mean number of reconstructed particles (right) as a function of the beam energy after applying ArborPFA. Two typical distributions of

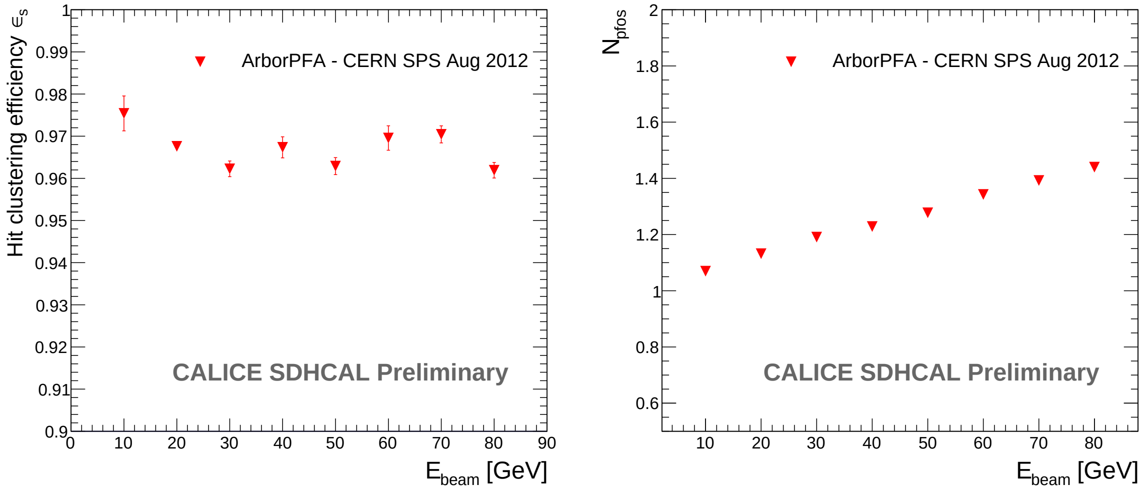

Figure 3: Hit clustering efficiency (left) and the mean number of reconstructed particles (right) after ArborPFA reconstruction on single pion shower events with the SDHCAL prototype.

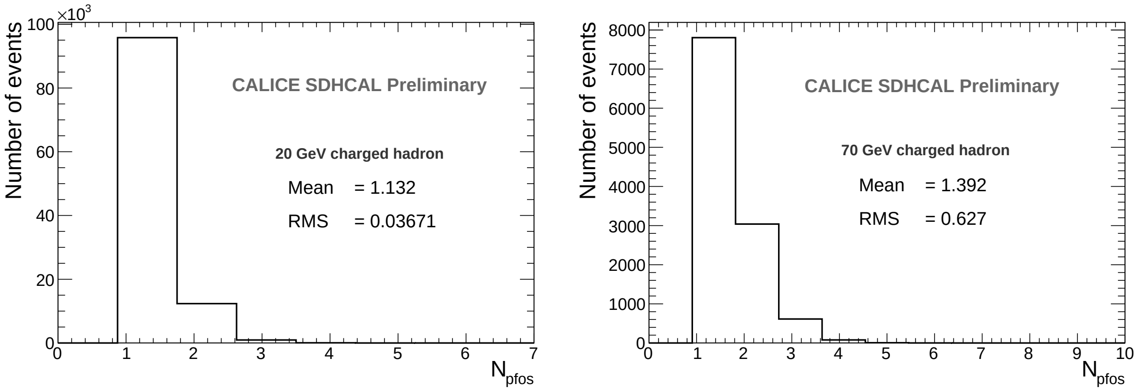

Figure 4: Distributions of the number of reconstructed particles for a 20 GeV single charged hadron (left) and a 70 GeV single charged hadron (right).

the number of reconstructed particles are shown in figure 4 as an example. The hit clustering efficiency shows a constant efficiency of over 96% over the whole beam energy range. Since the number of hits increases with the energy, the number of missed hits in the reconstructed charged particle also increases. Consequently, the mean number of reconstructed particles shows an increase which is directly due to shower splitting. This number grows up to 1.45 for particles at 80 GeV but has only a small impact on the reconstructed energy and energy resolution because the small additional clusters represent a small amount of energy.

Figure 5 shows the reconstructed energy and energy resolution of a single charged pion before running the ArborPFA program and the cluster associated to the track after re-

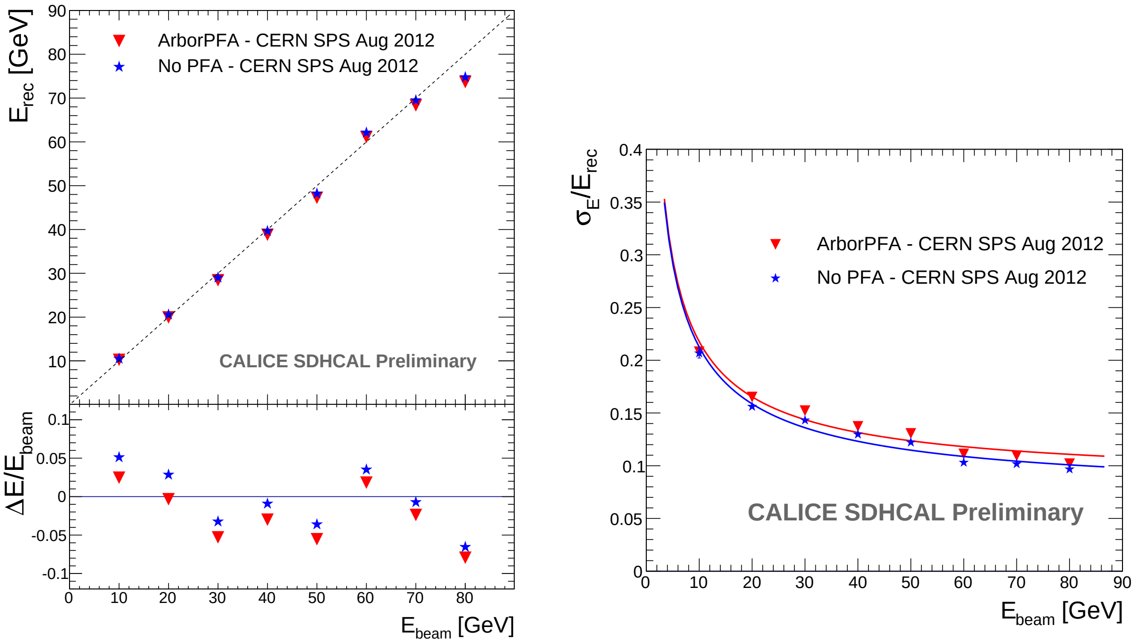

Figure 5: Reconstructed energy (left) and energy resolution (right) before (blue) and after (red) ArborPFA reconstruction on single pion shower event with the SDHCAL prototype.

construction.

These energy points are extracted using two fits of the energy distributions : i) a gaussian distribution fit over the full reconstructed distribution is first performed. The mean $\mu_{E,first}$ and width $\sigma_{E,first}$ are extracted, and ii) a second gaussian fit is then performed over the range [$\mu_{E,first}$-1.5*$\sigma_{E,first}$; $\mu_{E,first}$+1.5*$\sigma_{E,first}$]. From the latter, we extract the final values of the reconstructed energy and energy resolution defined as the mean μ_E and the width σ_E respectively of the gaussian fit (same procedure as applied in [4]). The deviation from linearity is shown below the reconstructed energy and is defined as :

$$\Delta E/E_{beam} = \Big(\mu_{rec} - E_{beam}\Big)/E_{beam} \tag{3}$$

The efficiency plot has shown that some hits are missing after reconstruction so it is expected to have a small energy decrease in the reconstructed energy. Nevertheless, the linearity is still within 8% as before applying the reconstruction. The energy resolution is also not so much affected after reconstruction.

5 Separation of two close-by hadronic showers

The ability of a particle flow algorithm to disentangle close-by showers is a key point for the reconstruction in detectors such as ILD of the ILC. To study the confusion between neutral and charged hadrons and the ability of the ArborPFA algorithm to disentangle them, we again use the same test beam data of the SDHCAL prototype. Two different pion showers are first overlaid in the same event and the ArborPFA algorithm is run on the overlaid event with the same parameters as for the single particle study. An analysis of the separation is then performed in order to extract the performance of the algorithm.

5.1 Overlay procedure and setup

In order to study the separation of nearby hadronic showers, two events from test beam data are overlaid in one event. We have chosen to overlay 10 GeV pion events and pion events at different energies from 10 GeV to 50 GeV, in steps of 10 GeV. The distance between shower entry points was taken between 5 cm and 30 cm with steps of 5 cm. The choice of this energy range is motivated by the fact that it is the typical single particle energy range foreseen at the ILC within jets [9].

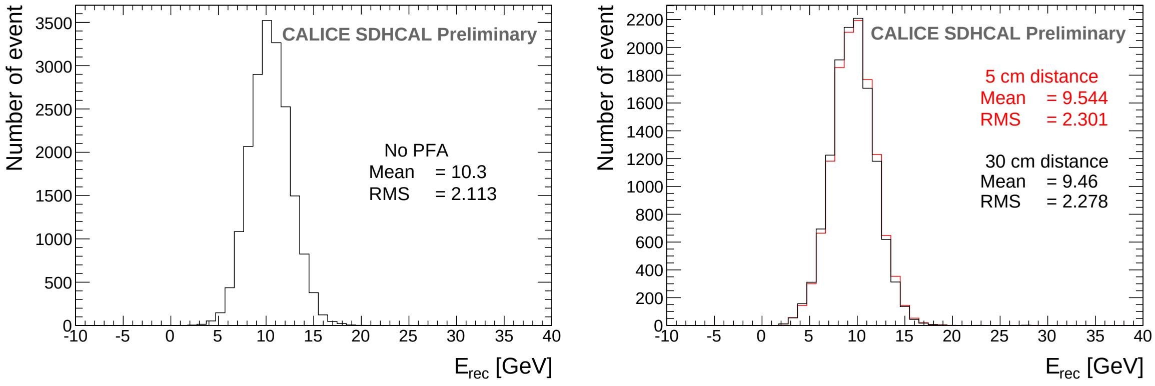

Figure 6: Left : The reconstructed energy of a 10 GeV charged hadron before the overlay procedure. Right : The reconstructed energy of the 10 GeV neutral hadron after the overlay procedure with a 50 GeV charged hadron with a separation distance of 30 cm (black) and 5 cm (red).

The overlay event algorithm is processed as follow :

1. The entering track segment of each shower is determined as for the single particle case. This allows to identify the entering points and starting points of each shower.

2. The hits belonging to the 10 GeV pion primary track segment are removed from the event in order to emulate a neutral hadron shower.

3. The two showers are then centred along the X and Y axis at the center of the calorimeter. No shift is performed on the Z direction (beam line).

4. The showers are then shifted along the X axis by a distance of -d/2 for the neutral hadron and +d/2 for the charged particle, where d is the distance to the calorimeter center in cm.

5. The two events are then overlaid. At this step a problem may occur : while mixing the showers in the event, pair of hits may overlap in the same cell. Knowing that we are using a semi digital readout and that the information of the deposit charge in each cell is not available in the data, we need to assign a new threshold by using an approximation. The most intuitive one is to keep the highest threshold of the two hits. Figure 6 (right) shows the reconstructed energy of the 10 GeV fake neutral hadron overlaid with a 50 GeV charged hadron at 30 cm distance (black) and 5 cm distance (red). The latter case is the worst that can appear in this study given the energy points and the distances we have chosen. By comparing the two plots, we can see that the effect of this approximation on the reconstructed energy is negligible.

6. The hits are tagged with respect to our initial showers. The overlaid hits are tagged differently so that the information on the overlaid hits can be retrieved after reconstruction.

7. A new event is created containing the overlaid showers and the entering point of the charged particle track after shifting. An example is shown on figure 7.

Figure 6 shows the reconstructed energy of a 10 GeV charged hadron before the overlay procedure (left). A small energy difference of approximatively -0.8 GeV is observed between the reconstructed energy before the overlay (left) and after the overlay (right) for the 30 cm case (black), due to the track segment hits removal while overlaying the two showers.

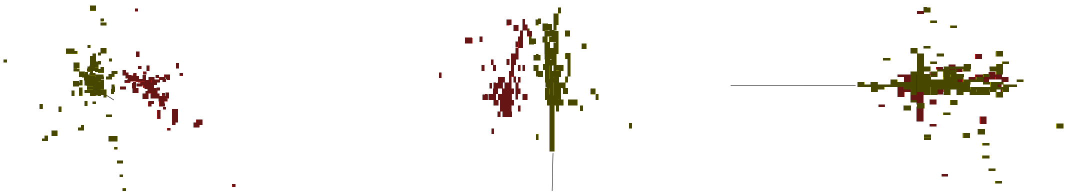

Figure 7: Display of a 10 GeV fake neutral hadron overlaid with a 30 GeV charged hadron separated by 20 cm in three different views (XoY on left, XoZ in center and YoZ on right). Colours correspond to the reconstructed PFOs after running the ArborPFA program. The black straight line is the fake track generated in front of the calorimeter.

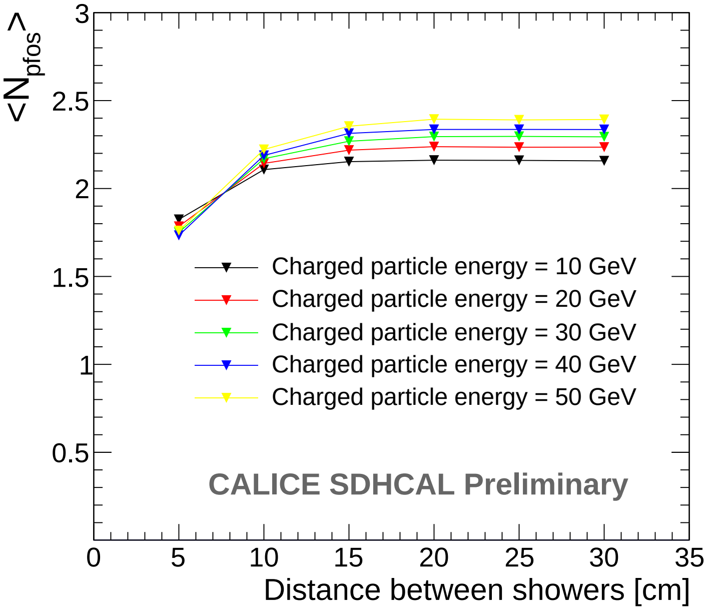

Figure 8: The mean number of PFOs after running the ArborPFA program on overlaid 10 GeV emulated neutral hadron and charged hadrons at different energies.

5.2 Overlaid particles analysis

Figure 8 shows the mean number of PFOs after running the ArborPFA program on a 10 GeV fake neutral hadron overlaid with a charged hadron at different energies and different separation distances. The behaviour at large separation distance with the charged particle energy matches the mean number of PFOs of the single particle study. We can also see that the sum of the mean number of PFOs for the single particle is compatible with the mean number of PFOs for the overlay. The mean number of PFOs is stable at large separation distances but slightly decreases at 5 cm from about 2.1 PFOs down to about 1.8 PFOs due to the showers overlaps and the resulting confusion.

To quantify the separation, we define the efficiency and the purity related to the reconstruction of one of the two related showers as :

$$\epsilon = \frac{Nhit_{good}}{Nhit_{ini,tot}} \tag{4}$$

$$\rho = \frac{Nhit_{good}}{Nhit_{rec,tot}} \tag{5}$$

with Nhit$_{good}$ the number of hits that initially belong to the particle and correctly assigned after reconstruction, Nhit$_{rec,tot}$ the total number of hits of the reconstructed shower and Nhit$_{ini,tot}$ the total number of hits of the particle before reconstruction.

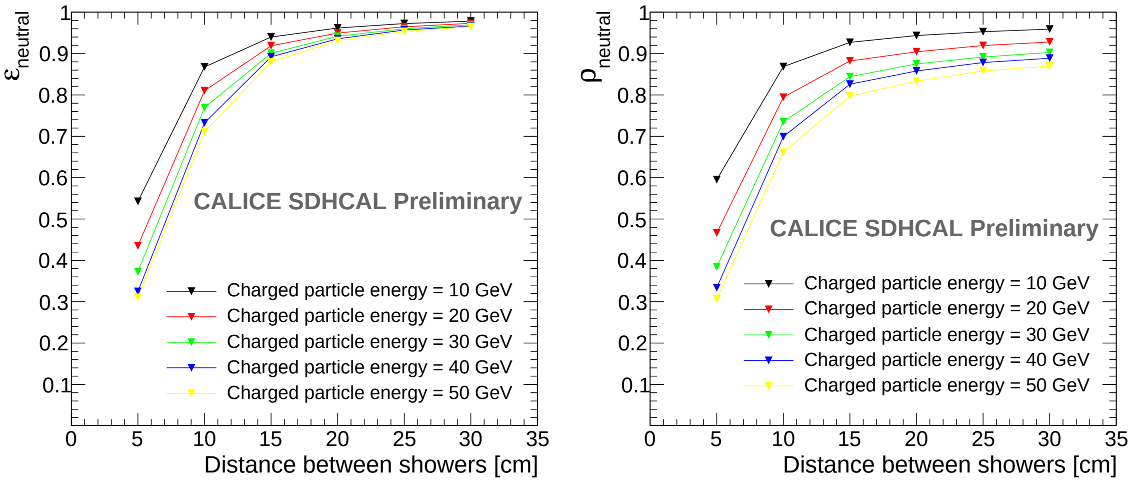

Figure 9: The efficiency (left) and purity (right) of the 10 GeV neutral hadron after reconstruction.

Figure 9 shows the efficiency (left) and the purity (right) of the neutral hadron for different charged particle energies and different separation distances. In the same way as for the mean number of PFOs, at small distances the two showers start to overlap and confusions appear in the reconstruction. Thus, some hits of the neutral hadron are assigned to the charged one (and vice versa) and the efficiency and purity decrease. At large separation distances, the purity does not tend to 100%. This is due to the last performed algorithm (small neutral fragment algorithm) which merges small neutral cluster fragments to their closest parent cluster, without considering the parent cluster size or energy. Since the number of neutral fragments for a single hadron particle increases with the energy, a non-negligible part of the charged hits is assigned to the neutral hadron, leading to a decrease of its purity.

Figure 10 (left) shows the fraction of events in which at least one neutral hadron has been reconstructed. As expected, the number of reconstructed neutral particles decreases with the separation distance. From 30 cm down to 15 cm, this fraction is stable and greater than 97%. At 10 cm, confusion becomes significant and the neutral hadron is sometimes merged with the charged one, leading to a small decrease of this fraction. At 5 cm, we can see that the fraction strongly depends on the charged particle energy and goes from 73% of reconstructed events for the 10 GeV charged particle case down to 60% at 50 GeV.

We define the reconstructed neutral energy as the sum of the neutral particle energies and the measured neutral energy as the estimated energy before reconstruction. Figure 10 (on the right) shows the mean difference between the reconstructed neutral energy and the measured neutral energy when at least one neutral hadron has been reconstructed. For the same reason as for the purity, the mean reconstructed neutral energy increases with the charged particle energy. This plot also shows a flat behavior of the reconstructed neutral energy with the separation distance. This means that the reconstruction of the

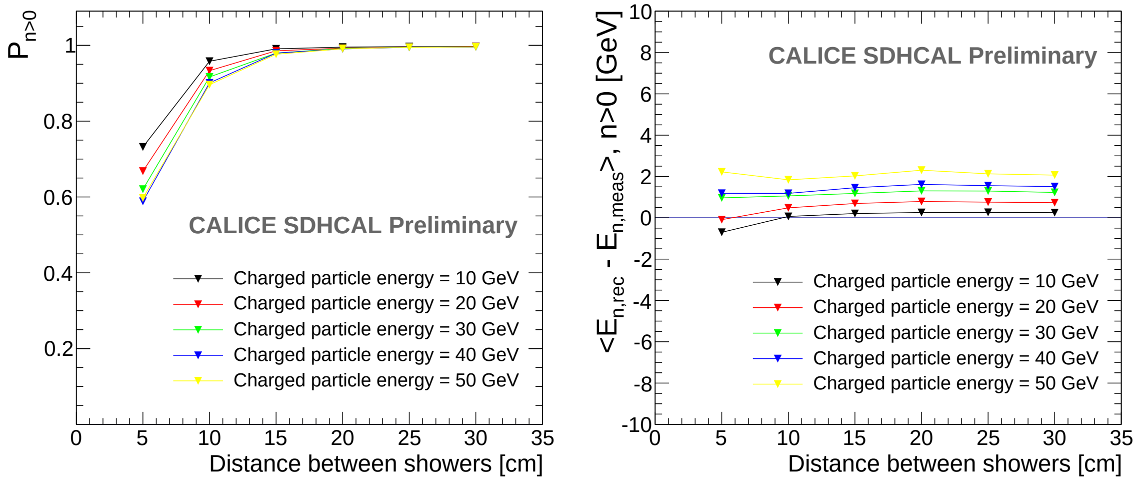

Figure 10: Left : The fraction of events where at least one neutral hadron has been reconstructed. Right : The mean difference between the reconstructed energy and the measured energy before reconstruction in events with at least one reconstructed neutral hadron.

neutral hadron at very small distance (5 cm) has a *binary-like* behaviour, either well reconstructed or completely merged with the charged hadron.

6 Summary

The ArborPFA, based on the tree-like structure of hadronic showers has been described.

A single particle study has been performed on SDHCAL using data taken at SPS H6 beam line at CERN [4]. Single hadron showers have been studied using the algorithm.

The results show a good efficiency with more than 95% of hits assigned to the reconstructed charged particle over the energy range of 10 to 80 GeV. The mean number of PFOs shows a linear increase with the energy from 1.1 PFOs at 10 GeV to 1.4 PFOs at 80 GeV. Around 4% of hits are not associated by the algorithm, leading to a small decrease in the clustered energy response and a small degradation of the energy resolution with respect to the case in which all hits are considered.

The ability of the algorithm to separate nearby hadronic showers was also investigated. Two different charged hadron showers with different energies from the same test beam data set have been overlaid in the same event with different separation distances. For the 10 GeV pion shower, the track segment inside the calorimeter has been identified and removed from the event in order to emulate a neutral hadron particle. The results show an efficiency of a neutral hadron recovery higher than 90% down to 10 cm separation distance where a non negligible confusion starts to appear.

The difference between the reconstructed energy and the measured energy of the neutral hadron in the case where at least one neutral hadron has been reconstructed shows an increase with the charged hadron energy for all the separation distances due to the small neutral fragment merging algorithm. For all separation distances, the difference stays constant and shows that, at small separation distances, the neutral hadron reconstruction has a binary-like behaviour, either a very good reconstruction or merged in the charged hadron.

This work will be extended shortly to include the electromagnetic calorimeter as well as the other sub-detectors in the framework of the ILD detector with the aim to separate charged and neutral hadrons produced in jets to study the possible improvement on PFA performances using this algorithm.

References

[1] J. Carwardine *et al.*, *International Linear Collider Technical Design Report.* 1) Executive Summary, 2) Physics, 3) Accelerator, 4) Detectors. 12 June 2013

[2] M. A. Thomson, *Particle Flow Calorimetry and the PandoraPFA Algorithm*, `phys.int-det/0907.3577`

[3] The CALICE Collaboration, *Tests of a particle flow algorithm with CALICE test beam data*, `phys.int-det/1105.3417`

[4] The Calice Collaboration, *Construction and commissioning of a technological prototype of a high-granularity semi-digital hadronic calorimeter*, `phys.int-det/1506.05316`

[5] M. Ruan, *Arbor, a new approach of the Particle Flow Algorithm*, Proceeding of CHEF 2013. `hep-ex/1403.4784`

[6] J. S. Marshall, M. A. Thomson, *The Pandora Software Development Kit for Pattern Recognition*, `phys.int-det/1506.05348`

[7] F. Gaede, *Marlin and LCCD: Software tools for the ILC*, Nucl. Instrum. Meth. **A559** (2006) 177-180

[8] ILCsoft, 2012. `http://ilcsoft.desy.de/portal`

[9] O. Lobban, A. Sriharan, R. Wigmans, *On the energy measurement of hadron jets*, Nucl. Instrum. Meth. **A495** (2002) 107-120

A ArborPFA algorithm

Before describing the algorithm in detail, a few definitions specific to ArborPFA need to be introduced :

Object An *object* is a calorimeter hit or a group of contiguous calorimeter hits within a layer that serves as a vertex for the ArborPFA algorithm. This was introduced for two reasons i) to provide a generalization of connections between *objects* without making any assumptions of what is contained in an *object*, ii) to overcome the pad multiplicity[2] inherent in the SDHCAL [4] and similar detectors.

Flow direction The flow direction is of two kinds : forward direction which is from upstream to downstream of the beam direction and backward direction for the opposite.

Connector A connector is a link between two *objects*. It has a weight and a direction.

Connector depth The connector depth is defined as the number of intermediary connectors linking two different objects.

Tree A tree is a set of *objects* connected in a tree topology, which means that for each object there is only one backward connector. An *object* without a backward connector is called a seed and an *object* without a forward connector is called a leaf.

Cluster A cluster is a set of trees.

Particle flow object (PFO) A particle flow object is a set of clusters and tracks [3], which corresponds to a reconstructed particle.

Note that in the following algorithm descriptions, some parameters are labelled by a name or symbol, whose values are given in a separate Appendix B.

[2] More than one pad could be fired when a particle crosses the gas gap.

[3] By *track* we mean one reconstructed by a tracking detector such as a TPC

A.1 Pre-clustering phase

Before building trees, we need to create objects to connect with each other.

Object creation When a particle goes through the detector, several pads can be fired in a single layer, leading to a multiplicity greater than 1. To overcome this problem, intra-layer groups of hits are assembled using the nearest neighbour clustering algorithm (corner neighbours not included). The result is shown on Figure 11 on the left. If a group contains more than 4 hits, it is split and each individual hit is considered as a separate object. This generally happens in the shower core. If a group contains 4 or less hits, it is used to define a single object. This is the case for a mip or more generally any isolated non-showering particle. Figure 11 on the right shows the output of this algorithm within a single layer with encircled hits forming objects.

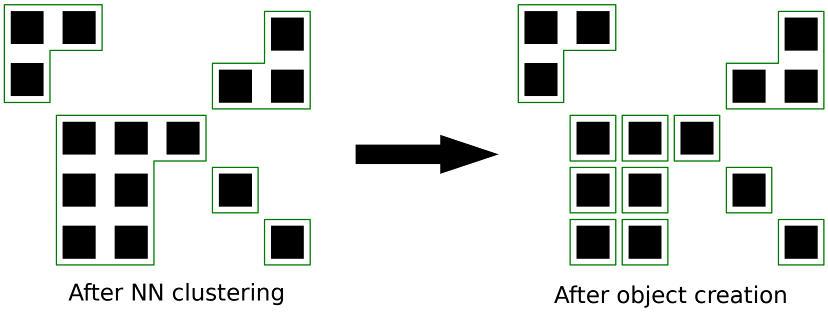

Figure 11: Schematic view of the object creation output within a single layer. Small groups of contiguous calorimeter hits are grouped together (encircled) in objects.

Track segment[4] candidate tagging In order to correctly reconstruct the primary track segment in the calorimeter, track segment candidate *objects* are identified and tagged for future treatment. For each object, we count the number of objects in the same layer within a distance of Δ_{mip}. If this number doesn't exceed $N_{obj,cut}$, the object is tagged as a track segment candidate object.

A.2 The main clustering phase - Connectors and trees

The main clustering algorithm consists of an iterative procedure using dedicated algorithms to create and remove connectors (connector loop). At the end of this step, all objects are arranged in a tree structure, which means that each object has at most one connector in the backward direction and 0 or more in the forward direction.

[4] By *track segment* we mean a track produced by a charged particle in the calorimeter such as Minimum Ionizing Particles (MIP)

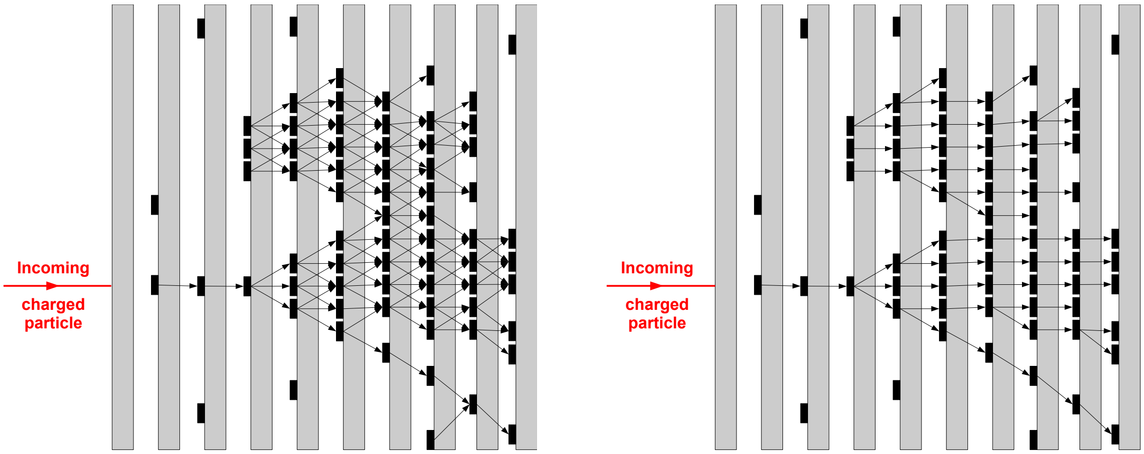

Figure 12: Schematic view of a neutral and a charged pion shower after the first connector seeding algorithm (left) and cleaning algorithm (right).

In the current implementation, the connector loop contains the following algorithms :

Primary track connection This algorithm aims to create connections between objects belonging to the primary track segment of charged particles in the calorimeter. It consists mainly of creating a sub-list of objects that are candidates for the primary track segments by using the objects tagged as track segment candidates and the track extrapolations on the front of the calorimeter. Once this list is built, the "*connector seeding 1*" algorithm and the "*connector cleaning 1*" algorithm are run on only the sub-list objects.

Connector seeding 1 We start by creating connections in the neighbourhood of each object. For each object, we look for other objects in the N$_{layers}$ next layers within a maximum distance Δ_{max} and we create connections between them. As an example, Figure 12 (left) illustrates the output of this algorithm.

Connector cleaning 1 Once connectors are created, we need to build a tree structure by keeping only one connector in the backward direction for each object. We define the reference direction of an object as :

$$\vec{C}_{ref} = w_{bck}.\sum_{\sigma}\sum_{b}\vec{c}_{b,\sigma} - w_{fwd}.\sum_{\delta}\sum_{f}\vec{c}_{f,\delta} \tag{6}$$

where :

- w_{bck} (w_{fwd}) is a global positive weight assigned to backward (forward) connectors
- $\vec{c}_{b,\sigma}$ ($\vec{c}_{f,\delta}$) is the direction of a backward (forward) connector at the connector depth σ (δ) from the considered object

The depth parameter σ has been fixed to 1 in all algorithms. The reference direction is a vector that goes in the backward direction and indicates the most probable direction for a unique backward connection. Then we need to assign which backward connector should be kept for the tree building. Thus, for each backward connector of an object, we define the κ parameter as :

$$\kappa \; = \; \left(\frac{\theta}{\pi}\right)^{p_\theta} . \left(\frac{\Delta}{\Delta_{max}}\right)^{p_\Delta} \tag{7}$$

where :

- θ is the angle between a backward connector and the reference direction of the considered object,
- Δ is the distance between the connected objects,
- p_θ (p_Δ) is a power parameter for the normalized angle (distance)

The κ parameter quantifies the alignment with the reference direction within the range [0,1]. Smaller is this parameter, higher the alignment will be. The power parameters p_θ and p_Δ are to be tuned depending on which variable we want to emphasize.

The chosen backward connector for the tree building is the one with the smallest κ parameter; all others are removed from the list. The removal of connectors is done at the end of the algorithm so that all connectors contribute to the evaluation of the reference direction.

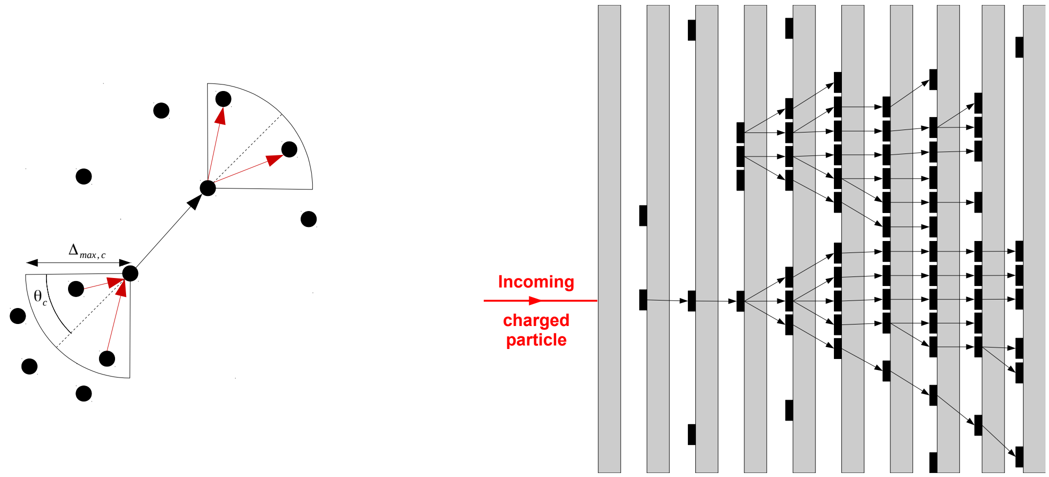

Figure 13: Left : Schematic view of the connector alignment procedure. In black, a considered connector and in red possible new connectors in backward and forward directions. Right : a neutral and a charged pion showers after the second connector cleaning algorithm.

Connector seeding 2 This second step of connector seeding starts from the tree structure obtained after the first connector cleaning algorithm. The goal of this second step is to create an alignment of connectors within the shower. For each connector, more forward connectors are created from the forward object of this connector by looking in a cone of half-angle θ_c and a maximum distance of $\Delta_{max,c}$. In the same way, additional backward connectors are created from the backward object of this connector. A schematic view of this step is shown in figure 13.

Connector cleaning 2 Here, we again need to clean up the backward connector list to end up with only one connector per object. This last algorithm is similar to the first connector cleaning except that the cleaning is done layer per layer. We start by the downstream layer and we perform the same cleaning procedure as described in the first connector cleaning within a single layer. Once the cleaning is done, the removal of unneeded connectors is done. By removing these connectors layer by layer, the computation of the reference vector will take into account only the cleaned connectors in the downstream direction. The depth parameter δ is also chosen to strictly higher than one in order to accentuate the alignment with the upstream connectors. We end up then with a tree structure again.

Tree building This step is straight-forward. Seed objects are identified and trees are built by recursively following the forward connected objects.

The following algorithms associate some of the trees with each other.

A.3 Association algorithms

Energy driven track cluster association The track to cluster association is performed using three different pieces of information, the cluster energy, the track momentum and the cluster topology. We first look at the track projection on the calorimeter front face. Two different cases may occur :

- the particle has interacted before the calorimeter or in the first layer. In this case, many seed objects are found in the N$_{layer}$ first layers at a maximum distance of $\Delta_{track-cluster_1}$ of the track projection. Seed objects are then sorted by their distance to the track projection. The clusters associated to their seeds are then associated to the track progressively starting from the closest one until the difference between the track momentum and the total cluster energy is minimized. The clusters are then merged since they belong to the same cluster structure.
- the particle produced a track segment at least in the N$_{layer}$ first layers and a seed object is found within a distance $\Delta_{(track-cluster)_1}$ to the track projection. Since only a cluster starting with a track segment has to be associated, an additional distance

cut $\Delta_{(track-cluster)_2}$ between seed objects and the track projection is applied. This decreases the confusion for small separation distances between nearby particles. The same track-to-cluster association and cluster merging is then performed as above.

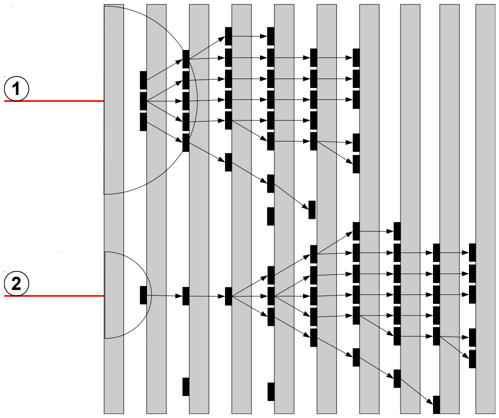

Figure 14: Schematic view of the energy driven track cluster algorithm.

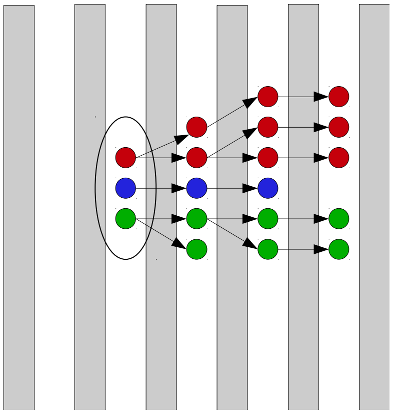

Figure 15: Schematic view of the neutral tree merging algorithm.

Figure 14 shows a schematic view of the two different scenarios. The upper one corresponds to the case where an early interaction is found, and the lower one where a primary track segment of a cluster is found.

Neutral tree merging This algorithm is designed for neutral particle interactions for which the first interacting layer contains a few seeds. Figure 15 shows a configuration in which three trees have been built (with three colours) for one neutral particle interaction. We can see that the seeds in the first interacting layer all belong to the same cluster. The trees having seeds closer than Δ_{seed} and positioned in the same layer are merged.

Pointing cluster association This step aims at associating neutral fragments (daughter cluster) to other fragments which may be charged or neutral clusters (parent clusters). We start by identifying the clusters that have at least $\mathrm{N}_{objects}$ objects in at least N_{layer} contiguous layers. The selected cluster could be either a parent or a daughter cluster. Then we proceed as follows :

1. A linear 3D straight line fit is performed over the position of all the hits of each cluster (without weights). This defines the axis of each cluster.

2. The clusters are sorted by their most downstream layers (most downstream hit in the cluster) l_{inner}.

3. Starting from the most downstream cluster i, we look for a parent cluster j for which it propagates further downstream ($\mathrm{l}_{inner,i} > \mathrm{l}_{inner,j}$)

4. Among these candidate parent clusters, we look for those for which $\mathrm{d}_{proj} < \mathrm{d}_{proj,cut}$ and $\theta_{i,j} < \theta_{i,j,cut}$ where :

    - d_{proj} is the distance between the candidate daughter cluster axis and the candidate parent cluster barycentre (line-to-point distance)
    - $\theta_{i,j}$ is the angle between the axis of the two clusters

    and we choose the cluster for which d_{proj} is minimal.

5. Among this same list of candidate parent clusters, we look for those satisfying the condition $\mathrm{d}_{cross} < \mathrm{d}_{cross,cut}$ and $\mathrm{d}_{closest} < \mathrm{d}_{closest,cut}$ where :

    - d_{cross} is the distance at closest approach (d.c.a) between the two cluster axes
    - $\mathrm{d}_{closest,i,j}$ is the closest distance between an object of the parent cluster and the point of closest approach of the cluster axes (distance at closest approach)

    and we choose the cluster for which d_{cross} is minimal.

6. We choose the best candidate parent cluster among the two previous methods above. Many cases may happen i) no parent cluster is found, then no parent cluster is assigned to this daughter cluster, ii) one of the two methods has found a parent cluster or the two methods provide the same parent, then we assign it to the daughter cluster, iii) the two methods have found a parent cluster but they are not the same one. In this case the closest candidate parent cluster among the two in terms of barycentre distance is assigned to the daughter cluster.

7. If no parent cluster has been found for a cluster, nothing is done.

8. If the parent cluster has no associated track, merge the two clusters, otherwise we define the variable Ψ as :

    $$\Psi = \left|\frac{p - E_{tot}}{f_{res}.\sigma_E.p}\right| \tag{8}$$

    where :

    - p is the track momentum of the parent cluster
    - E_{tot} is the total energy estimated from the combined hit list of the parent and daughter clusters
    - σ_E is the calorimeter energy resolution at the track momentum p
    - f_{res} is a multiplicative factor[5]

    We check then that the Ψ defined for the parent and daughter clusters is less than Ψ_{cut} and if the difference between p and E_{tot} after the cluster merging (parent + daughter clusters) decreases. The two clusters are merged if the previous conditions are satisfied.

[5] The parameter f_{res} is used to reduce or enlarge the accepted range of the difference p-E_{tot}. A higher value of this parameter will accept a merging with a higher difference p-E_{tot}.

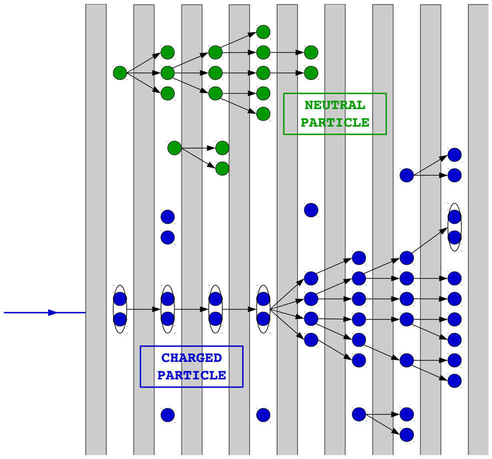

Figure 16: Schematic view of the final ArborPFA output.

Small neutral fragment merging At this stage, the main part of the shower of each particles has been identified. Only isolated objects and small tree structures that surround the showers are not associated. First, these small structures are identified if their size is less than N_{cut} objects. Then every small structure is merged within a shower, the closest in terms of barycentre distances.

Particle flow object creation Particle flow objects are built from the produced clusters after all the steps described above (Figure 16). Charged PFOs are built from clusters that have an associated track, while other clusters are considered as neutral PFOs.

B ArborPFA algorithm parameters

Object creation algorithm

Parameter name	value
MaxClusterSize	4
IntraLayerDistance	11 mm

- MaxClusterSize
  → The maximum intra layer cluster size to build an object with. Else the object is split in single calo hit objects
- IntraLayerDistance
  → The nearest neighbour intra layer clustering maximum distance

Track segment candidate tagging algorithm

Parameter name	value
MaxNNeighbors	6
IntraLayerNeighbourDistance (Δ_{mip})	50 mm

- MaxNNeighbors
  → The maximum number of neighbouring objects within a layer
- IntraLayerNeighbourDistance (Δ_{mip})
  → The maximum distance between two neighbours in a layer used for the neighbour counting

Primary track connection

Parameter name	value
ConnectionDistance	110 mm
BackwardConnectorWeight	2
ForwardConnectorWeight	3
OrderParameterAnglePower	1
OrderParameterDistancePower	5
MaxNEmptyConsecutiveLayers	3

- ConnectionDistance
  → The maximum connection distance used for the primary track connectors creation
- BackwardConnectorWeight (w_{bck})
  → The backward connector weight assigned for the reference vector computation
- ForwardConnectorWeight (w_{fwd})
  → The forward connector weight assigned for the reference vector computation
- OrderParameterAnglePower
  → The angle power parameter of the κ parameter while cleaning connectors
- OrderParameterDistancePower
  → The distance power parameter of the κ parameter while cleaning connectors
- MaxNEmptyConsecutiveLayers
  → The maximum consecutive empty layers to take into account for the connector seeding

Connector seeding 1

Parameter name	value
ConnectionDistance	45 mm

- ConnectionDistance
  → The maximum connection distance used for a connector creation

Connector cleaning 1

Parameter name	value
BackwardConnectorWeight	2
ForwardConnectorWeight	2
OrderParameterAnglePower	1
OrderParameterDistancePower	5
ReferenceDirectionDepth	1

- BackwardConnectorWeight
  → The weight of a backward connector assigned in the reference direction vector calculation.

- ForwardConnectorWeight
  → The weight of a forward connector assigned in the reference direction vector calculation.

- OrderParameterAnglePower
  → The θ angle power parameter used for the κ parameter computation

- OrderParameterDistancePower
  → The Δ distance power parameter used for the κ parameter computation

- ReferenceDirectionDepth
  → The forward connector depth used for the reference vector computation

Connector seeding 2

Parameter name	value
ConnectionDistance	65 mm
ConnectionAngle	0.7 rad

- ConnectionDistance
  → The maximum connection distance used for a connector creation

- ConnectionAngle
  → The maximum angle between two connectors

Connector cleaning 2

Parameter name	value
BackwardConnectorWeight	0.1
ForwardConnectorWeight	5
OrderParameterAnglePower	1
OrderParameterDistancePower	5
ReferenceDirectionDepth	2

- BackwardConnectorWeight
  → The weight of a backward connector assigned in the reference direction vector calculation.

- ForwardConnectorWeight
  → The weight of a forward connector assigned in the reference direction vector calculation.

- OrderParameterAnglePower
  → The θ angle power parameter used for the κ parameter computation

- OrderParameterDistancePower
  → The Δ distance power parameter used for the κ parameter computation

- ReferenceDirectionDepth
  → The forward connector depth used for the reference vector computation

Energy driven track cluster association

Parameter name	value
TrackToClusterDistanceCut1	75 mm
TrackToClusterDistanceCut2	55 mm
FirstInteractingLayerNSeedCut	15
TrackToClusterNLayersCut	3
TrackClusterPsi2Cut	3
Psi2SigmaFactor	1.5

- TrackToClusterDistanceCut1
  → The maximum distance between the track projection at calorimeter front face and a cluster seed. This distance is used to detect an early interacting cluster.

- TrackToClusterDistanceCut2
  → The reduced maximum distance between the track projection at calorimeter front face and a cluster seed. This distance is used when no early interacting cluster has been detected.

- FirstInteractingLayerNSeedCut
→ The cut on the number of cluster seeds found within a the distance TrackToClusterDistanceCut1 to detect an early cluster interaction.

- TrackToClusterNLayersCut
→ The number of inner layers to look for cluster seeds to associate.

- TrackClusterPsi2Cut
→ The ψ^2 cut applied while associating clusters to a track.

- Psi2SigmaFactor
→ The f_{res} factor on denominator used to compute the ψ^2 for track-to-cluster compatibility (see equation 8)

Neutral tree merging

Parameter name	value
SeedSeparationMerge (Δ_{seed})	25 mm

- SeedSeparationMerge (Δ_{seed})
→ The maximum distance between two cluster seeds within a layer to perform a cluster merging

Pointing cluster association

Parameter name	value
MinNObjects	4
MinNLayers	4
FitToBarycentreDistanceCut	30 mm
FitToBarycentreAngleCut	$\frac{\pi}{6}$ rad
FitToFitDistanceCut	20 mm
FitDistanceApproachCut	20 mm
Chi2NSigmaFactor	1.5
Chi2AssociationCut	1

- MinNObjects
→ The minimum number of objects within a cluster in order to to be candidate for the pointing cluster association

- MinNLayers
→ The minimum number of layers within a cluster (outermost - innermost + 1) in order to be candidate for the pointing cluster association

- FitToBarycentreDistanceCut
  → The cut applied on the distance between the daughter cluster fit and the parent cluster barycentre position (point-to-line distance)

- FitToBarycentreAngleCut
  → The cut applied on the angle between the daughter and parent cluster fits

- FitToFitDistanceCut
  → The cut applied on the distance between the daughter and parent cluster fits (line-to-line distance)

- FitDistanceApproachCut
  → The cut applied on the closest distance between a parent cluster object and the daughter cluster crossing point at the parent and daughter cluster fit closest approach.

- Chi2NSigmaFactor
  → The N_{res} factor on denominator used to compute the χ^2 for track-to-cluster compatibility (see equation 8) using the merged cluster (daughter + parent)

- Chi2AssociationCut
  → The χ^2 cut applied on the merged clusters compatibility with a track when associating a neutral daughter cluster with a charged parent cluster.

Small neutral fragment merging

Parameter name	value
MaximumDaughterNObject	20
LargeDistanceCut	1000 mm

- MaximumDaughterNObject
  → The maximum number of objects to consider the cluster as a small neutral fragment to merge it into a bigger parent cluster

- LargeDistanceCut
  → The maximum distance between a small neutral fragment and a potential parent cluster