The Pandora Software Development Kit for Particle Flow Calorimetry

J. S. Marshall, M. A. Thomson
Cavendish Laboratory, University of Cambridge, J. J. Thomson Avenue, Cambridge, CB3 0HE, United Kingdom.
E-mail: marshall@hep.phy.cam.ac.uk

Abstract. Pandora is a robust and efficient framework for developing and running pattern-recognition algorithms. It was designed to perform particle flow calorimetry, which requires many complex pattern-recognition techniques to reconstruct the paths of individual particles through fine granularity detectors. The Pandora C++ software development kit (SDK) consists of a single library and a number of carefully designed application programming interfaces (APIs). A client application can use the Pandora APIs to pass details of tracks and hits/cells to the Pandora framework, which then creates and manages named lists of self-describing objects. These objects can be accessed by Pandora algorithms, which perform the pattern-recognition reconstruction. Development with the Pandora SDK promotes the creation of small, re-usable algorithms containing just the kernel of a specific operation. The algorithms are configured via XML and can be nested to perform complex reconstruction tasks. As the algorithms only access the Pandora objects in a controlled manner, via the APIs, the framework can perform most book-keeping and memory-management operations. The Pandora SDK has been fully exploited in the implementation of PandoraPFA, which uses over 60 algorithms to provide the state of the art in particle flow calorimetry for ILC and CLIC.

1. Particle Flow Calorimetry
At a future high-energy lepton collider, such as the ILC[1] or CLIC[2], many of the interesting physics processes will produce final states containing multiple jets, which may be accompanied by charged leptons and/or missing transverse momentum. In order to perform precision physics measurements in these conditions, it is vital to be able to reconstruct the invariant masses of the jets; accurate jet mass measurements would prove to be a powerful tool for both reconstruction and identification of physics events. The goal for jet energy resolution at the ILC or CLIC is that it should be sufficient to allow separation of the hadronic decays of $W$ and $Z$ bosons via the reconstruction of di-jet invariant masses. This sets a challenging jet energy resolution goal of $3.5\%/\sqrt{E/GeV}$ for 50 – 500 GeV jets, which is unlikely to be achievable with a traditional approach to calorimetry[3].

Measurements of jet fragmentation at LEP provide detailed information about the particle composition of jets[4, 5]. In a typical jet, approximately 62% of the energy is carried by charged particles (mainly hadrons), whilst 27% is carried by photons, 10% by long-lived neutral hadrons and 1.5% by neutrinos. In a traditional approach to calorimetry, the jet energy would be obtained from the energies deposited in the electromagnetic and hadronic calorimeters (ECAL and HCAL respectively). This means that 72% of the energy of a typical jet would be measured
with a precision limited by the relatively poor HCAL resolution of \( > 55\% / \sqrt{E/\text{GeV}} \).

The particle flow approach to calorimetry aims to improve jet energy measurements by reconstructing the four-vectors of all visible particles in an event. The reconstructed jet energy is then the sum of the energies of the individual particles in the jet. As illustrated in Figure 1, this approach requires fine-granularity calorimeters and sophisticated software algorithms to trace the individual paths of particles through the detector. The energy and momentum for each particle can then be extracted from the detector subsystem in which we expect these measurements to be most accurate. Charged particle momenta are measured in the inner detector tracker, whilst photon energy measurements are extracted from the energy deposited in the ECAL, with typical resolution \( < 20\% / \sqrt{E/\text{GeV}} \). The HCAL is used to measure only the 10% of jet energy carried by long-lived neutral hadrons. Particle flow calorimetry can therefore offer a significant improvement to jet energy measurements, compared to traditional calorimetry.

Figure 1. The transition from traditional calorimetry, reconstructing jet energies purely from the energy deposited in the calorimeters, to particle flow calorimetry, which aims to reconstruct the four-vectors of all visible particles in an event.

2. Realising Particle Flow

Particle flow calorimetry requires the energy depositions from individual particles to be traced through the detector and cleanly distinguished from the depositions of other particles. As illustrated in Figure 2, there are a number of sources of possible confusion in identifying the paths of individual particles which can degrade jet energy measurements. It is the confusion, rather than the calorimetric performance, which is the limiting factor for particle flow calorimetry.

- Failure to resolve neutral particles (e.g. photons or neutral hadrons) from nearby charged particles will result in loss of energy. The neutral particle energy deposits are absorbed into the charged particle. The charged particle energy is then extracted only from measurements in the inner detector tracker.
- Failure to associate calorimeter energy deposits from a charged particle with the inner detector track created by the same particle will lead to double-counting of energy. The calorimeter energy deposits will be used to create a fake neutral particle, whilst the track will still be used to create the true charged particle.

In order to benefit from the improved jet energy resolution that particle flow calorimetry can offer, confusion must be reduced to the lowest possible levels. This requires fine-granularity detectors with accurate inner detector tracking, and calorimeters that can longitudinally separate electromagnetic and hadronic showers. The ECAL must therefore have a large ratio of nuclear interaction length to radiation length, with high longitudinal granularity to cleanly identify electromagnetic showers. The ECAL must also have a small Molière radius, to minimise the transverse spread of electromagnetic showers, and it must have transverse granularity of the order of the Molière radius. The HCAL must offer longitudinal and transverse segmentation,
Figure 2. Possible sources of confusion in a particle flow reconstruction, which can lead to energy loss or double counting energy deposits.

and it must have a small nuclear interaction length, in order to fully contain hadronic showers. One detector concept satisfying all of the above criteria, and which was designed specifically for particle flow calorimetry is ILD[6], illustrated in Figure 3.

Figure 4 shows the typical topology of a 250 GeV jet in ILD, with labels identifying a number of the constituent particles. The Figure shows inner detector tracks, representing the paths of charged particles in the Time Projection Chamber (TPC). These tracks can be extrapolated by eye and associated with clusters of calorimeter energy deposits in the fine granularity ECAL and HCAL. Photons produce energy deposits with characteristic longitudinal and transverse profiles in the ECAL and can be cleanly resolved by eye, due to their small transverse spread. HCAL clusters that cannot be associated with TPC tracks represent neutral hadrons. The fact that it is possible to reconstruct such particles so easily by eye suggests that the detector is indeed suitable for use with particle flow calorimetry. The challenge is then to develop software algorithms to automate the reconstruction.

Particle flow calorimetry demands high performance software; the final jet energy resolution depends strongly on both the detector granularity and the quality of the software algorithms that implement the particle flow reconstruction. The algorithms must be able to exploit the granularity to merge together energy deposits from individual particles, whilst making very few mistakes. The most sensible approach is to implement a large number of ‘decoupled’ pattern-recognition algorithms, each of which looks to identify and reconstruct specific particle topologies, whilst avoiding contributing to the confusion. The need to implement a large number of efficient pattern-recognition algorithms then motivates the need for a central software

Figure 3. (a). Cross-section of the top quadrant of ILD, as modified for proposed use at CLIC. (b). Reconstructed particles in a simulated $e^+e^- \rightarrow H^+H^- \rightarrow t\bar{b}b\bar{t}$ event in the CLIC-modified version of ILD. Source [2].
Figure 4. Typical topology of a 250 GeV jet in ILD. The granularity of the detector allows individual particles to be resolved by eye; the challenge is then to develop software algorithms to automate the reconstruction.

framework, which can help with memory-management and book-keeping issues. Such a framework would help to keep each algorithm simple and focused on its specific pattern-recognition tasks. Ideally the framework should be flexible and reusable, allowing operation with the numerous different software frameworks and detector concepts currently in existence for future high-energy lepton colliders.

3. Pandora SDK for Particle Flow

The Pandora Software Development Kit (SDK) for particle flow calorimetry is a robust and efficient framework for developing and running algorithms for particle flow reconstruction. It consists of a single framework library and a number of carefully-designed Application Programming Interfaces (APIs). It was designed with the twin aims of simplifying the development of efficient pattern-recognition algorithms and allowing easy application of existing algorithms to different detectors, different software environments or even different pattern-recognition problems. The software design means that the pattern recognition reconstruction can be divided into three distinct sections, which communicate via the Pandora APIs. This is illustrated in Figures 5 and 6.

A Pandora client application can use the APIs to pass details of the input tracks and calorimeter cells, which form the building blocks for the particle flow reconstruction problem. The client application runs in the user’s software framework and is responsible for isolating the Pandora framework and algorithms from the specific details of the user’s software and detector design. For each inner detector track and calorimeter cell, the client application must populate a Pandora API parameters instance. The parameters that must be specified are physics-based and are designed to enable the object to become ‘self-describing’. Examples of required parameters for inner detector tracks are impact parameters $Z_0$, $d_0$ and track states (position and momentum) at the start and end of the track, together with a projection to the front face of the calorimeter. For calorimeter cells, typical parameters are cell dimensions, positions and calibrated energy values. Each populated parameters instance is passed to the Pandora framework, via the CreateObject API. The full list of parameters must be specified for each object: failure to provide a single parameter will cause an exception to be thrown.

For each parameters instance passed to the Pandora framework, a Pandora manager instance will create a new self-describing Pandora track or Pandora calorimeter cell. There is a manager
for each of the objects used during the pattern-recognition: tracks, calorimeter cells, clusters and particle flow objects. Each manager class (the manager class is templated on the object type) is responsible for owning and manipulating instances of their respective objects, and for organising these objects into named lists. The managers offer simple, efficient and well-tested methods for performing a comprehensive set of simple operations: creating new objects, creating new object lists, moving objects between lists, merging lists, deleting lists, etc. These are low-level operations, which need not repeatedly feature in the actual algorithms that perform the pattern-recognition. Instead, higher level functions are created, which are simple combinations of the basic object-manipulation operations. These higher-level operations, which an algorithm will require, are cleanly wrapped up by a set of APIs to be used by the Pandora algorithms. The managers are extremely efficient, relying only on STL containers for their operations.

Figure 5. Pandora client applications can use Application Programming Interfaces (APIs) to pass details of inner detector tracks and calorimeter cells to the Pandora framework. The framework then manages named lists of self-describing tracks, calorimeter cells, clusters and particle flow objects, performing all memory-management and book-keeping operations.

As soon as parameters for all the tracks and calorimeter cells in an event have been passed to the Pandora framework, the client application can call the ProcessEvent API and the thread will be passed to the framework. The framework will simply call all of the Pandora algorithms configured to run this event, then return control to the client application. It is the algorithms which must perform the pattern-recognition reconstruction, and use the tracks and calorimeter cells to build clusters, then particle flow objects. The algorithms access and manipulate the Pandora objects by using the APIs to request services from the Pandora framework. For instance, a simple algorithm could ask for a named list of calorimeter cells (or just ask for the ‘current’ list of calorimeter cells), then start to group the cells together. The algorithm could ask to create a new cluster list, create new clusters, add calorimeter cells to the clusters and then, finally, to save the new cluster list under a specified name. Subsequent algorithms could ask for the cluster list, investigate which clusters point towards each other and ask to merge these clusters.

The strategy of allowing changes to the reconstruction objects to be made only via requests to the framework means that it is possible for the framework to monitor all the relevant changes and to perform all memory-management and book-keeping operations (e.g. which calorimeter cells
Pandora algorithms can use APIs to request high-level services from the Pandora framework. Typical requests would be to access or modify the objects, for instance creating or merging clusters. Allowing changes to these objects to be made only via requests to the framework keeps the algorithms simple and efficient and reduces the potential for errors. It also ensures that the algorithms can remain ‘clean’ and focused on the specific reconstruction task; the algorithm implementation can contain just the kernel of the specific pattern-recognition operations. This promotes the reuse of algorithms: parent algorithms can run daughter algorithms and, by altering the designated current object lists, allow the same algorithm code (possibly with a different configuration) to process e.g. objects in different regions of the detector. The algorithms are completely isolated from the user’s software framework and, in principle, need have no dependencies, as the Pandora objects are self-describing and can be queried to obtain a rich variety of useful properties.

Pandora algorithms are XML configured; each algorithm is registered with the Pandora framework, under a specified name, by the client application. The framework will then parse a Pandora settings XML file before event processing begins. This XML file should contain an ordered list of each of the algorithms to be run for each event. If an algorithm is designed to run a number of daughter algorithms, the configuration for the daughter algorithms should be nested within the parent algorithm XML. Every algorithm parameter is configurable, and an XML key can be specified within the relevant algorithm XML block to override any hard-coded default parameter values. Algorithms can be aided in their reconstruction via a number of re-usable helper classes or ‘algorithm tools’. These are again registered via the client application, at start-up, and configured via the Pandora settings XML file. Pandora interface classes are provided to allow a user to provide custom implementations of code to perform energy correction routines, particle identification operations, or to divide calorimeter cells into ‘pseudolayers’, which could e.g. increase with the number of integrated radiation lengths from the interaction point.

Pattern-recognition operations with Pandora can therefore be highly customised; Pandora is truly a Software Development Kit, rather than just a set of algorithms with some shared functionality. The client application is responsible for registering all of the possible algorithms and tools. The Pandora settings XML file is then responsible for specifying which of the algorithms and tools are used in order to process an event. The algorithms and tools that are registered could be compiled as part of any external library and could have any number of external dependencies. However, this need not be the case, and a large library of over 60
algorithms is provided, without any dependencies, for use with fine granularity detectors. The algorithms and tools in this ‘FineGranularityContent’ library can be registered via a single function call and are designed to be very generic and reusable: they are designed to work with just the Pandora object properties, i.e. tracks with specified track states and calorimeter cells that are simply space-points, with cell dimensions and energy values. These algorithms can be quickly applied to new pattern-recognition problems, with the creation of a short client application (from as little as 70 lines of C++). A smaller library of algorithms and tools relevant to detectors with more coarse granularity is also available.

The algorithms triggered by the Pandora settings XML file will typically be capable of gradually building particle flow objects from the input tracks and calorimeter cells. When the final algorithm has finished, for the current event, control will be passed back to the client application. The client application can call the GetParticleFlowObjects API to receive the final list of reconstructed particles. These objects are owned by the Pandora framework, and contain lists of the constituent Pandora tracks and Pandora clusters. However, mapping functionality is provided to allow the user to link from these objects back to the tracks and calorimeter cells in the user software framework. The user can then complete the reconstruction by writing out the objects in the relevant framework, before calling the Reset API to allow specification of the parameters for the next event to begin.

4. Pandora Algorithms
The reconstruction of events in detectors such as ILD uses over 60 different Pandora algorithms and tools. These algorithms are well-understood and extremely efficient, helped by the fact that the memory-management operations are all provided by the highly-optimised Pandora framework. The basic reconstruction operations performed by the default set of Pandora algorithms can be summarised as follows:

- Calorimeter cells are clustered using a simple cone-based clustering algorithm, which works from the innermost cells to the outermost cells in the detector. Clusters can be seeded by the projection of inner detector tracks to the front face of the calorimeter.
- The clustering algorithm is configured so as to split up the energy depositions from individual particles, rather than risk merging together the depositions from multiple particles at this early stage. The resulting fragments, or proto-clusters, are then carefully merged together by a series of algorithms that implement well-motivated topological rules.
- Calorimeter clusters are associated to inner detector tracks, by comparing the properties of the clusters (linear fits, helix fits, etc.) to the projected track states at the front face of the calorimeter.
- If the energy of a calorimeter cluster does not match the associated track momentum, the cluster can be split-up, or merged with nearby fragments, by a series of statistical reclustering algorithms. The calorimeter cells in the relevant clusters can be reclustered using a series of differently configured clustering algorithms in order to improve the track-cluster compatibility.
- Fragment-removal algorithms aim to remove neutral clusters that are really fragments of charged clusters, merging the relevant clusters together. The algorithms look for evidence of association between nearby clusters, evaluating the changes in track-cluster compatibility that would occur if the merge was made, together with other topological properties.
- Particle flow objects are formed. If a particle contains tracks and associated clusters, the particle properties are extracted from the tracks. For neutral particles, calorimeter information is used.
Particle identification algorithms flag the reconstructed particles with PDG codes, identifying charged leptons. Photon identification is considered throughout the reconstruction, but can be finalised at this stage.

The above algorithms are described in more detail in [3, 7]. The algorithms have been applied successfully to reconstruct jet energies in the ILD and SiD[8] models, both at the ILC and in their modified structures proposed for use at CLIC. The algorithms were used to perform particle flow reconstruction for all studies performed in the creation of the CLIC Conceptual Design Report: Physics and Detectors at CLIC[2].

5. Pandora Performance
To assess the performance of the Pandora particle flow reconstruction algorithms, fully simulated and reconstructed events in ILD were used. Single Z-like bosons were generated, with masses ranging from 91 GeV to 500 GeV (and up to 3 TeV for the modified version of ILD proposed for use at CLIC). The bosons decay at rest, to produce two mono-energetic jets, and the Pandora algorithms attempt to reconstruct all the visible particles in each event. In order to avoid a bias from jet reconstruction, no jet reconstruction is performed; instead, the full energy deposited in the detector $E_{jj}$ is analysed, i.e. the sum of the energies of the Pandora reconstructed particles. The resolution of the jet energy $E_j$ is then obtained by calculating the RMS$_{90}(E_{jj})$ and the mean$_{90}(E_{jj})$ from the data, then applying a factor of $\sqrt{2}$.

Figure 7(a) shows the total reconstructed energy, $E_{jj}$, for Z-like particles of mass 91 GeV, 200 GeV, 360 GeV and 500 GeV decaying at rest into two light quarks. Figure 7(b) shows the variation of the jet energy resolution as a function of the jet angle, for individual jet energies, $E_j$, of 100 GeV, 180 GeV and 250 GeV. The challenging jet energy resolution goal of $3.5%/\sqrt{E/\text{GeV}}$ is surpassed for these jet energies in all except the far forward region of the detector. Mean jet energy resolutions for the barrel region of the detector, $|\cos(\theta)| < 0.7$, are shown in Table 1.

![Figure 7](image-url)

**Figure 7.** (a). The total reconstructed energy for Z-like particles of mass 91 GeV, 200 GeV, 360 GeV and 500 GeV decaying at rest into two light quarks. (b). The variation of the jet energy resolution as a function of the jet angle, for jet energies of 100 GeV, 180 GeV and 250 GeV.

| Jet Energy  | 45.6 GeV | 100 GeV | 180 GeV | 250 GeV |
|-------------|----------|---------|---------|---------|
| RMS$_{90}(E_{jj})$/mean$_{90}(E_j)$ | 3.66 ± 0.05 | 2.86 ± 0.04 | 2.90 ± 0.04 | 3.02 ± 0.05 |

**Table 1.** Mean jet energy resolutions for the barrel region of ILD, $|\cos(\theta)| < 0.7$. 

Figure 8(a) shows a parameterisation of the particle flow energy resolution as a function of the jet energy. A separate curve displays the estimated contribution to the resolution from confusion, which increases with energy as the calorimeter occupancies increase and it becomes more difficult to resolve individual particles. A further curve shows the jet energy resolution obtained using a traditional calorimetric approach, using the total energy deposited in the calorimeters. The Pandora particle flow reconstruction provides a better energy resolution than the traditional calorimetric approach for the entire range of jet energies considered. Figure 8(b) refers back to the original motivation for particle flow calorimetry and accurate jet mass reconstruction: the desire to be able to separate the hadronic decays of W and Z bosons. The Figure shows the separation obtained by reconstructing the di-jet invariant masses for the processes $e^+e^- \rightarrow WW \rightarrow \mu\nu qq$ (where the muon is carefully removed from the event) and $e^+e^- \rightarrow ZZ \rightarrow \nu\nu qq$, for 500 GeV $W$ and $Z$ bosons in the ILD detector, as modified for proposed use at CLIC. The $W$ and $Z$ mass peaks can be cleanly resolved, with a 2.2$\sigma$ separation.

**Figure 8.** (a). Parameterisation of jet energy resolutions as a function of jet energy, for both particle flow and traditional calorimetry. Source [3]. (b). Separation of $W$ and $Z$ bosons achieved via di-jet invariant mass reconstruction in CLIC-modified version of ILD. Source [2].

**References**

[1] J. Brau (ed.) et al., *International Linear Collider Reference Design Report*, 2007, ILC-REPORT-2007-001.

[2] L. Linssen (ed.) et al., *Physics and Detectors at CLIC: CLIC Conceptual Design Report*, CERN-2012-03.

[3] M.A. Thomson, *Particle Flow Calorimetry and the PandoraPFA Algorithm*, Nucl.Instr. and Meth. A 611 (2009)

[4] I.G. Knowles, G.D. Lafferty, *Hadronization in Z0 decay*, J. Phys. G 23 (1997) 731.

[5] M.G. Green, S.L. Lloyd, P.N. Ratoff, D.R. Ward, *Electron-Positron Physics at the Z*, IoP Publishing, 1998.

[6] T. Abe et al., *The International Large Detector: Letter of Intent*, 2010, arXiv:1006.3396.

[7] J.S. Marshall, *Redesign of PandoraPFA*, Report at IWLC 2010, October 19, 2010, Geneva.

[8] H. Aihara et al., *SiD Letter of Intent*, 2009, arXiv:0911.0006, SLAC-R-944.