A framework for vertex reconstruction in the ATLAS experiment at LHC

Bouhova-Thacker, E.; Koffas, T.; Kostyukhin, V.; Liebig, W.; Limper, M.; Piacquadio, G.; Prokofiev, K.; Weiser, C.; Wildauer, A.

DOI
10.1088/1742-6596/219/3/032019

Publication date
2010

Document Version
Final published version

Published in
Journal of Physics. Conference Series

A framework for vertex reconstruction in the ATLAS experiment at LHC

2010 J. Phys.: Conf. Ser. 219 032019

(http://iopscience.iop.org/1742-6596/219/3/032019)

Journal of Physics: Conference Series **219** (2010) 032019 doi:10.1088/1742-6596/219/3/032019

A framework for vertex reconstruction in the ATLAS experiment at LHC

E.Bouhova-Thacker[1], Th.Koffas[2], V.Kostyukhin[3], W.Liebig[4], M.Limper[4], G.Piacquadio[5], K.Prokofiev[2], C.Weiser[5], A.Wildauer[6]

on behalf of the ATLAS Collaboration

[1] Lancaster University, Lancaster, LA1 4YB, UK
[2] PH Department, CERN, CH-1211 Geneva-23, Switzerland
[3] Physikalisches Institut der Universität Bonn, Nußallee 12, D-53115, Bonn, Germany
[4] Nikhef, Science Park 105, 1098 XG, Amsterdam, Netherlands
[5] Albert-Ludwigs-Universität Freiburg, Hermann-Herder-Strasse 3, D-79104, Freiburg, Germany
[6] IFIC Universitat de Valencia, Apartado de Correos 22085, E-46071, Valencia, Spain

W. Liebig has been supported by the Netherlands Organization for Scientific Research (NWO), research grant VIDI 680.47.218.

E-mail: `Kirill.Prokofiev@cern.ch`

Abstract. In anticipation of the first LHC data to come, a considerable effort has been devoted to ensure the efficient reconstruction of vertices in the ATLAS detector. This includes the reconstruction of photon conversions, long lived particles, secondary vertices in jets as well as finding and fitting of primary vertices. The implementation of the corresponding algorithms requires a modular design based on the use of abstract interfaces and a common Event Data Model. An enhanced software framework addressing various physics applications of vertex reconstruction has been developed in the ATLAS experiment. Presented in this paper are the general principles of this framework. A particular emphasis is given to the description of the concrete implementations, which are dedicated to diverse methods of vertex reconstruction.

1. Introduction

Presented in this paper is the framework for vertex reconstruction implemented in the reconstruction software of the ATLAS experiment at the Large Hadron Collider (LHC) of the European Organization for Nuclear Research (CERN). The modular design of the framework and the common Event Data Model are discussed in details. The implementations of reconstruction algorithms and their expected performances are demonstrated on several vertex topologies, such as photon conversions, secondary vertices in jets and exclusive J/ψ decays. The design and performance of parts of the framework responsible for the reconstruction of primary vertices was discussed in the previous note of this series [1] and will not be addressed here.

2. The ATLAS Inner Detector

The ATLAS Detector [2] is a multi-purpose particle detector in operation at the LHC at CERN. The aim of the ATLAS experiment is to study pp collisions at energies up to 14 TeV. The

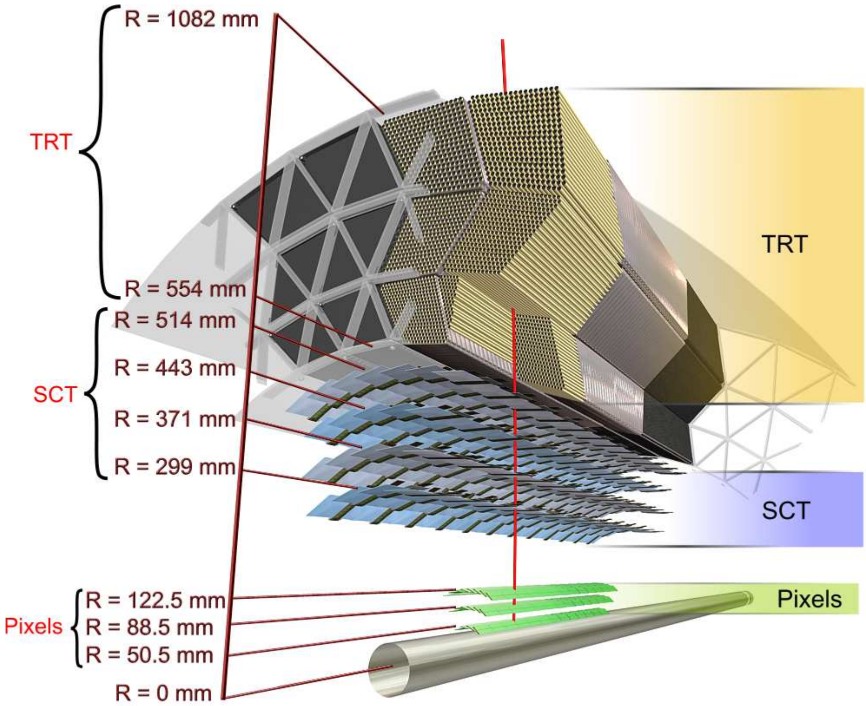

Figure 1. General layout of the ATLAS Inner detector.

detector is composed of several sub-detectors designed to study a variety of physics processes. For the reconstruction of vertices the Inner Detector (ID) is of most importance. Shown in Fig. 1 is the general layout of the ATLAS Inner detector. It is composed of three sub-detectors, which, from inside to outside, are:

- three barrel and 2×3 end-cap silicon pixel layers with an $R\phi$ resolution of approximately 10 μm and z resolution in barrel (R resolution in end caps) of approximately 115 μm;
- four barrel and 2×9 end-cap silicon microstrip layers (SCT) with an $R\phi$ resolution of approximately 17 μm and z resolution in barrel (R resolution in end caps) of approximately 580 μm;
- transition radiation tracker (TRT) with $e^{\pm}$ identification and an $R\phi$ resolution of approximately 130μm.

It provides on average $3 + 4 + 32$ measurements per charged particle trajectory, thus allowing for efficient reconstruction of tracks and vertices.

3. Overview of Vertex Topologies

Shown in Fig. 2 is the sketch representing the most important vertex topologies which can be produced in collisions of proton bunches at the LHC. In a typical collision, the signal high-p_t pp scattering will be superimposed with several minimum bias low-p_t events. Each of these interactions results in the production of a primary vertex in the interaction region of the Inner Detector. In addition to that, secondary vertex signatures such as photon conversions, cascade decays, V^0 decays and secondary vertices in jets may be produced. The efficient identification and reconstruction of these signatures is vital for many physics analyses.

The high accuracy of the ATLAS silicon tracker will allow to select jets originating from b-quarks by searching for tracks originating from transversely-displaced b-hadron decay vertices. The efficient detection and reconstruction of such vertices is essential to achieve a good b-tagging

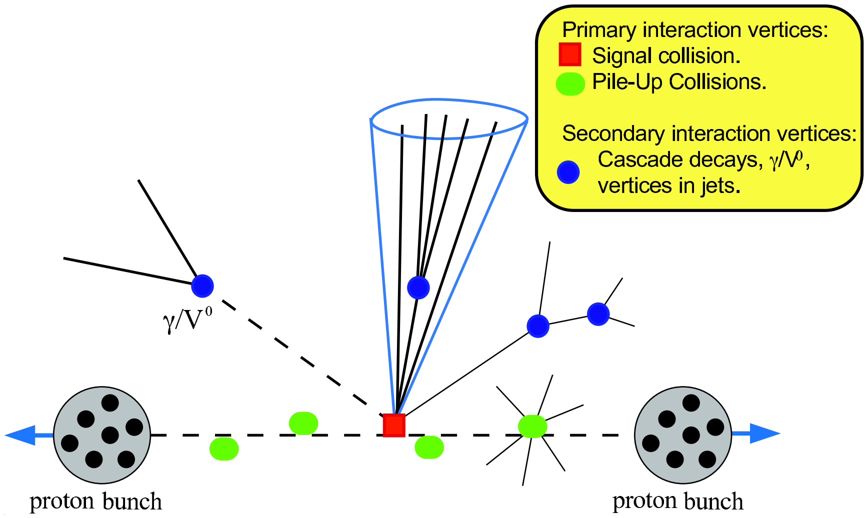

Figure 2. An overview of most important vertexing topologies produced in collisions of proton bunches.

performance. The fragmentation of a b-quark often results in a decay chain composed of a secondary vertex from the weakly decaying b-hadron and typically one or more tertiary vertices from c-hadron decays. The limited experimental resolution and high track density inside a jet therefore demand a dedicated vertex reconstruction.

Particle decays in flight and full decay chains are normally reconstructed using dedicated vertex finders. These algorithms exploit the assumed properties of the virtual particle and the conservation laws governing its decay, and use them to apply additional kinematic constraints during the vertex fit.

It is expected that about 40 % of all photons produced in pp collisions in ATLAS will convert in the material of the ID into electron-positron pairs [2]. The identification and reconstruction of photon conversions is thus important for many physics analyses. Reconstruction of conversions is also important to study the distribution of the material in the detector (e.g. detector description in simulation, calibration of calorimeters).

Finally, the reconstruction of two-body decays of long-lived neutral particles (V^0 decays) is crucial for several reconstruction and analysis strategies. It is used, for instance, to improve the b-tagging performance by rejecting the tracks with significant transverse impact parameters, but identified as coming from V^0 decays.

4. Framework Principle

The reconstruction of the vertex topologies mentioned in the previous section requires different approaches. The implementation of reconstruction strategies however should be done within one common software framework. The aim was to create a common modular software environment to allow the reconstruction of different vertex topologies within the same framework. In addition, it must be possible to use the algorithms implemented within the framework on both, reconstruction and analysis levels. Finally, the end-user should benefit from a common "look and feel" of the software when reconstructing different vertex topologies.

The principles mentioned above have been implemented in the framework for vertex reconstruction in the ATLAS Athena reconstruction software [3]. The framework is based on object-oriented C++ with Python steering. It benefits from a modular design based on the implementation of abstract interfaces for all tools and algorithms. The configuration of tools and algorithms is done during run-time through Python-based steering scripts. These scripts

Journal of Physics: Conference Series **219** (2010) 032019 doi:10.1088/1742-6596/219/3/032019

also define a concrete version of reconstruction algorithms to be used in a particular situation. This modular design with abstract interfaces allows for an easy exchange of components between algorithms, thus simplifying the adaptation to the physics process under investigation.

The implemented framework also benefits from the use of a common Event Data Model. For all strategies, the same group of C++ classes is used to describe the reconstructed vertices and their relation to other objects.

5. Event Data Model

As mentioned in the previous sections, the implemented framework is based on a common Event Data Model (EDM). The EDM is a set of data classes in which the information relevant to vertex reconstruction is stored. It is evident that reconstruction of different topologies requires significantly different level of details to be stored in various EDM objects. This is achieved by using inheritance throughout the data model. This approach has two important advantages:

- Quantities which are common to all vertex topologies are stored and retrieved the same way (common look and feel for users). This includes e.g. the vertex position and the vertex-track relations.
- Only the required amount of detail for a given vertex topology is stored in the objects and hence on disk. In addition, to reduce the usage of disk space, only the quantities which can not be re-calculated during data reading are retained and stored.

The EDM for vertex reconstruction also uses data classes of the general EDM for reconstruction of tracks whenever it has to deal with tracking quantities like track parameters or error matrices. End users and developers alike profit from this sharing of data classes.

Shown in Fig. 3 is the inheritance tree of main EDM objects of the ATLAS framework for vertex reconstruction. The simplest representation of the reconstructed vertex is provided by the

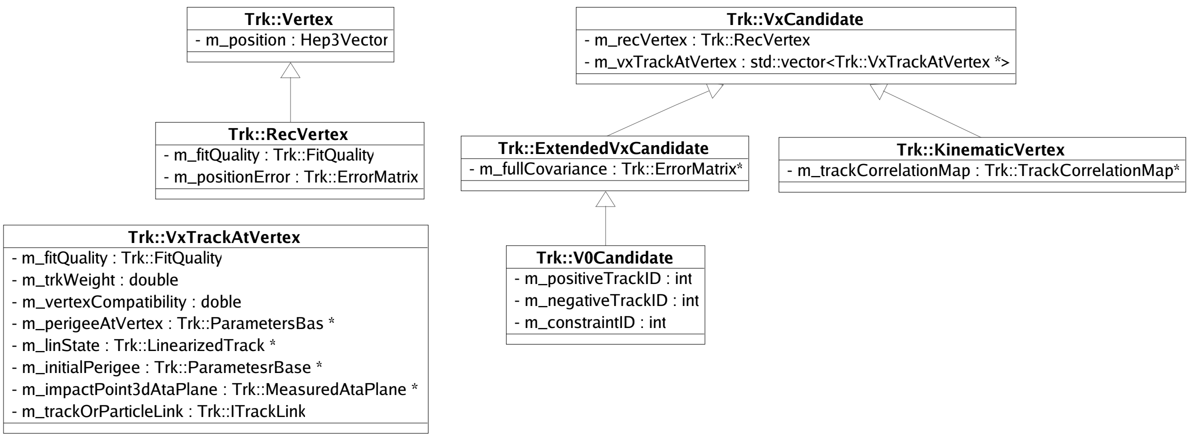

Figure 3. Main classes of Event Data Model of the ATLAS framework for vertex reconstruction.

RecVertex class. It stores the reconstructed position of the vertex and corresponding covariance matrix. The base representation of the reconstructed vertex returned by reconstruction algorithms is given by the *VxCandidate* class. This class stores the reconstructed vertex position, corresponding covariance matrix, χ^2 of the fit and related number of degrees of freedom and a vector of pointers to tracks which were fitted to this vertex. Various extensions of this class exist, representing the vertices reconstructed in constrained fits and results of reconstruction of V^0's and similar objects.

6. Common abstract interfaces

The common abstract interfaces are defined for all vertex reconstruction and related helper tasks. Each interface has several concrete implementations. The implementation to be used at each step of reconstruction is defined by the steering mechanism during run time. Shown in Fig. 4 are examples of abstract interfaces for different components of ATLAS vertex reconstruction. It can be noted that the tools responsible for vertex finding (association of tracks to a concrete

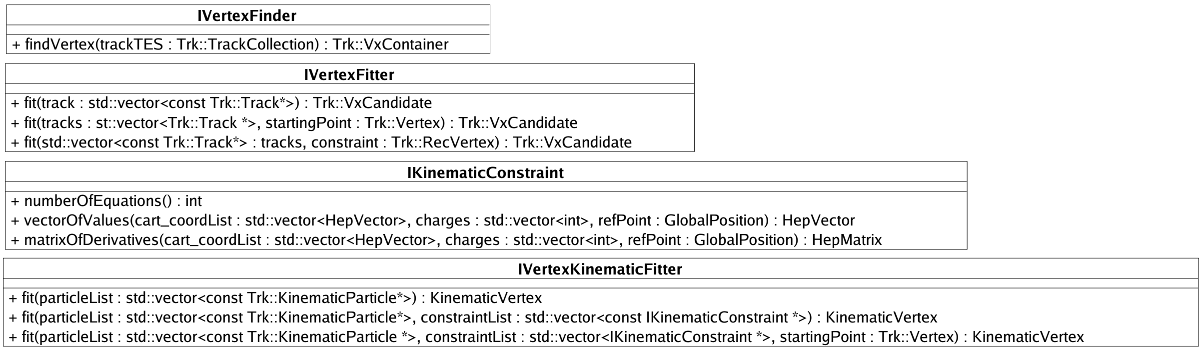

Figure 4. Main abstract interfaces of the ATLAS framework for vertex reconstruction.

vertex hypothesis) and vertex fitting (reconstruction of vertex position, covariance matrix, fit quality etc..) have different abstract interfaces. Indeed, in ATLAS these two stages of the vertex reconstruction are implemented separately and in most cases do not depend on each other. Separate interfaces were also implemented for helper tools updating vertex estimates with reconstructed trajectories, refitting trajectories with the knowledge of reconstructed vertices etc.

A separate set of abstract interfaces is created for constrained vertex fitting. The base class for implementation of linearized constraint equations is implemented separately from the minimization algorithm. The constraints to be applied in the particular vertex fit can thus be defined during run-time using the steering mechanism. The development of interfaces for constrained vertex fitting is discussed in more details in [4].

This structure of abstract interfaces, in conjunction with an event data model for vertex reconstruction, provides a common look and feel to the end-user who, regardless of the vertex topology, always works with the same EDM classes and interfaces. This design also allows for a high level of modularity and flexibility. Concrete implementations can be exchanged or added easily by modifying the external steering, without the need to update client software. In addition, many different approaches can be applied and tested in parallel to achieve the best result for a certain vertex topology. In the next section, several concrete implementations of strategies for vertex reconstruction are presented and preliminary results are discussed.

7. Examples of implementation

7.1. Reconstruction of photon conversions

Shown in Fig. 5 is the sequence diagram of an algorithm for the reconstruction of photon conversions in the Inner Detector. The use of abstract interfaces mentioned in the previous section is illustrated. The *ConversionFinder* algorithm uses an implementation of the *IVertexFinder* to find the conversion vertices. The latter tool uses implementations of *ITrackSelector* and *IVertexFitter* to pre-select signal tracks and reconstruct vertex candidates. This procedure is done in a loop where the used instance of the vertex fitter provides feedback to the finder concerning the “quality” of the vertex fit. The instance of the vertex finder then decides whether to accept the fitted vertex or not.

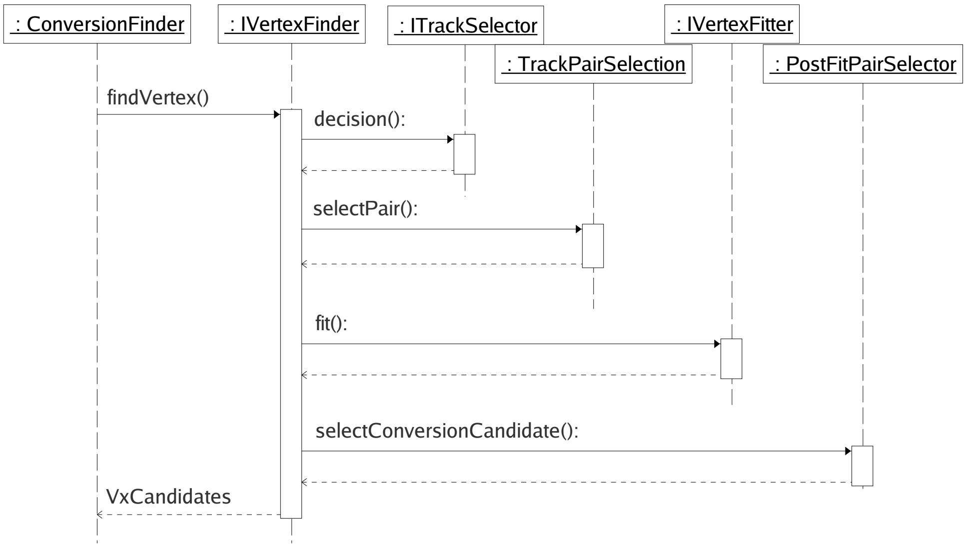

Figure 5. Sequence diagram showing the details of implementation of an algorithm for the reconstruction of photon conversions.

It should be noted that the reconstruction of converted photons normally involves the application of mass or angular constraint and thus can be CPU-consuming. A careful pre-selection of tracks and pair candidates is therefore necessary.

Shown in Fig. 6 (left) is the distribution of the residuals of the reconstructed radial position of conversion vertices for photons originating from simulated $H \to \gamma\gamma$ decays, where the mass of the Higgs boson is equal to 120 GeV. The constrained vertex fit, requiring tracks to have the same

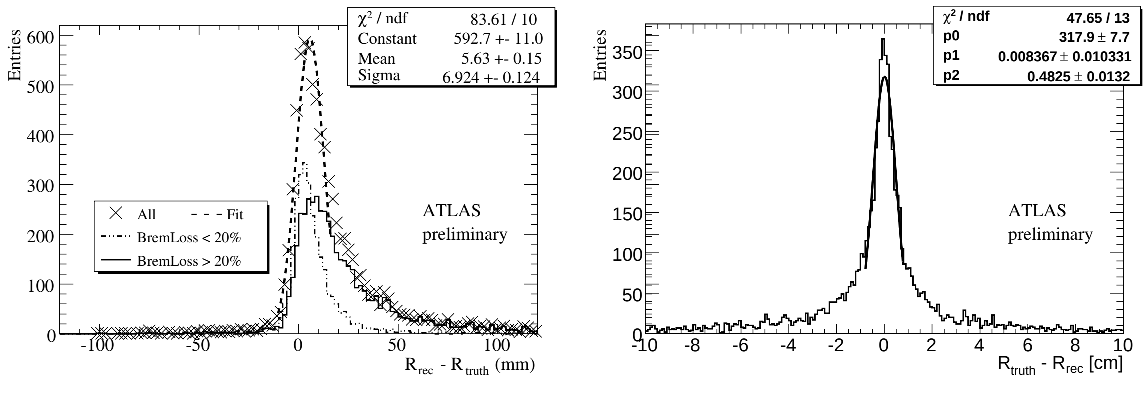

Figure 6. Distributions of residuals of transverse radius of reconstructed vertex. **Left:** conversions of photons produced in simulated $H \to \gamma\gamma$ decays, where the mass of the Higgs boson is equal to 120 GeV; **Right:** $K^0_S \to \pi^+\pi^-$ vertices.

direction at the vertex, has been used. The long tail to the right is due to bremsstrahlung losses of the two produced electron tracks, which reduce the individual track parameter reconstruction quality, hence affecting also the vertex fit results. To illustrate this, the radial position resolution with and without significant ($>$ 20%) losses due to bremsstrahlung, is plotted separately. An overall radial position resolution of approximately 7 mm has been achieved. The use of algorithms allowing reconstruction of tracks with recovery of energy losses is currently under investigation.

To compare with the reconstruction of photon conversions, a case without bremsstrahlung losses was selected. Shown in Fig. 6 (right) is the distribution of the residuals of the reconstructed radial position of the $K_S \rightarrow \pi^+\pi^-$ decay vertices. Instead of the angular constraint used for conversions, a direct mass constraint is implemented. The absence of a bremsstrahlung-related tail compared to that in Fig. 6 (left) is evident.

Shown in Fig. 7 are the overall track, track-pair and vertex reconstruction efficiencies for converted photons originating from $H \rightarrow \gamma\gamma$ events, where the mass of the Higgs boson is equal to 120 GeV. It can be noted that only tracks originating from a radial distance from the beam

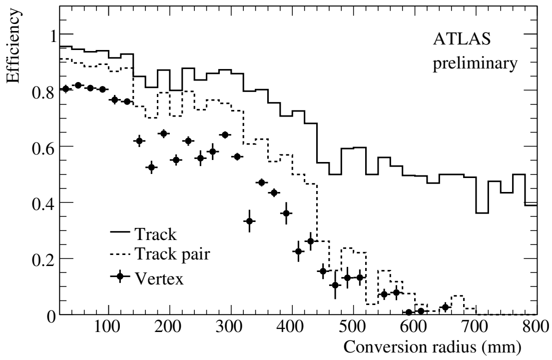

Figure 7. Track, track-pair and vertex reconstruction efficiency for converted photons originating from $H \rightarrow \gamma\gamma$ decays, where the mass of the Higgs boson is equal to 120 GeV, as a function of distance from the beam axis.

axis of up to approximately 800 mm can be efficiently reconstructed. However, the efficiency for reconstructing track pairs is significantly reduced at radial distances above 400 mm. Above this radius one needs to rely on the tracks reconstructed within the TRT detector only (see Sec. 2). The results in a reduced resolution in separating the conversion tracks.

This is due to the missing measurements from the pixel detector and the reduced number of measurements in the silicon strip detector. In addition, due to ATLAS tracker geometrical constraints, the track reconstruction efficiency is severely curtailed for pseudorapidity values of $|\eta| > 2$, although tracks are still reconstructed for up to $|\eta| = 2.5$. The conversion finding of ATLAS is described in more details in [5].

7.2. Constrained vertex fitting

The reconstruction of vertices which stem from a heavy flavor decay, a converted photon or the decay of a long-lived hadron, usually involves the application of kinematic constraints during the vertex fit. Most commonly, the application of additional constraints is implemented under an "all-in-one" fitting approach. An example of such an algorithm implemented in the ATLAS software framework is the VKalVrt package [6].

Recently, a new tool for constrained vertex fitting has been developed under the modular concept described in Sec. 6. This new tool, *VertexKinematicFitter*, is based on χ^2 minimization with Lagrange multipliers [7]. The minimization procedure finds parameters of tracks and a fitted vertex which simultaneously satisfy a vertex constraint and other kinematic constraints requested by the user. The modular approach allows the additional constraints to be implemented as separate classes with a common abstract interface.

Shown in Fig. 8 (left) is the resolution on the transverse momentum of muons produced in simulated $J/\psi \rightarrow \mu^+\mu^-$ decay before and after the vertex fit with a mass constraint. A

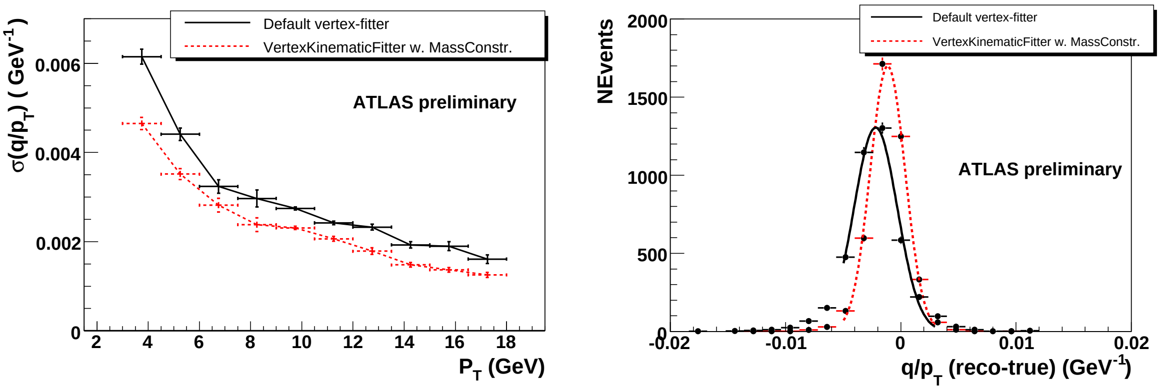

Figure 8. Left: Resolution on transverse momentum of the muon tracks originating from $J/\psi \rightarrow \mu^+\mu^-$ decay at the vertex reconstructed with the default vertex fitter and the VertexKinematicFitter with mass constraint; **Right:** Distribution of residuals of charge to transverse momentum ratio of muon tracks originating from $J/\psi \rightarrow \mu^+\mu^-$ decay before and after constraint fit.

clear improvement can be observed in the case of constrained fit over whole range of transverse momenta.

Constrained vertex fitting can also be used to help correcting shifts in reconstructed parameters induced by certain weak mode misalignments. These misalignments represent systematic deformations of the detector, affecting properties of reconstructed charged particles without increasing the hit residuals (i.e. the difference between the measured and predicted positions of hits). Shown in Fig. 8 (right) are the distributions of residuals of the reconstructed q/p_T of muon tracks, obtained with a conventional χ^2 fit and the mass constraint fit for the events reconstructed with a so-called 'curl' misalignment geometry [4]. It can be seen that the vertex fit with J/ψ mass constraint allows to recover some of the q/p_T shifts arising due to misalignment.

The application of this tool to the reconstruction of decay chains with one or several intermediate vertices is currently under investigation. The decay chain is meant to be reconstructed in a sequential way starting from the final state particles. The decayed particles reconstructed as a result of constrained vertex fits serve as an input for further steps of constraint fitting.

7.3. Reconstruction of vertices in jets

The algorithms for vertex finding in jets exploit the secondary vertex topology to enhance the b-tagging performance. At the moment, there are two such algorithms implemented in the ATLAS framework for vertex reconstruction. The first one relies on the assignment of all reconstructed tracks to a common geometrical vertex (*inclusive* vertex finder). The second one tries to identify the presence of a Primary Vertex (PV) $\rightarrow b \rightarrow c$ decay topology (*topological* vertex finder). The *inclusive* vertex finder is based on a vertex fitting method proposed by *P. Billoir* [8] and is implemented in the *VKalVrt* package [6]. The *topological* vertex finder uses an extension of the Kalman Filter formalism for vertex reconstruction developed in ATLAS and is implemented in the *JetFitter* package [9].

The initial finding strategy is common for both finders:

- Selection of displaced tracks;
- Reconstruction of all 2-track vertices;

- Removal of vertices and associated tracks which are compatible with γ, K_S and Λ decay hypotheses.

The *inclusive* vertex finder then tries to reconstruct a common geometrical vertex out of the surviving tracks. Tracks with a bad χ^2 contribution are removed iteratively from the fit, until the overall χ^2 is below a predefined threshold.

The *topological* vertex finder solves the pattern recognition problem by relying on the assumption that all tracks intersect a common PV $\rightarrow b-$hadron $\rightarrow c-$hadron flight axis, thus reducing a three-dimensional clustering problem to a one-dimensional one. The b-flight axis is initialized with the calorimetric jet direction and a first fit is performed assuming all tracks come from single vertices. This determines the b-flight axis direction and its intersections with the single tracks. Then an iterative clustering procedure is performed. Compatible vertices (with the primary vertex being one of them) are clustered in pairs, in decreasing order of compatibility, resulting in the end in a well-defined topology. The sequence diagram showing the details of implementation of this algorithm is shown in Fig. 9.

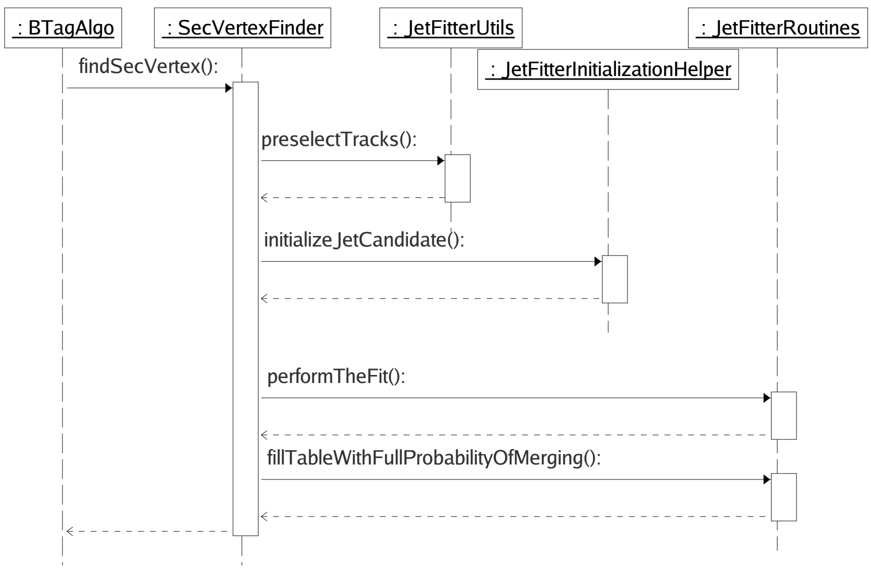

Figure 9. Sequence diagram showing the details of implementation of an algorithm for finding the secondary vertices in jets.

Shown in Fig. 10 (left) is the resolution achieved on the inclusive b-hadron decay vertex with respect to the true b-hadron position in the transverse plane (in the direction of flight of the b-hadron). The core resolution is about $400\mu m$, with a tail due to the contribution of tracks from c-hadron decays. The b-tagging performance achieved on $t\bar{t}$ and $t\bar{t}$ jet jet Monte Carlo events by the two secondary vertex-based algorithms, combined with the conventional pure impact parameter based algorithm, is illustrated in Fig. 10 (right). The results are normalized to the rejection of the conventional impact parameter-based algorithm. The rejection of light-quark jets is defined as the inverse of the probability to (mis)identify a light-jet as a b-jet. It can be noted that the use of the secondary vertex-based algorithms results in an improvement of the light-quark rejection by a factor up to 2-3.

8. Conclusions

Presented in this paper is a framework for vertex reconstruction in the ATLAS experiment at the LHC. An overview of general principles of the framework, its Event Data Model and abstract

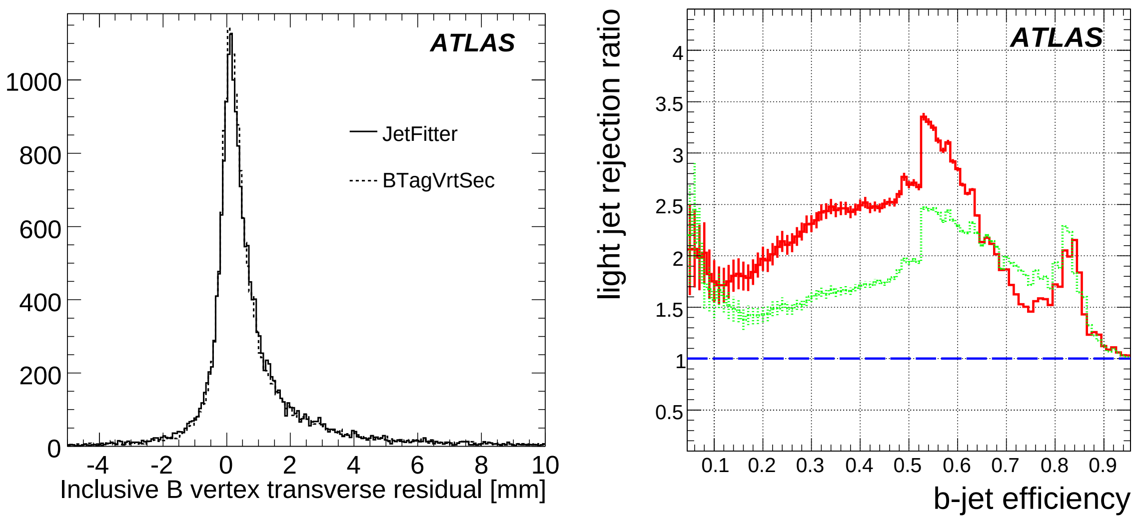

Figure 10. Left: Radial position resolution of the secondary vertex for the inclusive vertex finder (*BTagVrtSec*) and topological (*JetFitter*) vertex fitters; **Right:** Light jet rejection ratio vs b-jet tagging efficiency for inclusive fitter-based (dotted green line) and topological fitter-based (solid red line) b-tagging algorithms, combined with the conventional pure impact parameter based algorithm, normalized to the rejection of conventional impact parameter-based b-tagging algorithm (dashed blue line).

interfaces is given. The examples of implementation of algorithms for reconstruction of various vertex topologies such as photon conversions, reconstruction of vertices in jets and constrained vertex fitting are presented. The implementations of these algorithms is handled within a single modular software environment. This modular approach allows for an easy exchange of software tools between different algorithms and simplifies the implementation of new strategies. The use of a common Event Data Model for all strategies provides a “common look and feel” of different algorithms for the end-user.

The various components of this framework have been tested extensively under as real as possible conditions using latest Monte Carlo datasets. The simulation of these data sets includes overlaid minimum bias vertices to account for the presence of pile up, a displaced proton-proton collision point, detector misalignment and other conditions expected during the startup and run periods of LHC.

References

[1] G.Piacquadio, K.Prokofiev, A.Wildauer, J.Phys.Conf.Ser.119:032033,2008.
[2] The ATLAS Collaboration, G. Aad et al., *The ATLAS Experiment at the CERN Large Hadron Collider*, JINST 3 (2008) S08003.
[3] The ATLAS Collaboration, **URL:** https://twiki.cern.ch/twiki/bin/view/Atlas/WorkBook
[4] The ATLAS Collaboration, Auke Pieter Colijn et al. ATL-INDET-PROC-2009-003, Feb 2009.
[5] The ATLAS Collaboration, *Expected Performance of the ATLAS Experiment, Detector, Trigger and Physics*, CERN-OPEN-2008-020, Geneva, 2008.
[6] V.Kostyukhin, *VKalVrt - A package for vertex reconstruction in ATLAS*, ATL-PHYS-2003-031 (2003).
[7] P. Avery, `http://www.phys.ufl.edu/~avery/fitting.html`.
[8] P.Billoir and S. Qian, Nucl. Instr. and Meth. A311 (1992) 139-150.
[9] G.Piacquadio, C.Weiser, *A new inclusive secondary vertex algorithm for b-jet tagging in ATLAS*, CHEP 2007, Journal of Physics: Conference Series 119 (2008) 032032.