PROCEEDINGS OF SCIENCE

PoS(Vertex 2007)025

Track and Vertex Reconstruction in ATLAS

Wolfgang Liebig[*]
NIKHEF, PO Box 41882, 1009 DB Amsterdam, the Netherlands
E-mail: `Wolfgang.Liebig@cern.ch`

on behalf of the ATLAS Collaboration

High track densities and event rates in the very precise tracking devices of the ATLAS Detector at the LHC put stringent demands on the track and vertex reconstruction software in terms of performance, speed and maintainability over the experiment's lifetime. With this complex problem in mind the track and event reconstruction software in the ATLAS inner detector has been recently re-structured and extended, implementing a new, highly modular approach based on a global ATLAS event data model. The current reconstruction chain has been prepared for dealing with real data by validating its performance on simulated LHC collisions as well as participation in various detector commissioning programmes. This document gives an overview of the ATLAS tracking and vertexing algorithms and their performance as well as several underlying software aspects which are both essential for fully exploiting the data from the ATLAS inner detector.

Vertex 2007
23-28 September 2007
Lake Placid, NY, USA

---

[*] Speaker.

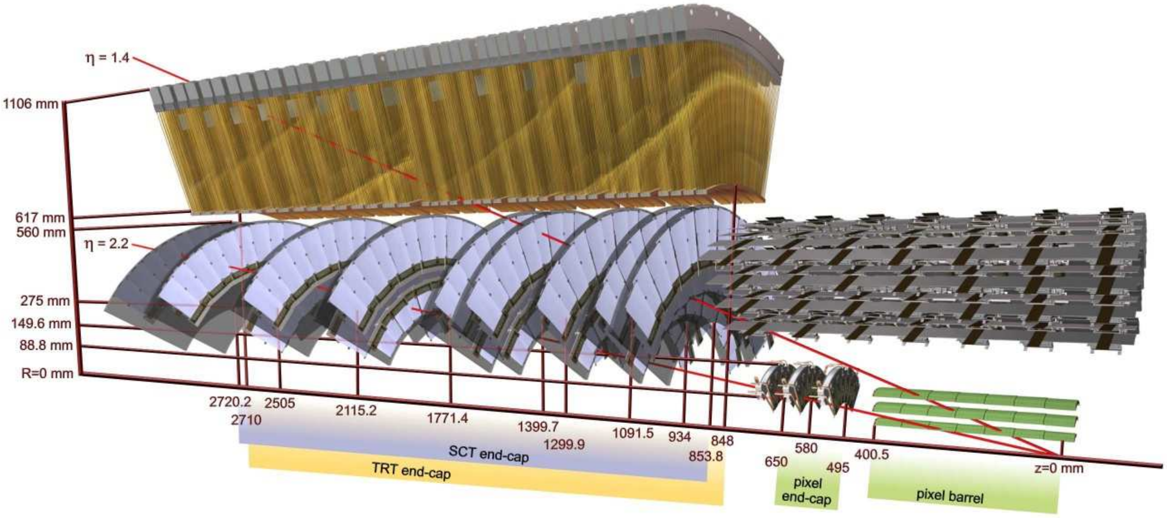

Figure 1: The ATLAS inner detector, exposing a sector in ϕ for the $z > 0$ hemisphere. The barrel TRT is omitted in this display. High energy tracks from the interaction point traverse on average 7 measurement layers of silicon modules and 36 straw tubes.

1. Introduction

The ATLAS experiment [1] is one of the two general-purpose detectors at the Large Hadron Collider at CERN, and is designed for searches for new particles as well as high-precision measurement of Standard Model processes. It has two independent large tracking devices, the inner detector (ID) and muon spectrometer, which were installed recently and are now being set up for LHC data-taking [2].

The ATLAS ID is housed in a 2 T solenoid magnetic field and consists of three technologies: a three-layered pixel detector (Pixels) closest to the interaction point followed by further layers of silicon strip detectors (SCT) and a straw tube tracker (TRT) measuring also transition radiation of traversing particles. The arrangement of concentric layers and endcap disks fitted with silicon modules and TRT straw tubes is shown in Fig. 1. The Pixel detector consists of identical modules segmented into 320×144 pixels, each with dimensions of $50\mu\text{m} \times 400\mu\text{m}$. Its intrinsic resolution in the bending co-ordinate is below $12\,\mu$m, since the charge collection measurement is exploited to adjust the cluster position. The SCT employs silicon wafers segmented into 768 strips with an average pitch of $80\,\mu$m. With a digital threshold setting, its resolution achieves $22\,\mu$m but is enhanced by fitting each module with two wafers glued back-to-back with a relative stereo angle of 40 mrad. The Pixel and SCT detectors have 80 M and 6.2 M channels. The TRT consists of an arrangement of 298K drift tubes with a $31\,\mu$m thick gold-plated tungsten wire at the center of each 4 mm wide straw. Its resolution is $140\,\mu$m and slightly worse only for tracks passing near the wire.

The track reconstruction software uses the position measurements from the aligned and calibrated ID to infer the position and momentum of the particles produced at the collision point and elsewhere in the detector. While it is only at the end of the reconstruction chain that the tracks from the ID and muon spectrometer are combined to form μ candidates, a large part of the track reconstruction software in both tracking systems has been designed in close collaboration as a common

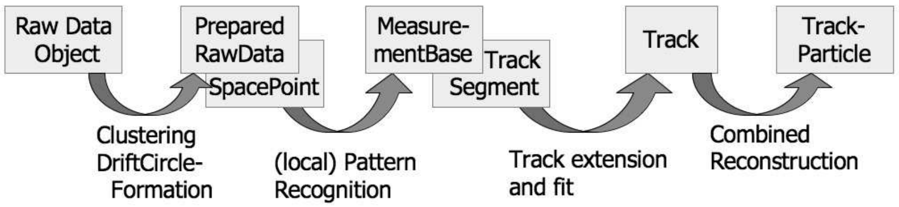

Figure 2: The object flow in the ATLAS tracking EDM in the way as it is implemented by inner detector and muon spectrometer.

project. The tracking data model is capable of representing measurements and reconstructed objects in both systems. Much of the algorithmic code and the standard calculations in tracking (e.g. parameter propagation, track fit) have been developed so that they work independently of the given technology and can be accessed from ID, muon spectrometer and physics object reconstruction. This efficiently avoids code duplication and inherently facilitates combined muon tracking. The vertex reconstruction software employs the same modular design principles and makes use of the tracking software wherever possible.

2. Tracking Software Structure and Algorithms

The existing monolithic algorithms to reconstruct tracks and vertices in the ATLAS ID met the performance goals on simulated data [1], but had their proprietary event data representation and lacked the abstract layer dividing a task from its concrete implementation, which was needed to cope e.g. with real data. Following the recommendation of an internal task force [3], the project has been migrated to utilise the new common event data model for tracking. It has also been integrated further into the ATLAS offline software framework Athena [4], such that algorithmic code is structured in a sequence of top algorithms which delegate repetitive tasks to common tools and use common services to retrieve detector description and calibration data.

2.1 Tracking Event Data

The data model for tracking in ATLAS is organised in a hierarchy of polymorphic classes reflecting the evolving knowledge about the event data during the subsequent off-line reconstruction steps [5]. According to Fig. 2, it starts with raw data objects and prepares the data for the track finding, e.g. provides clusters of pixels or strips with calibrated positions and uncertainties. A flexible track objects stores the associated measurements together with the fitted trajectory parameterisation

$$\mathbf{x} = (l_1, l_2, \phi, \theta, q/p)^T$$

and its uncertainties. The parameters l_1, l_2 are the local coordinates on a given surface, followed by the azimuthal and polar angles and the track curvature q/p. The track objects (or alternatively a more lightweight representation in physics analysis) form the inputs to the vertex reconstruction.

The measurements at different levels of calibration and reconstruction are defined by abstract base classes suitable for all tracking detectors in ATLAS, whereas detector-specific information is added by extending each class into the sub-detector software realm [5]. This design is the key component for forming the common, modular reconstruction software in the ID and muon spectrometer and serves for data storage as well as passing information through abstract interfaces.

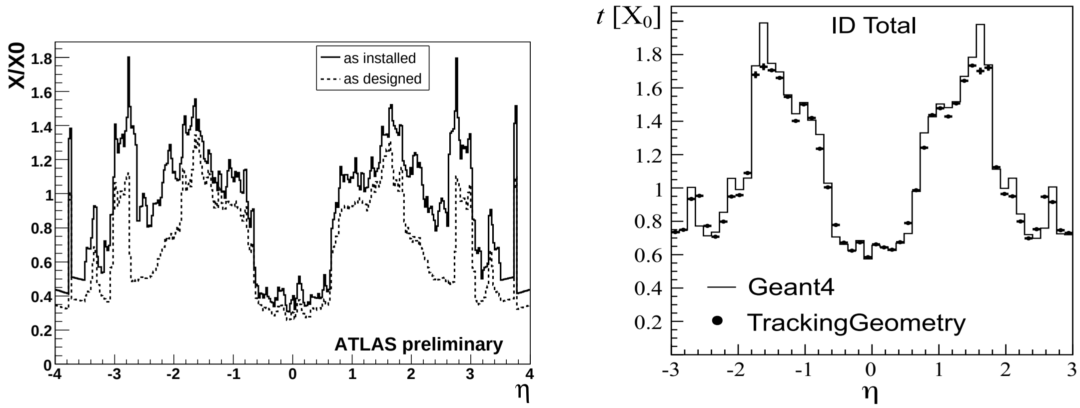

Figure 3: Left: Integrated radiation length (X_0) in the ID as function of pseudorapidity η for the currently installed detectors and for an older simulation which did not yet have information from measuring the actually built detectors. Right: To speed up the track reconstruction the material constants from the full simulation are mapped onto a simplified tracking geometry, which reproduces well the full information.

2.2 Detector Description and Conditions Data

Information about the detector is separated from the event data and provided by so-called Services. They are active during reconstruction as well as event simulation and supply both with the nominal positions and material constants of the read-out elements and every other object in the detector. There is also other, time-dependent meta-data such as the beam spot, calibration and alignment constants. Like the detector description, they are provided by a central database, but here a service updates the users of such data once the stream of processed data changes to a new “interval of validity” with potentially different constants.

A recent effort has brought the detector description and material constants in as close agreement with the installed detectors as possible, for example by adding more detail and by comparing weighed masses with the values calculated from the geometry. As a result the ID material budget has significantly increased, as shown in Fig. 3 (left). Precise material effects corrections during e.g. track parameter propagation and fitting are essential for fully exploiting the precise silicon detectors. At the same time it would be far too slow to determine those corrections on-the-fly from the full simulation, therefore a dedicated tracking geometry service has been created which maps the ID’s logical structure and material distribution onto a lightweight tracking geometry [6]. It has been validated and Fig. 3 (right) shows that the total radiation lengths in the simplified ID tracking geometry follow very closely that of the full simulation. While providing precise material effects to the track fit at very low CPU time cost, its material budget can be tuned on the data themselves while the detector placements are picked up automatically from the full geometry. The use of detector conditions in the reconstruction chain has recently been validated by means of an ATLAS-wide data challenge which used simulation with purposefully and realistically distorted detector constants, namely sensor alignment, material densities and magnetic field.

2.3 Track Reconstruction in the Inner Detector

During the 15-year long phase of preparing and improving the ID track reconstruction software since the conception of ATLAS, several competing reconstruction packages have been developed and maintained to this day. They understand the track reconstruction as one single task and each employed a proprietary data model and specific calibrations with only little access to detector conditions. This situation has been identified as a grave danger for reconstructing real LHC data and for the needed flexibility to adapt to specific detector set-ups or event topologies. Following a recommendation by an internal task force, a new modular approach has been commissioned in 2004, which allowed the existing packages to migrate their code into the new tracking framework (NEWT) and added also new techniques [7]. The modularity in NEWT is achieved through two concepts supported by the offline software framework: first, the track reconstruction is broken down into a sequence of independent algorithms; second, for each well-defined task a tool is invoked through an abstract interface. The detailed sequence of algorithms and their active tools are both configured at run-time. This has significantly increased the stability of the regular software releases and now allows to optimise the performance of single components and to adapt rapidly to early detector configurations from the commissioning phase.

ID Reconstruction sequence

First pre-processing algorithms generate silicon clusters, drift circles and space points as input to the pattern recognition. The track search then starts with an inside-out strategy:

(1) The track finder starts in the Pixel+SCT layers and searches for space-point triplets as track seeds, applying a search window with momentum and impact parameter cuts to limit the number of space point combinations. Candidates are extended by a local pattern recognition based on a simple combinatorial Kalman filter to build the full silicon track candidate.

(2) It is followed by an ambiguity solver algorithm which scores the candidates such that full tracks are favoured over small segments and the number of shared hits between tracks reduced. It performs a full track fit with precise material corrections and full propagation of track parameters and errors in the measured B-field.

(3) The next step extends the silicon tracks into the TRT and assigns drift radius measurements to the extended track, hereby solving the inherent left-right ambiguity from the drift radius.

(4) Finally a TRT-extension processor performs a full track fit and scores the new track to decide if the proposed extension is useful.

This sequence is complemented by an outside-in track search starting from yet unassigned TRT segments. It is mainly aimed at reconstruction of photon conversions in the detector and decay vertices of neutral particles, but is also able to recover the remaining trajectory after a catastrophic energy loss, which has not been compatible with the search cone applied in step (1). The ID also disposes of a lightweight alternative, CTBtracking [7], which performs both strategies in a low-multiplicity environment such as test beam and cosmic events.

Common Tracking Tools

The tools, on the other hand, provide most of the algorithmic code and cover detector-specific tasks (e.g. the cluster formation or the silicon-detector seeded pattern recognition in the ID) as well as the basic and often repetitive calculations, such as track parameter propagation or track fits.

The two track fitting techniques which are widely used in high energy physics, the global least-squares fit [8] and the Kalman filter [9], are both implemented in ATLAS. The global least-squares fit obtains the best estimate for the track parameters by minimising a (linearised) χ^2 function built

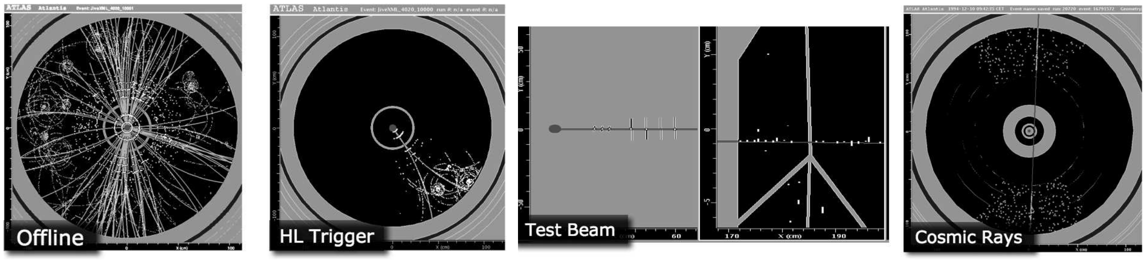

Figure 4: Track reconstruction in the ATLAS inner detector in different applications and detector set-ups (from left to right): fully simulated $pp \rightarrow t\bar{t}$ collisions, a region of interest processed by the high-level trigger, particles in the combined test beam 2004 and cosmics ray tracks in the readily installed TRT detector.

from the hit residuals at every measurement surface. Material effects are included as additional fitting parameters weighted by their variance, which in turn is estimated from the material parameters provided by the tracking geometry.

The Kalman filter [9] determines the trajectory vector by iteratively integrating all measurements along the track. Each filter step extrapolates the previous parameters to the current surface (taking material effects from the tracking geometry into account) and updates the parameters via the gain formalism [9]. It is followed by a smoother and outlier-rejection procedure. Additional techniques have been implemented as well, e.g. for the reconstruction of electron tracks with energy loss through bremsstrahlung [10].

Tracking in Different Detector Set-ups

The inner detector track reconstruction is designed to serve the needs of different applications and commissioning programmes. The same sequence as described above can be run to reconstruct full events offline and to provide tracks in a region of interest selected by the trigger, thus running under the ATLAS high-level trigger. It has also taken part in the past commissioning programmes, such as the 2004 combined test beam and cosmic data taken at the tracking integration facility or in the pit. The inner detector tracks in the barrel part perpendicular to the z axis are shown in Fig. 4 for all of the above applications. For example, to reconstruct the test-beam and cosmic ray data from the real detectors the track reconstruction has worked with the full ATLAS framework to access detector conditions and has validated the infrastructure to apply calibration and alignment corrections. The latter were determined by the various alignment algorithms under study for ATLAS [11].

3. Tracking Performance

Key quantities showing the performance of the new ID tracking, such as track efficiency, fake rate and parameter resolutions, have been studied on different types of simulated events. In addition they are continuously monitored as part of the daily automatic testing framework to catch possibly unwanted changes in one of the participating modules. Figure 5 shows the resolution of transverse momentum and impact parameters as function of pseudorapidity $|\eta|$ in a somewhat idealised setup (no misalignment effects) an without use of a beamspot constraint in the fit. Multiple scattering at lower momenta and inelastic scattering of hadrons in the detector material deteriorate the intrinsic detector resolution in the track parameters and cause the observed dependencies on

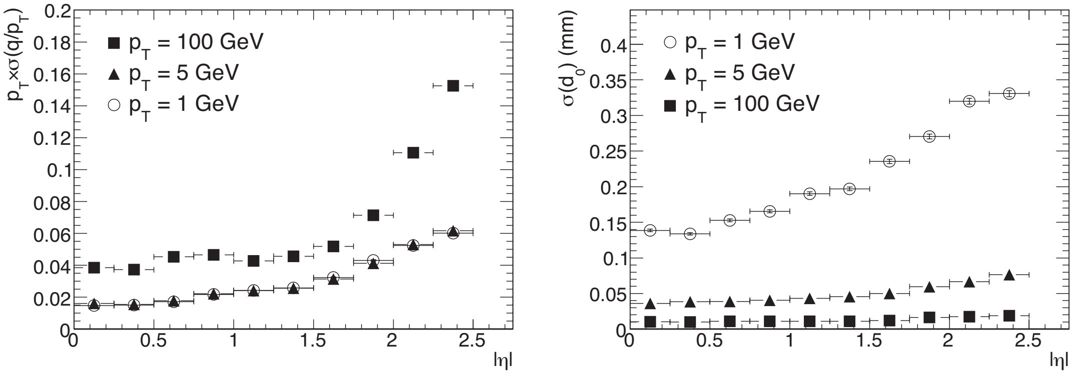

Figure 5: Resolution of the ID tracking in the relative transverse momentum for muons (left) and in the transverse impact parameter for pions (right), as a function of $|\eta|$.

$|\eta|$ and momentum. The efficiencies achieved by the current reconstruction software for charged tracks in multi-jet events traversing the silicon detectors are 98% in the barrel detectors and 96% in the end-caps. The full detector reconstruction efficiency taking into account also interaction with material and quality cuts on the reconstructed tracks ranges between 92% (barrel) and 80% (end-caps) whereas the rate of fake tracks stays well below 1%.

4. Vertex Reconstruction

The precise track parameter determination from the silicon detectors allows to extrapolate the tracks back to the interaction region and determine their probable point(s) of origin with high precision. Although the LHC beam spot will be very small with $\sigma_x = \sigma_y = 15\mu$m and $\sigma_z = 56$mm, this precision is not enough for optimal identification of b- and τ-jets and for analysis of several physics processes, e.g. for the Higgs boson discovery channel $H \rightarrow \gamma\gamma$. In addition several collisions are recorded per bunch crossing (up to 24 at $L = 10^{34}\mathrm{cm}^{-2}s^{-1}$) and potentially bias the triggered physics process. The vertex reconstruction software therefore estimates the number of primary proton-proton interactions in a single event and returns the fitted vertex positions with their associated tracks. Figure 6 (left) shows how a simple clustering of the z impact parameters from reconstructed tracks is able to separate the different vertices in one event, while for safely identifying the signal vertex other (e.g. kinematic) criteria have to be used as well.

Not all particles originate from the primary collision. B and D hadrons can decay in a measurable distance from the primary vertex and it is essential to identify their decay vertices for efficient tagging of b-jets. Here the vertexing software uses the same fitting techniques as for the primary vertex, while the search algorithm for secondary vertices is specific to the task. Finally also vertices of exclusive decays or photon conversions are searched for and fitted to facilitate reconstruction of specific topologies.

4.1 Vertex Reconstruction Software

In general the vertex reconstruction has been developed in parallel to the tracking software and also follows a modular structure, which now has both new developments and migrated existing code. The underlying event data model is composed of core classes for the basic vertexing

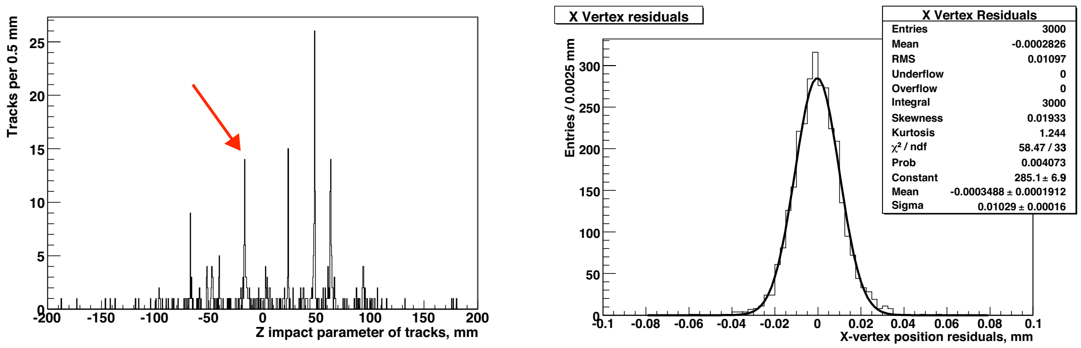

Figure 6: *Left:* Distribution of the z impact parameter of reconstructed tracks in a simulated $H(130) \rightarrow \gamma\gamma$ event at $L = 2 \cdot 10^{33}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$. The indicated cluster of tracks corresponds to the signal vertex. *Right:* Distribution of residuals in the x direction of reconstructed primary vertices in $t\bar{t}$ events.

objects and further extensions, which are used by specific vertexing applications. The core classes describe the vertex position, the track refitted under the vertex hypothesis and the vertex candidate associating such tracks to the vertex. Common interfaces have been defined for structural components like vertex finding, fitting and other, more specific applications. They work under all of the above vertexing tasks. Often several different implementations exist for one component, allowing to optimise the overall performance.

The main component implemented in this way is the vertex fit, of which four different implementations currently exist in ATLAS. Two of them follow the work of P. Billoir [12] and estimate the vertex position by approximating the equations of motion of a charged particle with their first-order Taylor expansions in terms of $\left(q/p\right)$. The two implementations differ in complexity. Another algorithm implements a conventional Kalman filter [13], hereby using a full analytical derivation of the equations of motion. The last algorithm is an iterative re-weighted Kalman filter which uses annealing iterations to downweight tracks that are less compatible with the vertex hypothesis [14]. This algorithm is more robust against outliers.

4.2 Primary Vertex Reconstruction

Two different algorithms are available for primary vertex finding in ATLAS. One follows the simple “fitting-after-finding” approach and works by clustering pre-selected tracks in the z-projection to determine the number of primary vertices. Then it reconstructs them using one of the different vertex fitter implementations. The other one can be characterised as “finding-through-fitting” and is called Adaptive Multi-Vertex Fitter [15]. It starts with a single seed and increases the number of seeds by forming new ones out of the outliers from the fit to the existing vertices. Then an iterative annealing procedure is used during the simultaneous fit of several vertices, such that a hard track-to-vertex assignment is approached. This approach achieves the best performance both in terms of efficiency of reconstructing the primary vertex and for the vertex position resolution. The latter, derived from the distribution of residuals, is shown for the transverse co-ordinate in Fig. 6 (right).

4.3 Secondary Vertex Reconstruction

Heavy flavour jets can be tagged using the characteristic lifetime of b-hadrons. ATLAS uses

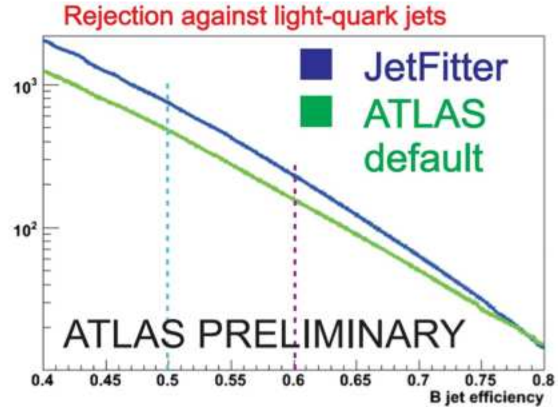

Figure 7: The ATLAS b-tagging measured in terms of rejection against light-quark jets determined on $pp\rightarrow t\bar{t}$ events.

a combination of two algorithms: one forms a discriminator based on the impact parameters of displaced tracks, the other one exploits the properties of explicitly reconstructed b-decay vertices. In this case the secondary vertices are reconstructed by a specific vertex finder [16], which uses the simple Kalman filter to obtain an inclusive single b-decay vertex.

However, the underlying hypothesis of a common geometrical vertex is not correct, since the b-decay and following c-decay vertex can be significantly apart. Given that the observed track multiplicities from each decay vertex are low, the b- and c-hadron decay positions can usually not be fitted independently. The solution which ATLAS has been studying recently therefore fits the B/D decay chain as separate vertices by constraining the tracks to lie on the b-hadron flight axis and clustering them along this axis. The algorithm is called JetFitter [17]. The algorithm then calculates the invariant mass, energy fraction and flight length significance for the so reconstructed vertices and creates a tagging likelihood, which is combined with the impact parameter tag in the same way as the default vertex tagging. The comparison of the default algorithm and the combination with JetFitter in Fig. 7 shows a significant gain in the rejection of light quark jets.

5. Conclusions and Outlook

The ATLAS ID has deployed a new, modular software for tracking and vertexing which allows to continuously improve its performance while it takes full part in the wide range of ATLAS activities to prepare the reconstruction for LHC turn-on. New techniques have been implemented such as the precise and fast material effects from the tracking geometry or the dedicated electron track fits. The same has happened in the vertex reconstruction with the Adaptive Multi-Vertex Finder and the secondary vertex reconstruction which identifies cascade vertices.

The testing of the tracking software with detector conditions support in several commissioning projects with data from the real and final detectors has greatly helped in validating the infrastructure to cope with realistic detector effects. At the same time these projects have profited from the new software which allowed reconstruction of tracks in a non-LHC environment with only little adaptation and thus have confirmed the modular approach. Further tests and improvements on simulated data and first reconstruction of cosmic tracks in the final and installed silicon detectors are planned in the coming months before the first collisions will be reconstructed.

6. Acknowledgements

The author is greatly indebted to his colleagues from the ID software group who helped developing, validating and extending the new ID tracking. He would like to thank G. Gorfine, A. Salzburger, G. Piacquadio and K. Prokofiev for kindly providing some of the figures. This work has been supported by the Netherlands Organisation for Scientific Research (NWO) under the research grant VIDI 680.47.218.

References

[1] The ATLAS Collab., CERN/LHCC/99-15 (1999).

[2] D. Robinson, these proceedings.

[3] V. Boisvert et al, ATLAS note ATL-SOFT-2003-010.

[4] The ATLAS Collab., CERN-LHCC-2005-022.

[5] F. Åkesson et al, ATL-SOFT-PUB-2006-004.
T. Cornelissen et al, ATL-SOFT-PUB-2007-003.
F. Åkesson et al, ATL-SOFT-PUB-2007-006.

[6] A. Salzburger et al, ATL-SOFT-PUB-2007-004.

[7] T. Cornelissen et al, ATL-SOFT-PUB-2007-007.

[8] G. Lutz, Nucl. Ins. and Meth. A273 (1988) 249.
T. Cornelissen et al, Proc. of CHEP 2007 # 144.

[9] R. Frühwirth et al, Nucl. Ins. and Meth. A262 (1987) 444.

[10] V. Kartvelishvili, in Proceedings of "XI Int. Workshop on Adv. Comp. and Analysis Techn. in Phys. Research", PoS(ACAT)046.

[11] T. Golling, these proceedings,
D. Dobos, these proceedings.

[12] P. Billoir, S.Quian, Nucl. Ins. and Meth. in Phys. Res. A311(1992) 139.

[13] R. Frühwirth, Nucl. Ins. and Meth. 225 (1984) 352.

[14] R. Frühwirth et al, Nucl. Ins. and Meth. in Phys. Res. A502 (2003) 699.

[15] K. Prokofiev et al, Proc. of CHEP 2007 #77.

[16] V. Kostyukhin, ATL-PHYS-2003-031.

[17] G. Piacquadio, C. Weiser, Proc. of CHEP 2007 #175.