Development of ATLAS primary vertex reconstruction for LHC Run 3

The ATLAS Collaboration

Increasing luminosity at the Large Hadron Collider (LHC) poses a challenge for primary vertex reconstruction in ATLAS. A rate of 60 or more inelastic proton-proton collisions per beam crossing was observed during the recently-completed Run 2 and even higher vertex density is expected in future runs. To meet this challenge, ATLAS has developed new tools: a Gaussian track density seed finder and an adaptive multi-vertex finder. The former constructs a simple but powerful analytic model of the track density along the beam axis to locate candidate vertices, and the latter applies a global approach to vertex finding and fitting, allowing vertices to compete for nearby tracks.

This note documents the strategy, optimization and preliminary performance of this new vertex reconstruction software, highlighting improvements in vertex finding efficiency, purity and spatial resolution under Run 3 pile-up conditions.

© 2019 CERN for the benefit of the ATLAS Collaboration. Reproduction of this article or parts of it is allowed as specified in the CC-BY-4.0 license.
1 Introduction

Precise reconstruction of primary vertices is of paramount importance in the ATLAS physics programme. Primary vertices are the locations of inelastic proton–proton (pp) collisions - the ultimate origins of all the reconstructed objects used in physics analyses. Accurate and efficient assignment of charged-particle trajectories to the correct primary vertex is essential to reconstruct the full kinematic properties of an interaction. Typically only one pp interaction in a given bunch crossing (often the interaction with the greatest transverse momentum $p_T$, the “hard-scatter”) is of interest, and its reaction products must be isolated from the dozens of others (“pile-up”) occurring simultaneously nearby.

In the years since Run 1, the increase in LHC instantaneous luminosity, and the average number of inelastic pp interactions per bunch crossing, denoted by $\mu$, has degraded vertex-finding performance. In preparation for the even higher vertex densities expected, ATLAS has developed a new Gaussian track density seed finder (GS), recommissioned an adaptive multi-vertex finder and fitter (AMVF) [1, 2], and tuned them to achieve sizeable improvements in vertex reconstruction efficiency and other metrics.

Sections 2 and 3 describe the Inner Detector of the ATLAS experiment and the Monte Carlo (MC) samples used for the present studies. Section 4 reviews the iterative vertex reconstruction strategy used during LHC Runs 1 and 2. Sections 5 and 6, respectively, document development and tuning of the GS and AMVF for Run 3. Section 7 compares the performance of new and old vertex-finding strategies.

2 The ATLAS Detector

ATLAS [3] is a multi-purpose detector at the LHC. The Inner Detector (ID) [4, 5] of the experiment covers the pseudorapidity\(^1\) range $|\eta| < 2.5$. The ID is responsible for efficiently and precisely measuring charged particle trajectories (tracks) in the region closest to the interaction point; these in turn allow reconstruction of primary and secondary vertices. It comprises three subsystems based on different detector technologies: the Pixel Detector, Semiconductor Tracker and Transition Radiation Tracker. The ID is surrounded by a superconducting solenoid providing 2 T axial magnetic field.

During the first LHC Long Shutdown, the Pixel system underwent significant upgrades; a new smaller beam pipe was installed together with an additional Pixel layer, called the Insertable B-Layer [6] which allows measurements closer to the interaction point. In addition, new Pixel Service Quarter Panels [7], providing more robust service connections and allowing the recovery of a number of inactive pixel modules, were installed. No significant changes to the ID are planned for LHC Run 3.

---

\(^1\) The ATLAS experiment uses a right-handed coordinate system with its origin at the nominal interaction point (IP) in the centre of the detector and the $z$-axis along the beam direction. The $x$-axis points from the IP to the centre of the LHC ring, and the $y$-axis points upward. Cylindrical coordinates $(r, \phi)$ are used in the transverse $(x, y)$ plane, $\phi$ being the azimuthal angle around the $z$-axis. Pseudorapidity is defined in terms of the polar angle $\theta$ as $\eta = -\ln \tan(\theta/2)$. 
3 Simulated Data Samples

3.1 Generation

This work is based on MC simulated data samples generated at centre-of-mass energy $\sqrt{s} = 13$ TeV. Each simulated event contains a single $pp$ scattering involving a large-momentum-transfer process referred to as the hard-scatter (HS), overlaid with a Poisson-distributed number of random inelastic $pp$ interactions (pile-up). HS interactions with two very different topologies are generated: top-quark pairs ($t\bar{t}$ in the following) and Higgs bosons produced via Vector Boson Fusion (VBF) and decaying to undetected particles ($\text{VBF } H \rightarrow 4\nu$ in the following). These two signal processes are representative of high- and low-multiplicity HS vertices, respectively. Each sample is produced with the average number of $pp$ interactions per bunch crossing $\langle \mu \rangle = 60$.

The top-quark pairs were generated using POWHEG [8, 9] interfaced to PYTHIA6 [10] with the Perugia 2012 tune [11]. The POWHEG $h_{\text{damp}}$ factor (a model parameter that controls the matching between the matrix element and the parton shower and effectively regulates the high $p_T$ radiation) was set equal to the top mass. The top-quark pairs were generated with a lepton filter, requiring at least one lepton in the final state. The VBF $H \rightarrow 4\nu$ interactions were generated using POWHEG interfaced to PYTHIA8 [12], with the AZNLO tune [13] and the CT10 parameterisation [14] of the parton density functions. All of the overlaid pile-up collisions were simulated using the soft QCD processes of PYTHIA8 with the A3 [15] set of tuned parameters and the NNPDF23LO parton density function set.

All generated events are processed with the ATLAS detector simulation framework [16], using the GEANT4 toolkit [17]. After full detector simulation, the MC events are reconstructed and analysed in the same manner as real data would be.

3.2 Vertex classification

To quantify the performance of vertex finding and fitting, vertices and charged tracks reconstructed from simulated data are compared to the true $pp$ interactions and particles generated in the Monte Carlo event. As described in Ref. [18], reconstructed tracks are associated to truth particles based on correspondence of simulated hits in the tracking detectors, and then traced back to simulated $pp$ interactions using stored truth particle histories. Reconstructed tracks with inconclusive matching information are classified as “fake”, while those not associated with any truth particle ($<1\%$ of all tracks) are ignored.

Each reconstructed vertex is classified based on the true origin, fit weights (see Sections 4.1 and 6.1) and $p_T$ of tracks assigned to it.

- **CLEAN**: At least 70% of the total track weight in the reconstructed vertex originates from a single simulated $pp$ interaction.
- **MERGED**: Less than 70% of the total track weight in the reconstructed vertex originates from any single simulated $pp$ interaction. Hence, two or more simulated interactions contribute significantly to the accumulated track weight.
- **SPLIT**: A single simulated $pp$ interaction contributes the largest track weight to two or more reconstructed vertices. The reconstructed vertex with the largest track $\sum p_T^2$ is classed as either CLEAN or MERGED, whilst the other(s) are labelled SPLIT.
• FAKE: Fake tracks contribute more weight to the reconstructed vertex than any simulated $pp$ interaction.

The quality of HS vertex reconstruction is also used to classify the event as a whole:

• CLEAN/MATCHED: the event contains a CLEAN reconstructed vertex originating from the true HS interaction, and the HS interaction does not contribute more than 50% of the accumulated track weight to any other vertex.

• LOWPU (low pile-up): the event contains a MERGED vertex with at least 50% of the accumulated track weight coming from the simulated HS interaction.

• HIGHPU (high pile-up): the event contains a MERGED vertex with its main contribution coming from a simulated pile-up interaction, and in which the HS interaction contributes between 1% and 50% of the accumulated track weight.

• PUREPU (pure pile-up): the event does not contain any reconstructed vertex with at least 1% accumulated track weight from the HS interaction.

4 Iterative Vertex Reconstruction

The ATLAS iterative vertex finder (IVF) was originally designed [18] for much lower $pp$ interaction multiplicity than expected in Run 3. This section briefly summarizes its features, for comparison with the new tools developed for higher luminosity.

4.1 Strategy

The inputs to vertex reconstruction are the luminous region (“beam spot”) location and size, and the subset of reconstructed tracks passing a quality selection designed to remove poorly-measured and fake tracks. Vertex-finding uses the signed radial and longitudinal impact parameters of reconstructed tracks, $d_0$ and $z_0$ respectively, measured from the center of the luminous region, and their uncertainties, $\sigma(d_0)$ and $\sigma(z_0)$ [19].

Iterative reconstruction proceeds one vertex at a time and involves four steps:

• Seed-finding Tracks not yet assigned to any vertex candidate are analysed to find the most likely position of a new primary vertex. The method is based on a “Fraction of Sample Mode with Weights” (FSMW) [20] mode-finder that recursively scans a weighted histogram of $z_0$ to estimate the point of maximum track density along the beam axis. Figure 1 shows an example of the histogram used in this mode search, along with the true locations and charged particle multiplicities of generated $pp$ interactions, for a Monte Carlo event.

• Track to seed assignment After a seed is found, the set of nearby tracks to fit is chosen based on impact parameter significance. Tracks whose closest approach to the seed is within $12\sqrt{\sigma^2(d_0) + \sigma^2(z_0)}$ are assigned to the vertex fit.
Figure 1: A representative Monte Carlo event as analysed by three different primary vertex seeding methods. The bottom plot shows the simulated locations of $pp$ interactions along the beamline, and the number of associated tracks passing quality cuts. The top three plots compare track density estimators computed by different methods. High densities correspond to possible vertex seed locations. ATLAS used the FSMW mode finder during Run 1 and Run 2. The medical imaging and Gaussian methods were both studied to improve performance at high luminosities.

- **Fitting** The helical parameters of tracks assigned to a candidate vertex are linearized at the seed position and fit using an adaptive Kalman filter [21, 22]. The beam spot location and size are used to constrain the transverse position of the fit, while the longitudinal position is unconstrained. Outliers are progressively de-weighted using a deterministic annealing schedule and the weight function in Equation (1). During annealing, the temperature ($T$) is decremented in stages, increasing the sensitivity of track weights to their compatibility ($\chi^2$) with the vertex position. By the end of the annealing process, when $T = 1$, compatible tracks within three standard deviations (corresponding to $\chi^2_{cut} = 9$) of the fit vertex will have weights close to unity, and incompatible outliers will have weights close to zero.

\[
\omega(\chi^2, T) = \frac{1}{1 + e^{-\frac{1}{2}(\chi^2_{cut}-\chi^2)/T}}
\]  

1

- **Acceptance/rejection** The new vertex candidate is accepted if the number of degrees of freedom (accounting for final track weights) is greater than three and at least two tracks used in the fit have weights greater than 0.01. If the vertex fit is rejected, all tracks tried in the fit are removed from the pool used for future iterations. If the vertex fit is accepted, associated tracks with fit weight greater than 0.01 or impact parameter significance less than 7 $\sigma$ (including both the track and fitted vertex
errors) are removed from the pool of tracks for subsequent iterations; tracks failing both compatibility
cuts are instead removed from the list of tracks associated with the vertex, and remain available to
form future vertices. Thus, regardless of outcome, the pool of remaining tracks for seeding is always
reduced by each iteration.

The IVF terminates when

• less than two tracks remain in the pool for seed-finding, or
• less than two tracks satisfy the seed proximity requirement for fitting, or
• the maximum allowed number of reconstructed vertices (currently 200) is exceeded (this is a failsafe
to handle data quality problems gracefully).

After vertex-finding, the primary vertex with the highest track $\sum p_T^2$ is identified as the default HS vertex.
Specific physics analyses may override this choice based on their own criteria.

4.2 Development

After Ref. [18], two changes to primary vertex reconstruction were made during Run 2. First, the efficiency
for pile-up vertices was slightly improved by eliminating an occasional failure mode in which tracks from a
secondary vertex could generate a spurious primary vertex candidate and cause premature termination of
vertex finding. Second, to reduce average event size on disk, the transverse momentum ($p_T$) threshold for
storing reconstructed tracks was raised from 400 MeV to 500 MeV. For consistency and reproducibility, the
$p_T$ cut used in vertex reconstruction was identically tightened, slightly reducing vertex finding efficiency
for low-multiplicity vertices. Figures 2(a) and (b), respectively, illustrate the effect of these two changes on
the number of reconstructed vertices.

5 Gaussian Track Density Seed Finder

Seed-finding is the first step in vertex reconstruction, and any vertex not (approximately) located by the
seed finder will be lost. The GS was the second of two new approaches studied.

A method based on medical imaging, in which a three-dimensional histogram of the track trajectories in
space is constructed and smoothed by Fast Fourier Transform, was tried first [25]. This method achieved
performance comparable to, but no better than, the mode-finding technique already in use. After gaining
experience with the medical imaging technique, several drawbacks became evident:

• It is a three-dimensional solution to a fundamentally one-dimensional problem.
• There is no obvious way to incorporate individual track measurement errors, which is highly
desirable.
• The calculation is CPU-intensive, non-linear, and must be completely redone when tracks are
removed by assignment to a vertex.
Figure 2: Evolution of vertex reconstruction performance due to software changes during Run 2. In (a), the dashed (solid) line shows the number of reconstructed vertices before (after) eliminating a failure mode which occasionally caused loss of low-multiplicity pile-up vertices [23]. In (b), the dashed (solid) line shows the number of reconstructed vertices before (after) raising the charged-track $p_T$ threshold from 400 MeV to 500 MeV to reduce the average event size on disk [24]. Each figure shows a representative $pp$ data sample reconstructed with two different software versions.

In practice, the Fourier-filtered three-dimensional track image was projected onto the beam axis, and local maxima of the resulting one-dimensional track density histogram were extracted as seed positions. Figure 1 shows an example of such a track density distribution computed by the medical imaging method. These studies suggested the advantages of finding a simpler and more transparent way to produce a similarly smoothed one-dimensional track density estimator directly.

5.1 Gaussian density function

Like the other two seed finders, the GS uses the density of tracks along the beam axis to estimate the most likely position of a $pp$ interaction vertex. Each track is modeled as a correlated radial and longitudinal Gaussian probability distribution $P(r, z)$ centred at $(d_0, z_0)$ and normalized to one, whose shape is characterized by the covariance matrix $\Sigma$ of the corresponding helical parameters:

$$\Sigma = \begin{pmatrix} \sigma^2(d_0) & \sigma(d_0, z_0) \\ \sigma(d_0, z_0) & \sigma^2(z_0) \end{pmatrix}.$$ 

A track’s seed-finding density can be evaluated at any longitudinal position $z$ on the beam axis, and is defined as $\rho(z) = P(0, z)$ where,

$$P(r, z) = \frac{1}{2\pi \sqrt{|\Sigma|}} e^{-\frac{1}{2}((r-d_0)(z-z_0))} \Sigma^{-1}((r-d_0)(z-z_0)).$$
The total density \( W(z) \) as a function of \( z \) is the sum of all nearby tracks:

\[
W(z) = \sum_{i \in \text{tracks}} P_i(0, z).
\]

Figure 1 shows an example of this Gaussian-smoothed track density function for a representative Monte Carlo event.

Neglecting correlations for the sake of illustration and rearranging the terms makes it easier to interpret the resulting simplified \( \rho(z) \):

\[
\lim_{\sigma(d_0; z_0) \to 0} \rho(z) = \left( \frac{1}{\sqrt{2\pi}\sigma(d_0)} e^{-\frac{1}{2}d_0^2/(\sigma^2(d_0))} \right) \left( \frac{1}{\sqrt{2\pi}\sigma(z_0)} e^{-\frac{1}{2}(z-z_0)^2/\sigma^2(z_0)} \right).
\]

In this case, the density function for a track is a normalized one-dimensional Gaussian distribution of width \( \sigma(z_0) \), centred at \( z_0 \) (the second term), with an additional \( z \)-independent multiplicative factor that reduces the track’s contribution based on its \( d_0 \) significance and measurement errors (the first term).

### 5.2 Peak search

The position of the global maximum of \( W(z) \) will be the seed position. The search can rely on the fact that this peak must lie near the location of some track. Further, since the density \( W(z) \) is a sum of known analytic functions, the first and second derivatives of \( W(z) \) at any point can be computed exactly any time \( W(z) \) itself is evaluated. \( W(z) \) and its derivatives are evaluated at the value of \( z_0 \) for each track. If the curvature at that point is negative, indicating we are in the neighborhood of a local maximum, the \( z \) step to the point where \( W'(z) = 0 \) is estimated. If the density function shape is assumed locally parabolic, this is just a Newton step:

\[
\Delta z = -\frac{W'(z)}{W''(z)}.
\]

If the density function shape is assumed locally Gaussian, the step is slightly different:

\[
\Delta z = \frac{W(z)W'(z)}{W^2(z) - W''(z)W(z)}.
\]

The Gaussian approximation gives slightly better convergence, and is used by default. After one step is computed, the trial position is adjusted and the process is repeated a second time, to refine the final location of the local maximum. Further iterations are unnecessary. The global maximum position and height are updated if the current local maximum has superseded them.

The seed finder could find all local maxima and return them in a single call, but instead seed-finding proceeds incrementally, with tracks removed from the seed track collection by the vertex finder in between calls. This requires more CPU time, but helps find seeds from low-multiplicity vertices whose maxima may be shadowed until the tracks from other nearby vertices are removed from the seeding pool. For efficiency, various computed values are cached and reused for each call to find a seed.
5.3 Peak width, error and track association

The Gaussian density functions also allow an accurate estimate of the seed width to be provided to the vertex finder. If the local shape of \( W(z) \) at the global maximum \( z_{\text{max}} \) is assumed Gaussian, then the width of that Gaussian is

\[
\sigma(z) = \sqrt{\frac{W(z_{\text{max}})}{W''(z_{\text{max}})}} \tag{2}
\]

This estimate is used to constrain the vertex fit to the approximate position of the seed, avoiding unwanted migration of the fit when tracks from other nearby vertices might be compatible. Although not yet exploited, this width estimate could allow also more discriminating track/seed association, by tailoring the acceptance window and/or track measurement quality cuts on a seed-by-seed basis.

In a small enough neighborhood of a peak for the Gaussian functions \( P_i \) of each track to be treated as approximately constant, solving \( \partial W / \partial z = 0 \) to find the location of the local maximum is equivalent to computing a weighted average (suppressing track indices for clarity):

\[
\bar{z} = \frac{\sum \tilde{w}(z_0/\sigma^2(z_0) - c^2 d_0/\sigma(d_0, z_0))}{\sum \tilde{w}/\sigma^2(z_0)},
\]

where \( \tilde{w}_i = \frac{P_i(0, \bar{z})}{(1 - c_i^2)} \) is the effective weight at the peak position \( \bar{z} \), and \( c_i = \frac{\sigma(d_0, z_0)}{\sigma(d_0)\sigma(z_0)} \) is the correlation coefficient of the helical impact parameters for track \( i \). The measurement error on this weighted average is then

\[
\sigma^2(\bar{z}) = \frac{\sum(1 + c^2)\tilde{w}^2/\sigma^2(z_0)}{(\sum \tilde{w}/\sigma^2(z_0))^2} \tag{3}
\]

Use of the weighted average and its uncertainty to tighten the longitudinal constraint on the vertex fit, essentially using the standard deviation of the mean instead of the standard deviation itself, will be studied in the future. In addition, the seed finder also knows each track’s contribution to the global maximum and this information could be used in place of a significance-based track to seed assignment cut.

6 Adaptive Multi-Vertex Reconstruction

The AMVF extends the adaptive (single) vertex fitter by allowing tracks to have weight in the fits of more than one nearby vertex. Through deterministic annealing, the weights generally evolve so that each track eventually only has non-negligible weight in a single compatible vertex. During this process, the vertex candidates compete for tracks by adjusting their positions. This technique was developed early in ATLAS but never used in production. It was originally recommissioned for studies of the phase-2 Pixel Detector upgrade, and showed considerable promise [26].
6.1 Strategy

The outline of the multi-vertex finder is identical to that of the IVF (see Section 4) but the steps themselves are quite different:

- **Seed-finding** Tracks not yet assigned to any vertex candidate are analysed to find the most likely position of a primary vertex using the GS (Section 5).

- **Track to seed assignment** After a seed is found, the set of nearby tracks to fit is chosen. One essential difference from the iterative single-vertex strategy is that all tracks passing the quality selection (not only unassigned tracks, but also tracks already assigned to one or more previously accepted vertex candidates) are eligible for assignment to new vertex candidates. Thus, unlike the IVF, each track may, and generally will, be assigned to multiple vertices. The track assignment criteria and their tuning are detailed in Section 6.2.

- **Fitting** The linearized helical parameters of assigned tracks are used to fit the position of the vertex candidate with a weighted adaptive Kalman filter, subject to transverse and longitudinal constraints provided by the beam spot and seed positions, respectively. Another important difference from the iterative strategy is that each time a new vertex candidate is fit, all other candidates linked to it (through a chain of tracks and vertices of any length) are also simultaneously refit.² Track weights \( \omega \) with each vertex \( i \) are annealed in six steps based on compatibility \( (\chi^2) \). After each decrease of the temperature \( T \), weights are recalculated and all connected vertices are refit. The total weight for a track across all assigned vertices is normalized to one, but for purposes of weight normalization (only) tracks are also given a notional three standard deviation (corresponding to \( \chi^2_0 = 9 \)) compatibility with “unassigned”:

\[
\omega_i(\chi^2_i, T) = \frac{e^{-\frac{1}{2}\chi^2_i/T}}{\sum_j e^{-\frac{1}{2}\chi^2_j/T} + e^{-\frac{1}{2}\chi^2_0/T}}
\]

- **Acceptance/rejection** The new vertex candidate is accepted if it satisfies three criteria. First, it must have at least two compatible tracks from the seeding track pool; a new vertex cannot be formed from only tracks that are compatible with some previously-found vertex. Track compatibility is defined as a \( \chi^2 \) probability greater than \( 10^{-4} \). Second, the new vertex’s fit position may not be within 3 \( \sigma \) of any other, based on the calculated fit errors of both vertices. Third, the weighted average of track weights in the fit \( (\sum \omega^2 / \sum \omega) \) must be greater than \( \frac{1}{2} \). If the vertex candidate is accepted, all compatible tracks are removed from the seeding pool. If the vertex candidate is not accepted, only the most compatible track is removed from the seeding pool. Thus, regardless of outcome, the pool of remaining tracks for seeding is always reduced by each iteration, ensuring eventual termination of vertex-finding.

Adaptive multi-vertex finding is complete when

- less than two tracks remain in the pool for seed-finding, or
- the seed finder is unable to return a seed, or
- the maximum allowed number of iterations is exceeded.

² Thus, if track 1 is assigned to both the new candidate and some other vertex A, and track 2 is shared between vertex A and vertex B, all three vertices will be (re)fit.
6.2 Development and tuning

Preliminary studies revealed that AMVF performance depended very strongly on the track/seed assignment criteria. In particular, loosening the assignment cut reduced the vertex-finding efficiency. This was unexpected, since the fitter is intended to handle cases where tracks are assigned to several possible vertices, yet initially the best performance was achieved by essentially disabling that feature. This section explains the investigation, understanding and correction of the problem, and optimization of the track assignment and other parameters of the AMVF for data-taking around \( \mu = 60 \).

The AMVF’s original track to seed assignment criterion was extremely simple - tracks with longitudinal impact parameters in a fixed-width, \( \Delta z = \pm 5 \) mm window were assigned to each new vertex seed, and included in the eventual fit. This very loose assignment was consistent with the philosophy of the fitter: to allow the fit itself to decide, through the adjustment of track weights during annealing, which tracks belong with which vertices. If the fitter worked as intended, the only drawback of this loose cut should be increased computational expense as more vertices become correlated by sharing tracks. Instead, large efficiency increases resulted from reducing this acceptance window to \( \Delta z = 0.5 - 1 \) mm. In effect, the “multi-vertex” fitter preferred to operate more like a single-vertex fitter, with as little sharing of tracks as possible. Valid seeds, which should have been reconstructable, were somehow being lost. Figure 3 shows the AMVF \( \bar{t}t \) performance as a function of \( \mu \) for two values of \( \Delta z \). For lower values of \( \Delta z \), more vertices are found, but many more interactions are split, with their tracks shared among several reconstructed vertices.

![Figure 3](image-url)

Figure 3: A comparison of reconstructed vertex classification for the AMVF with different acceptance windows \( \Delta z \). In (a) an acceptance window of \( \Delta z = 0.5 \) mm is used. In (b) an acceptance window of \( \Delta z = 3 \) mm is used.

To better understand the behavior, when a new seed fell within the track assignment window \( \Delta z \) of an existing vertex, its initial and final fit positions (\( z_i \) and \( z_f \), respectively) were compared to the nearby neighbor’s location on either or both sides (\( z_+ \) and/or \( z_- \)). A simple ratio measuring how far the new vertex moved toward its neighbors was computed:

\[
R_\pm = \frac{z_f - z_i}{z_\pm - z_i}.
\]

Values of \( R \) near 0 mean that the vertex remained close to its seed position, while values near 1 indicate that the vertex moved to the location of one of its neighbors. As Figure 4(a) shows, new candidates were
frequently migrating to their neighbors when track assignment windows overlapped. The incidence of
migration increased sharply as the $\Delta z$ window widened and overlaps became more frequent.

Careful scrutiny of the fitter logic revealed the cause: the annealing process begins by assigning all tracks
nearly equal weights. Imagine vertex A, with 40 tracks, and nearby vertex B, with 20 tracks. If the two
vertices are separated by less than the width of the track assignment window, both fits will include all 60
tracks. Since the fits begin with all tracks having nearly equal weights, both fits will prefer the location of
the 40-track vertex, and moreover, with a loose $\Delta z$ window, this will occur even if none of the tracks from
vertex A are compatible with vertex B, and vice versa.

Several approaches to preventing this problem were studied:

- **Significance-based assignment** The fixed-width acceptance window treats well-measured and poorly-
measured tracks identically. A cut based on impact parameter significance is a natural refinement.
Significance is defined as:

$$s = \sqrt{(d_0/\sigma(d_0))^2 + (z_0/\sigma(z_0))^2}.$$ 

In practice, a loose absolute $\Delta z$ cut is retained, to prevent poorly-measured tracks from being assigned
to large numbers of distant vertices.

- **Annealing schedule** Another way to reduce migration is to make the initial track weights more
sensitive to compatibility with the seed. To study this, an adjustable “annealing temperature power”
parameter (ATP) was introduced. Each temperature in the annealing schedule is raised to this power,
so values less than one result in lower annealing temperatures and greater sensitivity of weights to
vertex compatibility at each stage except the last.

- **Longitudinal constraint** The fitter has always used an $(x, y)$ constraint from the beam spot. A $z$
constraint is easily added, since the constraint is simply the initial Kalman filter state vector and its
covariance. The GS provides either the width, Equation (2), or the error on the centroid, Equation (3),
of each seed to adjust the strength of the constraint. The former, which is the looser of the two
possibilities, was used in these initial studies.

All were these changes were successful to some extent, but the last eliminated the problem entirely. Figure 4(b)
shows the migration metric $R$ with the optimized significance cut ($s < 5$), stiffened annealing
schedule (ATP = 0.5) and longitudinal constraint. Figure 5 compares the logic of the IVF and tuned
AMVF.

7 Performance

This section compares vertexing quality, efficiency, spatial resolution and track association for old and
new strategies, using the simulated $t\bar{t}$ and VBF $H \rightarrow 4\nu$ data described in Section 3. To generalize these
$\langle \mu \rangle = 60$ samples to a wider range of pile-up conditions, performance is sometimes shown as a function
of the local pile-up density near a vertex, defined as the number of generated vertices per unit length within a
symmetrical $\pm 2$ mm longitudinal window. The vertex and event classification categories from Section 3.2,
respectively, are used to quantify success in isolating individual $pp$ interactions, and the HS interaction
specifically.
Figure 4: Vertex migration metric $R$ (a) before and (b) after introduction of a track assignment cut on impact parameter significance ($s$), stiffer annealing temperature power (ATP), and a longitudinal constraint.

Figure 5: Comparison of the (a) IVF and (b) AMVF logic.
7.1 Merge probability

The AMVF’s improved performance is primarily due to its reduced tendency to merge nearby vertices. Figure 6 shows the longitudinal separation for nearby reconstructed vertices in simulated $t\bar{t}$ events, and highlights the AMVF’s superior ability to resolve vertices separated by a millimeter or less. The depletion around $\Delta z = 0$ is due to merging. The AMVF is better able to resolve vertices with smaller separations, resulting in fewer merged vertices. The small excess at $\Delta z = 0$ for the IVF is due to split vertices, which are absent with the AMVF.

![Graph showing distribution of longitudinal separation between pairs of nearby reconstructed primary vertices in simulated $t\bar{t}$ events, for the IVF and AMVF.]

Figure 6: Distribution of the longitudinal separation between pairs of nearby reconstructed primary vertices in simulated $t\bar{t}$ events, for the IVF and AMVF.

7.2 Hard-scatter vertex quality

Figure 7 compares the quality of HS vertex reconstruction and matching for the IVF and AMVF at $\langle \mu \rangle = 60$, showing the fraction of events in each HS classification category. The fraction of events with a CLEAN/MATCHED HS vertex is 5% higher for the AMVF.
7.3 Hard-scatter efficiency

Following Ref. [18], the HS reconstruction efficiency is the fraction of events in which the HS vertex is reconstructed and classified as CLEAN/MATCHED, LOWPU or HIGHPU. Figure 8 shows that the AMVF improves on the IVF’s already-high efficiency for $t\bar{t}$, and recovers about half the inefficiency of the IVF for the more challenging VBF $H \rightarrow 4\nu$ channel. The largest gains are in regions of highest vertex density, and the AMVF reduces pile-up dependence of the efficiency.

The HS selection efficiency is the fraction of events in which the highest $\sum p_t^2$ vertex contains the largest total weight from true HS tracks. Figure 9 compares selection efficiency as a function of local pile-up density for $t\bar{t}$ and VBF $H \rightarrow 4\nu$. The AMVF improves the selection efficiency for both $t\bar{t}$ and VBF $H \rightarrow 4\nu$, eliminating the dependence on pile-up density.

7.4 Hard-scatter spatial resolution

Figures 10 and 11 show the radial and longitudinal spatial resolutions calculated as the difference between the true and reconstructed vertex position in the MC simulation. Due to the beam spot constraint, little improvement is expected or observed in the radial direction, where the differences are sub-micron level. In the $z$ direction, the AMVF achieves 20% (10%) better resolution for $t\bar{t}$ (VBF $H \rightarrow 4\nu$).
Figure 8: Comparison of IVF and AMVF HS vertex reconstruction efficiency as a function of local pile-up density, for (a) $t\bar{t}$ and (b) VBF $H \rightarrow 4\nu$. The reconstruction efficiency is the fraction of events where the true HS vertex is successfully reconstructed (classified as CLEAN/MATCHED, LOWPU or HIGHPU).

Figure 9: Comparison of IVF and AMVF HS vertex selection efficiency as a function of local pile-up density, for (a) $t\bar{t}$ and (b) VBF $H \rightarrow 4\nu$. The selection efficiency is the fraction of events where the reconstructed vertex with highest $\sum p_T$ is the one containing the largest total weight from true HS tracks.
Figure 10: Comparison of IVF and AMVF HS vertex radial resolution as a function of local pile-up density for (a) $t\bar{t}$ and (b) VBF $H \rightarrow 4\nu$, obtained from the difference between the generator-level information and reconstructed primary vertex position in MC simulation.

Figure 11: Comparison of IVF and AMVF HS vertex longitudinal resolution as a function of local pile-up density for (a) $t\bar{t}$ and (b) VBF $H \rightarrow 4\nu$, obtained from the difference between the generator-level information and reconstructed primary vertex position in MC simulation.
7.5 Hard-scatter track efficiency and purity

Figure 12 shows the number of compatible ($\chi^2 < 9$) reconstructed tracks from the true HS interaction which are correctly assigned to the identified HS vertex, as a function of $\eta$. In the central region ($|\eta| < 1.25$) the AMVF and IVF have identical efficiency for $\bar{t}t$, while the AMVF correctly assigns 2% more HS tracks in VBF $H \rightarrow 4\nu$. At large $|\eta|$, the IVF tends to assign poorly-measured forward tracks to the first vertex they are compatible with (often the high-multiplicity HS), while the AMVF assigns them to the vertex they are most compatible with. As a result, the AMVF is less efficient in the forward region. Figure 13 shows the rate of HS contamination by incorrectly associated tracks from pile-up interactions. For both $\bar{t}t$ and VBF $H \rightarrow 4\nu$, the AMVF reduces pile-up track contamination by 25-70% over the entire range of $\eta$. 

Figure 12: Comparison of IVF and AMVF HS track efficiency as a function of $\eta$ for simulated (a) $\bar{t}t$ and (b) VBF $H \rightarrow 4\nu$ samples. The plot shows the number of correctly associated compatible reconstructed tracks per $\eta$ bin originating from the true HS vertex.

Figure 13: Comparison of IVF and AMVF HS track contamination as a function of $\eta$ for simulated (a) $\bar{t}t$ and (b) VBF $H \rightarrow 4\nu$ samples. The plot shows the number of compatible reconstructed tracks per $\eta$ bin originating from pile-up which are incorrectly associated with the HS vertex.
7.6 Pile-up vertex quality

Figure 14 compares the number of reconstructed vertices with each quality classification, for the IVF and AMVF at \( \langle \mu \rangle = 60 \). The AMVF reconstructs more CLEAN and MERGED vertices, with an average of one additional SPLIT vertex per event as well.

![Number of reconstructed vertices with each quality classification, for the IVF and AMVF at \( \langle \mu \rangle = 60 \).](image)

Figure 14: Number of reconstructed vertices with each quality classification, for the IVF and AMVF at \( \langle \mu \rangle = 60 \).

7.7 Pile-up efficiency

Figure 15 compares the average number of reconstructed vertices as a function of \( \mu \) in simulated \( t\bar{t} \) events, for the IVF and the AMVF. The upper dashed line shows the limit of 100% efficiency, while the lower dashed line shows what is theoretically achievable with the available reconstructed tracks after track quality selections. The number of AMVF vertices classed as CLEAN, MERGED, SPLIT and FAKE is also shown. At high \( \mu \), the AMVF recovers 35 – 50% of the reconstructable primary interactions lost by the IVF.

8 Conclusion

Future work will aim to fully exploit the power of the analytic seed-finding model and to diagnose and eliminate remaining sources of inefficiency in the vertex finder. Computational efficiency, which has not been a priority while validating the results, will eventually receive attention.
Figure 15: Average number of vertices reconstructed as a function of the number of $pp$ interactions per bunch crossing ($\mu$), in simulated $t\bar{t}$ events. For reference, the upper dashed line corresponds to perfect reconstruction efficiency, while the lower dashed line is a more conservative estimate of the maximum efficiency possible given the reconstructed tracks available to the vertex finder. Error bars on the data points are statistical uncertainties. Filled circles show the classification of AMVF vertices as CLEAN, MERGED, SPLIT and FAKE.

The Gaussian track density seed finder and adaptive multi-vertex fitter presented here outperform the previous ATLAS vertex reconstruction strategy, particularly in high pile-up environments where the improvement will be most relevant for Run 3 and beyond. Near $\langle \mu \rangle = 60$, 35 – 50% of the pile-up vertex and VBF $H \to 4\nu$ HS reconstruction inefficiency has been recovered.
References

[1] G. Piacquadio, K. Prokofiev and A. Wildauer, 
*Primary vertex reconstruction in the ATLAS experiment at LHC*, 


url: http://cds.cern.ch/record/331063.

url: http://cds.cern.ch/record/331064.


[9] S. Frixione, P. Nason and C. Oleari, 
*Matching NLO QCD computations with Parton Shower simulations: the POWHEG method*, 


[12] T. Sjöstrand et al., *An Introduction to PYTHIA 8.2*, 


*Phys. Rev. D* **82** (7 2010) 074024, 


