FAST TRACKING FOR THE ATLAS LVL2 TRIGGER

N. Konstantinidis*, M. Sutton, University College London, London, UK
J. Baines, D. Emeliyanov, Rutherford Appleton Laboratory, Chilton, Didcot, UK
F. Parodi, C. Schiavi, INFN, Genova, Italy
H. Drevermann, CERN, Geneva, Switzerland

Abstract

We present a set of algorithms for fast track reconstruction at the second level (LVL2) trigger of ATLAS, using three-dimensional space points ($\equiv$ hits) from the silicon trackers. The strategy is to determine the position z_0 of the interesting pp interaction along the beam axis prior to any track reconstruction and then retain only groups of hits which point back to that z_0 and perform combinatorial tracking only inside those groups. We give results and discuss the advantages of this approach, which is generic enough to be applicable to other multi-collision experiments. We also make a qualitative comparison with a complementary approach which is based on Look-Up Tables (LUT) and is also used in ATLAS.

INTRODUCTION

Online event selection is one of the major challenges for the LHC experiments, which are required to achieve a rejection of almost six orders of magnitude to bring the event rate from 40 MHz down to a manageable 200 Hz or so. To make things even more challenging, in addition to the pp interaction leading to the interesting physics process, i.e. the one responsible for the LVL1 accept (referred to as the *physics interaction* or *physics event* hereafter), several minimum bias interactions (referred to as the *pile-up interactions* or *pile-up events* hereafter) will occur at the same bunch crossing ($\sim$5 at "low" luminosity 2×10^{33} cm^{-2}s^{-1}, $\sim$25 at the design or "high" luminosity of 10^{34} cm^{-2}s^{-1}). This adds significantly to the complexity of the events.

The consequences are particularly severe for the tracking detectors and hence the corresponding algorithms. A typical ATLAS event at the LHC design luminosity contains about 20000 silicon hits, the majority of which come from low p_T tracks curling inside the trackers. Only a few percent of the hits are due to tracks from the physics interaction. This high hit occupancy, especially in the innermost tracking layers, has a cost both in tracking performance (due to hit mis-association) and in the execution times (due to combinatorics), a rather undesirable feature for triggering.

Overview of the ATLAS Trigger

The ATLAS Trigger System features a (typical in HEP) three-level architecture [1]. The first level (LVL1) trigger is based purely on custom-made hardware and uses reduced granularity information from the calorimeters and special muon trigger chambers. The decision for an event must be reached within about 2 μs, the latency of the LVL1 Trigger. The maximum sustainable output rate from LVL1 is 75 kHz (upgradeable to 100 kHz). When an event is accepted by LVL1, the data from this event are passed from the front-end electronics of the various sub-detectors to specially designed ReadOut Buffers (ROBs), where they remain (and can be accessed) during LVL2 processing. Event selection at LVL2 is based on specialised software algorithms, running on dedicated PC farms. The average processing time per event is expected to be around 10 ms. LVL2 is expected to bring the event rate down to about 1 kHz. If an event is accepted by LVL2, the corresponding data fragments from all ROBs are sent to the Event Builder, which builds the complete event and forwards it to the PC farm of the third Trigger selection stage, the Event Filter (EF). The average processing time at the EF will be on the order of a few seconds. At this stage, full calibration and alignment information is available for the data, which (together with the increased latency) allows the execution of precise offline-like algorithms for the final online selection. The LVL2 and Event Filter together are referred to as the ATLAS High Level Trigger (HLT). Overall, the HLT is expected to achieve an event rejection factor similar to LVL1 (i.e. about 500).

A novel feature in the LVL2 Trigger of ATLAS, which is key to the overall success of the Trigger, is that processing will be restricted to Regions of Interest (RoI). These are the regions of the detector where the LVL1 Trigger found some activity which lead to accepting an event. On average, there are about 2 RoIs per LVL1 accepted event and the average size of an RoI is about 2% of a total event. As a result, ATLAS will be able to minimise both the amount of data transfered from the ROBs to the LVL2 processors (hence the network traffic) and the processing time at LVL2. The size of the RoI depends on the physics signature. A typical size for an RoI containing a high-p_T electron is ($\Delta\eta = 0.2$, $\Delta\phi = 0.2$, $\Delta z = 22.4$ cm) and has the shape shown in Fig. 1b,c.

Tracking at the LVL2 Trigger

LVL2 is the earliest stage where (a) the data from the tracking detectors are available and (b) it is possible to combine information from different sub-detectors. These two features in addition to the availiability of full granularity calorimeter and precsion muon information, are responsible for the rejection power at LVL2.

---

* n.konstantinidis@ucl.ac.uk

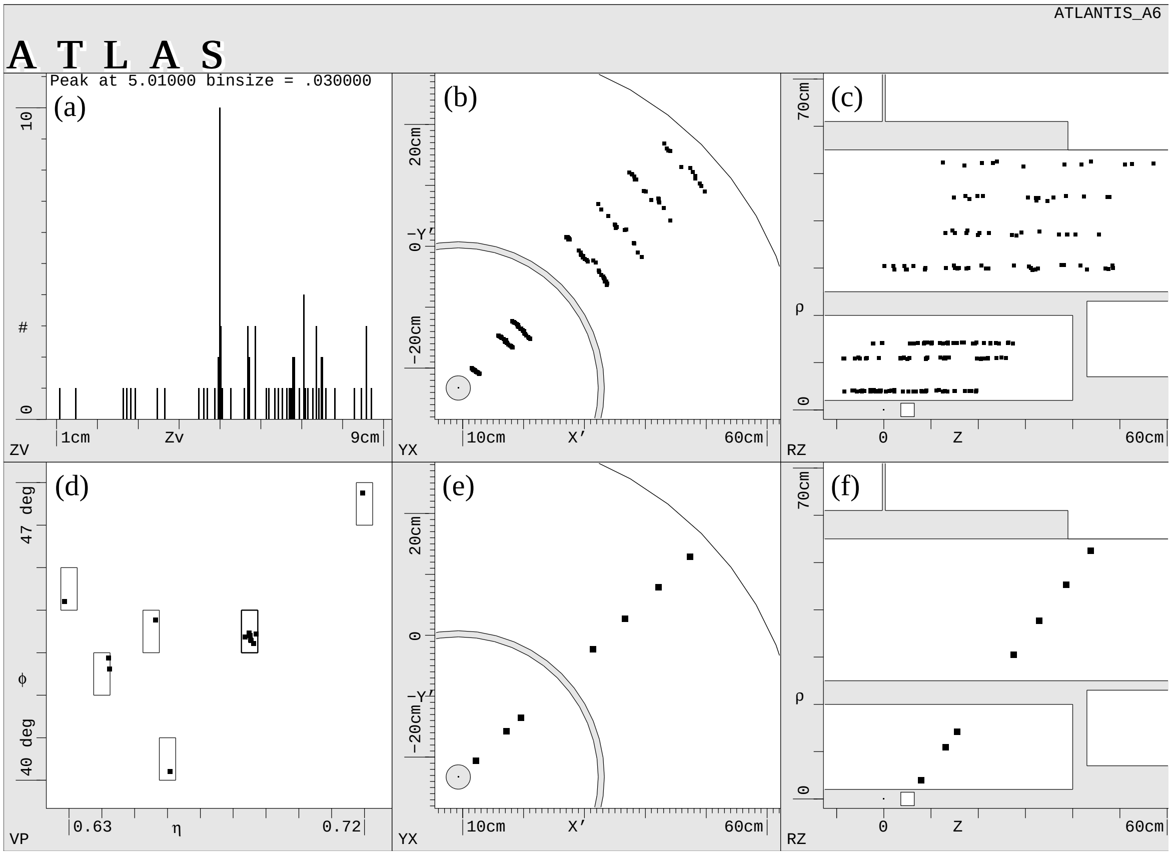

Figure 1: From an electron RoI: (a) the z-histogram (shown only around the initial z-position of the electron; the rest is flat), (b,c) x-y and ρ-z views of the RoI before pattern recognition, (d) the part of the 2D-histogram in (η, ϕ) containing the electron (only bins containing hits are drawn; the hits from the electron are all concentrated in one bin) and (e,f) x-y and ρ-z views of the RoI after the HitFilter.

Tracking is needed at LVL2 for verifying several signatures, each posing different requirements:

- Identification of high p_T electrons, muons and taus. High efficiency is required to reconstruct tracks which are then matched to information from outer detectors.
- B-Physics. High efficiency is required down to low p_T ($\sim$2 GeV/c) for the reconstruction of exclusive ($B \rightarrow \pi\pi$) or semi-inclusive ($J/\Psi \rightarrow \mu\mu$, ee) B hadron decays.
- Inclusive b-jet identification may be important if new physics appears in fully hadronic final states (such as $H \rightarrow hh \rightarrow b\bar{b}b\bar{b}$ in certain supersymmetric models). High purity tracking is required down to low p_T, to avoid fake tracks which may give fake b-signatures.

THE ALGORITHMS

As mentioned already, the main difficulty for performing tracking fast and efficiently is the large number of hits from tracks due to the pile-up interactions. The idea behind the algorithms presented in the following is to exploit the differences between the physics and pile-up interactions and reduce the number of hits with which combinatorial tracking has to be performed.

The main two differences between the physics event and pile-up are (a) they happen at different positions along the beam line ($\sigma_z \approx 6$ cm); and (b) tracks from the physics interaction have, on average, higher p_T than those from pile-up.

The zFinder

The general principle of the algorithm [2] is summarised in the following steps:

- The RoI is divided into many small slices in ϕ.
- In a given ϕ slice, each pair of hits from different layers is used to calculate a z by linear extrapolation to the beam line (this assumes a solenoidal magnetic field, where the helix trajectories of charged tracks are straight lines in the ρ-z projection). In order to avoid loss of efficiency due to binning effects (tracks on the boundary of bins would give fewer hit pairs)hit-pairs are formed from all the hits in two neighbouring phi slices.
- A one-dimensional histogram is filled with the z calculated for each hit pair.

- The z position of the physics event is taken to be the one corresponding to the z-histogram bin with the maximum number of entries.

An example of a z-histogram from an electron RoI is shown in Fig. 1a.

The key point of the algorithm is the division of the RoI in small ϕ slices. This has a double benefit: (a) it reduces drastically the combinatorics, hence minimising the quadratic time behaviour of the algorithm; and (b) it gives naturally more weight to high $\mathrm{p_T}$ tracks and therefore to the physics event as opposed to pile-up. This is because, in general, all hits from a high $\mathrm{p_T}$ track will be in the same (or adjacent) phi-slice(s), whilst those from a low $\mathrm{p_T}$ track will be distributed over several phi-slices, so the number of hit-pair combinations giving entries to the z-histogram will be significantly larger for the high $\mathrm{p_T}$ track.

The HitFilter

The hit filtering algorithm is based on the fact that all hits of a track of sufficiently high $\mathrm{p_T}$ are contained in a small solid angle in (η, ϕ) with its apex at the position of the primary interaction, in contrast to hits from tracks originating from different z positions. The principle of the algorithm can be described in the following steps:

- Given the z-position of the physics event, the η of all hits is calculated and a 2D-histogram in (η, ϕ) is filled.
- In each (η, ϕ) bin, the number N_L of different detector layers containing hits is counted. If N_L is above a given threshold all the hits in this bin are accepted, otherwise they are rejected.
- Hits from neighbouring bins are clustered into groups (this is done to eliminate binning effects). Often, a group contains the hits of just one track.

As can be seen, no hit combinations are used in the HitFilter and hence the execution time of the algorithm scales linearly with the hit occupancy. The size of the bins in η and ϕ can be adjusted according to the physics case. The size in η depends on the detector resolution in the z coordinate and the resolution on the reconstructed z position of the physics event. The size in ϕ determines a $\mathrm{p_T}$ cut-off, below which a track spans into many bins in ϕ and thus the algorithm starts to become inefficient. In ATLAS, for an η bin size of 0.004 and a ϕ bin size of 2°, the algorithm is essentially 100 % efficient for tracks with $\mathrm{p_T} > 2$ GeV/c, which is a cut-off commonly used for triggering.

About 95 % of the hits are rejected by the HitFilter in high luminisity RoIs, and the selected hits are returned in groups. Hence, the subsequent pattern recognition is facilitated since it can be performed in individual groups.

The GroupCleaner and Track fitting

A group resulting from the HitFilter may contain the hits from more than one track and/or random hits. The GroupCleaner exploits the fact that any triplet of hits from the same track can be used to extract the track parameters in the transverse plane: ϕ_0, 1/$\mathrm{p_T}$ and $\mathrm{d_0}$. The algorithm proceeds in the following steps:

- For all hit triplets satisfying basic selection criteria, the transverse track parameters are calculated and a 2D-histogram in (ϕ_0, 1/$\mathrm{p_T}$) is filled.
- The same procedure is followed as in the HitFilter, i.e. bins containing hits in more than a certain number of different layers are the signature for tracks. After clustering of neighbouring bins, the set of hits contributing to those bins forms a track candidate.
- Post-processing of tracks sharing hits: if track candidates share more than a certain number of hits, the longest one is retained and the rest are removed.

Finally, the track candidates are passed to a track fitter[4]. The fitter employs the extended Kalman filter which estimates the track parameters at perigee point only. Due to this, the track state propagation in the filter is reduced to the corrections due to the material effects, thus, resulting in significant speed-up of the fitter. The filter works with two types of input measurements – hits on the barrel surfaces and hits on the endcap discs orthogonal to the beam-line. During filtering, the algorithm calculates the expected χ^2 contribution to the fit for each hit and removes hits with large χ^2 values. The initial estimates of the track parameters are provided by the GroupCleaner. The Kalman filter algorithm is implemented using an object-oriented approach. The filtering process is performed by a set of filtering nodes – C++ objects capable of updating a track state by running their own dedicated (either "barrel" or "endcap") Kalman filters. Each filtering node is an instance of a C++ class derived for a specific measurement model from the common base class. The numerical operations necessary for the track state update are distributed between the derived and base classes so that those depending on the measurement model and, thus, amenable for the specific optimization, are performed by the methods of the derived class while all the generic operations are delegated to the base class. Such a distributed design allows deep optimization of the Kalman filter mathematics, in particular, model-dependent gain matrix calculation.

Performance results

We give here the results from benchmark studies demonstrating that the performance of the algorithms satisfies the requirements of the ATLAS LVL2 Trigger.

Using these algorithms in high-$\mathrm{p_T}$ (=40 GeV/c), single electron RoIs at high luminosity, where there are on average about 200 hits, the average execution time was estimated to be 1 ms (on a PIII 1 GHz processor). The resolution of the $\mathrm{z_0}$ from the zFinder is 200 μm and the overall efficiency for reconstructing the electron track is $\sim 95\%$.

Table 1 presents the results for a benchmark B-physics channel, $\mathrm{B_s} \rightarrow \mathrm{D_s}(\phi\pi)\pi$. In addition, the table shows the

Table 1: Comparing the performance of the algorithms when (3) or (2) pixel layers are used.

$B_s \to D_s(\phi\pi)\pi$	**(3)**	**(2)**
Signal efficiency (%)	68.7	68.0
Eff. wrt offline (%)	78.4	78.5
Bkg efficiency (%)	3.5	3.8

results of comparing the performance of the algorithms in the complete pixel system (three layers) and the initial layout (two layers). The only change that had to be made in the algorithms was to require hits in 4 (out of 6) layers in the second case, instead of 5 (out of 7). It can be seen that there is hardly any degradation in the signal efficiency and only a slight increase in the background level. This demonstrates that the algorithms are flexible and robust.

The linear time behaviour of the algorithms is shown in Fig.2, where the execution time is plotted as a function of the number of hits in the silicon trackers. This is one of the most elegant features of these algorithms.

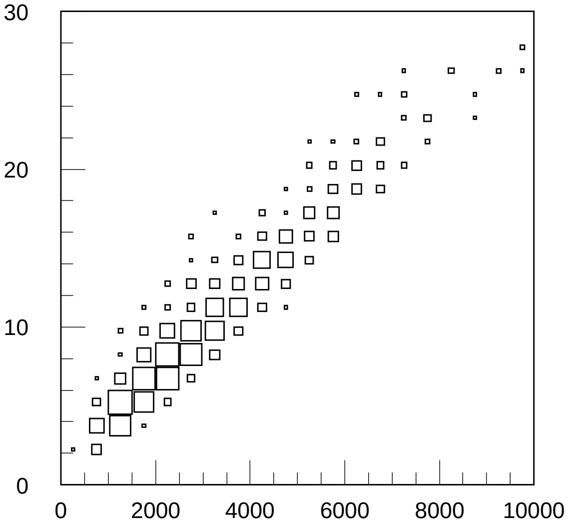

Figure 2: Execution time as a function of the number of hits for full reconstruction of the silicon trackers in B-physics triggers at low luminosity.

An alternative LUT-based approach

An alternative approach has been developed in ATLAS [5], employing LUTs to reduce combinatorics and using space points from only three layers of the silicon Trackers. The algorithm can be configured to use any three logical layers, with the first one always being the innermost pixel layer. The actual configuration can be chosen to optimise the performance for a specific physics signature.

The strategy is to build first track seeds from pairs of hits, with one hit always from the innermost pixel layer and a LUT connecting this layer with the next one out. The track seeds which pass certain selection criteria (e.g. minimum p_T) are then used to determine the z_0 of the interesting interaction by histogramming their extrapolated z positions along the beam axis as discussed above (several z_0 values are retained for better tracking efficiency). Track seeds consistent with the calculated z_0 are then extended further to a third layer, using another LUT, ambiguities are removed, and the remaining triplets form the track candidates, which are then fitted with a circle in the transverse plane and a straight line in (ρ, z).

This approach has given similar results to those described above, both in terms of physics and timing performance, and is in many ways complementary since it has different strengths and weaknesses. Most notably, it can be less sensitive to (or totally independent of) the SCT hardware performance, data quality and relative alignment to the Pixel detectors. On the other hand, it depends more critically on the good performance of the Pixel detector.

CONCLUSIONS

Track reconstruction at the ATLAS LVL2 Trigger is a challenging task, given the high occupancy of the tracking detectors due to the pile-up.

The strategy we have presented in this paper aims at rejecting most pile-up hits prior to track reconstruction, hence avoiding (almost) completely any combinatorial search until one is left only the interesting hits in small groups.

We have presented results demonstrating that this approach, implemented in the algorithms described here, give very satisfactory answer to the requirements of the ATLAS LVL2 Trigger. The algorithms are conceptually simple, flexible and robust and one of their most attractive features for online use is that they scale linearly with the hit occupancy in the silicon trackers.

The two complementary approaches presented provide the necessary flexibility to adapt to different conditions (machine and detector) especially at start-up and allow optimization of the trigger selections for specific physics signatures.

REFERENCES

[1] The ATLAS HLT-DAQ-DCS Technical Design Report (TDR), available from http://atlas-proj-hltdaqdcs-tdr.web.cern.ch/.

[2] N. Konstantinidis and H. Drevermann, ATLAS note ATL-DAQ-2003-034.

[3] TDR page 228 and references therein.

[4] D. Emeliyanov, ATLAS internal communication ATL-COM-DAQ-2004-012.

[5] TDR page 229 and references therein.