PROCEEDINGS OF SCIENCE

Impact of Tracker Layout and Algorithmic Choices on Cost of Computing at High Pileup

Vyacheslav Krutelyov[*]
University of California at San Diego
E-mail: Vyacheslav.Krutelyov@cern

Giuseppe Cerati
University of California at San Diego, FNAL

Matevz Tadel
University of California at San Diego

Frank Wuerthwein
University of California at San Diego

Avi Yagil
University of California at San Diego

High luminosity operation of the LHC is expected to deliver proton-proton collisions to experiments with average number of proton-proton interactions reaching 200 every bunch crossing. Reconstruction of charged particle tracks with current algorithms, in this environment, dominates reconstruction time and is increasingly computationally challenging. We discuss the importance of taking computing costs into account as a critical part of future tracker designs in HEP as well as the importance of algorithms used.

38th International Conference on High Energy Physics
3-10 August 2016
Chicago, USA

[*]Speaker.

PoS(ICHEP2016)194

1. Introduction

The Large Hadron Collider (LHC) operating at CERN is the primary facility for the experimental high energy physics at the energy frontier since 2009. It is expected to dominate the field for the next two decades or so [1]. General purpose collider detectors ATLAS and CMS are used to collect proton-proton collision data from the LHC and are expected to continue their operation on this time scale. The ATLAS and CMS collaborations operating these detectors have been analyzing collected data for variety of physics analyses and publish results in about a hundred publications each year. Among the most known results is the discovery of the Higgs boson in 2012. Computing resources needed to process, store, and analyze the data towards the publication play a large role in the experiment ecosystem and carry a significant fraction of the total operating budget.

The LHC has reached its nominal design luminosity in 2016 and operates at proton beam energy of 6.5 TeV since 2015. It is expected to reach its nominal design beam energy of 7 TeV as early as in 2018, towards the end of Run 2 period. Towards the end of 2016 the colliding beams had an average rate of about 50 proton-proton collisions per bunch crossing (pileup) at the beginning of the fills. The ATLAS and CMS detector data logging rate was about one kHz in each experiment. This data is processed offline first by event reconstruction algorithms, which deliver physics objects suitable for analyses or further post-processing. Among the algorithms, charged particle tracking takes more than a half of the total event reconstruction time. In order to process data at this rate and cope with the pileup conditions the experiments have already made major updates to the offline reconstruction algorithms in preparations for Run 2 with significant algorithmic and physics selection optimizations, most notably for needs of the charged particle tracking [2, 3].

In the coming years the instantaneous luminosity at LHC is expected to increase, with the largest increase coming after the transition to the high-luminosity LHC (HL-LHC) [1]. During HL-LHC operation the ATLAS and CMS experiments are expected to collect collision data with mean pileup reaching 200 and log the data for offline processing at a rate in the range of 5 to 7 kHz. In order to cope with the increased pileup the experiments are planning upgrades of major detector components [4, 5, 6, 7].

A combination of the increased pileup and the logged data rate will increase computing needs for the offline reconstruction. While the reconstructed output size on disk increases roughly proportionally with pileup, the reconstruction time grows as a higher power of pileup. The CPU needs can then become the dominant resource fraction in the total spending on computing. As an example, estimates done by CMS [5] suggest about a factor of ten shortage of CPU power to reconstruct all events with the current software under the assumption of the flat spending profile and expected future technology cost gains. In this scenario the estimated computing needs to run track reconstruction over a decade or so of HL-LHC operations can exceed the costs of the upgrade tracker detector construction. For this scenario to still be viable under the flat spending profile on computing, significant changes in algorithms used for tracking will need to happen, possibly at a cost to physics due to reduced phase space of reconstructed tracks. While the detector design is not yet final, further reduction in total cost may be achieved with detector design changes that aid reconstruction algorithms.

We consider two ingredients for the possible future charged particle tracking evolution towards HL-LHC: a tracking algorithm faster than the current conventional Kalman filter (KF) based tracking and the tracker layout design with pairs of sensitive layers placed closer than the current conventional tracker design of ATLAS or CMS with evenly-spaced layers. The analysis of the impact of the tracker layout is done in a simplified detector layout simulation corresponding to HL-LHC conditions. To quantify the impact of the tracker layout changes, we consider change in the purely combinatorial component in track building steps adjusted to keep track efficiency constant. Reduction of the combinatorial component corresponds to savings in the computing needs to run the track reconstruction.

2. Conventional Tracking and Alternatives

Three major steps can be identified for track reconstruction: track seeding, track building or pattern recognition, and track fitting. Track seeding aims to identify a possible starting point for the pattern recognition based on a small number of tracker hits and possible additional constraints for the proton-proton interaction origin. Charged particle track reconstruction in the ATLAS and CMS software relies on Kalman filter based approach [2, 3, 8]. The KF technique is applied to both the pattern recognition and track fitting. This is the conventional tracking approach.

The KF-based pattern recognition is outlined in a sketch in Fig. 1 (left). The initial seed state is propagated to the neighboring tracker detector layer. For every hit consistent with the propagated state a full track state update with the hit measurement is considered. Ambiguities are treated as alternative track candidates, including keeping the original candidate without a new hit. Uncertainties associated with material effects are accumulated as well. The procedure continues for every candidate to the last tracker layer, with some candidates terminating earlier after some threshold on the number of layers without actual measurements. Every propagation and update includes manipulation of the full five-parameter track state and its covariance matrix. At the end, the candidates are ranked and the best one is selected. Application of the KF technique at this step allows for mathematically most optimal accounting of the measurement uncertainties as well as the material effects.

After the pattern recognition, hits associated with the selected track candidate are used in the final track fit. Typically this Kalman fit is done in the forward and backward direction with the final track state estimate coming from a combination of the two (smoothing). Apart from the possible further cleaning and selection, a large fraction of the candidates passing the final track fit become the reconstructed tracks used in analyses.

During pattern recognition the fraction of layers with ambiguities grows with increasing pileup and the task becomes combinatorially complex. The number of candidates from combinatorial sources can become much larger than the number of final tracks. The computational cost of running the full KF-based pattern recognition can become prohibitive. A faster alternative can be considered. Compared to the pattern recognition step, the final track fitting step has essentially no combinatorial growth with pileup. The KF-based fitting remains feasible with increasing pileup.

The following aspects are important in the choice of a faster alternative to the KF-based pattern recognition: the reduction of high precision math for practical reasons; and maximization of locality of computations to specific detector regions. The latter is important to maximize utility of an implementation in a parallel computing environment, which is the driving technology trend with

GPUs, MIC, or similar. The KF pattern recognition which proceeds sequentially hit by hit with possible branching on each layer is hard to parallelize effectively [9]. An algorithm that reconstructs several parts of the track independently would allow for high degree of parallelization.

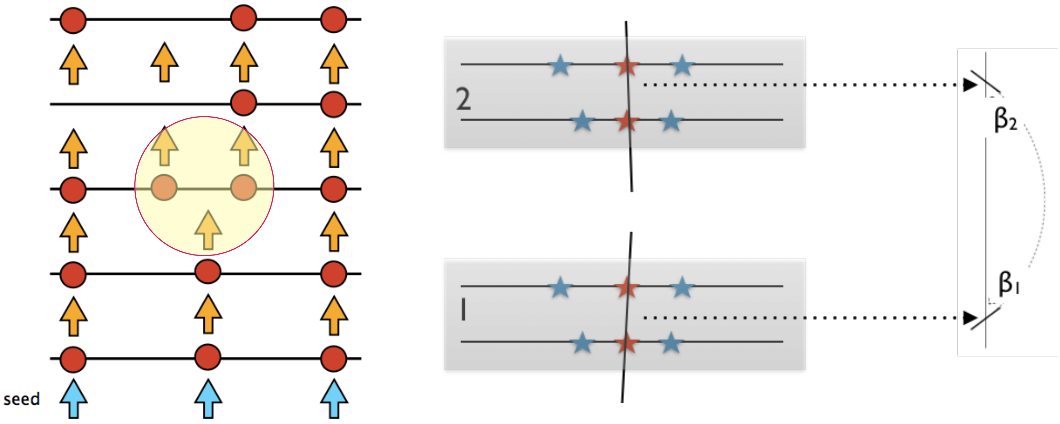

Figure 1: Left: illustration of track building with Kalman filter based tracking. Right: illustration of segment building and linking.

3. Segment linking as a fast tracking alternative

Segment linking algorithm outlined in Fig. 1 (right) is an example of a fast tracking algorithm, which can replace the seeding and the pattern recognition steps of the conventional KF-tracking. The implementation presented here is the most appropriate for track reconstruction in a uniform axial magnetic field in a tracker geometry with concentric measurement layers. The implementation idea comes from the segment linking algorithm used by the CDF collaboration [10]. For the study presented here we consider implementation corresponding to the central (barrel) region as can be applied to either the ATLAS or CMS trackers.

The main steps of the algorithm are the segment building, actual segment linking (combination of segment pairs to tracklets), and tracklet merging. Segment building provides line segments using measurements in two layers or more[1], which are separated well enough to provide high precision direction measurement. Segment linking step selects segments consistent with a helix of a track trajectory. In this steps segment pairs are selected if they are aligned along a straight line in rz direction (along the field lines) and if their direction difference is consistent with the four points of the segments aligned on the circle, as shown in Fig. 1 (right). The tracklets can be merged into track candidates by simply combining tracklets with shared segments.

The first two steps of the segment linking algorithm are implemented with rather simple math and logic. Segments are constructed with an assumption of a particle originating close to the proton-proton collision luminous region and having a momentum above a given threshold. This

[1] In the case of CMS tracker a pair of doublet layer hits can be used.

limits possible phase space of considered segments. The segments pairs are linked into tracklets with matching requirements taking into account segment position and direction resolution, material effects, and possible curvature of the trajectory. The selections are adjusted to maintain high efficiency above a p_T threshold, which was set at 1 GeV for this study. Estimates of the impact of the tracker layout on this algorithm were done using tracklets. The computational resource waste for the tracklet merging and further steps completing the full tracking are expected to be driven by the combinatorial complexity of the segment linking.

4. Layer Grouping Setup

Tracker hit distribution at high pileup on individual layers becomes rather uniform. Hits from particles with various momenta coming from separate proton-proton collisions dominate surrounding hit population for a given reconstructed charged particle track. Computing budget of an inclusive tracking algorithm is spent mostly on reconstruction of pileup tracks rather than on tracking in correlated contributions, as e.g. from high momentum hadronic jets. Segment building step, which is the least constrained part of the segment linking algorithm, has a large combinatorial component coming from surrounding pileup hits. It is clear, as illustrated in Fig. 2, that the smaller the distance between the measurement layers, the smaller is the chance to build a segment from unrelated hits. This suggests that combinatorial component in the segment linking should reduce if the layers are grouped. This works, as shown below, as long as the segment direction precision remains high.

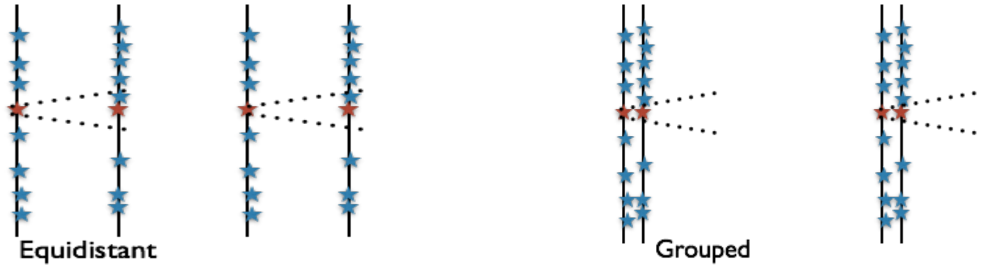

Figure 2: Illustration of segment building and linking in events with combinatorial background hits from pileup to contrast the equidistant layout (left) with the grouped layout (right). Segment building window, highlighted by dotted lines, is the same in both cases; it is defined by the acceptance needed for segment reconstruction. More combinatorial background hits are picked up in the equidistant layout and leads to more background segments.

We implement simplified tracker simulation with realistic hit placement, including resolution and material effects corresponding to the CMS upgrade tracker [5]. Simulation of top quark-antiquark pair production at a center of mass energy of 14 TeV with average pileup of 140 is used to study the impact of the tracker layout. Hit radii are adjusted from the reference geometry using known particle states associated to the hits followed by a short distance propagation to the alternative tracker layout. Layer outlines for two alternative layouts are shown in Fig. 3. To illustrate effects of the distance between layer pairs, we consider the layer separations of 16 cm and 2 cm. The larger value corresponds to the equidistant layout roughly matching the layer positions for the reference geometry. The smaller value of 2 cm is selected as a somewhat extreme case from the engineering design perspective, while it still corresponds to high precision of the reconstructed segment direction.

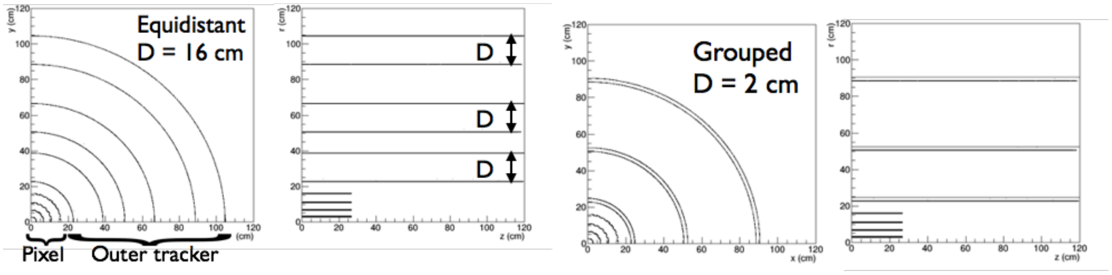

Figure 3: Illustration of mock geometry used in the layout impact study: the equidistant layout (left pair: *xy* and in *rz* view) and the grouped layout (right pair: *xy* and in *rz* view). Only the displayed barrel region is used in the study.

5. Results

The estimate of the computing resource wasted is done using the $\Delta\beta$ matching variable distribution defined in Fig. 1 (right) and computed at the final selection of the segment linking step. The distribution of $\Delta\beta$ on the sample of events with average pileup of 140 is shown for the two tracker layouts in Fig. 4. There are two components in the distribution of $\Delta\beta$: the narrow peak and the broad component under the peak. The narrow peak comes from track segments that belong to the same track. The broad component is from the combinatorial combinations of various kind: segments built from hits that do not belong to the same particle and segments from different particles combined. The final selection of the tracklets from pairs of compatible segments is the requirement on $\Delta\beta$ value to be close to the peak region as denoted with gray band.

The combinatorial component dominates the population of selected tracklets in the equidistant tracker layout case, compared to about a factor of nine smaller component for the grouped layout. The efficiency of genuine particle tracklets is essentially the same between the two cases. The peak width corresponding to genuine tracklets is roughly the same in both cases, which points to segment direction and matching quality preserved between the two cases. In a similar setup with a separation of 8 cm we find a reduction of the combinatorial component by a factor of three.

6. Summary

The results of this study show that a grouped tracker layer layout can significantly reduce the cost of computing needed to reconstruct charged particle tracks compared to the conventional choice of roughly equidistant layer separation. The tracker layout with the layer separation of 2 cm is shown to reduce the combinatorial component of tracking by a factor of nine compared to the equidistant outer tracker layout, similar to that currently planned by ATLAS and CMS. Considering comparable cost of the upgrade tracker construction and projected cost of running track reconstruction during the period of HL-LHC, there is an opportunity to reduce the total cost of the experiment.

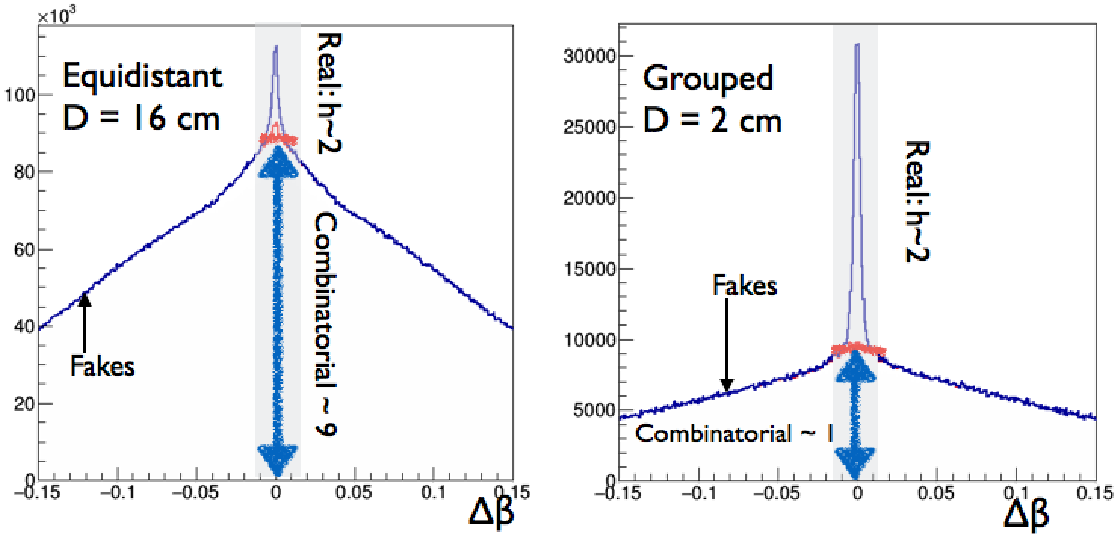

Figure 4: Distribution of $\Delta\beta$ at the last step of the segment linking algorithm for equidistant (left) and grouped (right) layout options. Entries in the peak above the combinatorial background indicate the population of the real segment pairs. The shaded region near the peak corresponds roughly to the selection of resulting linked segments. There is a factor of about nine reduction in the combinatorial background in the grouped layout compared to the equidistant case.

References

[1] HL-LHC project `http://hilumilhc.web.cern.ch/about/hl-lhc-project`.

[2] A. Salzburger [on behalf of the ATLAS Collaboration], *"Optimisation of the ATLAS Track Reconstruction Software for Run-2,"* J. Phys. Conf. Ser. **664**, no. 7, 072042 (2015), `http://dx.doi.org/10.1088/1742-6596/664/7/072042`.

[3] G. Cerati [on behalf of the CMS Collaboration], *"Vertexing and Tracking Algoritms at High Pile-Up,"* PoS Vertex **2014**, 037 (2015), `http://inspirehep.net/record/1393595`.

[4] ATLAS Collaboration, *"ATLAS Phase-II Upgrade Scoping Document,"* CERN-LHCC-2015-020, `https://cds.cern.ch/record/2055248`.

[5] D. Contardo *et al.* [on behalf of the CMS Collaboration], *"Technical Proposal for the Phase-II Upgrade of the CMS Detector,"* CERN-LHCC-2015-010, `https://cds.cern.ch/record/2020886`.

[6] D. Contardo *et al.* [on behalf of CMS Collaboration], *"CMS Phase II Upgrade Scope Document,"* CERN-LHCC-2015-019, `http://cds.cern.ch/record/2055167`.

[7] G. Auzinger [on behalf of the CMS Collaboration], *"Upgrade of the CMS Tracker for the High Luminosity LHC,"* CMS-CR-2016-268, `http://cds.cern.ch/record/2227193`.

[8] R. Fruhwirth, *"Application of Kalman filtering to track and vertex fitting,"* Nucl. Instrum. Meth. A **262**, 444 (1987), `http://dx.doi.org/10.1016/0168-9002(87)90887-4`.

[9] G. Cerati *et al.*, *"Kalman Filter Tracking on Parallel Architectures,"* arXiv:1605.05508 [physics.comp-ph], `https://arxiv.org/abs/1605.05508`.

[10] A. Mukherjee *"CTC and VTX Tracking,"* CDF Note 5490 (1995).