

Local reconstruction software for the CMS silicon strip tracker

R J Bainbridge¹, J Cole², J Fulcher¹, D Giordano³, C D Jones⁴
B Mangano⁵ and M Wingham¹

¹ Blackett Laboratory, Imperial College London, SW7 2BW, UK

² Rutherford Appleton Laboratory, Didcot, UK

³ INFN, Università di Bari, Italy

⁴ Cornell University, Ithaca, USA

⁵ University of California, San Diego, USA

E-mail: Robert.Bainbridge@cern.ch, Matthew.Wingham@imperial.ac.uk

Abstract. CMS has a two level trigger system, the first of which is hardware based, providing fast trigger decisions up to a rate 100 kHz. The second stage, known as the High-Level Trigger, is entirely software based and required to provide a trigger decision within 40 ms and a rejection factor of a thousand to achieve a write-to-disk rate of 100 Hz. One of the most CPU-intensive tasks within the High-Level Trigger is the reconstruction of tracking hits using raw data from the strip tracker. This study profiles the performance of the reconstruction algorithms. Even at low luminosities, an average processing time takes 5.5 s which already exceeds the HLT budget. A new schema, optimised for speed and performance, has been developed to reconstruct hits within *regions of interest* only. For the entire sub-detector, hit reconstruction times are reduced to 140 ms. Regional reconstruction is tested over $Z^0 \rightarrow e^+e^-$ events, by unpacking in η - ϕ windows of 0.16×0.16 around seeds identified in the calorimeter. In this case, only 2 ± 1 % of the silicon strip tracker raw data is reconstructed in 5 ± 3 ms whilst maintaining 99 % of the original dielectron trigger efficiency.

1. The CMS experiment

Compact Muon Solenoid (CMS) is one of the four main detectors at the Large Hadron Collider (LHC) based at CERN, Switzerland. It is a general purpose collider detector optimized to discover the Higgs Boson and signatures of physics beyond the Standard Model (SM). At the heart of CMS is a 13 m long, 5.9 m inner diameter superconducting solenoid, designed to produce nearly uniform magnetic fields of 4 T. Within the solenoid sits the LHC beam pipe, the tracking detector and the calorimeter. In order to deal with the high track multiplicities expected in the LHC environment and provide precision momentum measurements, CMS uses 9 to 12 layers of highly granular silicon microstrip detectors, resulting in a very large channel count. Three layers of silicon pixel detectors are placed closed to the interaction region for the precision spatial measurement of particle vertices. The calorimeter consists of both electromagnetic (Ecal) and hadronic (Hcal) components. Beyond the magnet sits an iron return yoke and a muon tracking system.

Sub-detector	Event data size (kB)	Channels
Pixel tracker	70	6.6×10^7
Strip tracker	500	10^7
Ecal	100	7.6×10^4

Table 1. Event data summary for the dominant sub-detectors at high luminosity running.

1.1. The event data

The CMS detector is unprecedented in terms of its size and complexity. At high luminosity running ($10^{34} \text{ cm}^{-2}\text{s}^{-1}$) event data sizes of ~ 2 MB are expected. Raw data are expected to contribute a significant fraction of this size. A breakdown on the contribution of the most dominant sub-detectors is shown in Table 1. Although the pixel detector has almost a factor 10 more channels than the strip tracker, larger occupancies in the latter result in a data rate almost 7 times larger [1].

1.2. The strip tracker readout system

The readout system is based on the front-end analogue APV25 readout chip [2], analogue optical links and an off-detector Front-End Driver (FED) processing board [3]. The APV25 chip samples, amplifies, buffers and processes signals from 128 channels of a silicon strip sensor at the LHC collision frequency of 40 MHz. On receipt of a Level-1 trigger, pulse height and bunch crossing information from pairs of APV25 chips are multiplexed onto a single line and the analogue data are converted to optical signals before being transmitted via optical fibres to the off-detector FED boards. The FEDs digitize, process and format the pulse height data from up to 96 pairs of APV25 chips, before forwarding zero-suppressed data to a computing farm for further event filtering.

1.3. The trigger

An LHC bunch crossing frequency of 40 MHz and large event data sizes necessitate a fast and efficient triggering system. The CMS trigger comprises two decision levels: Level-1 (L1) reduces the event rate to 100 KHz using calorimeter and muon information; the High Level Trigger (HLT) combines information from the full detector and reduces further to the write-to-disk rate of 100 Hz [4].

The HLT processes all events accepted by L1 in a single computer farm. The selection of events is optimized by rejecting un-interesting events as quickly as possible. With this in mind, each trigger path consists of several software modules, where each module fulfills a well defined task such as reconstruction or a trigger decision.

1.4. Track reconstruction

The raw data output from each FED encodes the basic hit information (pulse heights and corresponding strip numbers) necessary for tracking along with information on the bunch crossing number, the status of the tracker hardware and the status of the data acquisition system. Prior to reconstruction, these data are *unpacked* and the zero-suppressed strip information, known as a *digis*, are extracted.

Track reconstruction proceeds in two phases: local and global. In the first phase, localized *hits* are constructed from the extracted digis within the geometry of CMS. In the second phase, tracks are constructed from groups of these hits. Local reconstruction itself is a two-stage process

involving *clusterization* (grouping together neighbouring, gain corrected digis) and incorporation of the geometrical position of the hit. Global reconstruction can be subdivided into four steps: track candidate generation, track fitting, ambiguity resolution and smoothing. A track candidate defines the initial trajectory parameters and their errors. It can be provided in two ways: externally to the tracker using inputs from other detectors, such as the Ecal, or internally via the pixel hits. The pixel detector is best suited for track candidate formulation due to its low occupancy, the precision of its hits and its proximity to the beam pipe.

One pattern recognition algorithm used for track fitting is a combinatorial Kalman filter [5]. The filter proceeds from the coarse estimate of tracking parameters provided by the seed, determines which layers are compatible and extrapolates the trajectory to these layers according to the equations of motion of a charged particle in a magnetic field through material. As the trajectory encounters each hit, it is updated according to the Kalman filter formalism. If a stopping condition is satisfied, the algorithm is interrupted. It is especially useful for the HLT where, for example, the required accuracy is often reached after 5 or 6 hits.

Once the tracks are formed, ambiguities and instances of double counting are resolved. Since the full tracking information is only available at this point, any bias introduced with the imposed constraints is removed by a final fit. More details on the full tracking procedure can be found in [1].

1.5. The HLT time budget

During the first year of operating at low luminosity ($2 \times 10^{33} \text{ cm}^{-2}\text{s}^{-1}$), L1 is expected to provide an input trigger rate of 50 kHz to the HLT farm. The HLT farm will comprise 2000 CPUs, which results in an average time quota of 40 ms per event for each HLT processing node [6].

2. Local reconstruction software architecture

The HLT is implemented within the CMS offline reconstruction framework, CMSSW, which is designed around the Event Data Model (EDM) [7]. The basic premise of the EDM is that all user-defined types are contained within a single object called the *event*. All event products relate to a single triggered readout of the CMS detector. They are processed by passing the event object through a series of user-defined *plugin modules* within a CMSSW *path*. User access to auxiliary conditions data (e.g. cabling information) is given via an independent object called the *event setup*.

The CMSSW software design for the local phase of track reconstruction is shown in Figure 1. It describes the unpacking, clusterization and hit production modules interacting directly with the event and event setup. In the event data model, no interaction between software modules is allowed. Hence, there is no direct communication between the local reconstruction and track reconstruction software modules. Additionally, no information from the muon and calorimetry systems on the event topology is communicated to the tracking code.

2.1. Software performance

All studies in this document were performed in version 1.6.0 of CMSSW. Times quoted are wall-clock times measured on an a dual core AMD OpteronTM processor 248, with 2.14 GHz CPU and a 1 MB cache.

The software performance can be measured in two ways: through its timing cost to the HLT and through direct validation of the product digis against the input raw data. For the latter, an input source of fake raw data with configurable hit distributions has been developed. Though allowing the raw-to-digis module to be tested exhaustively, the clusterizer and hit producer cannot be validated in such a rigorous fashion. In general, the clusterizer algorithms reject digis below threshold and so the data are not directly comparable.

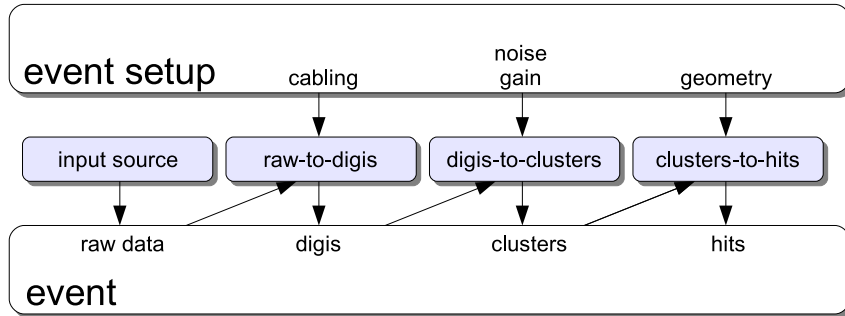


Figure 1. The software design for local reconstruction in the SST. Boxes represent CMSSW framework objects. Arrows represent the flow of data. The shaded boxes are plugin framework modules that perform a single event reconstruction task.

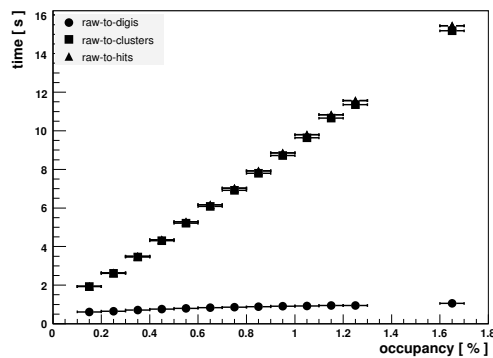


Figure 2. The variation in hit reconstruction time with strip occupancy for low luminosity minimum bias events. The mean cluster width for minimum bias events is 3.8 ± 0.3 strips.

The integrated reconstruction time up to the level of digis, clusters and hits is shown as a function of occupancy in Figure 2 for low luminosity minimum bias events. The time cost for the unpacking stage (identified in the plot as *raw-to-digis*) increases with strip occupancy, albeit with a relatively shallow gradient, due a single loop over all occupied digis stored within the raw data.

The time to clusterize the event data (*raw-to-clusters*) is strongly dependent on strip occupancy. This is to be expected since the algorithm performs two full loops over the zero-suppressed data. One to search for *seeds* and the second to find neighbouring strips above threshold. Multiple use of conditional statements also serve to make the clusterizer module the dominant time cost.

The hit production algorithm (*raw-to-hits*) is clearly the least time consuming. This is for several reasons: its task is simple with little memory overhead or CPU requirement; it deals with a smaller data-set since below threshold digis may be rejected by the clusterizer and it loops over clusters rather than digis reducing the number of loop iterations further still.

Since the expected mean tracker occupancy at low luminosity running is 0.56 % the corresponding mean local reconstruction time is 5.5 s. Although as few as 10 % of L1 accepted events are expected to require track reconstruction on the HLT [4], this still exceeds the total HLT budget (Section 1.5). To address these performance issues, the primary algorithmic inefficiencies and bottlenecks in the timing have been identified below:

- The majority of the unpacked raw data is not used within the tracking algorithms. A more sophisticated, *regional* approach to unpacking could be implemented to reduce this overhead, though not eliminate entirely since the smallest unpacking granularity is a single FED.¹
- At the heart of the clusterizer's inefficiency is its use of an initial loop over strips to find cluster seeds. A more streamlined algorithm is required to clusterize within a single loop.
- The templated container class *DetSetVector* $\langle T \rangle$ used by most CMSSW software modules within the reconstruction chain is effective, though highly inefficient when dealing with such large amounts of data. Every insertion into this collection forces a sort and copy. Since the number of insertions is large for the SST (for a low luminosity event, approximately 6×10^4 digis are generated and 1.5×10^4 clusters), a new container class is required.
- The entire software scheme enforces a copy of a large fraction of the raw data at both the unpacking and clusterization stages. Since C++ copy times are linear with the data size, this becomes a highly significant effect for a 10^7 channel detector. Furthermore, each stage in the low level reconstruction chain introduces a loop over channels. Integrating the unpacking and clusterizing algorithms within a single loop would serve to halve these effects.

3. Local reconstruction on demand

To address the issues itemized in Section 2.1 and hence streamline the local reconstruction chain, a new software architecture was devised. The SST modules are grouped into geometrical *regions*, defined by a grid on the η - ϕ plane (where η is the pseudorapidity and ϕ the azimuthal angle) with configurable dimensions. The new design allows raw data from *regions-of-interest* only to be considered. More specifically, any FED raw data packet with at least one channel connected to a region-of-interest is fully unpacked. A schematic of the regional concept is shown in Figure 3. These regions-of-interest can be defined by a dedicated CMSSW module, based on objects identified in external sub-detectors. Hit reconstruction is triggered by a request from any downstream module.

A new cluster container has also been developed. Storing clusters by region (numbered sequentially from zero) means read and write operations can be performed by direct access to the relevant memory address rather than through a search, which is inherently slow. A fixed container size also means the sort and copy involved with *DetSetVector* $\langle T \rangle$ insertions is no longer necessary.

A framework feature is that once data are stored within the event, they can no longer be changed. This becomes an obstacle when multiple tracking modules exist within a single path, each with an independent set of tracking regions-of-interest. For example, within the HLT where multiple reconstruction chains can be executed for every event. To overcome this, an empty cluster container is added (by a *clusterizer facility*) with access to the raw data and relevant calibrations. When the raw-to-hits module requests clusters, the container self-fills by unpacking and clusterizing all regions-of-interest within a single loop.² A schematic of the new software design is shown in Figure 4.

¹ In principle, unpacking a single channel is also possible, however since the raw data are zero-suppressed the location of each channel within the packet will always require one full scan.

² The inclusion of hit geometry has not been integrated within this loop. Instead, it must be performed independently. This avoids adding dependencies on the CMS geometry which greatly simplifies the software management.

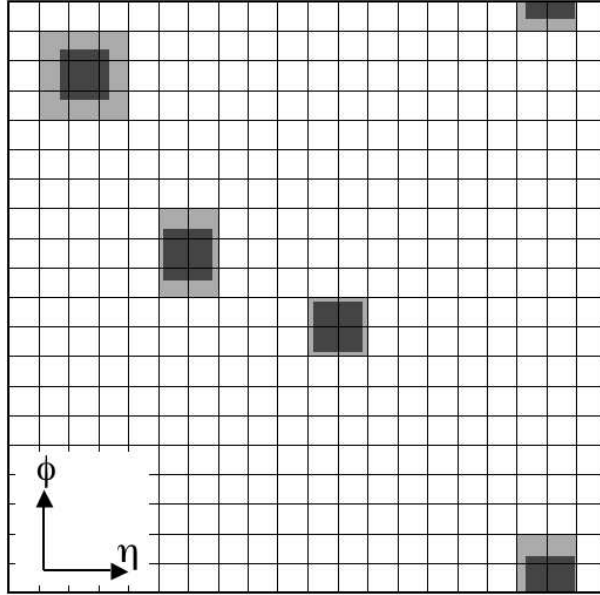


Figure 3. The *regional* view of the SST in the η - ϕ plane. Each square corresponds to a section of the detector containing up to 12 layers, or *regions*. Each dark grey shape represents a window of interest *seeded* by a physics object identified in an external subdetector. The window size is configurable and will vary with object flavour. Light grey squares are the corresponding *regions-of-interest*. The number of regions defined by a given window may vary with the position of the seed. Windows may also *wrap around* the ϕ dimension.

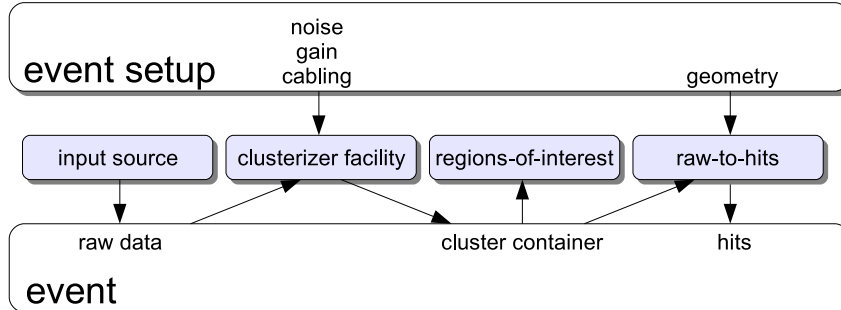


Figure 4. The new local reconstruction software design for the SST. Boxes represent CMSSW framework objects. Arrows represent the flow of data. The shaded boxes are plugin framework modules that perform a single event reconstruction task.

3.1. Software performance

The total number of tracker regions has a significant effect on the overall software performance. This is mainly due the overhead introduced by initialising the cluster container at the start of every event. Figure 5 shows a sharp increase in reconstruction times above a 20×20 configuration. Since each of these divisions is further divided into layers, this corresponds to ~ 4000 regions in total.

Figure 6 shows the distribution in hit reconstruction times for low luminosity minimum bias events when unpacking all regions. Over 1000 events, all times were measured at under 400 ms with a mean value of 140 ms. This is almost a factor 40 improvement on the previous

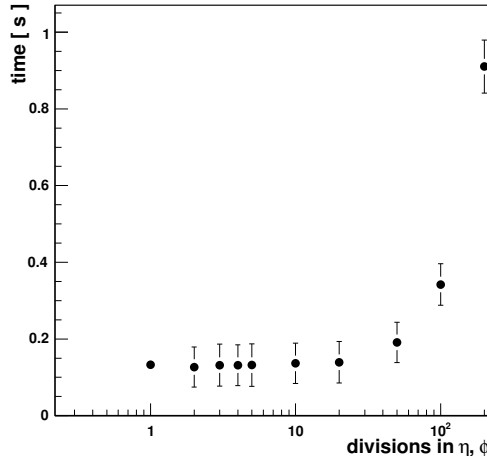


Figure 5. The variation in hit reconstruction time with regional granularity for minimum bias events at low luminosity. Here, the new regional scheme is used but in *global* mode. Reconstruction times increase rapidly above 20 divisions in η and ϕ , corresponding to a regional granularity $\Delta\eta \times \Delta\phi$ of 0.250×0.314 . For reference, a calorimeter tower in the barrel has granularity 0.0870×0.0873 .

reconstruction scheme. Since only 10 % of L1 accepted events are expected to require track reconstruction on the HLT, the average contribution per event is therefore ~ 14 ms. This is ~ 30 % of the full budget.

The variation of time with strip occupancy and mean cluster width is shown in Figure 7. The figure on the left shows a linear increase in reconstruction times with occupancy analogous to the trend of Figure 2. This also validates the fake cluster source described in Section 2.1 which is used in the figure on the right. Here the occupancy is fixed, hence the cluster width and total number of clusters are inversely proportional. Therefore, increasing cluster width results in a decreasing reconstruction time.

Table 2 gives a breakdown of the dominant time costs measured over a characteristic HLT path. Due to the nature of the new SST reconstruction schema, the hit reconstruction time is absorbed by the L2.5 track reconstruction module. Of the 60 ms measured here, ~ 50 % goes toward reconstructing hits. In comparison, the full Ecal hit production chain costs 22.6 ms with only $1/5^{th}$ the data rate (Table 1). The averaged times demonstrate the true timing cost of each module after considering the fraction of L1 events they are required for. They clearly show the strip tracker is no longer the primary reconstruction bottleneck. Also, for minimum bias events without pile-up the HLT quota of 40 ms has been achieved.

Figure 8 demonstrates the benefits seen with the regional approach. The fit is for a linear increase in time with regions-of-interest over evenly distributed fake clusters. An overhead of less than 5 ms exists in the case of no reconstruction being performed.

Over $Z^0 \rightarrow e^+e^-$ events, the calorimeter seeds used within isolated electron reconstruction were also used to define the tracker regions-of-interest. The dielectron trigger efficiency has been recorded for various $\Delta\eta \times \Delta\phi$ windows around these center-points. Figure 8 indicates that 99 % of the efficiency achieved with full hit reconstruction occurs with a 0.16×0.16 window. This corresponds to an array of dimension 2×2 , 3×3 , 2×3 or 3×2 , depending on the η, ϕ position of the seed.

A direct comparison of regional and global reconstruction is shown in Table 3 over $Z^0 \rightarrow e^+e^-$

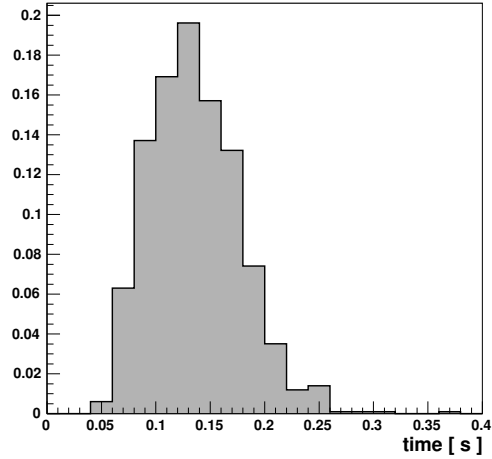


Figure 6. The spread in hit reconstruction time for minimum bias events at low luminosity. Here, the new regional software scheme is used, but in *global* mode.

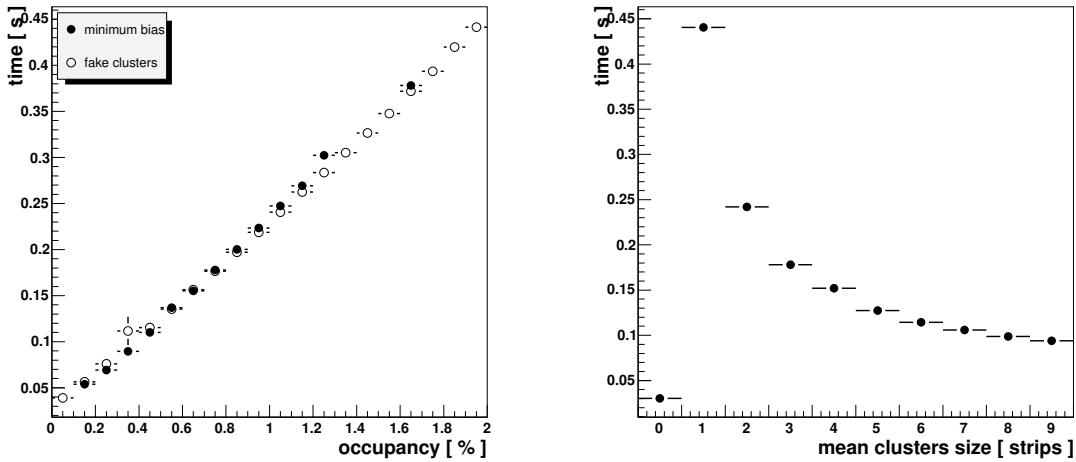


Figure 7. (left) The variation in hit reconstruction time with strip occupancy. The minimum bias events were generated at low luminosity giving a mean occupancy of 0.56 ± 0.18 % and a mean cluster width of 3.8 ± 0.3 strips. The fake clusters are all 4 strips in width and evenly distributed throughout the detector. (right) The variation in hit reconstruction time with cluster width for fake clusters. Here, occupancy is fixed at 0.56 %. For both figures the new regional software scheme is used, but in *global* mode.

HLT process	Running time (ms)	Averaged time (ms)	Fraction of all L1 events processed
Ecal unpacking	13.6	13.6	0.11
Ecal hit production	9	9	0.11
Hcal unpacking	1	1	0.11
Hcal hit production	3	3	0.11
Calorimeter tower maker	4.5	4.5	0.11
Jet reconstruction	3	3	0.11
Pixel unpacking	2	0.1	0.006
Pixel clustering	6	0.3	0.006
Pixel hit production	2	0.1	0.006
Pixel tracks	10	0.5	0.006
L2.5 regional seeding	11	0.5	0.006
L2.5 track reconstruction	60	2.7	0.006
L3 regional seeding	26	0.2	0.0008
L3 track reconstruction	280	2.0	0.0008
Total time	421	41.5	

Table 2. The time cost of the dominant processes within the *single tau plus missing E_T* HLT path over minimum bias events (no pile-up). The first number (Running time) represents the average wall-clock time for each process over a given event, while the second number (Averaged time) refers to the the average time per minimum bias event passing L1.[4]

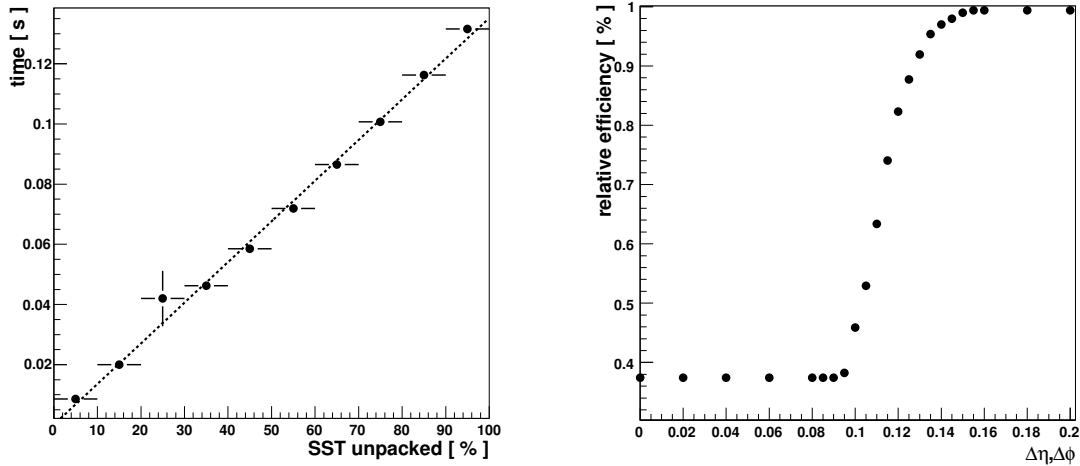


Figure 8. (left) The variation in hit reconstruction time as a function of the fraction of reconstructed regions. Fake, 4-strip clusters are evenly distributed throughout the tracker at low luminosity (0.56 % strip occupancy). (right) The variation in HLT dielectron trigger efficiency with reconstruction window size, $\Delta\eta \times \Delta\phi$. Here, efficiency is relative to that measured for global reconstruction over $Z^0 \rightarrow e^+e^-$ events.

	Regions-of-interest (%)	Occupancy (%)	Time (ms)
Global	100	0.24 ± 0.06	70 ± 10
Regional	2 ± 1	0.4 ± 0.1	5 ± 3

Table 3. Performance summary of the local reconstruction software in both *regional* and *global* modes over $Z^0 \rightarrow e^+e^-$ events without pile-up.

events. With a $0.16 \times 0.16 \Delta\eta \times \Delta\phi$ window only 2 ± 1 % of the tracker is reconstructed in 5 ± 3 ms. This is a factor 13 faster than the global approach.

4. Integration with the tracking algorithms

Work is ongoing to introduce these algorithms within the tracking code itself. In this way, detector regions of interest can be defined layer-by-layer using the projected track trajectory and its error. Information external to the SST would no longer be required, simplifying the software management considerably. Also, since all regions intersecting the track’s path are reconstructed by definition, trigger efficiencies would remain unaffected. In this sense it is the best possible scenario.

5. Summary

Local reconstruction of the CMS silicon strip tracker data (to the level of hits) on the HLT computer farm is subject to stringent requirements in terms of the HLT time budget (40 ms/event) and the raw data volume to be processed (0.5 MB/event). To minimize their contribution to the HLT budget, the local reconstruction algorithms have been redesigned, reducing the reconstruction time for minimum bias events at low luminosity from 5.5 s to 140 ms. Since approximately 10% of level-1 accepted events are expected to require track reconstruction, the average contribution per event is ~ 14 ms. This is ~ 30 % of the full budget.

Additionally, the new schema is also capable of reconstructing only physics regions-of-interest, in order to reduce the time cost further. For an example trigger path of $Z^0 \rightarrow e^+e^-$ events, 99 % of the original dielectron trigger efficiency is maintained whilst unpacking only 2 ± 1 % of the tracker in 5 ± 3 ms. Similar performances are expected for all trigger paths.

References

- [1] CMS Collaboration, “The CMS Physics Technical Design Report, Volume 1”, *CERN/LHCC 98-006* 1998
- [2] M. French *et al*, “Design and Results from the APV25, a Deep Sub-micron CMOS Front-End Chip for the CMS Tracker”, *Nuclear Instruments and Methods A*, 2001 359-365
- [3] J. Coughlan, “The Front-End Driver Card for the CMS Silicon Strip Tracker Readout,”, *CERN/LHCC 2002-034* 2002
- [4] CMS Collaboration, “CMS High Level Trigger”, *CERN/LHCC 2007-021* 2007
- [5] R. Fruhwirth, “Application of Kalman filtering to track and vertex fitting.”, *Nuclear Instruments and Methods A*, 1987 444-450
- [6] CMS Collaboration, “The TriDAS Project Technical Design Report, Volume 2: Data Acquisition and the High-Level Trigger ”, *CERN/LHCC 02-26* 2002
- [7] “CMSSW Application Framework” <https://twiki.cern.ch/twiki/bin/view/CMS/WorkBookCMSSWFramework>