PIXEL RECONSTRUCTION IN THE CMS HIGH LEVEL TRIGGER

S. Cucciarelli, CERN, Geneva, Switzerland, M. Konecki, University of Basle, Switzerland, D. Kotlinski, PSI, Switzerland, T. Todorov, IN2P3, Strasbourg, France

Abstract

The Pixel Detector is the innermost detector of the tracking system of the Compact Muon Solenoid (CMS) experiment. It provides the most precise measurements not only supporting the full track reconstruction, but also allowing the standalone reconstruction especially useful for the online event selection at High-Level Triggers (HLT). The HLT algorithms using the Pixel Detector are presented, including pixel track reconstruction, primary vertex finding, Tau identification, isolation and track seeding.

INTRODUCTION

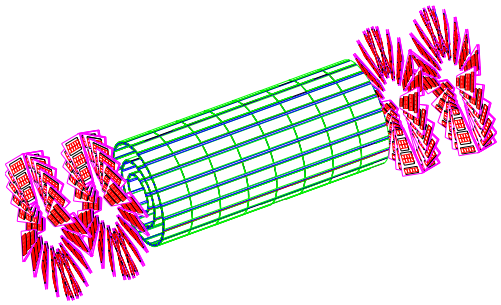

Figure 1: The CMS Pixel Detector Layout.

Fast and efficient tracking and algorithms for primary-vertex finding are necessary for the High Level Trigger. The Pixel detector is the most suitable to provide track candidates, because it has a very good spatial resolution of reconstructed three dimensional hits. The typical spatial resolution is around 8 and 15 μm in the transverse and longitudinal coordinate respectively [1]. The Pixel Detector is close to the interaction point, thus it minimizes multiple scattering and it has a low occupancy.

The Pixel detector layout considered here consists of three barrel layers with two endcap disks on each side, as shown in Fig. 1. The three barrel layers are located at mean radii 4.4, 7.3 and 10.2 cm and are 53 cm long. The two disks are placed at 34.5 and 46.5 cm from the interaction point. To achieve a similarly good resolution of the vertex position in the transverse and the longitudinal planes, a design with a rectangular pixel shape of dimensions 150 $\times$ 100 μm^2 and thickness 300 μm is used. To enhance the spatial resolution by analog signal interpolation the effect of charge sharing induced by the large Lorentz drift in the 4T magnetic field is used. Hence the detectors are deliberately not tilted in the barrel layers but are tilted in the end disks resulting in a turbine like geometry. The whole Pixel system consists of about 1400 detector modules arranged into half-ladders of four identical modules each in the barrel, and blades with seven different modules each in the endcaps. A more detailed description of the Pixel layout can be found in Ref. [2].

PIXEL TRACK FINDING

The track finding based on pixel hits consists of two steps: defining a set of hit pairs compatible with a track and from these pairs making a prediction of the third hit. Two precise hits (pairs) are enough to define a track seed, but the ghost rate is high. Track candidates based on three pixel hits (triplets) allow to do primary vertex reconstruction and to define simple algorithms for online event selection, even if they are not fully efficient.

The search for track candidates based on two pixel hits is performed inside a region of interest with kinematic constraints (Tracking Region). A Tracking Region is defined by a direction, a vertex point from which tracks are expected to originate, a minimal transverse momentum (p_T), a maximal allowed closest distance from the beam in the transverse plane and a maximal allowed distance from the vertex along the beam line. In addition a range of tolerance in the pseudorapidity (η) and in the azimuthal angle (ϕ) are considered. The first hit search is in the outermost layer within the η and ϕ allowed ranges. The compatible hits in the outermost layer are used as constraint to search for second hits in one of the two innermost layers. The analytical prediction of the azimuth angle and its tolerance are used to find compatible hit pairs. The compatibility is successively checked in the longitudinal plane using the constraint from the defined tracking region.

A pair of hits together with the kinematic constraints from the tracking region are used to predict the third compatible hit to be attached to the track candidate. The two layer combinations that are chosen always guarantee the possibility of having three pixel hits in a track. The triplet finding starts in the transverse plane selecting hits within a $\Delta\phi$ window around the predicted ϕ value. The selected hits are further constrained in the r-z plane within a Δz and Δr windows around the line passing through the hit pair. Finally the ϕ constraint is re-checked using the prediction from the circle approximated with a parabola [3]. The performance of the triplet finding algorithm is shown in Figures 2 and 3 in terms of efficiency to find single muon tracks. In Fig. 2 the efficiency is shown as a function of the transverse momentum of the simulated tracks,

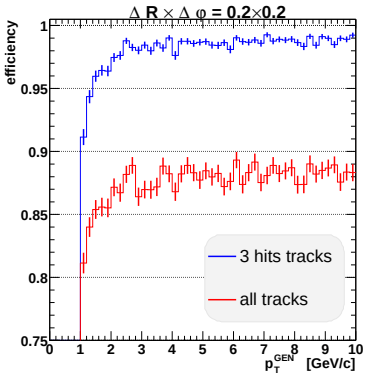

Figure 2: Triplet finding efficiency as a function of p_T with respect to the simulated single muon tracks that have at least three pixel hits (upper line) and with respect to all the simulated tracks in the detector (lower line).

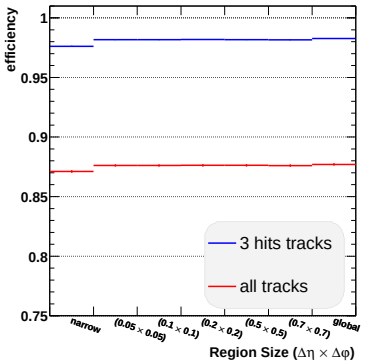

Figure 3: Triplet finding efficiency for different tracking region sizes with respect to the simulated single muon tracks that have at least three pixel hits (upper line) and with respect to all the simulated tracks in the detector (lower line).

and in Fig. 3 as a function of the region's of interest size in the η-ϕ plane. The upper lines refer only to the single muon tracks that have three hits in the Pixel Detector, thus they reflect the algorithmic efficiency of the triplet finding method. The lower lines refer to all the simulated single muon tracks, thus they also include effects due to the detector inefficiencies. The algorithmic efficiency is around 98% for tracks with p_T greater than 2 GeV/c and independently of the size of the tracking region. If detector effects are also considered, the efficiency with respect to all the simulated tracks is around 88% for transverse track momentum above 2 GeV/c.

The CPU time is shown in Fig. 4 for different sizes of the region of interest. The time refers to $\mathrm{q\bar{q}}$ events in the high luminosity environment and the regions of interest are defined around the jet direction. The time is measured on a 2.8 GHz Pentium IV. About 50% of the time is spent for reconstructing, accessing and sorting hits. The gain on processing time going from a regional search to the full acceptance of the Pixel Detector (global region) is more than a factor 5.

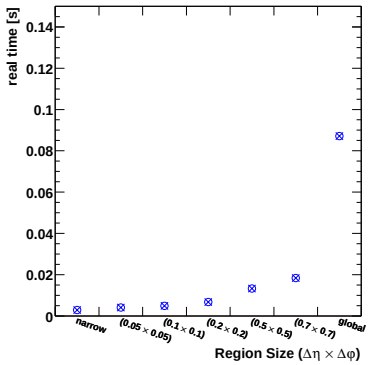

Figure 4: CPU time of the triplet finding for different sizes of the tracking region with $\mathrm{b\bar{b}}$ event at high luminosity, time has been measured on a 2.8 GHz Pentium IV.

PRIMARY VERTEX FINDING

The primary-vertex finding based on the pixel hits provides a simple and efficient method for the primary-vertex (PV) position measurement. This measurement is subsequently used for track seeding and in many High-Level Trigger (HLT) analyses. It must therefore be fast and precise enough. For this reason primary-vertex finding is reduced here to a one-dimensional search along the z axis. The two primary-vertex finding algorithms which are described in the following, refer to hit triplets found in the full Pixel detector acceptance. It is also possible to restrict the triplet finding to selected regions of the Pixel detector, in order to make the vertex finding faster.

The search for primary vertex along the z axis is based on the longitudinal impact point z_{IP} evaluation from tracks made of three hits (a more detailed description of the track parameter evaluation can be found in [4]). Figure 5 shows

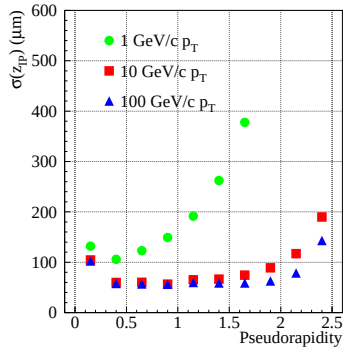

Figure 5: Resolution of the longitudinal impact point from the helix parametrization, as a function of the pseudorapidity and for for p_T values 1, 10 and 100 GeV/c.

the $z_{\rm IP}$ resolution as a function of the pseudorapidity for single muon tracks with different transverse momenta p_T. For high p_T tracks in the barrel region the longitudinal impact point is evaluated with a resolution of $\sim 60\,\mu$m. Only pixel tracks reconstructed with p_T in excess of 1 GeV/c and a transverse impact point smaller than 1 mm are used for the vertex finding.

The output of the PV finding algorithm is a list of primary vertex candidates, denoted *PV Clusters*. Among these candidates, the *closest* primary vertex is defined as that closest in z to the simulated signal PV and the *tagged* primary vertex as that chosen by the reconstruction algorithm. For a given event, the primary vertex (tagged or closest) is *found* if it is reconstructed inside a window of 500 μm around the true PV position. The PV-finding efficiency is the fraction of events with a *found* (tagged or closest) primary vertex. The closest PV-finding efficiency evaluates the ability of the algorithm to find a PV candidate. The tagged PV-finding efficiency evaluates the ability of the algorithm to identify the signal PV of the event.

Histogramming Method

The histogramming method progressively merges tracks close to each other in $z_{\rm IP}$, to form primary-vertex candidates. The track longitudinal impact points, $z_{\rm IP}$, are first histogrammed in 5000 bins in a ± 15 cm window around the nominal interaction point. Only the non-empty bins are kept, and their position is computed as the track $z_{\rm IP}$ weighted average. These non-empty bins are then scanned along z. A PV cluster is defined as a continuous set of consecutive bins separated by less than a certain threshold Δz. The z position of the PV cluster, $z_{\rm PV}$, is determined by averaging the $z_{\rm IP}$ of all tracks associated to this cluster. A cleaning procedure is applied to each PV cluster, rejecting the tracks distant from the PV-cluster position by more than $z_{\rm offset}$ standard deviations, i.e., such that $|z_{\rm IP} - z_{\rm PV}| < z_{\rm offset} \cdot \sigma_{z_{\rm IP}}$, where $\sigma_{z_{\rm IP}}$ is parametrized as a function of the η and p_T of the track. The z position of the PV clusters is recomputed as a weighted $z_{\rm IP}$ average of the remaining tracks.

For each PV cluster, the quantity $S = \sum p_T'^2$ is computed, where the sum runs over all the tracks associated to the cluster and

$$p_T' = \begin{cases} 0 & \text{if } p_T < p_T^{\rm min}, \\ p_T & \text{if } p_T^{\rm min} < p_T < p_T^{\rm max}, \\ p_T^{\rm max} & \text{if } p_T > p_T^{\rm max}; \end{cases} \qquad (1)$$

where $p_T^{\rm min}$ is typically around 2 GeV/c and $p_T^{\rm max}$ is around 10 GeV/c. The PV cluster with the largest S value is called the *tagged* PV, by definition. In the S evaluation, the tracks with a very small p_T (below $p_T^{\rm min}$) likely originating from pileup events, are not considered. A threshold is set at high momentum ($p_T^{\rm max}$) not to overweight vertices with very few high-momentum tracks, determined with a poor resolution.

Especially at high luminosity, the performance of the algorithm depends on the Δz parameter and only mildly on the $z_{\rm offset}$ parameter. Figure 6 shows the PV-finding efficiency for different values of Δz, for both the closest and tagged primary vertices. The best performance of the algorithm is reached for small values of the merging parameter due to the pollution from pileup events at high luminosity. Indeed, for large Δz values, many bins are merged together and the PV cluster is associated to many tracks either coming from other vertices or which are incorrectly reconstructed. The averaged z_{PV} value is therefore far from the true position, and the PV is subsequently not *found*.

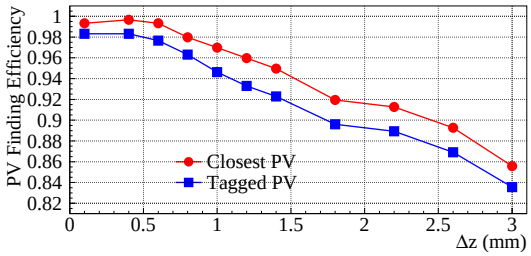

Figure 6: The PV-finding efficiency of the histogramming method. Efficiencies for the closest (circle) and tagged (square) primary vertex of the event are shown as a function of the merging parameter Δz, for high luminosity $\rm q\bar{q}$ events with E_t =100 GeV.

Divisive Method

The same set of tracks as for the histogramming method is used in the divisive method. In this method, the tracks are ordered according to increasing $z_{\rm IP}$. The ordered list is scanned to form a PV cluster until a pair of consecutive tracks separated by more than a certain threshold $z_{\rm sep}$ is found, at which point another PV cluster is built.

For each initial PV Cluster, an iterative procedure is applied to discard tracks not compatible with it. Tracks are discarded according to the $z_{\rm offset}$ parameter as explained in the previous section, and the cluster position is recomputed. The procedure iterates until each remaining track is declared compatible with its associated PV cluster position. Discarded tracks are recovered to form a new PV cluster and the above procedure is applied. New PV clusters are built iteratively, until the number of remaining tracks is smaller than $N_{\rm Tk}^{\rm min}$. (Here, the choice $N_{\rm Tk}^{\rm min} = 2$ is made.) The tagged PV cluster is defined as in Section , i.e., according to the largest value of S. The performance of the divisive PV-finding in a high luminosity environment is sensitive to the value of the $z_{\rm sep}$ parameter. For large values of $z_{\rm sep}$ (above 1 mm) the closest and tagged PV-finding efficiencies decrease. In this case, it may happen that the initial PV cluster contains tracks coming from two vertices, therefore the z-PV position is between the two and most of the tracks are discarded at the next iteration. Values of the PV-finding efficiency above 95% are reached for values of the separation parameter around 500 μm or below.

Performance Comparison

The two PV-finding algorithms using tracks made of three pixel hits reconstruct the z position of the primary vertex with an efficiency close to 100 % at high and low ($2{\cdot}10^{33}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$) luminosities. The PV z position is reconstructed with a resolution of about 50 μm and 40 μm at the high and low luminosity respectively for both the histogramming and the divisive methods.

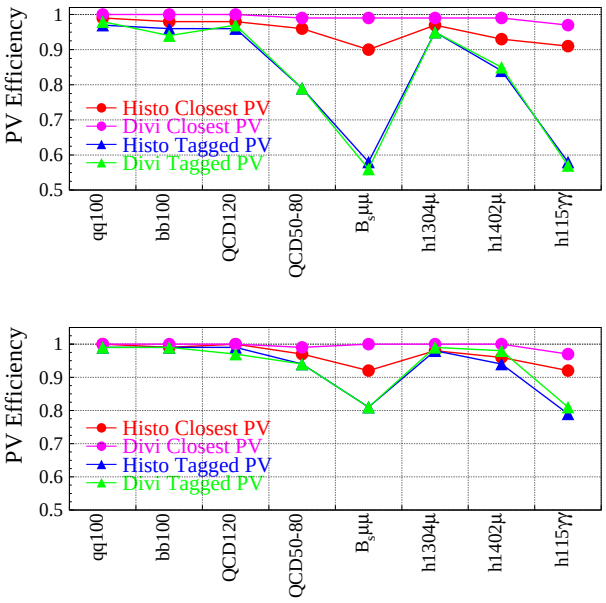

Figure 7: Tagged (circle) and Closest (triangle) PV-finding efficiencies of the histogramming and divisive methods, for different samples of simulated events at high (top) and low (bottom) luminosities.

The performance comparison between the two algorithms is presented in Fig. 7, where the closest and tagged PV-finding efficiencies of both the histogramming and the divisive methods are shown for different simulated event samples at high and low luminosities. The divisive method gives better closest PV-finding efficiencies, while in terms of tagged PV-finding the two algorithms are comparable. Closest PV-finding efficiencies are very close to 100 % for the different samples considered here. Tagged PV-finding efficiencies are significantly below 100 % for events like $\mathrm{h} \to \gamma\gamma$ and $\mathrm{B_s} \to \mu\mu$, where the small average number of charged particle tracks does not allow the signal PV to be always distinguished from pileup primary vertices. Other methods to find the most likely signal PV specific to these physics channels are under investigation.

The average time per event needed for the track parameter evaluation is about 7 ms per event. The average time for the primary-vertex finding is 0.7 ms per event, for both the histogramming and divisive methods. The time was measured on a 2.8 GHz PentiumIV and for $\mathrm{q\bar{q}}$ events with $E_T^{\mathrm{Jet}} = 100$ GeV at high luminosity. The time quoted does not include the contributions from the hit reconstruction and the triplet finding.

AN APPLICATION: HLT τ

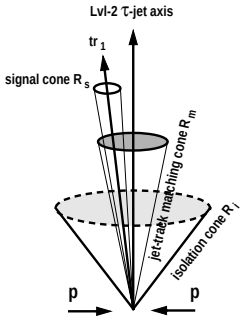

Figure 8: Sketch of the principle of τ-jet isolation using charged particles tracks reconstructed in the Pixel Detector.

The principle of τ-jet identification [5] is shown in Fig. 8. The direction of the τ-jet is defined by the axis of the calorimetric jet. Pixel track candidates in a matching cone R_m around the jet direction and with a p_T above a threshold (typically 1 GeV/c) are considered in the search for signal tracks. A narrow signal cone R_s is defined around the leading signal track and any other tracks inside the R_s cone is supposed to come from the τ-decay. Tracks with p_T above a threshold are searched for inside a larger cone around the leading track, called isolation cone R_i. If no tracks are found in the isolation cone, except from the ones which are already in the R_s cone, the isolation criterion is fulfilled. The performance of the τ-trigger based on pixel

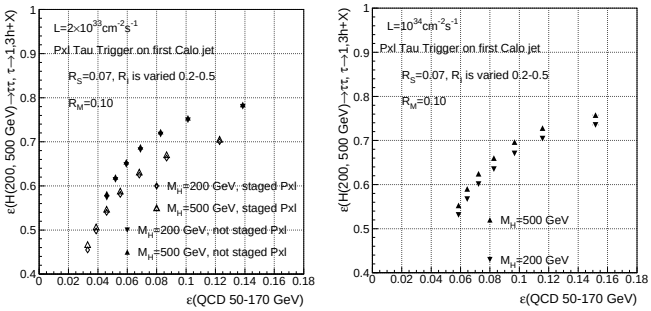

Figure 9: Efficiency of the τ-trigger for the first calorimetric jet in $\mathrm{A/H} \to 2\tau \to 2\tau$-jet, for two Higgs masses, as a function of the efficiency for QCD di-jet background events at at a luminosity of $2\mathrm{x}10^{33}$ cm6$-$2s^{-1} (left) and $10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ (right).

track candidates is shown in Fig. 9 in terms of signal efficiency for the channel $\mathrm{A/H} \to 2\tau \to 2\tau$-jet versus the efficiency for QCD di-jet background events for the high and low luminosities. The size of the isolation cone varies

from 0.2 to 0.5 and two values of the Higgs mass have been considered. At low luminosity also the staged Pixel layout consisting of 2 barrel layers and 2 endcap disks, has been considered.

REFERENCES

[1] S. Cucciarelli, D. Kotlinski, T. Todorov, CMS Note 2002/049.

[2] The CMS Collaboration, “CMS Tracker Technical Design Report”, CERN/LHCC 1998/06.

[3] M. Hansroul, H. Jeremie, D. Savard, NIM A270 (1998) 490.

[4] S. Cucciarelli, M. Konecki, D. Kotlinski, T. Todorov, CMS Note 2003/026.

[5] D. Kotlinski, A. Nikitenko, R. Kinnunen, CMS Note 2001/017.