

A new technique for the reconstruction, validation, and simulation of hits in the CMS Pixel Detector

M. Swartz*, **D. Fehling**, **G. Giurgiu**, **P. Maksimović**

Department of Physics and Astronomy, Johns Hopkins University, Baltimore, Md., USA
E-mail: morris@jhu.edu

V. Chiochia

Physik Institut der Universität Zürich-Irchel, 8057 Zürich, Switzerland

New techniques for the reconstruction/validation and the simulation of hits in the pixel detectors of the Compact Muon Solenoid (CMS) Experiment are described. The techniques are based upon the use of pre-computed projected cluster shapes or “templates”. A detailed simulation called Pixelav that has successfully described the profiles of clusters measured in beam tests of radiation-damaged sensors is used to generate the templates. Although the reconstruction technique was originally developed to optimally estimate the coordinates of hits after the detector became radiation damaged, it also has superior performance before irradiation. The technique requires a priori knowledge of the track angle which makes it suitable for the second in a two-pass reconstruction algorithm. However, the same modest angle sensitivity allows the algorithm to determine if the sizes and shapes of the cluster projections are consistent with the input angles. This information may be useful in suppressing spurious hits caused by secondary particles and in validating seeds used in track finding. The seed validation is currently under study but has the potential to significantly increase the speed of track finding in the CMS reconstruction software. Finally, a new procedure that uses the templates to re-weight clusters generated by the CMS offline simulation is described. The first tests of this technique are encouraging and when fully implemented, the technique will enable the fast simulation of pixel hits that have the characteristics of the much more CPU-intensive Pixelav hits. In particular, it may be the only practical technique available to simulate hits from a radiation damaged detector in the CMS offline software.

Vertex 2007: 16th International Workshop On Vertex Detectors
24-28 September 2007
Lake Placid NY, USA

*Speaker.

1. Introduction

Hybrid pixel detectors are now coming into widespread use at the CERN Large Hadron Collider (LHC). Although they function similarly to strip detectors, they also differ in some significant ways and these differences affect the hit reconstruction algorithms. One significant difference is that the pixel cells are not capacitively coupled. There is no coupling matrix to understand and therefore the charge sharing is caused entirely by charge collection and induction effects in the detector substrate. Another difference is that the two-dimensional clusters contain hit position and track angle information. The fine sampling of the cluster enables new possibilities in signal processing that can suppress the deleterious effects of large delta rays. Because the new generation of pixel detectors are “n+ in n” devices, they collect electrons and must cope with potentially large Lorentz drift effects. This fact implies that the usual silicon strip reconstruction technique based upon the integral of the “ η -distribution” [1] does not completely self-calibrate (there is an unknown integration constant) and requires additional calibration. The new generation of pixel detectors at the LHC will be exposed to much more radiation than previous generations of silicon tracking detectors. The resulting space charge and signal trapping effects will modify the responses of these detectors significantly during their operational lifetimes. A reconstruction technique that can account for changing detector response is clearly desirable. This talk describes a new hit reconstruction technique [2] that has been developed for the pixel detector [3] of the Compact Muon Solenoid (CMS) Experiment [4]. The technique also provides hit validation information and can be adapted to re-weight simulated clusters to account for radiation damage.

2. Pixel Clusters

The deposition of charge by a track having angles α and β with respect to the local x - and y -axes of a barrel module is shown in Fig. 1. The primary track deposits approximately 25,000 electron-hole pairs per 300 μm of track length more or less uniformly in the y -direction. For highly inclined tracks, about 12,500 pairs are deposited in each 150 μm wide pixel column. The n-in-n sensors collect electrons which have a large Lorentz angle ($\sim 23^\circ$ at 150 V bias [5]) in the 4 T magnetic field of CMS. The charge from the larger local z -side of the sensor typically drifts by more than a pixel x -width into the adjacent row of pixels producing clusters with the typical shape shown in Fig. 2. The track projection is shown as the dashed red line on the cluster. Note that the track center, shown as the cross, is contained in a pixel that does not have enough charge (the threshold is approximately 2.5k electrons) to trigger its readout. The primary ionization process produces large fluctuations in charge along the track. As should become clear from the following discussion, any pixel signal larger than the most probable one for a full track-traversal of the pixel does not contain useful position information. Energetic delta rays often cross pixel cell boundaries causing strong charge correlations between adjacent pixel cells and sometimes causing unusual cluster shapes.

The single most important feature of pixel clusters is that the shape of the x -projection of the cluster is independent of the y -position of the hit and the y -size of the cluster (independent of the angle β). Similarly, the shape of the y -projection of the cluster is independent of the x -position of the hit and the x -size of the cluster (independent of the angle α). This x - y factorization is a

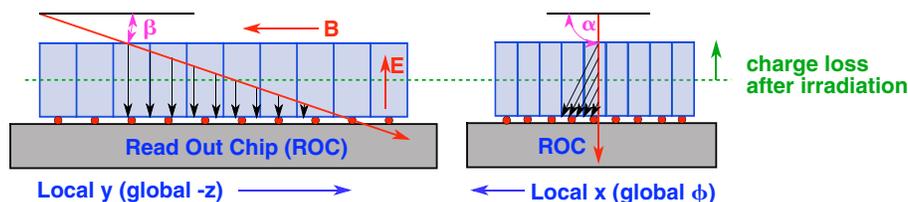


Figure 1: Geometrical and Lorentz-drift induced in charge sharing in a “flipped” pixel barrel module.

consequence of the facts that the field configurations in the pixels don’t couple the two coordinates except perhaps in the corners of the cells where there are small 2-D focusing effects and that the pixels have a periodic structure. This property of the system is heavily exploited by the standard CMS reconstruction algorithm [6] and by the template algorithm described in this note. They sum the x - and y - charge projections of the two-dimensional clusters and treat the projections or profiles independently. The y -profiles for a large sample of $\beta = 15^\circ$ tracks that were measured from several test sensors are shown in Fig. 3. Note that the unirradiated sensor (fluence $\Phi = 0$) has a rectangular profile with well defined edges. The average signal in the interior pixels of the projection is constant as expected. These (projected) pixels contain no precise position information. A simple analysis shows that if one assumes that the statistical uncertainty on a signal $s(y)$ is proportional to \sqrt{s} , then the uncertainty on the parameter y , δy , is given by the following expression,

$$\delta y = C \frac{\sqrt{s(y)}}{ds/dy} \quad (2.1)$$

were C is a constant. This suggests that most of the position information is contained in the smaller signals near the cluster edges where the slope is largest. After irradiation to fluence $\Phi = 8 \times 10^{14} \text{ n}_{\text{eq}}/\text{cm}^2$, charge trapping causes the cluster to have a bias-voltage-dependent shape. Note that although charge is preferentially lost from the “far” end of the cluster, the interior pixels now contain position information. A summary of the key features of the pixel clusters follows:

- The shapes of x - and y -projections of the two-dimensional pixel clusters are independent.
- There is no precise position information in very large pixel signals. Once the maximum signal is exceeded, one only learns about the likelihood of energetic delta ray emission (still useful information).
- The best position information is contained in the small pixel signals near the cluster ends.

3. Standard Reconstruction Technique

The standard technique for the reconstruction of pixel hits that is implemented in the CMS offline software (CMSSW) is an “eta-like” technique that uses the signals in the first and last pixels of the x and y cluster projections $P_{F/L}^{x/y}$. The use of the first and last projected pixel charges reduces

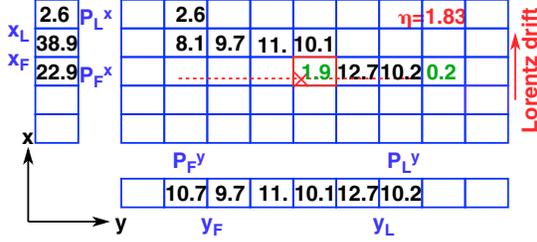


Figure 2: Cluster shape for the barrel hit shown in Fig. 1. The signals in each pixel are given in kiloelectrons. Those shown in green are below the readout threshold. The track projection is shown as the dashed red line. The x - and y -projections are also shown as one-dimensional arrays. The coordinates of the boundaries between the first and second pixels (x_F/y_F) and the next-to-last and last pixels (x_L/y_L) and the charges of the first and last pixels $P_{F/L}^{x/y}$ are also shown.

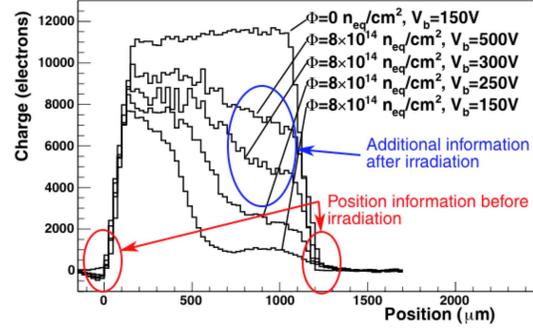


Figure 3: Charge collection profiles for $125 \times 125 \mu\text{m}^2$ test sensors illuminated by a $\beta = 15^\circ$ test beam. An unirradiated sensor (fluence $\Phi = 0$) is compared with a heavily irradiated sensor (fluence $\Phi = 8 \times 10^{14} \text{ n}_{\text{eq}}/\text{cm}^2$) operated at several bias voltages.

the sensitivity of the procedure to delta ray emission which becomes quite likely in long clusters. The reconstructed hit coordinates in each projection are given by the following expressions [6],

$$x_{\text{rec}} = \frac{x_F + x_L}{2} + \frac{P_L^x - P_F^x}{P_L^x + P_F^x} \cdot \frac{W_{\text{eff}}^x(\cot \alpha)}{2} - \frac{\Delta_x}{2} \quad (3.1)$$

$$y_{\text{rec}} = \frac{y_F + y_L}{2} + \frac{P_L^y - P_F^y}{P_L^y + P_F^y} \cdot \frac{W_{\text{eff}}^y(\cot \beta)}{2} - \frac{\Delta_y}{2} \quad (3.2)$$

where: $x_{F/L}$ and $y_{F/L}$ are the coordinates of the first/second and last/next-to-last pixel boundaries (defined in Fig. 2), W_{eff}^x and W_{eff}^y are the total charge widths in the end pixels (defined below), and Δ_x and Δ_y are the maximum Lorentz-drift in the x - and y -directions. Note that Δ_y vanishes in the pixel barrel but is non-zero in the pixel endcaps. The effective charge widths in the end pixels of the two projections are given by the following expression

$$W_{\text{eff}}^x(\cot \alpha) = |T \cot \alpha + \Delta_x| - (x_L - x_F) \quad (3.3)$$

$$W_{\text{eff}}^y(\cot \beta) = |T \cot \beta + \Delta_y| - (y_L - y_F) \approx \frac{\text{pitch}_F^y + \text{pitch}_L^y}{2} \quad (3.4)$$

where: T is the sensor wafer thickness, and $\text{pitch}_{F/L}^y$ are the pitches of the first and last pixels in the y -projection. These expressions are valid for all clusters even those that contain the double-size pixels that are present at the edges of the readout chips. The use of the average pitch size to approximate W_{eff}^y makes it insensitive to the track direction and appropriate for the first pass of a two pass hit reconstruction algorithm without sacrificing much resolution. Problems do arise,

however, when equations 3.1 and 3.2 are used to reconstruct hits in a radiation damaged detector. After an exposure of 6×10^{14} n_{eq}/cm², the residual distributions develop biases of 30-50 μm and the resolutions are significantly worsened. To overcome these difficulties, a new technique that uses a priori information to fit the entire projected cluster shapes was developed. It is based upon a detailed simulation that was developed to interpret several beam test measurements. The following sections describe the simulation and the new simulation-based reconstruction technique.

4. Pixelav Simulation

The detailed sensor simulation, Pixelav [8], incorporates the following elements: an accurate model of charge deposition by primary hadronic tracks (in particular to model delta rays) [9]; a realistic electric field map resulting from the simultaneous solution of Poisson's Equation, carrier continuity equations, and various charge transport models; an established model of charge drift physics including mobilities, Hall Effect, and 3-D diffusion; a simulation of charge trapping and the signal induced from trapped charge; and a simulation of electronic noise, response, and threshold effects.

Several of the Pixelav details described in [8] have changed since they were published. The commercial semiconductor simulation code now used to generate a full three dimensional electric field map is the ISE TCAD package [10]. The charge transport simulation originally integrated the position and velocity equations which required very small step sizes to maintain stability. It was modified to integrate only the position equation by using the fully-saturated drift velocity,

$$\frac{d\vec{r}}{dt} = \frac{\mu \left[q\vec{E} + \mu r_H \vec{E} \times \vec{B} + q\mu^2 r_H^2 (\vec{E} \cdot \vec{B}) \vec{B} \right]}{1 + \mu^2 r_H^2 |\vec{B}|^2} \quad (4.1)$$

where $\mu(\vec{E})$ is the mobility, $q = \pm 1$ is the sign of the charge carrier, \vec{E} is the electric field, \vec{B} is the magnetic field, and r_H is the Hall factor of the carrier. The use of the fully-saturated drift velocity permits much larger integration steps and significantly increases the speed of the code. A final speed enhancement results from the implementation of adaptive step sizing in the Runge-Kutta integrations using the Cash-Karp embedded 5th-order technique [11]. Pixelav was developed to use the vector (SIMD) processing on the PowerPC G4 and G5 families of processors. A port to the less capable Intel SSE architecture has recently been performed. Early testing indicates that the speed of the ported code running on a 2.8 GHz Xeon is approximately 50% of the speed achieved on a 2.5 GHz G5 processor.

The simulation was originally written to interpret beam test data from several unirradiated and irradiated sensors. It was extremely successful in this task, demonstrating that simple type inversion is unable to describe the measured charge collection profiles in irradiated sensors and yielding unambiguous observations of doubly-peaked electric fields in those same sensors [12]. In these studies, charge collection across the sensor bulk was measured using the "grazing angle technique" [13]. As is shown in Fig. 4, the surface of the test sensor was oriented by a small angle (15°) with respect to the pion beam. Several samples of data were collected with zero magnetic field and at temperature of -10°C . The charge measured by each pixel along the y direction sampled a different depth z in the sensor. Precise entry point information from the beam telescope was used to produce finely binned charge collection profiles.

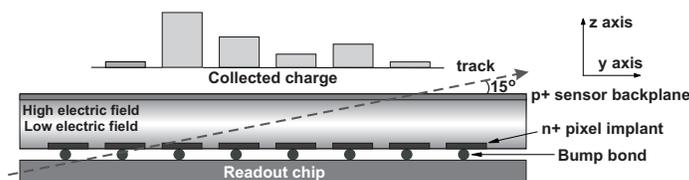


Figure 4: The grazing angle technique for determining charge collection profiles. The charge measured by each pixel along the y direction samples a different depth z in the sensor.

The charge collection profiles for a sensor irradiated to a fluence of $\Phi = 5.9 \times 10^{14} \text{ n}_{\text{eq}}/\text{cm}^2$ and operated at bias voltages of 150 V, 200 V, 300 V, and 450 V are presented in Fig 5. The measured profiles are shown as solid dots and the Pixelav-simulated profiles are shown as histograms. They are compared with Pixelav simulations based upon the electric field produced by a tuned two-trap model [12].

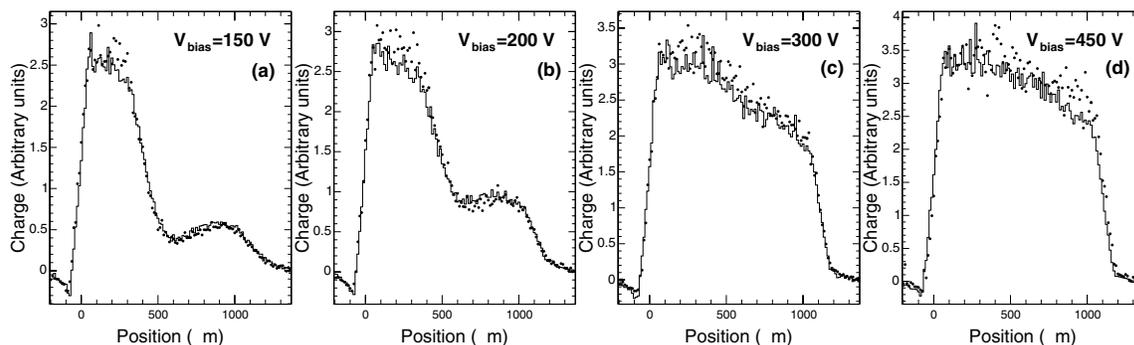


Figure 5: The measured charge collection profiles at a temperature of -10°C and bias voltages of 150 V, 200 V, 300 V, and 450 V are shown as solid dots for a fluence of $5.9 \times 10^{14} \text{ n}_{\text{eq}}/\text{cm}^2$. The two-trap double junction simulation is shown as the solid histogram in each plot.

The simulation describes the measured charge collection profiles well both in shape and normalization (the charge scale of the data is uncertain at the 10% level). The apparently unphysical “wiggle” observed at low bias voltages is actually the signature of a doubly peaked electric field having a minimum near the midplane of the sensor and maxima at the n^+ and p^+ implants. The relative signal minimum near $y = 700 \mu\text{m}$ corresponds to the minimum of the electric field z -component, E_z , where both electrons and holes travel only short distances before trapping. This small separation induces only a small signal on the n^+ side of the detector. At larger values of y , E_z increases causing the electrons drift back into the minimum where they are likely to be trapped. However, the holes drift into the higher field region near the p^+ implant and are more likely to be collected. The net induced signal on the n^+ side of the detector therefore increases and creates the local maximum seen near $y = 900 \mu\text{m}$.

5. Template Reconstruction Algorithm

The template-based reconstruction algorithm is a procedure that translates pre-stored cluster

projection shapes, also called “templates”, across measured cluster projections to find the best fit and hence an estimate of the hit position in both x and y . The Pixelav simulation is used to generate the templates which are stored as functions of $\cot \alpha$ and $\cot \beta$ along with large quantities of auxiliary information in a template object. The following sections describe this procedure.

5.1 Motivation

One of the original motivations for the template-based reconstruction algorithm was the realization that radiation damage would significantly change the charge sharing functions of the detector during large portions of its useful life. Any reconstruction algorithm that was not tunable would become biased and non-optimal as the detector ages. Another motivation was the observation [shown in Fig. 3] that the interior pixel signals in the y -projections of long barrel clusters would acquire position sensitivity as the detector ages. The “Standard” algorithm uses only the end pixels of the projections which is nearly optimal before aging but becomes less so after irradiation. The implementation of an algorithm that uses all of the (projected) pixel information was an obvious choice. Since Pixelav had demonstrated that it could describe the behavior of a heavily irradiated detector and since we had demonstrated that we could tune that description, it seemed obvious to base a more capable algorithm on the detailed simulation. This has numerous advantages in implementation over a purely data-driven approach. Once the detailed simulation has been tuned, it can generate cluster shapes, predict resolutions, and provide goodness-of-fit normalizations for a large range of track angles and cluster charges independently of other detector subsystems and their state of operation (ie alignment). In effect, Pixelav becomes a “software test beam” replacing the very limited pixel beam test data available.

5.2 First Pass Template Generation

The template algorithm requires a-priori knowledge of the projected cluster shapes as functions of $\cot \alpha$, $\cot \beta$, and the hit position. This information is extracted from 30000-event samples simulated by Pixelav at fixed track angle and random hit position. The charge distributions of the samples manifest significant Landau tails due energetic delta ray emission. Since delta-rays distort the cluster shapes, the template generation procedure utilizes only those events having less than the average cluster charge Q_{avg} . This retains approximately 70% of the (asymmetrically-distributed) sample and yields an accurate determination of the projected cluster shapes as caused by the geometrical, charge drift, trapping, and charge induction effects. Note that the determination of the average cluster shapes is quite insensitive to the exact value of the cluster charge requirement.

The template generation is done in two passes. The first pass processing is described in this section and the second pass is described in section 5.3. During the first pass, the x - and y -projections of simulated clusters corresponding to fixed track angles are averaged into respective 7-pixel by 9-bin and 21-pixel by 9-bin arrays. By construction, the struck pixel is labelled as pixel 0 in both projections. The simulated x and y coordinates of the hit are each binned in bins of width 0.125 pixel pitch. The bins are chosen so that the middle bin is centered on the pixel center and the end bins are centered of the pixel boundaries. This yields 9 bins spanning the pixel 0 where the end pixels differ by a full pixel pitch as shown for the y -projection in Fig. 6. The average signal profile for all hits in each bin is stored in the 7-pixel or 21-pixel direction of the arrays. A template

entry therefore consists of the average signal $S_{i,j}^{y/x}$ in each projected pixel i and bin j at fixed α and β angles. The y -templates derived from simulated samples corresponding to unirradiated and heavily irradiated (fluence $\Phi = 6 \times 10^{14} \text{ n}_{\text{eq}}/\text{cm}^2$) $\cot\beta = 1.97$ samples are shown in Fig. 6. Note that trapping reduces the projected signals but produces apparently larger clusters from charge induction. The application of the 2500 electron readout threshold actually reduces the observed cluster size.

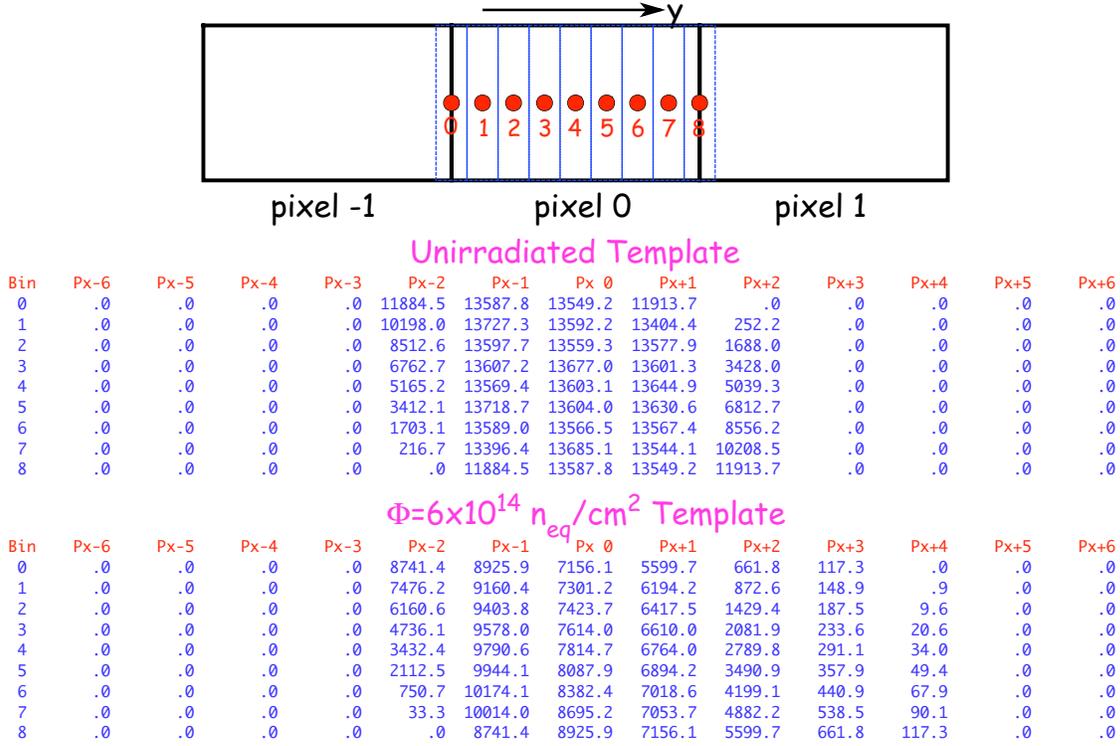


Figure 6: The signal averages $S_{i,j}^y$ of 13 of the pixels in the y -projection of $\cot\beta = 1.97$ barrel clusters for each of 9 bins in the y hit position. They are shown for unirradiated and heavily irradiated sensors ((fluence $\Phi = 6 \times 10^{14} \text{ n}_{\text{eq}}/\text{cm}^2$).

The same procedure is also used to calculate the expected rms, $\Delta S_{i,j}^{y/x}$ of the average signals. The $\Delta S_{i,j}^{y/x}$ for the unirradiated $\cot\beta = 1.97$ sample are plotted vs the projected signals in Fig. 7. The signal/rms points for pixels from either side of the cluster projection are shown as different colors. The signals from the “near” side, the side with the shorter carrier drift path to the readout chip, are shown as red points. The signals from the “far” side, the side with the longer carrier drift path, are shown as blue points. The two sets of points are fit to independent functions of the form

$$\Delta S_{i,j}^{y/x} = \sqrt{a + bS_{i,j}^{y/x} + c(S_{i,j}^{y/x})^2 + d(S_{i,j}^{y/x})^3 + e(S_{i,j}^{y/x})^4} \quad (5.1)$$

where a-e are constants. The best fits to the near and far side data are shown as red and blue solid curves in Fig. 7. Note that there are essentially no differences between the cluster ends for an unirradiated sensor and that the RMSs scale dominantly as $\Delta S_{i,j}^{y/x} \propto \sqrt{S_{i,j}^{y/x}}$. The same information

is shown for the irradiated sensor in Fig. 8. Note that the fluctuations of the far end are significantly reduced by charge trapping and that the scaling of $\Delta S_{i,j}^{y/x}$ is approximately linear in $S_{i,j}^{y/x}$.

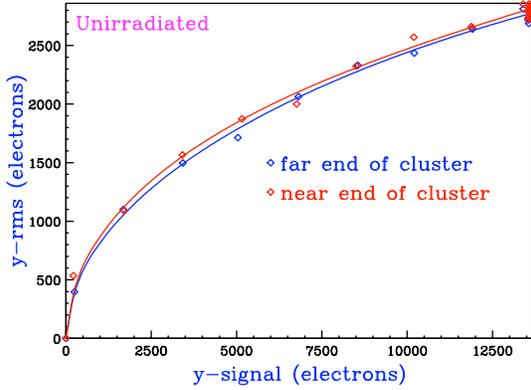


Figure 7: The rms versus signal for the y-projected pixels of the unirradiated $\cot\beta = 1.97$ sample. The signals from the near side, the side with the shorter carrier drift path to the readout chip, are shown as red points. The signals from the far side, the side with the longer carrier drift path, are shown as blue points. The solid curves are best fits to equation 5.1.

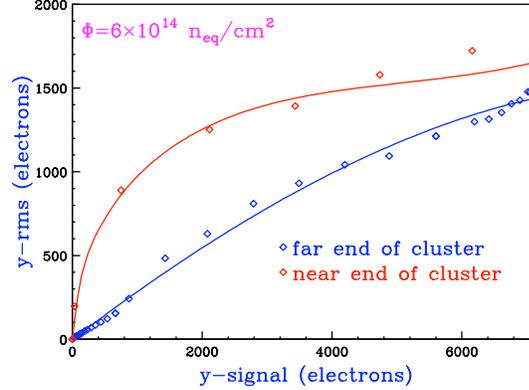


Figure 8: The rms versus signal for the y-projected pixels of the $\cot\beta = 1.97$ sample irradiated to $\Phi = 6 \times 10^{14} \text{ n}_{\text{eq}}/\text{cm}^2$. The signals from the near side are shown as red points and the signals from the far side are shown as blue points. The solid curves are best fits to equation 5.1.

An identical procedure is applied to the x-projection of each cluster at each set of track angles. As was discussed in Section 2, the shapes of the x-projections are independent of $\cot\beta$ and depend upon $\cot\alpha$ only. The normalization of the projected x-signals does depend upon $\cot\beta$, however, the fitting algorithm discussed in Section 5.3 is insensitive to the normalization. Therefore, a single set of x-projections spanning the relevant range in $\cot\alpha$ is sufficient to fit all clusters. The predicted RMS uncertainties of the signals do depend upon $\cot\beta$, however, they do so in a scalable way. This is shown in Fig. 9 where the rms and average x-signals are plotted for three values of $\cot\beta$ corresponding to the three values of pseudorapidity (η): 0.5, 1.5, and 2.0. The best fits for the near and far cluster ends at $\eta = 2.0$ are scaled by the factor $\sqrt{Q_{\text{avg}}(\eta)/Q_{\text{avg}}(2.0)}$ and shown as the dashed ($\eta = 1.5$) and dotted ($\eta = 0.5$) curves. It is clear that a single set of rms functions can be scaled to other values of $\cot\beta$. The actual implementation of the x-template interpolation makes use of this property.

The first pass of the template generation algorithm produces 9-bin templates in both x and y; 5-parameter descriptions of the x-rms and y-rms functions for both near and far ends of the clusters; the average charge Q_{avg} ; and maximum signals for the x- and y-projections, S_{max}^x and S_{max}^y . These are stored in individual files for each each set of track angles. The barrel track angles are chosen to sample the y-cluster length $T \cot\beta$ in 0.25 pixel increments from 0 pixels ($\eta = 0$) to 11.5 pixels ($\eta = 2.5$) [it was found that coarser 0.5 pixel sampling lead to interpolation errors and resolution loss at the 5% level for the worst cases (midway between the points)]. Since displaced vertices produce acceptance tails to $\eta = 2.9$ and the long clusters in this region are very expensive

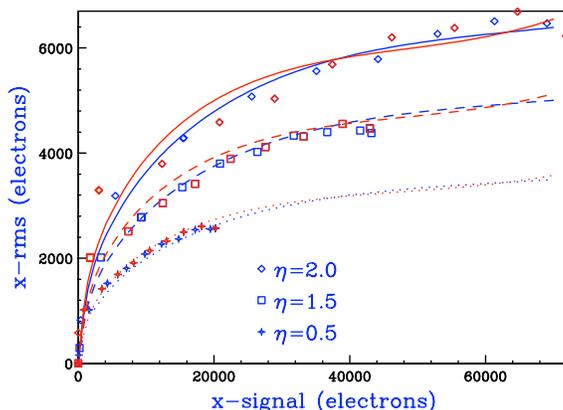


Figure 9: The rms and average x -signals are plotted for three values of $\cot\beta$ corresponding to the three values of pseudorapidity: 0.5, 1.5, and 2.0. The best fits for the near and far cluster ends at $\eta = 2.0$ are scaled by the factor $\sqrt{Q_{avg}(\eta)/Q_{avg}(2.0)}$ and shown as the dashed ($\eta = 1.5$) and dotted ($\eta = 0.5$) curves.

computationally, coarser 0.5 pixel sampling was chosen from 11.5 to 18 pixel y -cluster lengths. The $\cot\alpha$ values are chosen to sample $\alpha' = \alpha - \pi/2$ in 0.075 radian increments from -0.225 to 0.225 radians in the barrel.

5.3 Second Pass Template Generation: Reconstruction Algorithm

The second pass of the template generation uses the pre-stored results of the first pass to apply the actual template reconstruction algorithm to the same data samples used to generate the 9-bin templates. The second pass generates information on biases, errors, corrections, and goodness-of-fit that are combined with the results of the first pass to build a 448 kB ascii template summary file that represents a given set of operating conditions as simulated by Pixelav.

5.3.1 Philosophy and Strategy

A simple description of the template algorithm is that it translates and fits the pre-tabulated projected cluster shapes to the measured projections of real data clusters to estimate the best hit position. This is a loaded statement because the measured signals have large fluctuations caused by delta rays. The delta rays also produce strong correlations in the fluctuations of adjacent pixels. A correct statistical treatment involves considerable technical complexity. Luckily, one can appeal to the observation made in Section 2 that the hit position information is contained primarily in the small signals. Large signals carry no precise information and are also likely to involve the large fluctuations that complicate any analysis. The use of the low Q events to make the 9-bin templates avoided the effects of the fluctuations on the template shapes. Similar advantages in the reconstruction of the entire sample can be achieved by limiting the size of individual pixel signals. The analysis in Section 5.2 also produced expected signal rms's. These are obviously highly biased quantities that apply only to the smaller signals, those that carry position information. However, they are appropriate weights in the definition of a chisquare function that compares the measured cluster shapes with the template shapes. To avoid the complexity of correlations between projected pixel neighbors, a simple diagonal chisquare function is defined. A diagonal chisquare

function cannot be correctly normalized even for clusters with truncated pixel charges, however, the goodness-of-fit criterion can be approximately normalized to account for the deficiencies of the approach. This is a somewhat academic discussion because the performance of the algorithm is quite insensitive to the weighting of projected pixel signals in the chisquare function.

Finally, one should note that the template reconstruction algorithm makes the implicit assumption that there is prior knowledge of the track direction before the algorithm is invoked. The algorithm is therefore suitable for a second pass to refine the estimates of hit position and its uncertainty. This is not a major constraint because of the high granularity of the pixel detector and the relatively large spacing between pixel planes. Any multi-plane pixel-based reconstruction algorithm should be able to establish a sufficiently accurate knowledge of the track direction to achieve the full resolution of this algorithm.

5.3.2 Description of the Template Algorithm

The following is a description of the template algorithm. The reader should note that the template-based approach implicitly incorporates all of the relevant detector physics into the templates themselves. Lorentz drift manifests itself as an offset in the projected cluster shapes with respect to bin number. Non-uniformity of the Lorentz drift modifies the shapes of the templates. Charge loss and trapping makes them asymmetric. This implies that although the templates themselves depend upon which projection is being analyzed, the actual procedure does not depend upon the projected direction. The following description applies to the general reconstruction of a pixel cluster. Not all steps are needed for the second-pass template processing. The differences between these cases are noted.

Preliminary Template Processing: The first step is to interpolate the templates and auxiliary information in $\cot \alpha$ and $\cot \beta$. Simple linear interpolation in $\cot \beta$ is used for all y -related quantities. The x -template is interpolated linearly in $\cot \alpha$ only whereas other x -related quantities are interpolated in both $\cot \alpha$ and $\cot \beta$. Parameterized quantities are not interpolated until after the entire function has been evaluated at each $(\cot \alpha, \cot \beta)$ point. The interpolation step is unnecessary for the second-pass template processing because the requisite information was prepared during the first-pass processing.

The 9-bin templates in x and y are shifted by ± 1 and ± 2 pixels to span the 5 central pixels of the cluster for the possible locations of the x - and y -hit coordinates. This is illustrated in Fig. 10 for the unirradiated y -template shown in Fig. 6. The resulting templates now have 41 bins so that bins 4, 12, 20, 28, and 36 correspond to hit positions at the centers of pixels -2, -1, 0, 1, and 2, respectively. The templates are also padded with zeros to increase their lengths to 25 pixels in y and 11 pixels in x to match the size of the working buffers used to contain the cluster data.

Preliminary Cluster Processing: The total charge of the two-dimensional input cluster is calculated before the individual pixel charges are truncated to a maximum size given by the angle-interpolated value of S_{max}^y . After this truncation (also called “decapitation”) step, the 1-D projections $P_i^{y/x}$ are calculated. These working buffers have lengths 11 in x and 25 in y to accommodate the following processing procedure. Any double pixels are expanded to occupy 2 adjacent elements in the projection arrays where each contains one half of the total double-pixel charge. The

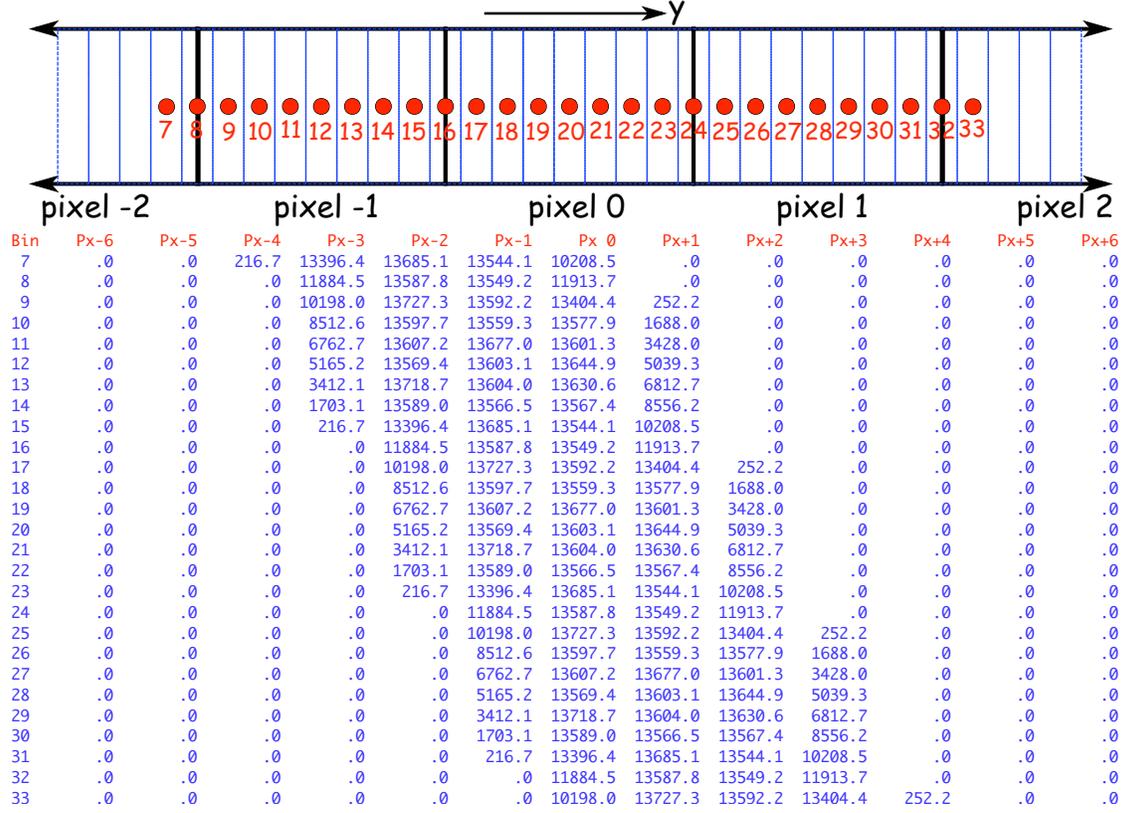


Figure 10: The signal averages $S_{i,j}^y$ of 13 of the pixels in the y -projection of $\cot\beta = 1.97$ unirradiated barrel clusters for 27 of the 41 bins in the y -hit position after shifting the 9-bin template by ± 1 and ± 2 pixels.

first and last pixels of the projections are identified and the clusters are shifted to center them in the projection arrays (the shifts $\text{shift}_{y/x}$ are stored for later use). A set of double pixel flags is also shifted to track the locations of the expanded double pixels. These flags are then used to modify the interpolated templates by replacing the contents of the corresponding adjacent single pixels by their average value. The entire procedure of replacing a single double-pixel with a pair of half-signal single pixels has exactly the same pull in the final chisquare analysis as would have a single entry for a double pixel in the limit that the rms uncertainty on the pixel signals $\Delta P_i^{y/x}$ scale as $\sqrt{P_i^{y/x}}$. Note that the second-pass template processing assumes that all pixels are single size.

A key idea in the template-based algorithm is the recognition that there is important information in the absence of information. Since the readout chip is zero-suppressed, all pixels at the periphery of a cluster must have signals less than the readout threshold P_{min} . To force the fitting procedure to recognize this fact, the two pixels adjacent to each end of of the projected clusters are set equal to $P_{min}/2$ and they are assigned uncertainties $P_{min}/2$. These four “pseudo-pixels” improve the resolution of the algorithm. The use of doubled pseudo-pixels helps to ensure that misaligned clusters and templates always have large values of chisquare even when the input clusters are small.

Initial Chisquare Minimization: The basic goal of the procedure is to translate the expected cluster shape until it best matches the observed cluster shape. This is shown in Fig. 11 where the

y-projection of a cluster is shown as the set of magenta data points. We note that the signal at pixel +2 falls below the readout threshold and is replaced by a green pseudo-pixel. The basic cluster shape, encoded in 1/8 pixel bins is shown as the blue histogram. It is clear that translating the cluster to the left by 2/8 bins would produce much better agreement and suggests that true hit is likely to be at -2 bins in pixel 0. To allow for less than perfect signal height calibration, we allow the overall normalization of the template or of the data to float. This is accomplished by evaluating the following chisquare function for some or all of the template bins,

$$\chi^2(j) = \sum_i \frac{(P_i^{y/x} - N_j S_{i,j}^{y/x})^2}{(\Delta P_i^{y/x})^2} \quad (5.2)$$

$$N_j = \sum_i \frac{P_i^{y/x}}{(\Delta P_i^{y/x})^2} \bigg/ \sum_i \frac{S_{i,j}^{y/x}}{(\Delta P_i^{y/x})^2}$$

where the projected pixel uncertainties $\Delta P_i^{y/x}$ are calculated using equation 5.1 from the pre-stored parameters and interpolated in $\cot \alpha$ and $\cot \beta$ if appropriate (not needed in second-pass template generation). The actual χ^2 minimization search can be performed in several ways that trade-off speed for robustness. The slowest and most robust search evaluates equation 5.2 for each of the 41 bins and finds the absolute minimum. A faster and still secure alternative is to limit the search to the central 25 bins if there are no double-pixels at the ends of the projected cluster and to use the central 33 bins if there is an end double-pixel. Still faster but having slightly less than optimal resolution is to search every fourth bin for a minimum and then to expand the search just to the second nearest neighbors to find another minimum and then to the nearest neighbors of until a group of 3 consecutive bins has been evaluated and a minimum established. This minimization scheme is roughly four times faster than the slowest one but relies on the smooth parabolic shape of the χ^2 function. It works well for most clusters with single-size pixels but must be started at the finer step size for those with double-size pixels.

Position Estimation for Single Pixel Projections: The procedure described to this point is applied to all cluster projections and always results in the bin number and value of the chisquare minimum. For single pixel cluster projections, the chisquare value is stored and a simplified position estimation is performed. The reconstructed position of the hit, y_{rec} or x_{rec} , is determined by correcting the position given by the pixel center for the centering step ($\text{shift}_{y/x}$) and for the bias $D_{1/2}^{y/x}$ where the subscript indicates single and double pixels separately,

$$y_{rec} = y_{pix} - \text{shift}_y - D_k^y \quad (5.3)$$

$$x_{rec} = x_{pix} - \text{shift}_x - D_k^x. \quad (5.4)$$

The bias is determined from the average residual of all one-pixel clusters during the second-pass template generation. It is also calculated separately for single double-pixel clusters by merging adjacent rows and columns of the Pixelav events. This is done with two adjacent pixel pairings to span all possible situations. The bias calculation automatically corrects single pixel clusters for Lorentz-drift and for bias caused by radiation damage which can cause two-pixel clusters to become single pixel clusters. The same procedure is also used to calculate the rms spreads in hit residual for

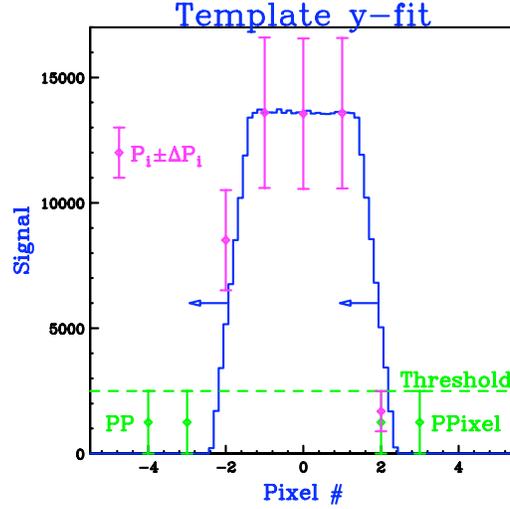


Figure 11: The y -projection of a cluster is shown as the set of magenta data points. The signal at pixel +2 falls below the readout threshold and is replaced by a green pseudo-pixel. The basic cluster shape, encoded in $1/8$ pixel bins is shown as the blue histogram.

single pixel clusters, $\Delta_{1/2}^{y/x}$. These can differ significantly from the usual product of $(12)^{-1/2}$ and the pixel pitch because single pixel clusters often occur only in limited regions hit position depending upon incident track angles and Lorentz drift. The uncertainties on the reconstructed pixel hits, σ_y and σ_x , are taken to be $\Delta_{1/2}^y$ and $\Delta_{1/2}^x$, respectively, for one-pixel projections.

Note that the quantities $D_{1/2}^{y/x}$ and $\Delta_{1/2}^{y/x}$ are interpolated in $\cot\beta$ and $\cot\alpha$ in ordinary hit processing whereas they are actually generated during second-pass template processing.

Position Estimation for Multiple Pixel Projections: For multiple pixel projections, the bin number of the chisquare minimum is used to seed a two-bin interpolation calculation to refine the knowledge of the chisquare minimum in terms of a continuous parameter. This is done by defining the bins adjacent to the minimum bin as bins l and h as is shown in Fig. 12. The chisquare function is then redefined in terms of a linear combination of the functions $S_{i,l}^{y/x}$ and $S_{i,h}^{y/x}$. For simplicity, we drop the x/y superscripts from the quantities and then express the chisquare function for each projection as

$$\chi^2 = \sum_i \frac{\{P_i - N[(1-r)S_{i,l} + rS_{i,h}]\}^2}{\Delta P_i^2} \quad (5.5)$$

$$r = \frac{\sum_i P_i (S_{i,h} - S_{i,l}) / \Delta P_i^2 \sum_i P_i S_{i,l} / \Delta P_i^2 - \sum_i P_i^2 / \Delta P_i^2 \sum_i S_{i,l} (S_{i,h} - S_{i,l}) / \Delta P_i^2}{\sum_i P_i^2 / \Delta P_i^2 \sum_i (S_{i,h} - S_{i,l})^2 / \Delta P_i^2 - [\sum_i P_i (S_{i,h} - S_{i,l}) / \Delta P_i^2]^2} \quad (5.6)$$

where N is a common normalization factor and r is a dimensionless ratio that is bounded by 0 and 1 and determines the position of the χ^2 minimum between the centers of bins l and h . The resulting estimates of the hit position are given by the following expressions,

$$y_{rec} = y_{bin}[l] + r(y_{bin}[h] - y_{bin}[l]) - \text{shift}_y \quad (5.7)$$

$$x_{rec} = x_{bin}[l] + r(x_{bin}[h] - x_{bin}[l]) - \text{shift}_x \quad (5.8)$$

where $y_{bin}[i]$ and $x_{bin}[i]$ are the x and y positions of bin i . The templates corresponding to bins l and h for the example shown in Fig 11 are shown as the blue solid and red dashed histograms in Fig. 13. It is clear that a something close to an $r = 0.5$ combination of the templates will yield the best fit.

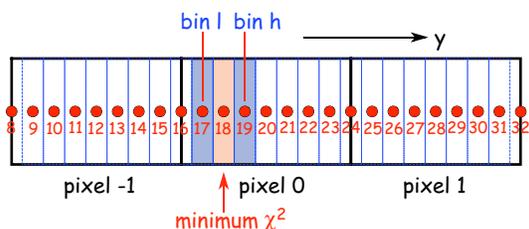


Figure 12: The definition of bins l and h for the example shown in Fig 11.

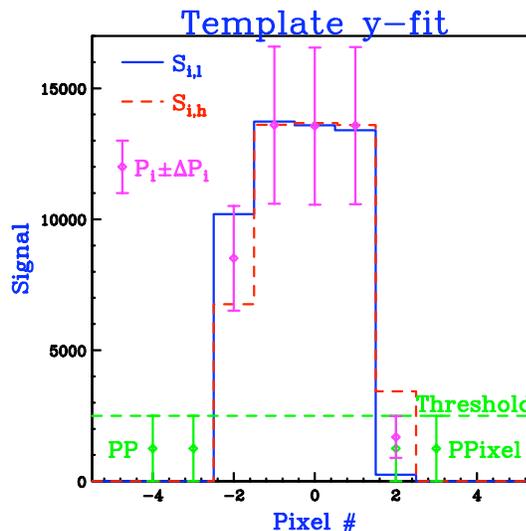


Figure 13: The templates corresponding to bins l and h for the example shown in Fig 11 are shown as the blue solid and red dashed histograms.

Residual Corrections: The second template pass also generates and stores final bias and resolution information for each set of track angles and charge bin. In actual operation, the means of the residual distributions are interpolated in the track angles and are used to correct the final position estimates. The interpolated rms widths are used to estimate the uncertainties of the position estimates. This choice includes any non-Gaussian tails that may be present and represents a better estimate of the true resolution than the Gaussian fit parameters.

Chisquare Probabilities: The second-pass of the template generation stores the averages of the minimum y- and x-chisquare functions for later use in the calculation of goodness-of-fit probabilities. This single parameter is adequate to reproduce the actual distributions.

5.4 Performance

Unfortunately, very limited beam test data are available for the final pixel geometry and read-out chip. All of the available data were collected at normal incidence or near normal incidence so that all clusters have one or two pixel sizes. Therefore, the characterization of the template algorithm depends almost entirely upon simulated data. The native performance of template algorithm was studied by reconstructing large samples clusters generated by Pixelav with random positions and track angles. This work is summarized in Section 5.4.1. There are a number of effects that

are more easily studied with the full CMSSW simulation than with the standalone code. These include the effects of double-size pixels, the effects of detector edges, and the effects secondary particle production upstream the pixel sensors. These are discussed in Section 5.4.2. The production of secondary particles yields cluster shapes that are inconsistent with the track angles and reconstructed hit coordinates that are not well correlated with the position of the primary track. These can be suppressed by the use of the goodness-of-fit information generated by the template algorithm. This is discussed in Section 5.4.3. The reader should note that all of the resolution plots shown in Section 5.4 are root-mean-square (rms) quantities and include the effects of tails.

5.4.1 Native Performance

The performance of the template algorithm is compared with that of the standard algorithm by plotting the rms y - and x -residuals for a sample of reconstructed barrel clusters generated by Pixelav as shown in Fig. 14. The residuals are plotted as functions of pseudorapidity for the two cluster charge (Q) bands $1.5 > Q/Q_{avg} > 1$ (30% of all clusters) and $1 > Q/Q_{avg}$ (70% of all clusters). The clusters were simulated for an unirradiated physical sensor (includes focusing effects near the $n+$ implants) operated at 150 V bias. The rms residuals are used to measure the effects of non-Gaussian tails on the performance of the algorithms. Note that the template and standard algorithms perform similarly in the lower charge band which has less delta-ray activity. Near $\eta = 0$, the projected y -clusters consist of single pixels and have poor resolution. Near $\eta = 0.5$, the y -projections consist of two-pixel clusters and the y -resolution is quite good. It then worsens at larger η where the template algorithm has approximately 10% better resolution. The x -resolutions for the lower charge band improve with increasing η (and increasing Q). The algorithms perform comparably at low η and diverge a bit at large η where the template resolution is about 20% better than the standard resolution. In the larger charge band where there is increased delta-ray activity, the template algorithm has significant advantages over the standard algorithm at nearly all pseudorapidities.

The two algorithms were also compared using a Pixelav-generated sample of clusters from a heavily irradiated physical sensor ($\Phi = 6 \times 10^{14} \text{ n}_{eq}/\text{cm}^2$) operated at 300 V bias. A calibrated template is used to reconstruct these events. The Lorentz-shift used by the standard algorithm is reduced from $121 \mu\text{m}$ to $75.3 \mu\text{m}$ to account for the higher operating bias and the loss of charge sharing caused by trapping. The resulting rms residuals are plotted versus η in Fig. 15 for the cluster charge bands $1.5 > Q/Q_{avg} > 1$ and $1 > Q/Q_{avg}$. We note that the resolutions of both algorithms are degraded, but template algorithm is less affected (as it was designed to be). In particular, the standard algorithm develops large η -dependent bias in the y -direction after irradiation which is reflected in the significant degradation of the y -resolution. The template algorithm has a much smaller intrinsic bias that is automatically corrected.

5.4.2 Performance in CMSSW

The template algorithm was also tested using samples of events with six 20 Ge V muons generated by the CMSSW simulation. Special templates corresponding to the simpler sensor physics in the CMSSW simulation were generated using a highly simplified electric field map that is uniform and does not include focusing effects near the $n+$ implants. Although the CMSSW simulation has

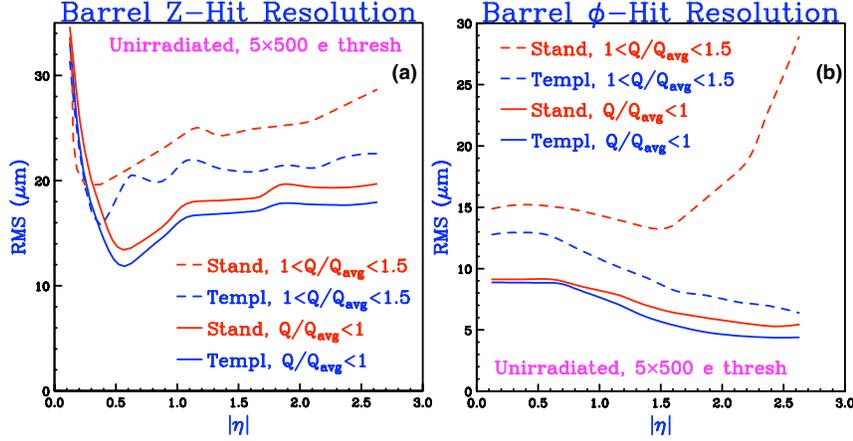


Figure 14: The rms y-residuals (a) and x-residuals (b) of a sample of reconstructed barrel clusters for the template (blue) and standard (red) algorithms are plotted versus pseudorapidity for the cluster charge bands $1.5 > Q/Q_{avg} > 1$ (dashed lines) and $1 > Q/Q_{avg}$ (solid lines). The event sample is generated by Pixelav and models an unirradiated detector operated at 150 V bias.

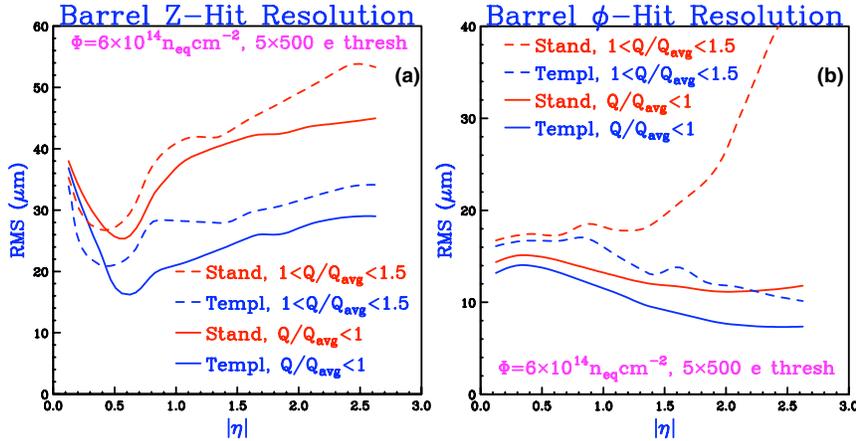


Figure 15: The rms y-residuals (a) and x-residuals (b) of a sample of reconstructed barrel clusters for the template (blue) and standard (red) algorithms are plotted versus pseudorapidity for the cluster charge bands $1.5 > Q/Q_{avg} > 1$ (dashed lines) and $1 > Q/Q_{avg}$ (solid lines). The event sample is generated by Pixelav and models a detector with significant radiation-damage ($\Phi = 6 \times 10^{14} \text{ n}_{eq}/\text{cm}^2$) operated at 300 V bias.

a simplified model of the sensor physics, it does correctly model the geometry of the entire tracking system and the spatial distribution of vertices. The resolution of the pixel tracking system is affected by the presence of double-size pixels at readout chip boundaries and by the sensor edges. Additionally, the CMSSW simulation includes showering by the primary charged particles as they transit the detector.

This problem is illustrated in Fig. 16 which shows the CMSSW-generated cluster charge distribution for 10 0.25-slices of pseudorapidity. The charge distributions for primary muons are shown in red and black and the charge distribution for secondary electrons is shown in magenta. Edge

clusters appear as muons with low cluster charge and become more prominent in the larger- η bins. The secondary electrons are present in all slices but are particularly pronounced at large η where they comprise nearly 20% of all hits. The presence of reconstructed secondary clusters on tracks is somewhat smaller but still comprises approximately 6% of hits in the largest η slice. To help suppress the secondary clusters, the template code stores a minimum charge Q_{min} for each cot β entry. If the cluster charge is found to be less than Q_{min} , a flag is set. This allows the rejection or flagging of high- η , low-charge clusters with no loss of efficiency but does not suppress the low- η secondaries. A more powerful discriminant based upon template probabilities is discussed in Section 5.4.3.

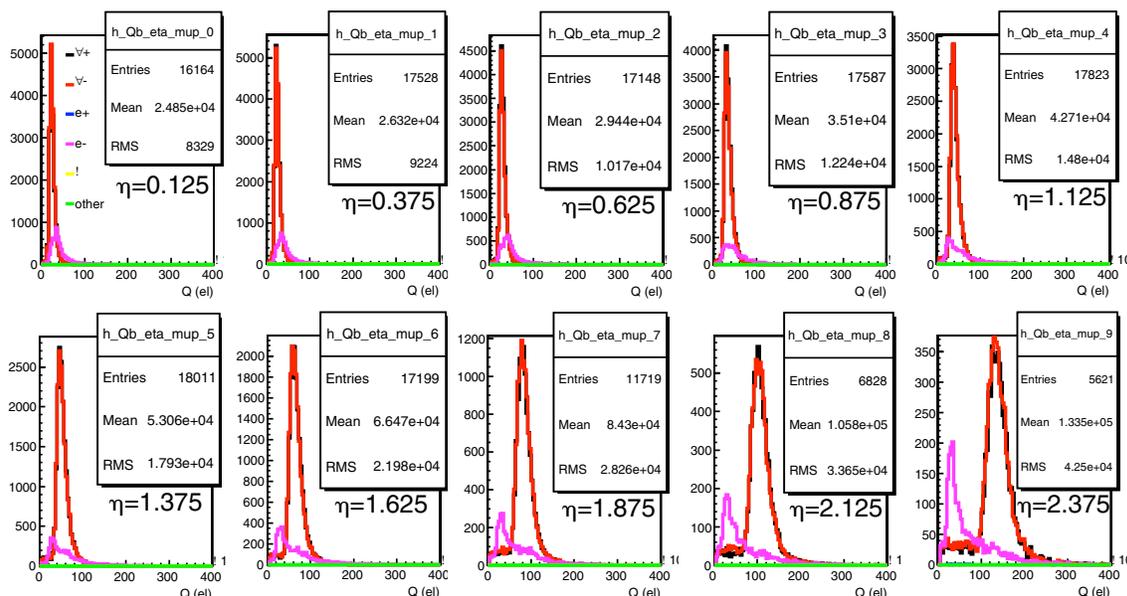


Figure 16: The charge distribution of clusters (not necessarily track associated) in 10 slices of η for primary muons (red/black) and secondary electrons (magenta) from a sample of CMSSW-generated muon events.

A comparison of the template and standard algorithms operating on a sample of CMSSW-generated clusters with all effects is shown in Fig. 17. The rms resolutions in the two charge bands are shown in the standard algorithm (red lines), the template algorithm (blue lines), and the template algorithm after the low charge clusters have been removed with a simple charge cut (green lines). Note that the template algorithm still outperforms the standard one with all effects present and improves further when the low charge clusters are removed.

5.4.3 Cluster Shape Information

The template algorithm was designed to use the a-priori cluster shape information generated by Pixelav to optimize the resolution of the hit reconstruction. As part of the process, it minimizes the chisquare function defined by equations 5.2 and 5.5. This also provides information that can be used to test the compatibility of the observed cluster shapes with the shapes expected for the input track angles. In order to interpret the minimum chisquare information, the simple charge-bin-dependent one-parameter description discussed in Section 5.3.2 was implemented. These are

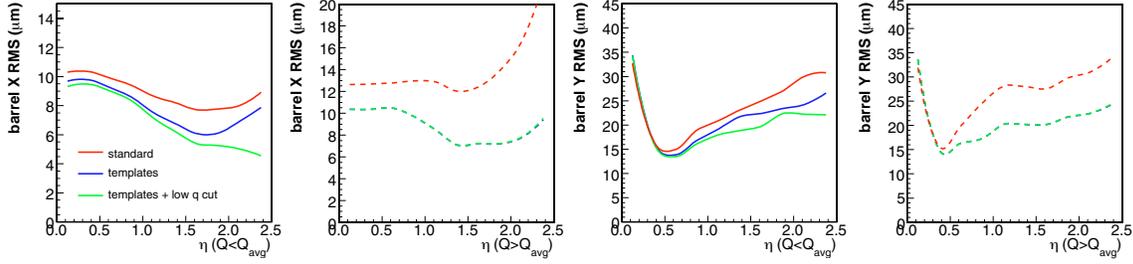


Figure 17: The rms y -residuals and x -residuals of a CMSSW-generated sample of standard- and template-reconstructed barrel clusters are plotted versus pseudorapidity for the cluster charge bands $1.5 > Q/Q_{avg} > 1$ and $1 > Q/Q_{avg}$. The red curves show the standard algorithm, the blue curves show the template algorithm and the green curves show the template algorithm after the removal of the low charge clusters. Note that the green and blue curves are coincident for the $1.5 > Q/Q_{avg} > 1$ band.

interpolated in the track angles and are applied to the problem in the usual way to estimate the chisquare tail probabilities in each projection,

$$\text{Prob}_{y/x} = 1 - \Gamma(\bar{\chi}_{y/x}^2/2, \chi_{y/x}^2/2) \quad (5.9)$$

where: Γ is the incomplete Gamma function, $\bar{\chi}_{y/x}^2$ is the expected average of the distribution, and $\chi_{y/x}^2$ is the minimum determined from the template algorithm.

There are two distinct uses for the goodness-of-fit information. The first is to validate that the cluster is likely to have been produced by the transit of primary charged particle. It was shown in Section 5.4.2 that some of the clusters observed in the CMSSW simulation are produced by secondary showers. These can have the wrong cluster charge or shape and should not be included in reconstructed tracks. The template probabilities are a useful tool to reject these. The second application is to test the compatibility of the cluster with the track angle hypotheses. In principle, this could be a powerful constraint in track seeding and is discussed in Section 5.6. In either use case, it is essential to understand the fraction of clusters lost to the y - or x -probability requirements. These inefficiencies are estimated from a large sample of Pixelav clusters and are plotted as functions of the base 10 logarithm of the minimum probability in Fig. 18. The inefficiencies arising from the clusters with large delta ray activity in the largest charge band, $Q/Q_{avg} > 1.5$, are shown as dashed curves. Since these events comprise only 4.5% of the entire sample, it is clear that poorly measured events with large delta rays are disproportionately removed by reasonable values of the minimum probabilities. Furthermore, as one might expect given the factorization of the y - and x -projections, the inefficiencies associated with y - and x -probabilities are largely independent. The total inefficiency of separate y -probability and x -probability requirements is quite close to the sum of the inefficiencies of the individual requirements.

The utility of a minimum probability requirement is illustrated in Fig. 19 which shows the same CMSSW-generated cluster charge distributions that were shown in Fig. 16 after the application of the requirements $\text{Prob}_y > 10^{-3}$, $\text{Prob}_x > 10^{-3}$. We note that the secondary electron induced clusters are eliminated at large η and are suppressed at smaller η . The low charge distribution of edge clusters is also removed. The particular probability requirements shown in Fig. 19 are not

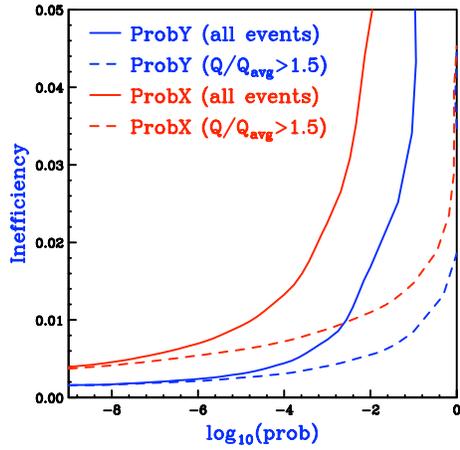


Figure 18: The inefficiencies of y - (blue) and x -probability (red) are shown as functions of the base 10 logarithm of the minimum probability as solid lines. The inefficiencies arising from the loss of poorly measured clusters in the largest charge band, $Q/Q_{avg} > 1.5$, are shown as dashed curves.

optimized. It is clear that one would like to remove hits due to secondaries at the earliest stage of track finding. This suggests that the probabilities should be used in track seed finding and that any probability cuts must be optimized for that purpose. This is discussed in Section 5.6.

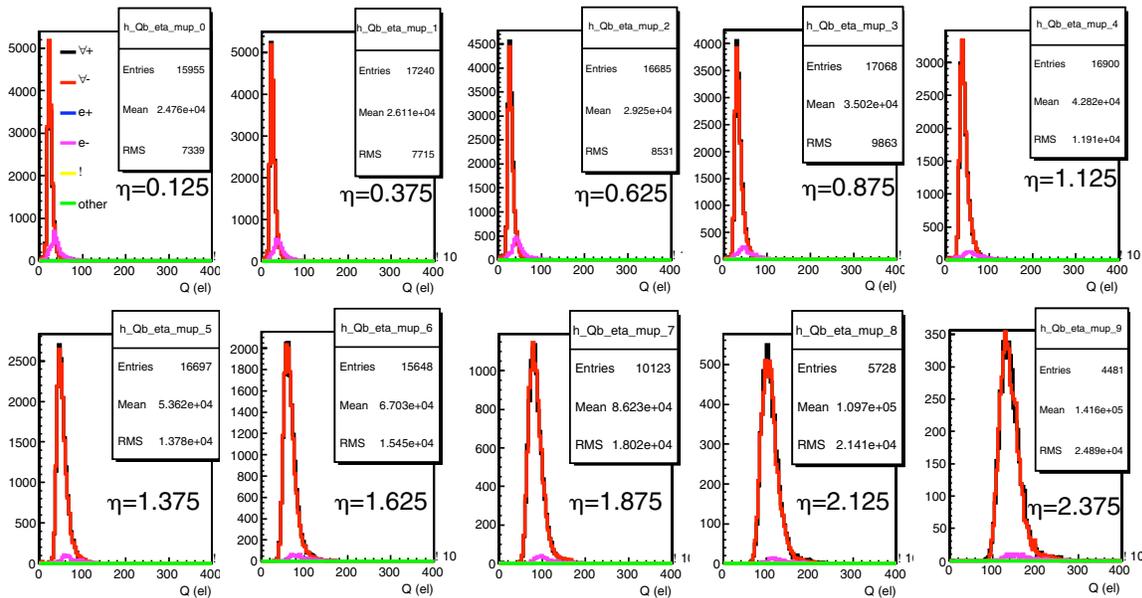


Figure 19: The charge distribution of clusters (not necessarily track associated) in 10 slices of η for primary muons (red/black) and secondary electrons (magenta) from a sample of CMSSW-generated muon events after the application of minimum y -probability and x -probability requirements of 10^{-3} .

POS(VERTeX 2007)035

5.5 Tuning and Calibration

The template algorithm is based upon the premise that the Pixelav simulation can be tuned to accurately describe the pixel sensors as they are gradually damaged by exposure to the large LHC radiation field. This premise rests upon the demonstrated success in the modeling of charge collection profiles measured with test sensors irradiated to several fluences [12]. The parameters of the double-junction model were tuned by hand until the simulation reproduced the profiles measured at the fluence $\Phi_0 = 6 \times 10^{14} \text{ n}_{\text{eq}}/\text{cm}^2$ as is shown in Fig. 5. Although this was extremely tedious, it was also shown that the parameters of the model could then be scaled to lower fluences using separate scale factors for the acceptor density N_A , donor density N_D , and trapping rates $\Gamma_{e/h}$:

$$N_A(\Phi) = R_A(\Phi)N_A(\Phi_0), \quad N_D(\Phi) = R_D(\Phi)N_D(\Phi_0), \quad \Gamma_{e/h}(\Phi) = R_\Gamma(\Phi)\Gamma_{e/h}(\Phi_0) \quad (5.10)$$

where the scale factors R are given by the following expressions,

$$R_\Gamma(\Phi) = \frac{\Phi}{\Phi_0}, \quad R_A(\Phi) = R_\Gamma(\Phi)(1 + \alpha), \quad R_D(\Phi) = R_\Gamma(\Phi)(1 - \alpha) \quad (5.11)$$

and where α depends upon the fluence. The scale factors that were determined at the fluences $2 \times 10^{14} \text{ n}_{\text{eq}}/\text{cm}^2$ and $0.5 \times 10^{14} \text{ n}_{\text{eq}}/\text{cm}^2$ are plotted in Fig. 20. These provide essentially a one-parameter prescription to tune the model to intermediate fluences and should greatly expedite the calibration process.

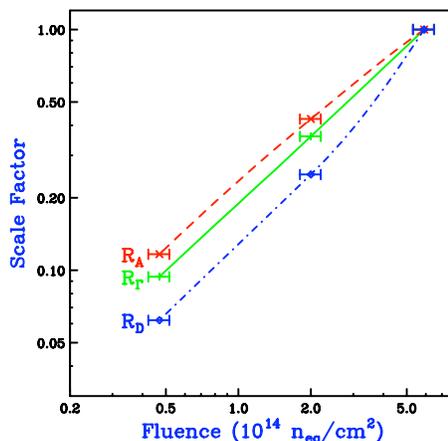


Figure 20: The scale factors R_A , R_D , and R_Γ as defined in equations 5.10 and 5.11 are plotted as functions of fluence.

The actual calibration procedure is to repeat the beam test measurements in-situ in CMS. This requires that samples of large- η tracks and pixel clusters be recorded at a series of pixel bias voltages. It has already been shown [14] that due to displaced primary vertices, it is possible to acquire such samples even in the central barrel modules. In principle, the in-situ measurement could acquire large statistics in only a few hours of dedicated operation at several bias voltages. Since it will require several years of operation to reach fluences comparable to Φ_0 , the calibration procedure will not have to be performed frequently. However, because the readout chip is zero-suppressed, the very useful information in the small tails of the charge collection profiles will not

be available. This is illustrated in Fig. 21 which shows the effect of the readout threshold upon the charge collection profile that was measured in the beam test with an un-suppressed prototype readout chip. The essential tail information is visible at only one bias setting. This implies that finer, carefully-targeted voltage scans will be required to calibrate the sensor model.

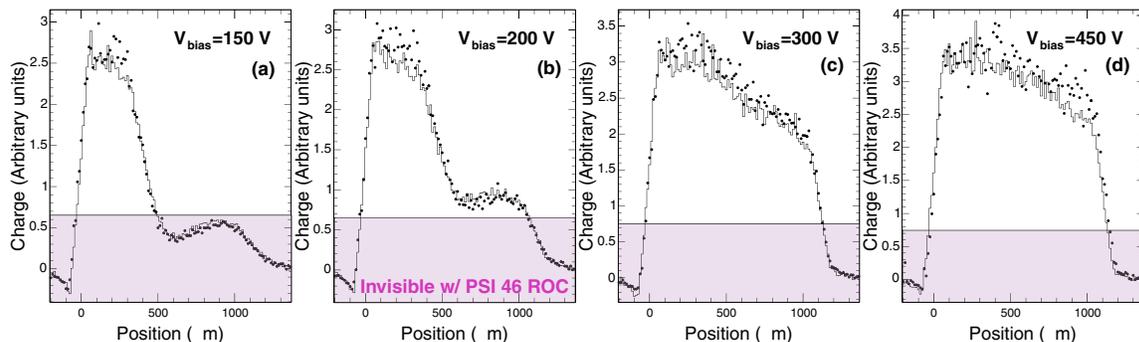


Figure 21: The measured charge collection profiles at a temperature of -10°C and bias voltages of 150 V, 200 V, 300 V, and 450 V are shown as solid dots for a fluence of $5.9 \times 10^{14} \text{ n}_{\text{eq}}/\text{cm}^2$. The two-trap double junction simulation is shown as the solid histogram in each plot and the shaded region shows the effect of the readout threshold. Any signal dot inside the shaded region will be invisible in-situ in CMS.

Unfortunately, the template calibration cannot be automated. The calibration procedure will always require some iteration and hand adjustment of the modeling parameters (the complexity of the model tuning is described in [12]). We are planning to develop a suite of CMSSW software packages to facilitate this. After the implementation of the template-based simulation package in CMSSW (see Section 6) it will be possible to develop this software using data from simulated irradiated sensors.

5.6 Track Seeding

The sensitivity of the template algorithm to the cluster shapes was discussed in Section 5.4.3 in the context of identifying secondary electron backgrounds. The template probabilities test the consistency of the observed cluster shapes with the shapes expected for the input angle hypotheses. This technology can also be used to test the consistency of the observed shapes with the angle hypotheses. Pixel doublets and triplets are used to seed the Kalman Filter track finding algorithm. Each pair of pixel hits defines the β -direction of a possible track and each triplet of hits defines the α -direction of a possible track. The cluster shapes crudely measure both of these angles and can, in principle, be used to “validate” possible track seeds. This idea is sketched schematically in Fig. 22 which shows (not to scale) the y - and x -projections of a possible triplet of pixel hits. We note that the lengths of both cluster projections in the middle layer are inconsistent with the triplet hypothesis. Rejecting inconsistent track seeds before the Kalman Filter is invoked can significantly reduce the track-finding time.

The development of a template-based “seed cleaner” is in progress. The Modified Pixel Seeder processes pixel doublet seeds generated by the Global Pixel Seeder by applying the template algorithm to both pixel hits and by requiring that the y -probabilities exceed 10^{-3} . The results of the application of the seeder and the seeder/cleaner combination to a sample of 750 simulated $t\bar{t}$

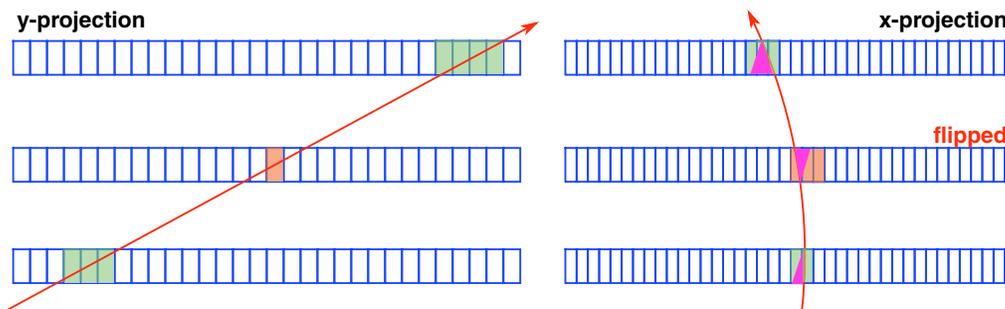


Figure 22: A schematic diagram of a pixel triplet seed and a track hypothesis. Note that the projected sizes of the clusters (shaded green) in the outer layers are consistent with the track angles whereas the projected sizes of the cluster (shaded red) in the middle layer are not consistent with the track angles.

events are summarized in Table 1. The cleaner reduces the number of seeds by more than a factor of two and reduces the tracking time by almost a factor of two. Taking into account the additional overhead of the template algorithm, the total time for seeding and tracking is reduced by 40%. The application of template-based procedure loses about 1.6% of the tracks from the original sample. The quality of the lost tracks is unknown and is currently under study.

Table 1: Comparison of the normal pixel seed builder (Global Pixel Seeder) and a template-based seed cleaner (Modified Pixel Seeder) for a sample of 750 simulated $t\bar{t}$ events.

Quantity	Global Pixel Seeder	Modified Pixel Seeder
Total Seeds (10^3)	1085	476
Total Tracks (10^3)	37.6	37.0
Seeding Time	0.13 s/event	0.19 s/event
Tracking Time	1.80 s/event	0.96 s/event
Total Time	1.92 s/event	1.15 s/event

The use of the template algorithm for seed cleaning has the additional advantage that it can be calibrated to match the degrading performance of the pixel detector. At the very least, this should keep the efficiencies of the seed-cleaner reasonably constant in time even if the rejection power for poor seeds declines. In actuality, the converse may be true. As the detector ages, the template algorithm will acquire the ability to distinguish between positive and negative $\cot\beta$ due to the asymmetry of charge trapping. This may actually improve the ability to reject some backgrounds. The use of the template algorithm to validate pixel seeds should also improve the resolution of track parameters even in the first pass of the track finding since it would automatically reject non-primary particles and would improve the resolution of the reconstructed hits used in the first-pass track fitting.

6. Template-Based Simulation Algorithm

Radiation damage will significantly degrade the performance of the pixel system during its

useful lifetime. It is obviously desirable to reproduce the radiation-induced changes in detector response in the CMSSW simulation. The modeling of irradiated sensors by the vectorized Pixelav code requires approximately 1.5 s per hit on a 2.5 GHz G5 processor and more than 3 s per hit on a 2.8 GHz Xeon processor. This code is obviously much too slow to be integrated into the CMSSW pixel simulation. It is clear that another approach is required. An obvious idea is to try to modify the clusters generated by the standard CMSSW simulation to exhibit the effects of radiation-damage by using the information stored in the templates. In principle, this is a straightforward procedure. It is complicated by the fact that the templates store one-dimensional projections of the two-dimensional clusters but the simulation generates two-dimensional clusters. The following sections describe a technique that can modify the 2-D clusters to achieve the changes in the 1-D projections predicted by the ratios of the templates for the generated and desired events.

The template-based simulation technique has the advantages that it utilizes the same calibration and modeling developed for the reconstruction thereby eliminating a separate calibration for the simulation and it ensures that the simulation and reconstruction processes remain synched as the detector ages.

The template-based simulation technique described in this section is less developed than is the template-based reconstruction technique but a proof of principle has recently been developed and tested. This C++ procedure is analogous to the reconstruction procedure but is not yet fully implemented in the CMSSW simulation.

6.1 Description of the Template-Based Simulation Algorithm

The template-based simulation algorithm re-weights individual pixel signals to modify the one dimensional projections as suggested by the ratios of the one-dimensional templates. This is possible because the number of pixels in a typical cluster N is usually (97% of all cases) less than or equal to the number constraints M provided by the one-dimensional projections. The following algorithm is designed to identify and re-weight the “core” of the cluster. Additional pixels from delta rays are treated in an ad-hoc manner. The procedure retains the fluctuations inherent in the generated clusters but modifies the average projected cluster shapes as suggested by the template information.

Cluster Preparation: The inputs to the algorithm are the CMSSW-generated two-dimensional cluster, track angles, and hit position. The first step is to prepare the input cluster based upon information from templates corresponding to the physics model of the CMSSW simulation. Using the generated track angles and hit position, y - and x -templates corresponding to the “generating” CMSSW model are interpolated and labeled G_i^y, G_j^x . The columns and rows having template signal larger than 50% of the readout threshold define the “inside” region where the re-weighting problem will be formulated. The pixels of the CMSSW input cluster are then categorized and sequentially labeled as inside I_k or outside O_l pixels as shown in Fig. 23. In order to avoid the effects of large signal fluctuations on the re-weighting procedure, the inside signals are then truncated at the same angle-dependent maximum signal $I_{max}(\cot\beta)$ used in the reconstruction procedure. The truncated signals \tilde{I}_k are then summed into y - and x -projections P_i^y and P_j^x as shown in Fig. 24.

Formulation of Re-weighting Problem: The next step is to use the input track angles and hit position to interpolate the y - and x -templates, T_i^y and T_j^x , which correspond to the “target” physical

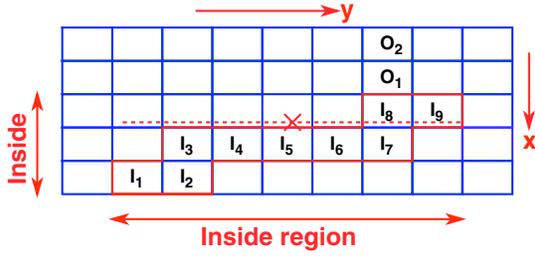


Figure 23: The categorization of the inside I_k and outside O_l pixels in the input cluster .

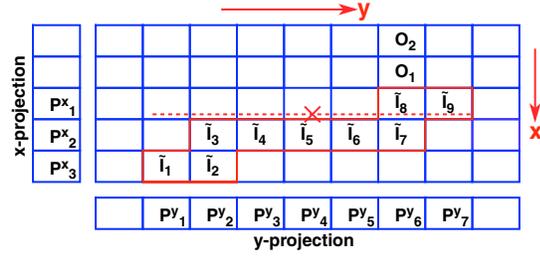


Figure 24: The truncated inside pixels are summed into y - and x -projections.

model. These templates would normally correspond to the Pixelav model of an irradiated detector. The goal of the procedure is to find the N pixel weights r_k that modify the truncated pixel signals so that the re-weighted signals $r_k \tilde{I}_k$ have the y - and x -projections $P_i^y T_i^y / G_i^y$ and $P_j^x T_j^x / G_j^x$ as shown in Fig. 25.

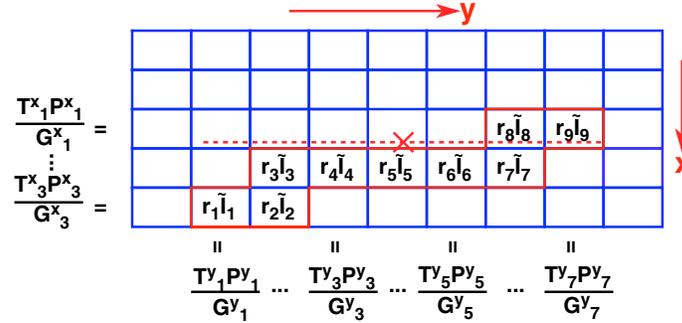


Figure 25: Formulation of the pixel re-weighting problem.

The cluster re-weighting problem is therefore a linear problem in N unknown r_i with M conditions (M is the sum of the numbers of columns and rows in the inside region) and can be expressed in matrix form

$$\mathbf{A} \cdot \mathbf{r} = \mathbf{b} \quad (6.1)$$

where the $M \times N$ matrix \mathbf{A} is composed of truncated signals \tilde{I}_k , the N -vector $\mathbf{r} = (r_k)$, and the M -vector \mathbf{b} contains the re-weighted projections $P_i T_i / G_i$. Unfortunately, standard techniques for the solution of this problem like Singular Value Decomposition (SVD) often yield unphysical (negative) values for the weights. Note however, that when $M > N$, SVD actually minimizes the least square difference $|\mathbf{A} \cdot \mathbf{r} - \mathbf{b}|^2$. Clearly, we would like to do exactly this but with the additional constraints that all r_k be positive. This is a standard problem in the field known as Quadratic Programming. Our problem can be cast into standard form by subtracting a constant from the least square difference and minimizing the new function L ,

$$L = |\mathbf{A} \cdot \mathbf{r} - \mathbf{b}|^2 - |\mathbf{b}|^2 = \mathbf{r}^T \cdot \mathbf{A}^T \cdot \mathbf{A} \cdot \mathbf{r} - 2\mathbf{b}^T \cdot \mathbf{A} \cdot \mathbf{r} = \mathbf{r}^T \cdot \mathbf{Q} \cdot \mathbf{r} + 2\mathbf{c}^T \cdot \mathbf{r} \quad (6.2)$$

where the $N \times N$ symmetric matrix \mathbf{Q} is given by $\mathbf{Q} = \mathbf{A}^T \cdot \mathbf{A}$ and the N -vector \mathbf{c} is given by $\mathbf{c} = -\mathbf{A}^T \cdot \mathbf{b}$. For this study, we were able to find a non-commercial code written mostly in C++ called OOQP for Object-Oriented Quadratic Programming [15]. It is available under a GPL-like license from the University of Chicago and relies upon the BLAS3 linear algebra package [16] and also upon the fortran-coded MA27 sparse linear solver from the HSL archive [17].

Note that double pixels are treated differently in the re-weighting procedure than in the reconstruction procedure. The expansion of double pixels into pairs of single-size pixels simplified the coding of the reconstruction procedure. A similar treatment would only complicate the re-weighting procedure. Therefore, the generated and target templates are modified to model the double pixels by merging appropriate adjacent columns or rows. This is done starting at the central “struck” pixel and proceeds in both directions away from the central pixel.

Final Re-Weighting The actual re-weighting of the cluster depends upon the integers M and N . If M is larger than or equal to N (97% of all cases), OOQP is used to solve for r and the weights are applied to the un-truncated inside pixels. Any outside pixels are reweighted using the weight applied to the nearest inside pixel. This procedure is illustrated in Fig. 26. If OOQP fails to find a solution (approximately 0.15% of all cases) or if M is less than N , a simpler re-weighting is performed. If the number of columns is equal to or larger than the number of rows (normal case), the columns of cluster are re-weighted using the weights T_i^y/G_i^y . If the number of columns is less than the number of rows, the rows are reweighted using the weights T_j^x/G_j^x .

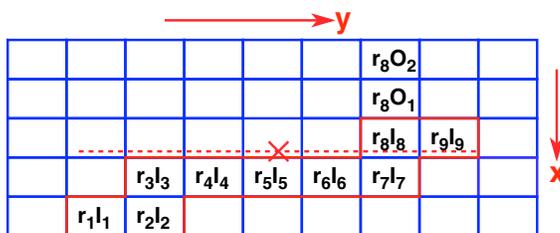


Figure 26: Application of weights to un-truncated inside pixels and outside pixels.

6.2 First Tests: Performance and Speed the Template-Based Simulation Algorithm

The simulation algorithm was tested using clusters generated by Pixelav with a simplified electric field map corresponding to the CMSSW simulation. They were re-weighted to model a sensor that was irradiated to a fluence of $6 \times 10^{14} \text{ n}_{\text{eq}}/\text{cm}^2$. The re-weighted clusters were reconstructed using the template algorithm and the residuals were compared with those produced by reconstructing full Pixelav simulations of the irradiated sensor. Initial testing showed that attempts to re-weight these CMSSW-like events lead to residual distributions with offsets in the 10-11 μm range. It was noted that the input clusters were generated assuming that the detector was operated at the nominal 150 V bias whereas the output template corresponded to an irradiated sensor operated at 300 V bias. The different bias voltage leads to a different average Lorentz drift and therefore different typical topologies of the output clusters. To overcome this problem, another set of CMSSW-like events was generated with Pixelav that corresponded to the uniform field approximation for a detector operated at 300 V bias. The Lorentz angle was reduced from 23° to 16.1°

making the topologies of the generated clusters closer to those of the irradiated detector [5]. Using the new event stream and its corresponding template, the residual offsets of the re-weighted and reconstructed events were reduced to values less than $1 \mu\text{m}$. The resulting rms residuals from this second attempt are compared with those from the full Pixelav simulation of the irradiated sensor as shown in Fig. 27. The local y (global z) residuals for the two simulations are very similar. The local x (global ϕ) residuals are similar but exhibit a somewhat different η dependence. It is clear that the simulation of irradiated sensors will require that the parameters of the CMSSW simulation be “matched” with individual templates to achieve the best results.

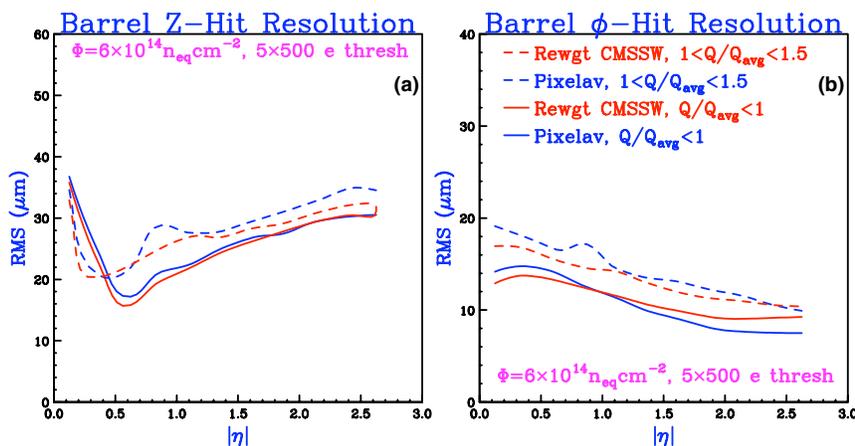


Figure 27: The rms y -residuals (a) and x -residuals (b) of Pixelav-generated (blue) and reweighted CMSSW-like (red) samples of reconstructed barrel clusters using the template algorithm are plotted versus pseudo-rapidity for the cluster charge bands $1.5 > Q/Q_{avg} > 1$ (dashed lines) and $1 > Q/Q_{avg}$ (solid lines). Both samples model an irradiated detector ($\Phi = 6 \times 10^{14} \text{ n}_{eq}/\text{cm}^2$) operated at 300 V bias.

The speed of the algorithm is dominated by the time needed to solve the quadratic programming problem. As tested on a 2.5 GHz G5 processor, the simulation can process approximately 3300 clusters per second which yields 0.3 ms/cluster. The CMSSW digitizer has a tested speed of 12 ms/cluster on a 2.5 GHz Xeon processor [18]. Even allowing for a 10-20% difference in processor speed and the overhead from the unwritten CMSSW interface, the speed of the re-weighting procedure is easily sufficient for use in CMSSW and should be much faster than the 1.5(3) s/cluster speed of the vectorized Pixelav simulation operating on available G5(Intel) processors.

6.3 Future Development

This technique is currently the only alternative for the simulation of the Pixel system after it has become radiation-damaged. We regard the results shown in Fig. 27 as an encouraging proof of principle, but we intend to study and develop the algorithm further before final implementation. Additionally, the CMSSW interface for the template simulation has not yet been designed. Since the algorithm reweights already-generated clusters, there are a number of possible implementation schemes. It could be invoked from within the pixel digitizer. This has the advantage that it could be applied before the response of the readout electronics and electronic noise are simulated. It is also possible to create an independent module that could reweight “noisy” clusters after generation

and before clusterization. The performance of this scheme should be slightly worse than the first, but it simplifies the implementation.

7. Conclusions

This note describes new techniques for the reconstruction/validation and the simulation of pixel hits. The techniques are based upon the use of pre-computed projected cluster shapes or “templates”. A detailed simulation that successfully described the profiles of clusters measured in beam tests of radiation-damaged sensors is used to generate the templates. Although the reconstruction technique was originally developed to optimally estimate the coordinates of hits after the detector became radiation damaged, it also has superior performance before irradiation. The technique requires a priori knowledge of the track angle which makes it suitable for the second in a two-pass reconstruction algorithm. However, the same modest angle sensitivity allows the algorithm to determine if the sizes and shapes of the cluster projections are consistent with the input angles. This information may be useful in suppressing spurious hits caused by secondary particles and in validating seeds used in track finding and has the potential to significantly increase the speed of track finding in the offline reconstruction. The use of the template algorithm at the seeding level would also remove spurious hits from tracks and might further reduce resolution tails.

The implementation of the template reconstruction algorithm in CMSSW is well advanced. It can already be used to reconstruct simulated data. A suite of calibration tools needs to be developed and some additional but straightforward implementation enhancements are needed before the template algorithm could be used to reconstruct real data. Investigations of its use in seeding and track finding are just beginning.

Finally, a new procedure that uses the templates to reweight clusters generated by the CMSSW simulation was described. The first tests of this technique are encouraging and when fully implemented, the technique will enable the fast simulation of pixel hits that have the characteristics of the much more CPU-intensive Pixelav hits. In particular, it may be the only practical technique available to simulate hits from a radiation damaged detector in CMSSW. Additional work is required to finish the algorithm development and to integrate it into CMSSW.

8. Acknowledgements

This work was supported by the National Science Foundation under Grant No. PHY-0457374.

References

- [1] E. Belau, *et al.*, “Charge Collection in Silicon Strip Detectors”, Nucl. Instrum. Meth. **214**, 253 (1983).
- [2] Additional details are reported in CMS Note-2007/033, November 2007. Available: http://cmsdoc.cern.ch/doc/notes/docs/NOTE2007_033.
- [3] D. Bortoletto (CMS Collaboration) “The CMS pixel system”, Nucl. Instrum. Meth. A **579**, 669 (2007).
- [4] CMS Technical Proposal, CERN/LHCC 94-38, LHCC/P1 15 December, 1994.

- [5] A. Dorokhov *et al.*, “Tests of silicon sensors for the CMS pixel detector,” Nucl. Instrum. Meth. A **530**, 71 (2004) [arXiv:physics/0311050].
- [6] The expressions given here are generalizations of similar ones found in S. Cucciarelli and D. Kotlinski, “Pixel Hit Reconstruction, CMS IN 2004/014, May 2004 (CMS Internal Note).
- [7] The CMS Offline Software Project is described at http://cms.cern.ch/iCMS/jsp/page.jsp?mode=cms&action=url&urlkey=CMS_OFFLINE.
- [8] M. Swartz, “CMS pixel simulations”, *Nucl. Instr. Meth.*, **A511**, pp. 88-91, 2003.
M. Swartz, “A Detailed Simulation of the CMS Pixel Sensor”, CMS Note 2002/027, July 2002. [Online]. Available: http://cmsdoc.cern.ch/doc/notes/docs/NOTE2002_027.
- [9] H. Bichsel, Nucl. Instrum. Meth. A **562**, 154 (2006); H. Bichsel, *Rev. Mod. Phys.* **60**, 663 (1988); and private communication (see <http://faculty.washington.edu/hbichsel/MCCsi.html>).
- [10] TCAD 9.0 User’s Manual, Synopsys, Inc., Mountain View CA. USA, <http://www.synopsys.com>.
- [11] J.R. Cash, and A.H. Karp, *ACM Transactions on Mathematical Software*, **16**, 201 (1990); see also W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, “Numerical Recipes in C”, pg 717.
- [12] V. Chiochia, *et al.*, “Simulation of heavily irradiated silicon pixel sensors and comparison with test beam measurements,” IEEE Trans. Nucl. Sci. **52-4**, 1067 (2005), e-print: arXiv:physics/0411143; M. Swartz, *et al.*, “Observation, modeling, and temperature dependence of doubly peaked electric fields in irradiated silicon pixel sensors,” Nucl. Instrum. Meth. A **565**, 212 (2006), e-print: arXiv:physics/0510040.
- [13] B. Henrich, *et al.*, CMS Note 1997/021, March 1997. http://cmsdoc.cern.ch/documents/97/note97_021.pdf
- [14] L. Wilke, *et al.*, private communication.
- [15] E. M. Gertz and S. J. Wright, “Object-Oriented Software for Quadratic Programming”, *ACM Transactions on Mathematical Software* **29**, 58 (2003); online resources: <http://pages.cs.wisc.edu/~swright/ooqp/>.
- [16] L. S. Blackford, *et al.*, “An updated set of basic linear algebra subprograms (BLAS)”, *ACM Transactions on Mathematical Software* **28**, 135 (2002); online resources: <http://www.netlib.org/blas/>.
- [17] HSL Archive can be found at <http://hsl.rl.ac.uk/archive/hslarchive.html>.
- [18] D. Kotlinski, private communication.