

A combined track and hit finding method for a TPC based on local road search with pad pulses

Claus Kleinwort, DESY, Germany

July 10, 2017

Abstract

This document describes a combined local track and hit finding algorithm using a road search with pad pulses developed for a prototype TPC with pad readout aiming for an improved double hit resolution.

1 Introduction

A combined track and hit finding method based on road search with pad pulses has been developed especially for the data taken by the Large Prototype TPC (LPTPC) [1] with GEM-based pad readout at the DESY test beam [2] and implemented as a MarlinTPC [3] processor to improve the double hit resolution. To study this, data collected with a scattering target in front of the TPC to produce events with multiple close-by tracks has been used. Pad pulses describe the timing and size of the pad signals from ionizing particles.

In this setup the drift direction perpendicular to the readout plane is along the Z-coordinate, the tracks roughly move in the X-direction and the pad rows measure Y-coordinates.

The readout plane is segmented into modules. For each module separately track segments and hits are build using both the measurement in the pad row and in the drift

direction. The segments are found by a road search with pad pulses and the hits are constructed from adjacent pad pulses in a row. In this process the tracking information is used to split hits if pads belong to different segments or join hits over empty pads if they belong to the same segment. This allows to find hits only a few pads apart. Otherwise the hit finding is performed similar to the standard ROWBASEDHITFINDERPROCESSOR in MarlinTPC. Matching segments of neighbouring modules are combined to track candidates fitted by a simple helix.

In section 2 the basic concepts are presented, section 3 describes the sequence of processing steps, section 4 shows a brief performance test and section 5 explains the steering parameters of the ROWBASEDPADPULSEROADSEARCHPROCESSOR.

2 Concepts

2.1 Radial ordering

For the pad based readout modules of the LPTPC the pad rows are on concentric circles in the XY-plane. The position in a row measures an azimuthal coordinate and the row number defines the radius and is used as radial ordering parameter:

- The hits on a track are ordered by row number.
- Tracks may have only a single hit in each row.
- For a set of hits the number of rows defines a size.
- The difference of row numbers defines a distance.

2.2 Helical track model

Assuming a constant magnetic field in Z-direction track segments are described by a circle [4] in the XY- and a straight line in the ZS-projection using the average of the positions of the first and last hit as reference point. The five track parameters \mathbf{p} with covariance matrix \mathbf{V}_p defined there are the curvature and for both projections the directions at and distances to the point of closest approach in XY. For zero magnetic field straight lines are fitted in both projections.

2.3 Local charge density

For each pad pulse the average pad charge from adjacent pads in the same row is used to separate them into *leading pulses* as local charge maxima and *trailing pulses*. Only the leading pulses are used for the track finding to reduce combinatorics and to seed hits.

2.4 Pad pulse road

A pad pulse road is a straight line in three dimensions defined by a seeding pair of leading pad pulses or a helix defined by multiple leading pad pulses containing all leading pad pulses within a maximum distance. It is characterised by the number of rows and the row density.

2.5 Pad pulse segment

A pad pulse segment is a collection of pad pulses described by a straight line or helix fit. Pulses on multiple pads in a row are allowed.

2.6 Pad pulse ambiguity

During the segment finding pad pulses can be associated to multiple segments. Before the hit finding those ambiguities have to be resolved.

2.7 Pad pulse hit

A pad pulse hit is seeded by a leading pulse and is the combination of pulses on adjacent pads in a (single) row uniquely associated to the same or no pad pulse segment.

2.8 Equivalence classes

Equivalence classes are used to combine segments to tracks. For a set of segments a *matching* relation is defined for pairs of elements. An equivalence class is the subset of all directly or indirectly related elements. The original set is decomposed in disjunct subsets.

3 Sequence

3.1 Input

Input from each module is the list of (calibrated) TPC pad pulses with pad positions, drift times and pulse charges as illustrated in figure 1. The TPC geometry described by GEAR is used to calculate the azimuthal measurement directions and to define neighbouring modules. In addition, the steering parameters provide a parametrisation of the hit measurement errors ($\sigma^2 = \sigma_0^2 + z \cdot D_{\text{eff}}^2$), the drift velocity to convert drift time into distance and the factor to translate charge from ADC values to primary electrons.

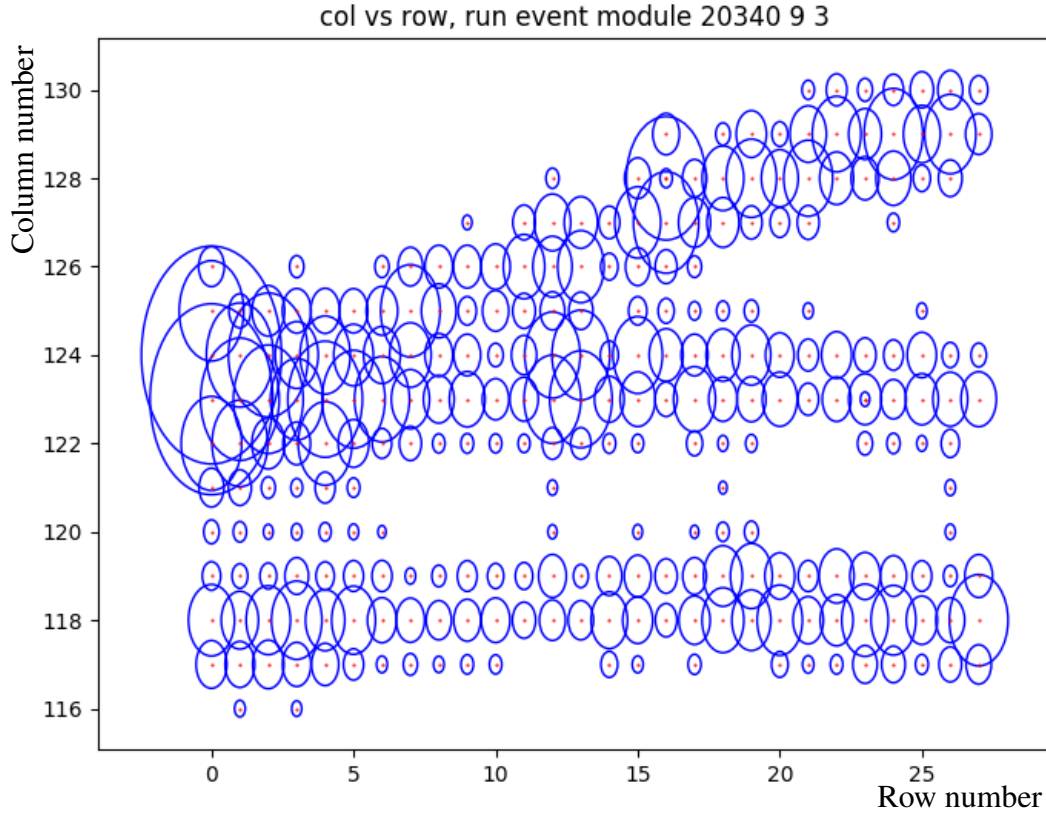


Figure 1: DESY GEM example module with 262 pad pulses. The dots indicate the pad positions and the areas of the elliptical markers the pulse charges.

3.2 On module level

3.2.1 Pulse classification

The local charge density is used to divide the pad pulses into *leading* and *trailing* pulses. For each pad pulse i with charge Q_i in column c_i all pulses j in the same row with $|c_j - c_i| \leq n$ and compatible drift times define its *neighbourhood*. The maximum column difference n should reflect the transversal diffusion and chosen differently for data with and without magnetic field. With N_{pad} as number of pads with pulses in the neighbourhood the relative pulse charge is defined as:

$$Q_{i,\text{rel}} = \frac{N_{\text{pad}} \cdot Q_i}{\sum_j Q_j} \quad (1)$$

Isolated pulses have $N_{\text{pad}} = 1$ and $Q_{i,\text{rel}} = 1$. For leading pulses $Q_{i,\text{rel}} > 1$ is required as depicted in figure 2.

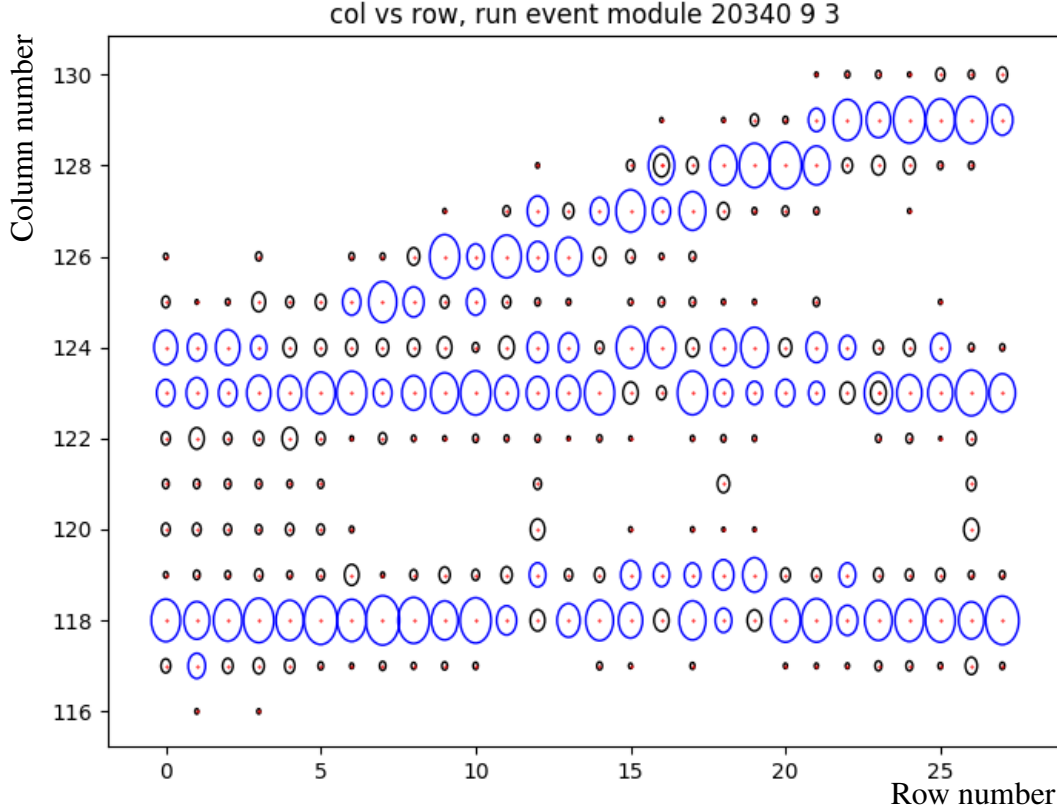


Figure 2: Relative charge in DESY GEM example module. The dots indicate the pad positions and the areas of the elliptical markers the *relative* pulse charges. The 97 leading pulses are shown with blue markers.

3.2.2 Iterative road search

Input are the leading pad pulses. From those all pulses which have the maximum absolute charge in their neighbourhood are selected as *seeding* pulses and sorted in descending order by their relative charge. Optionally pulses marked by the pulse finding as *multiple pulse candidate* or rows with a too large number of seeding pulses are excluded to reduce the combinatorics (e.g. for curling tracks). The road search is performed in two steps:

Initial road: From the sorted list of seeding pad pulses for each pair of unused pulses with a minimal row distance a linear pad pulse road is constructed. The combinatorics scales with the square of the number of seeding pad pulses. The width of the road is few times the pad width and the pad centers are used as coarse position measurement in the XY-plane. From all pulses inside the road the number of rows n_{row} and the first r_{first} and last r_{last} row number are calculated. A road with too few rows or too small row density $n_{\text{row}} / (r_{\text{last}} + 1 - r_{\text{first}})$ is rejected. In addition, there is

a cut on the sum of the squared XY-residuals normalised to the number of pulses. For an accepted road the pulses inside are fitted by a straight line or helix.

Refined road: From the fit parameters of an accepted initial road a refined linear or helical road is constructed. This time the charge weighted average pad position in the pulse neighbourhood is used as *improved XY-measurement*. There are the same requirements on the rows and XY-residuals. An accepted refined road defines a pad pulse segment and all contributing pulses are marked as used by that segment.

If there are unused seeding pad pulses left the procedure restarts with looking for the next initial road.

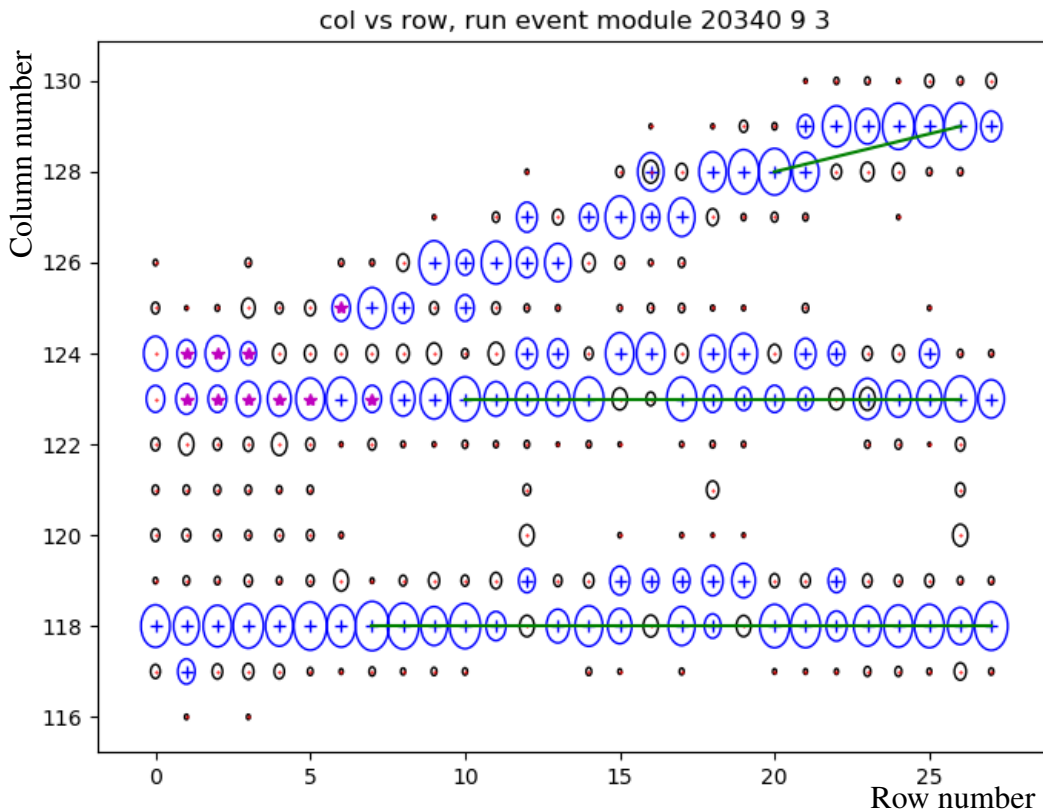


Figure 3: Road search in DESY GEM example module. The dots indicate the pad positions of unused pulses, the crosses of pulses associated to one segment and the stars of pulses associated to multiple segments. The areas of the elliptical markers are proportional to the relative pulse charges (leading pulses in blue). Three partly overlapping segments have been found. The pulse pairs seeding the initial roads are connected by green lines.

3.2.3 Resolution of ambiguities

The road search can associate pad pulses to multiple segments (see figure 3) meaning that segments may share pulses. This may happen if the distance between different tracks gets small. Furthermore, for a single track several largely overlapping segments (seeded by different pairs of pulses) could be reconstructed in difficult cases. After all segments in a module have been found, those ambiguities are resolved in several steps:

Segment cleaning: For all segments the fraction of *ambiguous pulses* (assigned to multiple segments) is calculated. If there are segments with a too large fraction (e.g. larger than 50%) the segment with the largest fraction and all its pulse associations are removed. The fractions are recalculated and the procedure is repeated until no more segment has a too large fraction.

Removal of ambiguous pulses: The ambiguous pulses are removed from all segments and the affected segments are refitted with the remaining pulses.

Reassignment of ambiguous pulses: The ambiguous pulses are reassigned to the previously assigned segment with the closest distance in the XY-plane. Segments with reassignments are refitted again with all pulses.

3.2.4 Addition of trailing pad pulses

For all pad pulse segments a road search with the trailing pad pulses (using the improved XY-measurements) is performed and the pulses inside the roads are associated to them. In case of ambiguities again the segment closest in the XY-plane is selected.

3.2.5 Hit finding

The hit finding as illustrated in figure 4 is performed independently for each row. In a row the leading pad pulses are ordered decreasingly by their charge.

A new hit is seeded by the leading pulse with the largest charge not yet used in a hit. This can be associated to a segment or not. Further unused pad pulses compatible in drift time are added by moving first upwards and afterwards downwards in column (number) until a pulse associated to a different segment (as the seeding pulse) has been found. The hit finding stops at empty pads if the gap is too large or the pulse after the gap doesn't belong to the same segment. Optionally there is a cut on the maximum column distance to the seeding pulse. Hits with too few pad pulses are ignored, otherwise the pad pulses are marked as used by that hit.

As hit parameters the XY-position is determined from the average pad position weighted with the charge, the Z-position is taken from the seeding pulse, the hit charge is the sum of the pulse charges (converted from ADC values to primary electrons) and

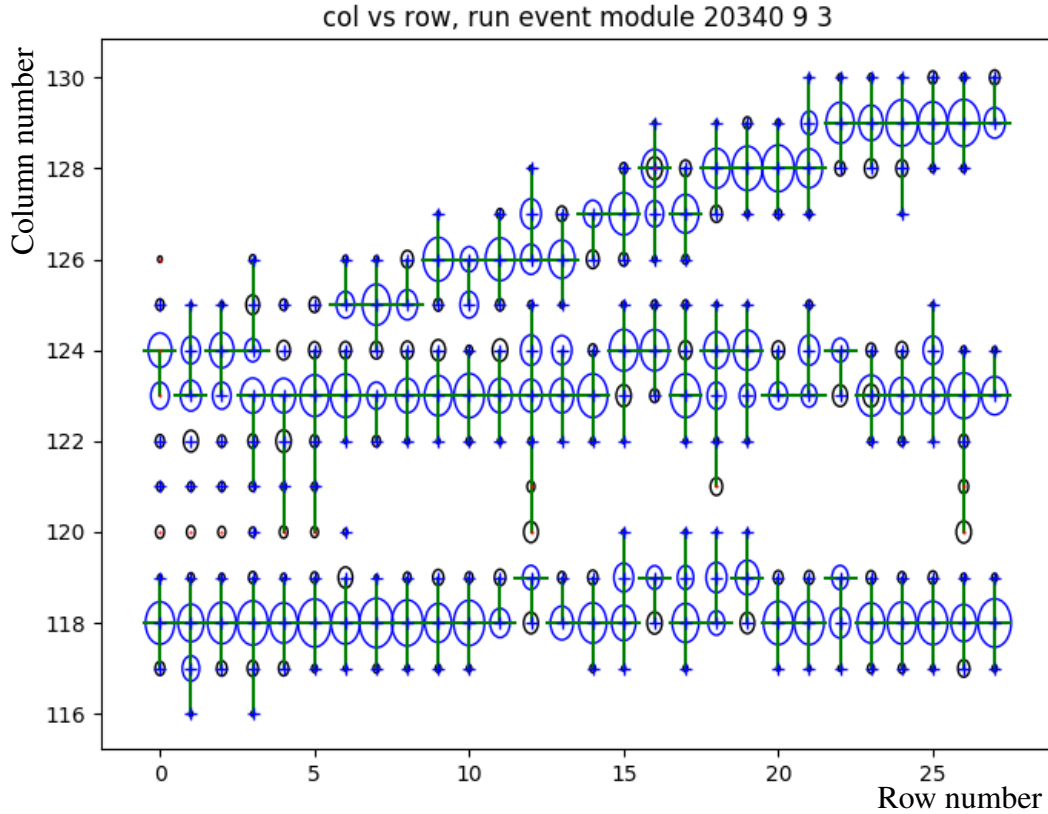


Figure 4: Hit finding in DESY GEM example module. The dots indicate the pad positions of unused pulses and the crosses of pulses associated to a segment. The areas of the elliptical markers are proportional to the relative pulse charges (leading pulses in blue). The vertical green lines connect the pulses in a hit and the horizontal ones mark the seeding pulses. From the 79 found hits 78 are associated to one of the three segments.

the covariance matrix for the hit position is constructed from the corresponding steering parameters (and the Z-Position). In addition, the module ID is encoded into the hit (cell) ID, several quality flags are copied from the contributing pulses and new flags for multiple hit candidates or hits with dead channels or hits at the module border are defined.

After the new hit has been accepted or rejected, the procedure is repeated with the largest still unused leading pulse.

The solution of ambiguities in the assignment of pad pulses to hits using the pad pulse to segment associations is based on proximity. For close-by seeding pad pulses the corresponding hits can only grow outwards leading to biases in the position determination. A possible improvement would be to fit instead the charge distribution in the

pad row with multiple pad response functions.

3.2.6 Track segment definition

For each pad pulse segment a *track* segment is defined (and refitted). Instead of pad pulses it consists of all the hits seeded by pulses from the pad pulse segment.

3.3 Combination of modules

Track segments from individual modules are combined to track candidates as illustrated in figure 5. Each segment is compared to all segments of neighbouring modules where the row gap in-between is less than the average segment size. Track candidates are build from the equivalence classes of matching segments. For each class the list of segments is replaced by a track candidate with the combined list of hits and the track parameters calculated from the weighted average of the segment parameters at a common reference point. For comparing two segments the (n_{par}) parameters and the covariance matrices are propagated to the midpoint of their reference points (as new reference point) and compared there:

$$(\mathbf{p}_{mid,1} - \mathbf{p}_{mid,2}) \cdot (\mathbf{V}_{p,mid,1} + \mathbf{V}_{p,mid,2})^{-1} \cdot (\mathbf{p}_{mid,1} - \mathbf{p}_{mid,2})^t / n_{par} < \chi_{seg}^2 \quad (2)$$

Pairs of segments passing this cuts are added to a list of segment matches.

After all segment matches have been found, they need to be validated to avoid track merging. For each segment appearing in the matches the list of all matching segments is build. This list is checked for multiple segments in the same module with overlapping row ranges. In case of overlaps the corresponding matches are invalidated. The set of remaining valid matches is used to define the equivalence classes.

3.4 Output

For each hit a TPC hit and for each track candidate a TPC track is constructed. The track parameters and the covariance matrix are transformed to the convention used by LCIO [6] (different order and signs). As reference point the position of the first hit is used. This ensures a proper value of the magnetic field. Alternatively, the point of closest approach (PCA) to the origin in the global frame can be used. For the test beam setup this could be outside of the magnet where a curvature parameter makes no sense anymore.

4 Performance test

This combined track and hit finder comprises the functionality of four standardly used MarlinTPC processors for hit finding, proper encoding of the module ID, calculation of

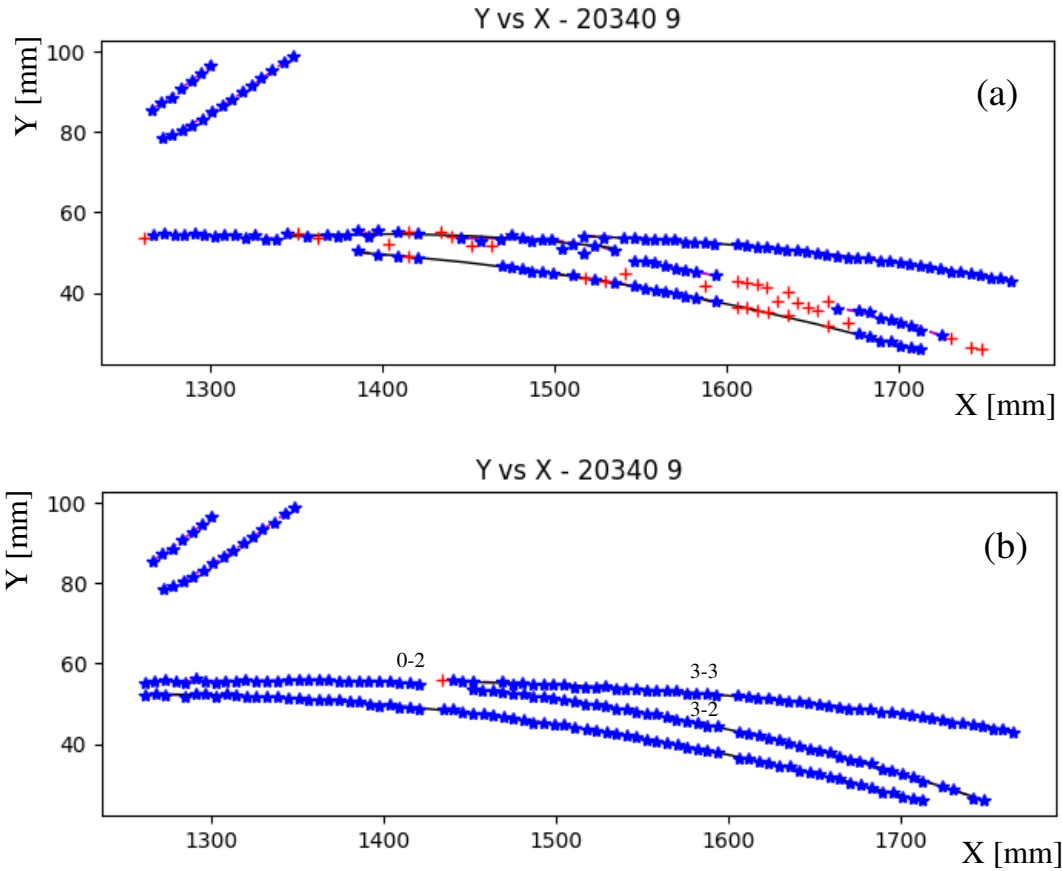


Figure 5: DESY GEM example event. Unused hits are marked by red crosses, hits on tracks by blue stars and (combined) track segments by black lines. The upper plot (a) shows the result of the standard hit finder and the triplet track finder [5] and the lower one (b) with this method. Especially in the first module ($X < 1400$) many hits are missing or distorted in the first case (a). For this method (b) the track segment '0-2' matches both with '3-2' and '3-3'. To avoid the merging of these two segments in a similar row range the corresponding matches are invalidated and ignored resulting in three shorter but proper tracks instead of one long merged.

hit (position) errors and track finding, e.g. by the triplet finder [5]. The performance of these two choices for the LPTPC with three DESY GEM modules has been compared briefly for sets of tracks without and with magnetic field B :

Run 19094: In average there are 747 pulses and 2.18 tracks (from the triplet finder) per event and B is zero.

Run 20380: In average there are 384 pulses and 2.13 tracks per event and $B = 1 T$.

In each case a scattering target in front of the TPC had been used to enhance multiple track events. Due to different transversal diffusion the number of pulses per track differs almost by a factor two. Table 1 compares the timing. This method is about a factor three to four faster.

MarlinTPC processor	run 19084, B=0 T	run 20380, B=1 T
ROWBASEDHITFINDER	7.18	4.91
TPCHIT_FIXCELLIDS	0.50	0.42
TPCHIT_FIXERRORS	0.38	0.31
ROWTRIPLETBASEDTRACKFINDER	0.83	0.70
This method, track and hit finding	2.61	1.45
This method, hit without track finding	1.32	0.70

Table 1: Time spent per event in milliseconds on a DESY workgroup server for hit and track finding in MarlinTPC with the standard processor chain or this method for DESY GEM data.

Figure 6 shows the average difference of the number of hits found on tracks by this method and the standard hit and track finding processor chain as function of the track multiplicity. For data without magnetic field this method finds about 5 additional hits per track. With magnetic field the difference is about a factor two smaller as the hits are narrower due to smaller transversal diffusion meaning less overlapping of hits and the tracks are bent apart. For single track events there is (almost) no difference.

5 Steering parameters

The steering parameters with the defaults indicated in parentheses and alternative suggestions for data without magnetic field in square brackets are:

InputTrackerPulses ("TPCPulses"): The name of the input collection of TPC pad pulses.

OutputTrackerHits ("TPCHits"): The name of the output collection with the found hits.

OutputTracks ("TPCTracks"): The name of the output collection with the found tracks.

SkipRoadSearch (false) Flag for skipping road search, to test hit finding without any tracking information.

BFieldScaleFactor (1.0): Scale factor for the magnetic field (map), allows to switch off the magnetic field (section 2.2).

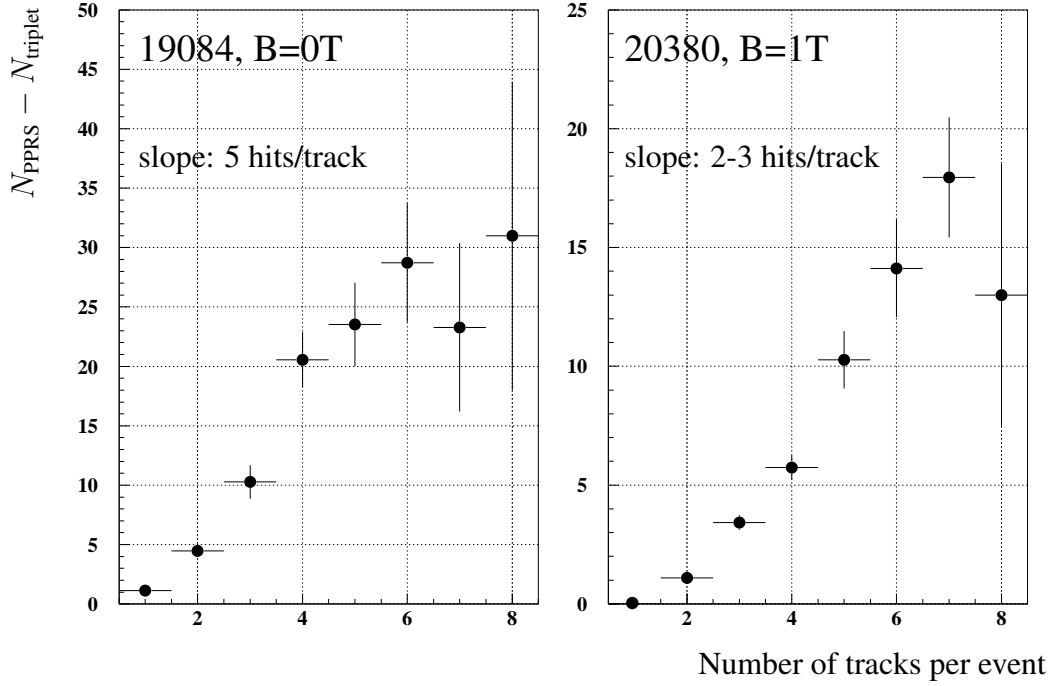


Figure 6: Difference of number of hits on track found by this method (PPRS) and the standard hit finder followed by the triplet finder as function of the track multiplicity for data without (left) and with (right) magnetic field.

DriftVelocity (74.): Drift velocity for drift time to distance conversion (section 3.1).

MaxColDiffNeighbourhood (1) [2]: Maximal column difference to pulse in neighbourhood (n in section 3.2.1).

MaxTimeDiff (80.): Maximum drift time difference [ns] in neighbourhood or hit (sections 3.2.1, 3.2.5).

MinRowDiff (5): Minimum row difference for pulse pair seeding initial road (section 3.2.2).

SkipMultiplePulses (true) Flag for skipping multiple pulse candidates as seeds for initial road (section 3.2.2).

MaxSeedsInRow (10): Maximum number of seeding pulses in row (section 3.2.2).

MaxDxyRoad (2.) [4.]: Maximum distance to road in XY (section 3.2.2).

MaxDzRoad (3.): Maximum distance to road in Z (section 3.2.2).

MinRowNum (6): Minimum number of rows for pad pulse segment (section 3.2.2).

MinRowDensity (0.8): Minimum row density for pad pulse segment (section 3.2.2).

MaxVarXY (0.25) [2.]: Maximum average of squared XY-residuals for pad pulse segment (section 3.2.2).

MaxOverlapFraction (0.5): Maximum overlap fraction of pad pulse segment (section 3.2.3).

MaxColDiffHit (0): Maximum distance of pad to seed in hit or zero (section 3.2.5).

MinNumPulses (2): Minimum number of pulses in hit (section 3.2.5).

MaxColGap (1): Maximum column gap in hit(section 3.2.5).

OffsetCol (0.0085): Hit position variance: offset σ_0^2 in column direction (sections 3.1, 3.2.5).

SlopeCol (0.00011) [0.000397]: Hit position variance: diffusion D_{eff}^2 in column direction (sections 3.1, 3.2.5).

OffsetDrift (0.25): Hit position variance: offset σ_0^2 in drift direction (sections 3.1, 3.2.5).

SlopeDrift (0.): Hit position variance: diffusion D_{eff}^2 in drift direction (sections 3.1, 3.2.5).

OffsetRow (0.): Hit position variance: offset σ_0^2 in row direction (sections 3.1, 3.2.5).

SlopeRow (0.): Hit position variance: diffusion D_{eff}^2 in row direction (sections 3.1, 3.2.5).

ChargeConversionFactor (1.): Charge conversion factor from ADC values to primary electrons (sections 3.1, 3.2.5).

SegmentMatchingChi2Cut (30.): χ^2 cut normalised to number of track parameters (5 for magnetic field on, 4 for off) for segment matching (χ_{seg}^2 in section 3.3).

ReferencePointAtPca (false) Flag for using the point of closest approach (PCA) as track reference point instead of position of first hit (section 3.4).

6 Summary

A simple and fast combined track and hit finding method for a TPC based on local road search with pad pulses has been presented. It relies on proper (calibrated) pad pulse data. For the LPTPC test beam data it demonstrates an improved hit separation capability down to a distance of few pads exploring the tracking information. An implementation in MarlinTPC is available as the `ROWBASEDPADPULSEROADSEARCHPROCESSOR`.

7 Outlook

Options for further improving the method are proper fits to the charge distribution in a row to resolve ambiguities or the usage of a database to access the list of known dead or noisy pads or parameters like the drift velocity.

References

- [1] P. Schade, J. Kaminski on behalf of the LCTPC collaboration, A large TPC prototype for a linear collider detector, Proceedings of the 12th International Vienna Conference on Instrumentation, *Nucl. Instr. and Methods A*, 628:128, 2011.
- [2] LPTPC collaboration, A time projection chamber with GEM-based readout, *Nucl. Instr. and Methods A*, 856:109, 2017.
- [3] MarlinTPC Homepage: <https://znwiki3.ifh.de/MarlinTPC/>
- [4] V. Karimäki, Effective circle fitting for particle trajectories, *Nucl. Instr. and Methods A*, 305:187, 1991.
- [5] C. Kleinwort, A track finding method for a TPC based on triplet chains, LC-TOOL-2014-004, <http://www-flc.desy.de/lcnotes>
- [6] T. Krämer, Track Parameters in LCIO, LC-DET-2006-004, <http://www-flc.desy.de/lcnotes>