LC-TOOL-2014-006

http://www-flc.desy.de/lcnotes

A track finding method for a TPC based on fast Hough transformation

Claus Kleinwort, DESY, Germany

April 28, 2014

Abstract

This document describes a global track finding algorithm using a fast Hough transformation [1] developed for a prototype TPC with pad readout and later extended for pixelated readout.

1 Introduction

A track finding method based on fast Hough transformation has been developed especially for the data taken by the Large Prototype TPC (LPTPC) [2] with pad readout at the DESY test beam [3] and implemented as a MarlinTPC [4] processor.

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.

With a Hough transformation the position measurements of the hits are represented by manifolds in a n-dimensional parameter space. This is usually subdivided in a large number of equidistant bins and the track finding corresponds to looking for the bins with maximal content. Following the approach of [1] the parameter space is mapped onto the unit hypercube and the measurements correspond to hyperplanes. The hypercube is split in each dimension into two child cubes and those containing some minimum content are split again recursively. Therefore a much smaller number of bins (cubes) has to be evaluated allowing to use the complete set of track parameters simultaneously. For hyperplanes it is especially simple to check the intersection with the child cubes.

In section 2 the basic concepts are presented, section 3 describes the sequence of processing steps and section 4 some extensions, section 5 show some performance tests and section 6 explains the steering parameters of the ROWBASEDFASTHOUGHTRANSFORMATIONPROCESSOR.

2 Concepts

2.1 Hypercubes

The n-dimensional range of accepted track parameters $\mathbf{p}$ is mapped onto the unit hypercube at the origin : $|p_i| \leq \frac{1}{2}, i = 1..n$. This root cube is divided in each dimension i into two child cubes with centers $c_i = \pm\frac{1}{4}$. For each of the 2^n child cubes the hits with intersecting transforms (hyperplanes) have to be determined. The cubes containing some minimal number of hits are split recursively until the content is compatible with a valid solution (single track). This requires each time to shift and scale the transforms as those cubes become a new unit hypercube with center zero in the recursive process.

2.2 Hyperplanes

Hyperplanes in the parameter space as transforms of the measurements of the hits are *linear* combination of the parameters and can be described by a unit normal vector $\mathbf{n}$ and the distance of closest approach η:

$$\eta = \mathbf{p} \cdot \mathbf{n} = \sum_i p_i \cdot n_i \quad \text{or} \quad d + \mathbf{p} \cdot \mathbf{n} = 0 \quad \text{with distance} \quad d = -\eta \tag{1}$$

The hyperplane intersects the unit hypercube at the origin if the maximal component of the point of closest approach ($\eta\,\mathbf{n}$) is inside:

$$\max_i(|\eta \cdot n_i|) \leq \frac{1}{2} \quad \text{or} \quad |d| \leq \frac{0.5}{\max_i(|n_i|)} \tag{2}$$

The distance d_i with respect to a child cube with center $\mathbf{c}_i$ is:

$$d_i = d + \mathbf{n} \cdot \mathbf{c}_i \tag{3}$$

With the child cube as new unit hypercube this will become the new $d = 2\,d_i$. The maximal component of the unit normal vector and the distance corrections ($\mathbf{n} \cdot \mathbf{c}_i$) for all possible child cubes are the same in each step of the recursive splitting process and need to be calculated only once and can be stored with the hyperplane.

2.3 Track parametrisation

As the tracks move in the LPTPC mainly in the x-direction a parametrisation $y(x)$ and $z(x)$ based each on a series of Legendre polynomials L_i is used. As those map the range $[-1, +1]$ onto itself and p_i is from $[-0.5, +0.5]$ each term $p_i \cdot L_i$ is in $[-0.5, +0.5]$ too. Therefore for all hits subject to this method a linear range adjustment has to be performed:

$$x \to u \in [-1, +1],\ y \to v \in [-0.5, +0.5],\ z \to w \in [-0.5, +0.5] \tag{4}$$

The (adjusted) pad plane measurement $v(u)$ is described by a series of orthogonal Legendre polynomials L_i up to the second order (first order in case of zero magnetic field) and the drift time measurement $w(u)$ by a series up to the first order:

$$v(u) = \sum_{i=0}^{i=2} p_{i+1} \cdot L_i(u),\quad w(u) = \sum_{i=0}^{i=1} p_{i+4} \cdot L_i(u) \tag{5}$$

The orthogonality of the L_i reduces the correlations between the parameters p_i. In case u would be uniformly distributed in $[-1, +1]$ they would be uncorrelated. The corresponding polynomials are:

$$L_0(u) = 1,\quad L_1(u) = u,\quad L_2(u) = \frac{3}{2}u^2 - \frac{1}{2} \tag{6}$$

From the parametrisations two hyperplanes with normals $\mathbf{l}_v$ and $\mathbf{l}_w$ can be constructed:

$$\mathbf{l}_v = (L_0(u), L_1(u), L_2(u), 0, 0),\quad \mathbf{l}_w = (0, 0, 0, L_0(u), L_1(u)) \tag{7}$$

The normalised unit normal vectors and distances are:

$$\mathbf{n}_v = \frac{\mathbf{l}_v}{|\mathbf{l}_v|},\ d_v = -\frac{v}{|\mathbf{l}_v|},\ \mathbf{n}_w = \frac{\mathbf{l}_w}{|\mathbf{l}_w|},\ d_w = -\frac{w}{|\mathbf{l}_w|} \tag{8}$$

In this case the largest component of the normal vector $\mathbf{l}$ is $L_0(u) = 1$ yielding:

$$\max_i(|n_i|) = \frac{1}{|\mathbf{l}|} \tag{9}$$

In summary the space point defined by a hit is represented by a pair of hyperplanes in disjunct subspaces of a five (or four) dimensional parameter space.

2.4 Radial ordering

The row number of a hit is used as radial ordering parameter to sort the hits. For a set of hits the number of rows measure a size and the difference of row numbers define a distance.

3 Sequence

The general strategy is to find the largest track, remove its hit from the process and to iterate until no more track is found.

3.1 Input

Input are the positions of all hits not yet assigned to a track ordered by row number. A constant magnetic field is assumed to describe the tracks by a helix or straight line.

3.2 Preprocessing

First the number of correlated rows is calculated. This are rows with hits for which a hit exists in some of the previous rows. This kind of correlation is expected from tracks, but not from random hits. If this number is too small the process stops.

From all hits the minimal and maximal value of the x-coordinate is calculated to dynamically scale x to $u \in [-1, +1]$. For the measurement directions y and z the steering defines some ranges for a static scaling to v and $w \in [-0.5, +0.5]$. The ratio of the sizes of these ranges should somehow reflect the ratio of the measurement resolutions to have similar resolution in v and w.

3.3 Construction of root hypercube

For each hit the pair of hyperplanes is constructed according to (8). The root hypercube is defined by all intersecting pairs of hyperplanes:

$$|d_v| \leq 0.5 \cdot |\mathbf{l}_v| \;\text{ and }\; |d_w| \leq 0.5 \cdot |\mathbf{l}_w| \tag{10}$$

Alternatively only the hyperplanes from one of the measurement parametrisations $v(u)$ and $w(u)$ can be used in a parameter space of dimension two or three.

3.4 Recursive splitting

The hypercube is being split recursively:

1. For all of the 2^n child cubes with centers $\mathbf{c}_i$ the intersection with all the hyperplane pairs is checked using an increased distance cut to allow for some overlap to avoid binning effects:

   $$2\,|d_v + \mathbf{n}_v \cdot \mathbf{c}_i| \leq 0.75 \cdot |\mathbf{l}_v| \;\text{ and }\; 2\,|d_w + \mathbf{n}_w \cdot \mathbf{c}_i| \leq 0.75 \cdot |\mathbf{l}_w| \tag{11}$$

   As the two hyperplanes of each pair are in disjunct subspaces (with dimensions $n-2$ and 2) instead of 2^n only $2^{n-2} + 2^2$ different cases have to be checked.

2. Only child cubes containing some minimum number of rows (relative to the total number of correlated rows) are considered further.

3. The remaining child cubes are sorted by decreasing size (number of rows) and increasing spread of the distances ($\langle d_v^2 \rangle + \langle d_w^2 \rangle$). The idea is to analyse the most promising child cubes first.

4. If a minimum splitting level (defining the two track resolution) has been reached the hit density is evaluated for the ordered child cubes. This is the number of hits normalised to the distance of first and last row. If this value is between the expected hit efficiency and purity the process stops and the list of hits in this child cube defines a track candidate.

5. Otherwise the child cube becomes a unit hypercube ($d \rightarrow 2 \cdot (d + \mathbf{n} \cdot \mathbf{c})$) and is split itself except if a maximum splitting level or maximum number of cubes has been reached. In this case the process stops without having found a track candidate.

If a candidate has been found the hits assigned to it are marked as used and the process restarts with the preprocessing (3.2).

3.5 Postprocessing

Similar to [6] a track segment is build from the list of hits of the candidate from the final child cube in the recursive splitting process. All hits in rows where the number of hits does not exceed some limit define the segment. The segment is fitted with a circle [5] (or a straight line in case of zero magnetic field) in the XY- and a straight line in the ZS-projection using in addition to the positions the directions and resolutions of the measurements. With those additional hits can be assigned based on a χ^2 cut as in [6].

3.6 Output

From the fitted parameters and the list of hits a TPC track is constructed for each candidate. At the first hit as reference point the parameters and covariance matrix are converted to the LCIO parametrisation [7].

4 Extensions

4.1 Pixelated readout

The major change in this method for pixelated readout planes like [8] is to allow more than one hit per row on a track. Nevertheless a decent cut in the number of hits per row provides some protection against delta rays. In addition the measurement directions and errors have to be calculated now in a cartesian instead of a polar (pad) geometry.

4.2 Full TPC

One idea to apply this method to a full TPC would be to split the events in azimuthal bins. For each bin the fast Hough transformation could be run in a local XY-coordinate system and the results would need to be combined later. Additionally this would need a study of the performance of the track parametrisation (equation (5)) for tracks with large curvature (e.g. curlers).

5 Performance

The performance of this method for the LPTPC has been compared with the triplet finder [6] for several datasets with pad based readout in a magnetic field of 1 Tesla:

Run 19103 DESY GEM modules with single tracks. The average track multiplicity is about 1.2.

Run 19121 DESY GEM modules with multiple tracks produced by a lead block in the beam line in front of the TPC. The average track multiplicity is about 2.0.

Run 18900 Micromegas modules for a case of low single hit efficiency.

The comparison includes the number of tracks per event, the number of hits (rows) per track and the timing. The triplet finder is typically about a factor two faster (table 1).

Track finder	number of parameters	run 19103	run 19121
Triplet chains	5	0.51	0.84
FHT, v(u) & w(u)	5	0.77	1.81
FHT, only v(u)	3	0.44	0.76
FHT, only w(u)	2	0.18	0.22

Table 1: Time spent per event in milliseconds on a DESY workgroup server for track finding in MarlinTPC with fast Hough transformation (FHT) or the triplet method for DESY GEM data with varying track multiplicities. For the FHT different sets of measurement parametrisations corresponding to different number of (track) parameters have been used.

The figures 1 and 2 compare DESY GEM data with different track multiplicities. In both cases this methods finds slightly less but larger tracks with more hits. Probably fewer tracks are split into several pieces. A case of a low single hit efficiency of about 60% in shown in figure 3 for Micromegas data. The triplet finder returns mainly several short track pieces with less than 20 hits while this method finds usually more complete tracks with around 40 hits in agreement with the expectation for this single hit efficiency. In figure 4 the number of hypercubes built and inspected to find a track are displayed for different datasets. In most cases this could be done with the minimum number and the average is between 20 and 40.

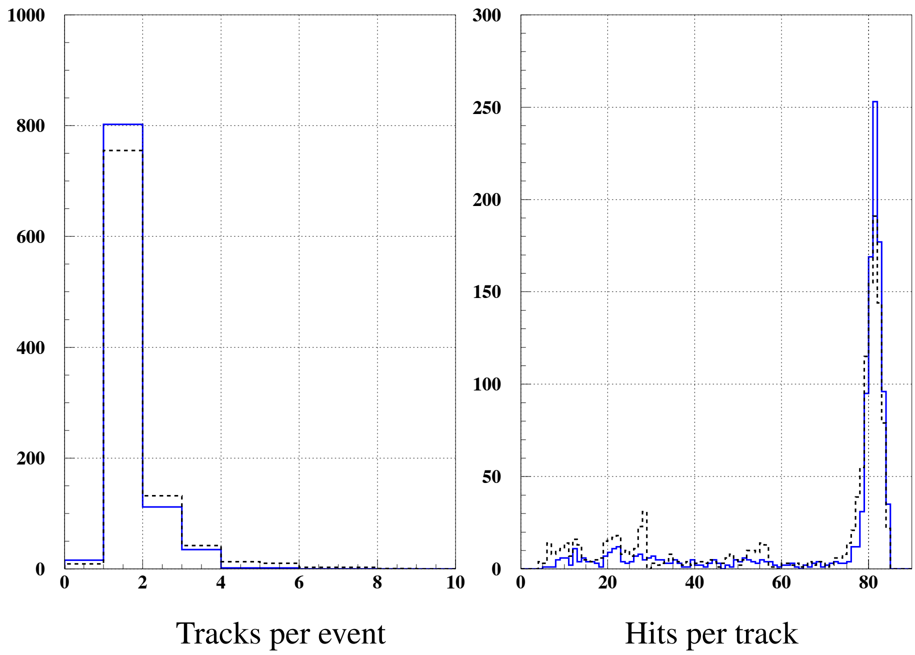

Figure 1: Number of tracks per event (left) and number of hits per track (right) for triplet (dashed) and this (solid) track finder for LPTPC run 19103. The maximum possible number of hits is 84.

Finally the method has been tried for data with pixelated readout [8]. In the 1024 rows typically tracks with 500-700 hits and a radial track length of close to 1000 rows are found as illustrated in figure 5.

6 Steering parameters

The steering parameters with the defaults indicated in parentheses are:

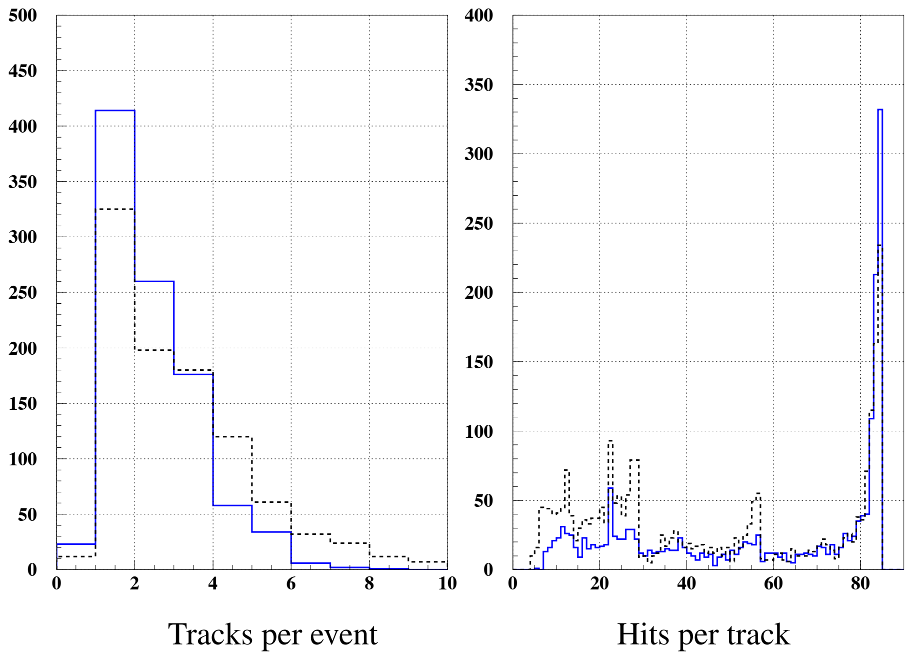

Figure 2: Number of tracks per event (left) and number of hits per track (right) for triplet (dashed) and this (solid) track finder for LPTPC run 19121. The maximum possible number of hits is 84.

InputHits ("TPCHits"): The name of the input collection of TPC hits .

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

BFieldScaleFactor (1.0): Scale factor for the magnetic field (map), allows to switch off the magnetic field (section 3.5).

CenterXYMeasurement (60.): Center of XY (anode plane) measurements (section 3.2).

RangeXYMeasurement (200.): Range of XY (anode plane) measurements (section 3.2).

CenterXZMeasurement (300.): Center of XZ (drift time) measurements (section 3.2).

RangeXZMeasurement (600.): Range of XZ (drift time) measurements (section 3.2).

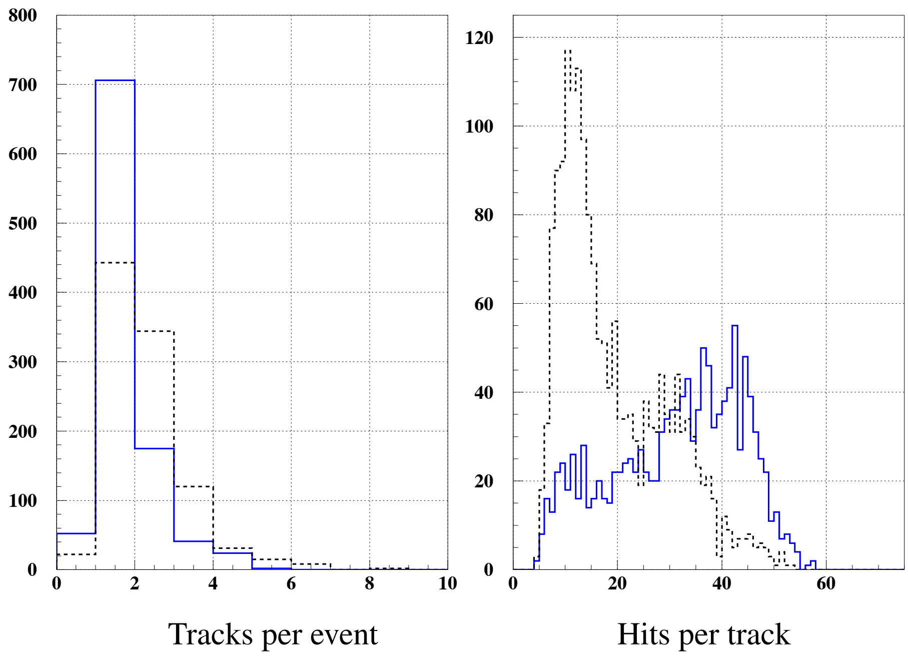

Figure 3: Number of tracks per event (left) and number of hits per track (right) for triplet (dashed) and this (solid) track finder for LPTPC run 18900. The maximum possible number of hits is 72. The peak around 40–45 corresponds to a single hit efficiency of about 60%

UseXYMeasurement (true): Flag for using XY (anode plane) measurements for hyperplanes (section 3.3).

UseXZMeasurement (true): Flag for using XZ (drift time) measurements for hyperplanes (section 3.3).

MinimumRows (8): Minimal number of (correlated) rows (section 3.2).

MaximumRowDifference (1): Row correlation parameter (maximal distance of correlated rows) (section 3.2).

FractionOfRows (0.8): Relative minimum cube content (ratio of number of rows in cube to total number of correlated rows) (section 3.4 item 2).

MaximumCubes (250): Maximum number of (hyper) cubes to evaluate (section 3.4 item 5).

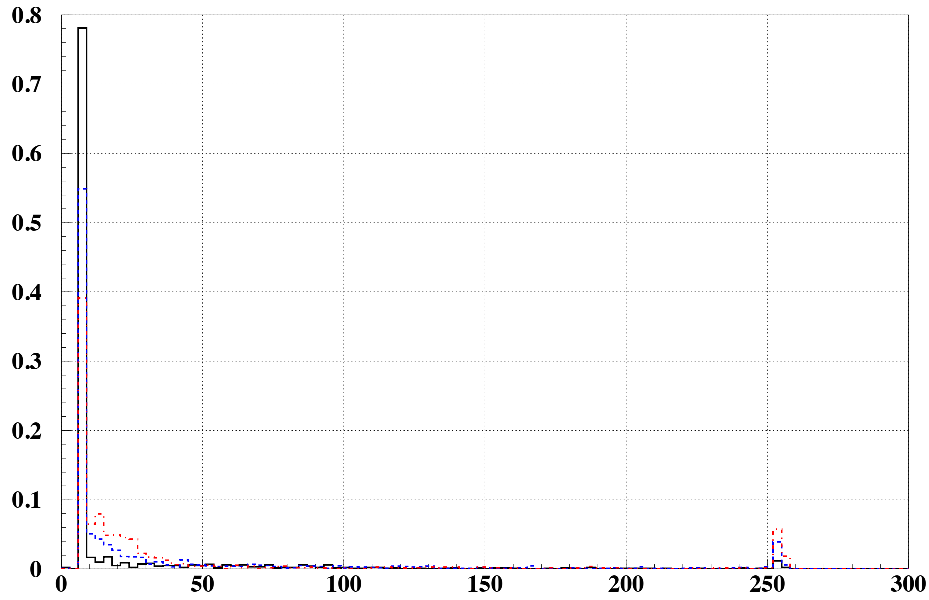

Figure 4: Number of hypercubes analysed (per track) for runs 19103 (solid), 19121 (dashed) and 18900 (dash-dotted). The peak at 6 corresponds to the minimal value and the number of possible cubes is $(2^5)^5 \approx 3 \cdot 10^7$ (for a minimum splitting level of 5 and five parameters). For values above 250 the method has stopped without finding a track.

MinimumLevel (5): Minimum number of (hyper) cubes splittings (section 3.4 item 4).

MaximumLevel (8): Maximum number of (hyper) cubes splittings (section 3.4 item 5).

EfficienyCut (0.8): Minimum hit density (hits/track-length) (section 3.4 item 4).

PurityCut (1.1): Maximum hit density (hits/track-length) (section 3.4 item 4).

MaxHitsPerRow (1): Maximum number of hits per row (section 3.5).

UnusedHitMatchingChi2Cut (20.): χ^2 cut for matching unused hits in XY and Z (section 3.5).

UnusedHitMaxGapCut (4): Maximal (row) gap for matching unused hits in XY and Z (section 3.5).

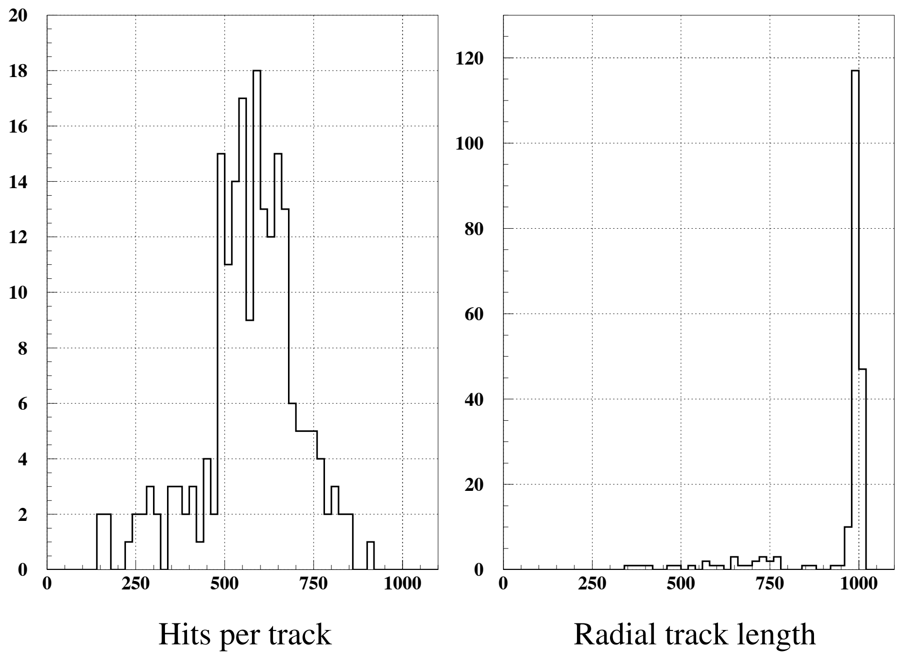

Figure 5: Number of hits per track (left) and radial track length (right) for GridPix [8] data. The maximum possible number of rows is 1024.

EncodedModuleID (true): Flag for encoding of module ID in CellID0 (section 3.1).

ReferencePointAtPca (false) Flag for using the point of closest approach (PCA) as track reference point instead of position of first hit (section 3.6).

7 Summary

A simple and fast global track finding method based on a fast Hough transformation for curved or straight tracks capable of using the complete set of parameters has been presented. Adjusting the steering parameters especially to the measurement ranges and the single hit efficiency it can be applied to various types of data (in terms of quality or readout technology). In comparison to the triplet finder [6] it finds slightly fewer but longer tracks and is about a factor two slower. An implementation in MarlinTPC is available as the ROWBASEDFASTHOUGHTRANSFORMATIONPROCESSOR.

References

[1] H. Li, M. A. Lavin, and R. J. Le Master, Fast Hough Transform: A Hierarchical Approach, *Computer Vision, Graphics, and Image Processing* **36**, 139–161 (1986)

[2] 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.

[3] A. Münnich, Performance of DESY GEM Module in Testbeam Measurements, Talk given at LCWS 2013, Tokyo, November 2013, http://www.icepp.s.u-tokyo.ac.jp/lcws13

[4] MarlinTPC Homepage: https://znwiki3.ifh.de/MarlinTPC/

[5] V. Karimäki, Effective circle fitting for particle trajectories, *Nucl. Instr. and Methods A*, 305:187, 1991.

[6] C. Kleinwort, A track finding method for a TPC based on triplet chains, LC-TOOL-2014-004, http://www-flc.desy.de/lcnotes

[7] T. Krämer, Track Parameters in LCIO, LC-DET-2006-004, http://www-flc.desy.de/lcnotes

[8] J. Kaminski, Development of a TPC readout based on GridPix, Talk given at the 7. Detector Workshop of the Helmholtz Alliance, Göttingen, March 2014, http://indico.desy.de/conferenceDisplay.py?confId=9389