Algorithms for Online Tracking in PANDA

CHEP 2018

10.7.2018  TOBIAS STOCKMANNS
Boundary Conditions

- Two magnets
- 4 tracking systems

- No time structure in beam
- Up to 20 MHz interaction rate
- Online tracking and event selection

- < 10 tracks / event
- Mainly p, π, K, e, µ
Simulated Events

MVD
- 4 barrel layers
- 6 disks
- Mixed pixels / strips
- 3D space points
- \(< 30 \mu m\) point resolution
- \(< 10 \text{ ns}\) time resolution

STT
- \(~ 4000\) straws
- Dense packaging
- \(< 150 \mu m\) resolution
- Isochrones
- 250 ns drift time
- 2D space point
- No start time

Time series of MVD and STT detector signals in PANDA (view along beam axis)
**Triplet Finder**

- Find 2 out of 3 combinations of hits in pivot layers and primary vertex
- Calculate circle
- Add hits in other layers by distance to circle
- Straight line fit for $z$-component
- Select track candidates by number of hits

+ Simple algorithm
+ No fitting needed
- No isochrone information
- Limited on primary tracks

- Optimized for GPUs together with Nvidia Application Lab

Marius Mertens(FZJ), Andrew Adinetz(FZJ), Andreas Herten(FZJ)
• Non-bunched: packages of single events send
• Bunched: Fixed time bunch with many events
- Find (unambiguous) neighbours with hits → generates tracklets
- Find connections between tracklets
- Fit connected tracklets and take best matching one (Circle fit via Riemann Surface)
- Add the ambiguous hits
- Correct for isochrones
- Straight line fit for z-component

+ All hits processed at once
+ Primary and secondary tracks
+ General clustering algorithm
  - Isochrone information in an additional stage
  - Only suitable for STT

- Optimized for GPU
Cell Track Finder

Processing time per event

incl. data transfer

Process neighbourhood relations
(EvaluateState & EvaluateMultiStates)

<table>
<thead>
<tr>
<th>Version</th>
<th>Runtime per event</th>
<th>Ratio of CPU Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>4.56 ms</td>
<td>100 %</td>
</tr>
<tr>
<td>GPU (algorithm)</td>
<td>0.575 ms</td>
<td>12.08 %</td>
</tr>
<tr>
<td>GPU (+ event)</td>
<td>0.045 ms</td>
<td>0.94 %</td>
</tr>
</tbody>
</table>


t [ms]
Circle Hough

- Calculate for all hits all circles through origin which are tangent to isochrone
- Fill x, y coordinate of circle center into histogram
- Find peaks
- Straight line fit for z-component

+ All hits in parallel
+ Isochrone information naturally in
+ Works for STT and MVD

- Limited on primary tracks

Andreas Herten (FZJ),
Ludovico Bianchi (FZJ)
Pair Circle Hough
Circle Hough
Track Finding on FPGA

- Road Finding

- Circle Fitting

\[ E^2 = \sum (x_i^2 + y_i^2 + a x_i + b y_i + c)^2 \left(1/d_i\right)^2 \]

- Straight Line Fitting

\[ E^2 = \sum (\Phi_i + k z_i + \Phi_0)^2 \left(1/d_i\right)^2 \]

- Hits processed sequentially
- Isochrone information used
- T0 extraction possible
- Only suitable for STT
- Optimized for FPGA

Yutie Liang (JLU Giessen)
Performance

- Efficiency > 90 %
- $\sigma_{pt} : \sim 3.2\%$  $\sigma_{pz} : \sim 4.2\%$ .
- 7 $\mu$s/event (6 tracks)
- Tracking strategy in case of missing T0.
Summary

• Tracking in PANDA challenging
• Several different algorithms exist in different states of maturity
• No best candidate obvious
• Common test environment finished

• Next steps:
  • Finalize the algorithms
  • Test all on common data sample
  • Chain or combine algorithms
  • Machine Learning as alternative?
Circle Hough

Coverage

Mean = 0.95
Stddev = 0.11

Relative $p_t$ difference

Mean = -1.04
$\sigma = 3.36$
T0 determination

T0 shift = \( \Sigma di / N \) / \( \text{const} \)
(N: number of hits di: signed distance of circle to track)

T0 shifted by:  
-50 ns: \(-47.0\pm4.0 \) ns  
-20 ns: \(-19.8\pm2.1 \) ns  
10 ns: \(9.4\pm2.5 \) ns  
20 ns: \(19.0\pm2.3 \) ns

Extracted T0:
-47.0\pm4.0 \) ns
-19.8\pm2.1 \) ns
9.4\pm2.5 \) ns
19.0\pm2.3 \) ns
Tracking Algorithm -- Road Finding

Hit:  Seg_ID (3 bits) + LayerID (5 bits) + Tube_ID (6 bits) + Arrival time

1: Start from inner layer
2: Attach neighbour hit to tracklet layer by layer

✓ Boundary between two segments.
✓ Number of neighbor:
  4 in axial layer; 6 in stereo layer
Known: \( x_i, y_i, d_i \)

Question: To determine a circle,

\[ x^2 + y^2 + ax + by + c = 0 \]

Method: Minimize the equation

\[ E^2 = \sum (x_i^2 + y_i^2 + a x_i + b y_i + c)^2 (1/d_i)^2 \]

1) Circle para.

\[
\begin{pmatrix}
S_{xx} & S_{xy} & S_x \\
S_{xy} & S_{yy} & S_y \\
S_x & S_y & N
\end{pmatrix}
\begin{pmatrix}
a \\
b \\
c
\end{pmatrix}
= \begin{pmatrix}
-S_{xxx} - S_{xyy} \\
-S_{xxy} - S_{yyy} \\
-S_{xx} - S_{yy}
\end{pmatrix}
\]

\( S_x = \sum x_i \)
\( S_{xx} = \sum x_i x_i \)
\( S_{xxx} = \sum x_i x_i x_i \)

2) Track quality.

\[
\chi^2 = \frac{1}{n} \sum \left( \frac{x_i^2 + ax_i + y_i^2 + by_i}{d_i^2} \right)
\]

\[ = \frac{1}{n} \sum \left( \frac{x_i^2}{d_i^2} + \frac{ax_i}{d_i^2} + \frac{y_i^2}{d_i^2} + \frac{by_i}{d_i^2} \right) \]
Known: \( z_i, \Phi_i, d_i \)

Question: To determine a line,

\[ \Phi + kz + \Phi_0 = 0 \]

Method: Minimize \( E^2 = \sum (\Phi_i + kz_i + \Phi_0)^2 (1/d_i)^2 \)

\[
\begin{pmatrix}
S_{zz} & S_z \\
S_z & 1
\end{pmatrix}
\begin{pmatrix}
k \\
\phi_0
\end{pmatrix}
= \begin{pmatrix}
-S_{\phi z} \\
-S_{\phi}
\end{pmatrix}
\]
Extract $T_0$ from Tracking

$T_0$ shift = $\Sigma di / N / \text{const}$

$N$: number of hits
$di$: signed distance of circle to track

$T_0$ shifted by: 
-50 ns: 
-20 ns: 
10 ns: 
20 ns: 

Extracted $T_0$: 

-47.0±4.0 ns
-19.8±2.1 ns
9.4±2.5 ns
19.0±2.3 ns
Performance at FPGA

For one event with 100 hits (6 tracks): 7 µs

<table>
<thead>
<tr>
<th>Logic Utilization</th>
<th>Used</th>
<th>Available</th>
<th>Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Slice Flip Flops</td>
<td>25,022</td>
<td>50,560</td>
<td>49%</td>
</tr>
<tr>
<td>DCM autocalibration logic</td>
<td>14</td>
<td>25,022</td>
<td>1%</td>
</tr>
<tr>
<td>Number of 4 input LUTs</td>
<td>33,120</td>
<td>50,560</td>
<td>65%</td>
</tr>
<tr>
<td>DCM autocalibration logic</td>
<td>8</td>
<td>33,120</td>
<td>1%</td>
</tr>
<tr>
<td>Number of occupied Slices</td>
<td>21,563</td>
<td>25,280</td>
<td>85%</td>
</tr>
<tr>
<td>Number of FIFO16/RAMB16s</td>
<td>148</td>
<td>232</td>
<td>63%</td>
</tr>
<tr>
<td>Number used as RAMB16s</td>
<td>148</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of DSP48s</td>
<td>124</td>
<td>128</td>
<td>96%</td>
</tr>
</tbody>
</table>

Hit reading (100 cc)  
Road finder (50 cc X 6)  
Pt extraction (120 cc X 2 X 6)  
Pz extraction (60 cc X 2 X 6)
Performance test
Summary and Outlook

- $\sigma_{pt} : \sim 3.2\% \quad \sigma_{pz} : \sim 4.2\%$.
- $7 \mu s/\text{event (6 tracks)}$
- Tracking strategy in case of W/O T0.

Sitcp: Ready