U2 Downstream tracking studies

Arthur Hennequin – arthur.hennequin@cern.ch

Objectives

Evaluate what impact a pixel UT would have on downstream tracking:

- What pixel size ?
- How many layers ?
- Layer placement.
- Ghost rates ? Efficiencies ?
- Data structures ?

Setup and Sample

Stack starting point (initiated by Tim Evans):

- Gaudi: v38r0
- Detector: run5-tmp
- LHCb: run5
- Rec: run5
- Moore: run5
- Lbcom: master

Sample:

- Boole-test-FTDRCutOff-Extended.digi - minbias, produced by Mark Whitehead and Renato Quagliani

MT seeding

In run3, seeds are comming only from the SciFi. They are reconstructed using the HybridSeeding algorithm.

In run5, the seeds will come from different combination of pixels and fibres detectors in the MT and will be reconstructed using 3 algorithms:

- Pix-Pix (contributes $\sim$150 downstream tracks / event)
- Fib-Fib (contributes $\sim$170 downstream tracks / event)
- Pix-Fib (contributes $\sim$50 downstream tracks / event)

The seeds will then be merged into a single track container and given as input to the downstream tracking.

Fake MT seeding

The algorithms for MT reconstruction are currently under development. It is therefore interesting to have a "cheated" version that uses MC informations to create seed tracks.

The FakeTrackingMT creates tracks from MCParticles if:

$$\#_{hits} \geq 4 \vee (2 \times \#_{pixels} + \#_{fibres}) > 10$$

The created tracks can contain either or both pixels and fibres hits.

MT Seeding (cheated)

```
**** MTSeed                      7532 tracks including        0 ghosts [ 0.00 %], Event average  0.00 % ****
  01_hasT                     :   3733 from    4251 [ 87.81 %]     1 clones [ 0.03 %], purity: 99.99 %, hitEff: 99.98 %
  02_long                     :   2672 from    4265 [ 62.65 %]     0 clones [ 0.00 %], purity:100.00 %, hitEff:100.00 %
  03_long_P>5GeV              :   1619 from    2133 [ 75.90 %]     0 clones [ 0.00 %], purity:100.00 %, hitEff:100.00 %
  04_long_fromB               :      5 from       5 [100.00 %]     0 clones [ 0.00 %], purity:100.00 %, hitEff:100.00 %
  05_long_fromB_P>5GeV        :      3 from       3 [100.00 %]     0 clones [ 0.00 %], purity:100.00 %, hitEff:100.00 %
  06_UT+T_strange             :    341 from     561 [ 60.78 %]     0 clones [ 0.00 %], purity:100.00 %, hitEff:100.00 %
  07_UT+T_strange_P>5GeV      :    133 from     152 [ 87.50 %]     0 clones [ 0.00 %], purity:100.00 %, hitEff:100.00 %
  08_noVelo+UT+T_strange      :    193 from     309 [ 62.46 %]     0 clones [ 0.00 %], purity:100.00 %, hitEff:100.00 %
  09_noVelo+UT+T_strange_P>5GeV :   82 from      95 [ 86.32 %]     0 clones [ 0.00 %], purity:100.00 %, hitEff:100.00 %
  10_UT+T_SfromDB             :     10 from      18 [ 55.56 %]     0 clones [ 0.00 %], purity:100.00 %, hitEff:100.00 %
  11_UT+T_SfromDB_P>5GeV      :      1 from       1 [100.00 %]     0 clones [ 0.00 %], purity:100.00 %, hitEff:100.00 %
  12_noVelo+UT+T_SfromDB_P>5GeV :    1 from       1 [100.00 %]     0 clones [ 0.00 %], purity:100.00 %, hitEff:100.00 %
  13_hasT_electrons           :   1450 from    2337 [ 62.05 %]     7 clones [ 0.48 %], purity: 99.94 %, hitEff: 99.92 %
  14_long_electrons           :     82 from     374 [ 21.93 %]     0 clones [ 0.00 %], purity:100.00 %, hitEff:100.00 %
  15_long_fromB_electrons     :      2 from       4 [ 50.00 %]     0 clones [ 0.00 %], purity:100.00 %, hitEff:100.00 %
  16_long_electrons_P>5GeV    :     30 from      36 [ 83.33 %]     0 clones [ 0.00 %], purity:100.00 %, hitEff:100.00 %
  17_long_fromB_electrons_P>5GeV :    2 from       2 [100.00 %]     0 clones [ 0.00 %], purity:100.00 %, hitEff:100.00 %
```

The reconstructible definition used in this checker is $(\#_{MThits} \geq 4)$

Fake Downstream Tracking

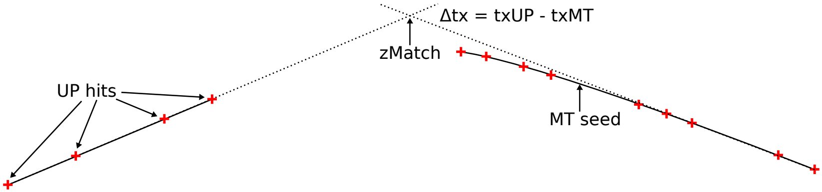
Similarly, a fake downstream tracking can be defined. The goal is to provide a base for the developement of the algorithm, and allow to measure the impact of design choices individually.

Downstream tracking (cheated)

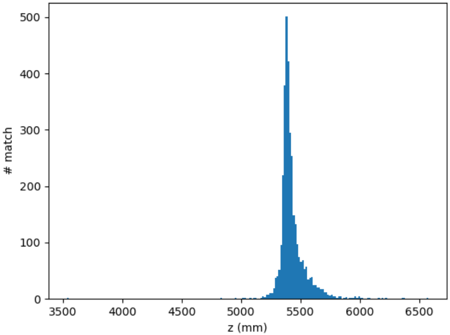
```
**** Downstream                     3391 tracks including              0 ghosts [ 0.00 %], Event average  0.00 % ****
  01_UT+T                       :   2877 from     3005 [ 95.74 %]      1 clones [ 0.03 %], purity:100.00 %, hitEff: 99.98 %
  02_UT+T_P>5GeV                :   1604 from     1704 [ 94.13 %]      0 clones [ 0.00 %], purity:100.00 %, hitEff:100.00 %
  03_UT+T_strange               :    338 from      352 [ 96.02 %]      0 clones [ 0.00 %], purity:100.00 %, hitEff:100.00 %
  04_UT+T_strange_P>5GeV        :    132 from      143 [ 92.31 %]      0 clones [ 0.00 %], purity:100.00 %, hitEff:100.00 %
  05_noVelo+UT+T_strange        :    191 from      199 [ 95.98 %]      0 clones [ 0.00 %], purity:100.00 %, hitEff:100.00 %
  06_noVelo+UT+T_strange_P>5GeV :     81 from       88 [ 92.05 %]      0 clones [ 0.00 %], purity:100.00 %, hitEff:100.00 %
  07_UT+T_fromDB                :     37 from       38 [ 97.37 %]      0 clones [ 0.00 %], purity:100.00 %, hitEff:100.00 %
  08_UT+T_fromBD_P>5GeV         :     18 from       18 [100.00 %]      0 clones [ 0.00 %], purity:100.00 %, hitEff:100.00 %
  09_noVelo+UT+T_fromBD         :      7 from        7 [100.00 %]      0 clones [ 0.00 %], purity:100.00 %, hitEff:100.00 %
  10_noVelo+UT+T_fromBD_P>5GeV  :      1 from        1 [100.00 %]      0 clones [ 0.00 %], purity:100.00 %, hitEff:100.00 %
  11_UT+T_SfromDB               :      9 from        9 [100.00 %]      0 clones [ 0.00 %], purity:100.00 %, hitEff:100.00 %
  12_UT+T_SfromDB_P>5GeV        :      1 from        1 [100.00 %]      0 clones [ 0.00 %], purity:100.00 %, hitEff:100.00 %
  13_noVelo+UT+T_SfromDB        :      6 from        6 [100.00 %]      0 clones [ 0.00 %], purity:100.00 %, hitEff:100.00 %
  14_noVelo+UT+T_SfromDB_P>5GeV :      1 from        1 [100.00 %]      0 clones [ 0.00 %], purity:100.00 %, hitEff:100.00 %
```

The reconstructible definition used in this checker is $(\#_{UPhits} \geq 3 \wedge \#_{MThits} \geq 4)$

Downstream tracking

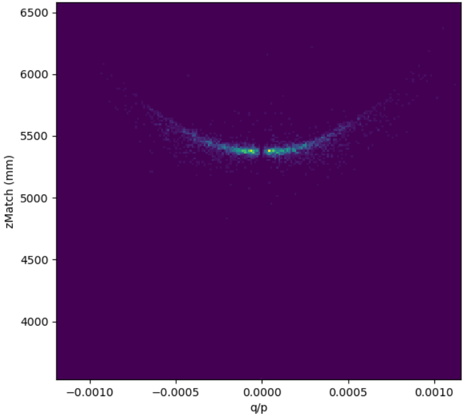


Kink position (zMatch)



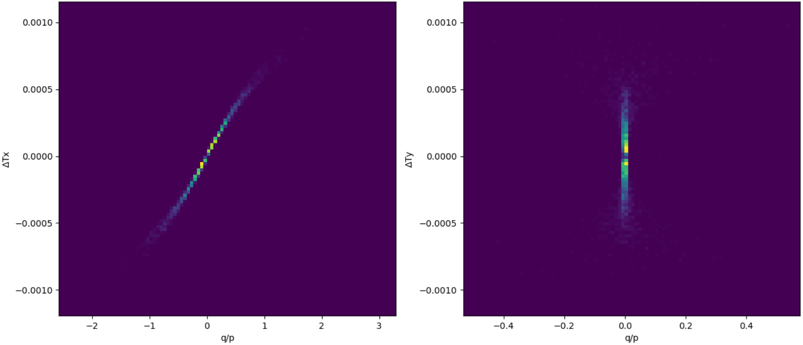
Peak at z = 5380 mm

zMatch vs q/p



6500
6000
5500
5000
4500
4000
zMatch (mm)
−0.0010
−0.0005
0.0000
0.0005
0.0010
q/p

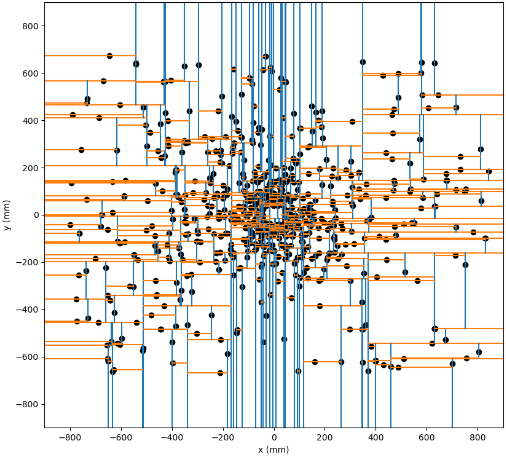
True Δtx Δty vs q/p



Hits data structure

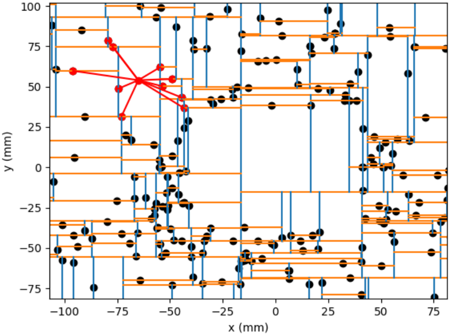
- In run3, UT hits are stored in a grid of sorted strips (in x)
- In run5, UP hits will be pixels, and fast lookup will be very important
- Simulation estimate $O(800)$ pixel hits per layer in UP
- Hit density is not uniform (concentrated around the beamline) $\Rightarrow$ need for an adaptive data structure

KDTree



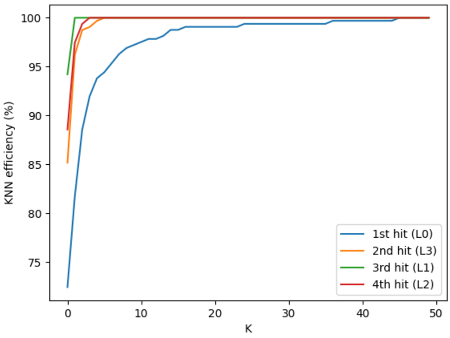
- Space partitioning data structure for organizing points in a k-dimensional space. (here 2D)
- For N points:
  - static creation: $O(N \times log(N))$
  - addition / removal of a point: $O(log(N))$
  - K nearest neighbors: $O(K \times log(N))$
- $O(800)$ hits per layer in UP

KNN



Example KNN query with K=10, zoomed on center region

KNN efficiency



- Find 1st hit using true zMatch and assuming origin at (0,0,0)
- Find 2nd hit using 1st hit and zMatch as straight line constraints
- Find 3rd and 4th hit using 1st and 2nd hit as straight line constraints

Conclusion

Done:

- Fake algorithms to produce MT seeds and downstream tracks
- MC checking sequence to measure efficiencies

Future work:

- Fit the MT seeds and compute a state from the hits informations
- Tune search windows in each layers
- Measure efficiencies