

Pattern Recognition FTS @Cracow University of Technology work in progress status report

Jerzy Jaworowski, Krzysztof Korcyl, Mateusz Michałek, Joanna Płażek,
Piotr Poznański

Four methods

- evaluation of methods:
 - 1)Pattern Matching
 - 2)Hough Histogram
 - 3)Circle-Line-Tangent Filter
 - 4)Circle-Tangent Region
- Final solution may consist of a combination of those

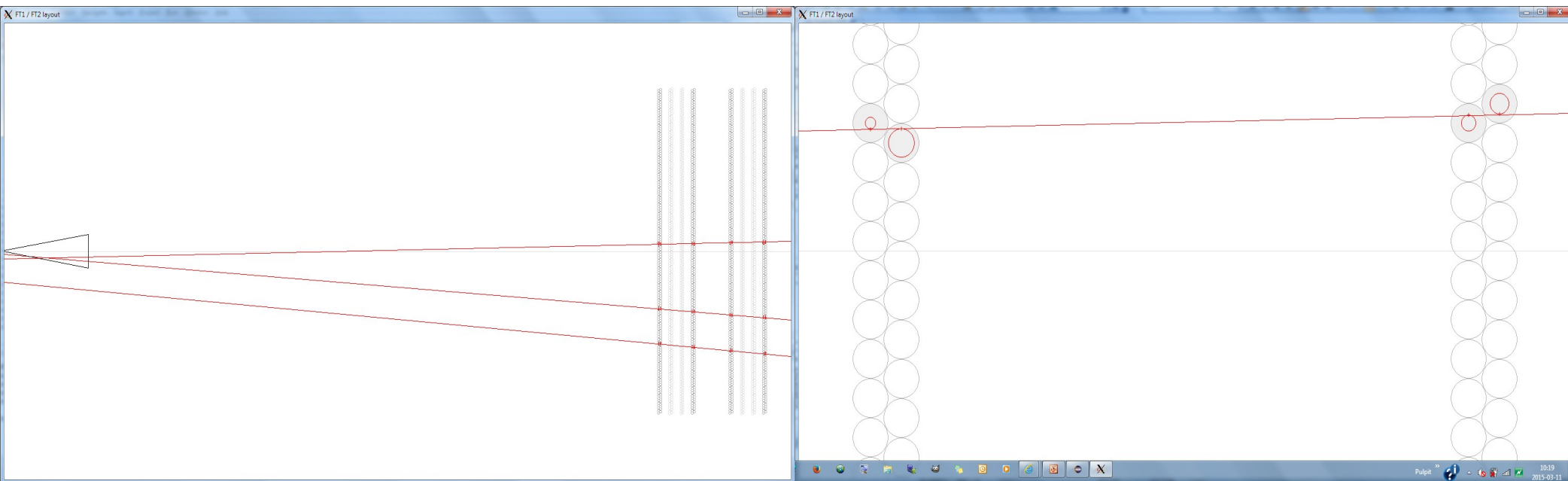
Pattern Matching Method

Segments and Data Organisation

- **FT 1, 2 layers : 1, 2, 7, 8 & 9, 10, 14, 15 (2D)**
- **Pair (Key, Data)** - stored in hash table
($\log_2 n$ search time)
- **Key:** 16 segments (16 bytes - 2 bytes per layer)
- 1 segment – relative tube index on layer (0-131)
- **Data:** double[2] - a,b (slope-intercept from of linear equation, $x=a*z+b$)

Pattern Generation

- precision **0.01 mm** step
- for **FT1** segment – **19102** patterns (will reduce < **16000**)



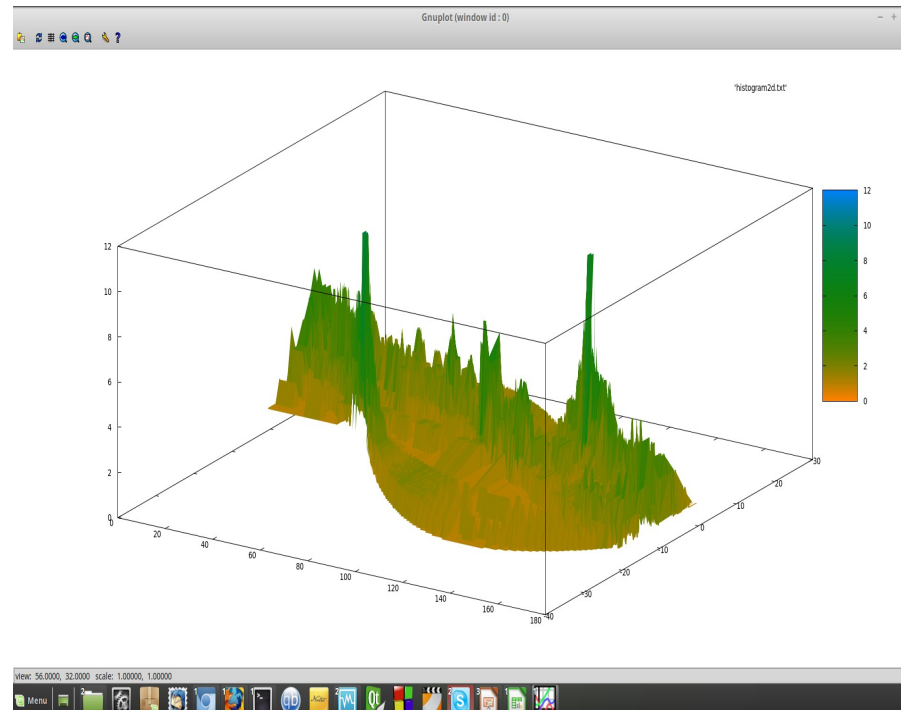
Pattern Matching

- 1.If exact pattern found – stop.
- 2.If not, return first pattern with key not less than searched one ('close' pattern).
- 3.Evaluate mean-square distance between event and close pattern (two different distance measure algorithm's tested : Levenshtein and custom one).
- 4.Repeat step 3 for enclosing keys to determine local minimum

Hough Histogram Method

Segments and Histo Generation

- Implemented for **FT 1, 2, 5, 6 (2D)**
- using **drift circles**
- lexical-distance based maxima search



Segments and Histo Generation

- TBDaTY:
 - **FT3 and FT4** (conformal transformation vs 3D Histogram)
 - **3D** (3-4D Histogram?)
 - check efficiency using **straw centers** instead of drift circles

Circle-Line-Tangent Filter Method

Method Description

- Method is **global FT1 – FT6 (2D)**
- Steps:
 1. Choose (any combination of) 3 points from FT3, FT4
 2. Create circumcircle over choosen points
 3. Use filters to verify created circle
- FtsPointZ and FtsPointX used as input by now
- aiming at: using straws centers

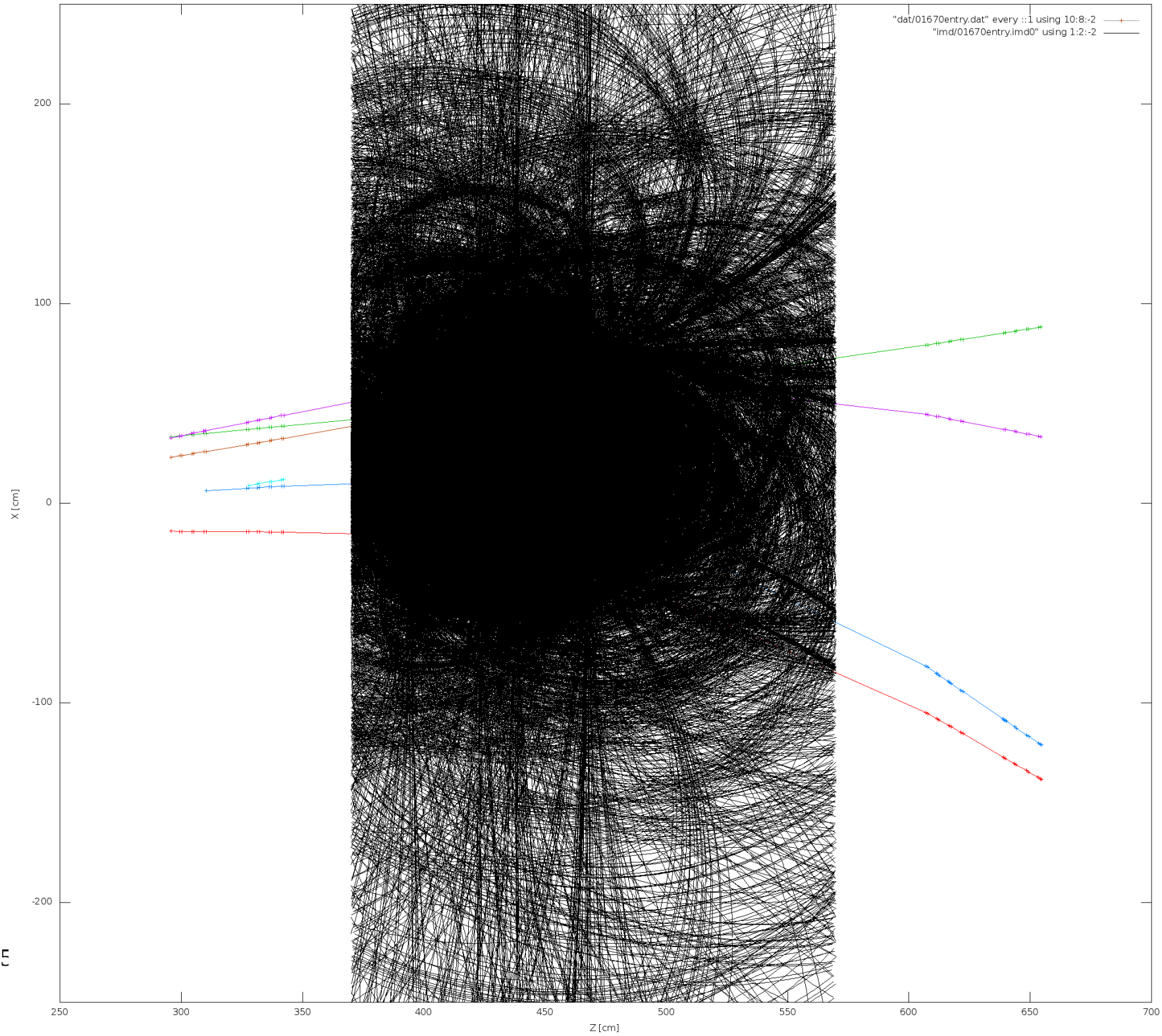
Filters

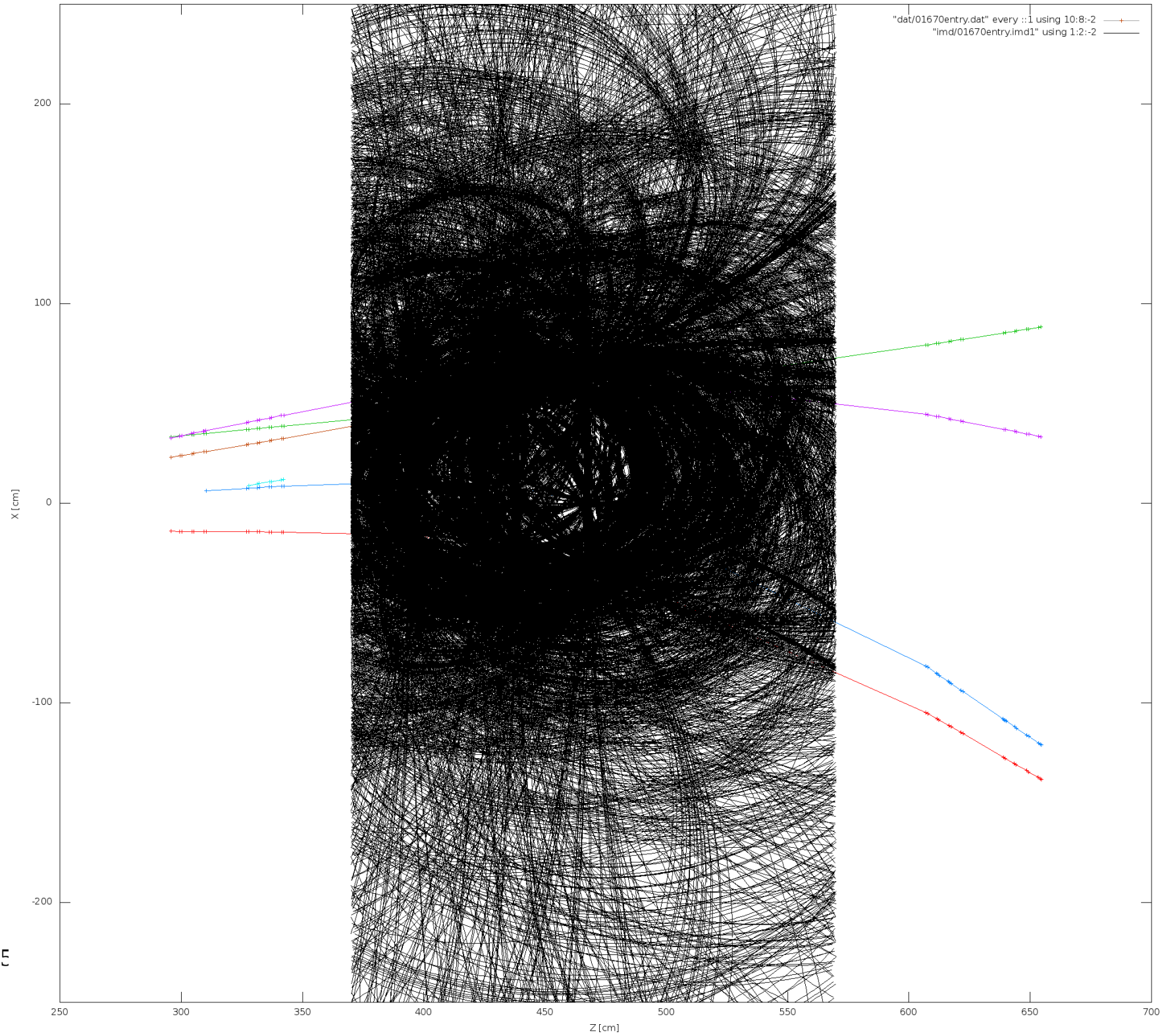
- 1.radius within range $<50,5000>$ cm
- 2.center far away from beam pipe
- 3.circle passes through 6 hits
- 4.circle has tangent in FT5, FT6 which passes through 6 hits
- 5.circle has tangent in FT1, FT2 which passes through 6 hits

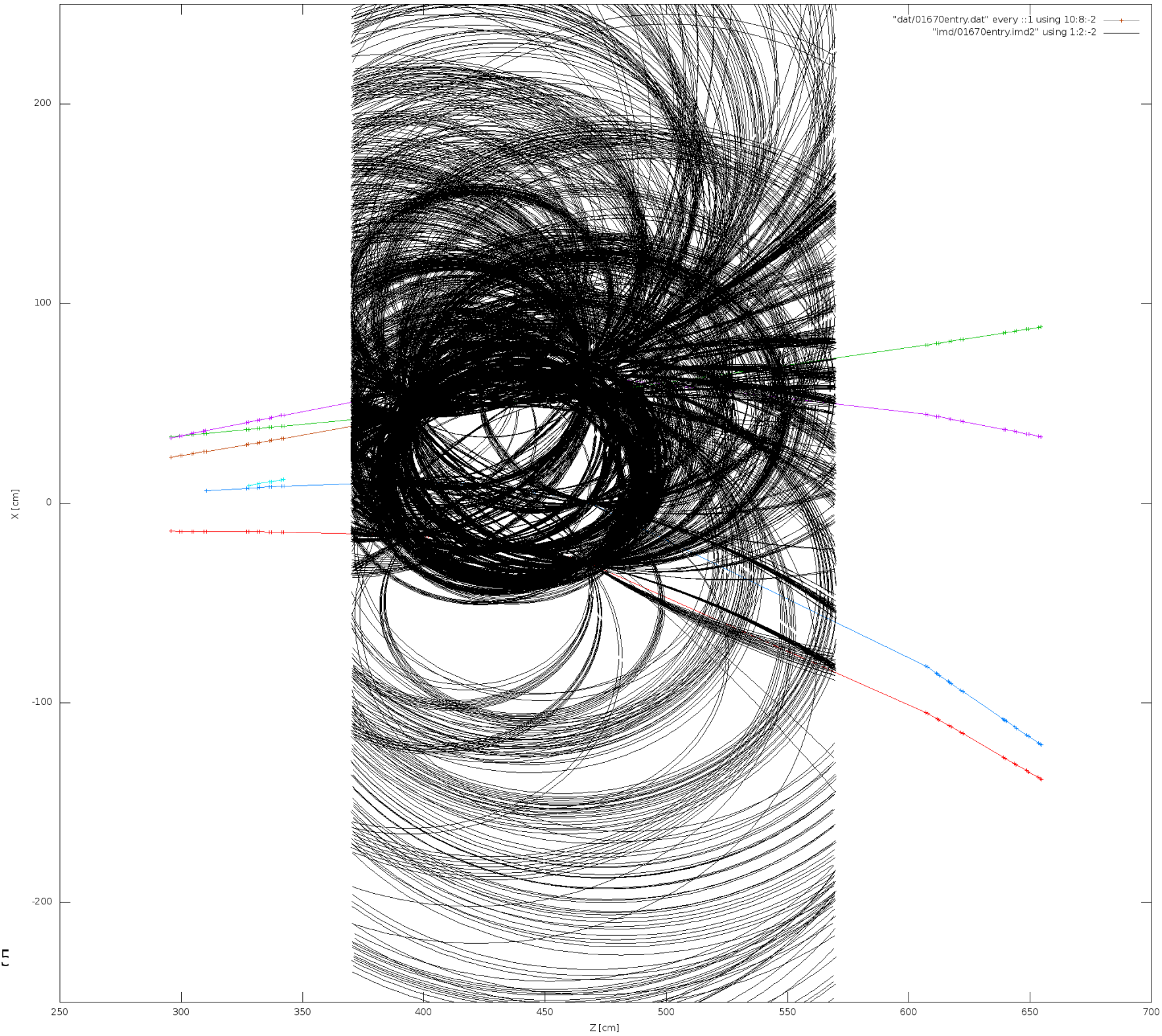
Execution in numbers

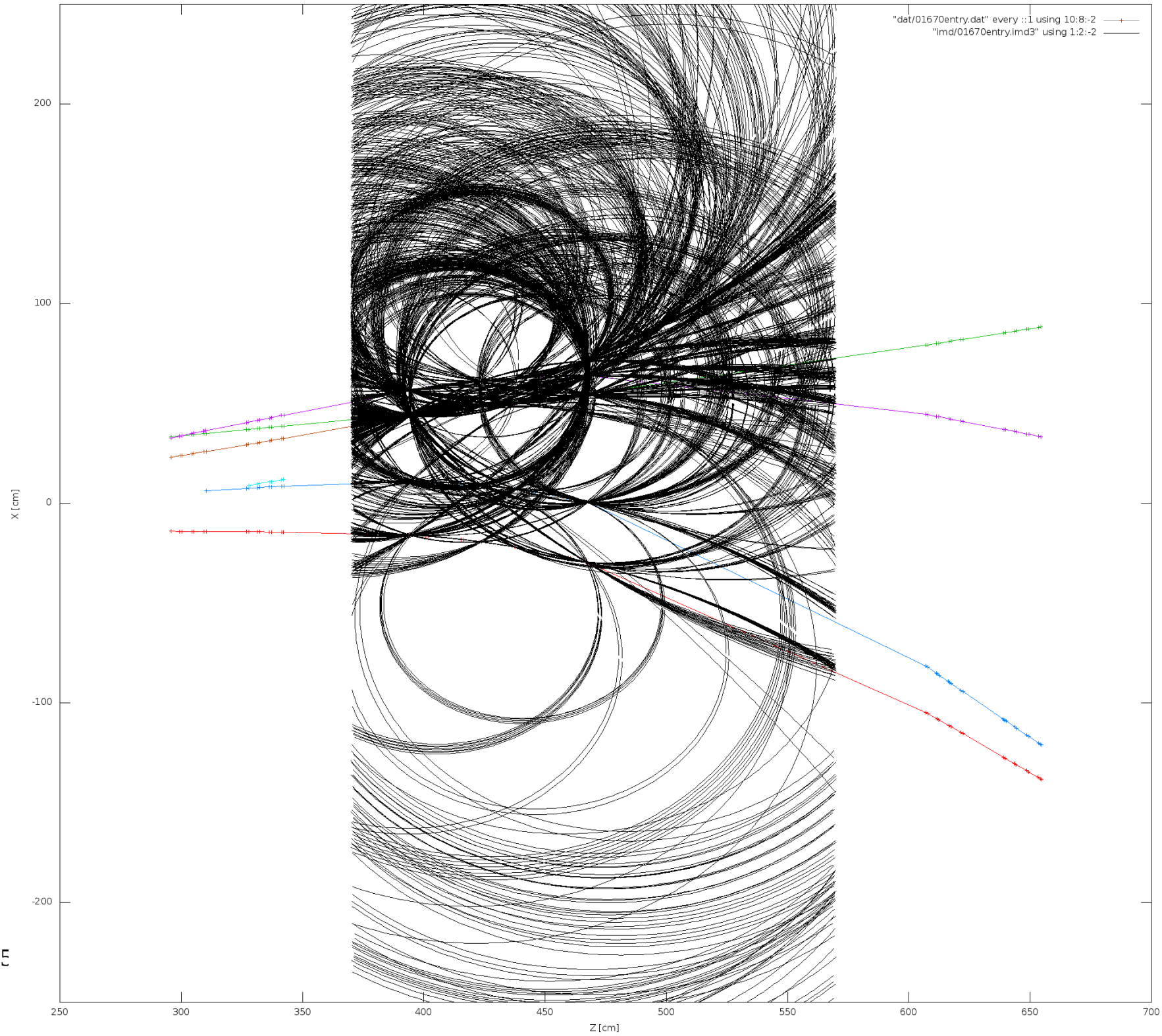
- **34 hits** generate **5984** and after filtering **84 circles**
 - 1.radius filter dropped **3543** circles
 - 2.circle center filter dropped **1840** circles
 - 3.six point circle (0.010cm cutoff) filter dropped **480** circles
 - 4.FT56 tangent filter (0.800cm cutoff) dropped **37** circles
 - 5.FT12 tangent filter (0.800cm cutoff) dropped **0** circles

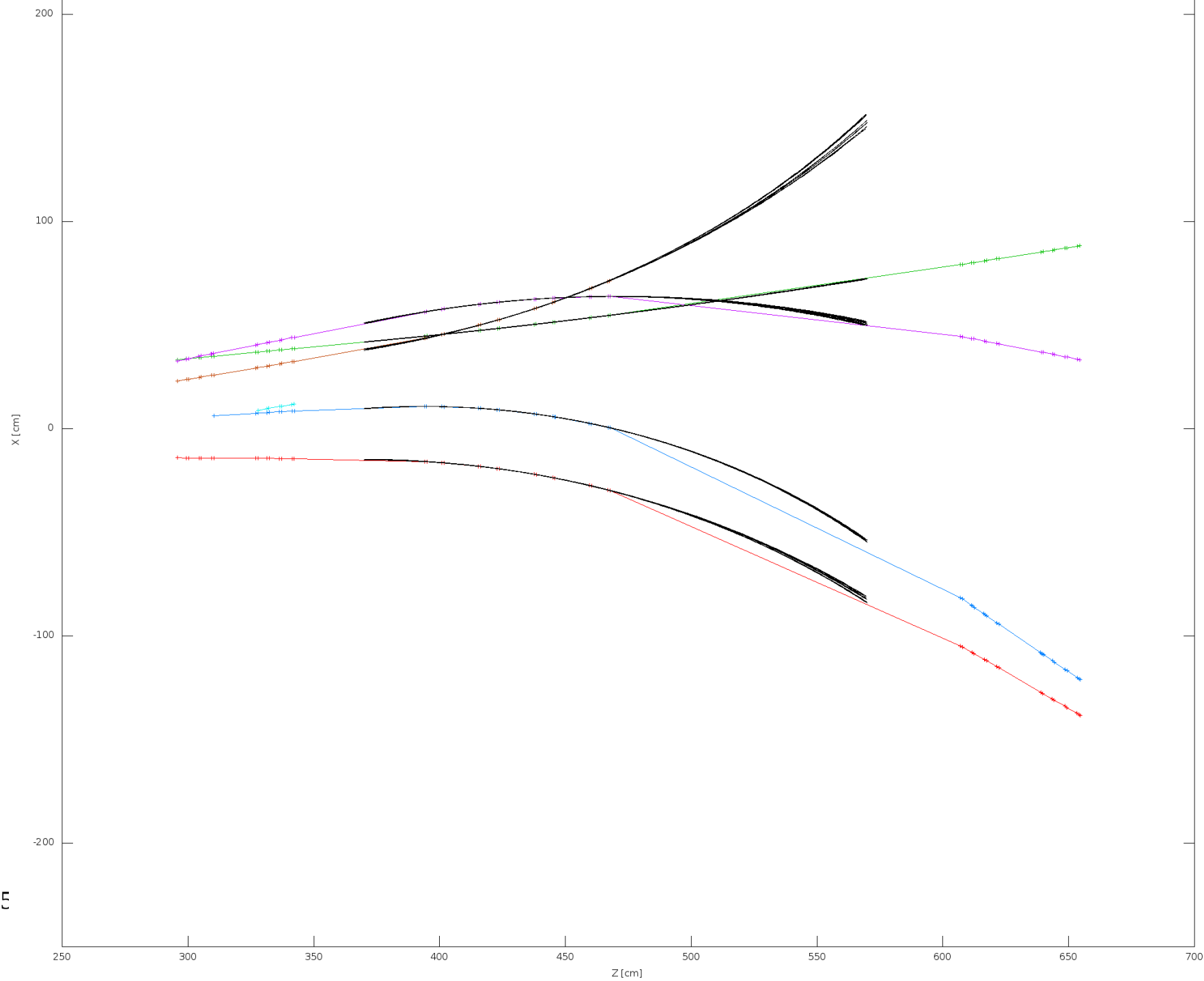
single threaded, Intel Core i5 M520@2.40GHz 10.18ms

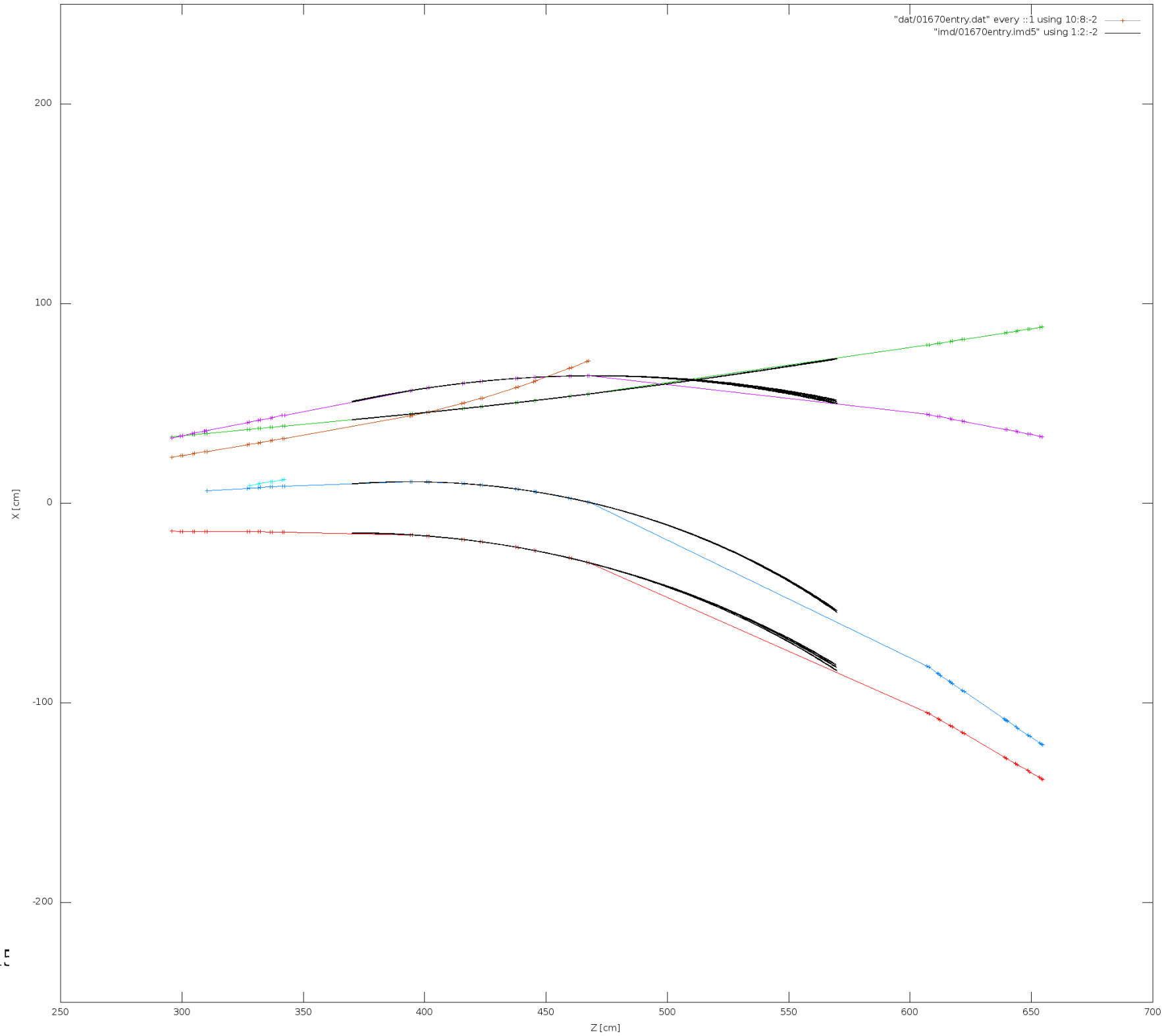


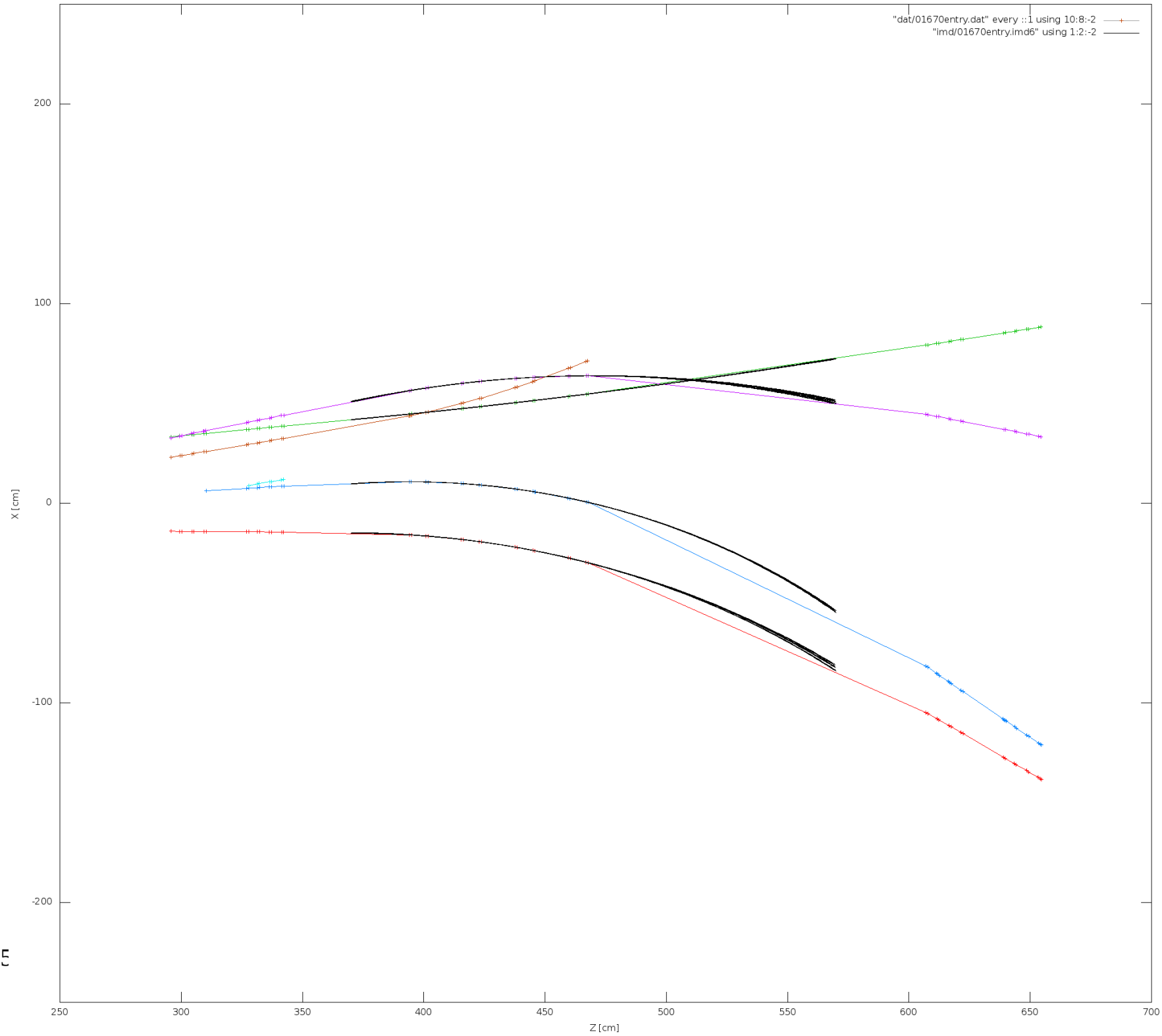








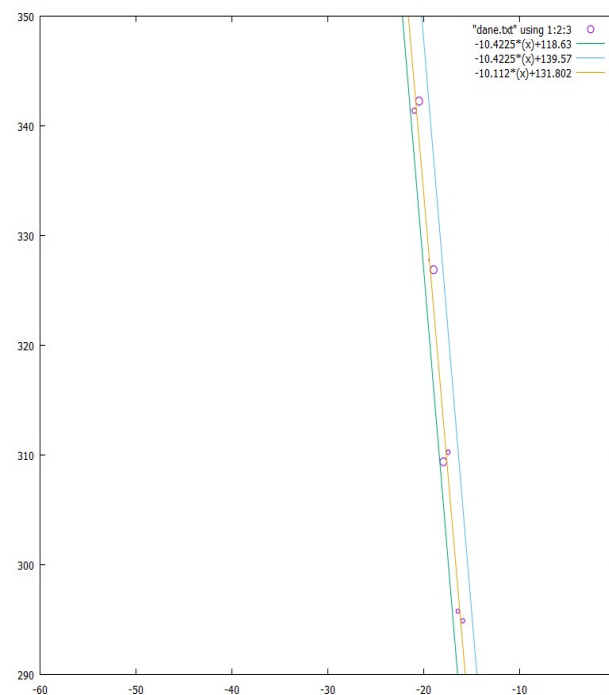
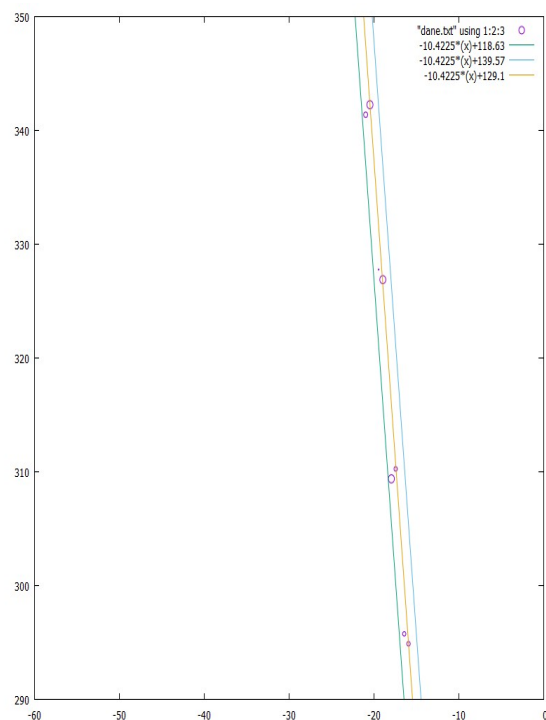




Circle-Tangent Region Method

Method Description

- Implemented for **FT 1, 2, 5, 6** (2D)
- Using most external layer hits in a segment, create paths, where other hits are searched for.



issues encountered (showstoppers)

Errors in simulation data:

- wrong straw numbers
- wrong geometry (z-coordinate value mismatch with FTS description)

Further Investigations

- add 3D (take into account skewed straws)
- add FT3, FT4
- evaluate applicability combination of methods
- once the issue of test data sorted-out will adjust methods and test efficiency
- optimisation and parallelisation