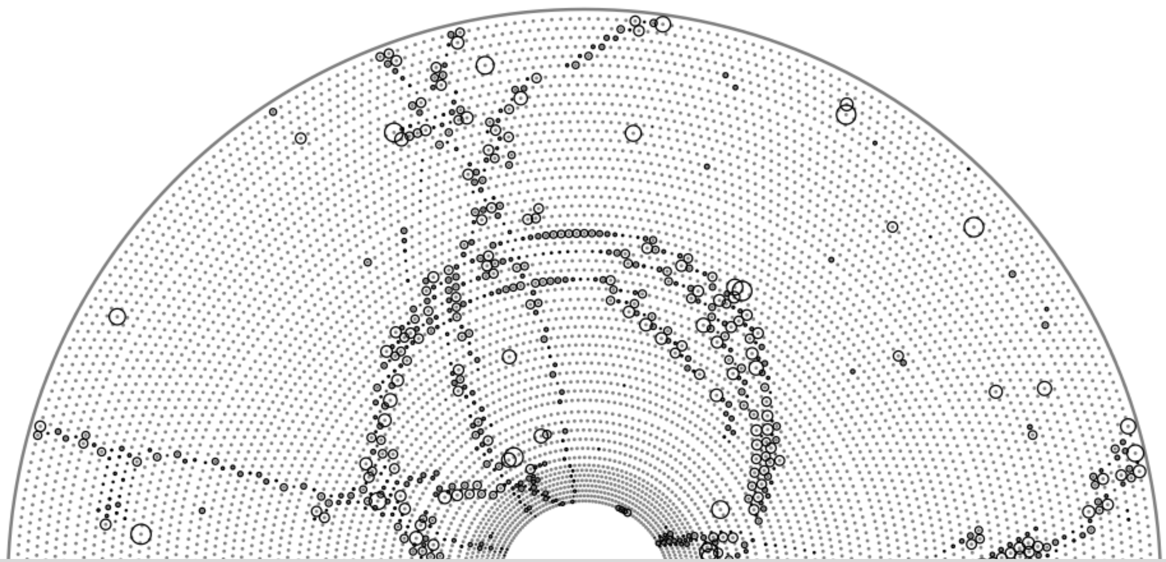
Applying Legendre transformation method for Belle II tracking

DPG Spring Meeting, Particle Physics division

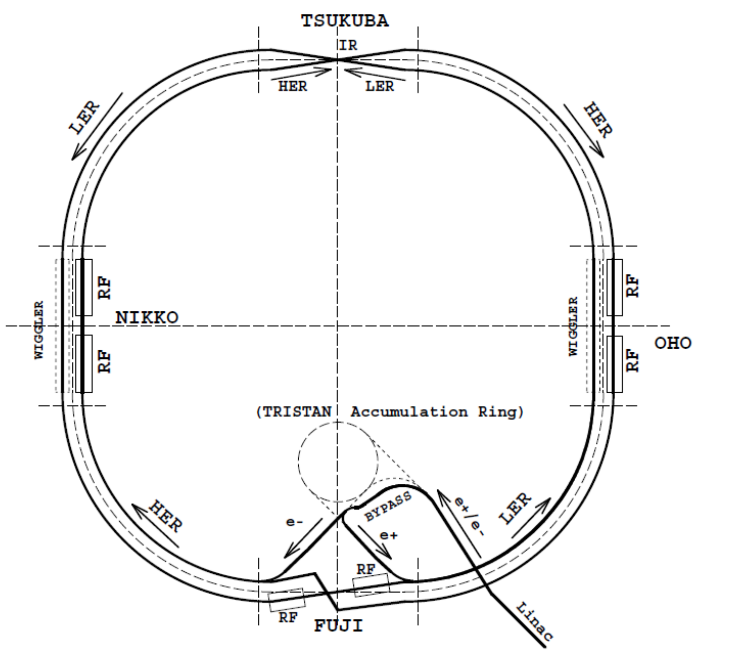
Viktor Trusov, Michael Feindt, Martin Heck, Thomas Kuhr, Pablo Goldenzweig

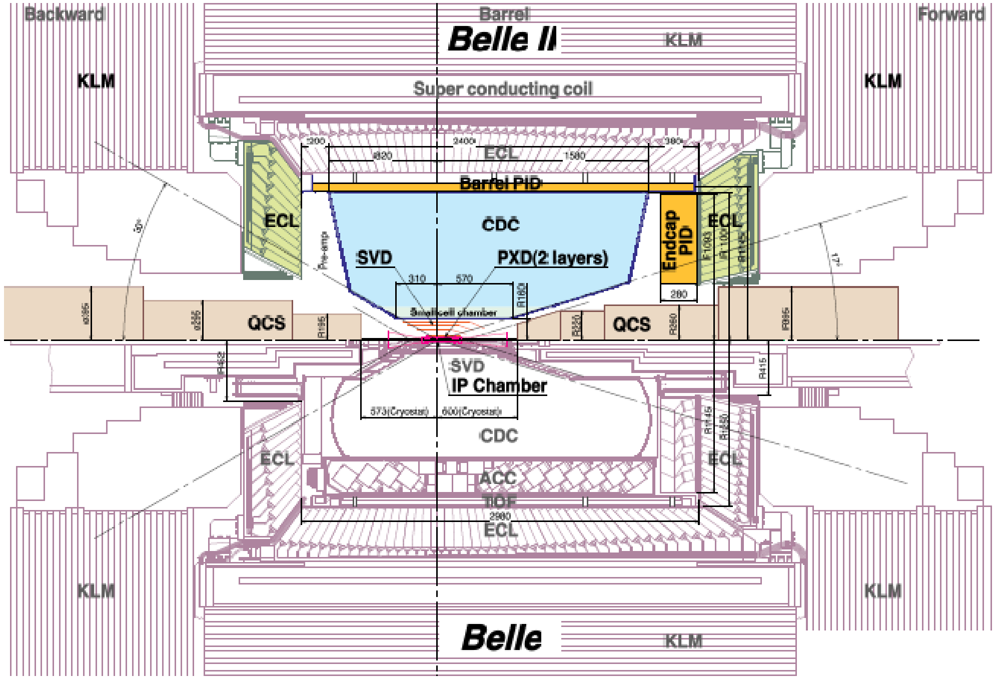
Karlsruhe Institute of Technology (KIT)



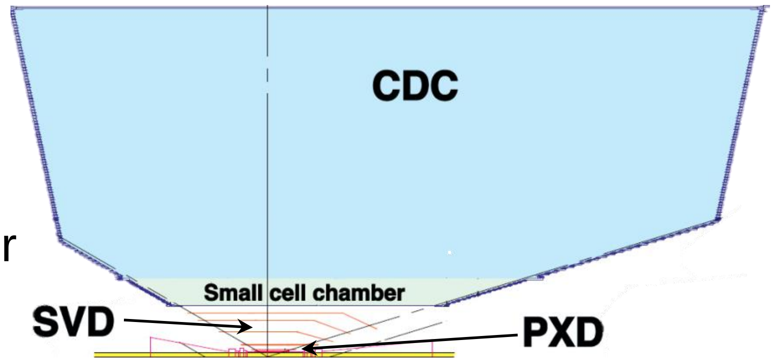
Belle II experiment

- SuperKEKB will be asymmetric electron-positron collider (SuperB-factory)
  - $E(e^-) = 7\ \mathrm{GeV};\ E(e^+) = 4\ \mathrm{GeV}, \sqrt{s} = 10.58\ \mathrm{GeV}$
- Belle II is experiment on SuperKEKB
- Design luminosity: $8 \times 10^{35} cm^{-2} s^{-1}$ (50 times higher than KEKB)





Tracking

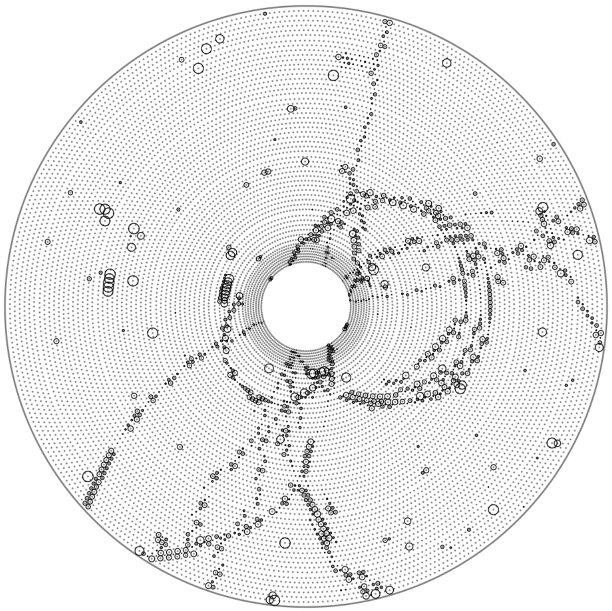


- Belle 2 tracking consist of:
  - PXD – Silicon pixel detector
  - SVD – Double sided silicon strip detector
  - CDC – Small cell central drift chamber

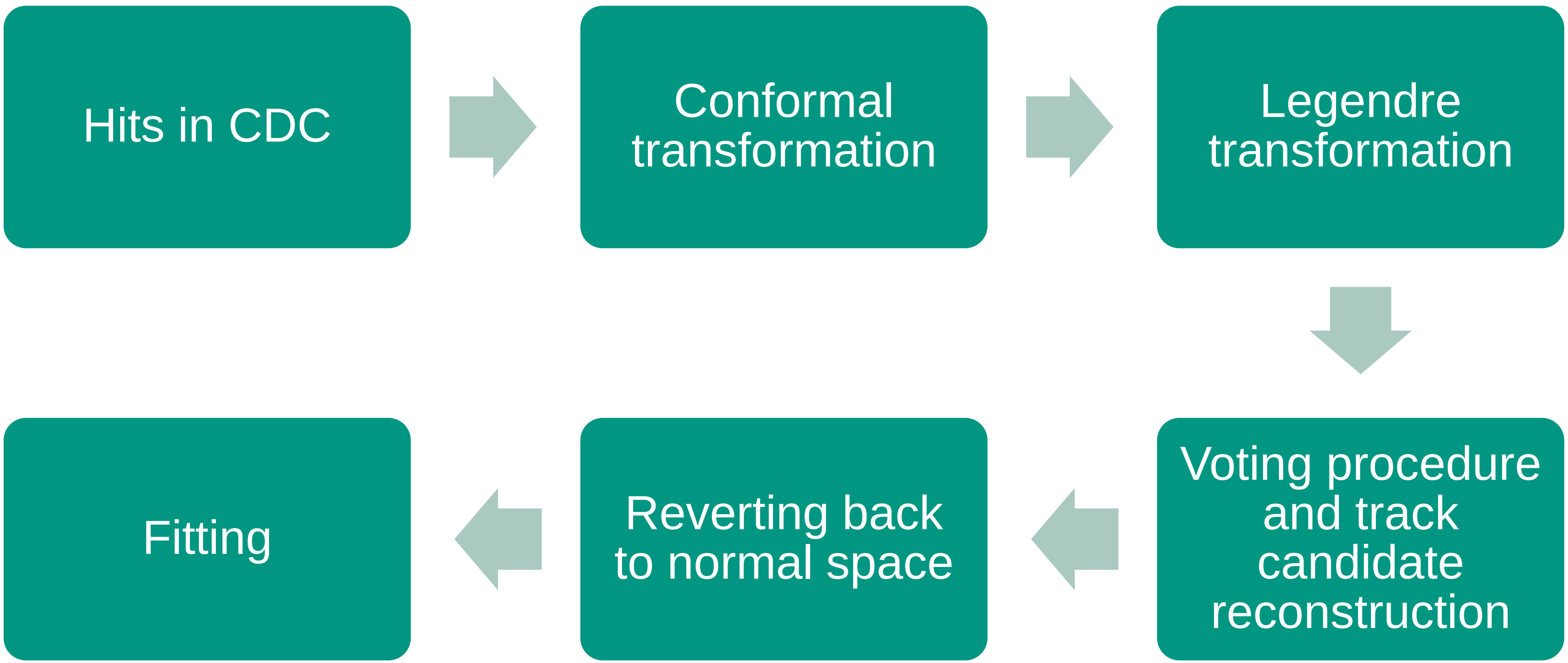
- Main goals of CDC tracking:
  - reconstruct charged tracks and measure their momenta precisely
  - provide particle identification information using measurements of energy loss within its gas volume
  - it provides efficient and reliable trigger signals for charged particles

Tracking

- Charged particles ionize gas in drift chamber, and electrons are drifting in electrical field to sensible wires and produce hits
- Drift distance for each hit can be calculated using the drift time which electrons need to reach the wire
- The task of track finding is to determine which hits belongs to a common track
- There exist few methods for hit pattern recognition
  - We present a method of track finding which based on reconstruction of linear hit patterns in conformal space



Chain of the method

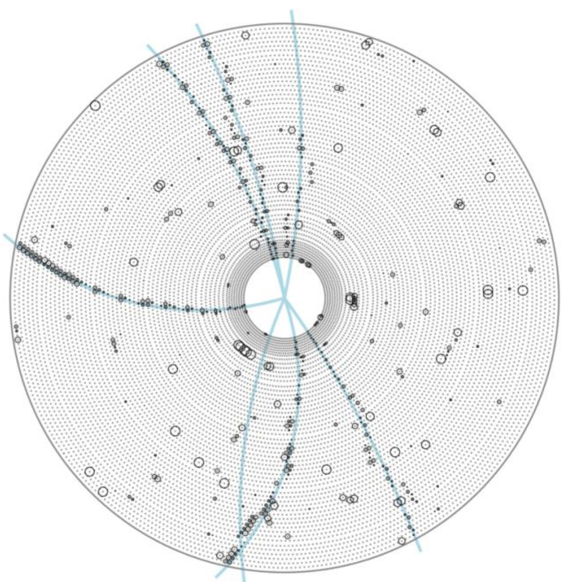


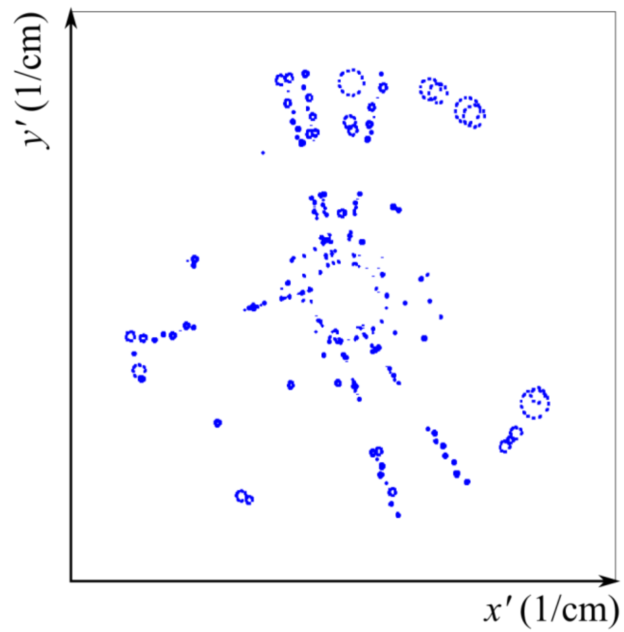
Conformal transformation

- Conformal transformation which transforms circles through origin into lines:

$$x' = \frac{2x}{x^2 + y^2}$$

$$y' = \frac{2y}{x^2 + y^2}$$



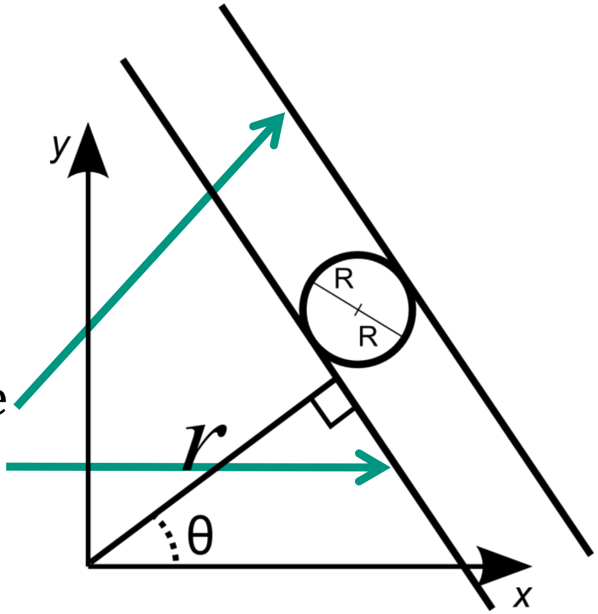


Transformation into Legendre space

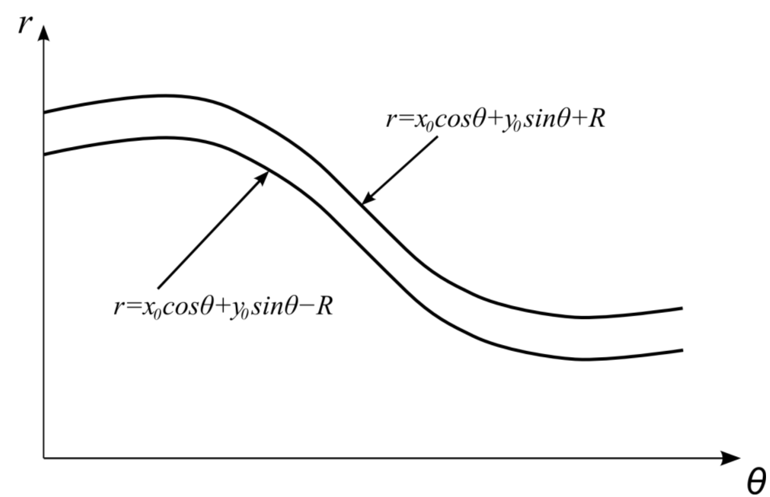
- The method is based on applying Legendre transformation to each drift circle in conformal space
- Legendre transformation of the circle can be written in next form:

$$f(x) \overset{\mathcal{L}}{\leftrightarrow} \begin{cases} r = x_0 + \cos\theta + y_0 \sin\theta + R & \text{for concave} \\ r = x_0 + \cos\theta + y_0 \sin\theta - R & \text{for convex} \end{cases}$$

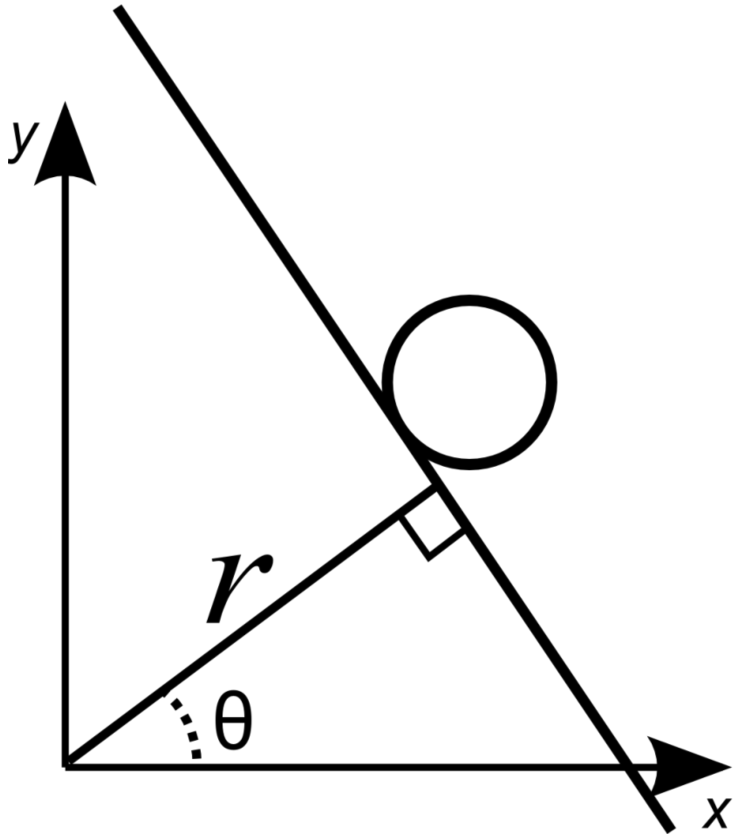
which presents tangents to the circle

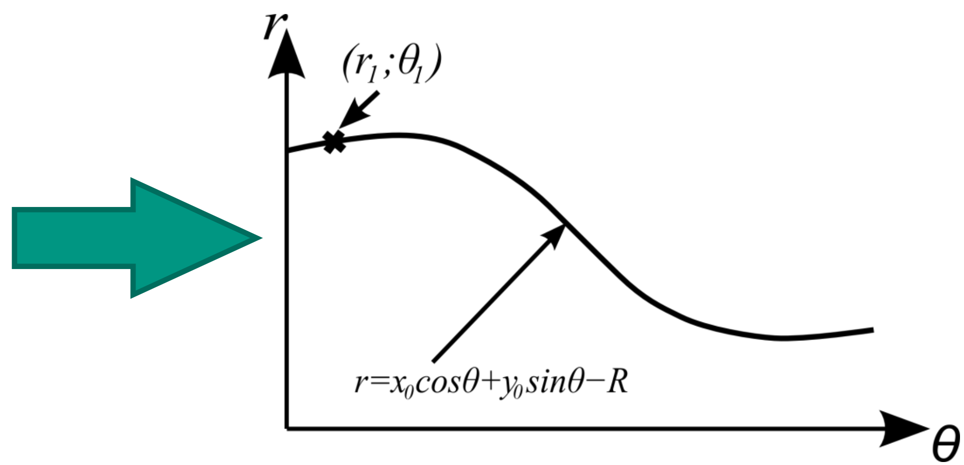


- Representation of the circle in the r, θ Legendre transformation space



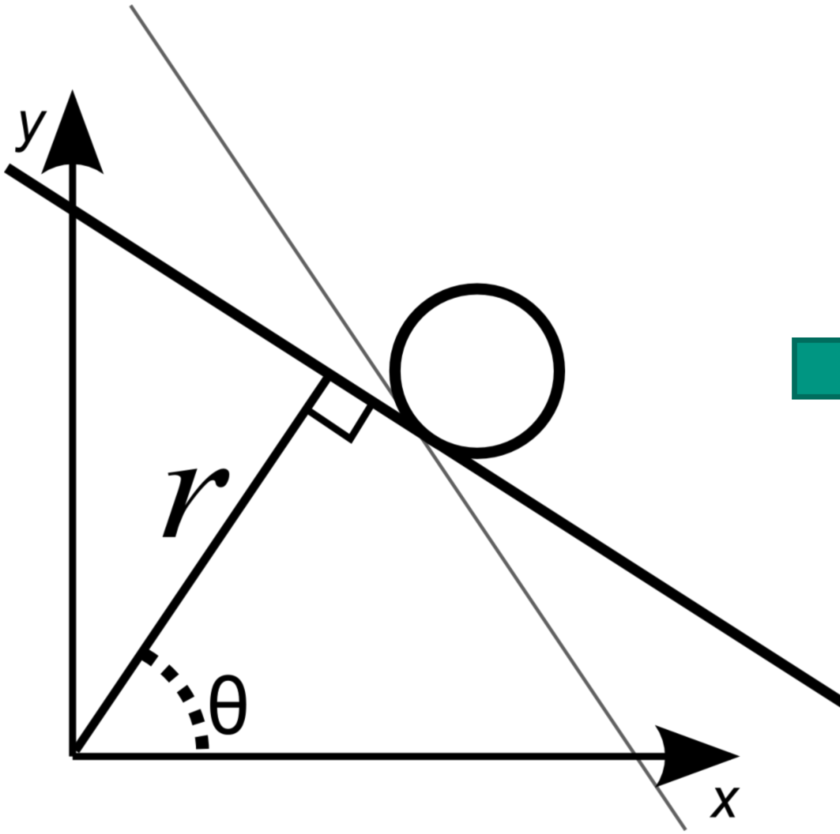
Transformation into Legendre space

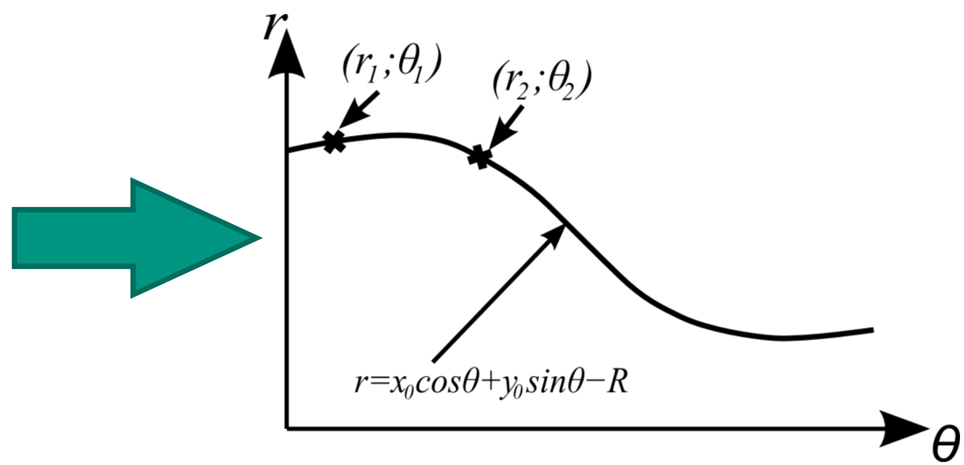




Each tangent has corresponding point in r, θ Legendre transformation space

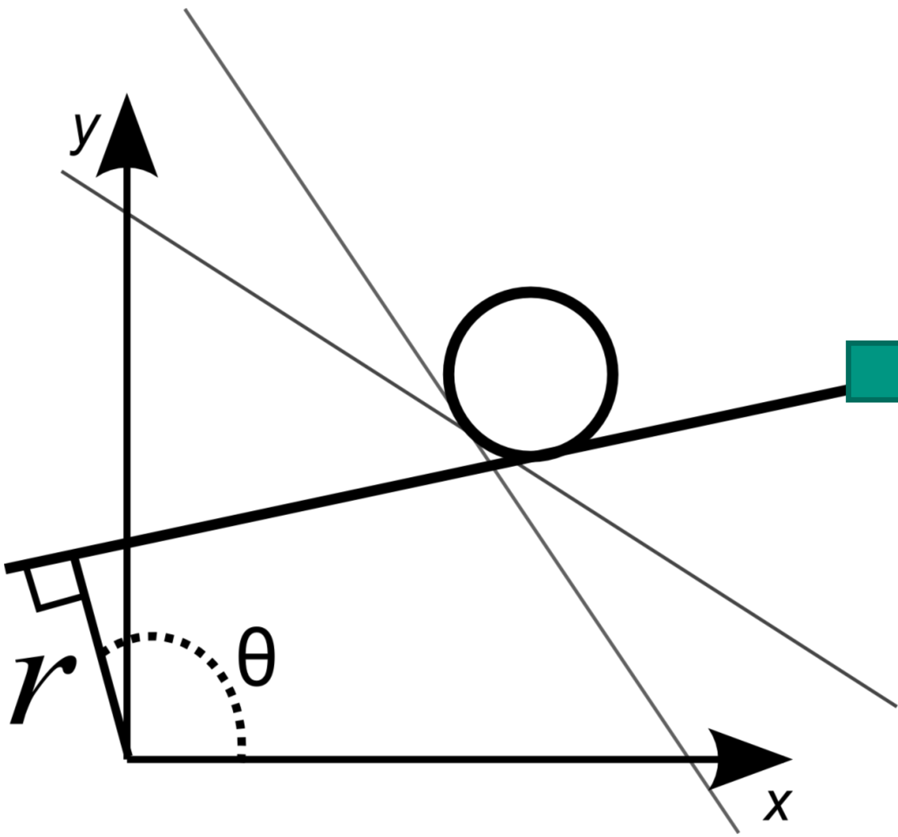
Transformation into Legendre space

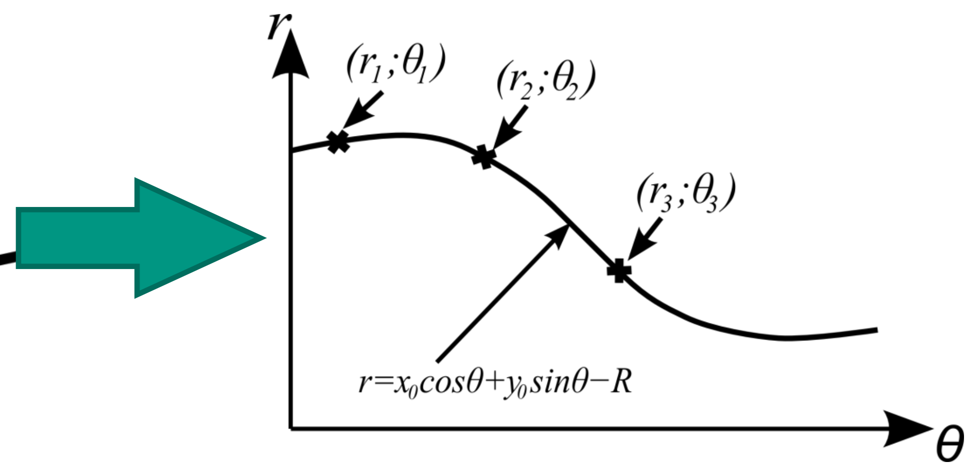




Each tangent has corresponding point in r, θ Legendre transformation space

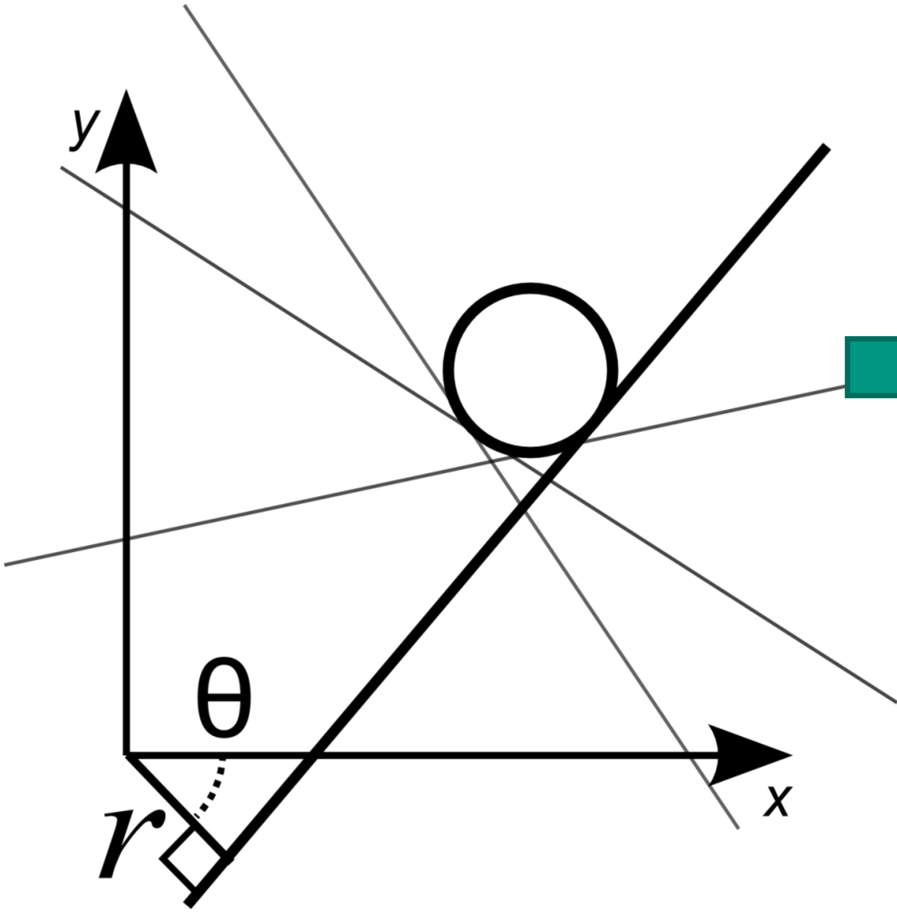
Transformation into Legendre space

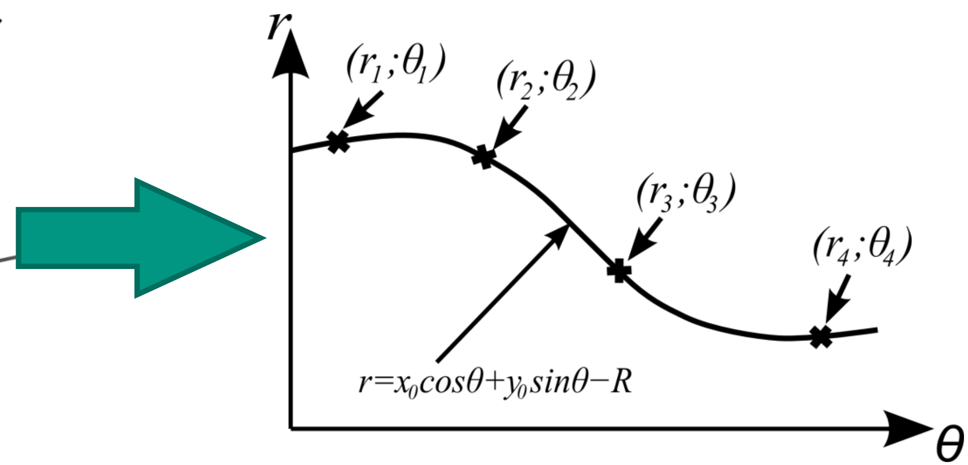




Each tangent has corresponding point in r, θ Legendre transformation space

Transformation into Legendre space

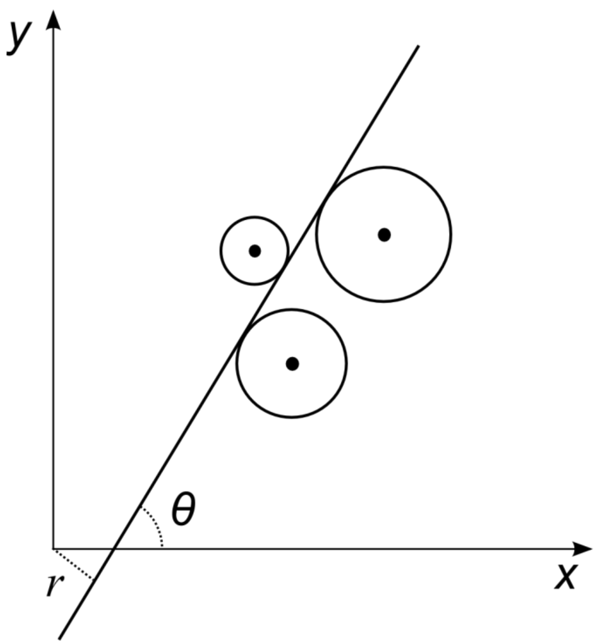


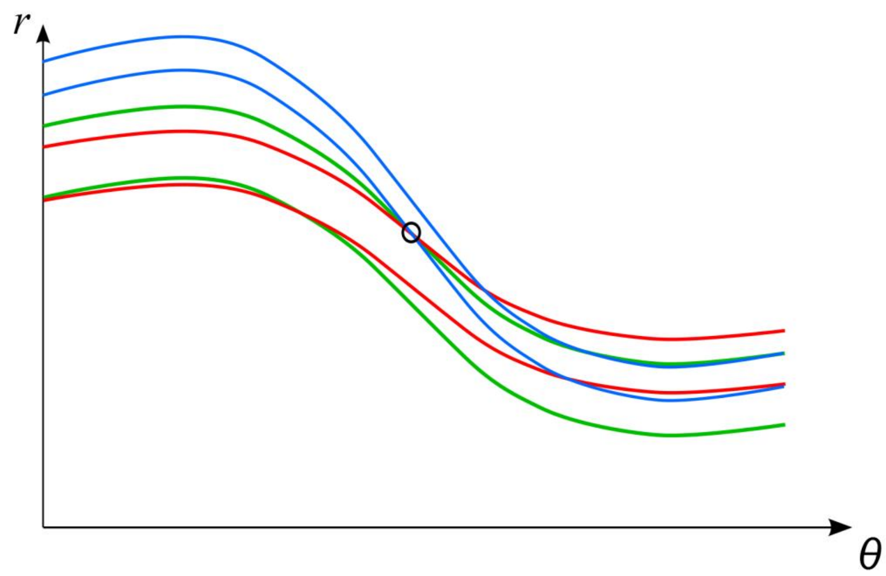


Each tangent has corresponding point in r, θ Legendre transformation space

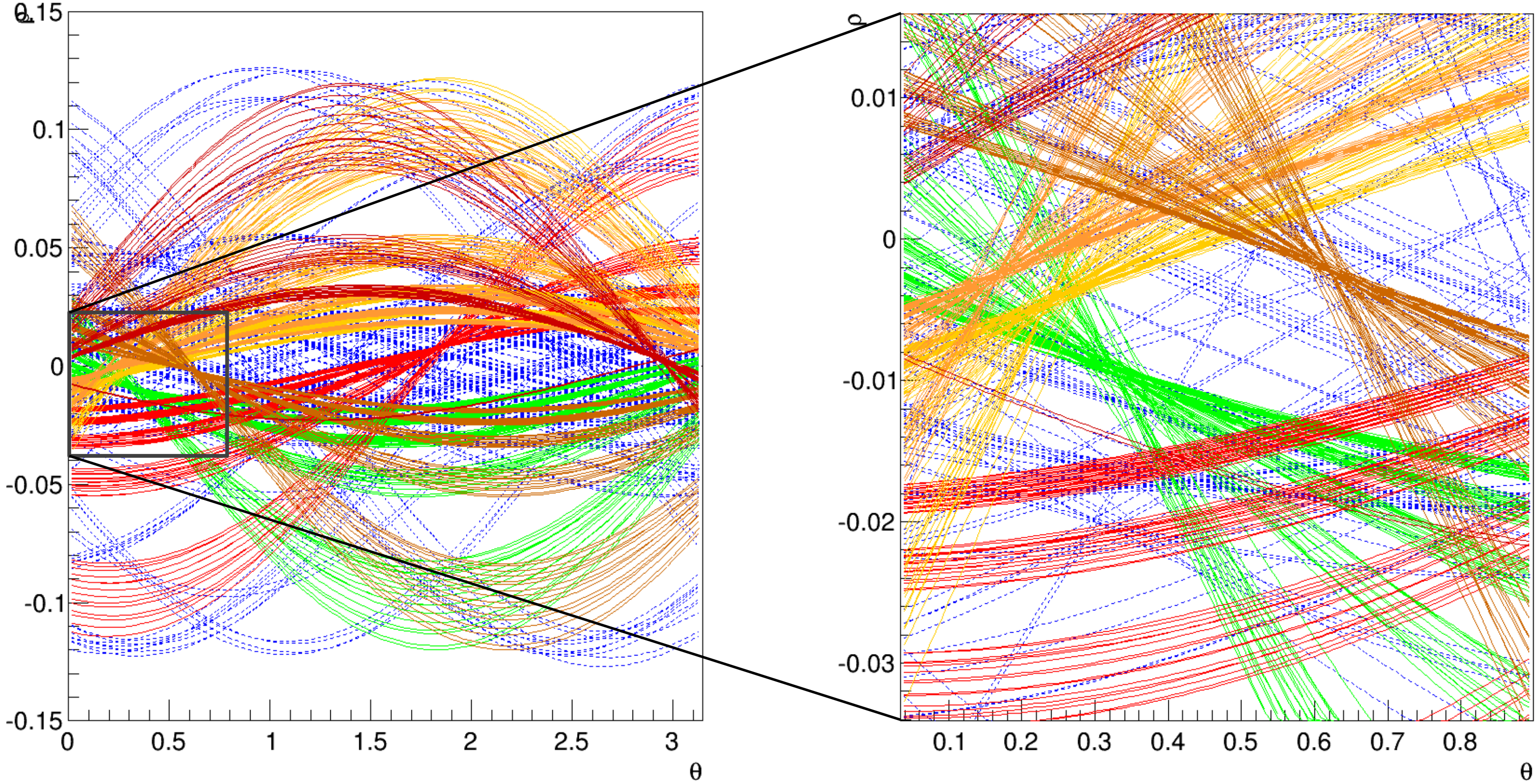
Finding of tracks

- The point of most sinogram intersection in r, θ space represents parameters of the common tangent to each drift circle belonging to the track



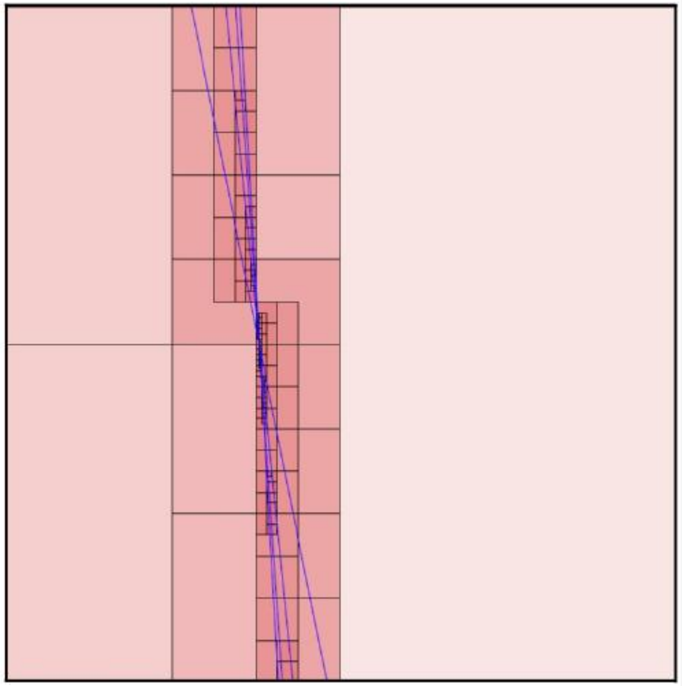


Sinograms of simulated event



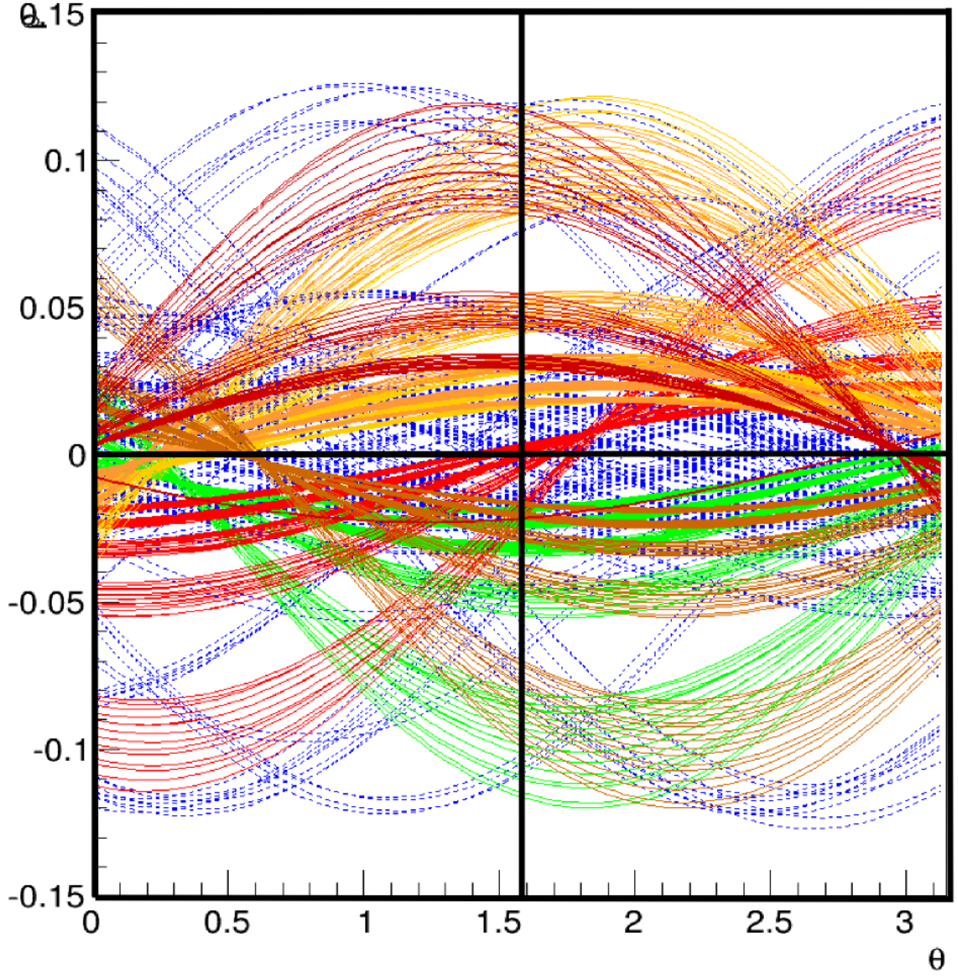
Voting or "How to find the point of most intersections?"

- Let each drift circle *vote* for a set of possible parameters in Legendre space
  - Bin acquires vote if sinogram of drift circle passes through it
- Voting algorithm (based on 2-D binary search):
  - Split (r, θ) space into 4 bins
  - Accumulate votes in each bin
  - Select bins which passes threshold on number of votes
  - Continue bin splitting and voting until desired (r, θ) resolution reached
- Bin with the most of votes (hits) indicates hits pattern of track candidate



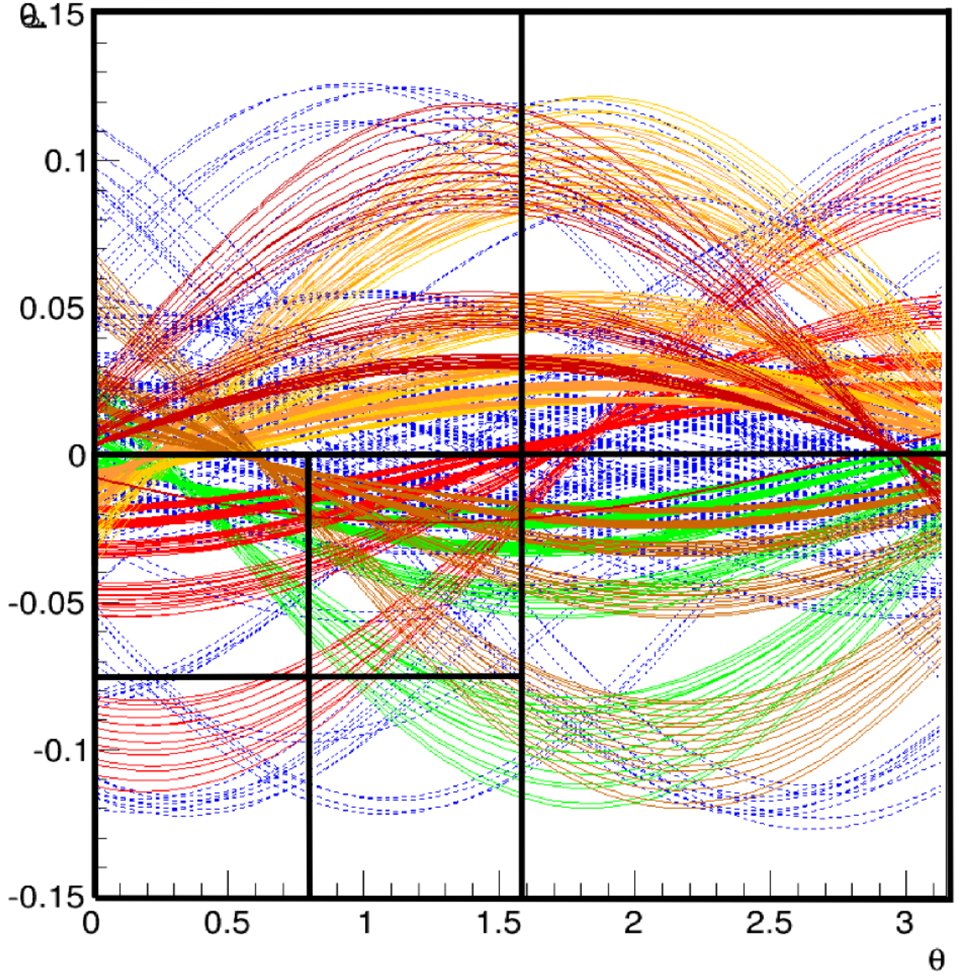
Voting

- **Step 1**



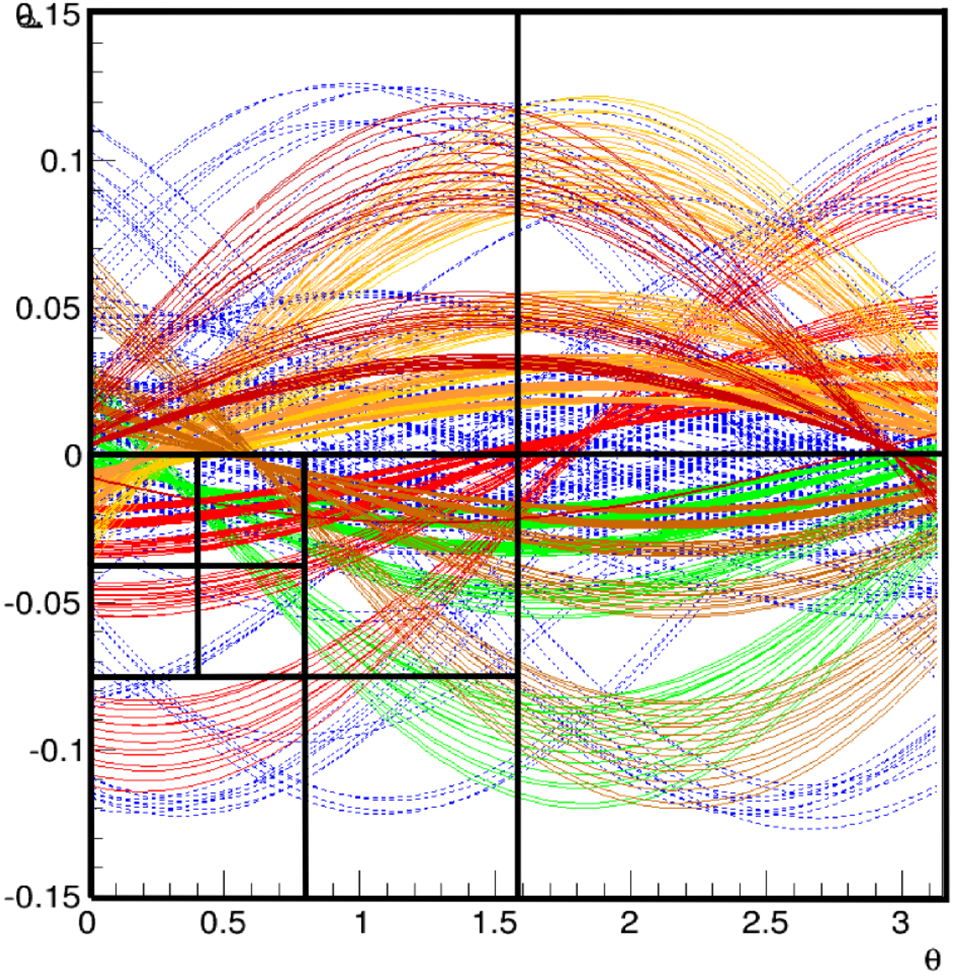
Voting

- **Step 2**



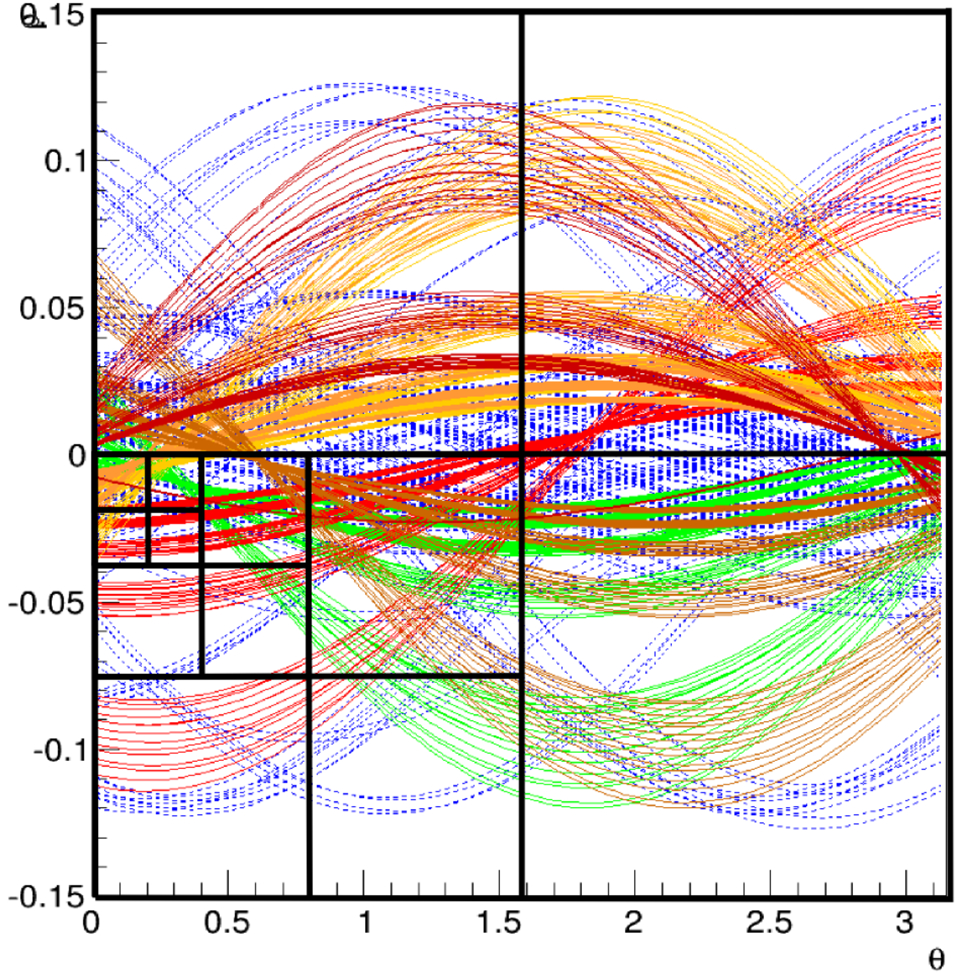
Voting

- **Step 3**



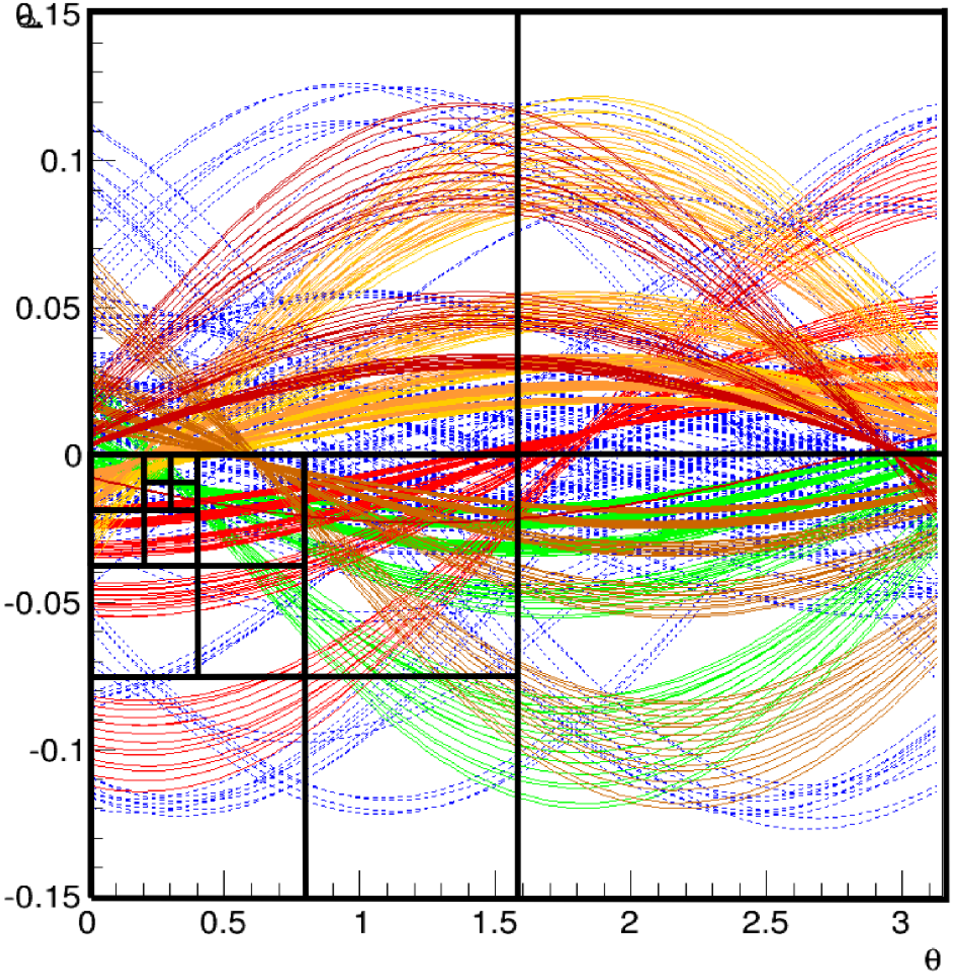
Voting

- **Step 4**



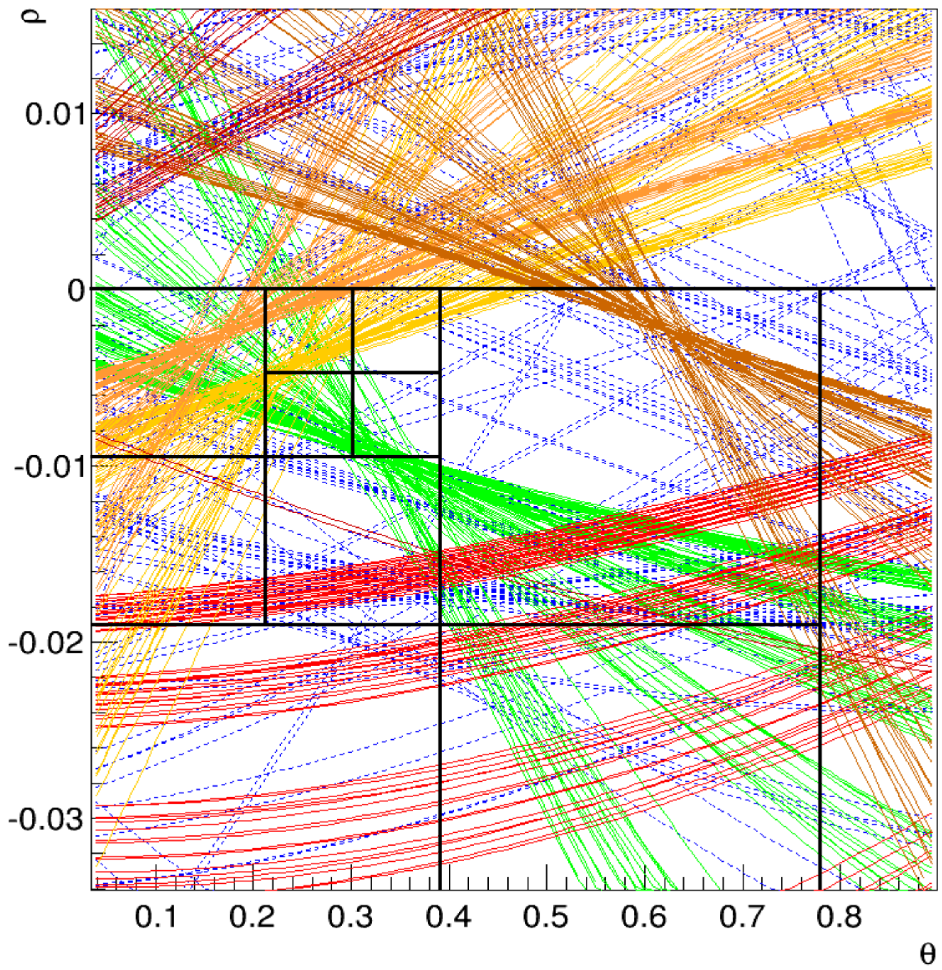
Voting

- **Step 5**



Voting

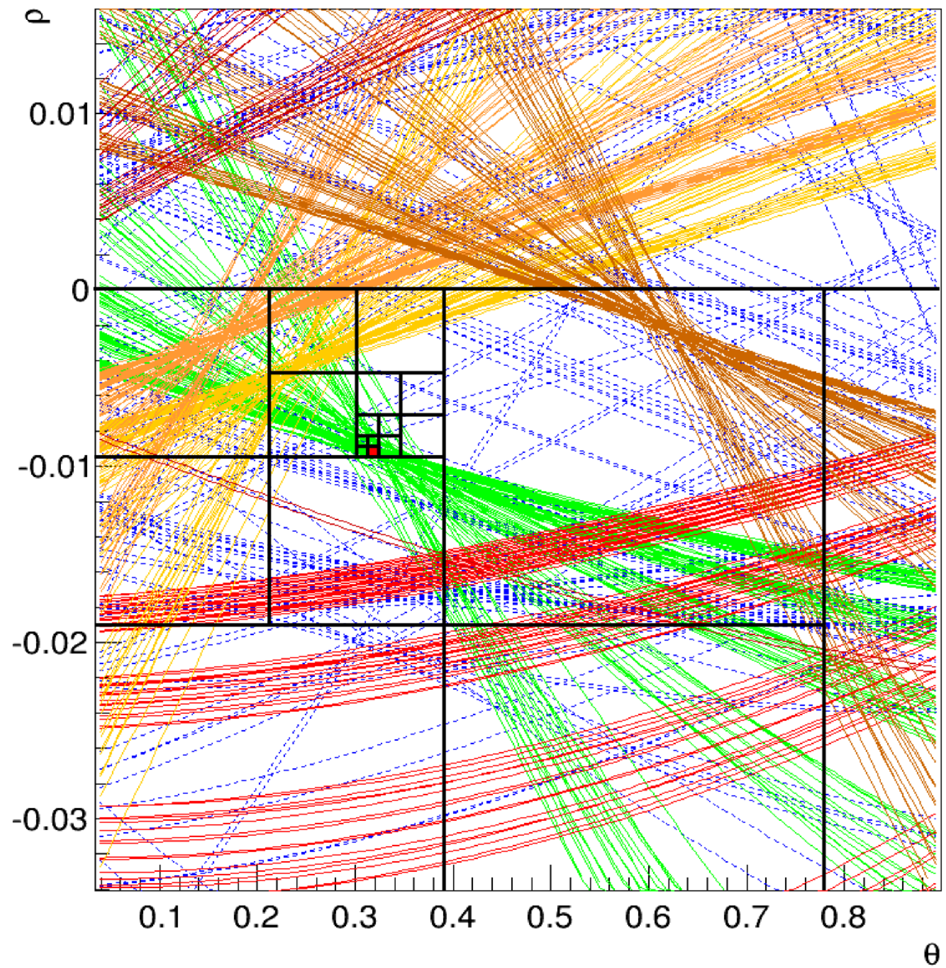
- **Step 6**



Voting

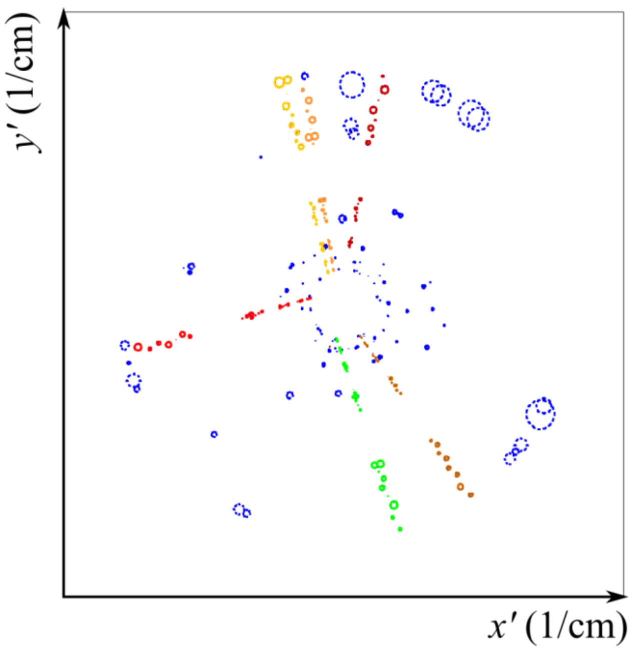
- **Step 10**

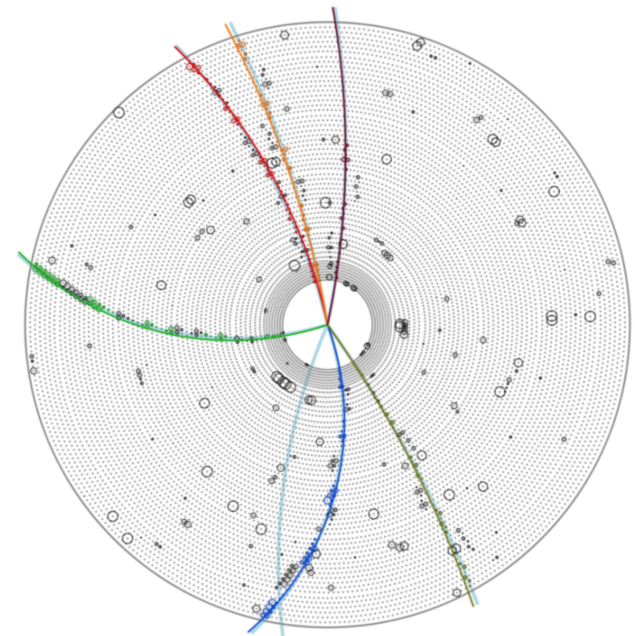
Last step, track candidate parameters are defined



Actual results of algorithm

- Simulated event: $B^- \rightarrow D^0(\rightarrow K^- \pi^+)\,\pi^- + \text{beam background}$



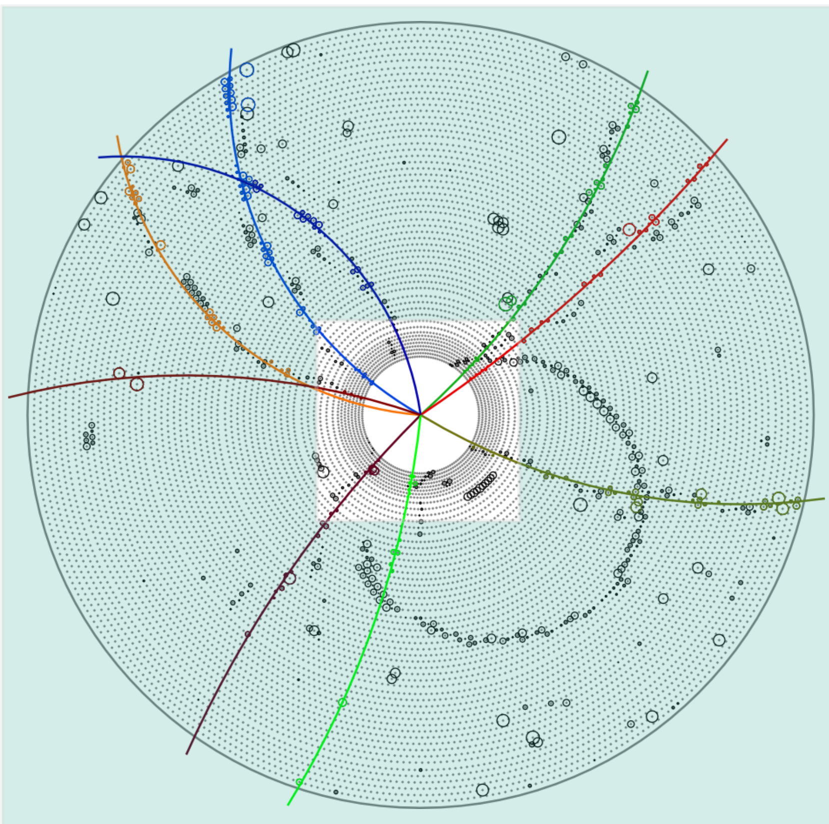


Working only with tracks originating close from IP

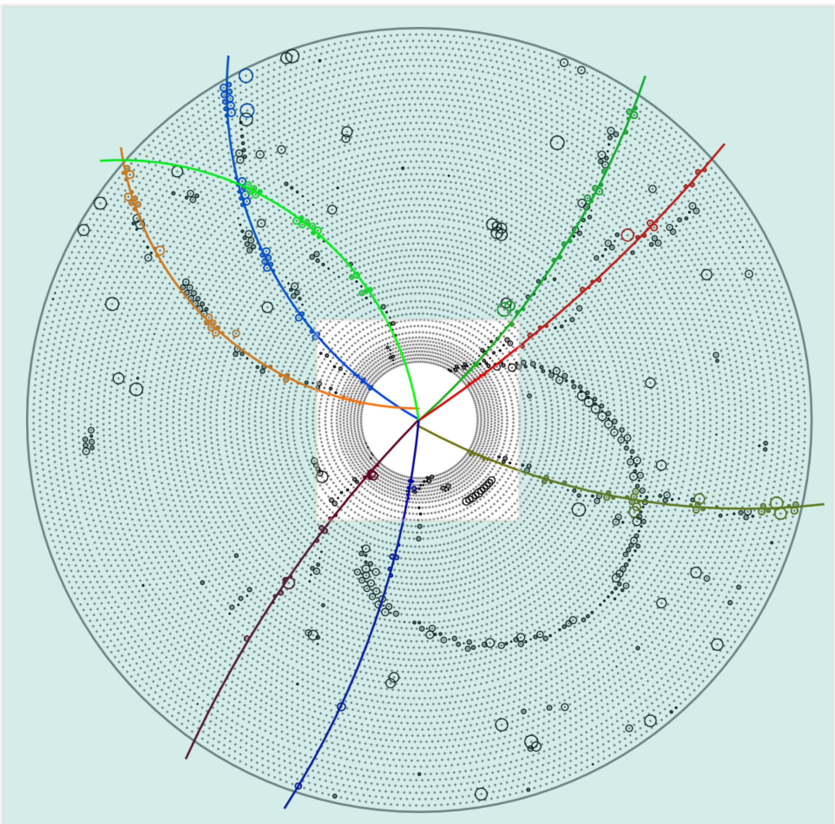
Efficient for finding tracks with high momentum

Track fitting

- Finder uses circular track fitting based on solving system of linear equation



Before



After

Conclusions

- The method can perform fast track finding
- Works quite well with high-p_t tracks
- Limited to tracks originating from IP
- Highly efficient for track finding

Thank you for your attention!