Current Status of SpiRIT-TPC Tracking

이정우

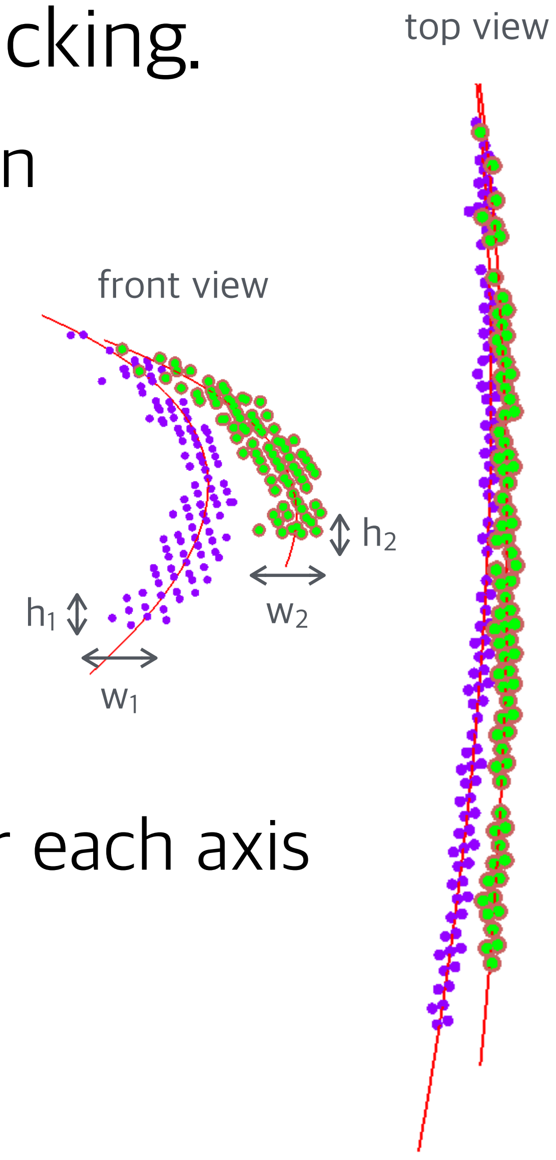
2016.7.5

What happend to Tracking

- First algorithm of track finding was taken from FOPIROOT(FOPI software); Riemann Tracking.
- Pros and cons of using Riemann tracking
    - pros: Save development time by using approved(?) software.
    - cons: Risk comming from property difference of FOPI-TPC and SpiRIT-TPC had to be taken.
- What is needed to use Riemann Tracking? Hit-Clusters.

Hit Clustering

- Why hit-cluster has to be given to Riemann tracking.
  1. In proximity correlator, only the distance between hit-track is compared to constant cut value.
  2. For non-clustered hits, distribution sigma for
     a) dispersion-axis (width) and
     b) perpendicular-axis (height) are different.



- Need of pre-tracking: Curve tracking
  1. Two axis, width and height are divided. Sigma for each axis are caculated to used as cut value.
  2. Not qualitative for full track finding.
  3. Built for hit-clustering.

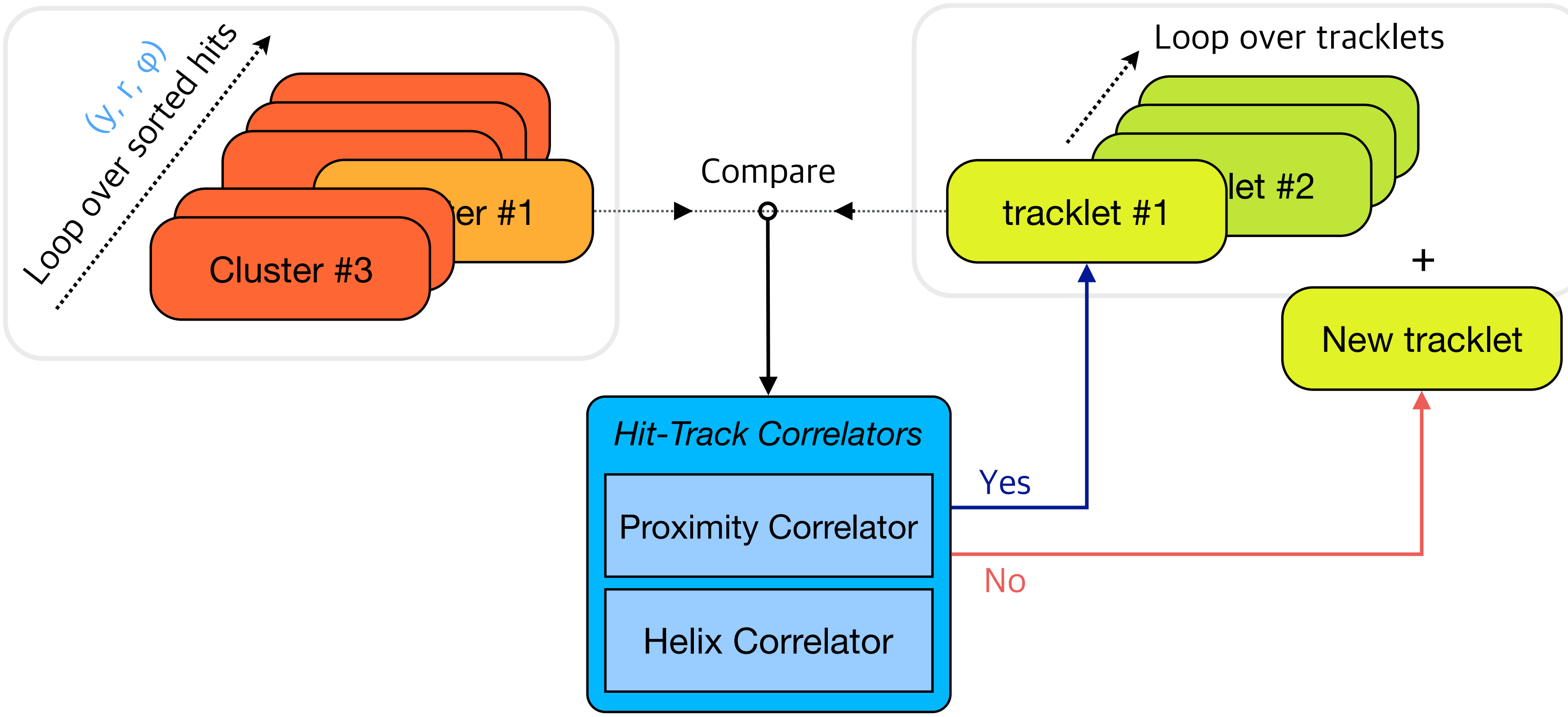
What happened to Tracking

Riemann Tracking

- Fast(non-iterative) and accurate circle fit, using Riemann sphere mapping.

What happened to Tracking

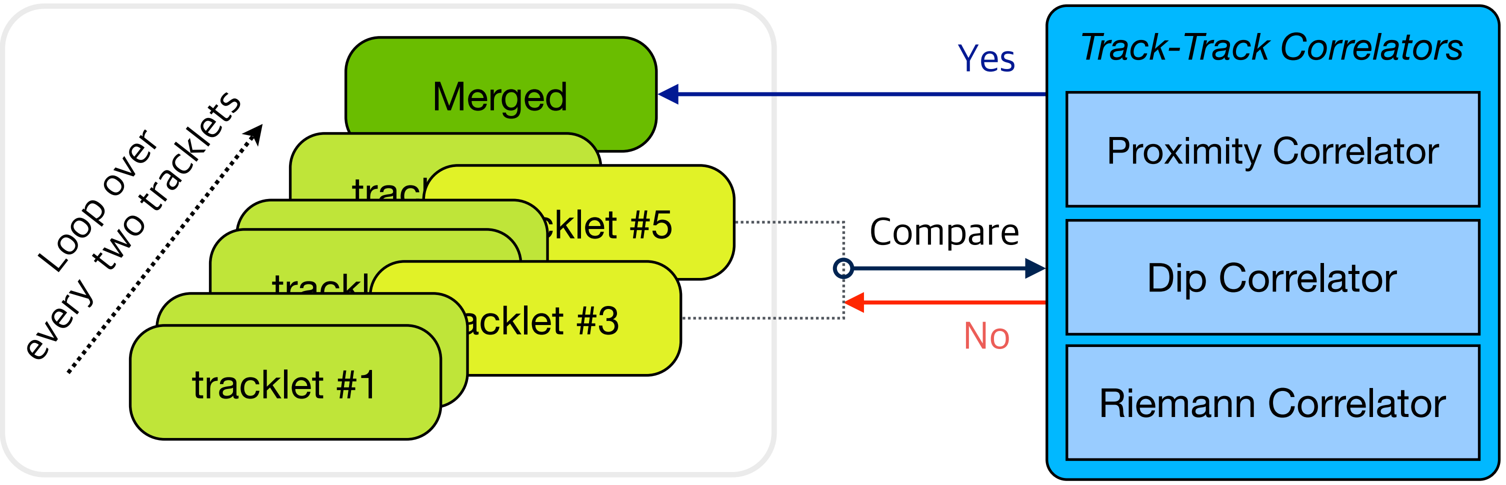
Riemann Tracking



- Hit belong to one track cannot belong to other track.
- Hard to distinguish fake/stable track while building tracks simultaneously.
- Broken tracks are not avoidable.

What happened to Tracking

Riemann Tracking



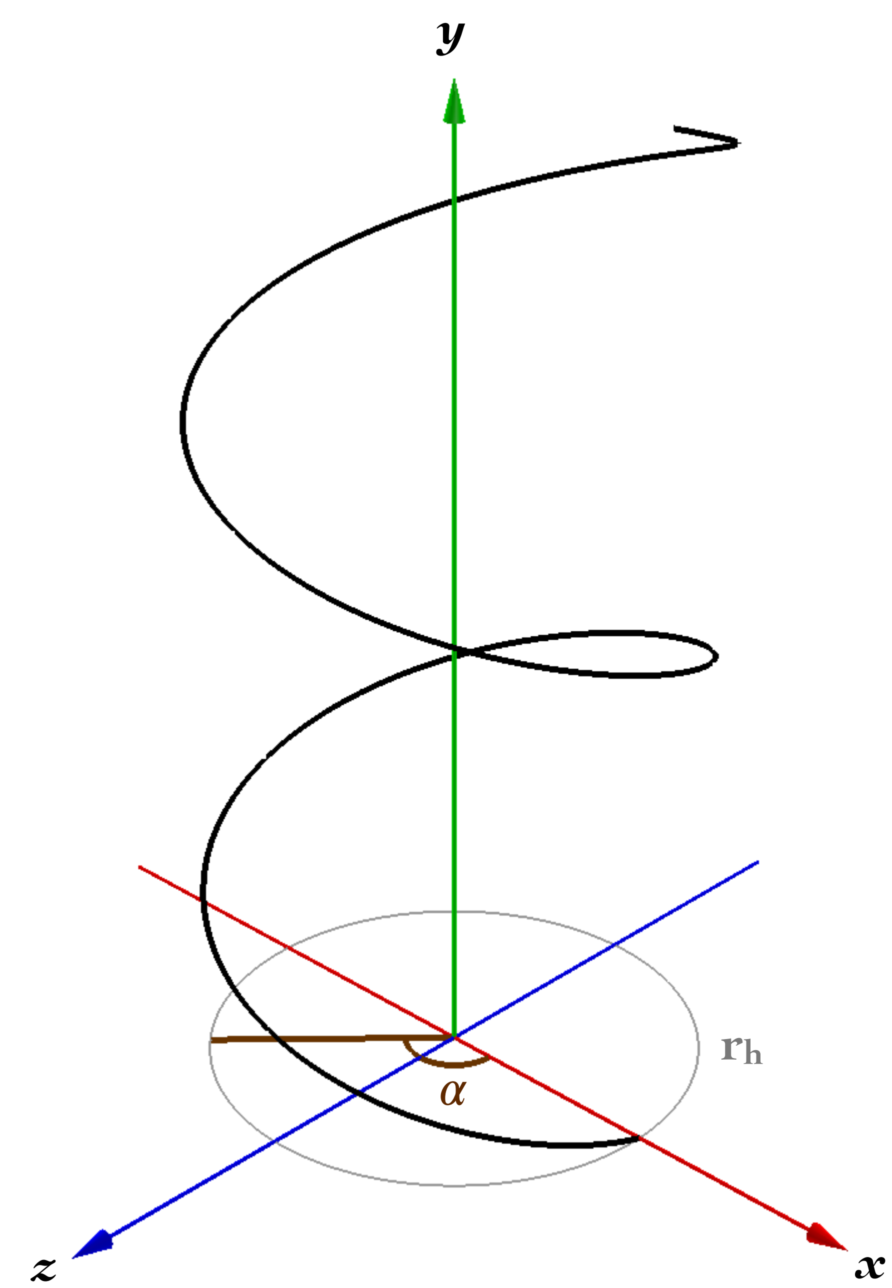
- In ideal situation, tracklets from one physical track should be merged. But reallity is different.
- Many times, following effects make trouble
  - Bad position resolution (caused by saturation)
  - Bad circle fit (comming from fixed position of Riemann sphere)
  - Bad clustering (my fault)

Summary until now

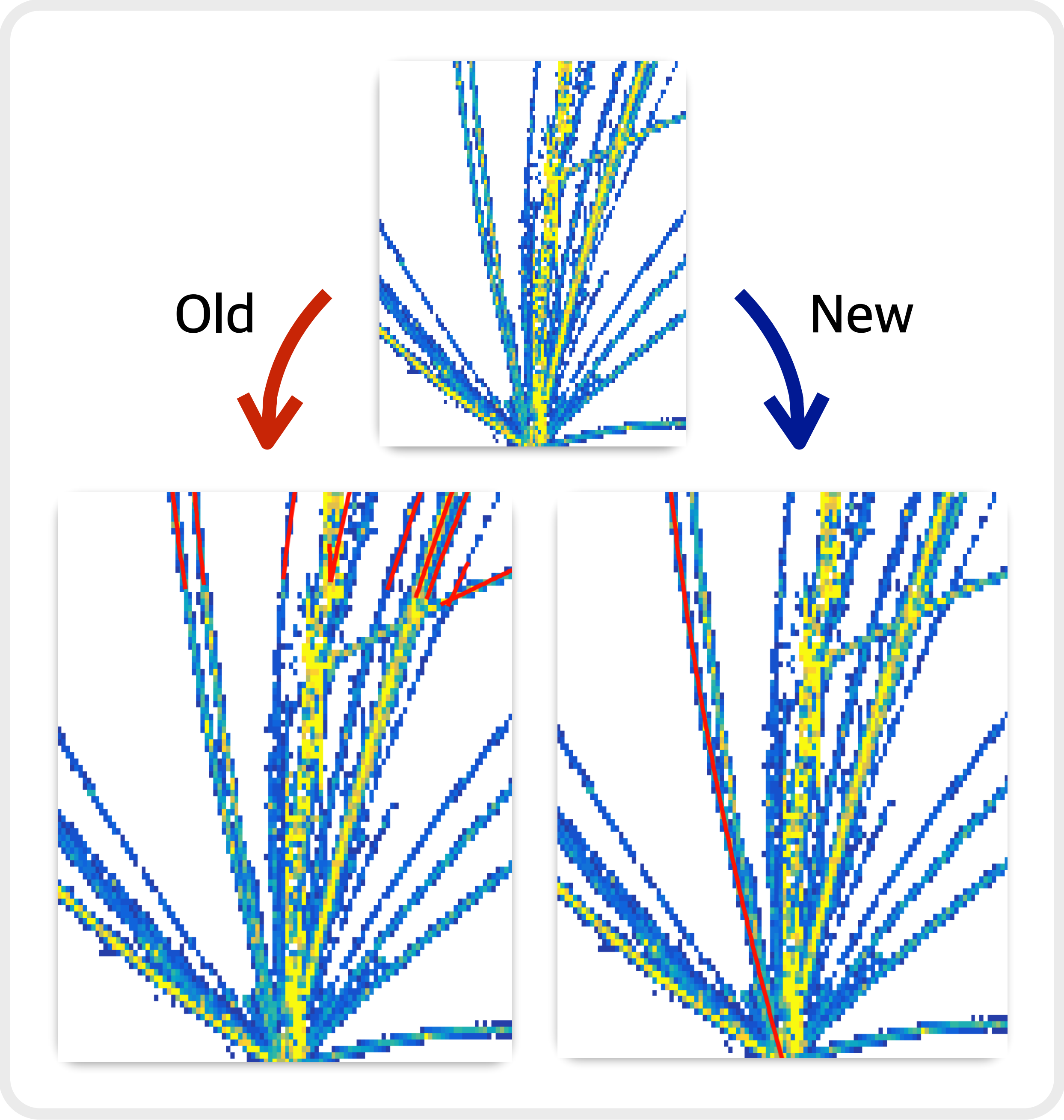
- Untill now
  1. We tried to use Riemann tracking
  2. Curve tracking was developed for clustering
  3. Broken tracks are not avoidable.
- Problems
  1. Dense system
  2. Bad position resolution (staturation)
  3. Parameterization of Riemann tracking.
- What we learned
  1. Riemann fit (circle fit) including possibility of improvement.
  2. Advantage of width and height axis.

Helix Tracking

- [x] Full control of the code.
- [x] Build full track one by one.
- [x] Helix to straight line map.
- [x] Use advantage of width of the track comming from electron dispersion.
- [x] Use self-update parameters.
  - Riemann sphere position and radius.
  - Proximity cut.
- [ ] Deal with shared hit.
- [x] Clustering for Genfit.

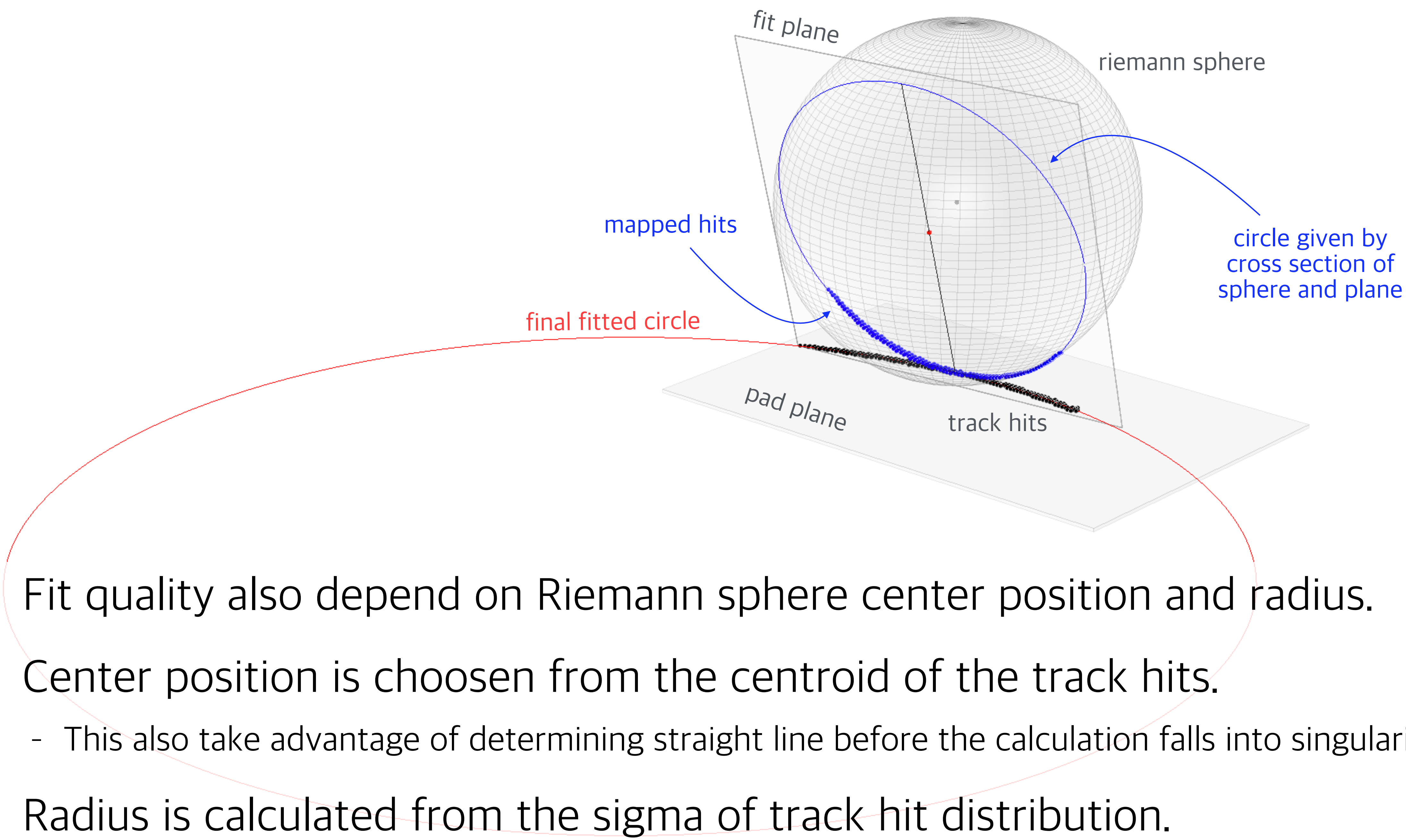


Event Map



- One to one 2D mapping from pad(row, layer) to pad hits.
- This enables one to build one full track before another track is built.
- New possibility of finding hits and continue building track from extrapolated position using event map.
- Used hits are left in the event map so other tracks also have chance to check the correlation.

Improvement of Riemann Fit



- Fit quality also depend on Riemann sphere center position and radius.
- Center position is choosen from the centroid of the track hits.
  - This also take advantage of determining straight line before the calculation falls into singularity.
- Radius is calculated from the sigma of track hit distribution.

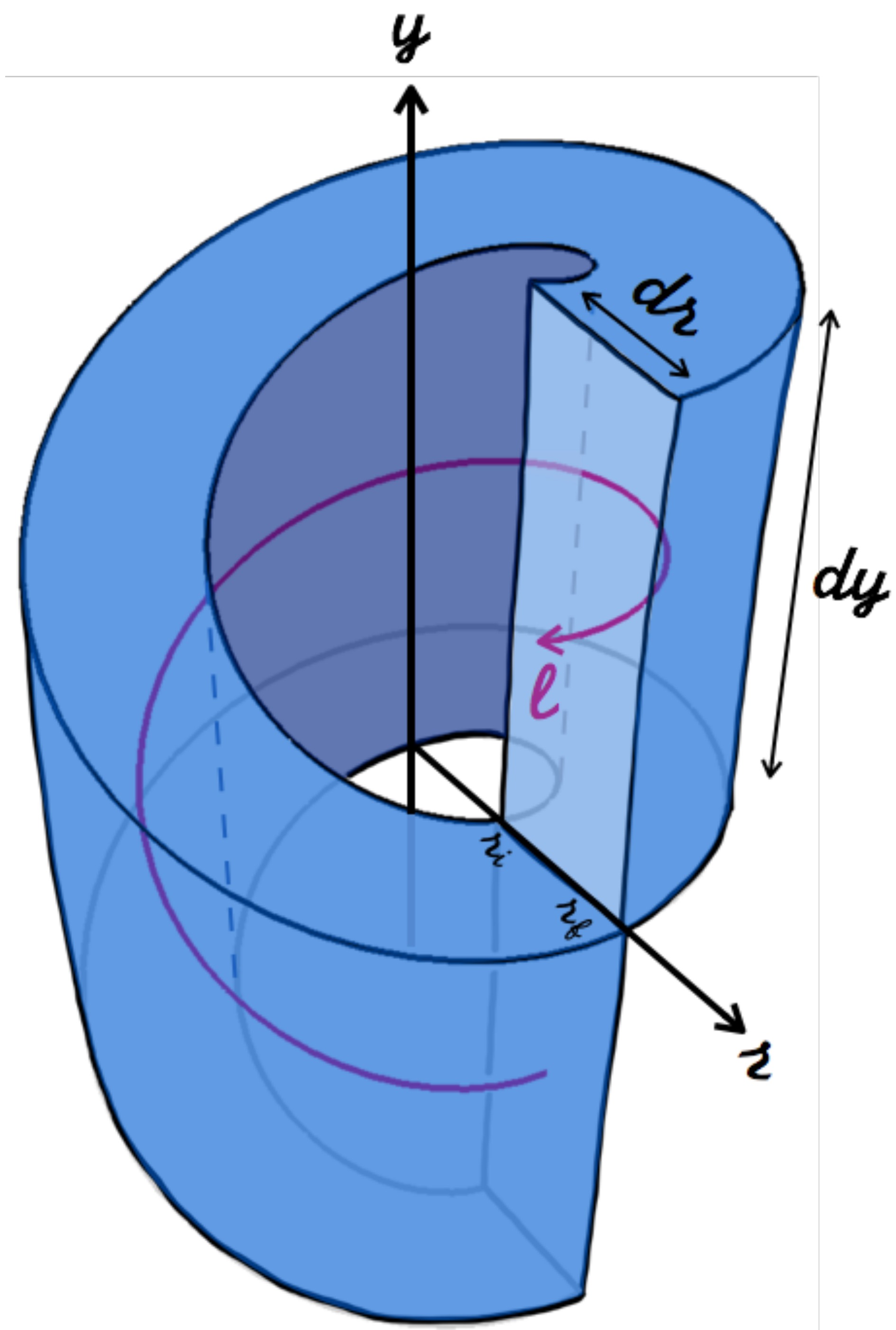
Hit-Track Correlation

- The only correlation used in track finding is the [shortest distance from point to helix] from two-axis in plane, perpendicular to track propagation direction.

- Distance from point to helix is known as numerical problem, equivalent to solving Kepler's equation using Newton's method.
  - Computing the distance from a point to a helix and solving Kepler's equation - Nuclear Instruments and Methods in Physics Research A 598 (2009) 788-794)

- Rather than choosing numerical method, the problem choosen to solving [shortest distance from point to straight line].

Real Space

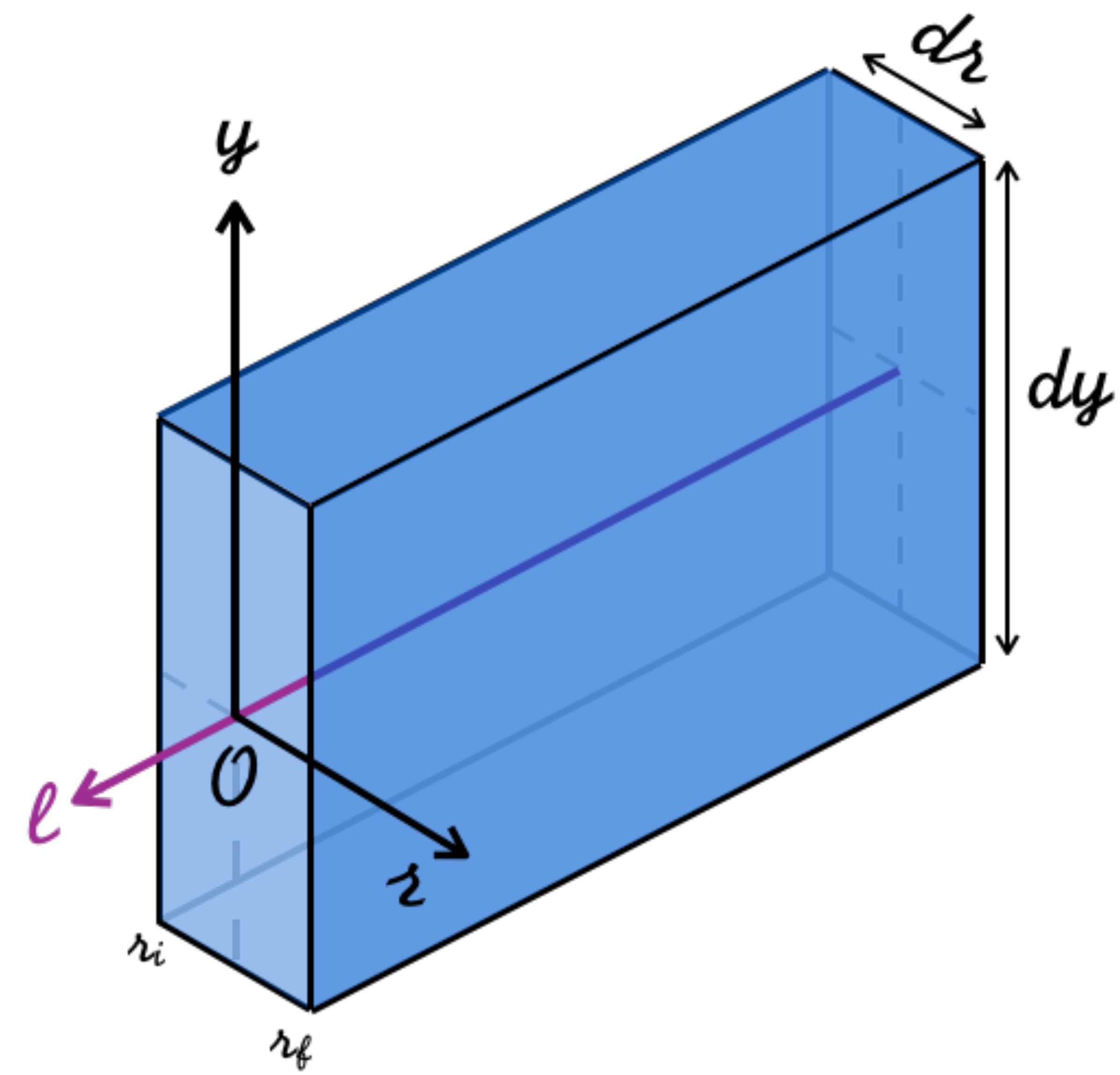


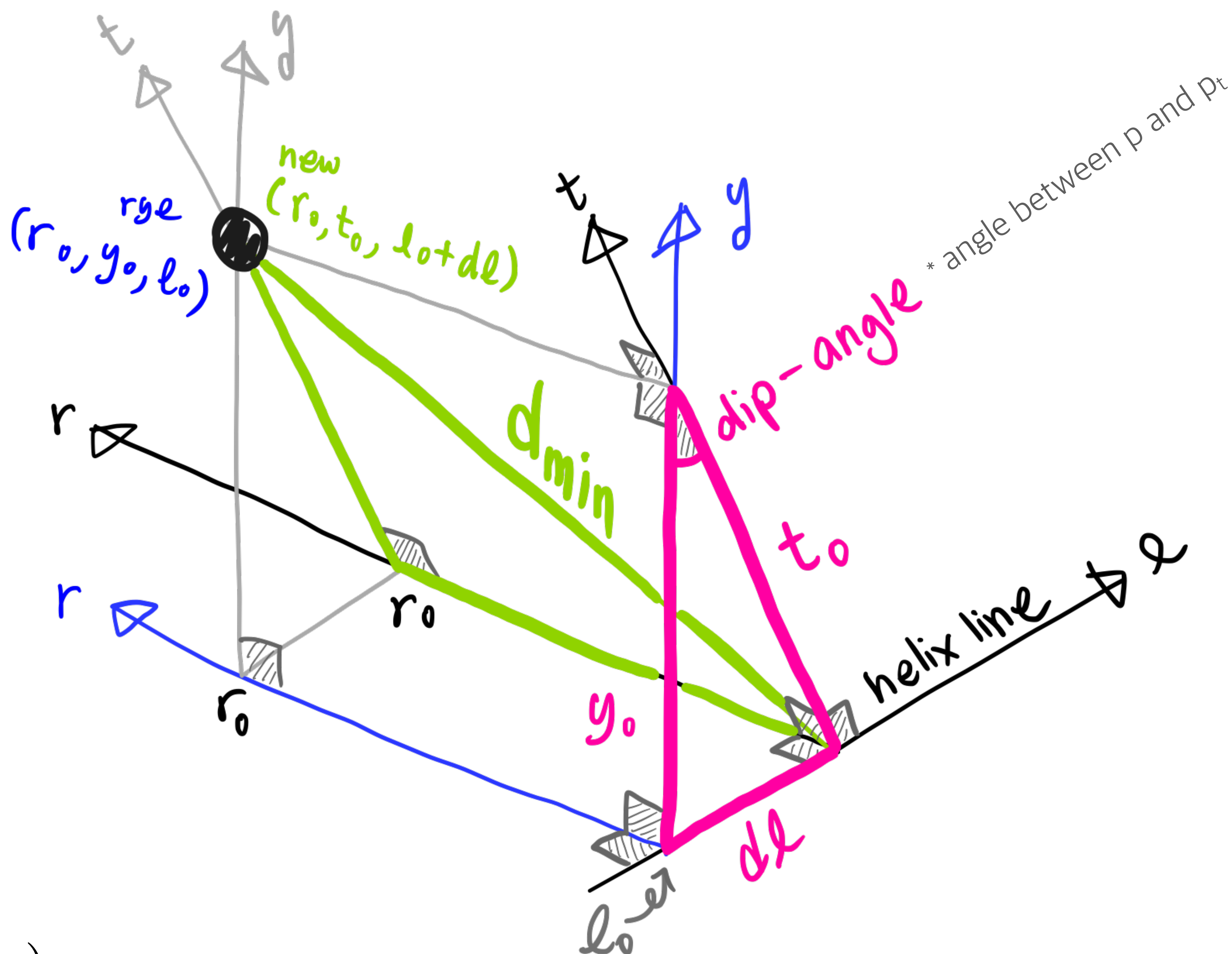
$0 < \alpha < \pi$

$r_i < r < r_f$

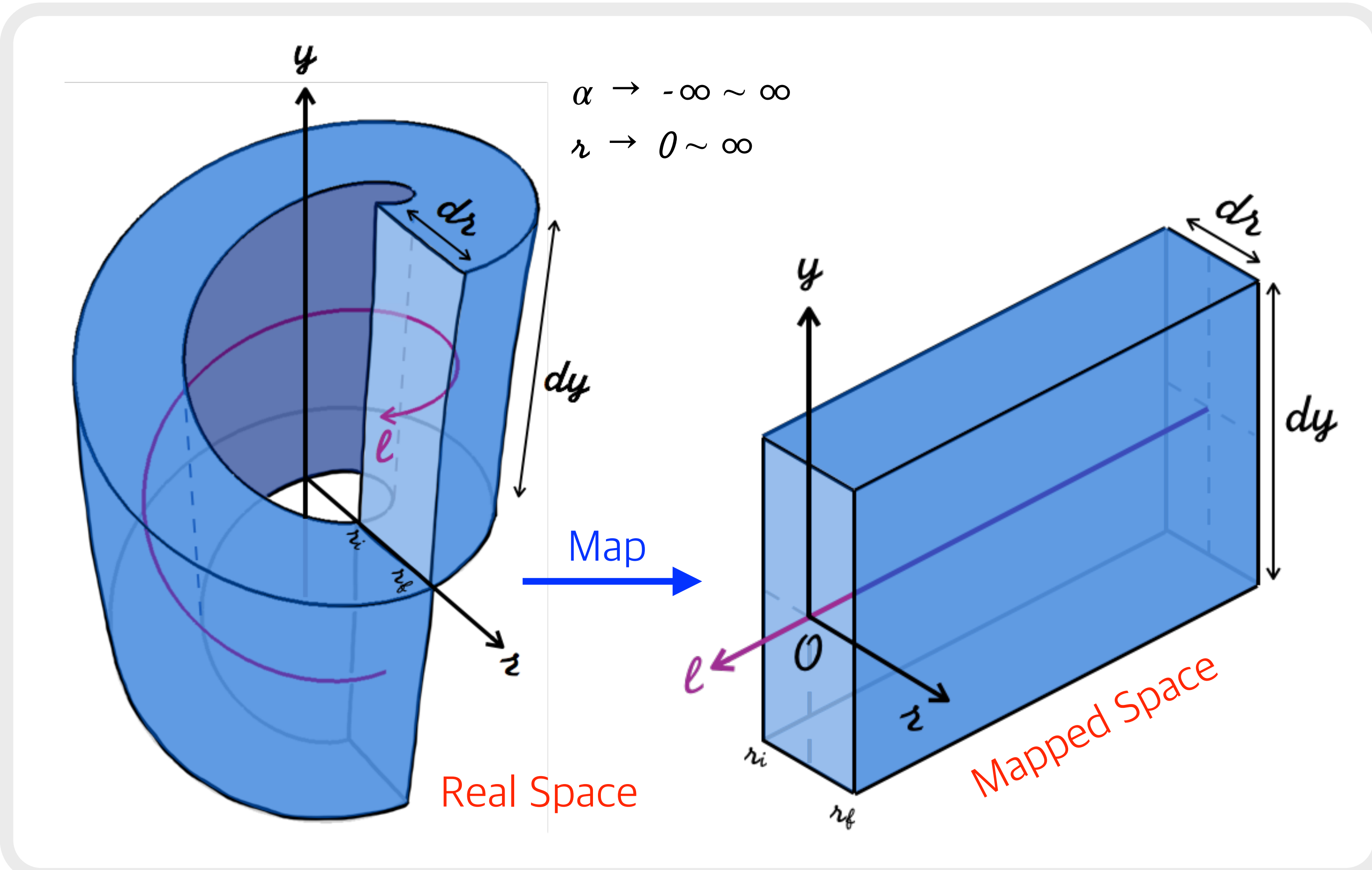
* y-length of helix propagated in one period,

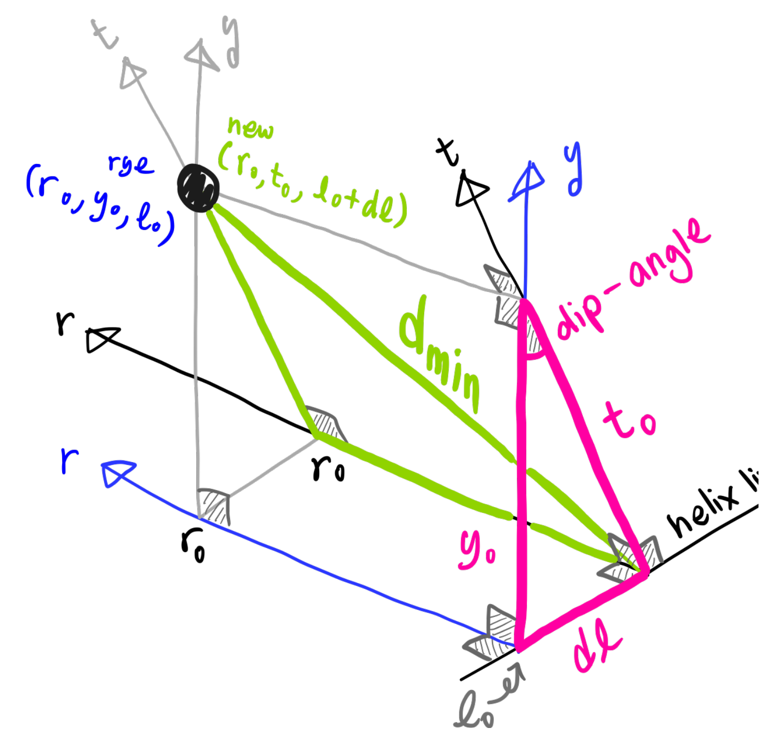
Mapped Space



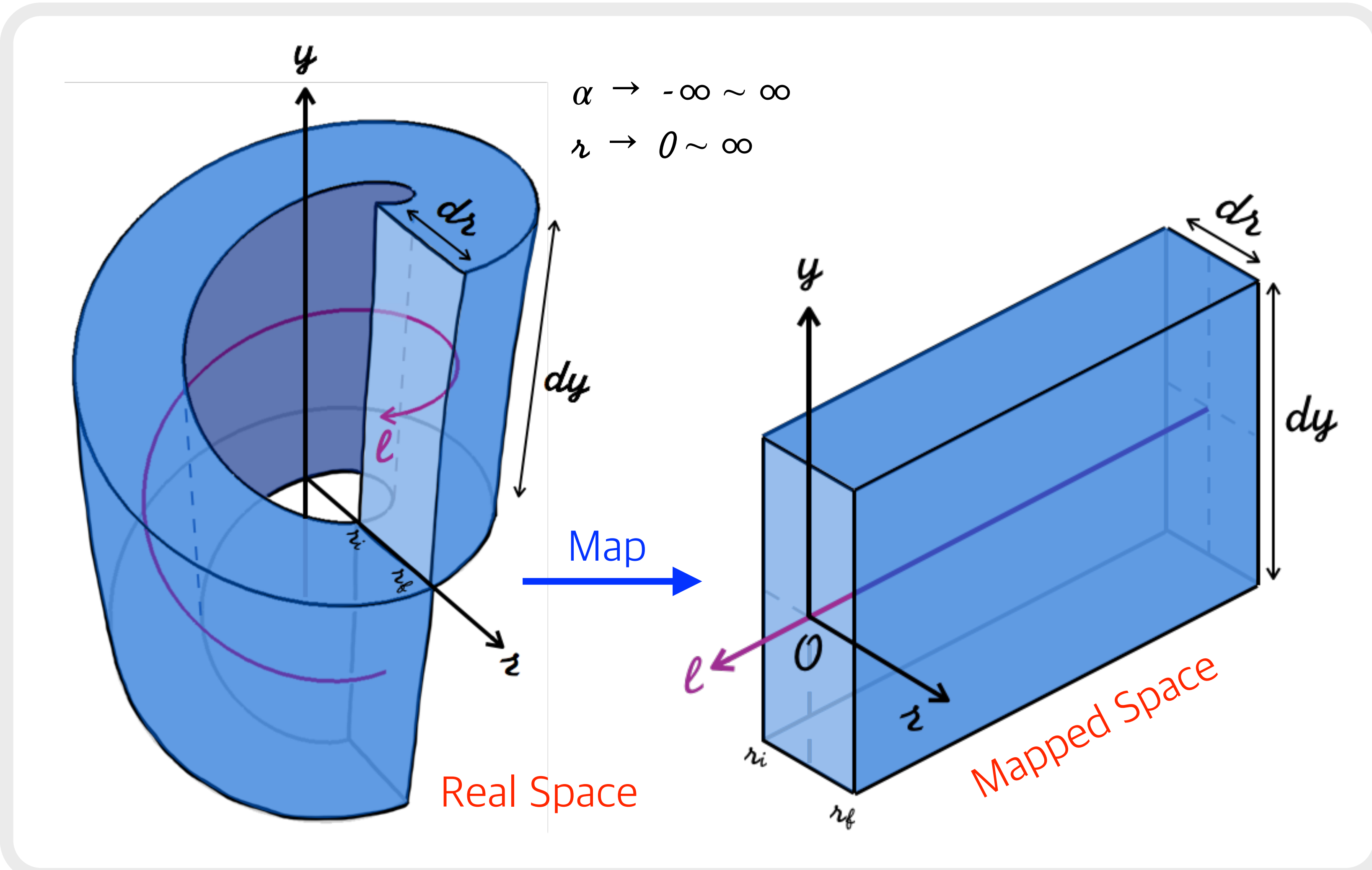


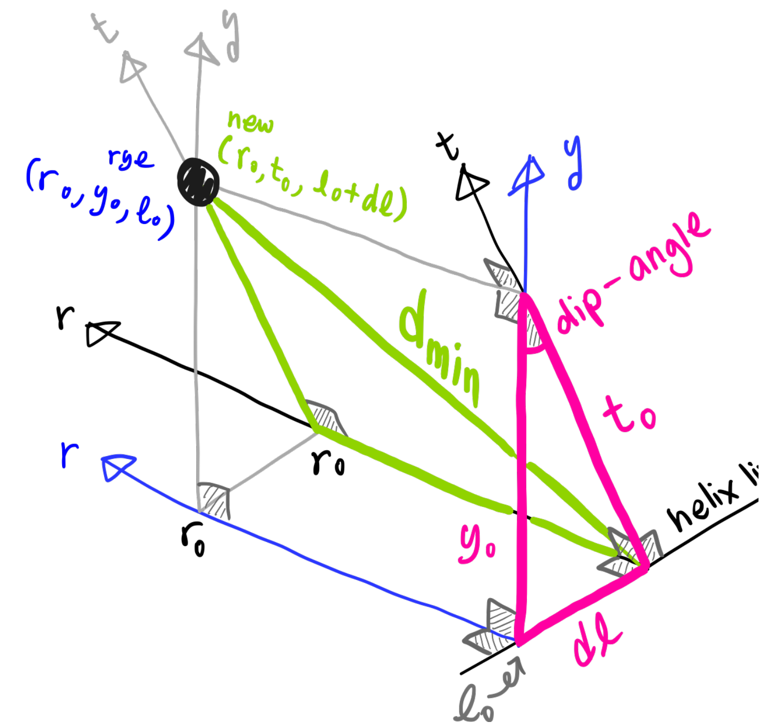
$$r' = r_0$$
$$t' = y_0 \cos(dip)$$
$$\ell' = \ell_0 + helicity \times y_0 \sin(dip)$$





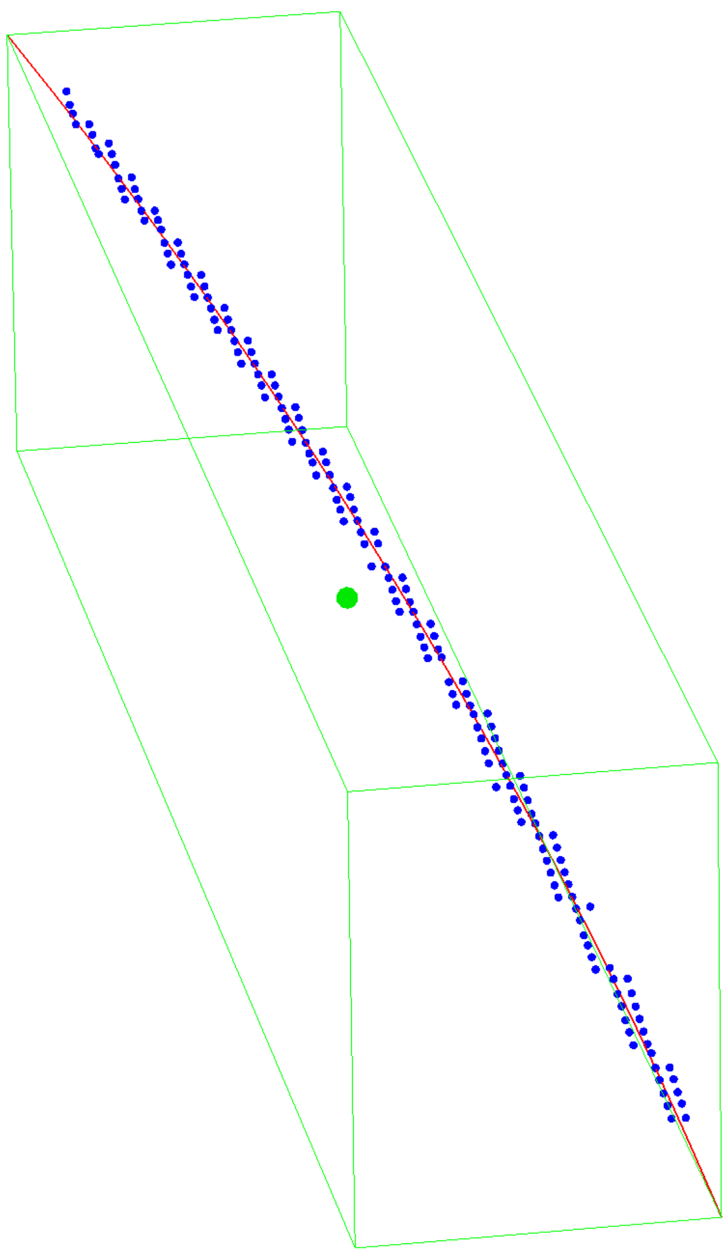
- Map space by straightening curled space : l-axis(helix line) becomes straight line. (x, y, z) → (r, t, ℓ)
- Origin in the mapped space is sitting on the l-axis, where α-angle becomes 0.
- Plane defined by r and t is flat in mapped space but not in real space which tells [shortest distance from point to helix] is not same as [shortest distance from point to line in mapped space].

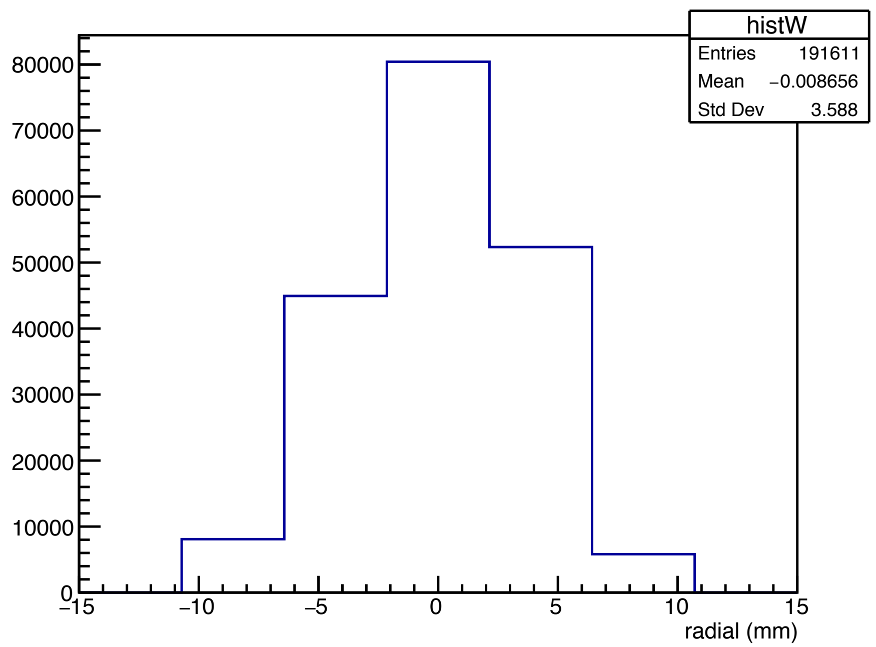


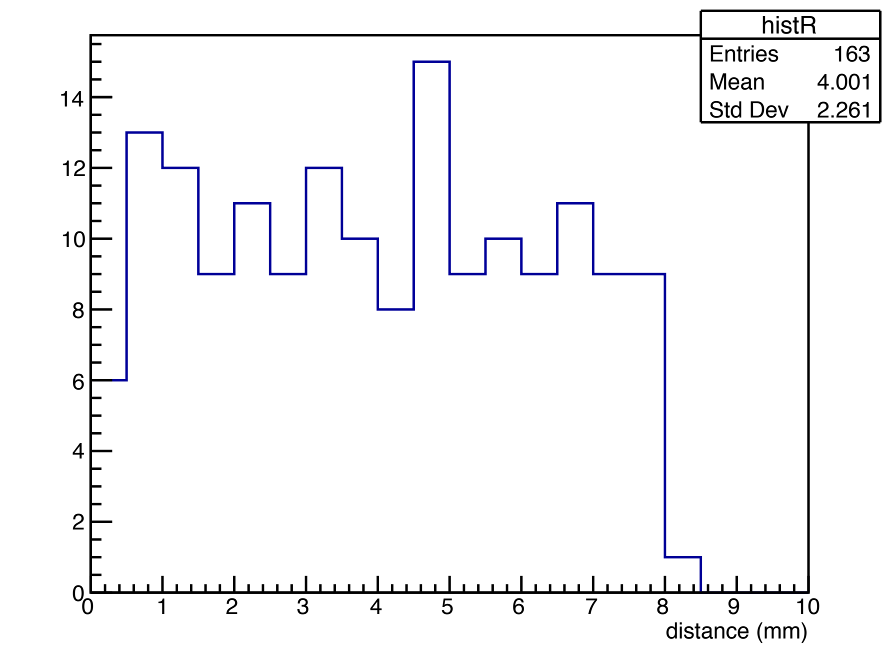


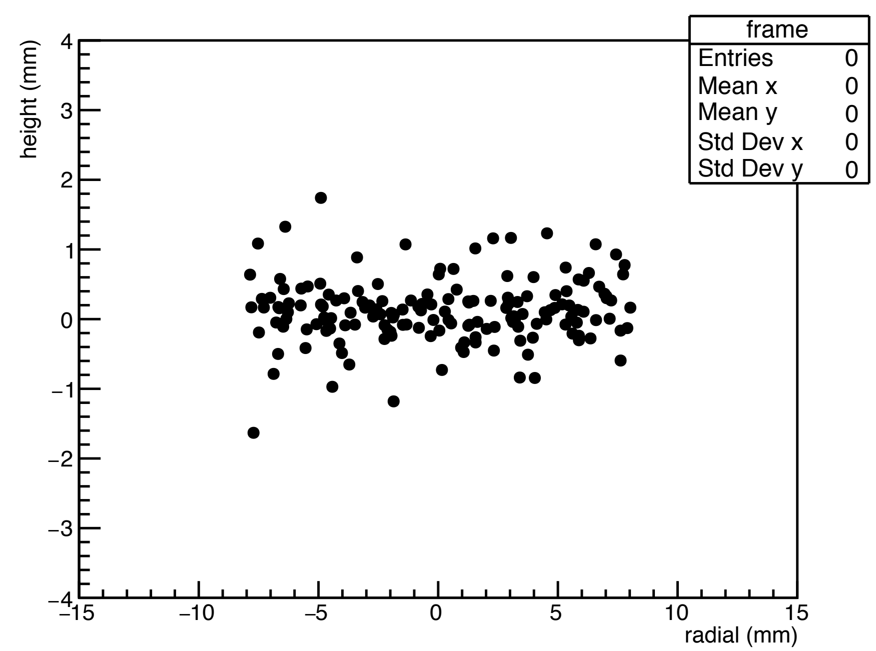
- Advantage by using position of point in new system (r, t, ℓ) is 1) distance in width-direction(r), 2) distance in heigth direction(t) and 3) length along the track(ℓ).
- It is possible to use this mapping instead of calculating [distance from point to helix] because hit-track correlation cut parameters are self-updated from mapped system which makes no difference in track finding.

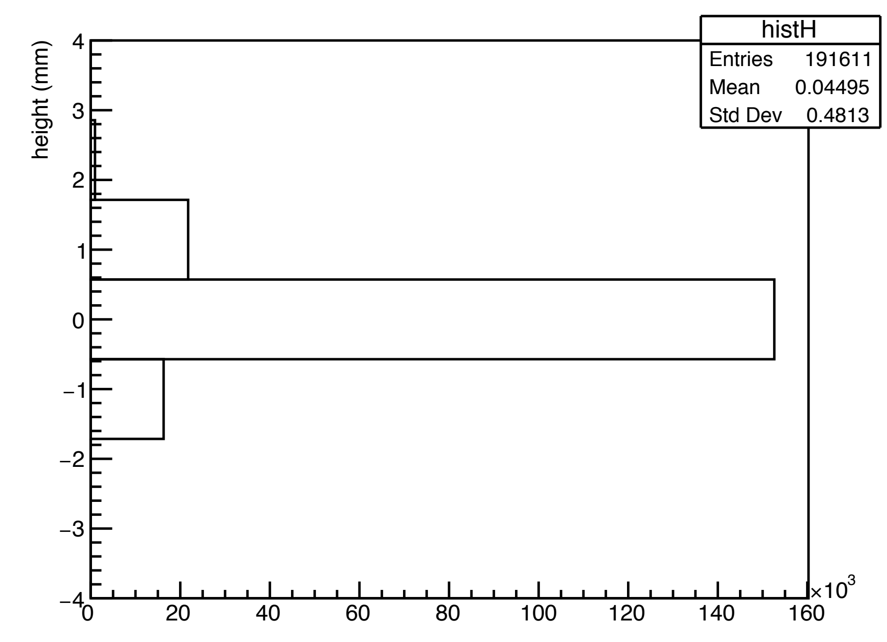
Hit Map Quality Check



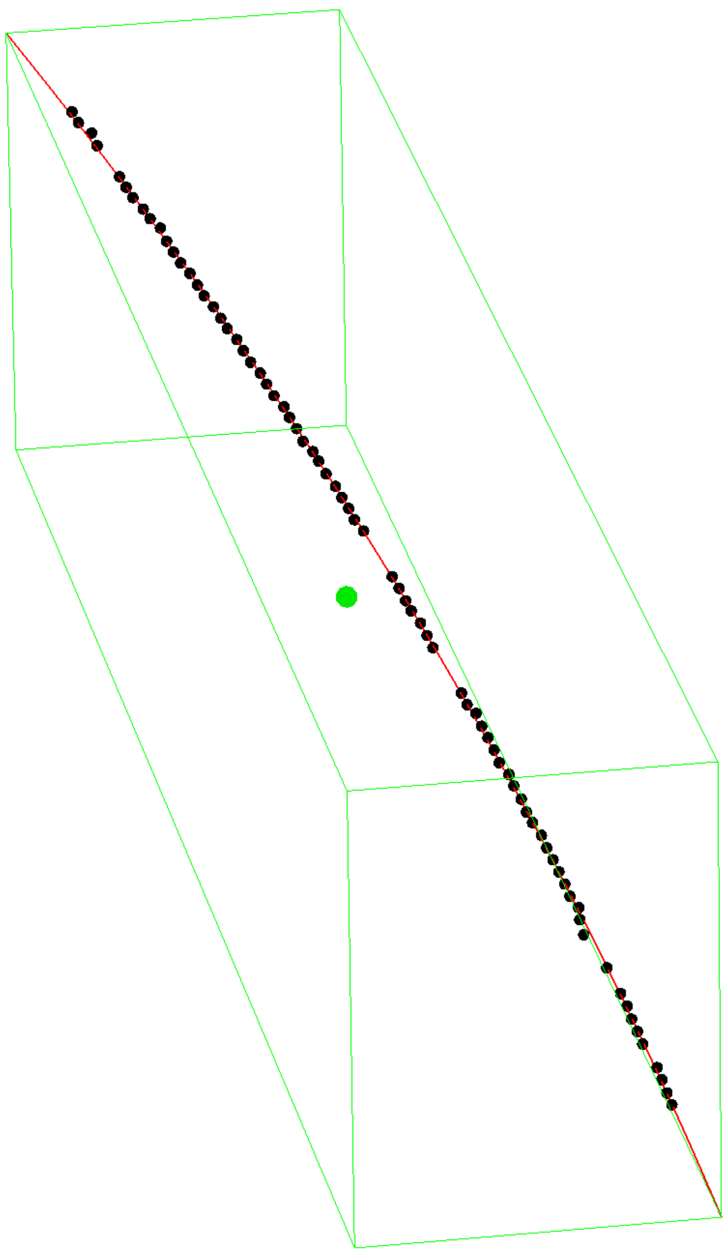


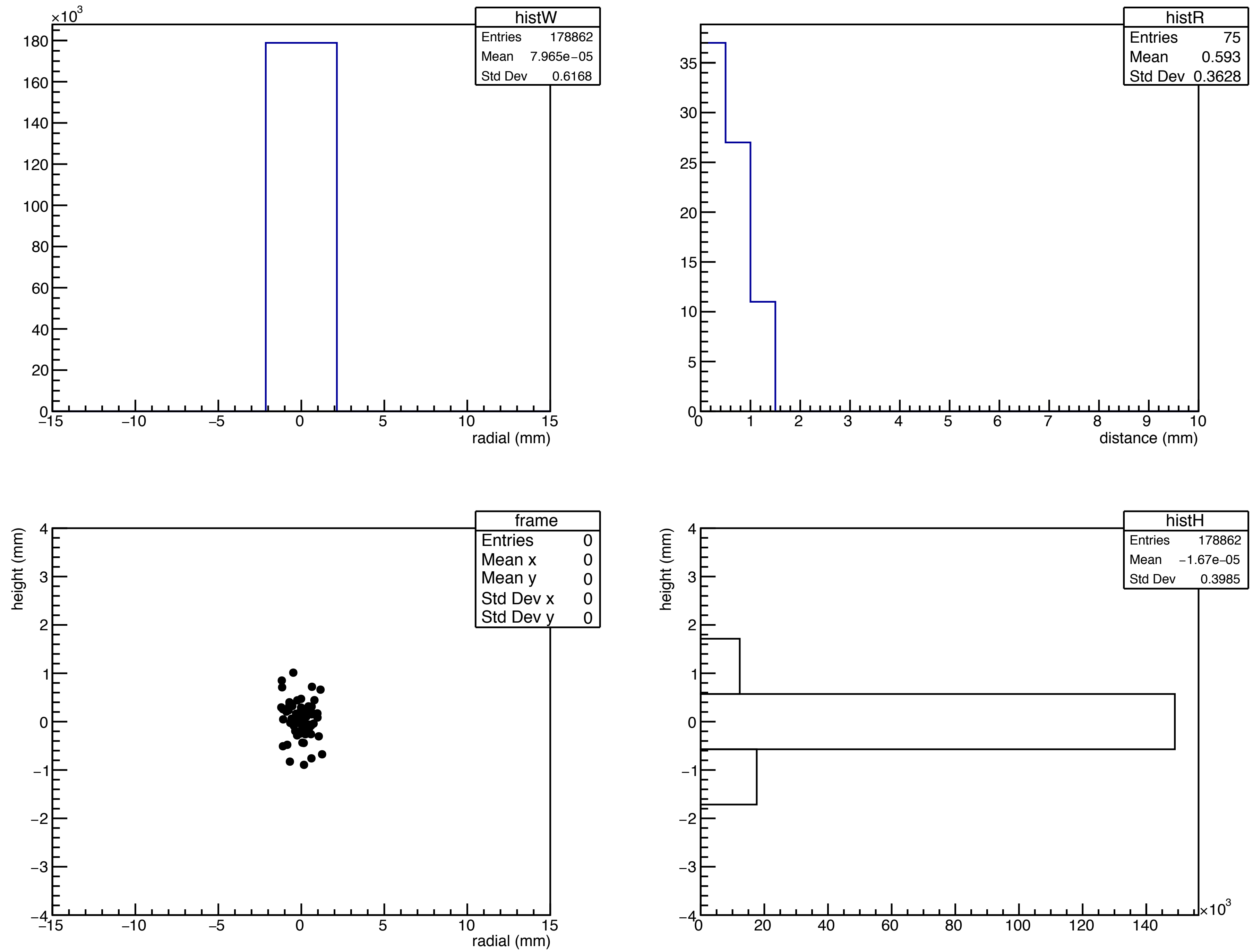




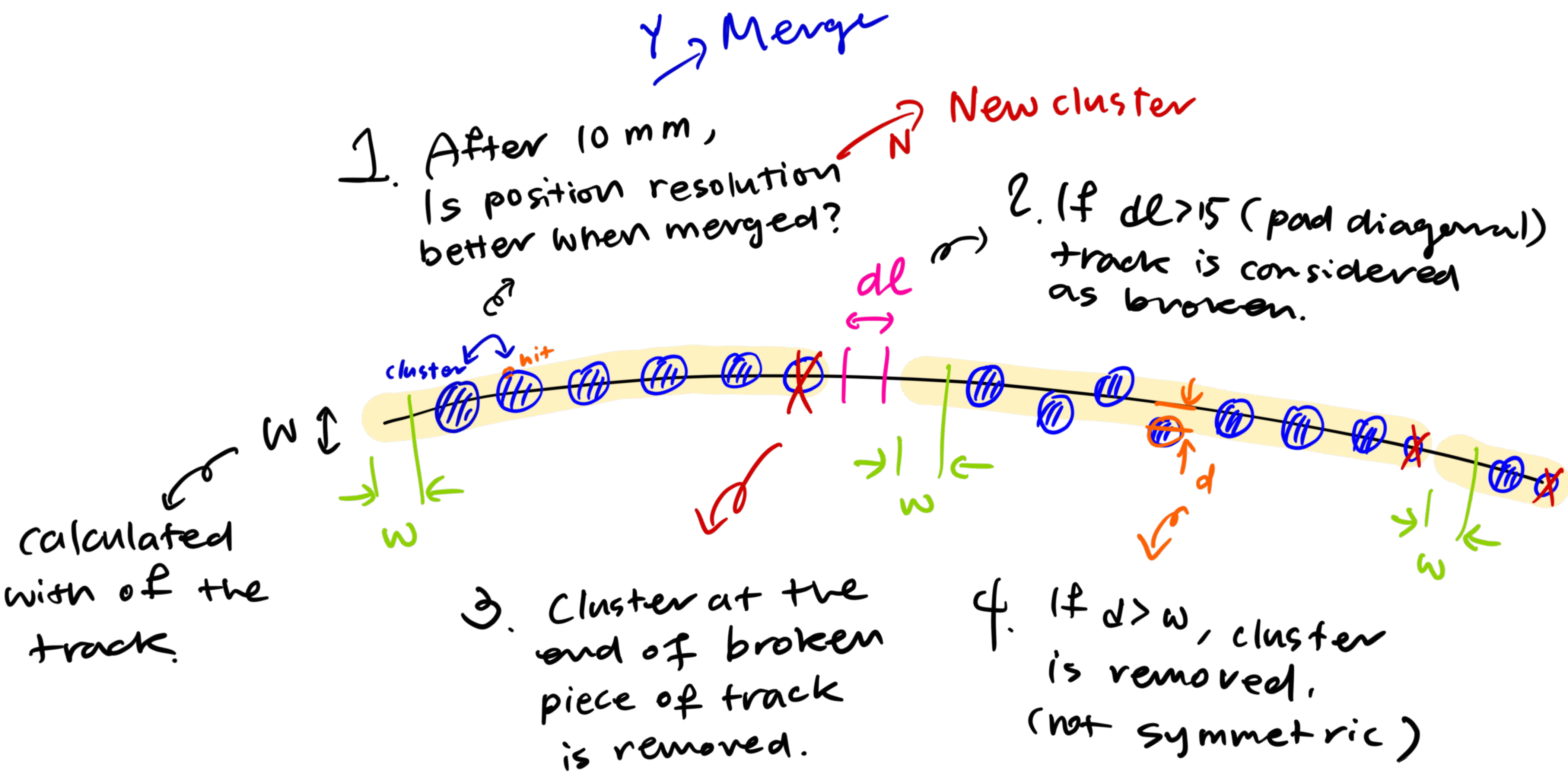


Hit-Cluster Map Quality Check

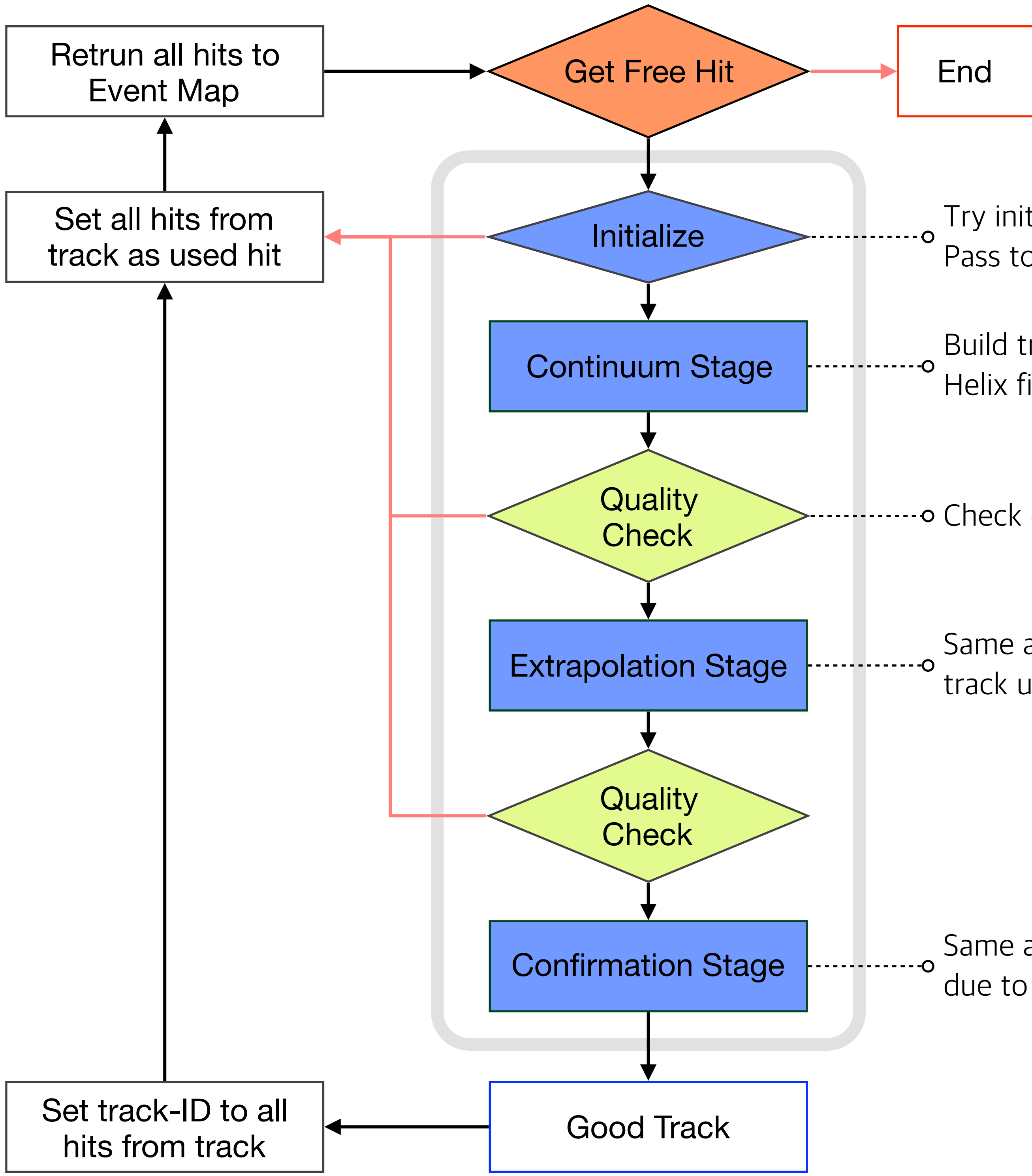




Hit-Clustering



Track Finding Algorithm



Try initialization(build track) by adding only the neighboring pad hits.
Pass to next stage if hit distribution is big enough to fit as helix.

Build track by adding only the neighboring pad hits.
Helix fitting is done to see the mathching quality of with the track.

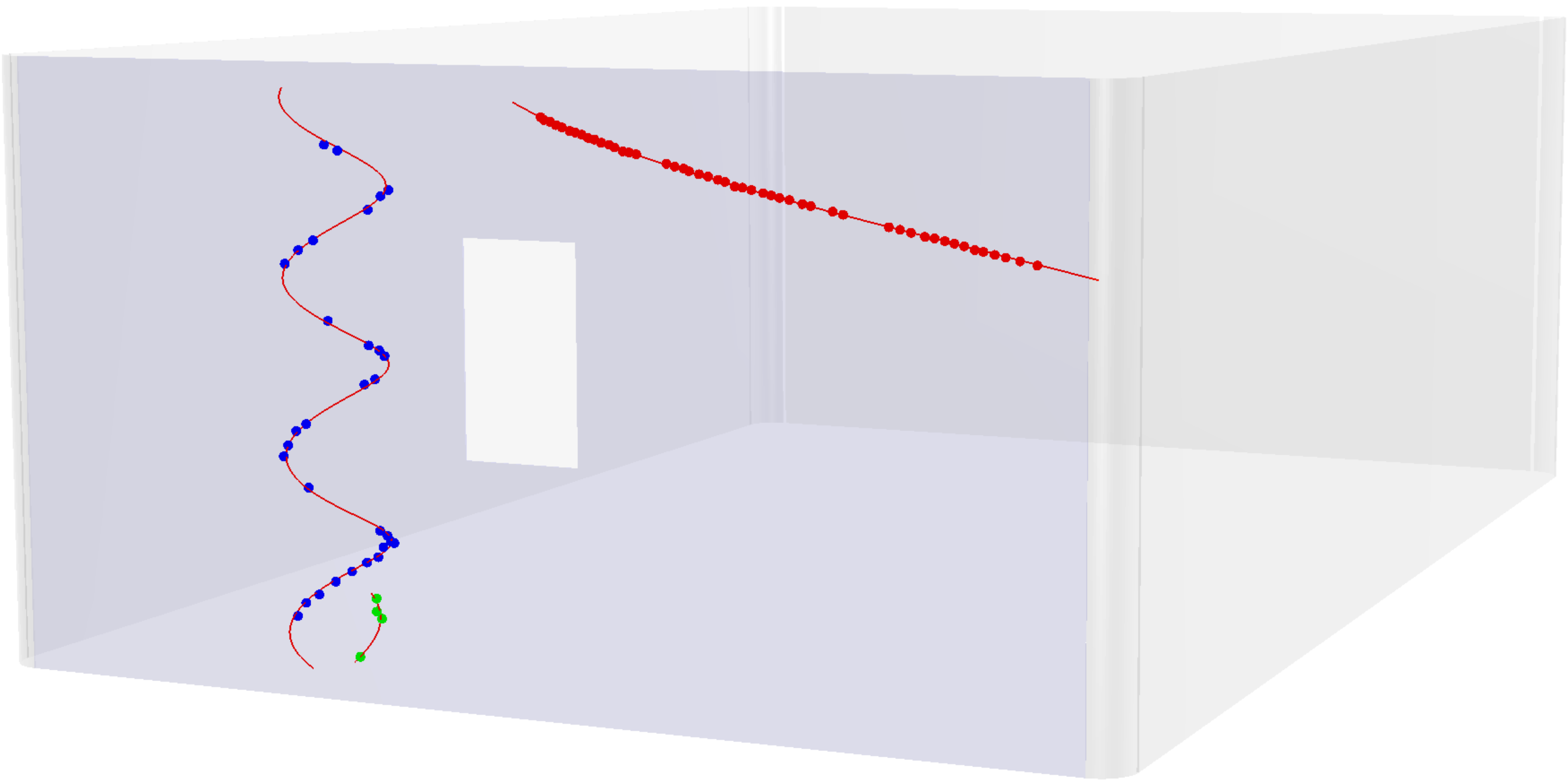
Check quality(continuity, length) of the track.

Same as continuum stage, but hits are gethered by extrapolating track until the track reaches end of TPC.

Same as extrapolation stage, but find or remove hits due to with stable fitting of track.

- **Free hit** : Hit that was never used to build track.
- **Used hit** : Hit that was used to build once or more.

Helix!



Summary

- New tracking is developed and being tested for first release.
- New tracking has advantage in
  - ☑ Full control of the code.
  - ☑ Build full track one by one.
  - ☑ Helix to straight line map.
  - ☑ Use advantage of width of the track comming from electron dispersion.
  - ☑ Use self-update parameters.
    - Riemann sphere position and radius.
    - Proximity cut.
  - ☑ Clustering for Genfit.
- We can find helix.