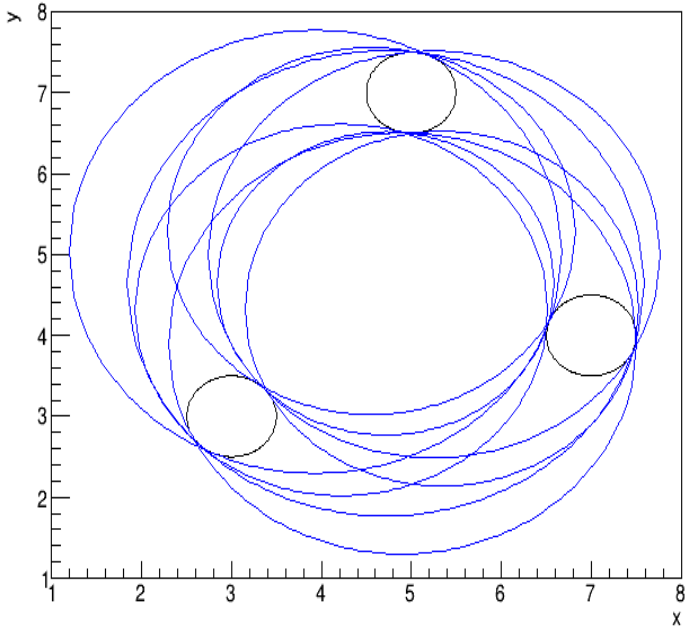
APOLLONIUS TRIPLET TRACK FINDER
A TRACK FINDER FOR SECONDARIES

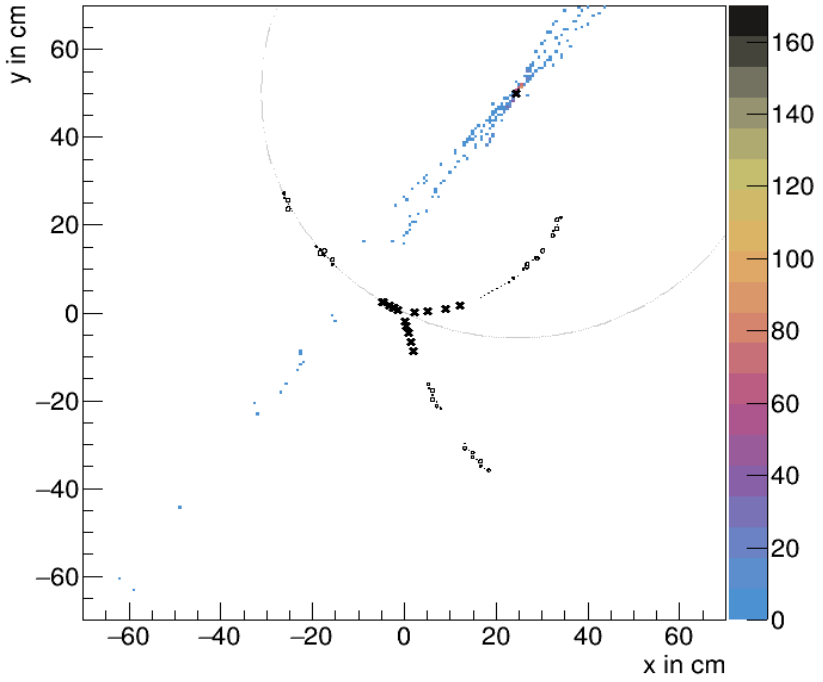
26.10.2021 I ANNA ALICKE | PANDA COLLABORATION MEETING

MOTIVATION

- Interesting hyperon decays often lead to secondary particles (e.g. Λ-decays)
- HoughTrackFinder could be extended to secondaries

Apollonius is not restricted to IP





MOTIVATION

- Interesting hyperon decays often lead to secondary particles (e.g. Λ-decays)
- HoughTrackFinder could be extended to secondaries
  - Huge combinatorics: increase from $\binom{n}{2}$ to $\binom{n}{3}$
  - 3D-Hough space is needed for circle parameters (x, y, r)

→ Slow due to high combinatorics

→ Slow because of 3D-Maximum finding

Solvable by GPU

→ High memory consumption due to sufficient high resolution in 3D-Hough space

Strong bottleneck to use the GPU efficiently

MOTIVATION

- Interesting hyperon decays often lead to secondary particles (e.g. Λ-decays)
- HoughTrackFinder could be extended to secondaries
  - Huge combinatorics: increase from $\binom{n}{2}$ to $\binom{n}{3}$
  - 3D-Hough space is needed for circle parameters (x, y, r)

→ Slow due to high combinatorics

→ Slow because of 3D-Maximum finding

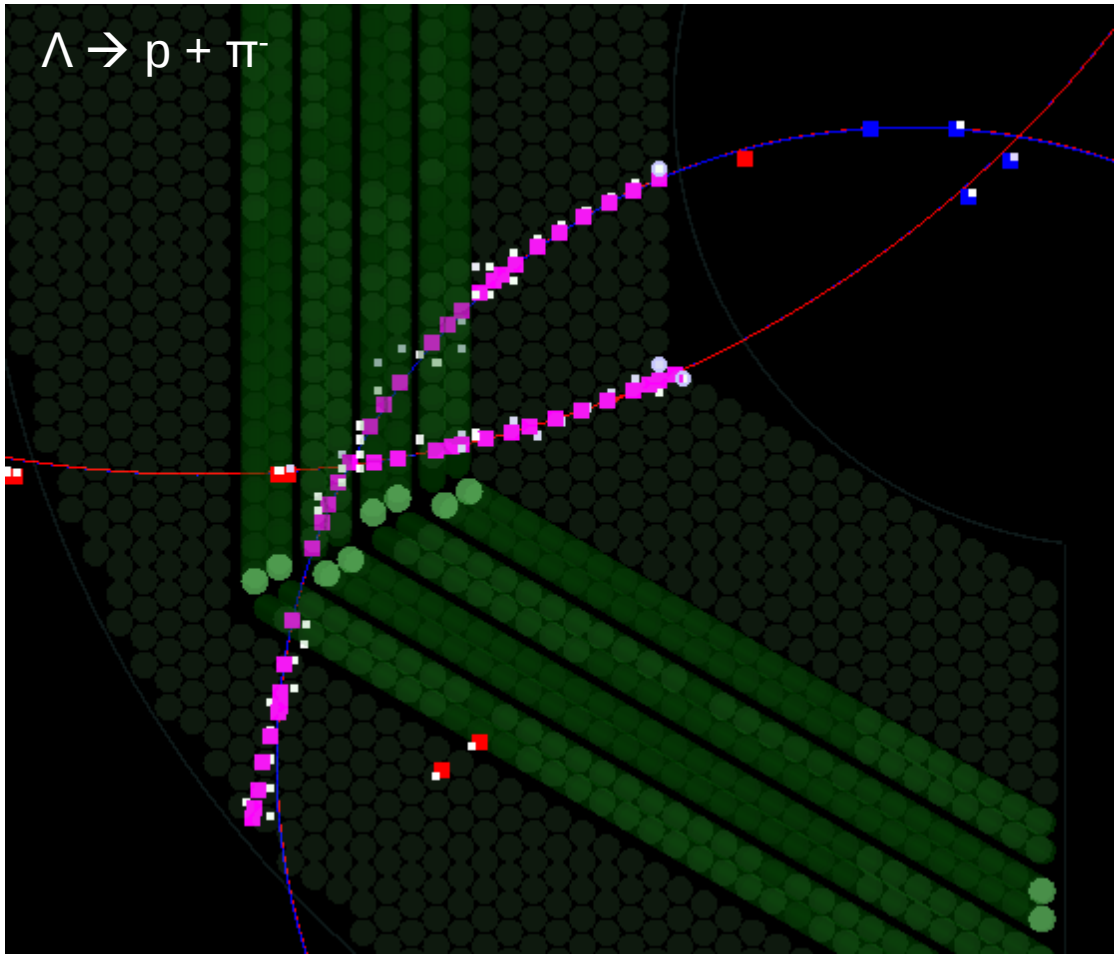
Solvable by GPU

→ High memory consumption due to sufficient high resolution in 3D-Hough space

Strong bottleneck to use the GPU efficiently

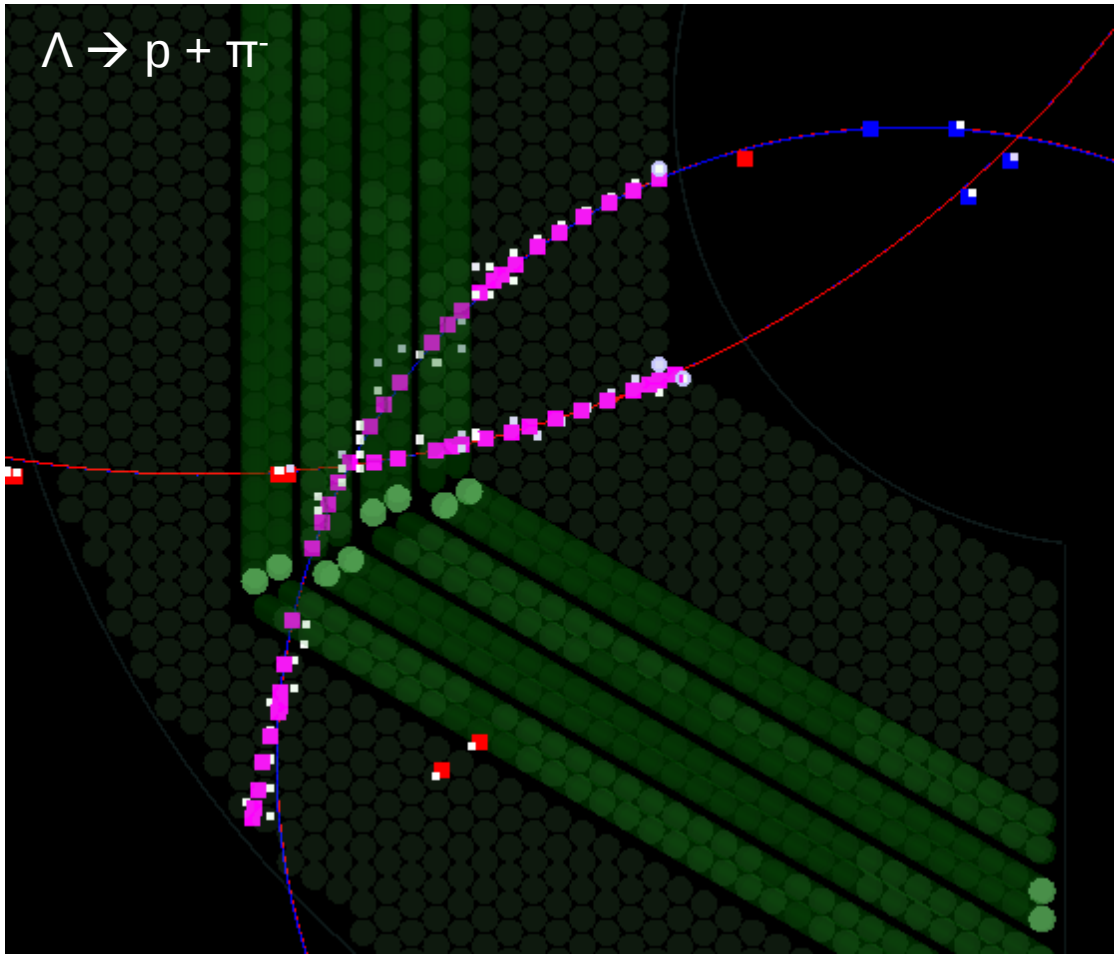
➔ Can we find another idea to use the Apollonius calculation more efficiently?

APOLLONIUS TRIPLET TRACK FINDER



- Basic idea
  - Select three STT hits
  - Calculate Apollonius circles
  - Add other STT hits which are close to circles
  - Select best solution(s)

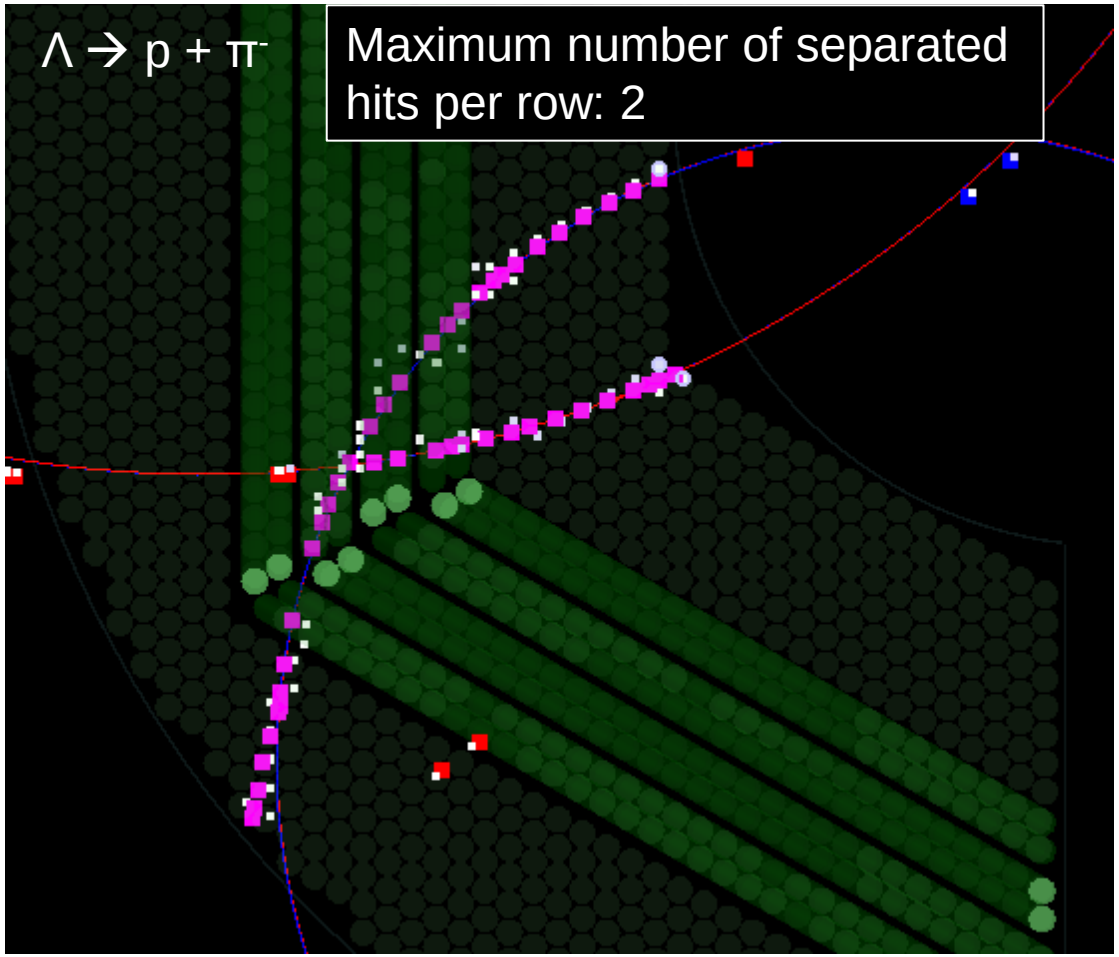
APOLLONIUS TRIPLET TRACK FINDER



- Basic idea
  - Select three STT hits
  - Calculate Apollonius circles
  - Add other STT hits which are close to circles
  - Select best solution(s)

- First Step „Select three STT hits“ already most important step:
  - Combine all STT hits are too many combinations to be fast → Selection has to be done
  - If selection is too tight efficiency will drop

APOLLONIUS TRIPLET TRACK FINDER

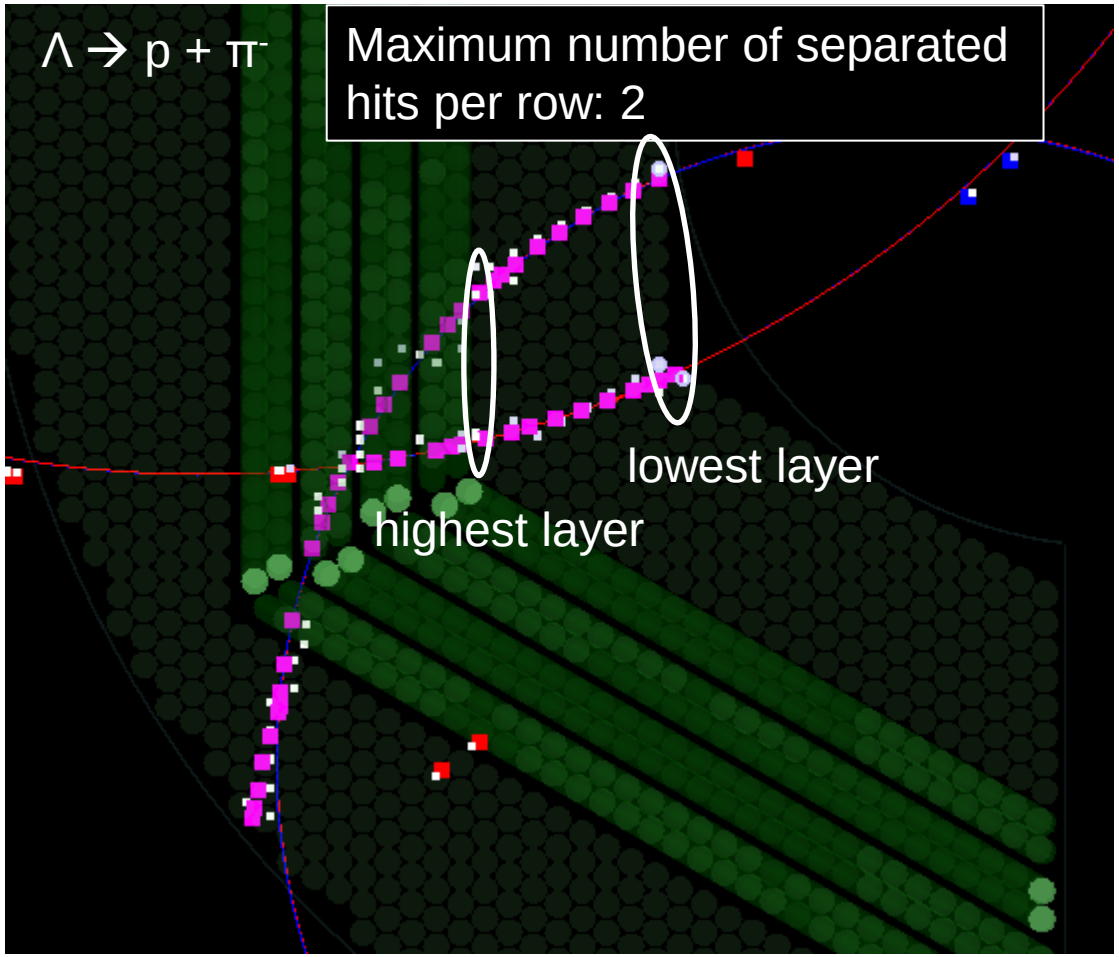


- Basic idea
  - Select three STT hits
  - Calculate Apollonius circles
  - Add other STT hits which are close to circles
  - Select best solution(s)

- First Step „Select three STT hits“ already most important step:
  - Combine all STT hits are too many combinations to be fast → Selection has to be done
  - If selection is too tight efficiency will drop

➔ How to do the selection?
  - Cellular Automaton / Phi Selector to find ranges of connected STT hits
  - Analyse STT hits by separated hits per row

APOLLONIUS TRIPLET TRACK FINDER

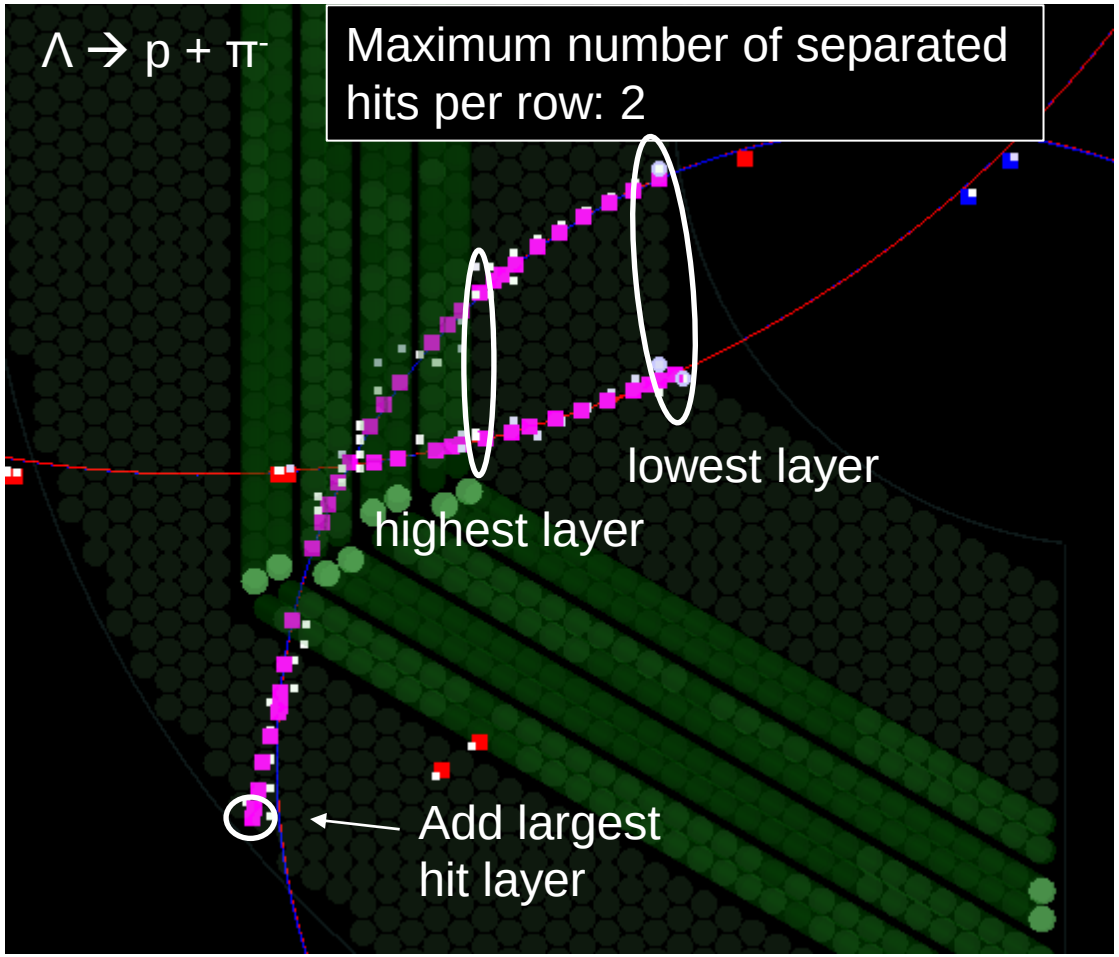


- Basic idea
  - Select three STT hits
  - Calculate Apollonius circles
  - Add other STT hits which are close to circles
  - Select best solution(s)

- First Step „Select three STT hits“ already most important step:
  - Combine all STT hits are too many combinations to be fast → Selection has to be done
  - If selection is too tight efficiency will drop

➔ How to do the selection?
  - Cellular Automaton / Phi Selector to find ranges of connected STT hits
  - Analyse STT hits by separated hits per row
  - Select all hits from „inner“, „outer“ and „mid“ row

APOLLONIUS TRIPLET TRACK FINDER

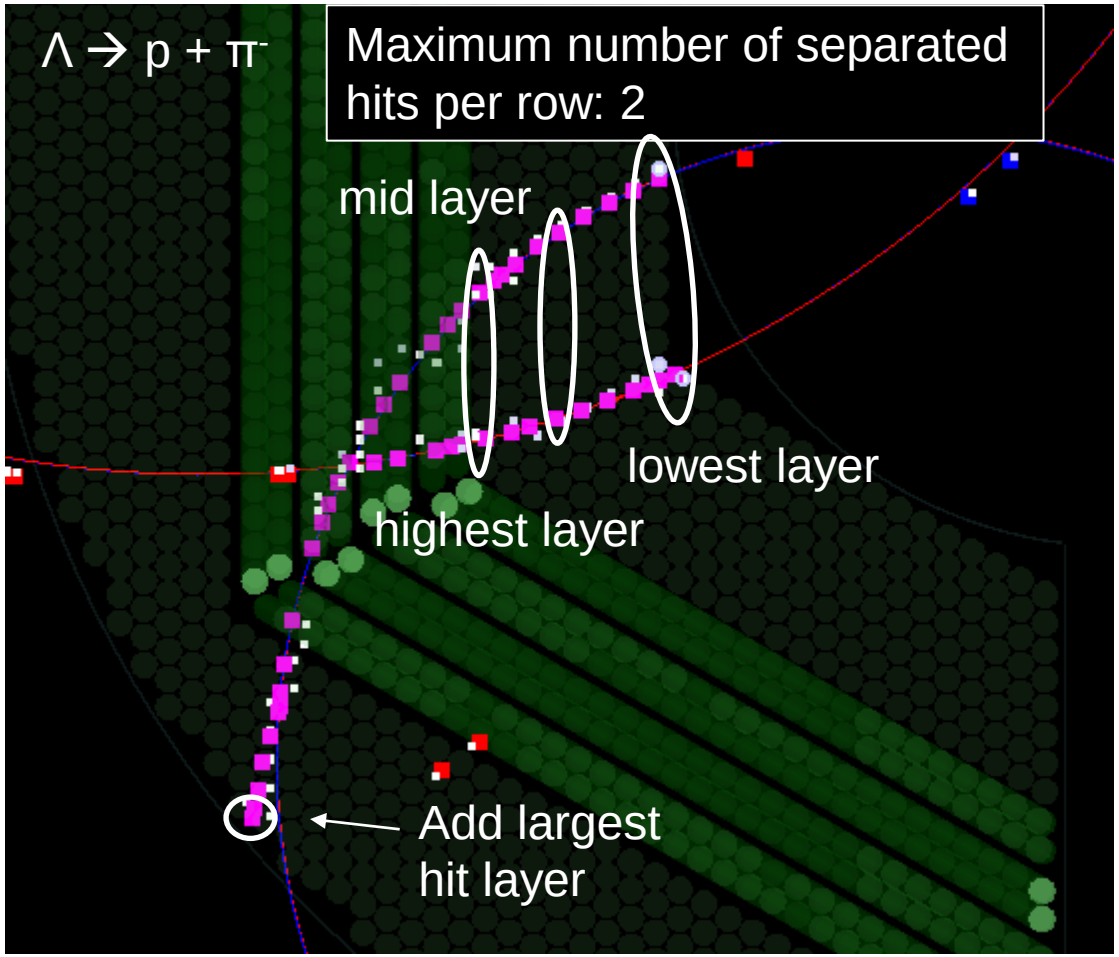


- Basic idea
  - Select three STT hits
  - Calculate Apollonius circles
  - Add other STT hits which are close to circles
  - Select best solution(s)

- First Step „Select three STT hits“ already most important step:
  - Combine all STT hits are too many combinations to be fast → Selection has to be done
  - If selection is too tight efficiency will drop

➔ How to do the selection?
  - Cellular Automaton / Phi Selector to find ranges of connected STT hits
  - Analyse STT hits by separated hits per row
  - Select all hits from „inner“, „outer“ and „mid“ row

APOLLONIUS TRIPLET TRACK FINDER

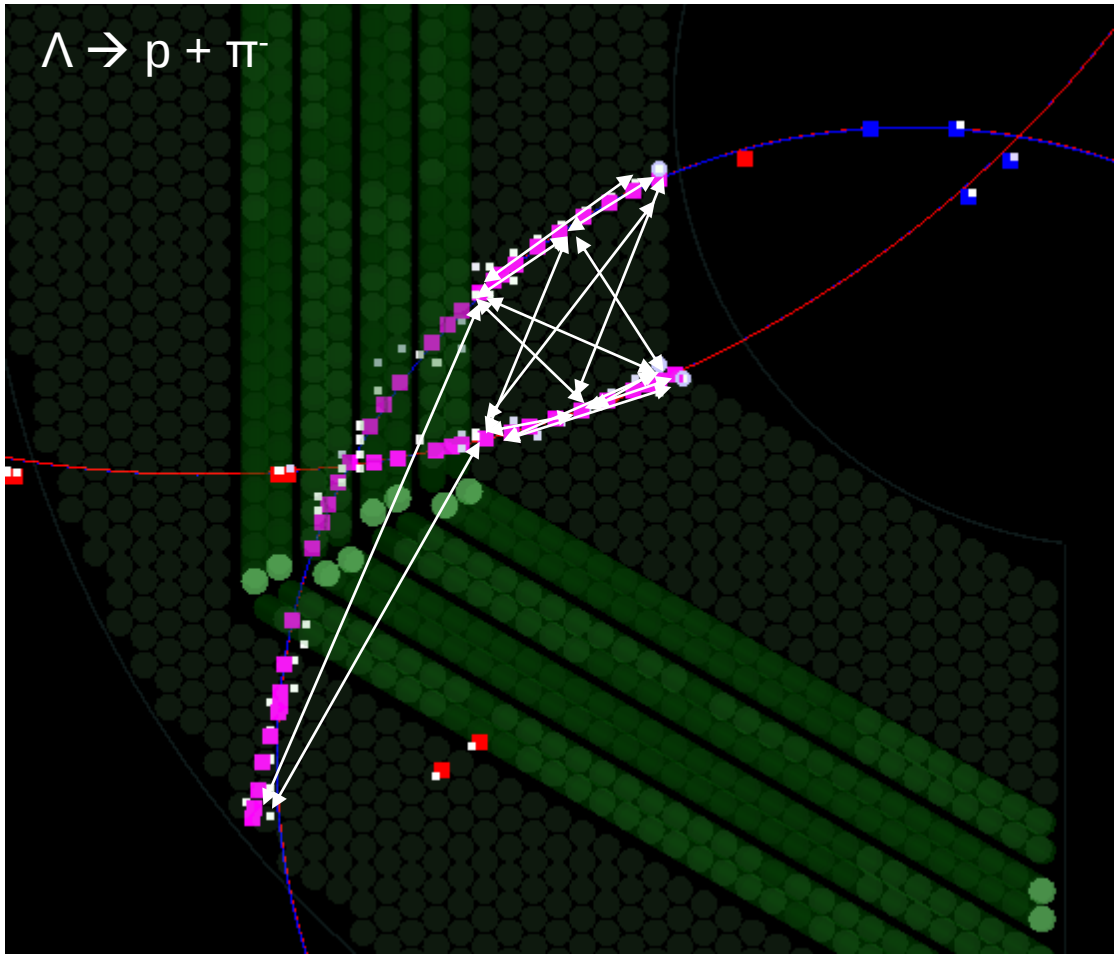


- Basic idea
  - Select three STT hits
  - Calculate Apollonius circles
  - Add other STT hits which are close to circles
  - Select best solution(s)

- First Step „Select three STT hits“ already most important step:
  - Combine all STT hits are too many combinations to be fast → Selection has to be done
  - If selection is too tight efficiency will drop

➔ How to do the selection?
  - Cellular Automaton / Phi Selector to find ranges of connected STT hits
  - Analyse STT hits by separated hits per row
  - Select all hits from „inner“, „outer“ and „mid“ row

APOLLONIUS TRIPLET TRACK FINDER



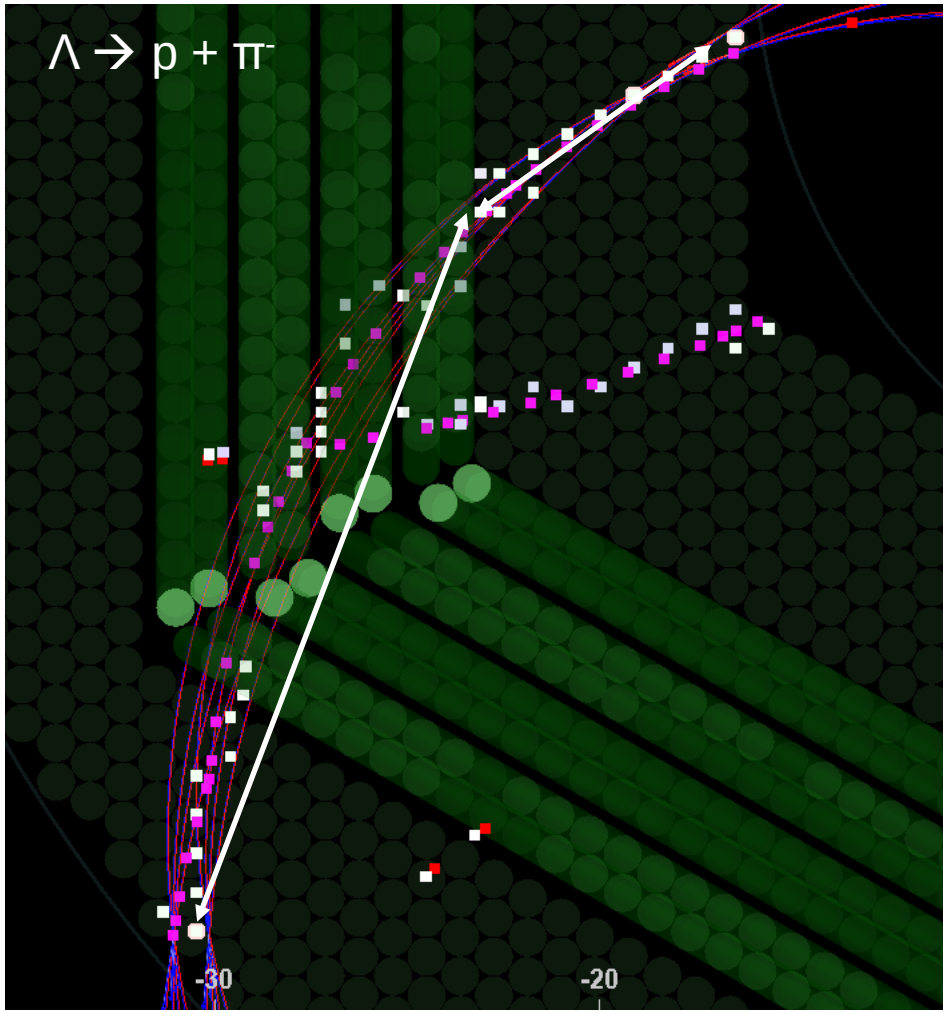
- Basic idea
  - Select three STT hits
  - Calculate Apollonius circles
  - Add other STT hits which are close to circles
  - Select best solution(s)

- First Step „Select three STT hits“ already most important step:
  - Combine all STT hits are too many combinations to be fast
    → Selection has to be done
  - If selection is too tight efficiency will drop

➔ How to do the selection?

- Create triplet from one hit out of each group

APOLLONIUS TRIPLET TRACK FINDER



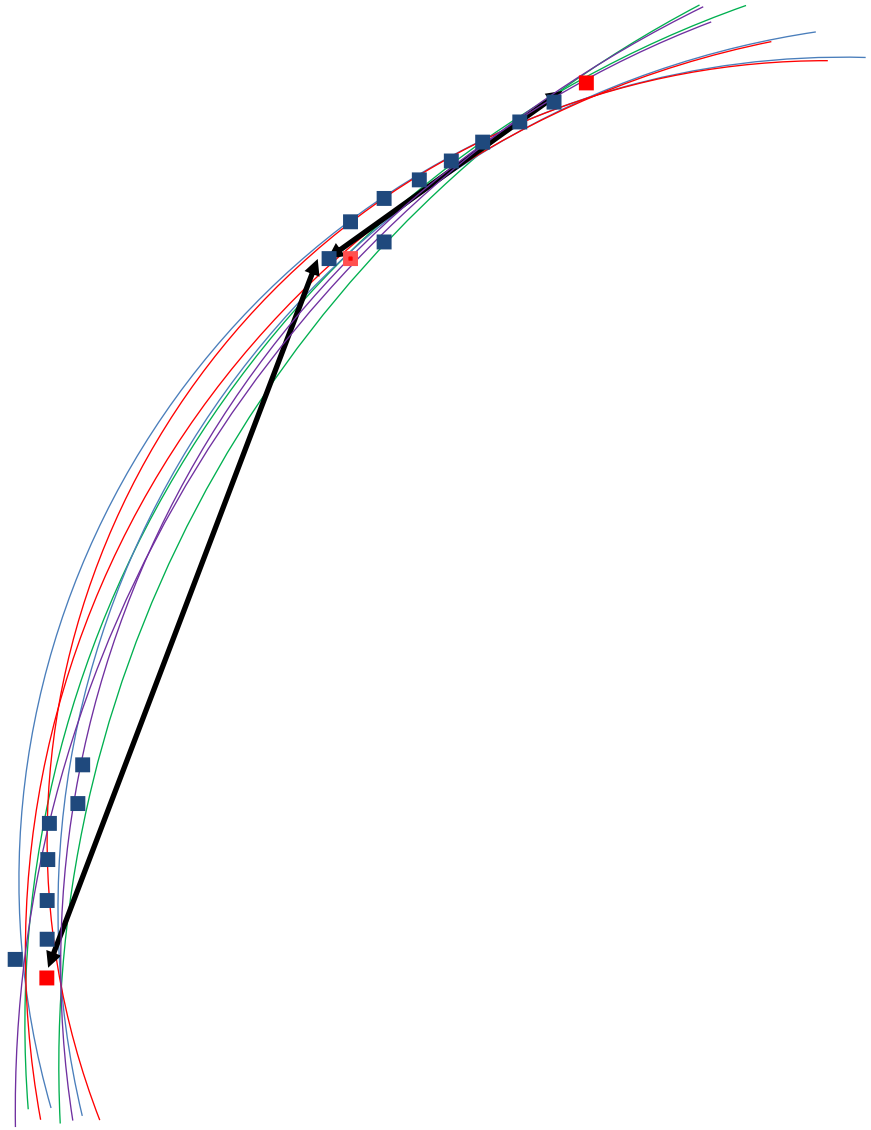
- Basic idea
  - Select three STT hits
  - Calculate Apollonius circles
  - Add other STT hits which are close to circles
  - Select best solution(s)

- First Step „Select three STT hits“ already most important step:
  - Combine all STT hits are too many combinations to be fast
    → Selection has to be done
  - If selection is too tight efficiency will drop

➔ How to do the selection?

- Create triplet from one hit out of each group
  - Calculate 8 Apollonius circles

APOLLONIUS TRIPLET TRACK FINDER



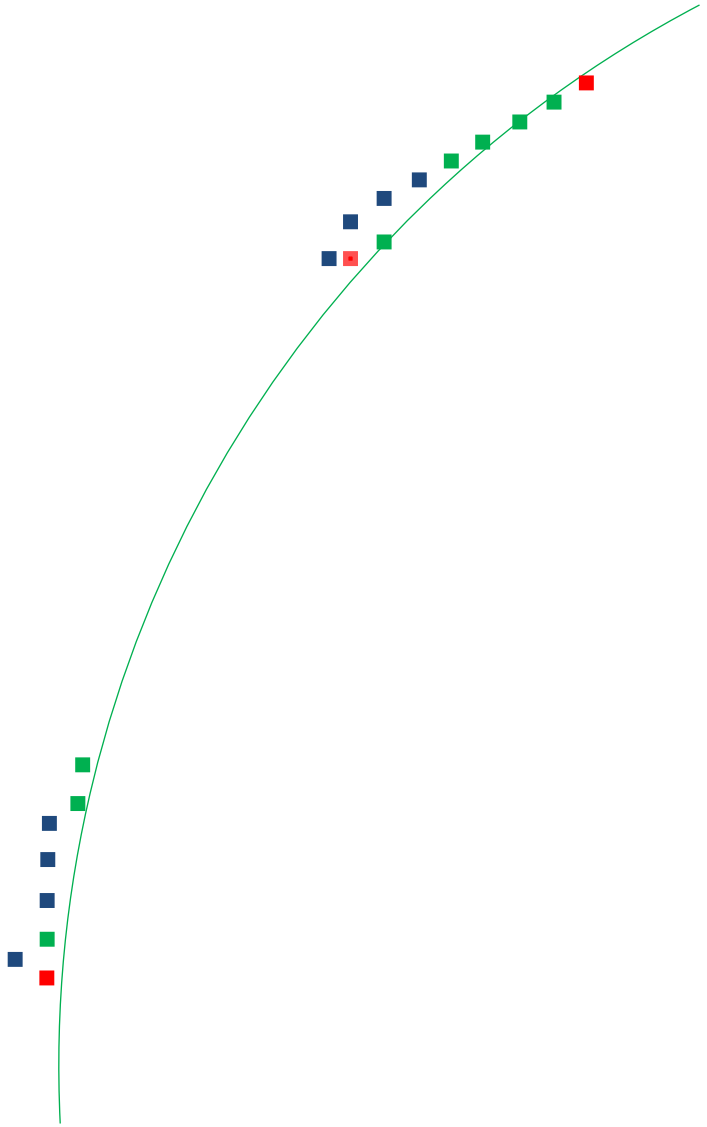
- Basic idea
  - Select three STT hits
  - Calculate Apollonius circles
  - Add other STT hits which are close to circles
  - Select best solution(s)

- First Step „Select three STT hits“ already most important step:
  - Combine all STT hits are too many combinations to be fast
    → Selection has to be done
  - If selection is too tight efficiency will drop

➔ How to do the selection?

- Create triplet from one hit out of each group
  - Calculate 8 Apollonius circles
  - Assign STT hits to circles

APOLLONIUS TRIPLET TRACK FINDER



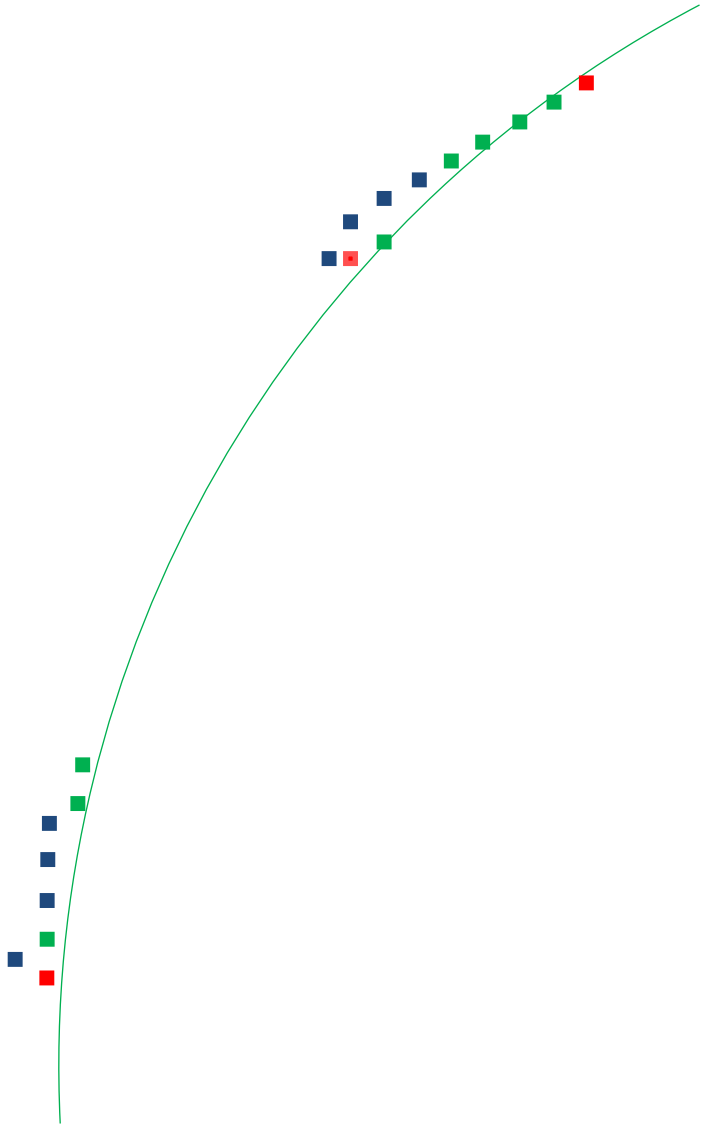
- Basic idea
  - Select three STT hits
  - Calculate Apollonius circles
  - Add other STT hits which are close to circles
  - Select best solution(s)

- First Step „Select three STT hits“ already most important step:
  - Combine all STT hits are too many combinations to be fast
    → Selection has to be done
  - If selection is too tight efficiency will drop

➔ How to do the selection?

- Create triplet from one hit out of each group
  - Calculate 8 Apollonius circles
  - Assign STT hits to circles

APOLLONIUS TRIPLET TRACK FINDER



- Basic idea
  - Select three STT hits
  - Calculate Apollonius circles
  - Add other STT hits which are close to circles
  - Select best solution(s)

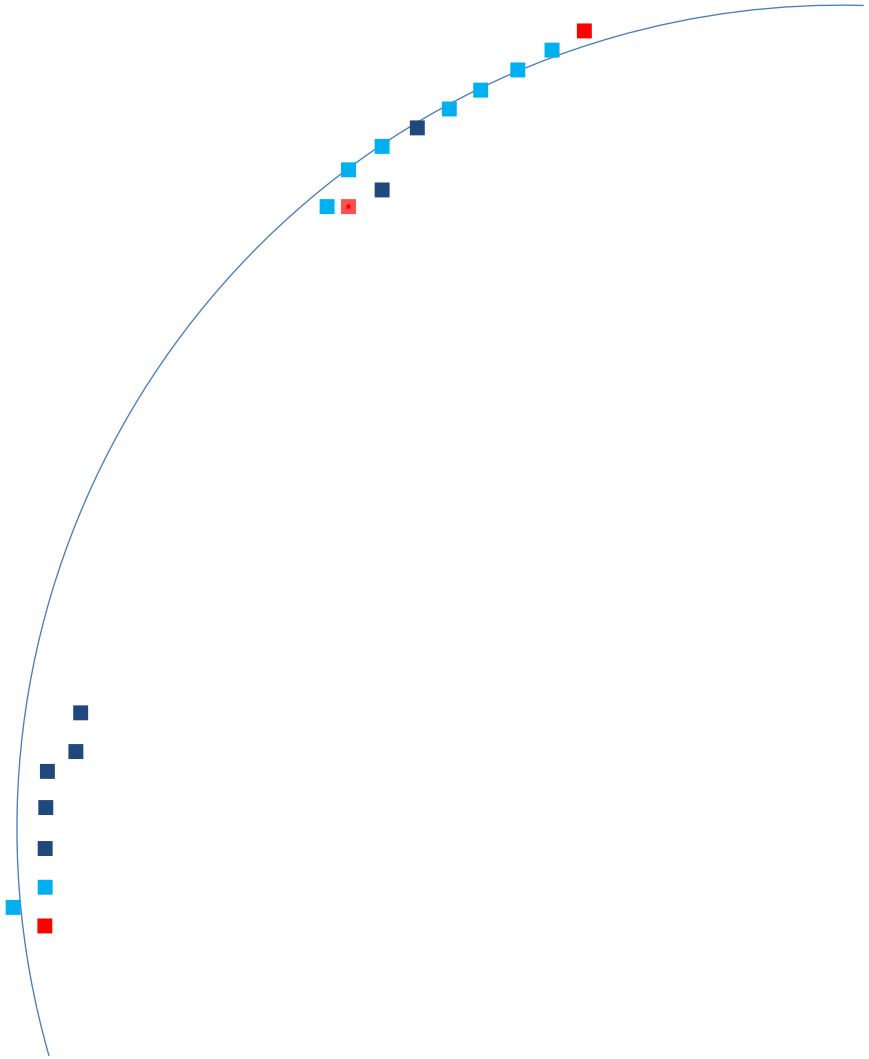
- First Step „Select three STT hits“ already most important step:
  - Combine all STT hits are too many combinations to be fast
    → Selection has to be done
  - If selection is too tight efficiency will drop

➔ How to do the selection?

- Create triplet from one hit out of each group

➔ How to decide which is the proper circle?
  - Continuity check

APOLLONIUS TRIPLET TRACK FINDER



- Basic idea
  - Select three STT hits
  - Calculate Apollonius circles
  - Add other STT hits which are close to circles
  - Select best solution(s)

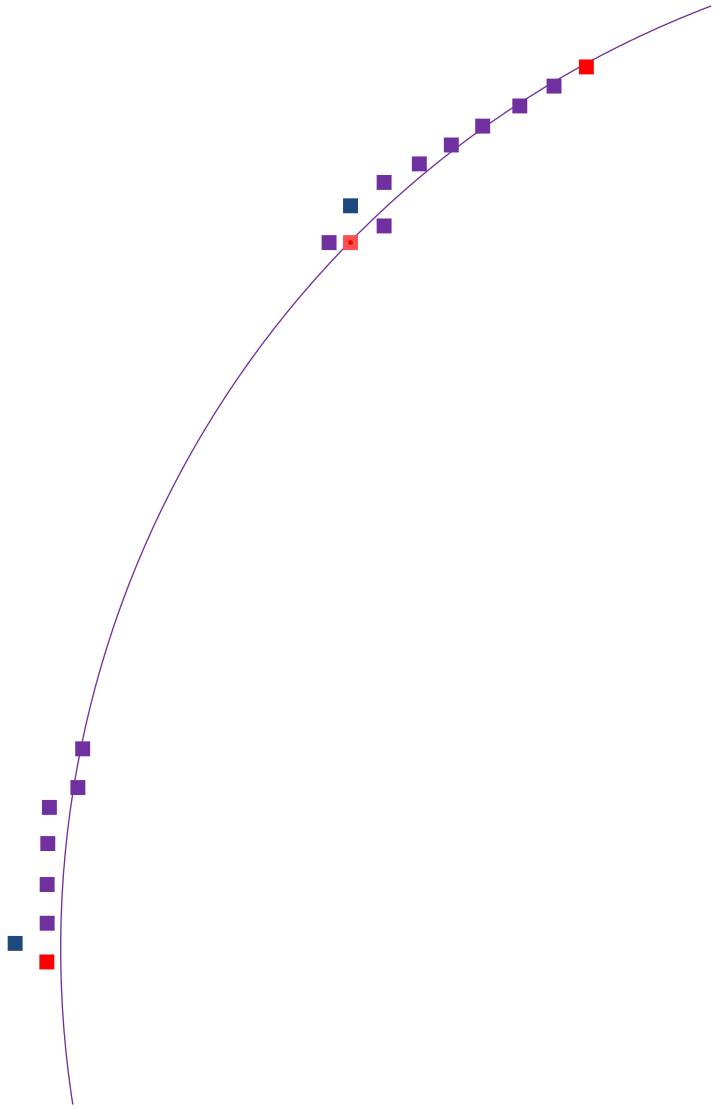
- First Step „Select three STT hits“ already most important step:
  - Combine all STT hits are too many combinations to be fast
    → Selection has to be done
  - If selection is too tight efficiency will drop

➔ How to do the selection?

- Create triplet from one hit out of each group

➔ How to decide which is the proper circle?
  - Continuity check

APOLLONIUS TRIPLET TRACK FINDER



- Basic idea
  - Select three STT hits
  - Calculate Apollonius circles
  - Add other STT hits which are close to circles
  - Select best solution(s)

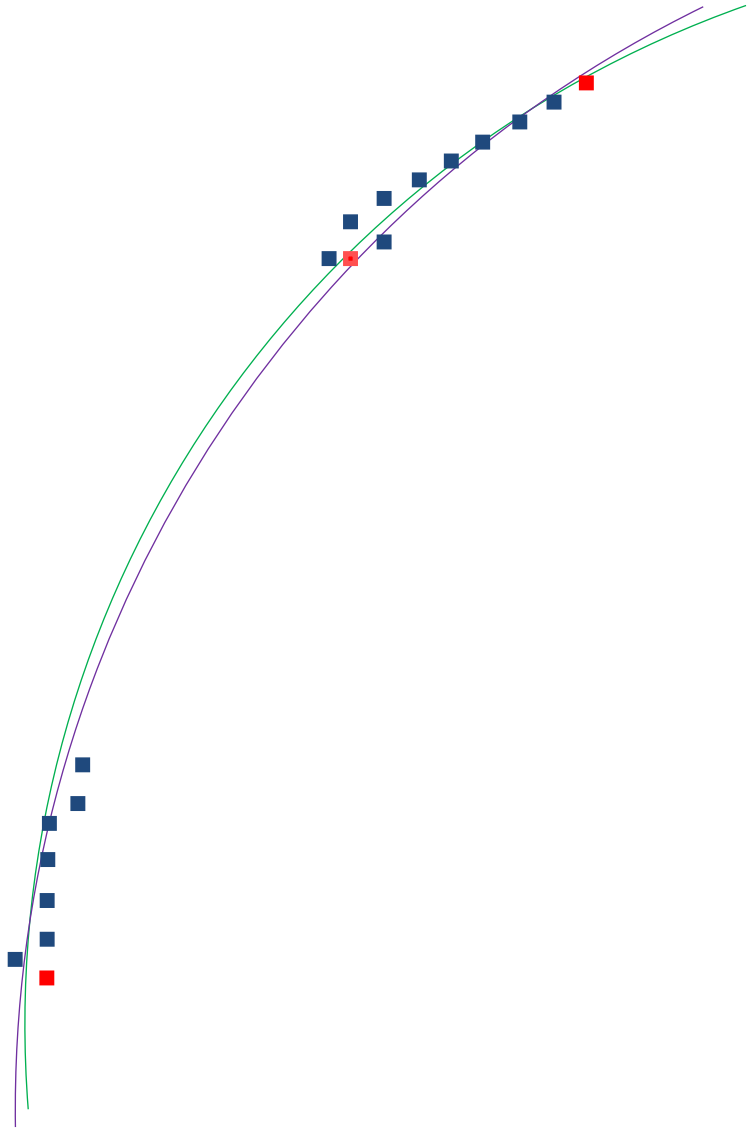
- First Step „Select three STT hits“ already most important step:
  - Combine all STT hits are too many combinations to be fast
    → Selection has to be done
  - If selection is too tight efficiency will drop

➔ How to do the selection?

- Create triplet from one hit out of each group

➔ How to decide which is the proper circle?
  - Continuity check

APOLLONIUS TRIPLET TRACK FINDER



- Basic idea
  - Select three STT hits
  - Calculate Apollonius circles
  - Add other STT hits which are close to circles
  - Select best solution(s)

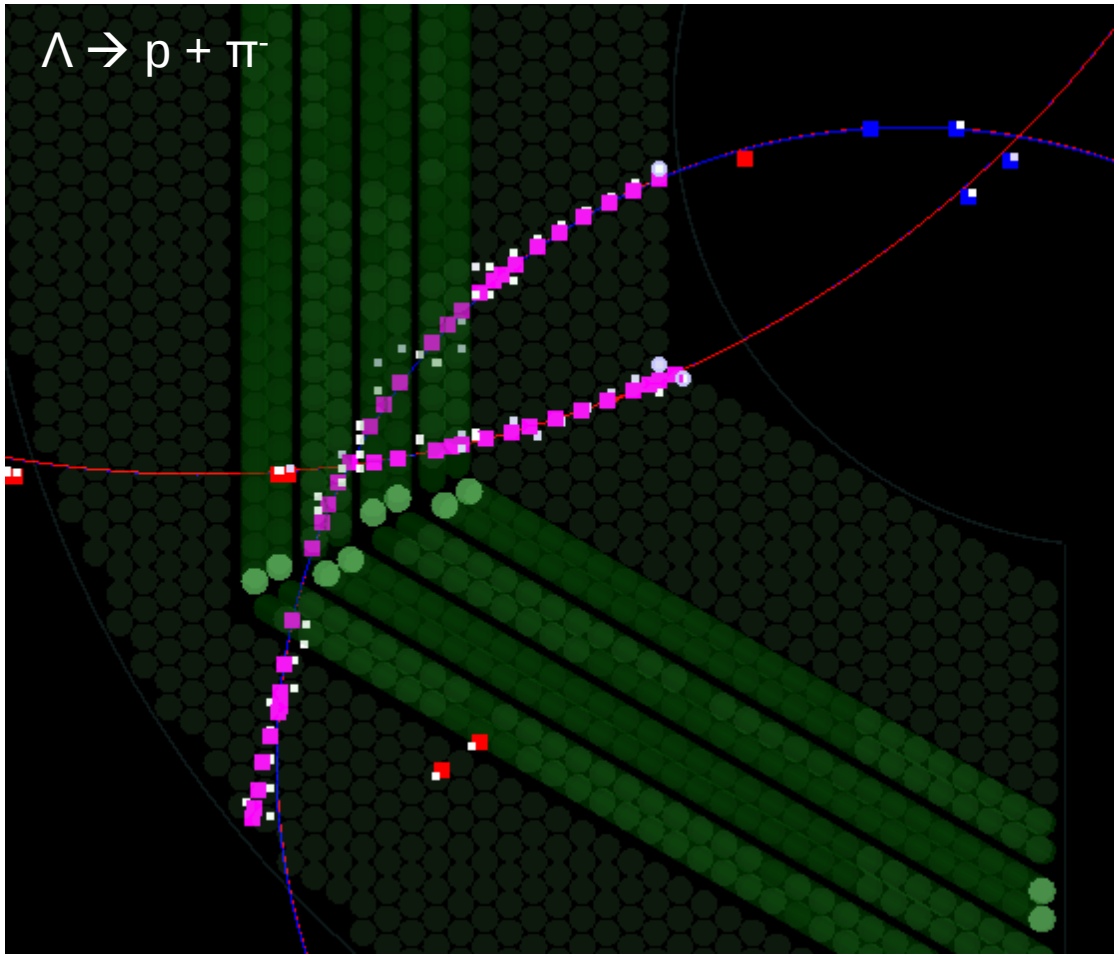
- First Step „Select three STT hits“ already most important step:
  - Combine all STT hits are too many combinations to be fast
    → Selection has to be done
  - If selection is too tight efficiency will drop

➔ How to do the selection?

- Create triplet from one hit out of each group

➔ How to decide which is the proper circle?
  - Continuity check
  - Number of hits in track
  - Quadratic distance of hits to circle

APOLLONIUS TRIPLET TRACK FINDER



- Basic idea
  - Select three STT hits
  - Calculate Apollonius circles
  - Add other STT hits which are close to circles
  - Select best solution(s)

- First Step „Select three STT hits“ already most important step:
  - Combine all STT hits are too many combinations to be fast
    → Selection has to be done
  - If selection is too tight efficiency will drop

➔ How to do the selection?

- Create triplet from one hit out of each group

➔ How to decide which is the proper circle?
  - Continuity check
  - Number of hits in track
  - Quadratic distance of hits to circle

➔Find good cut criterion is challenging

FIRST RESULTS

$1000\ \Lambda\overline{\Lambda}$ - events at 3 GeV/c

	Efficiency				Runtime	Ghosts	Clones
	Λ	P	π^-	$\overline{\Lambda}$			
Triplet Track Finder	83.7 % (108 / 129)	95.6 % (326 / 341)	85.3 % (466 / 546)	100 % (15 / 15)	581.5 s	43.1 %	161.9 %

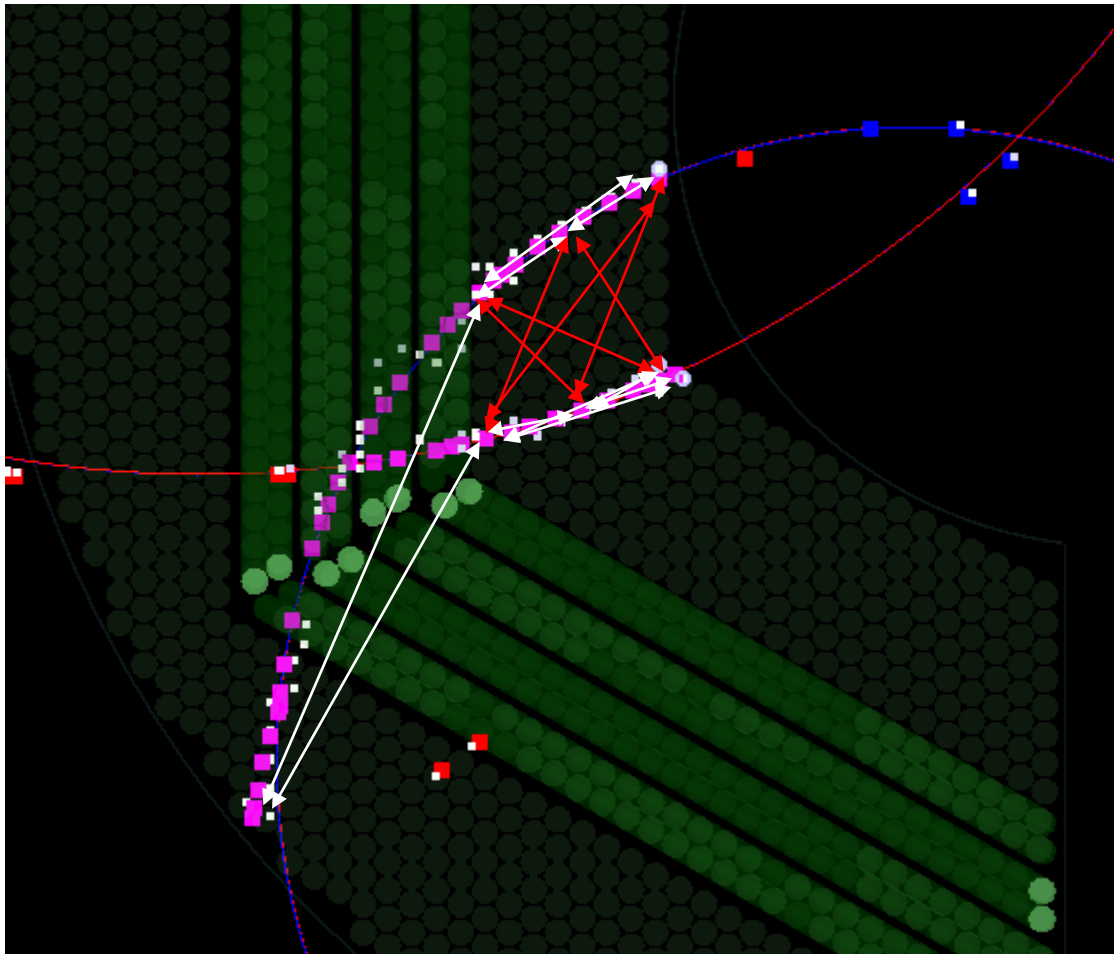
FIRST RESULTS

$1000\ \Lambda\overline{\Lambda}$ - events at 3 GeV/c

	Efficiency				Runtime	Ghosts	Clones
	Λ	P	π^-	$\overline{\Lambda}$			
Triplet Track Finder	83.7 % (108 / 129)	95.6 % (326 / 341)	85.3 % (466 / 546)	100 % (15 / 15)	581.5 s	43.1 %	161.9 %

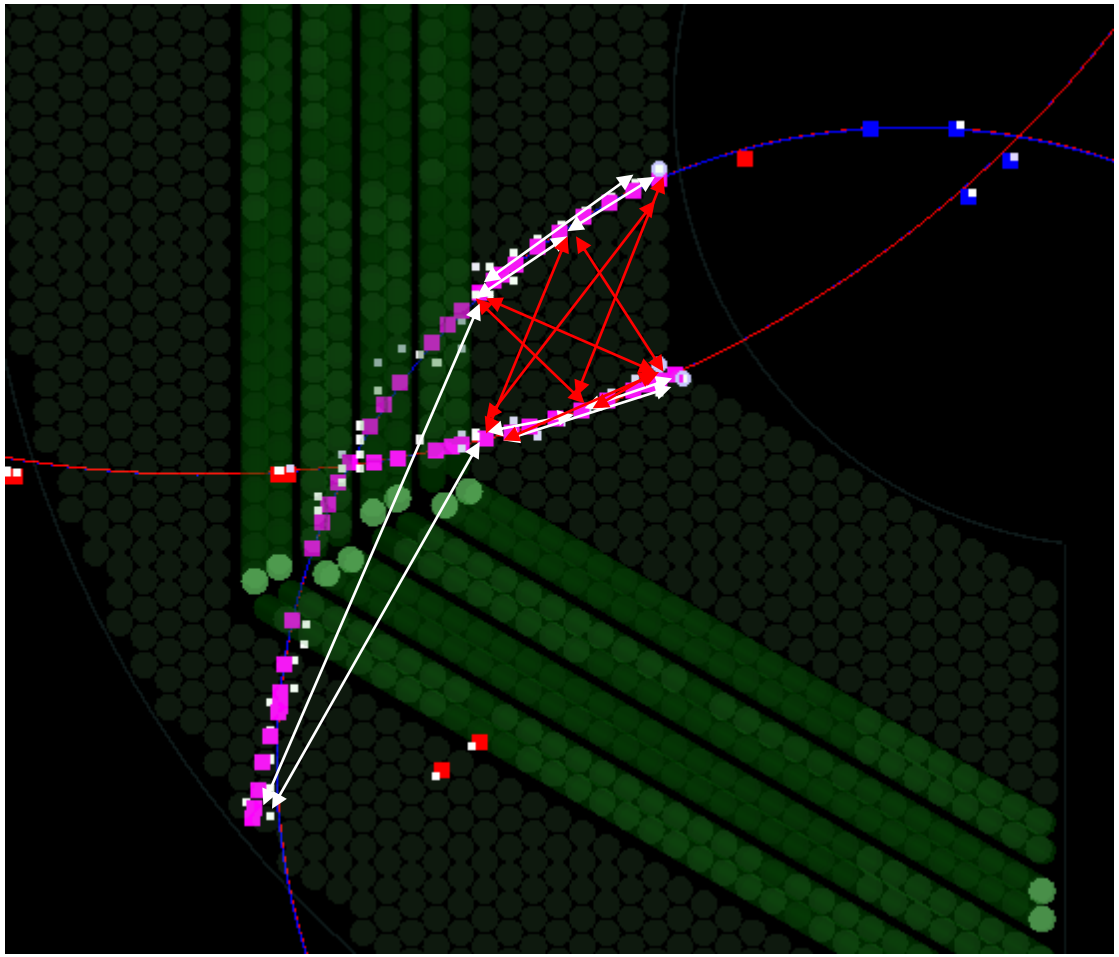
Reduce possible combinations

APOLLONIUS TRIPLET TRACK FINDER



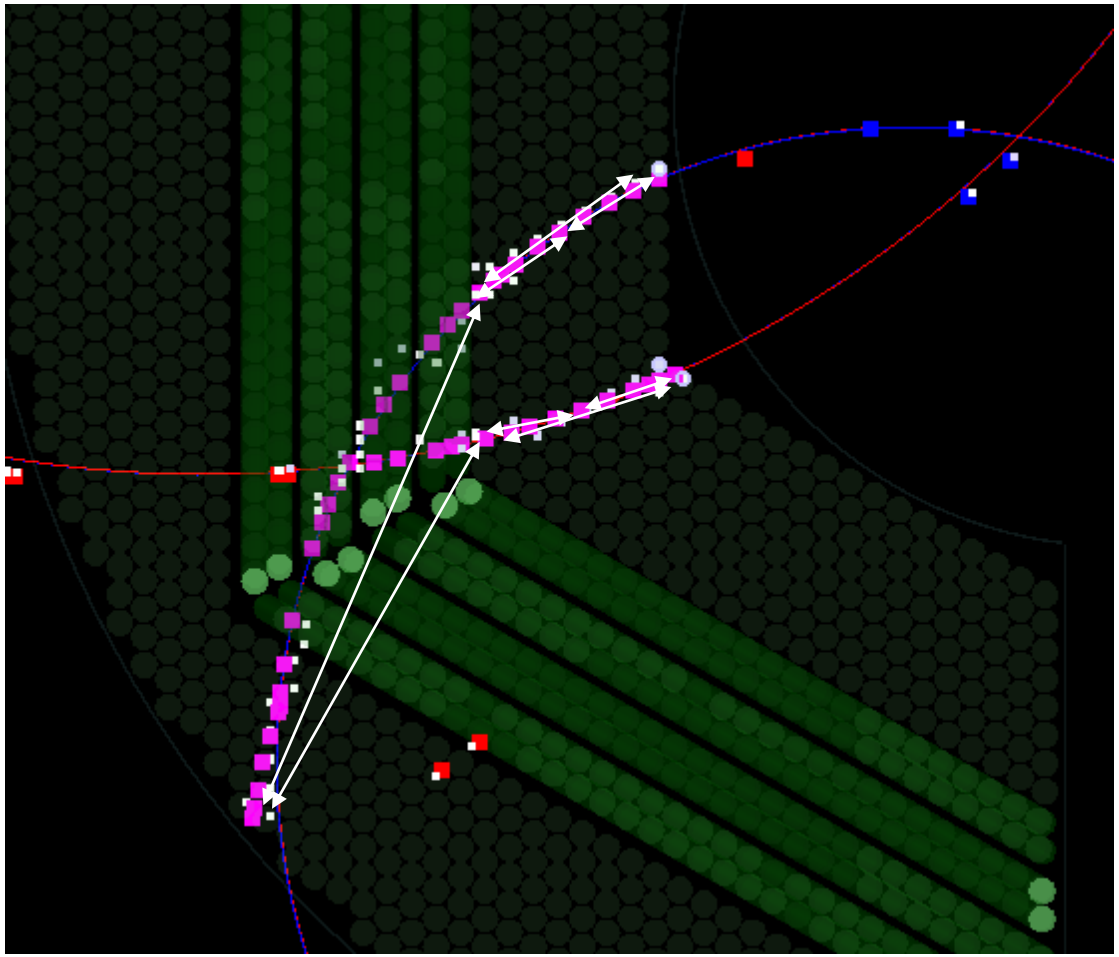
- Use CA Tracklet information
  - If first-mid or mid-last in one CA Tracklet avoid all other possibilities

APOLLONIUS TRIPLET TRACK FINDER



- Use CA Tracklet information
  - If first-mid or mid-last in one CA Tracklet avoid all other possibilities
  - If several hits of all first, mid or last hits are in one CA tracklet use only one of the

APOLLONIUS TRIPLET TRACK FINDER



- Use CA Tracklet information
  - If first-mid or mid-last in one CA Tracklet avoid all other possibilities
  - If several hits of all first, mid or last hits are in one CA tracklet use only one of the
  - Example:
    → Reduction from 18 to 4 combinations

FINAL RESULTS

$1000\ \Lambda\overline{\Lambda}$ - events at 3 GeV/c

	Efficiency				Runtime	Ghosts	Clones
	Λ	P	π^-	$\overline{\Lambda}$			
Triplet Track Finder	83.7 % (108 / 129)	95.6 % (326 / 341)	85.3 % (466 / 546)	100 % (15 / 15)	581.5 s	43.1 %	161.9 %
With reduction	80.6 % (104 / 129)	94.4 % (322 / 341)	83.3 % (455 / 546)	100 % (15 / 15)	166.7 s	25.8 %	99.2 %

FINAL RESULTS

$1000\ \Lambda\bar{\Lambda}$ - events at 3 GeV/c

	Efficiency				Runtime	Ghosts	Clones
	Λ	P	π^-	$\bar{\Lambda}$			
Triplet Track Finder	83.7 % (108 / 129)	95.6 % (326 / 341)	85.3 % (466 / 546)	100 % (15 / 15)	581.5 s	43.1 %	161.9 %
With reduction	80.6 % (104 / 129)	94.4 % (322 / 341)	83.3 % (455 / 546)	100 % (15 / 15)	166.7 s	25.8 %	99.2 %

Caused by curling tracks

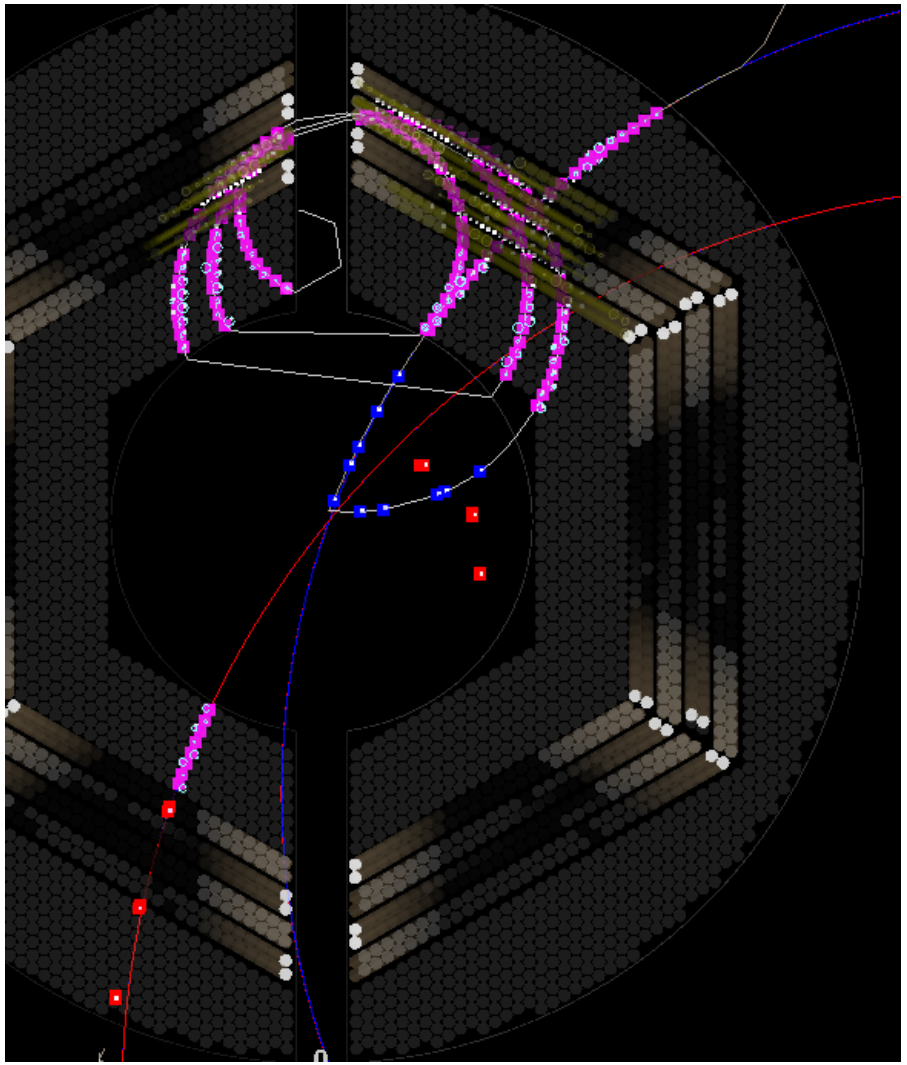
FINAL RESULTS

$1000\ \Lambda\bar{\Lambda}$ - events at 3 GeV/c

	Efficiency				Runtime	Ghosts	Clones
	Λ	P	π^-	$\bar{\Lambda}$			
Triplet Track Finder	83.7 % (108 / 129)	95.6 % (326 / 341)	85.3 % (466 / 546)	100 % (15 / 15)	581.5 s	43.1 %	161.9 %
With reduction	80.6 % (104 / 129)	94.4 % (322 / 341)	83.3 % (455 / 546)	100 % (15 / 15)	166.7 s	25.8 %	99.2 %
Without curling tracks	95.7 % (22 / 23)	96.9 % (249 / 257)	89.1 % (33 / 37)	100 % (9 / 9)	55.4 s	3.2 %	7.2 %

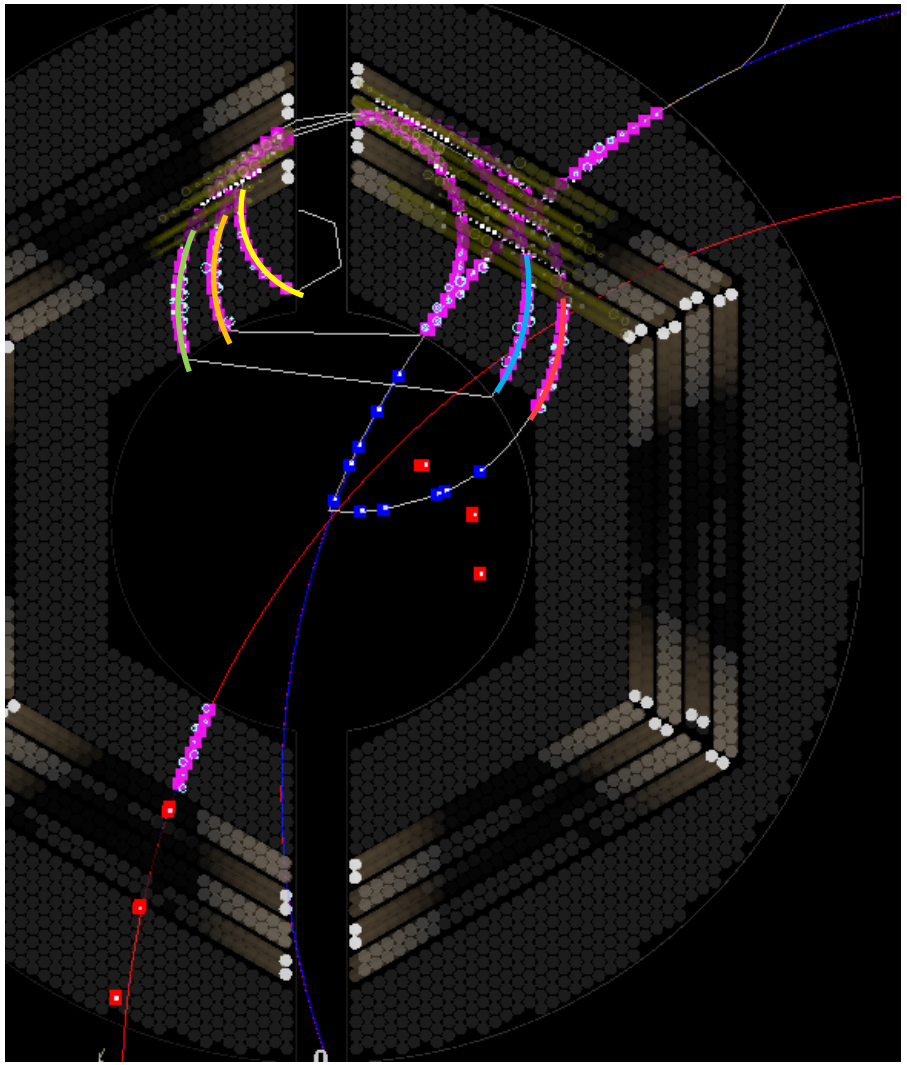
➔ Curling tracks essential for high Lambda efficiency
➔ Modification of algorithm necessary to deal with curling tracks

POSSIBLE IMPROVEMENTS: CURLING TRACKS



- Curling tracks are found too often or create too many possible combinations
  ➔ high ghost and clone rate

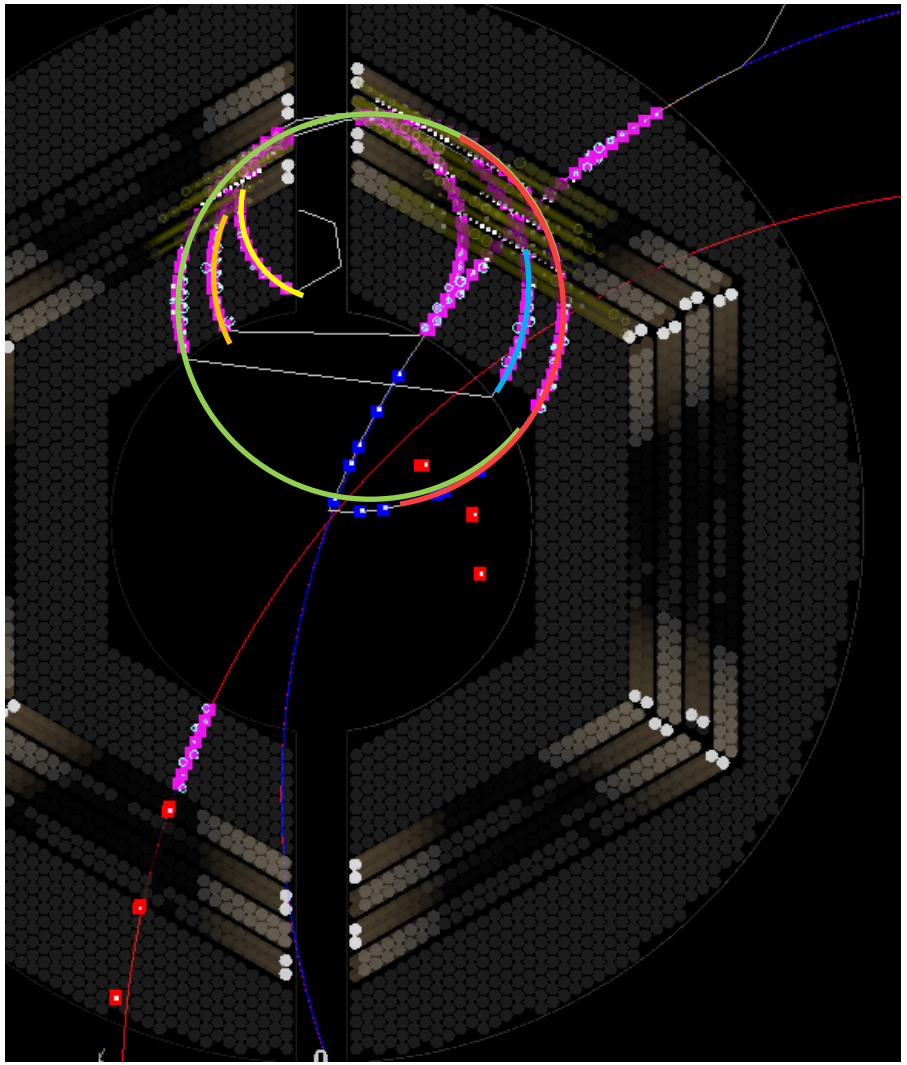
POSSIBLE IMPROVEMENTS: CURLING TRACKS



- Curling tracks are found too often or create too many possible combinations
  ➔ high ghost and clone rate

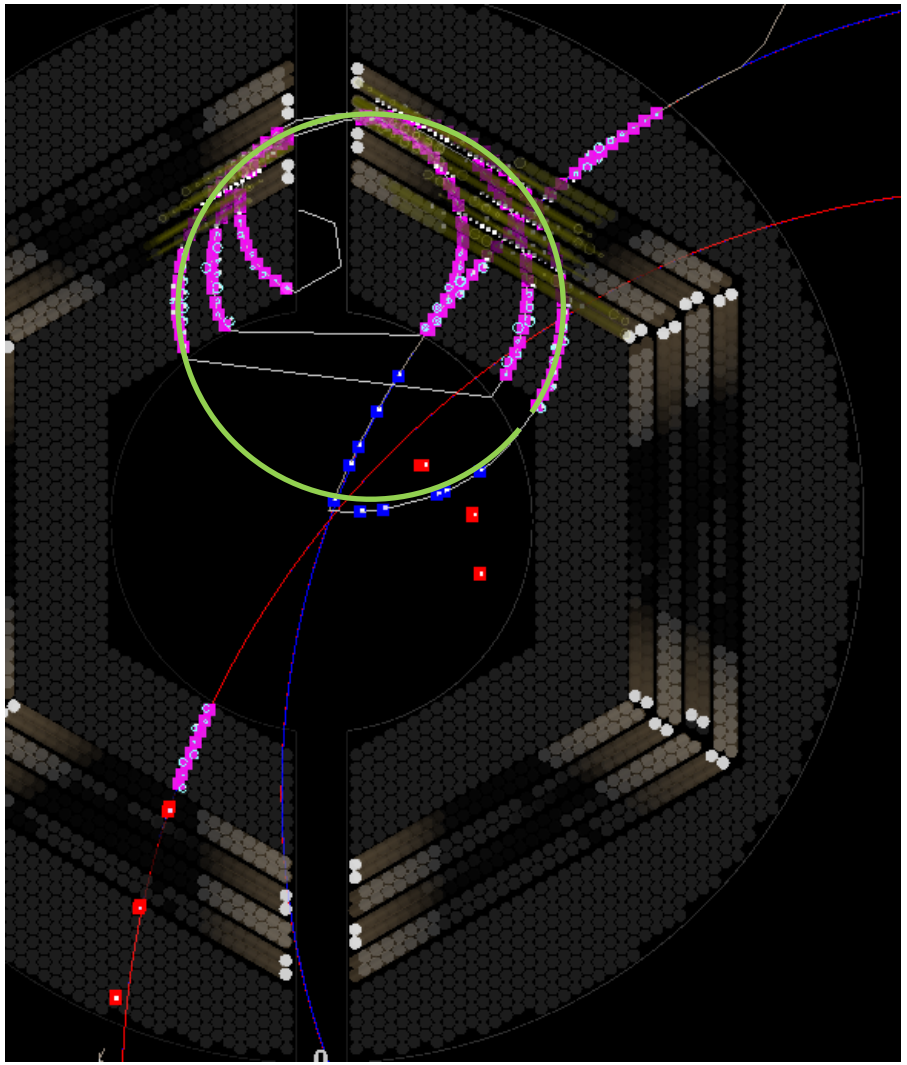
- Divide all created subtracks (e.g. Apollonius Triplet Finder without skewed or CA)

POSSIBLE IMPROVEMENTS: CURLING TRACKS



- Curling tracks are found too often or create too many possible combinations
  ➔ high ghost and clone rate
- Divide all created subtracks (e.g. Apollonius Triplet Finder without skewed or CA)
- Merge if possible: green and red are nearly identical

POSSIBLE IMPROVEMENTS: CURLING TRACKS



- Curling tracks are found too often or create too many possible combinations
  ➔ high ghost and clone rate
- Divide all created subtracks (e.g. Apollonius Triplet Finder without skewed or CA)
- Merge if possible: green and red are nearly identical
- Remove all circles which are inside the found circle

SUMMARY

- New track finder for secondary particles:
  - Finds 80.6% of all reconstructible Λ – particles ($1000\ \Lambda\bar{\Lambda}$ - events at 3 GeV/c)
  - Ghost and Clone ratio high due to curling tracks
  - Curling track reconstruction is under development
- ApolloniusTripletFinder could also be used for primaries?

COMPARISON TO BARREL TRACK FINDER

1000 FTF - events at 7 GeV/c

	Efficiency		Runtime	Ghosts	Clones
	primaries	secondaries			
Triplet Track Finder	81.6 %	60.3 %	581.5 s	40.3 %	90.4 %
With reduction	80.6 %	57.8 %	166.7 s	28.4 %	64.2 %
Barrel Track Finder	83.7 %	31.2 %	17.5 s	20.7 %	20.7 %

a bit worse

significantly better

Further development for curling tracks

SUMMARY

- New track finder for secondary particles:
  - Finds 80.6% of all reconstructible Λ – particles ($1000\ \Lambda\bar{\Lambda}$ - events at 3 GeV/c)
  - Ghost and Clone ratio high due to curling tracks
  - Curling track reconstruction is under development
- ApolloniusTripletFinder could also be used for primaries
  - Efficiency for primaries is similar
  - For secondaries is much better
  - Runtime, ghosts and clones under development

THANK YOU VERY MUCH FOR YOUR ATTENTION!