Bundesministerium
für Bildung
und Forschung

GOETHE
UNIVERSITÄT
FRANKFURT AM MAIN

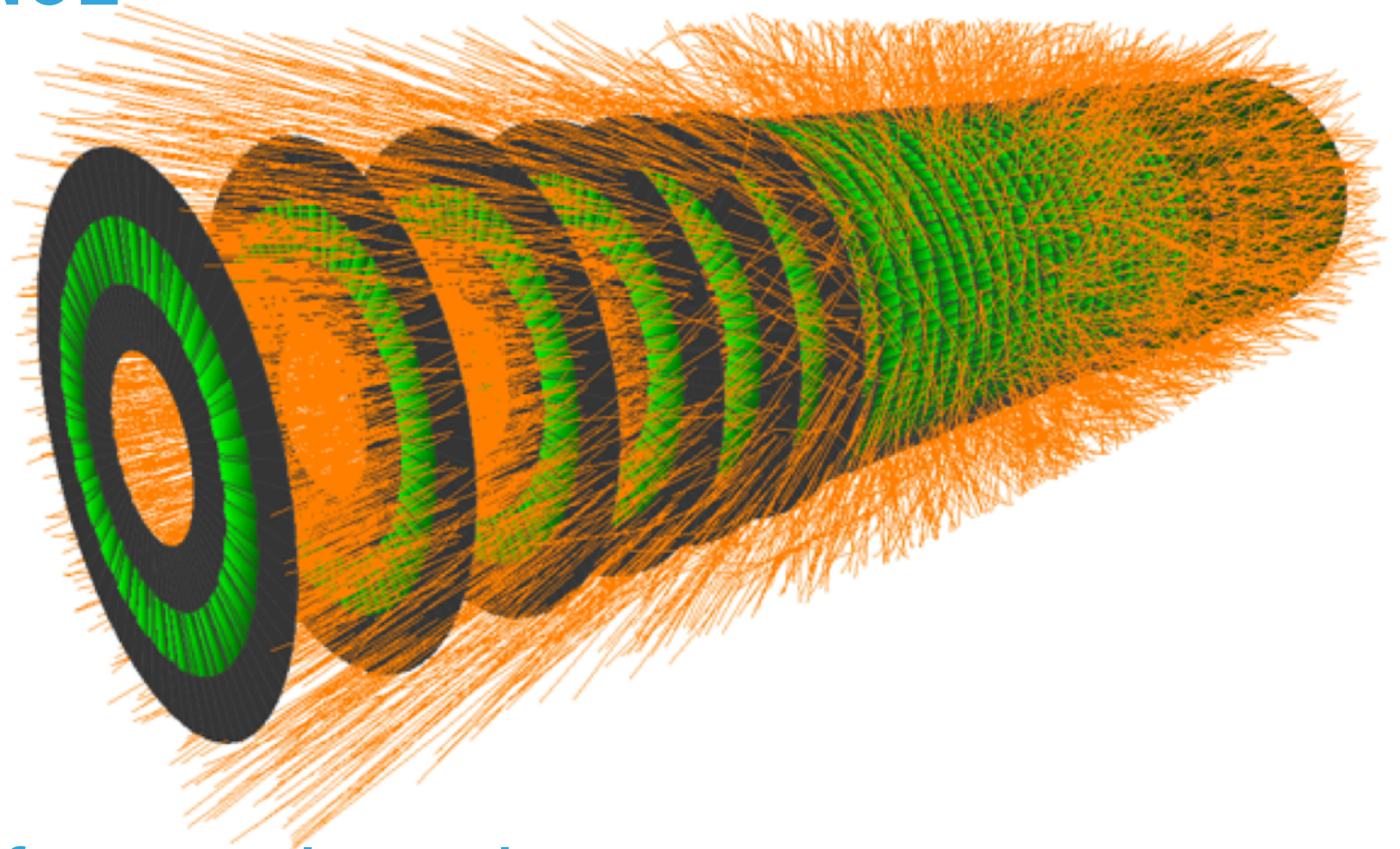
MIKADO APPROACH FOR THE TRACKML TRACKING CHALLENGE

SERGEY GORBUNOV [1,2]

[1] Johann Wolfgang Goethe-University Frankfurt am Main
[2] FIAS Frankfurt Institute for Advanced Studies

- **3-RD PLACE ACCURACY PHASE**
- **1-ST PLACE THROUGHPUT PHASE**

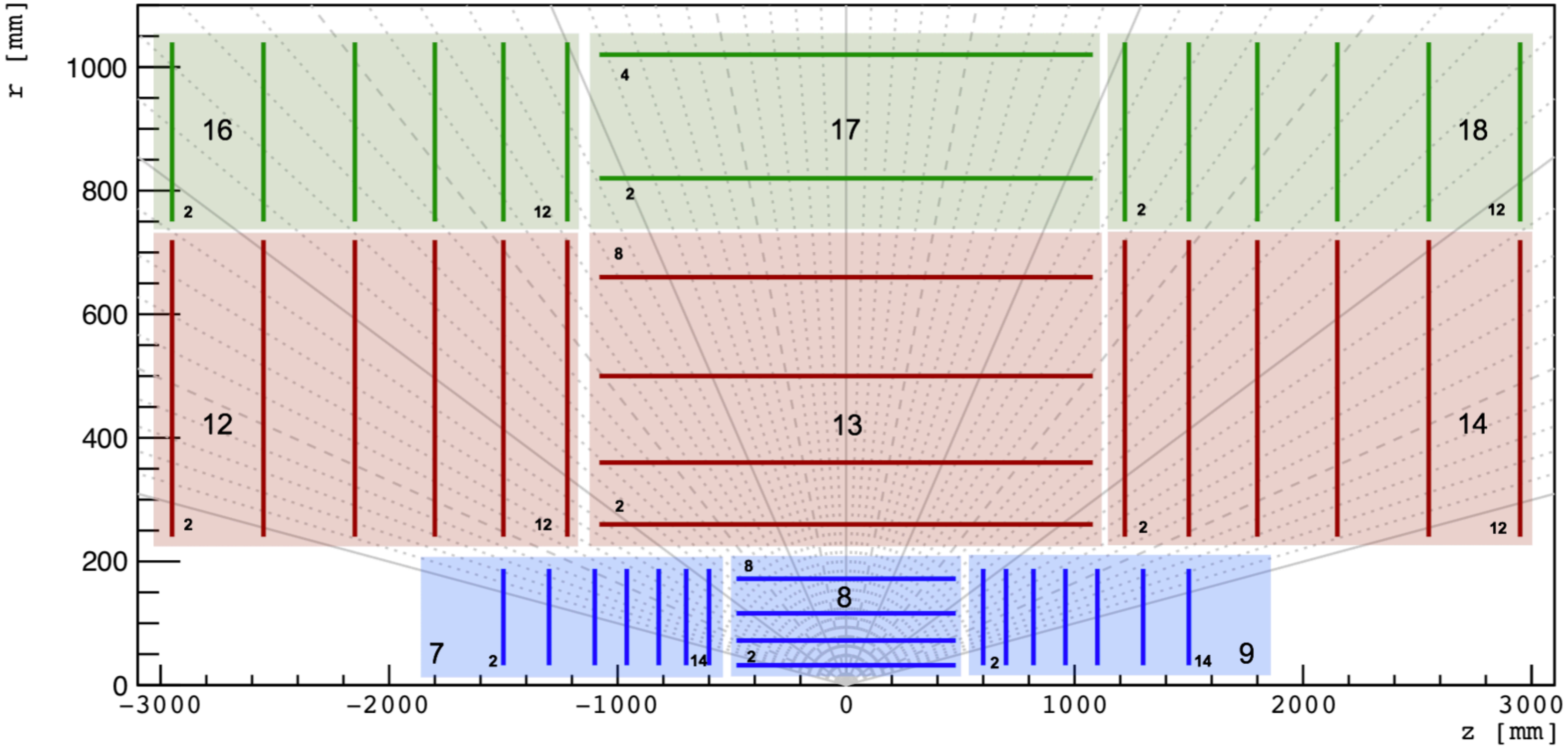
PARTICLE TRACKING CHALLENGE

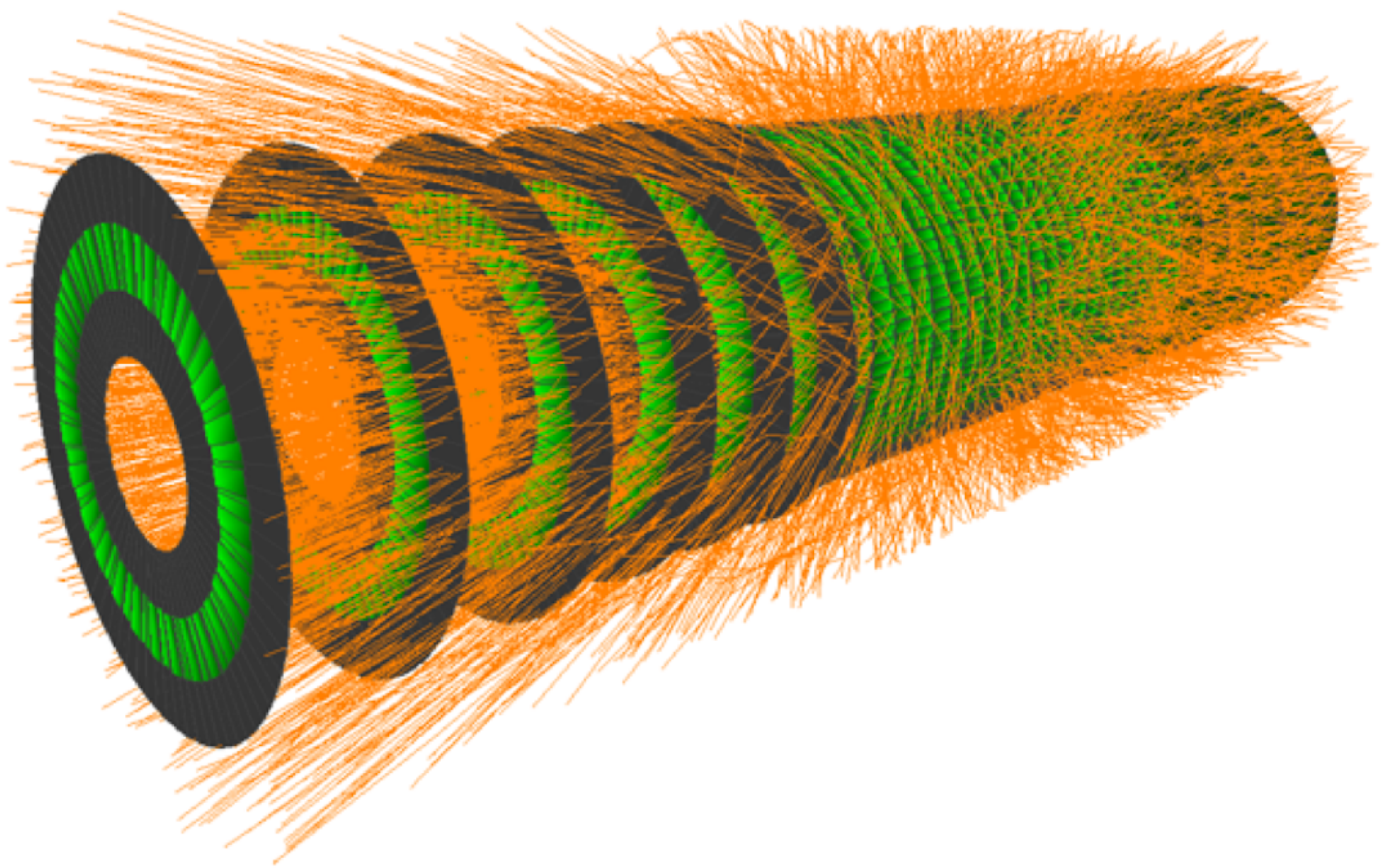


GOAL: develop novel techniques for particle track reconstruction in „typical“ CERN detector

- Realistic detector
- Incomplete data in order to give a head start for ML algorithms
- Open competition! All algorithms were welcome.
- 2 competition phases: Accuracy and Throughput
- Money prizes: $12k / $8k / $5k and $7k / $5k / $3k

PARTICLE TRACKING CHALLENGE





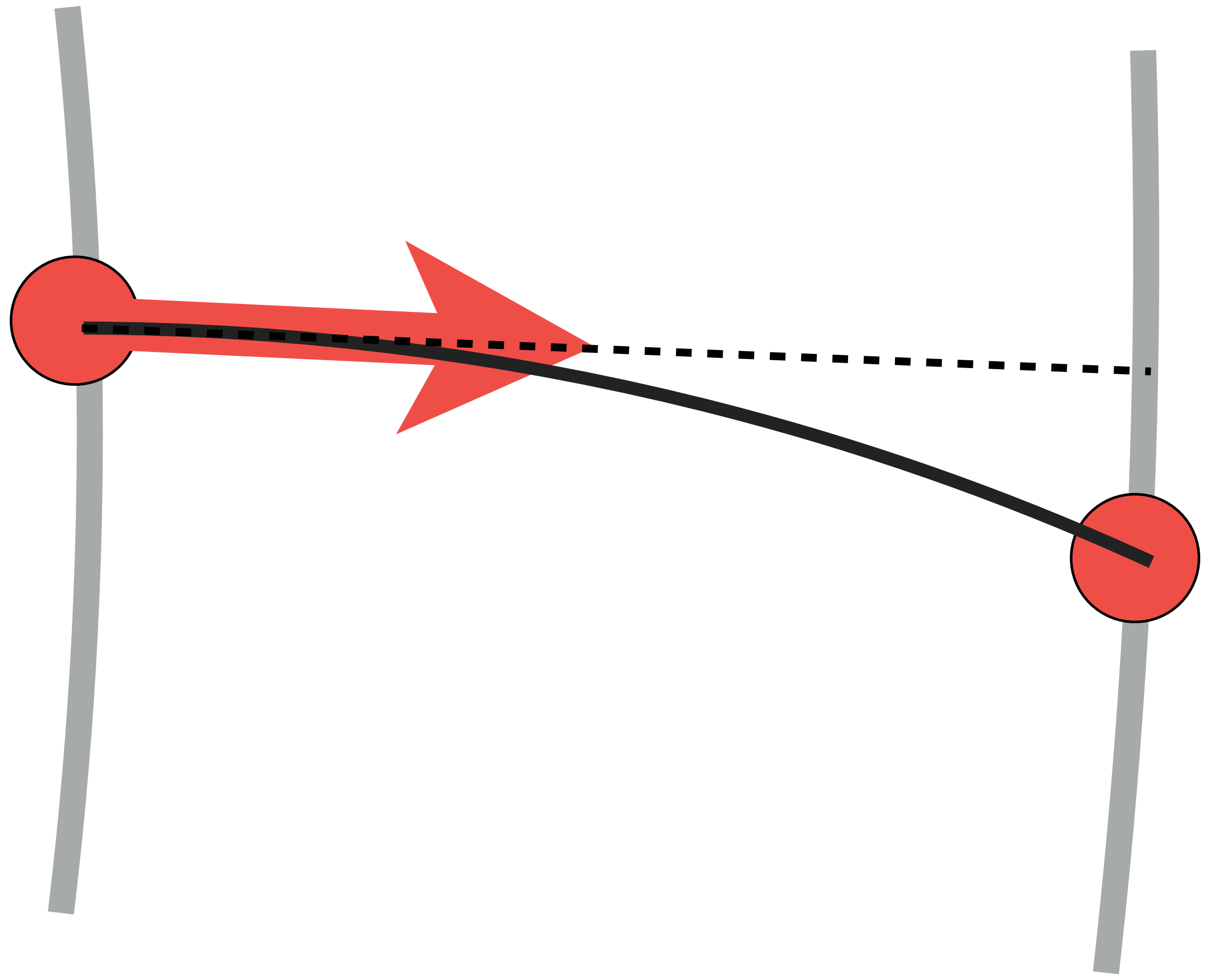
- complicated detector geometry
- high dense events
- strict requirements for efficiency: 100%== find all the hits on all the tracks
- non-uniform magnetic field (unknown!)
- multiple scattering in material (unknown!)
- varied resolution (unknown!)

BEFORE THE START: CHEATING

The magnetic field and the detector material was not given to us in order to make it harder to use conventional algorithms.

BUT..

We are scientists :) We are used to extract knowledge from the observations.



- true positions + momentum => curvature => magnetic field
- field + deviations from the ideal helix => detector material

A bit overdoing? It depends..
The 2-nd place competitor (**demelian**) reconstructed physical properties of the solenoid magnet. Then he lets the magnet produce the field.

ALL THE INFORMATION IS FOUND, WHICH WAY TO GO ?

APPLY SOME EXISTING ALGORITHM

- doesn't sound interesting
- a lot of work
- future use is in question
- however, new experience

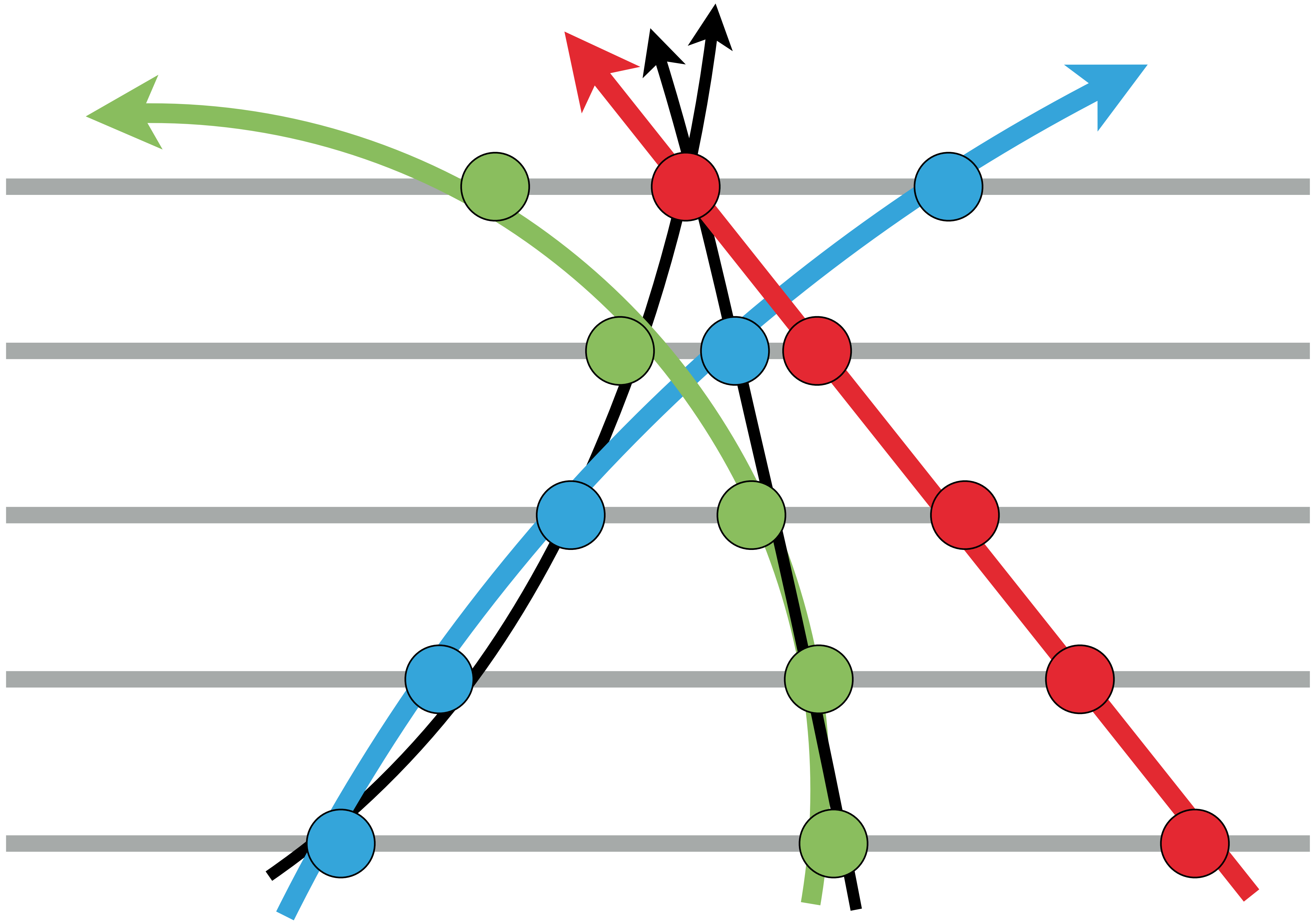
THE 2-ND PLACE (D.EMELIANOV):
CELLULAR AUTOMATON + KALMAN FILTER WITH RUNGE-KUTTA EXTRAPOLATION + NEW FEATURES

TRY TO INVENT SOMETHING SIMPLE

- lot of trying, no guarantee
- might be fun
- could be used later

THE 1-ST PLACE (ME):
NEW MIKADO APPROACH: CRAZY STRATEGY WITH A PRIMITIVE COMBINATORIAL ENGINE

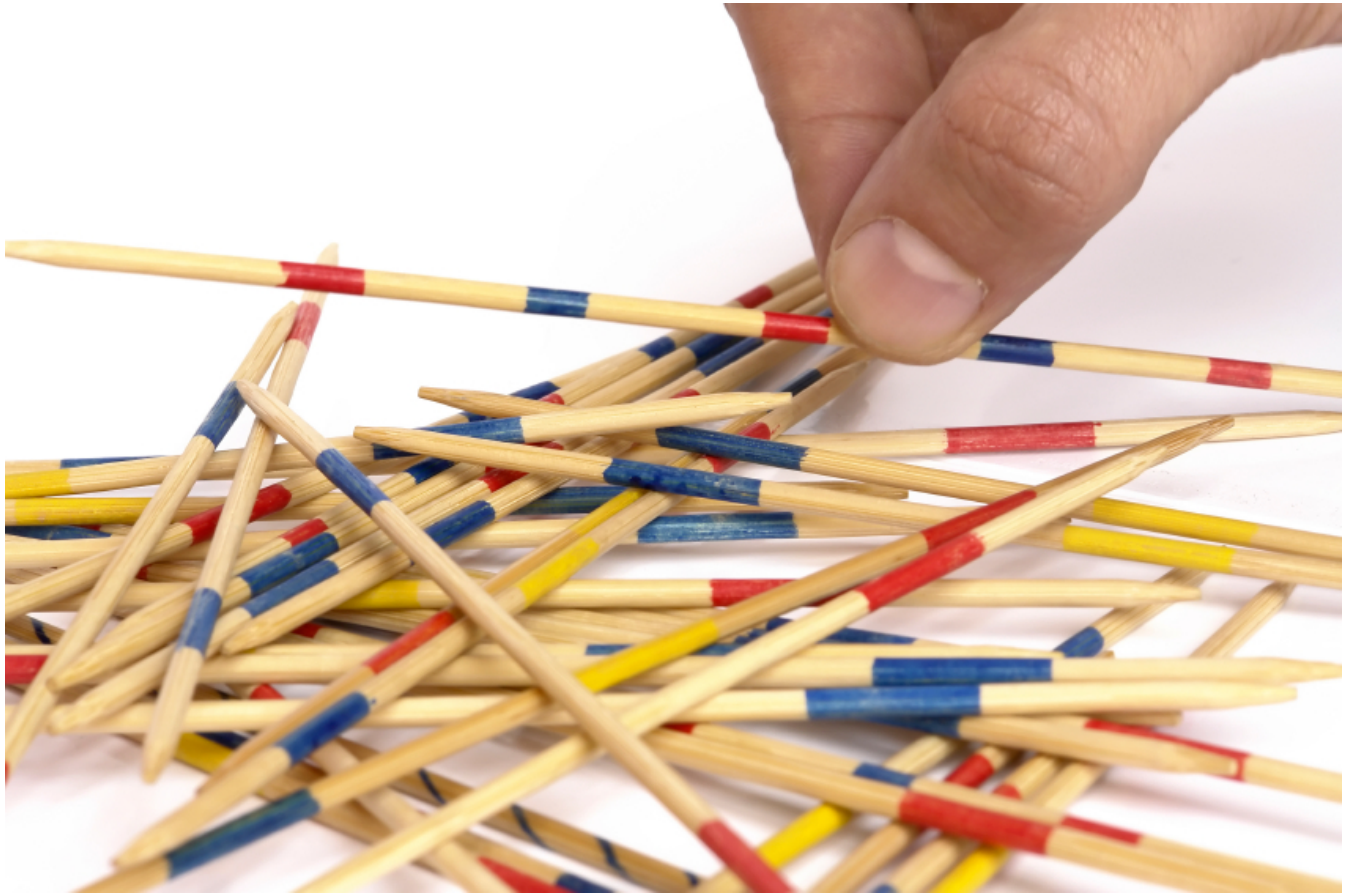
USUAL TRACKING STRATEGY



Strategy: Find all reasonable candidates; then select the better ones.

Problem: It needs accurate math.

Tracks are different.
To distinguish between good and bad candidates one needs precise, flexible mathematics which is applicable to all kind of tracks.



MIKADO STRATEGY

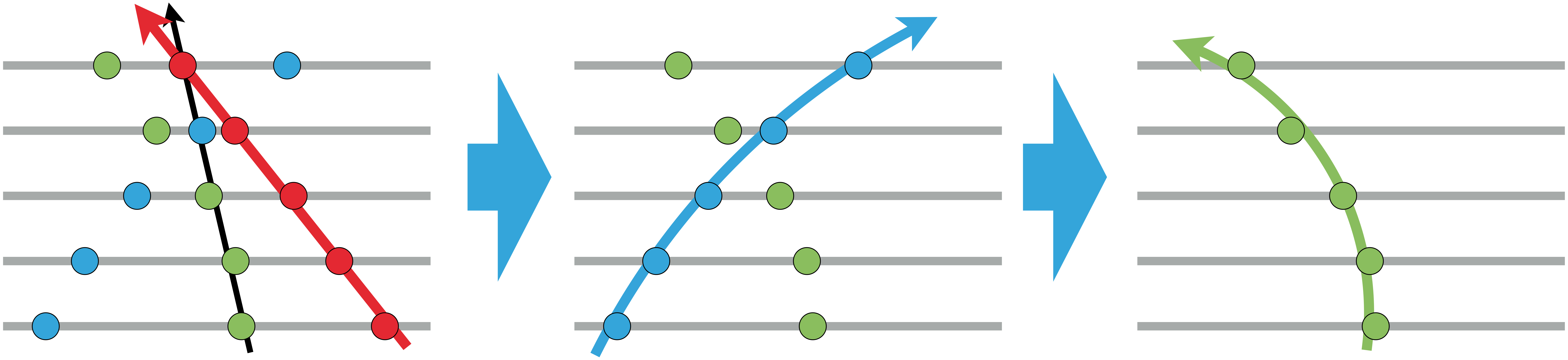
- reconstruct data in small portions, trying to not destroy the rest
- each pass reconstructs similar tracks -> same quality criteria are applied -> simple math
- 60 reconstruction passes
- each pass process little combinatorics, therefore is extremely fast

MIKADO STRATEGY: ILLUSTRATION

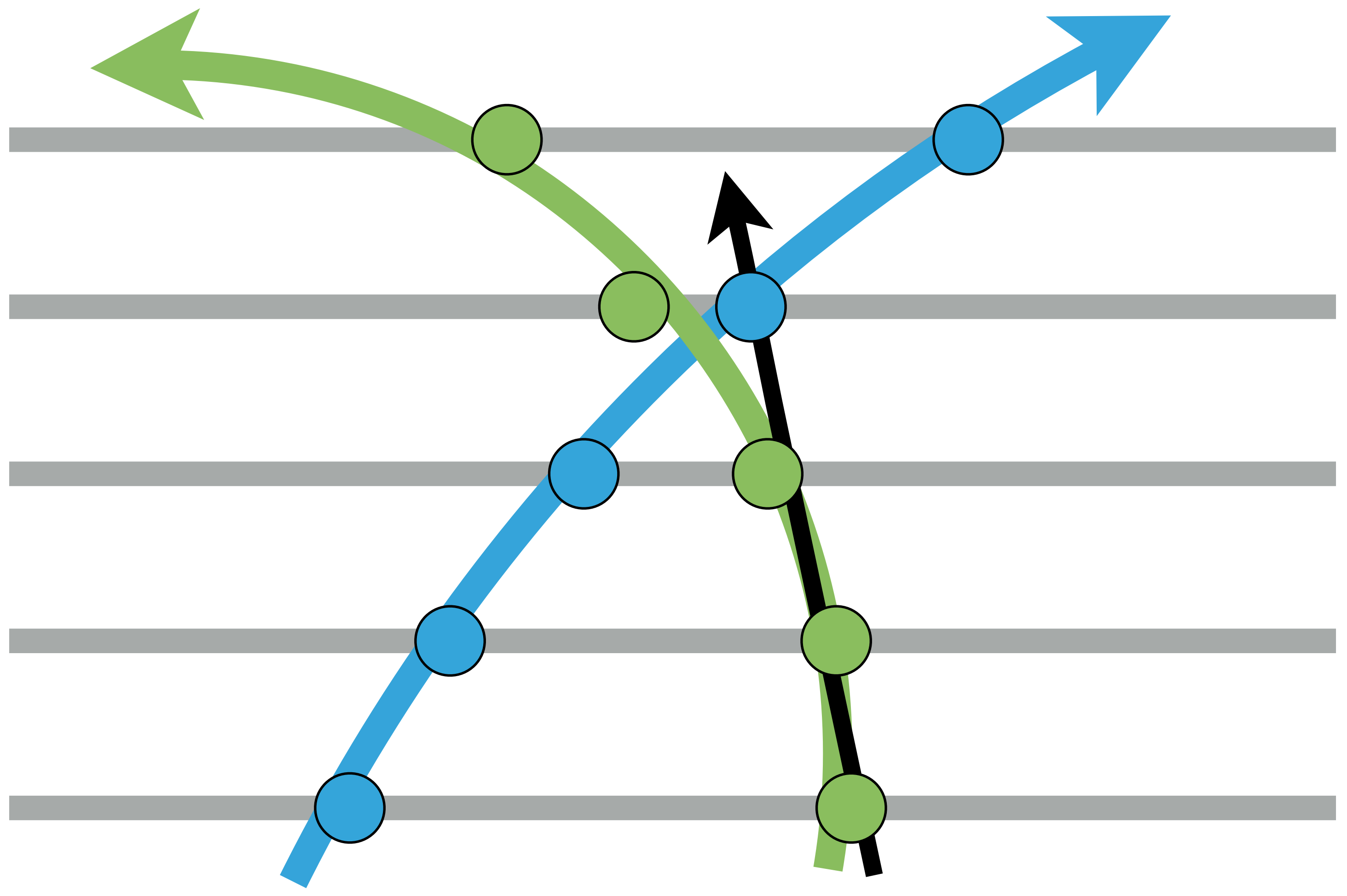
First, find straight tracks..

Then curved tracks..

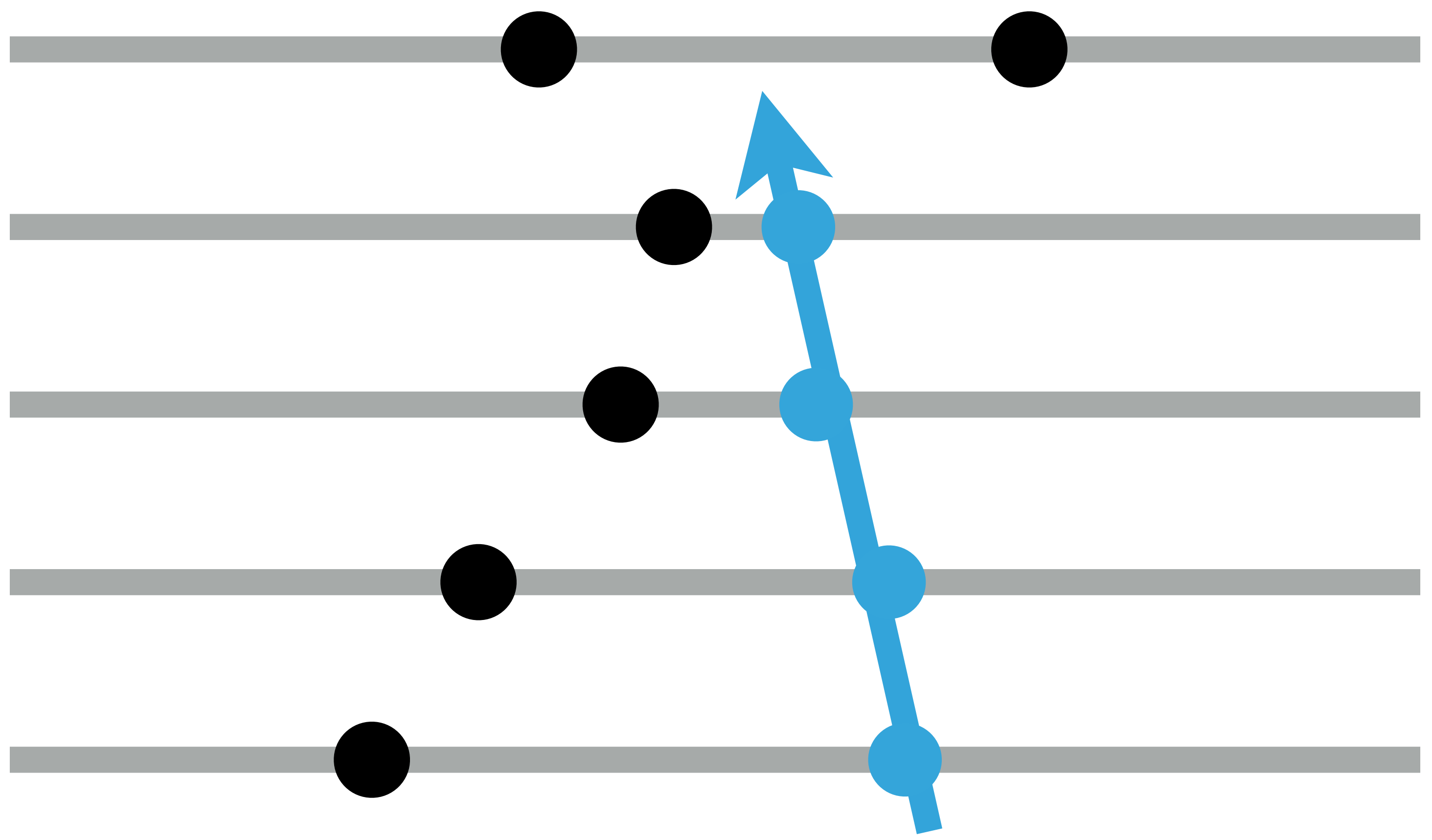
Then more curved tracks ...



all good candidates



currently visible candidates



PROBLEM: MIKADO ALGORITHM IS „BLIND“

Problem: the algorithm is „blind“. During a current pass it sees only small fraction of all possible track candidates.

Solution: don’t be greedy:)
Don’t try to keep everything that has been reconstructed.
At each pass take only clean, well-determined tracks.
Optimize passes for purity, not for efficiency.

LET MIKADO PASSES REFLECT THE TRACK SELECTION CRITERIA

TRACK PRIORITY

1. fast tracks with >=10 hits
2. fast tracks with >=9 hits
3. fast tracks with >=8 hits
4. slower tracks with >=10 hits
5. slower tracks with >=9 hits
6. slower tracks with >=8 hits
7. fast tracks with >=7 hits
8. slow tacks with >=10 hits
9. ....

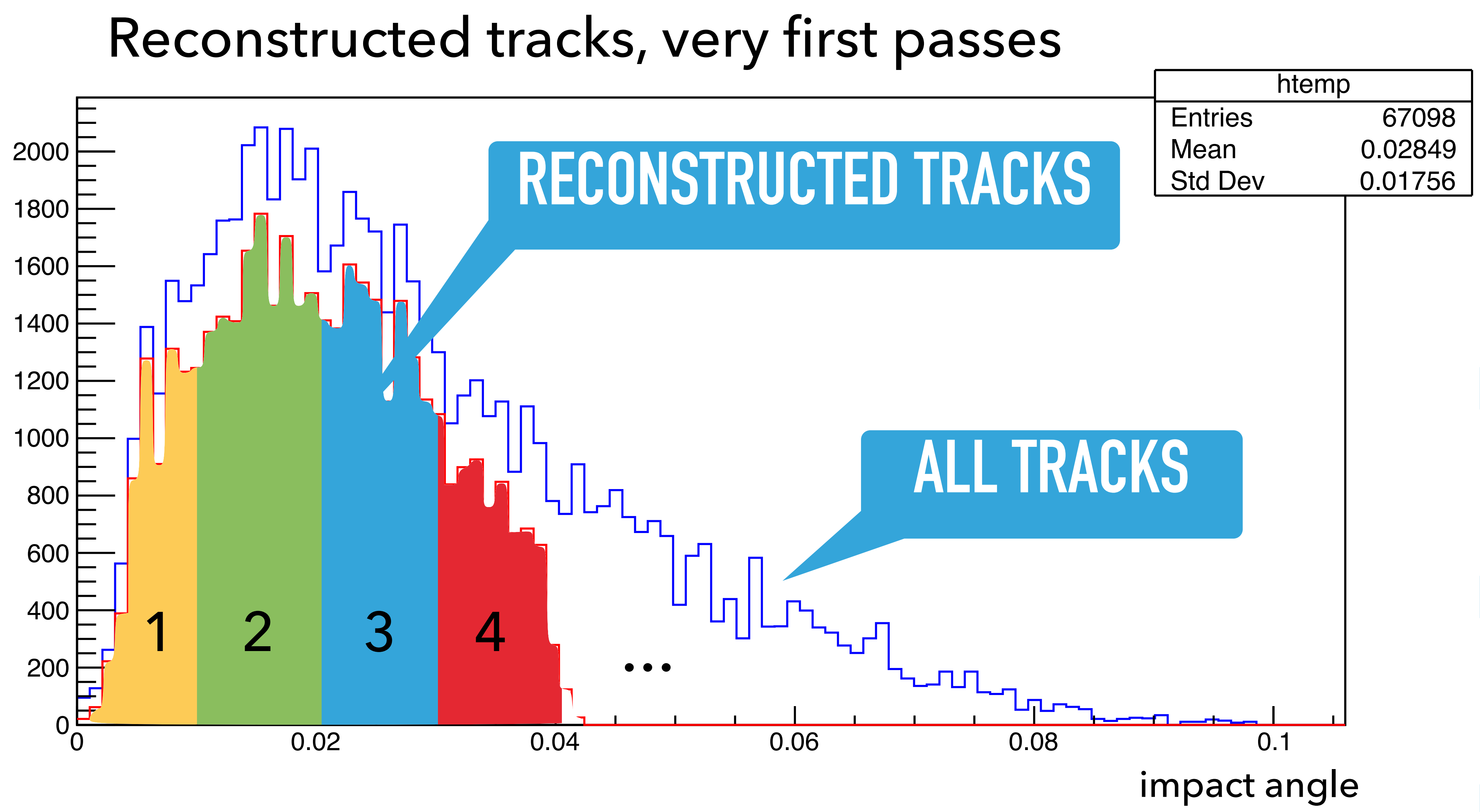
MIKADO PASSES

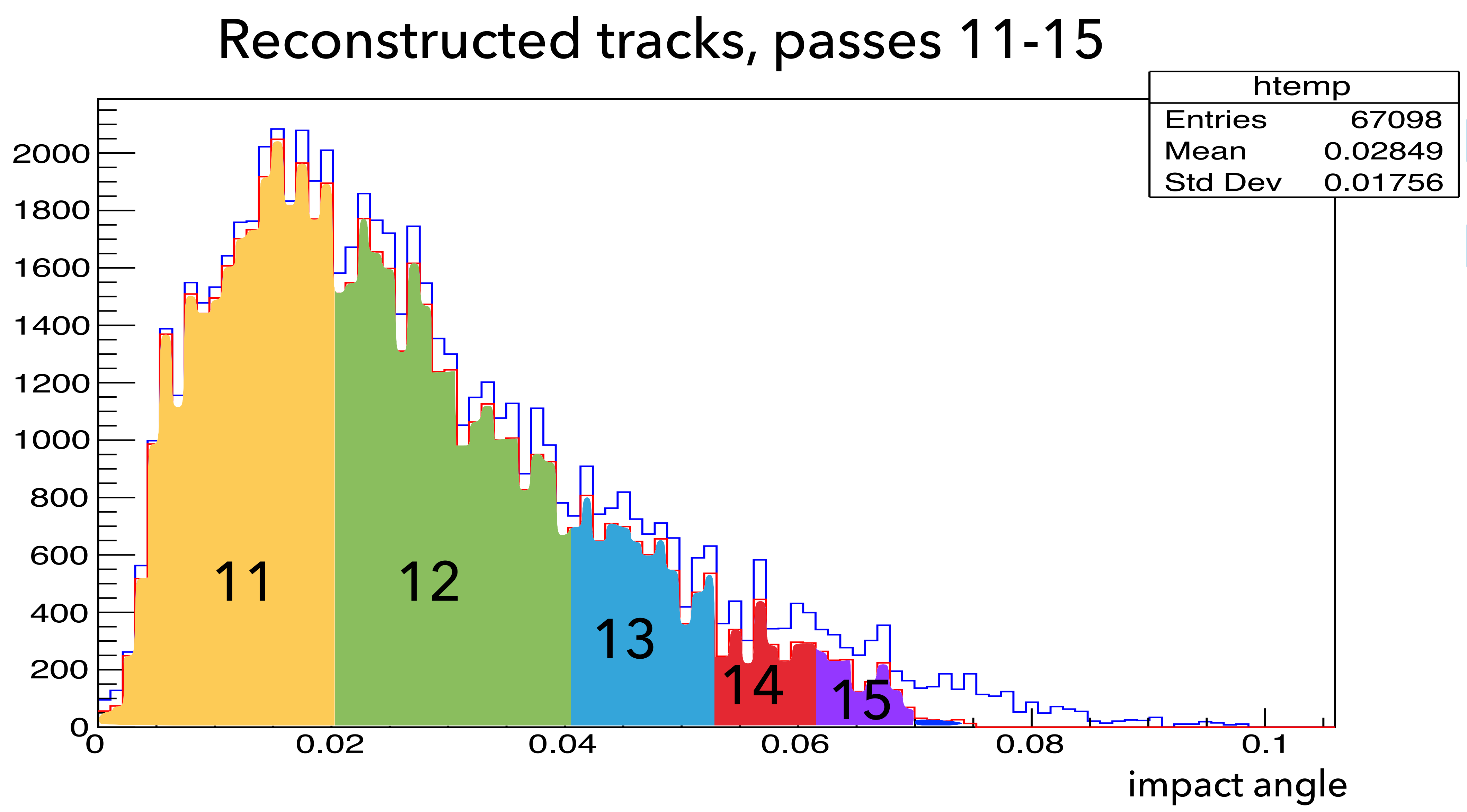
pass 1: fast tracks, cut at 8 hits

pass 2: slower tracks, cut at 8 hits

pass 3: fast tracks, cut at 7 hits

pass 4: slow tracks, cut at 10 hits

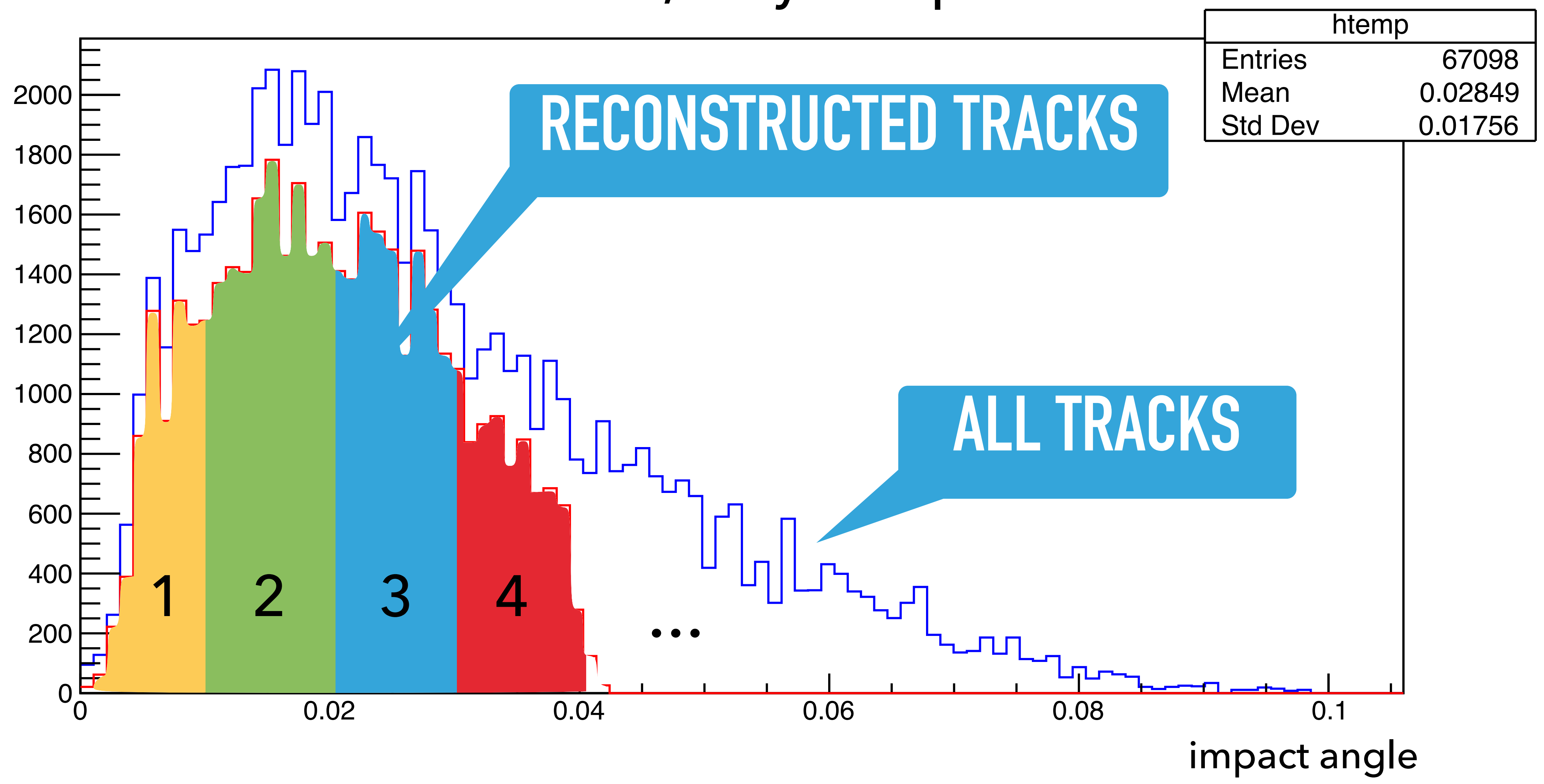




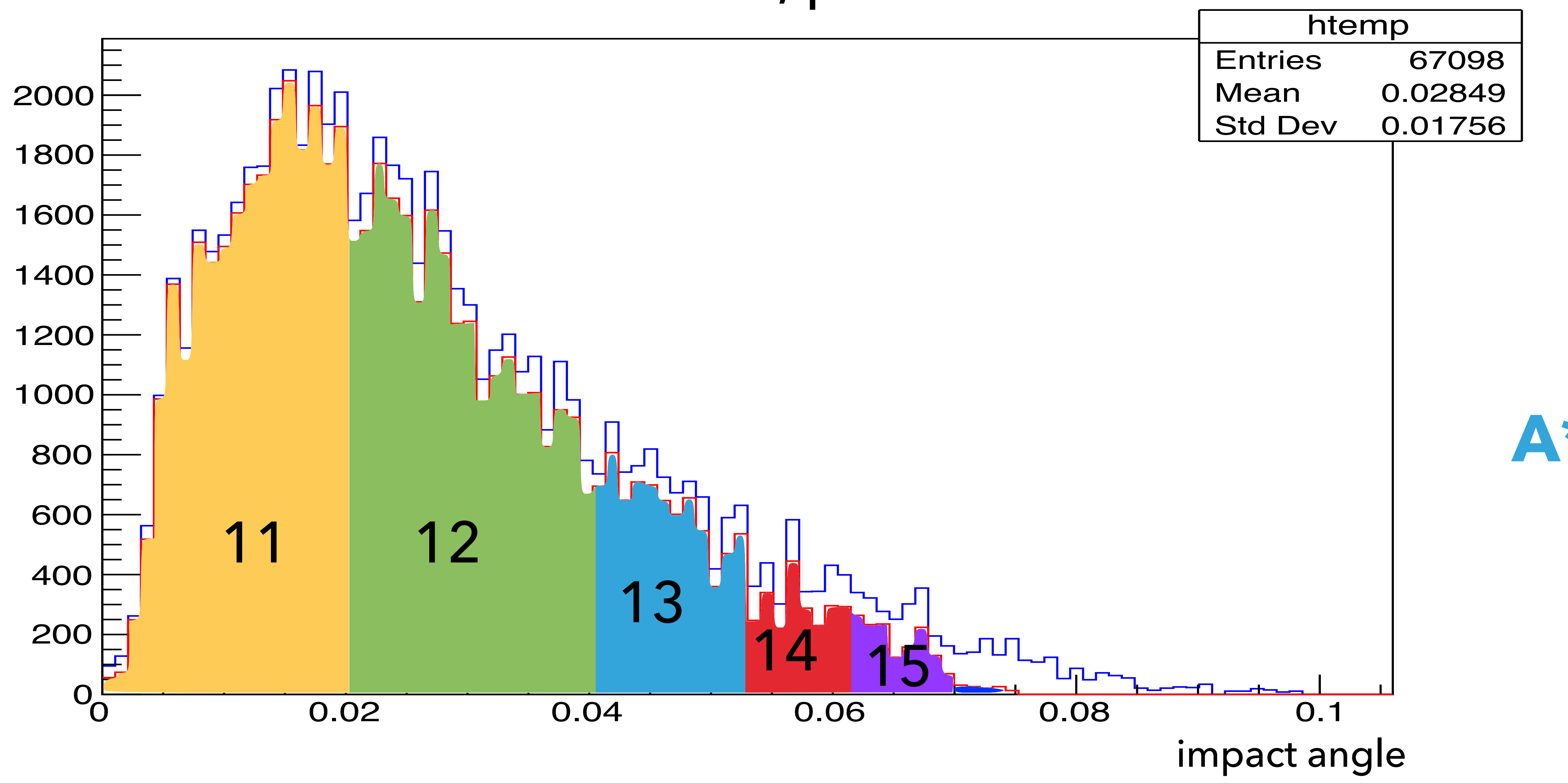
MIKADO STRATEGY

- Reconstruct data in small portions. It is very helpful for simple approaches to split the task into subtasks.
- Generic strategy, potentially applicable for any algorithm
- Combinatorial complexity remains constant among the passes
- 60 reconstruction passes with simple mathematics
- Each pass process little combinatorics, therefore is extremely fast

Reconstructed tracks, very first passes



Reconstructed tracks, passes 11-15



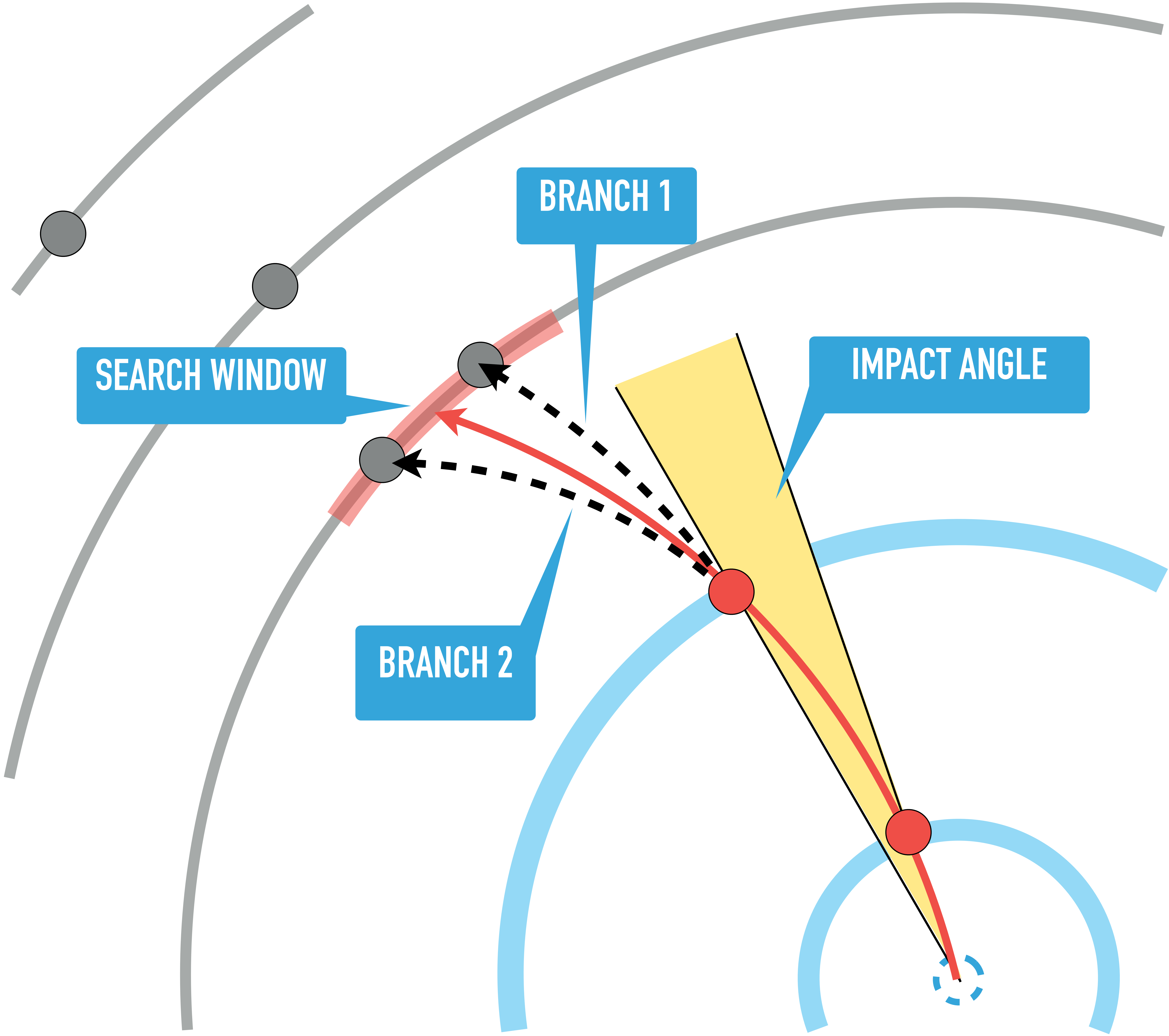
CALIBRATION OF CUTS (LEARNING)

- calibrate passes one by one, optimizing a partial result
- calibrate parameters one after another
- seeding layers and impact angle are fixed, search windows are calibrated
- optimise for purity, not for efficiency
- optimisation criteria tuned manually, the optimisation is done automatically by home-made algorithm
- variable mixture of true/noise hits
- 60x300 parameters in total - needs statistics

optimisation criteria:

A*w_reconstructed - B*w_lost - C*w_wrong_assigned -> max

GOOD NOT GOOD BAD

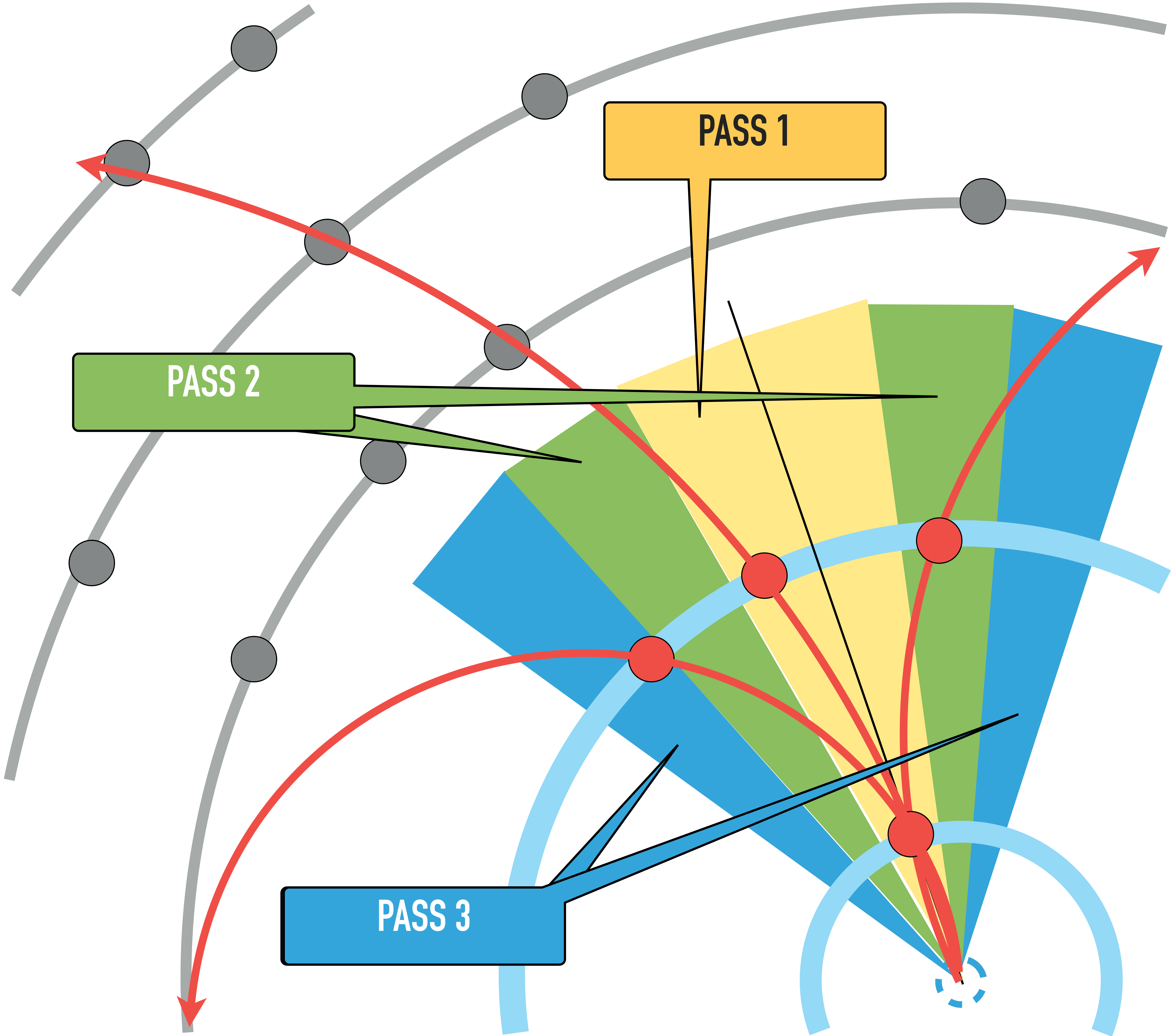


SEEDING LAYERS

- 2 fixed obligatory layers
- fixed angular cuts for the second hit
- try to create a track with all 2-hit combinations

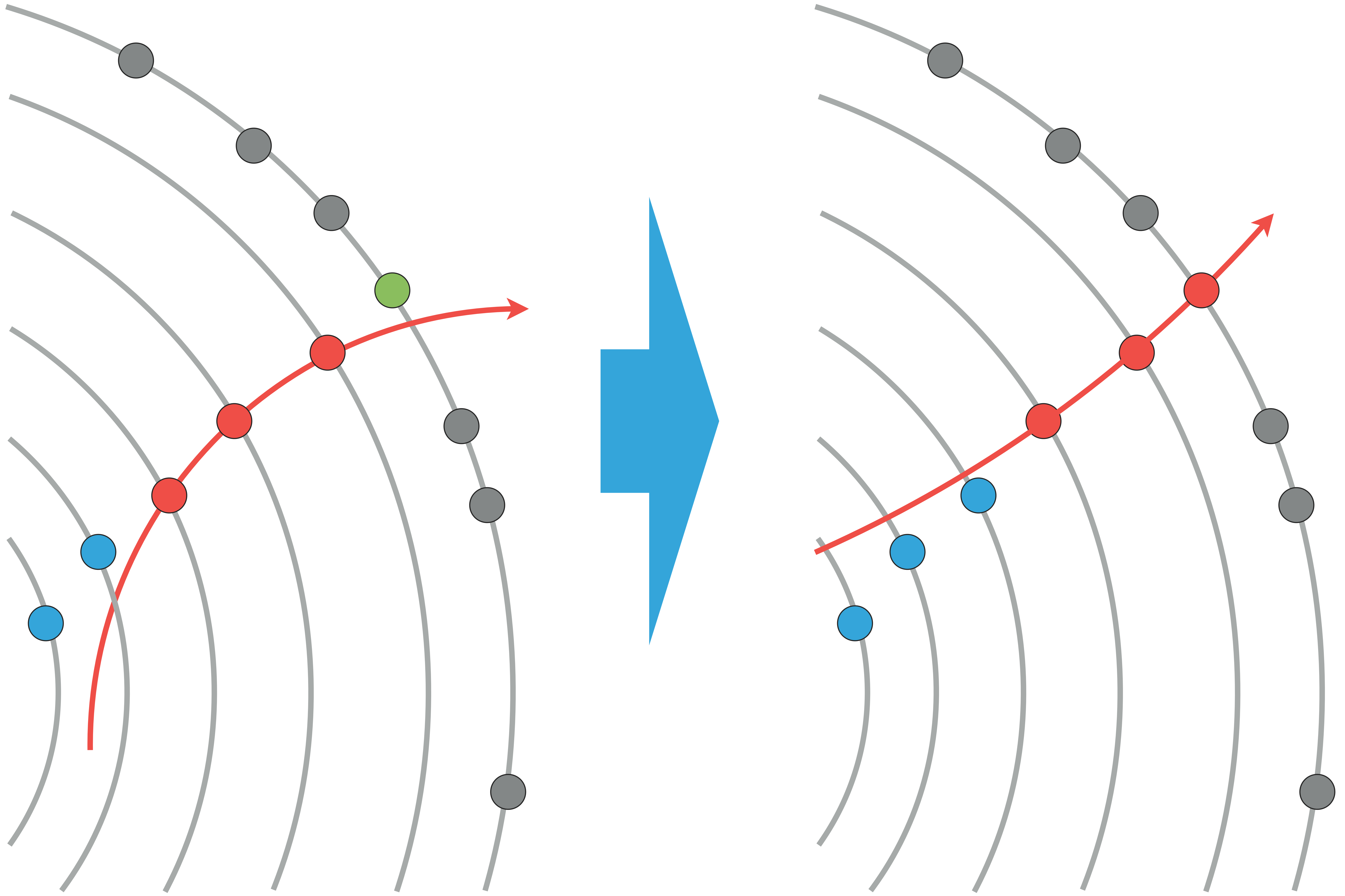
PROLONGATION LAYERS

- search in layers along the track
- fixed search windows
- make combinatorial branch at every hit (optionally: only one branch with the best hit)



RECONSTRUCTION PASSES

- we don’t process same combinatorics twice

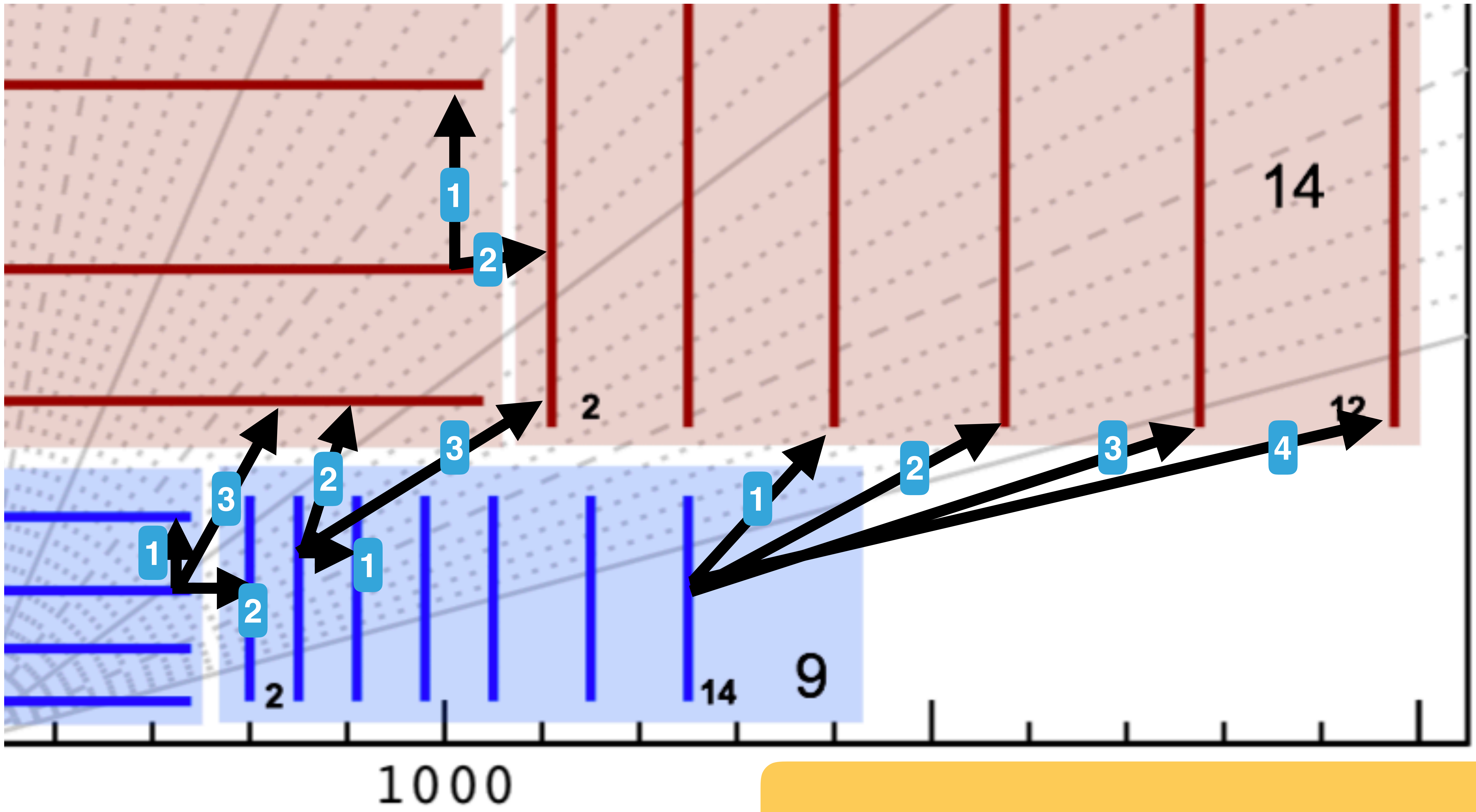


TRACK MODEL

- local 3-hit helix

TRACK PROLONGATION

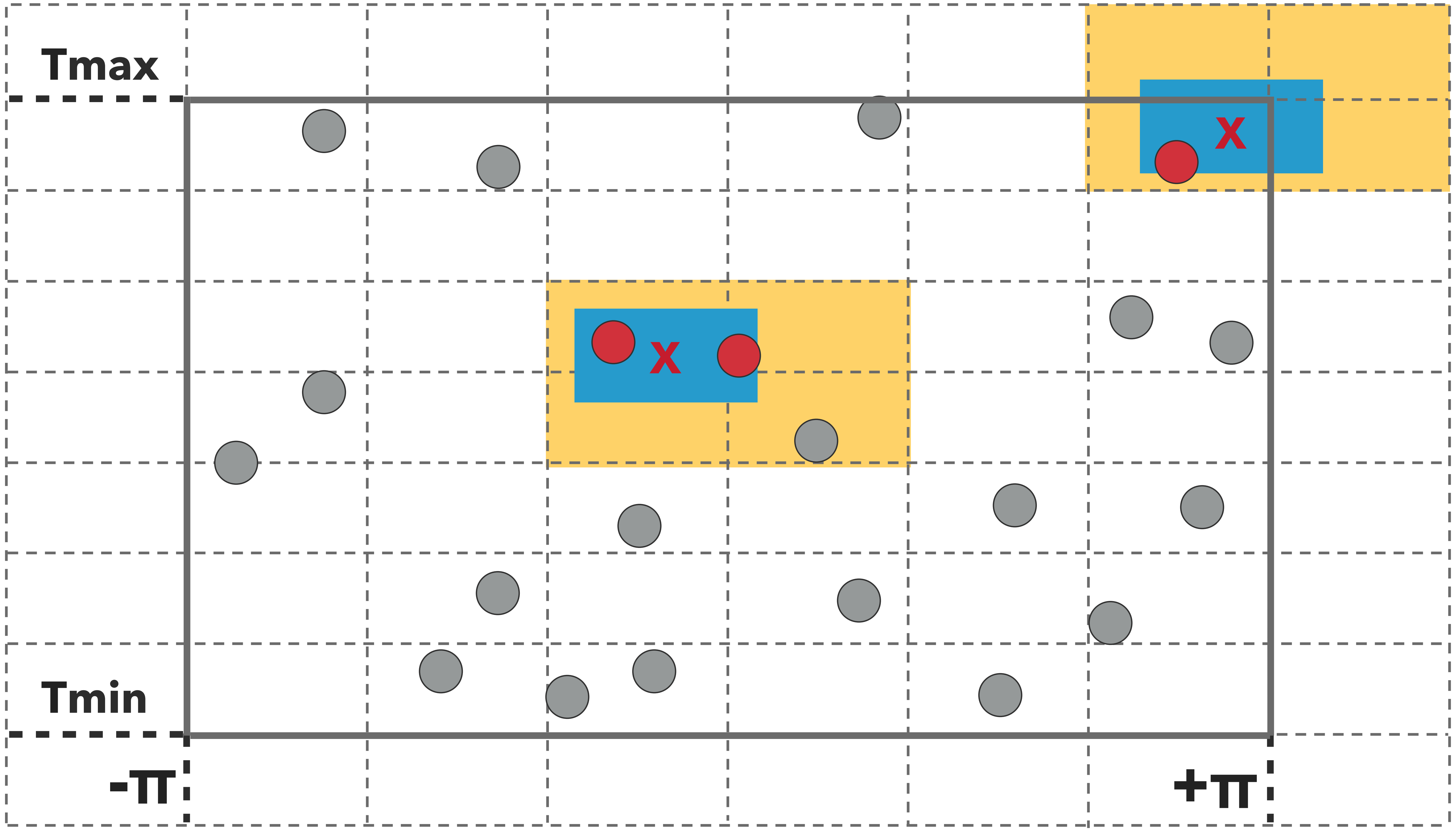
- cross the next layer
- pick a hit
- refit with the new hit



LAYER TOPOLOGY

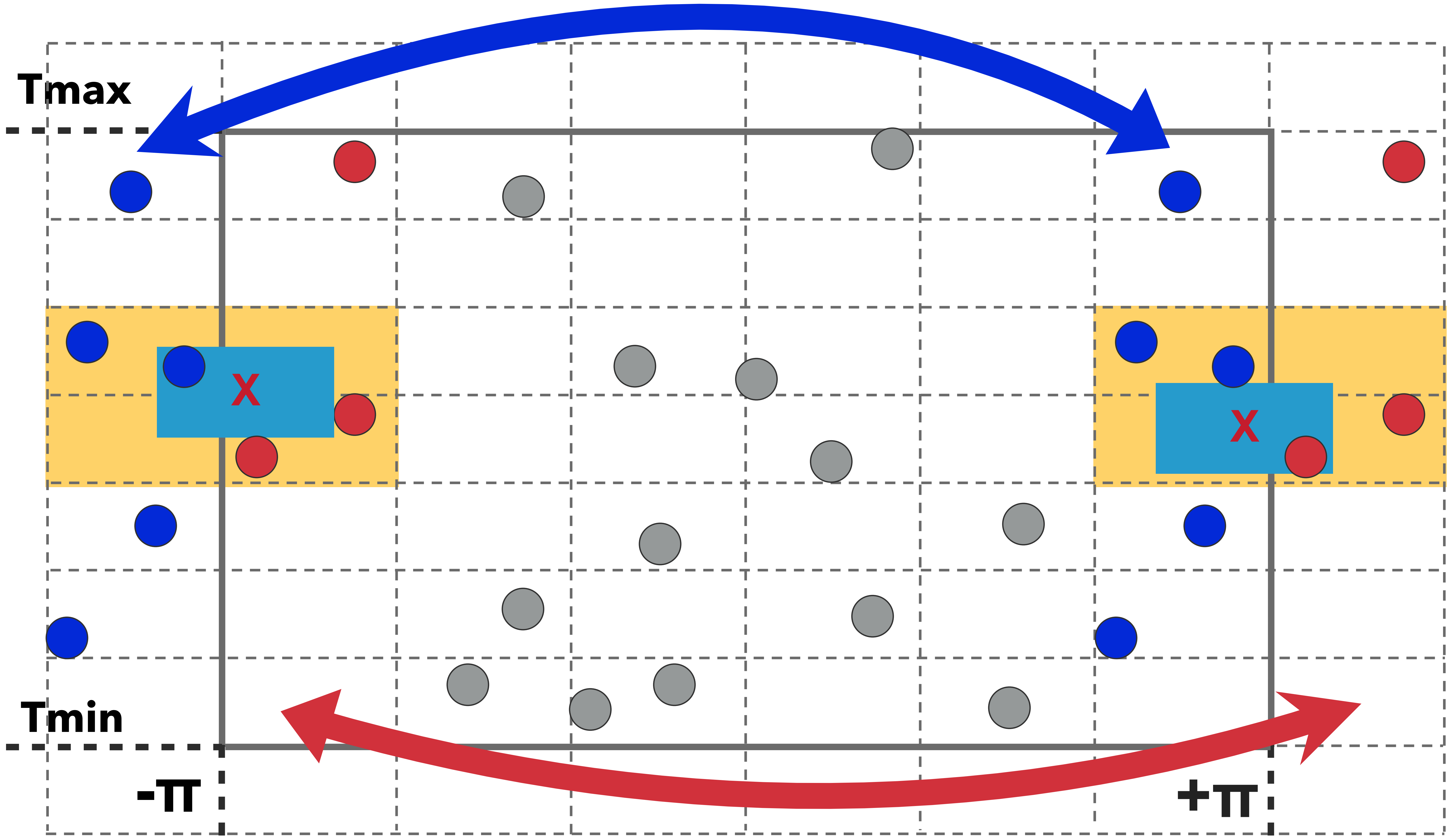
- define possible next layers according to the geometry
- prolongate to 2-nd neighbour ONLY when no intersection with the 1-st neighbour

The 3-rd place competitor (**cloudkitchen**) determines the event topology automatically from the ground truth data



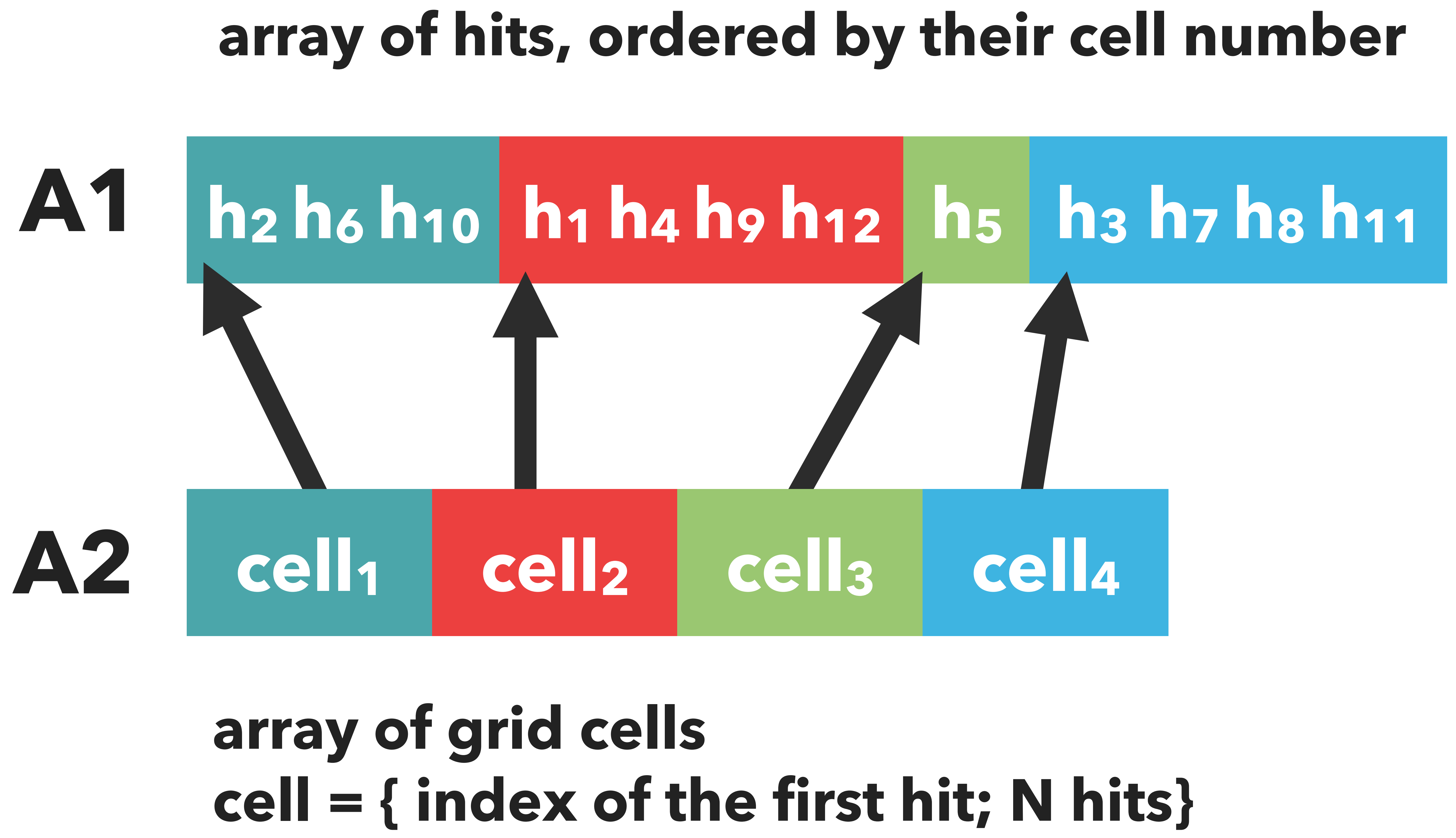
REGULAR 2D GRID

- the algorithm uses individual fixed-size search area on each detector layer
- grid spacing = search area
- the search is always performed in four grid cells
- fast calculation of the left-bottom search cell via modulo
- max N cells ≈ N hits
- empty rows around the layer



OVERLAPS IN Φ

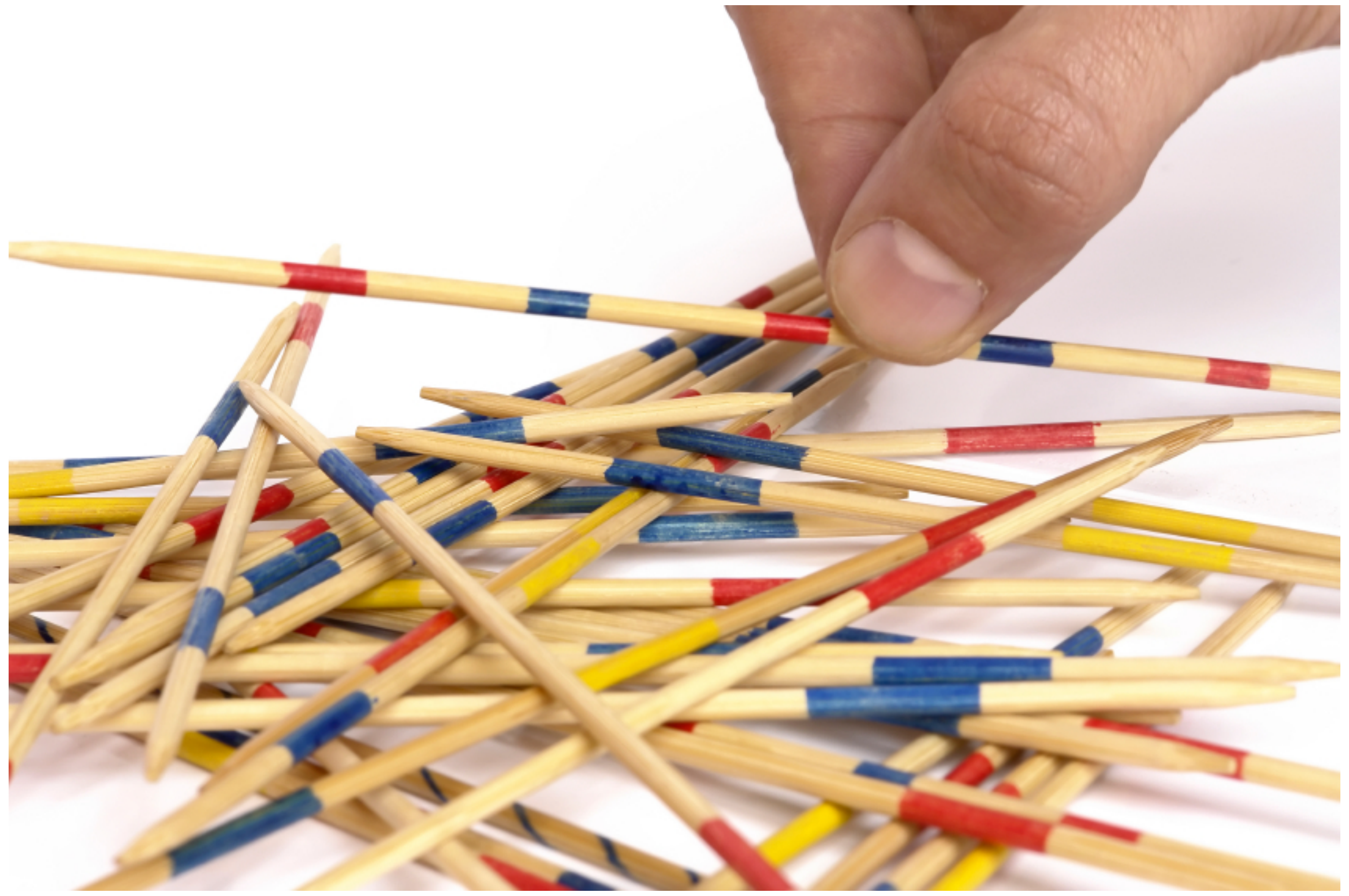
- two extra rows in φ with duplicated hits
- no special treatment of ±π regions

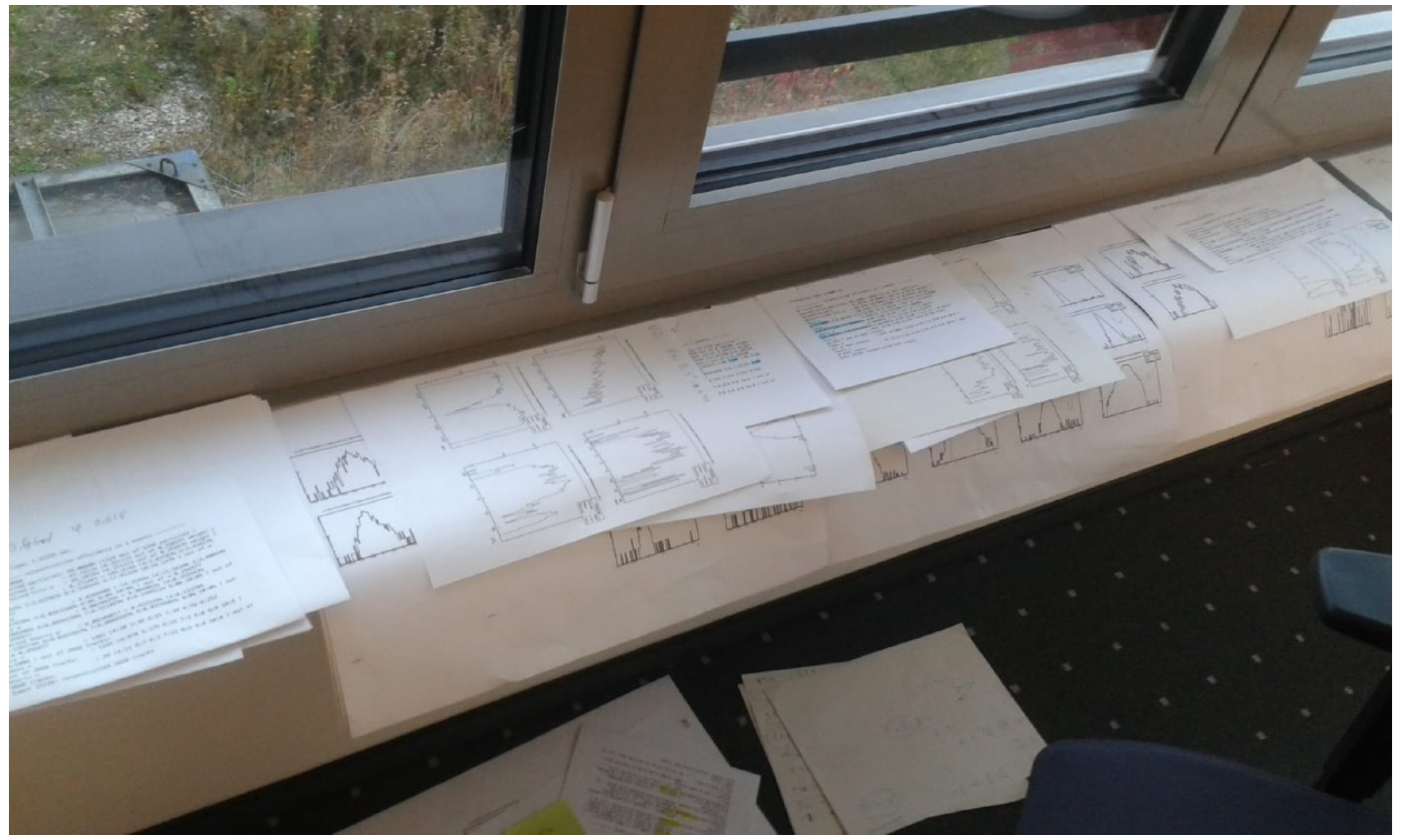


CREATION OF THE GRID

- loop over the hits: count n hits / cell
- loop over the cells: set pointers
- loop over the hits: copy hits to array

WHAT CAN BE IMPROVED





track selection criteria

- consider hit density around the track
- use the full track fit for the final selection
- full fit after 4-5 hits
- …

tuning of parameters

- external library for multi-dimensional optimization
- event-based parallelization on many computers
- automatically adjust Pt&angular ranges for the reconstruction passes

THE MIKADO APPROACH SUMMARY

- 1-st place
- 94.4% efficiency
- reconstruction time 0.56 sec / event

FUTURE OF THE TRACKML CHALLENGE

ACTS project: CERN official test case for tracking algorithms and hardware. Endless competition, open to everyone.