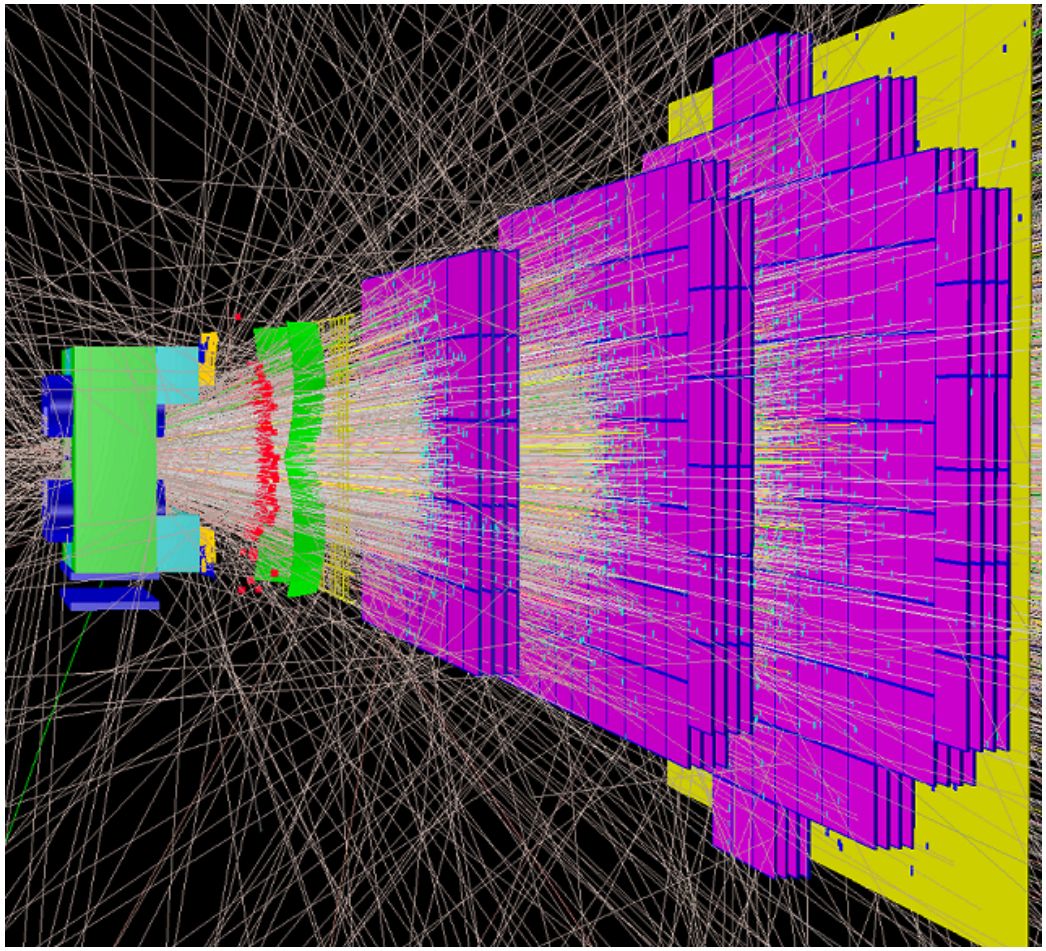
Event reconstruction algorithms for modern HEP experiments

Andrey Lebedev
LIT JINR / GSI IT
Alushta'14, June 2-7, 2014

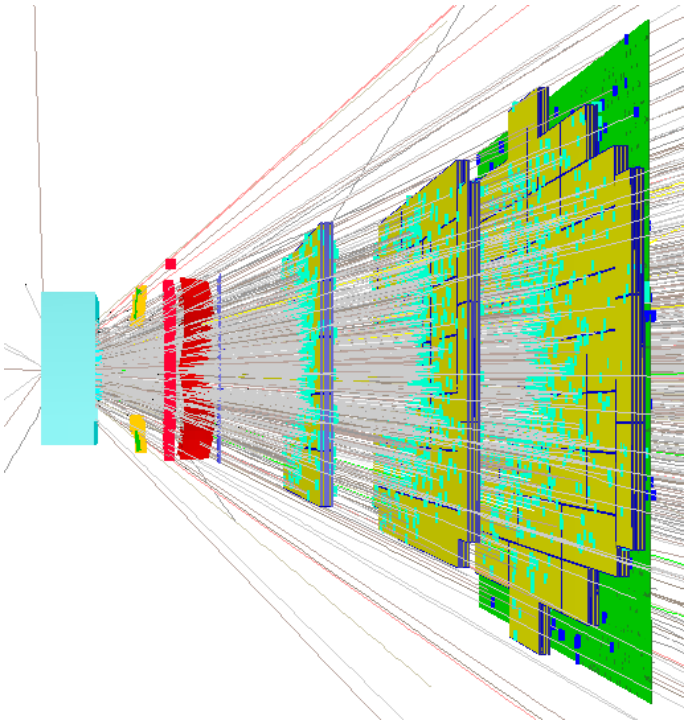
The problem



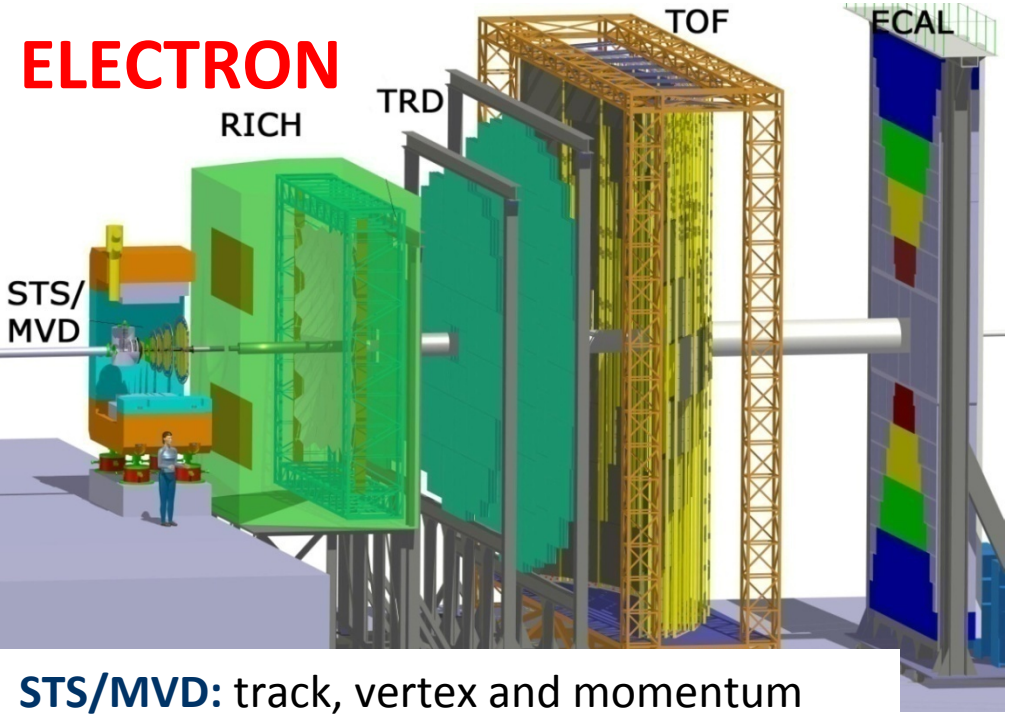
About 1000 charged particles in central Au+Au collision at a beam energy of 25 AGeV in the CBM acceptance.

What is event reconstruction?

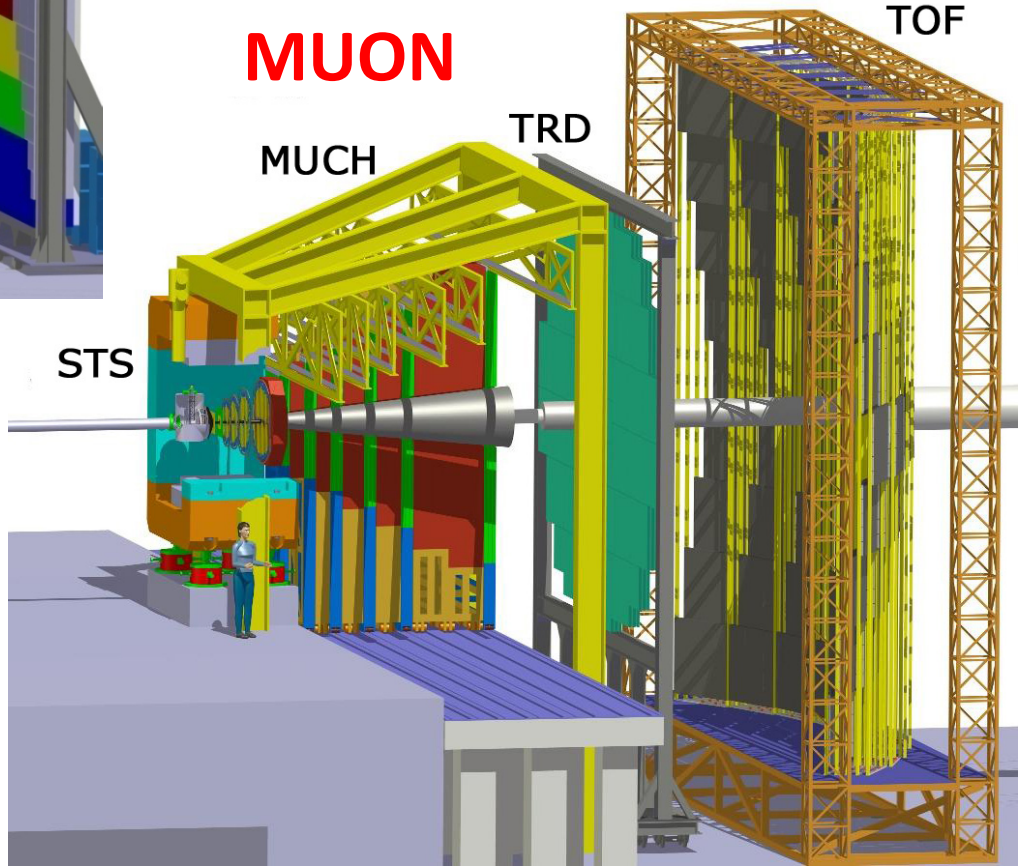
- The aim of the event reconstruction procedure is to describe a collision in terms of tracks and particles.
- In high energy physics experiments this is usually a large set of algorithms and software which has to be developed.
- It usually includes algorithms for:
  - Pattern recognition: tracking, vertexing, clustering, ring recognition etc.
    - Hough transform;
    - Kalman filter;
    - Cellular automata;
    - Neural networks;
  - Parameter estimation
  - Hypothesis testing
    - Likelihood;
    - Artificial neural networks;
    - Boosted decision trees;



CBM experiment



- comprehensive measurement of hadron and lepton production in *pp, pA* and *AA* collisions from **8-45 AGeV** beam energy
- fixed target experiment



STS/MVD: track, vertex and momentum reconstruction

RICH: electron identification **OR**
MUCH: muon identification

TRD: global tracking, electron identification

TOF: time of flight measurement for hadron identification

ECAL: photons and neutral particles

Track reconstruction

- Determine location, direction and momentum of charged particles

① Track propagation

- No magnetic field, magnetic field, detector material

② Track fit

- Kalman filter
- Least Squares Fit

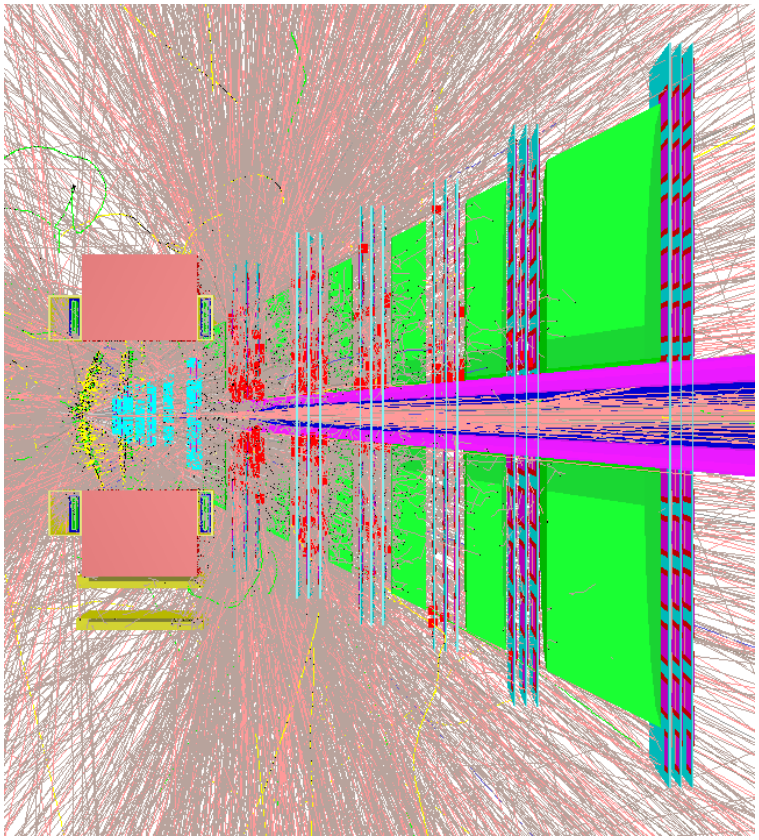
③ Track finding

- Conformal mapping
- Hough transform
- Artificial neural networks
- Elastic neural networks
- Cellular automaton
- Track following
- Kalman filter

Challenges for tracking

Peculiarities for CBM:

- large track multiplicities
  - up to **1000** charged particles per reaction in +/- 25˚ acceptance
- high reaction rate
  - up to **10 MHz**
- measurement of rare probes → need for fast and efficient trigger algorithms
- high hit density
- large material budget – especially in the muon detector
- complex detector structure, overlapping sensors, dead zones
- enormous amount of data: foreseen are currently 1Gb/s archiving rate (600Tb/week)



Central Au+Au collision at 25 *A*GeV (UrQMD + GEANT3)

→ fast tracking algorithms are essential

Kalman filter (KF)

The Kalman Filter is an efficient recursive filter that estimates the state of a linear dynamic system from a series of noisy measurements.

KF advantages:

- KF is suitable for combined track finding and track fitting;
- KF is fast in comparison to global LSF fit;
- KF is optimally suits to handle many measurements in presence of multiple scattering and other energy loss effects.

KF proceeds recursively by alternating two steps:

① Prediction
- Propagate state vector and covariance matrix to the next layer, increment covariance matrix by contributions from multiple scattering and energy loss.

② Update
- Compute a weighted mean of the extrapolation and the observation. The track parameters after each update step is the best estimate of the trajectory based on the measurements incorporated so far.

KF mathematics

The transport of a state at node k − 1 to a state at node k is described by the propagation relation:

$$\mathbf{x}_k = F_{k-1}\mathbf{x}_{k-1} + \mathbf{w}_{k-1} \qquad C_k = cov\{\mathbf{x}_k\}$$

$\mathbf{x}_k$ state vector in layer k (local track parameters)

F_{k-1} transport matrix (local track model)

$\mathbf{w}_{k-1}$ process noise (multiple scattering)

In case of fixed target experiment state vector it is convenient to define state vector as follows:

$$\mathbf{x}(z) = \left(x, y, t_x, t_y, q/p \right)^T \qquad t_x = \frac{\partial x}{\partial z},\ t_y = \frac{\partial y}{\partial z}$$

KF mathematics

Measurement equation:

$$m_k = H_k \mathbf{x}_k + e_k \qquad V_k = cov\{\mathbf{e}_k\}$$

m_k measurement in layer k

e_k

H_k measurement model

For example for pixel detectors which measure X and Y coordinates:

$$H_k = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \qquad V_k = \begin{pmatrix} d_{xx}^2 & d_{xy} \\ d_{xy} & d_{yy}^2 \end{pmatrix}$$

d_{xx}, d_{yy} и d_{xy} measurement errors

KF mathematics

Prediction

$$\mathbf{x}_k^{k-1} = F_{k-1}\mathbf{x}_{k-1}$$

$$C_k^{k-1} = F_{k-1}C_{k-1}F_{k-1}^T + Q_{k-1}$$

Prediction step equals to the track propagation.

Update

KF gain matrix:

$$K_k = C_k^{k-1}H_k^T\left(V_k + H_kC_k^{k-1}H_k^T\right)^{-1}$$

Update of state vector and cov matrix

$$\mathbf{x}_k = \mathbf{x}_k^{k-1} + K_k r_k^{k-1}$$

$$C_k = (1 - K_kH_k)\,C_k^{k-1}$$

Residual:

$$r_k = m_k - H_k\mathbf{x}_k = (1 - H_kK_k)\,r_k^{k-1}$$

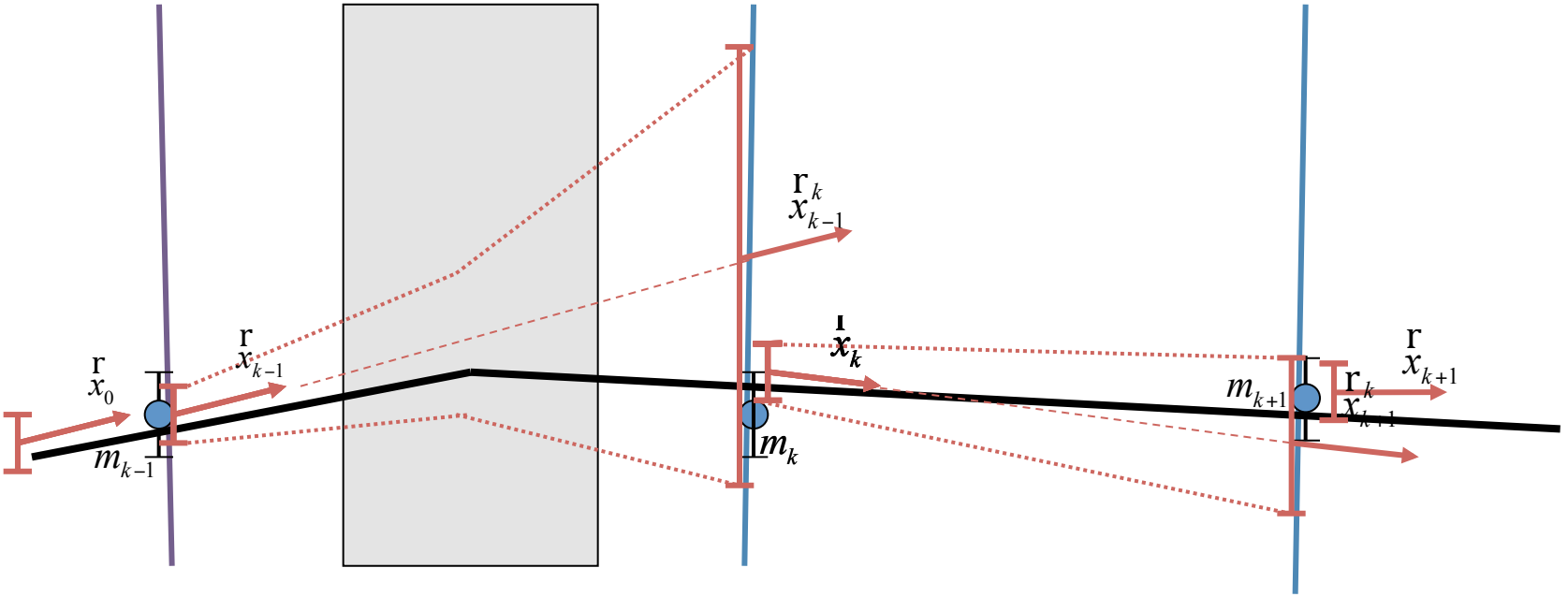
$$R_k = (1 - H_kK_k)V_k = V_k - H_kC_kH_k^T$$

Chi-square test:

$$\chi_+^2 = r_k^T R_k^{-1} r_k$$

$$\chi_k^2 = \chi_{k-1}^2 + \chi_+^2$$

Kalman filter illustration



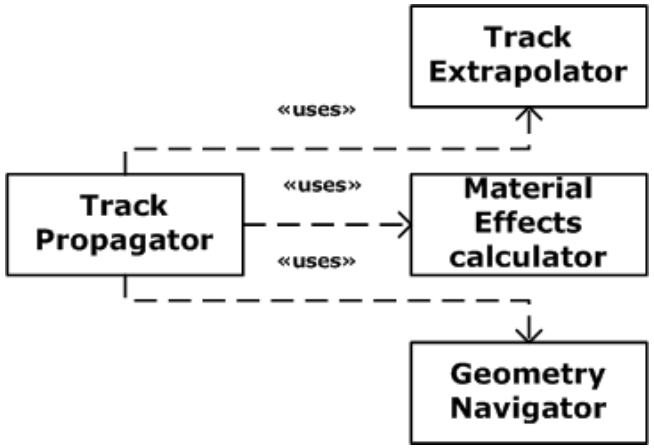
- Particle scatters in a material layer between layer k and k − 1.
- The track fit accounts for this effect by increasing the predicted error on the state vector x_{k-1} with Q^k_k.
- The measurement, m_k, pulls the state back to the true trajectory, resulting in a filtered track state, x_k.

Track propagation

The **track propagation algorithm** calculates the average trajectory and its errors in a covariance matrix while taking into account physics processes which may change the trajectory, i.e. energy loss, multiple scattering and also the influence of a magnetic field.

① Geometrical extrapolation according to the equation of motion for a charged particle.

② Calculation of the multiple scattering and energy loss for a particle passing through the detector material.



Geometrical extrapolation

- **Straight line** in magnetic field free regions.

$$x = x_0 + t_x \Delta z$$
$$y = y_0 + t_y \Delta z$$

$$F = \begin{pmatrix} 1 & 0 & \Delta z & 0 & 0 \\ 0 & 1 & 0 & \Delta z & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Geometrical extrapolation

- In a **magnetic field** the equation of motion for a charged particle is solved applying the 4th order Runge-Kutta method.

Transport of the state vector.

$$\frac{d\mathbf{x}}{dz} = \frac{d}{dz}\begin{pmatrix} x \\ y \\ t_x \\ t_y \\ q/p \end{pmatrix} = \begin{pmatrix} t_x \\ t_y \\ (kq/p)\,A_x \\ (kq/p)\,A_y \\ 0 \end{pmatrix}$$

$$A_x = \sqrt{1+t_x^2+t_y^2}\left(t_x t_y B_x - \left(1+t_x^2\right)B_y + t_y B_z\right)$$

$$A_y = \sqrt{1+t_x^2+t_y^2}\left(\left(1+t_y^2\right)B_x - t_x t_y B_y - t_x B_z\right)$$

Calculation of the covariance matrix.

$$F = \frac{\partial \mathbf{x}}{\partial \mathbf{x_0}}$$

This can be solved using numerical differentiation or integration along the “zero trajectory”. In the second method 4th order Runge-Kutta method can be used.

Adding material

– Correction of the mean values is only due to the energy loss

- ionization: Bethe-Bloch

$$\left(\frac{dE}{dx}\right)_{ionization} = -Kz^2\frac{Z}{A}\frac{1}{\beta^2}\left[\frac{1}{2}\ln\frac{2m_e c^2\beta^2\gamma^2 T_{\max}}{I^2} - \beta^2 - \frac{\delta(\beta\gamma)}{2}\right]$$

- bremsstrahlung: Bethe-Heitler

$$\left(\frac{dE}{dx}\right)_{bremsstrahlung} = -\frac{E}{X_0\rho}\left(\frac{m_e}{M}\right)^2$$

- Total energy loss

$$\Delta = \left(\frac{dE}{dx}\right)_{ionization} + \left(\frac{dE}{dx}\right)_{bremsstahlung} \qquad \Delta E = \Delta\rho l$$

$$E = E_0 + \Delta E = \sqrt{M^2 + p_0^2} + \Delta E \qquad q/p = q/\sqrt{E - M^2}$$

Adding material

- Corrections in the covariance matrix from multiple scattering:

Estimation of the scattering angle:

$$\Theta_0 = \frac{13.6MeV}{\beta pc} z \sqrt{\frac{l}{X_0}} \left[1 + 0.038 \ln \frac{l}{X_0} \right]$$

Corrections in the covariance matrix:

$$Q(\Delta z) = \begin{pmatrix} Q_{33}\frac{\Delta z^2}{3} & Q_{34}\frac{\Delta z^2}{3} & Q_{33}\frac{\Delta z}{2} & Q_{34}\frac{\Delta z}{2} & 0 \\ \dots & Q_{44}\frac{\Delta z^2}{3} & Q_{34}\frac{\Delta z}{2} & Q_{44}\frac{\Delta z}{2} & 0 \\ \dots & \dots & Q_{33} & Q_{34} & 0 \\ \dots & \dots & \dots & Q_{44} & 0 \\ \dots & \dots & \dots & \dots & 0 \end{pmatrix}$$

where

$$Q_{33} = (1 + t_x^2)(1 + t_x^2 + t_y^2)\Theta_0^2$$

$$Q_{44} = (1 + t_y^2)(1 + t_x^2 + t_y^2)\Theta_0^2$$

$$Q_{34} = t_x t_y (1 + t_x^2 + t_y^2)\Theta_0^2$$

Put it all together

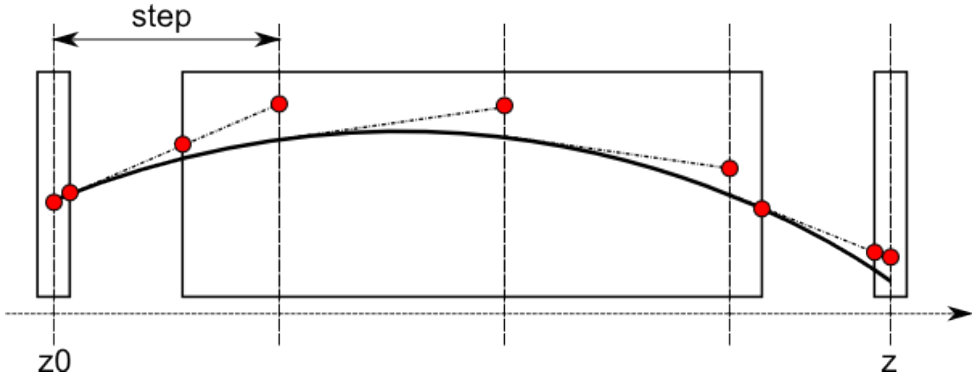
- In order to include material effects correctly in the track propagation algorithm one needs to identify which materials in the complex detector geometry have been traversed by the particle.
  - This is done by the geometry navigation algorithm which finds intersection points with detector elements in a certain interval (*z*1,*z*2) along a straight line.
  - Implementation can be based on ROOT TGeo package.

The Algorithm:

Trajectory is divided into steps. For each step:

– – · Straight line approximation for finding intersections with different materials (geometry navigator)

——— Geometrical extrapolation of the trajectory



Material effects are added at each intersection point

Track fit QA

Usually the correctness of track propagation and track fit algorithms is determined using residual and pull distributions.

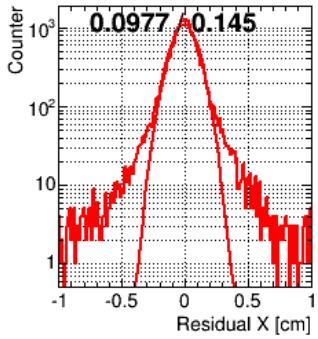
$$residual_x = x_{MC} - x_{rec}$$

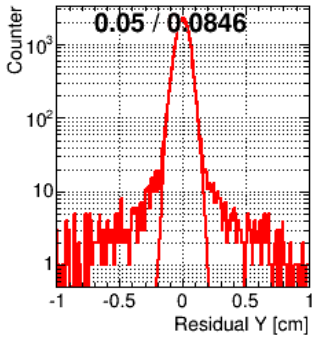
Residual shows the difference between MC and estimated track parameter.

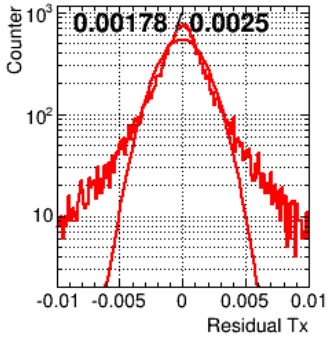
$$pull_x = \frac{x_{MC} - x_{rec}}{\sigma_x}$$

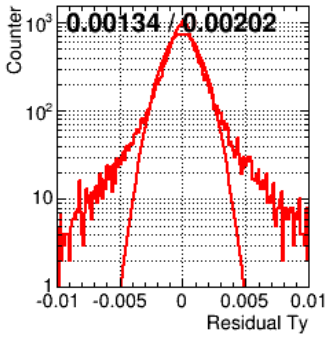
Pull shows the correctness of the error estimation. The variance has to be 1, mean has to be 0.

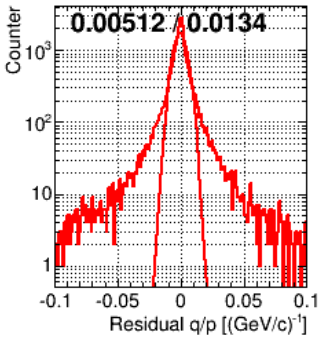
Example from CBM TRD

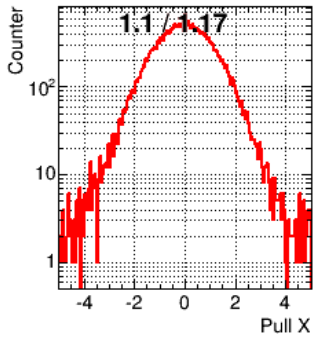


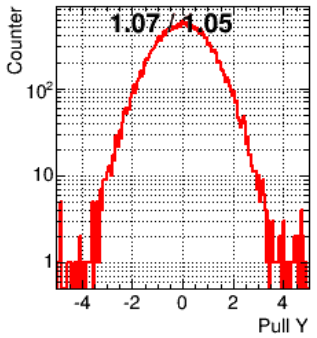


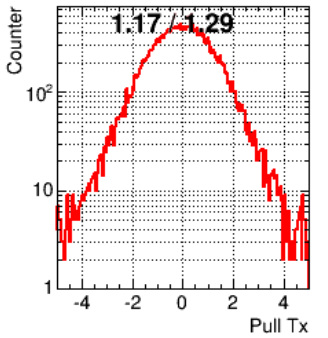


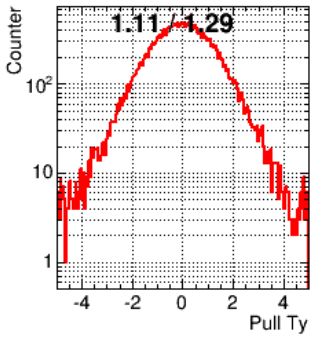


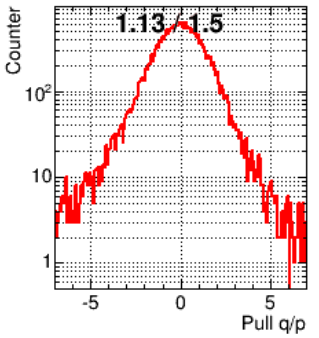




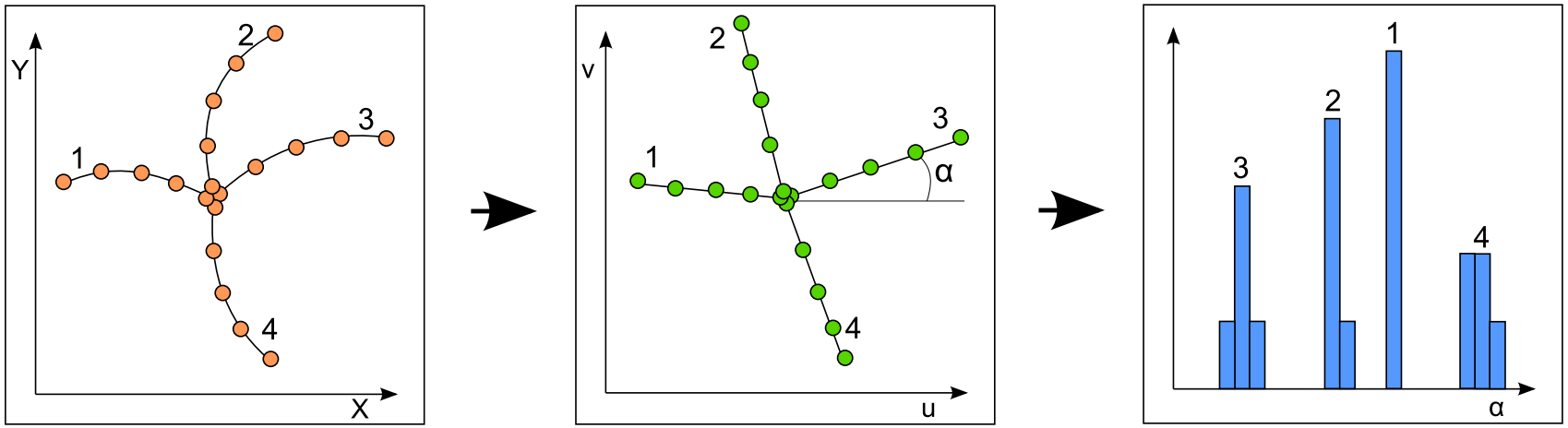








Conformal mapping

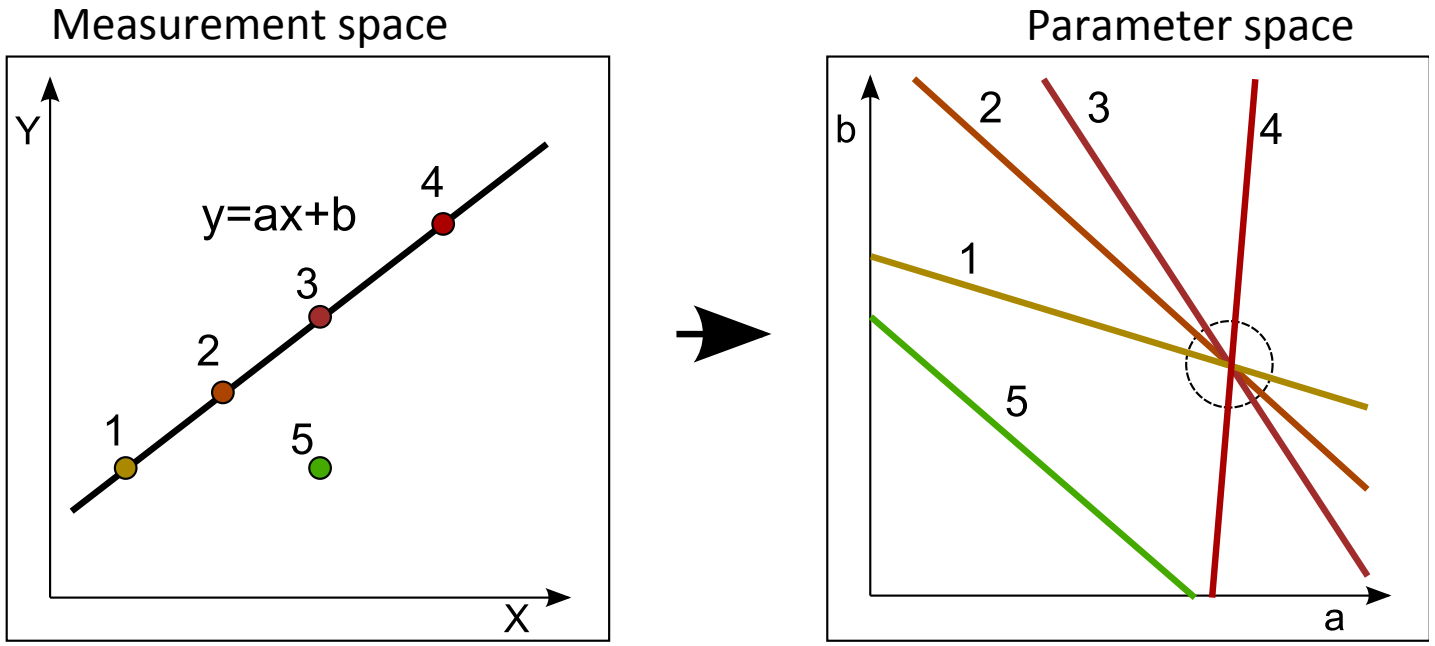


- Conformal mapping is used to simplify track topology.
- For example in collider experiment with a solenoid, where tracks are circular trajectories.
  1. **Conformal mapping.** Transform circles into a straight lines: $u = x/(x^2+y^2)$; $v = -y/(x^2+y^2)$.
  2. **Histogramming.** Collect histogram of azimuth angles, find peaks in the histogram, collect hits into tracks.

Hough transform

The Hough Transform is a standard method for curve recognition in digital images.

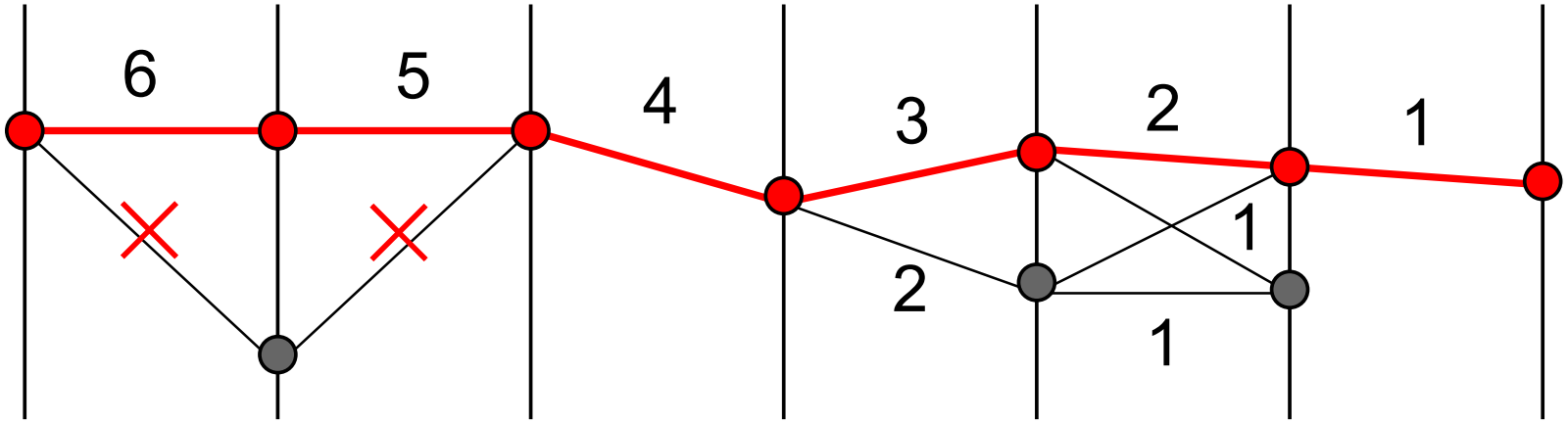
Straight line example



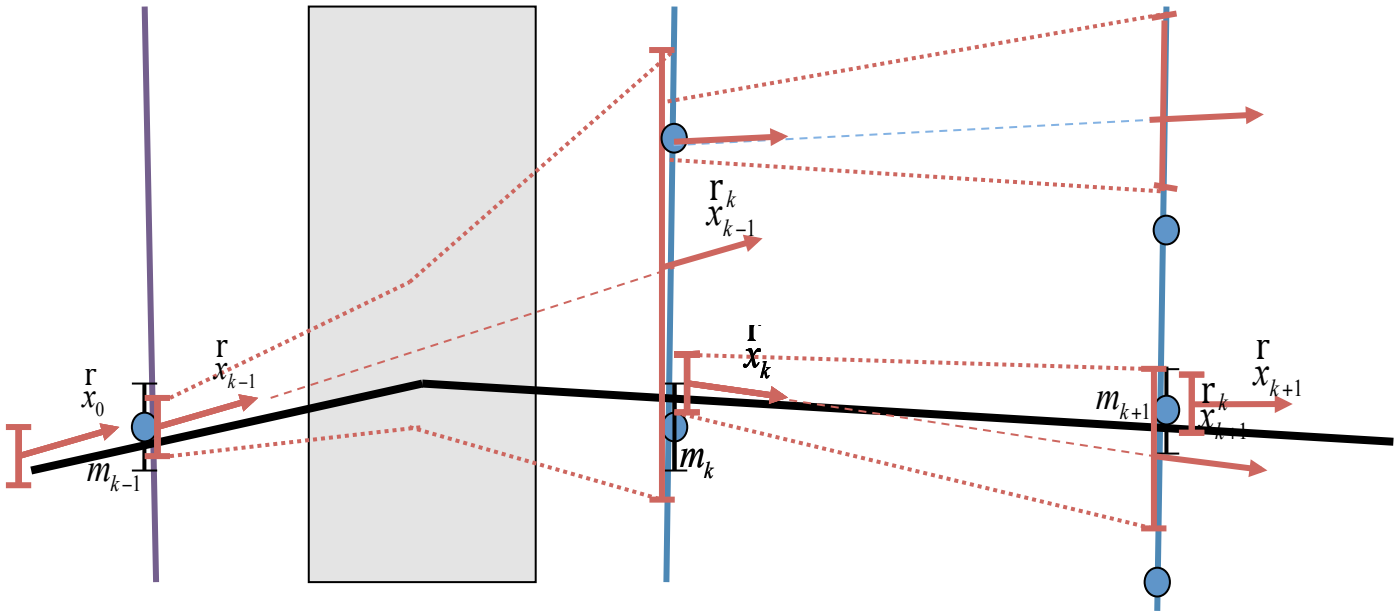
- A point (x_0,y_0) in the measurement space (XY) is transformed into a straight line $b = y_0 - ax_0$ in parameter space (AB).
- Points on the straight line $y = a_0x + b_0$ in measurement space are transformed into lines intersecting in the point (a_0, b_0) in parameter space
- If parameter space is discretized, intersections of lines can be found by histogramming.

CA tracking

- Track finding: Cellular Automaton method (I.Kisel)
  - Game “Life”
  - Algorithm:
    - Create segments between hits based on the track model
    - Switch from hits to segments
    - Find neighboring segments (adjacent segments which can belong to one track)
    - Count segment belonging to the same track
    - Create track candidates
    - Create tracks
- Track fitting: Kalman filter



Track following



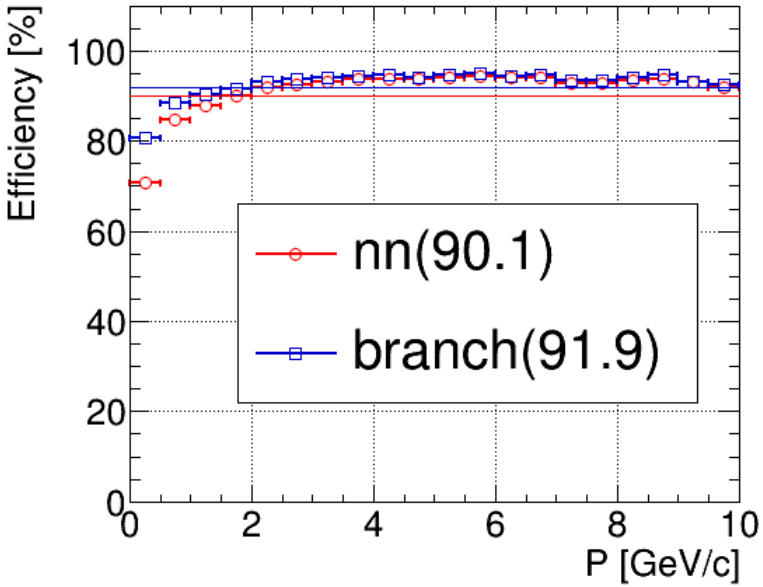
- Kalman Filter (KF)
- Validation gate
- Two hit-to-track association techniques
  - nearest neighbor: attaches the closest hit from the validation gate
    - Fast and easier to implement
  - branching: creates branch for each hit in the validation gate, select the best branch at the end
    - Slower due to much higher combinatory but more efficient in some cases

Track finding QA

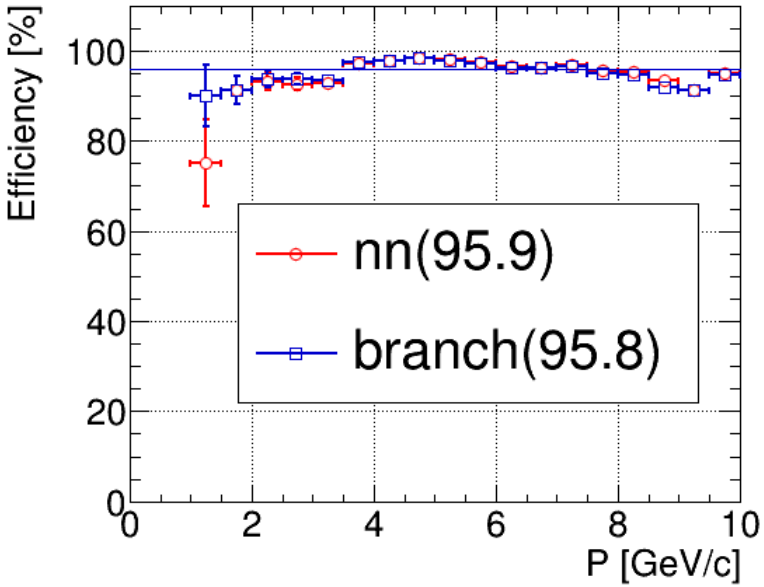
For the efficiency calculation the level of the correspondence between found and simulated tracks is estimated.

Tracking efficiency = (number of reconstructed tracks) / (number of tracks in the acceptance)

Track reconstruction in CBM TRD



Muon track reconstruction in CBM MUCH

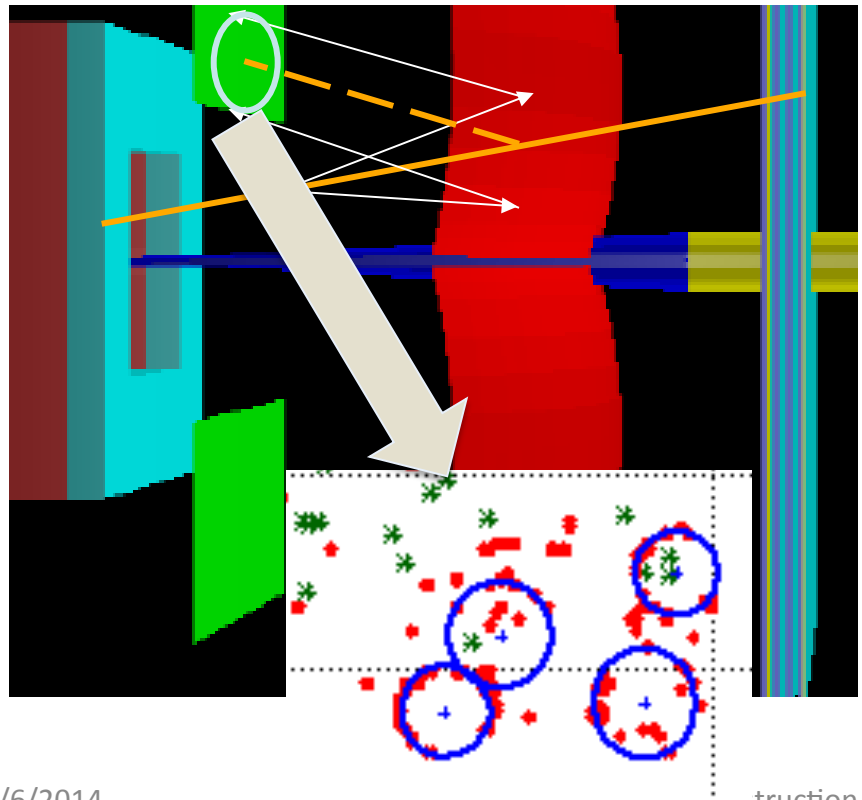


Event reconstruction in RICH

RICH = Ring Image CHerenkov

Whenever a charged particle passes through a medium characterized by a refractive index n, with a velocity v that exceeds the speed of light in that medium, Cherenkov radiation is emitted under a constant angle θ to the particle track:

$$\cos\theta = \frac{1}{\beta n} \quad , \text{ where } \beta = \frac{v}{c}.$$

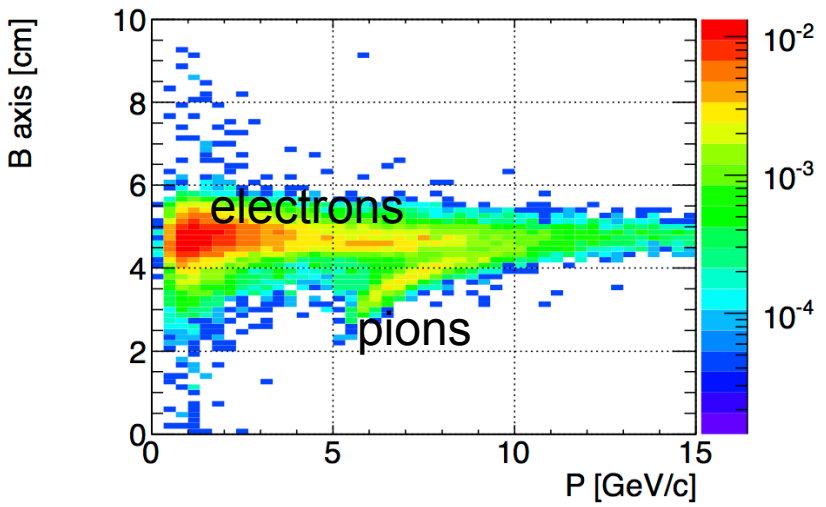


✓Event reconstruction includes:

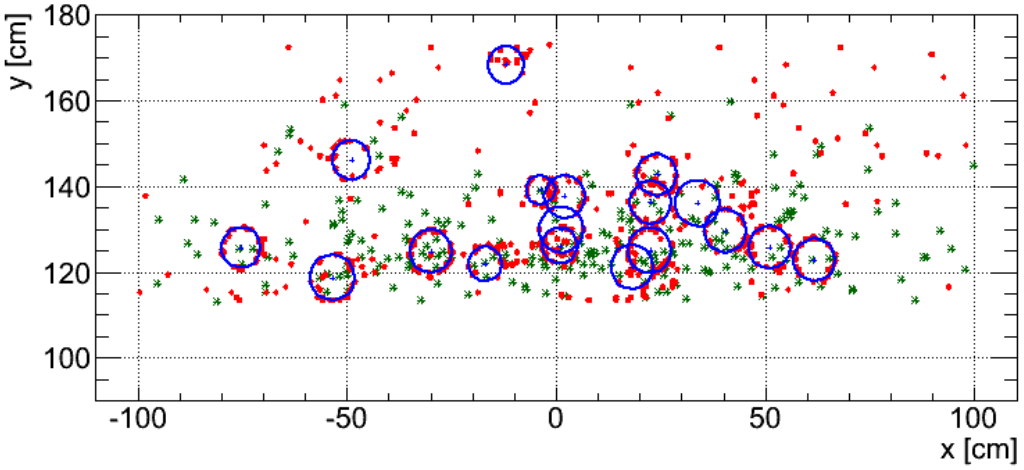
① Recognition of Cherenkov rings

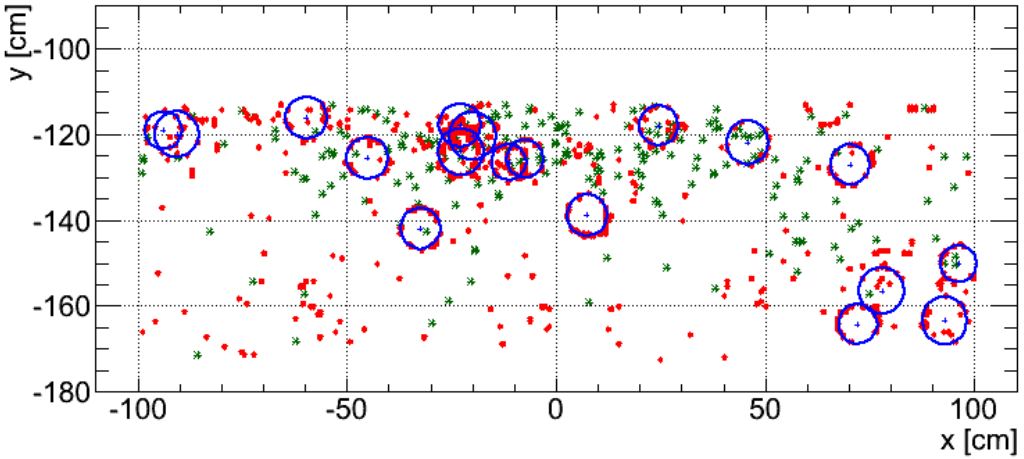
② Parameter estimation

③ Particle identification



Example of the RICH event



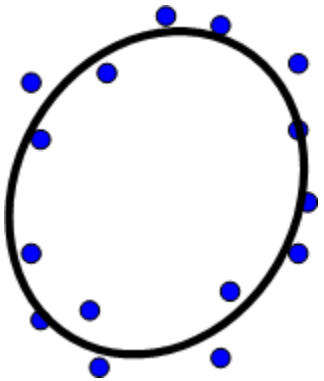


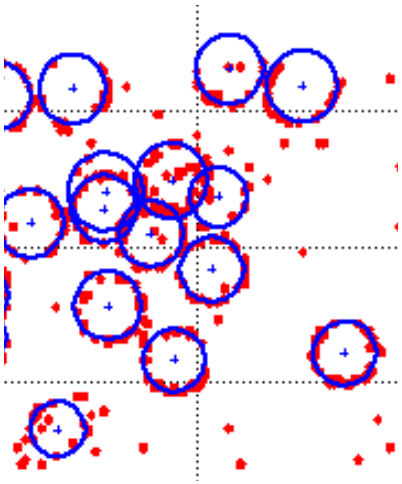
Red – RICH hits
Blue – reconstructed rings
Green – track projections

Central Au-Au collision at 25 AGeV

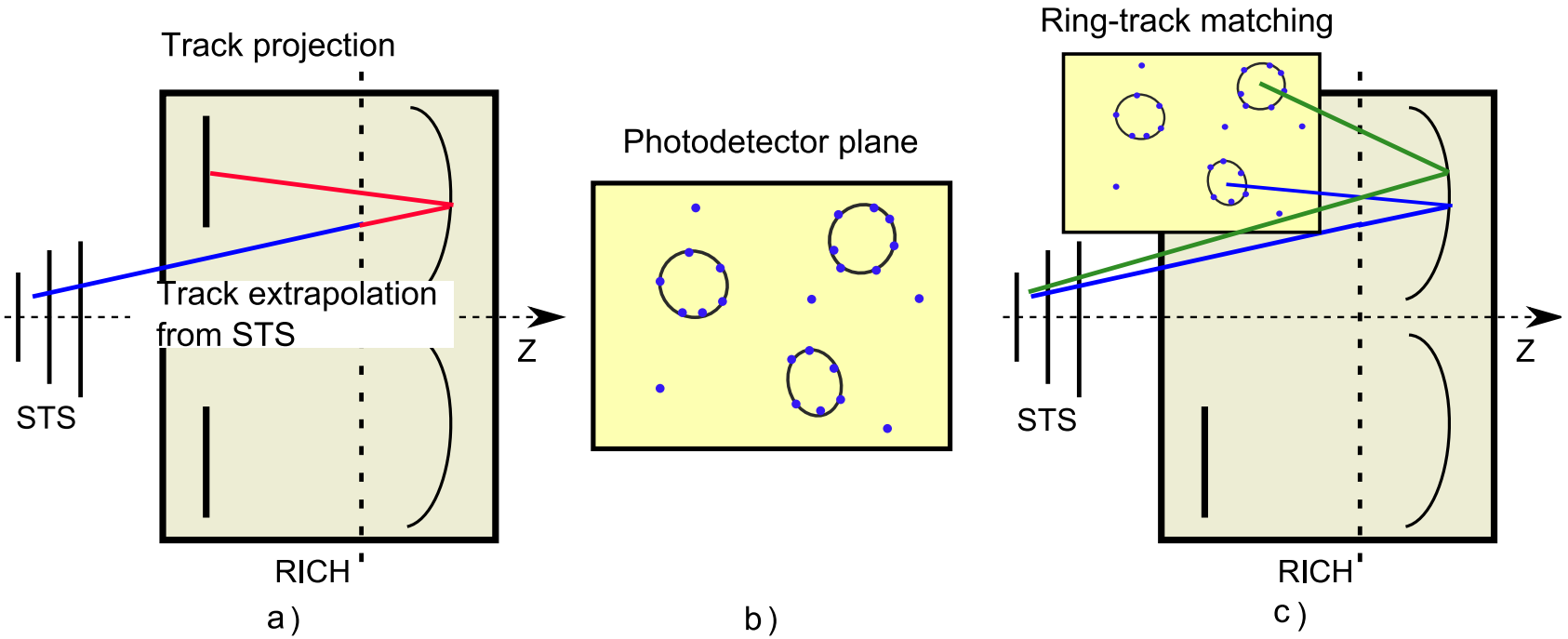
Challenges

- **Large number of hits in each event** including noise hits (around 1100 hits / event)
- **High ring density**, especially in the inner part of the photodetector, overlapping rings
- **Different number of hits per ring** (from 5 to 45) -> hard to reconstruct rings with small number of hits
- **Elliptic shape of the rings** (mean B/A = 0.9)
- **Fuzzy ring shape** -> optical distortions, photodetector granularity (0.58x0.58cm^2 around 10% of ring radius), residue of magnetic field
- **High interaction rate** -> algorithm must be fast.



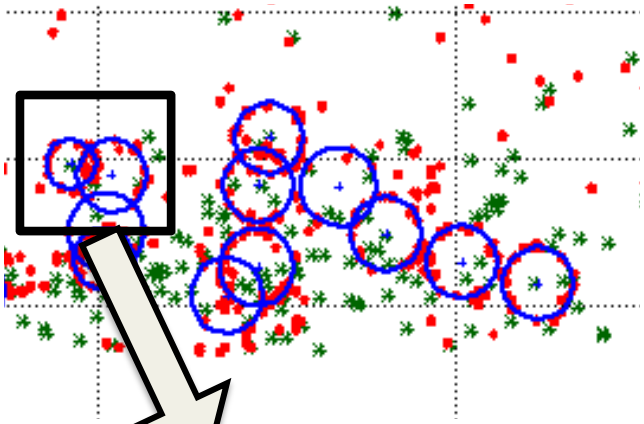


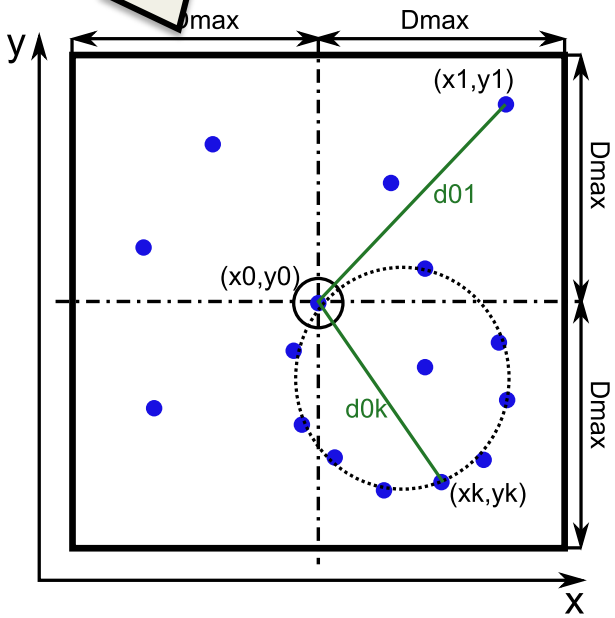
Reconstruction in the CBM RICH



- a) schematic view of the STS and RICH detectors, STS track extrapolation and projection onto the photon detector plane
- b) the photon detector plane with hits and reconstructed rings
- c) RICH rings and STS tracks matching

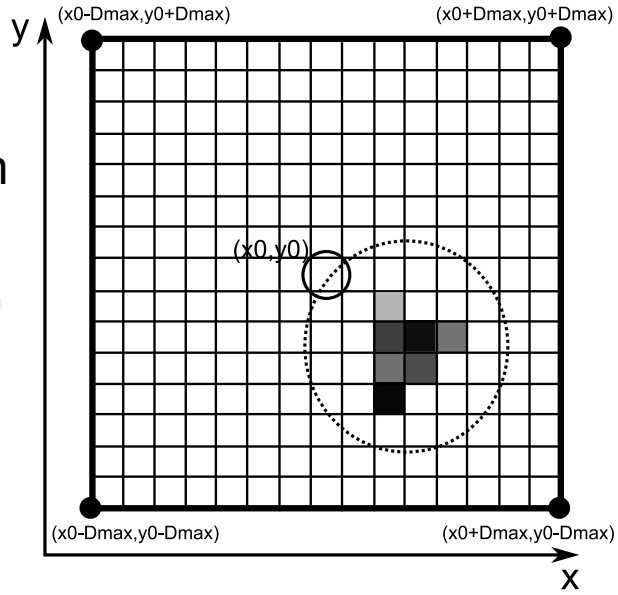
Ring recognition





Hough Transform:
large combinatorics => slow
Localized Hough Transform:
much less combinatorics => fast

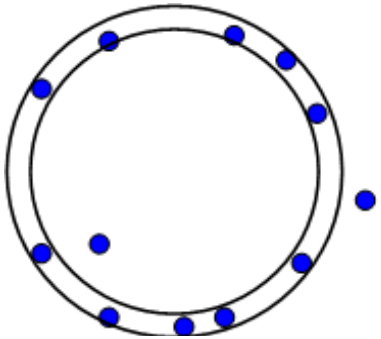
Hough Transform

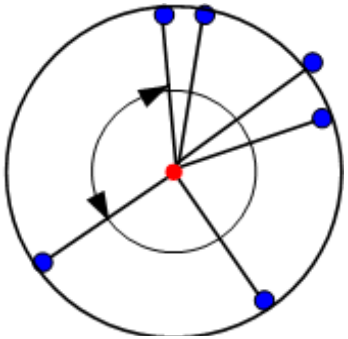


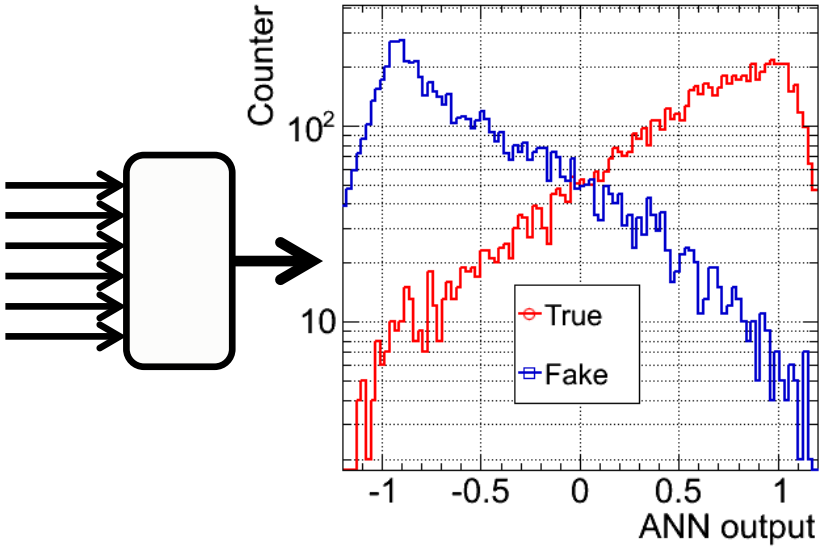
Ring selection

Ring quality parameters:

1. **number of hits in ring;**
2. **chi-squared;**
3. **number of hits in a small corridor around the ring;**
4. **biggest angle between neighboring hits;**
5. **position of ring on photodetector plane;**
6. **radius**







- ***Artificial Neural Network (ANN)*** derives ring quality from six parameters.
- The ANN output provides a ring quality parameter whether ring-candidate was found correctly or not.

- ✓ **Final selection of rings from ring-candidates to the output array**
- ✓ **Check for shared hits between ring-candidates**
- ✓ **Reject candidate with worse quality if it shares more than 25% of hits with a ring candidate with better quality.**

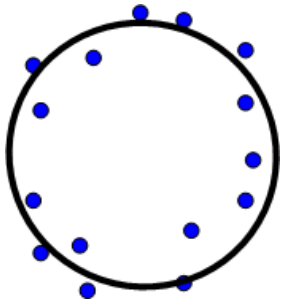
Circle and ellipse fitting

Circle fitting – ring finding

- **program realization of the COP (Chernov-Ososkov-Pratt), based on the minimization of the functional**

$$\bar{M}(a,b,R) = \sum_{i=1}^{n} \left[\left((x_i - a)^2 + (y_i - b)^2 - R^2 \right)^2 / 4 * R^2 \right]$$

- **Newton method for nonlinear equations with one variable is used**
- **3-4 iterations**
- **algorithm is very robust to the initial parameters**



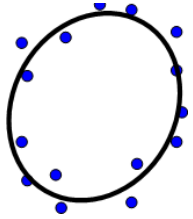
Ellipse fitting – final ring fitting

Rings in the photodetector plane have a slight elliptic shape

Conic section

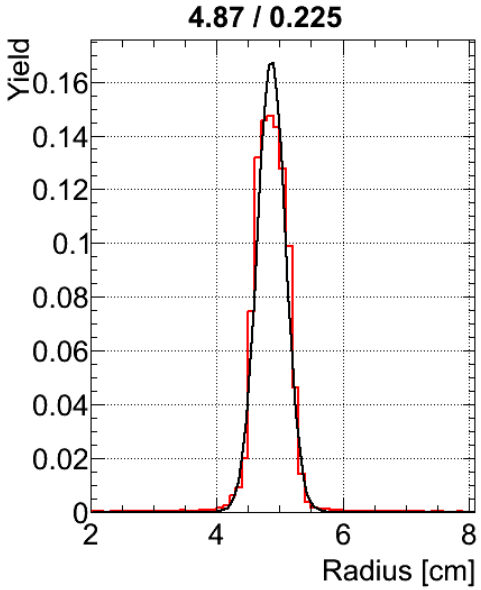
$$P(\mathbf{X}) = Ax^2 + Bxy + Cy^2 + Dx + Ey + F$$

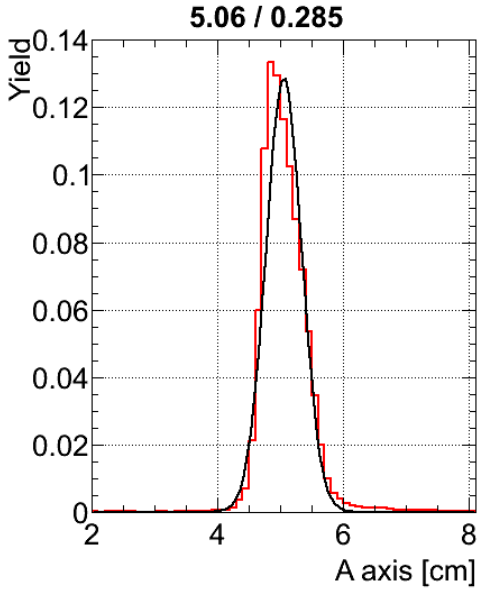
- **Taubin method is used**
- **Minimize P(x) by A,B,C,D,E,F, but measuring deviations along normals to the curve.**
- **non-linearity is avoided by Tailor expansion**
- **non-iterative very fast direct algorithm**
- **no need of start values for parameters**

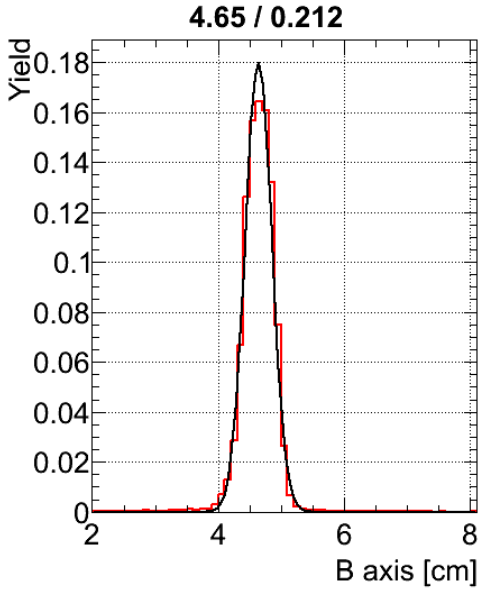


Mean B/A for CBM RICH rings ***= 0.9***

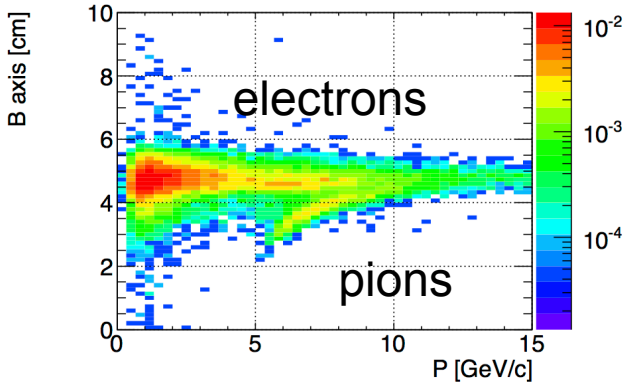
Results of ring fitting



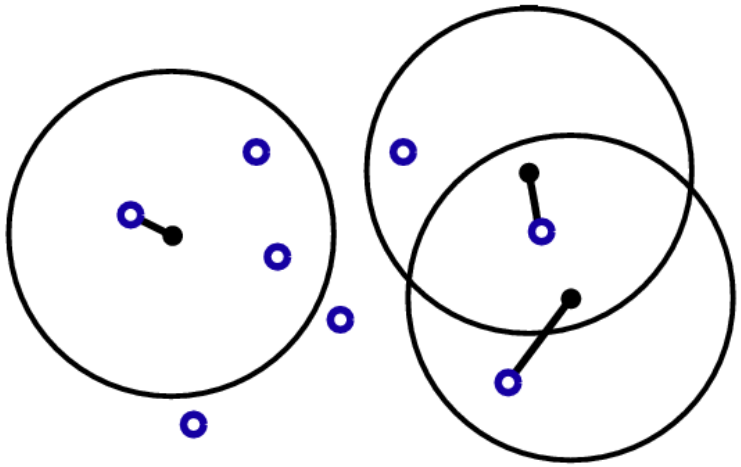


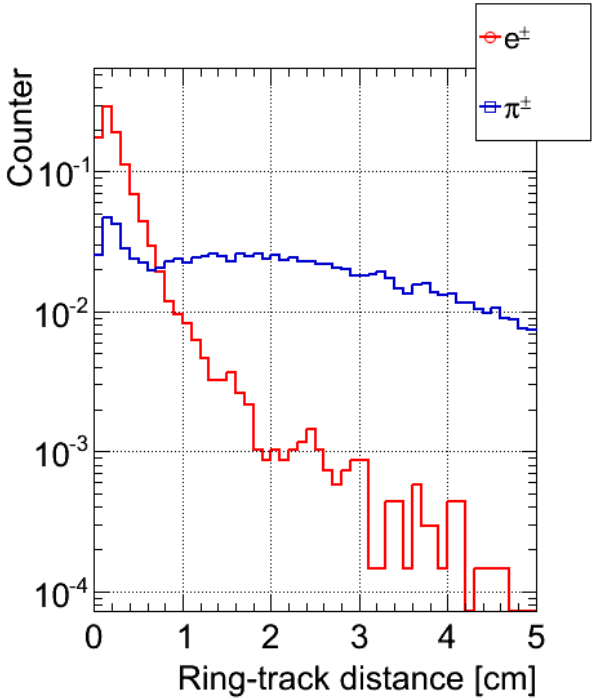


- The ring resolution is one of the most important parameter for a good electron identification in the RICH.



Ring-track matching

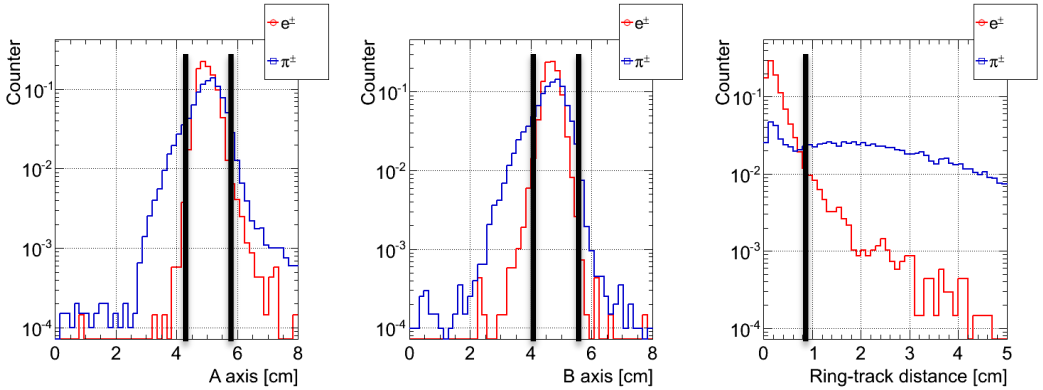




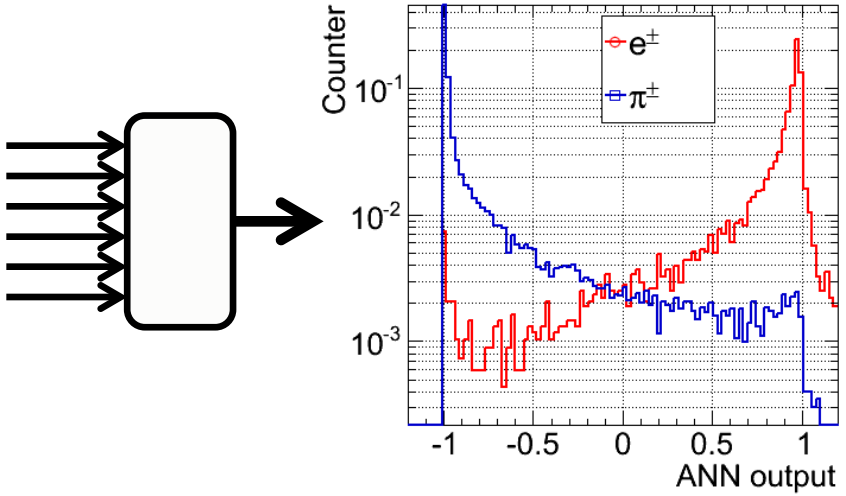
- In theory the track projection should be located in the ring center but in reality it is not always true.
- Matching is based on the shortest possible distance between ring centers and extrapolated track positions.

Electron identification

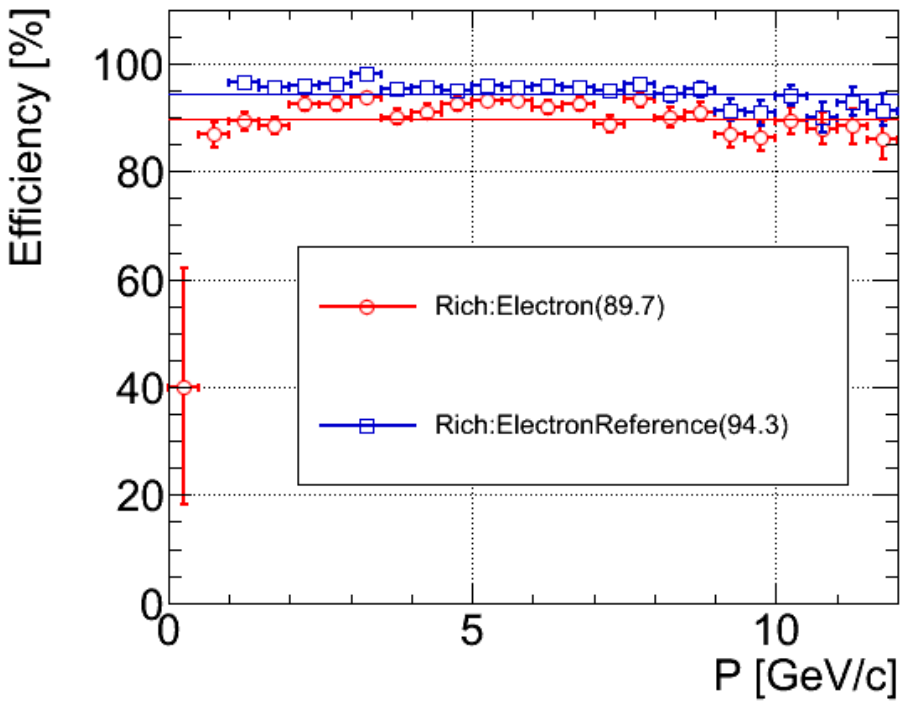
- Input parameters:
  - A axis, B axis and rotation angle of ellipse
  - Radial angle and radial position
  - Chi2 of ellipse fit
  - Number of hits
  - Distance to closest track
  - Momentum of the track



- **Artificial Neural Network (ANN) derives output value from nine input parameters.**
- **The ANN output provides a parameter: whether ring was identified as electron or as pion.**
- **The cut on ANN output value depends on the required electron identification efficiency (90% by default).**



Ring reconstruction results

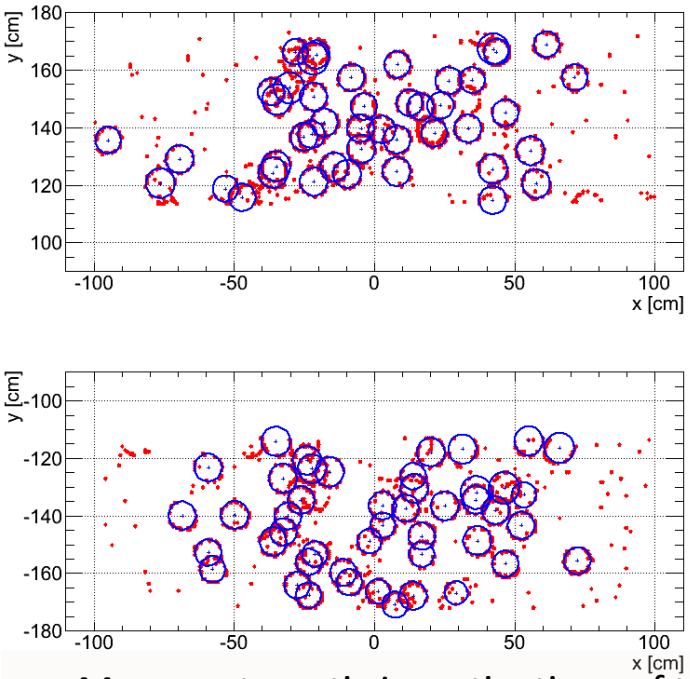


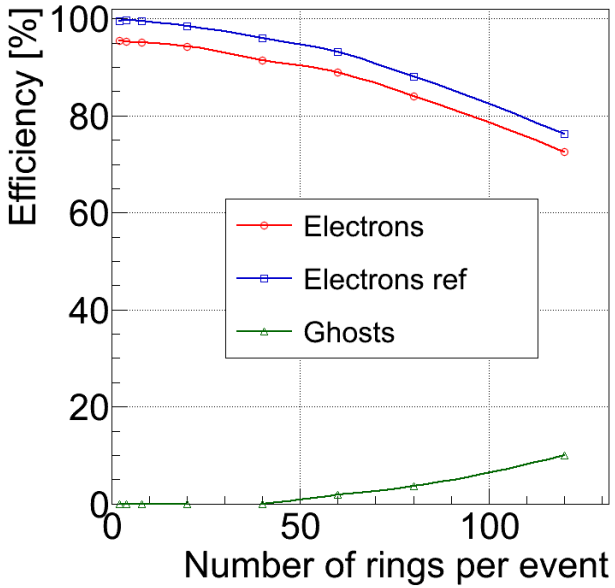
Simulation: UrQMD @ 25 AGeV + 10e+/-, 1kevents

Number of ghost rings is 7/event, and 1.8/event after STS matching

- **Efficiency = Number of correctly found ring/number of accepted rings**
- **Electron - electron rings with at least 7 hits**
- **ElectronReference - electron rings with at least 15 hits and momentum more than 1GeV/c**

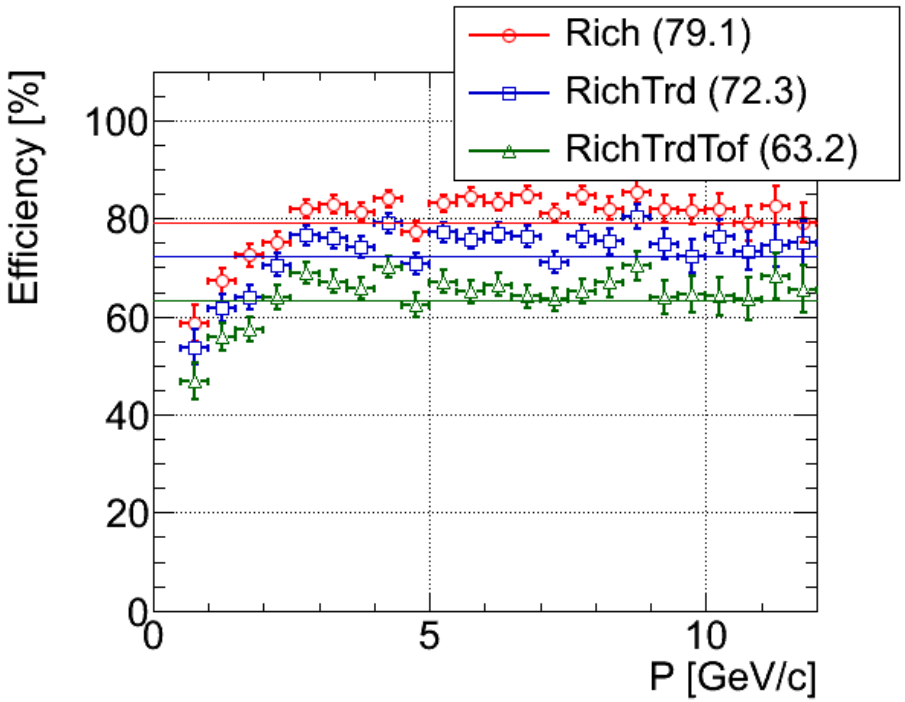
Systematic studies of the reconstruction algorithm

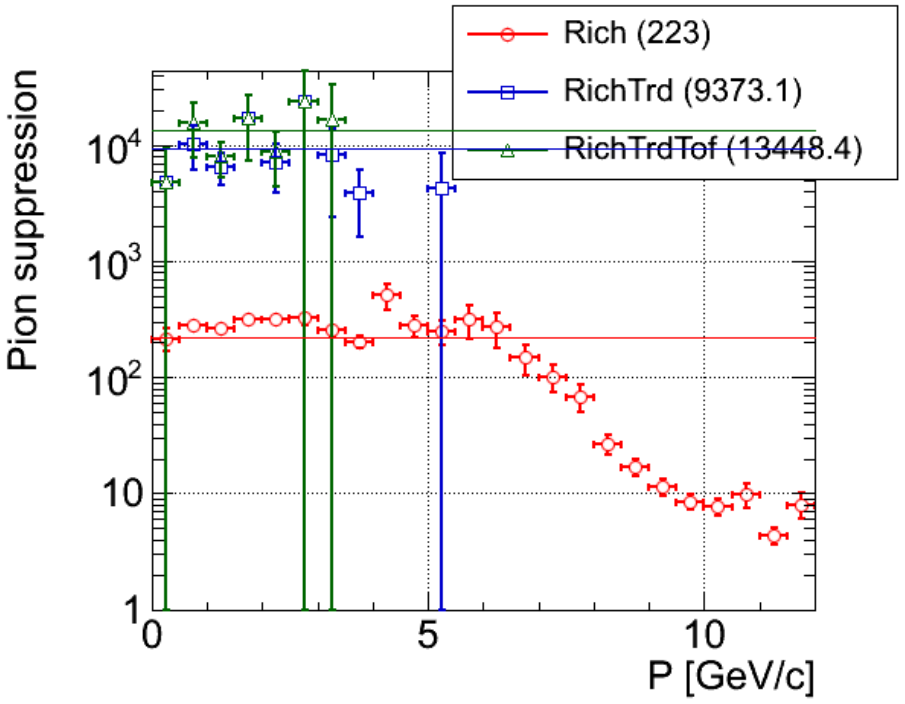




- Many systematic investigations of the ring reconstruction algorithm were performed in order to define limitations and robustness of the algorithm to different experimental factors and obtained more precise characteristics of the RICH detector.
  - reconstruction efficiency in high ring density environment
  - less number of hits in rings
  - additional hit error
  - number of noise hits

Electron identification results

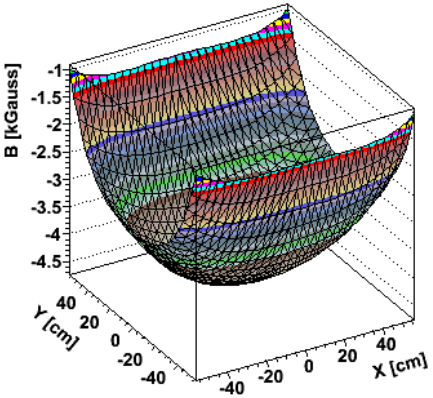


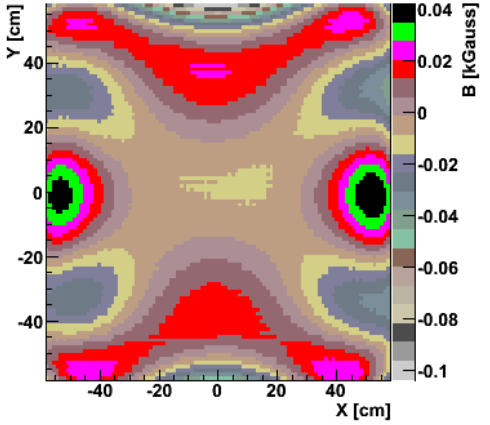


- The RICH detector alone yields a pion suppression factor of 400 at an electron identification efficiency of 75% for momentum range from 0 to 6 GeV/c. In combination with TRD a factor 13k is reached at 63% efficiency.
- Pion suppression is number pions which were reconstructed in STS and have track projection in the RICH divided by the number of pions identified as electron

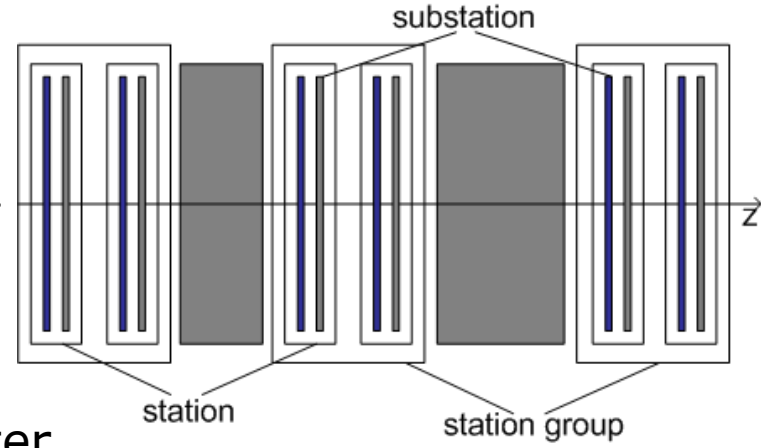
Optimization of the algorithm

- Minimize access to global memory
  - Approximation of the 70 MB large magnetic field map
    - 5 degree polynomial in the detector planes
    - parabola between the stations





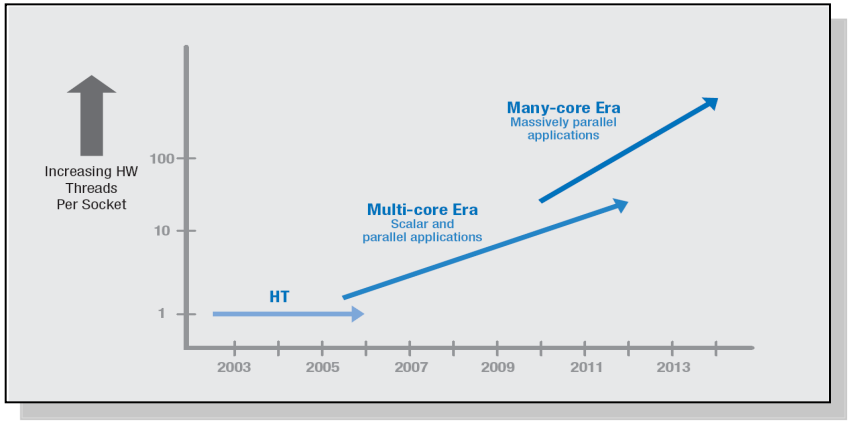
- Simplification of the detector geometry
  - Problem
    - Monte-Carlo geometry consists of 800000 nodes
    - Geometry navigation based on ROOT TGeo
  - Solution
    - Create simplified geometry by converting Monte-Carlo geometry
    - Implement fast geometry navigation for the simplified geometry



- Computational optimization of the Kalman Filter
  - From **double** to **float**
  - Implicit calculation on non-trivial matrix elements
  - Loop unrolling
  - Branches (`if then else` ..) have been eliminated

All these steps are necessary to implement SIMD tracking

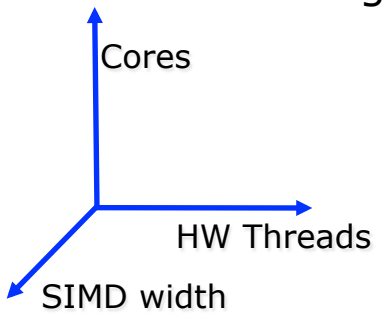
Parallelism

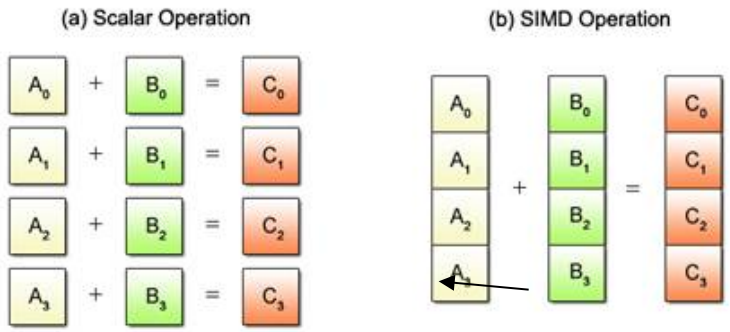


S. Borkar et al. (Intel), "Platform 2015: Intel Platform Evolution for the Next Decade", 2005.

- We´re already at the age of parallel computing
- Parallel computing relies on parallel hardware
- Parallel computing needs parallel software
- So parallel programming is very important
  - new way of thinking
  - identification of parallelism
  - design of parallel algorithm
  - implementation can be a challenge

- SIMD – Single Instruction Multiple Data
  - CPU's have it!
  - **Today**: SSE - 128 bit registers
    - 4 x float
  - **Future**: AVX, LRB
    - AVX: 8 x float
    - LRB: 16 x float
  - Benefits:
    - X time more operations per cycle
    - X time more memory throughoutput
- Multithreading
  - Many core era coming soon…
  - Tool for CPU: Threading Building Blocks



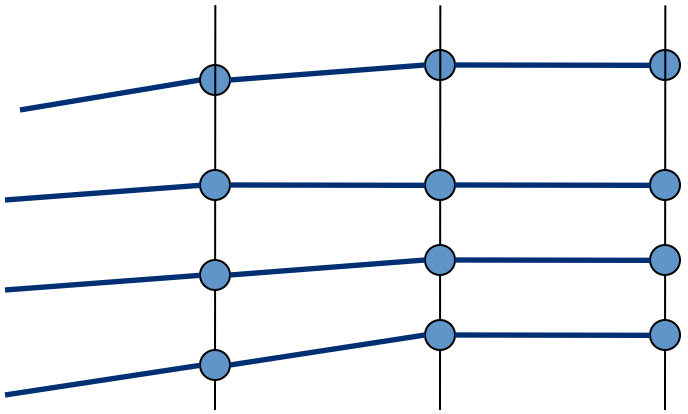


4 concurrent add operations

SIMDization of track fitter

- Vectorization of the tracking data
  - X_{vector} contains (X_0,X_1,X_2,X_3)
- Vectorization of the track fitter algorithm
  - SSE instructions were overwritten in a header file
- All tracks are independent and fitted by the same algorithm: 4 tracks packed into one vector and fitted in parallel

Finally SIMD version of the track fitter can fit 4 tracks at a time!

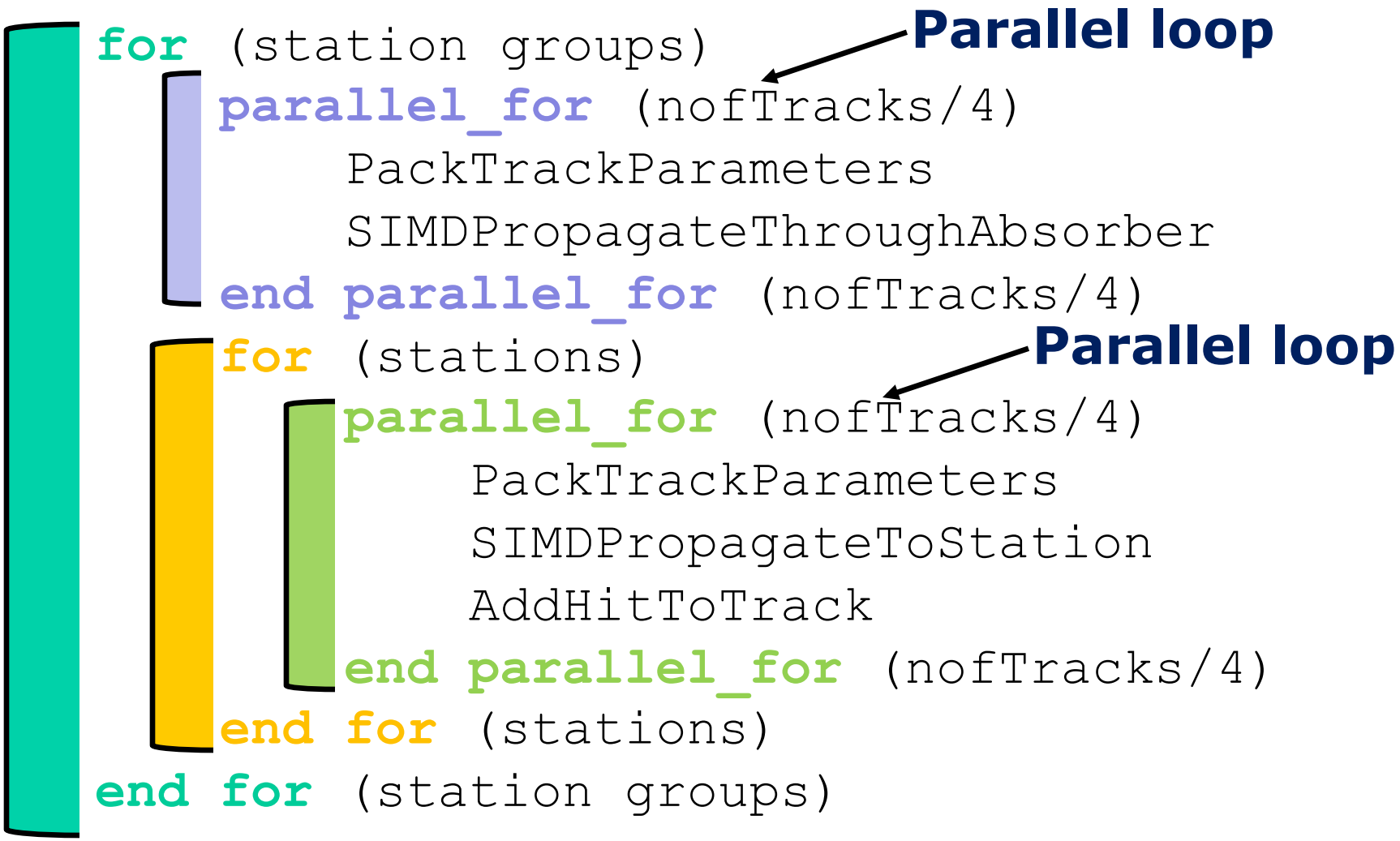


Serial tracking implementation

```
for (nofTrackSeeds)
    Follow each track through the detector
    Pick up nearest hits
end for (nofTrackSeeds)
```

Can not be directly used in SIMD approach, so it has to be significantly restructured!

Parallel tracking



Performance of the track fit

Track fit quality

Residuals					Pulls				
X [cm]	Y [cm]	Tx $*10^{-3}$	Ty $*10^{-3}$	q/p $*10^{-3}$ [GeV^{-1}]	X	Y	Tx	Ty	q/p
0.38	**0.39**	**9.1**	**8.7**	**3.4**	**1.02**	**0.99**	**1.08**	**1.08**	**0.92**

Speedup of the track fitter

	Time [μs/track]	Speedup
Initial	**1200**	**-**
Optimization	**13**	**92**
SIMDization	**4.4**	**3**
Multithreading	**0.5**	**8.8**
Final	**0.5**	**2400**

Throughput: $2*10^6$ tracks/s

Computer with 2xCPUs Intel Core i7 (8 cores in total) at 2.67 GHz

Performance of the parallel tracking

Simulation:

• 1000 UrQMD events at 25 AGeV Au-Au collisions + 5 μ+ and 5 μ- embedded in each event

	Initial version	Parallel version
Efficiency [%]	**94.7**	**94.0**

Speedup of the track finder

	Time [ms/event]	Speedup
Initial	**730**	**-**
Optimization	**7.2**	**101**
SIMDization	**4.8**	**1.5**
Multithreading	**1.5**	**3.3**
Final	**1.5**	**487**

Computer with 2xCPUs Intel Core i7 (8 cores in total) at 2.67 GHz

Thank you
for your attention ☺