GPGPU projects in Wuppertal

-

Workshop on GPUs, Hamburg
16.04.2013

Thijs Cornelissen, Sebastian Fleischmann, Peter Maettig, Manuel Neumann

Bergische Universität Wuppertal

Projects for GPGPU processing

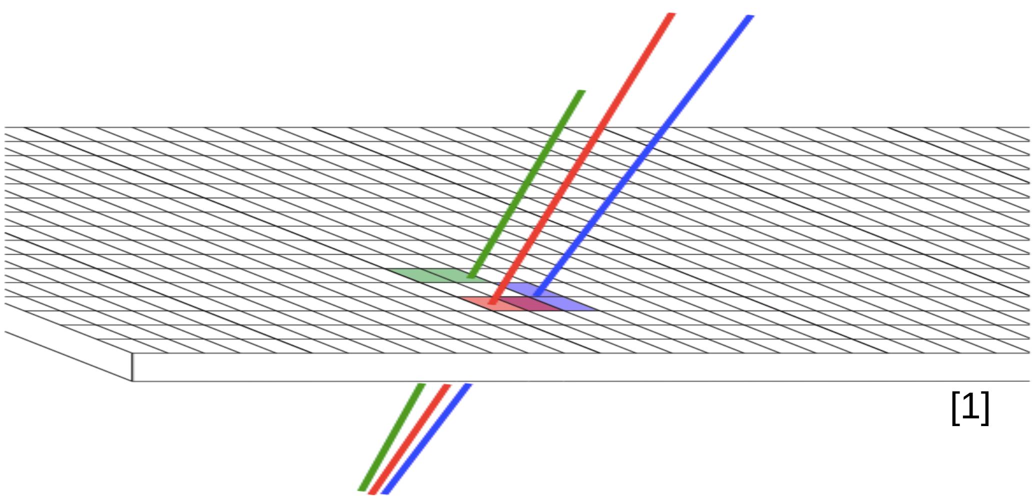
- Overall goal: Gain experience in parallelisation of algorithms in ATLAS offline reconstruction
- **1st Project:** Cluster splitting with neural networks
  - NN implementation in Toolkit for Multivariate data Analysis (TMVA, [0]) on GPUs
  - Collaboration with FH Niederrhein (P. Ueberholz)
  - Preparation of measured data as input for track reconstruction
- **2nd Project:** Track fitting
  - Collaboration with FH Muenster (N. Wulff)
  - Fit track seeds to measurements
  - Multi Track Fitter
    - Kalman filter
    - Global χ^2 fitter

1st Project: Cluster Splitting using neural networks

- Close by particles leave charge clusters in pixel detector that may overlap
- Previous cluster seeking method did not solve the issue of accidentally merged clusters

- Merged clusters will reduce tracking performance
  - Too many shared measurements will lead to rejection of tracks
  - Resolution decreases due to less precise reconstructed objects

- New approach uses charge distribution to split clusters according to estimated track count
- A neural networks calculates probability that a cluster is caused by multiple particles



[1]

New cluster splitting

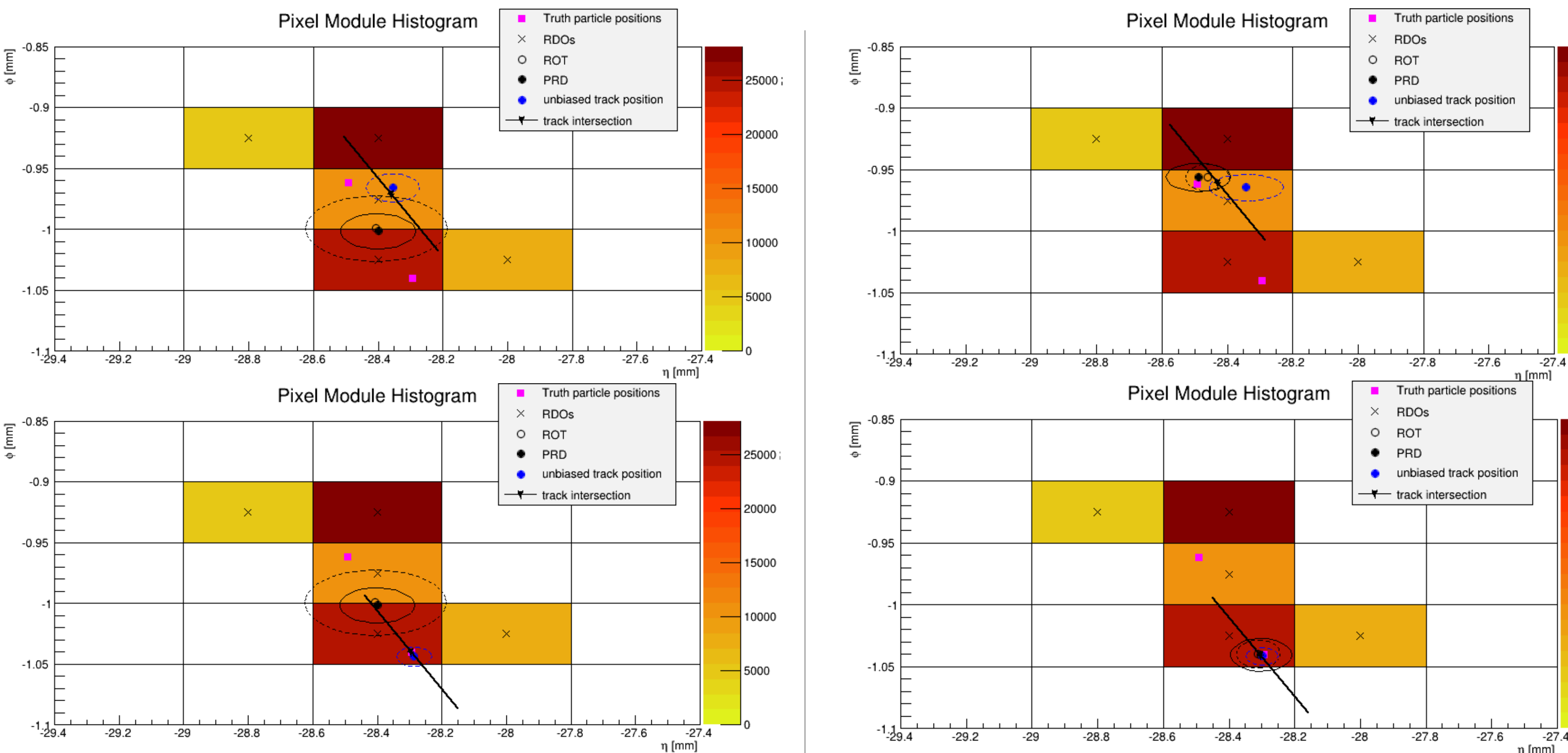
- Uses charge information of single pixels in cluster
    - Estimate number of passing tracks of cluster
    - “ position of each track
    - “ uncertainty of measurement
- Inputs:
    - 7x7 pixel information centred using charge weights
    - Longitudinal length of pixel cell
    - Estimated track direction (angle to surface normal)
- Multi Layer Perceptron (MLP) with
    - 3 output nodes (1 for each number of tracks)
    - about 60 input variables
- Try to use GPUs for network training
    - Parallelise weight calculation
    - Train different networks in parallel
    - Starting from GPU-based (CUDA) MLP implementation in TMVA from Edinburgh [SOURCE]
        - Extension to multiple output nodes
        - Rewrite in OpenCL

Manuel Neumann
Workshop on GPUs
15. - 16. April 2013

BERGISCHE UNIVERSITÄT WUPPERTAL

Cluster splitting performance (I)

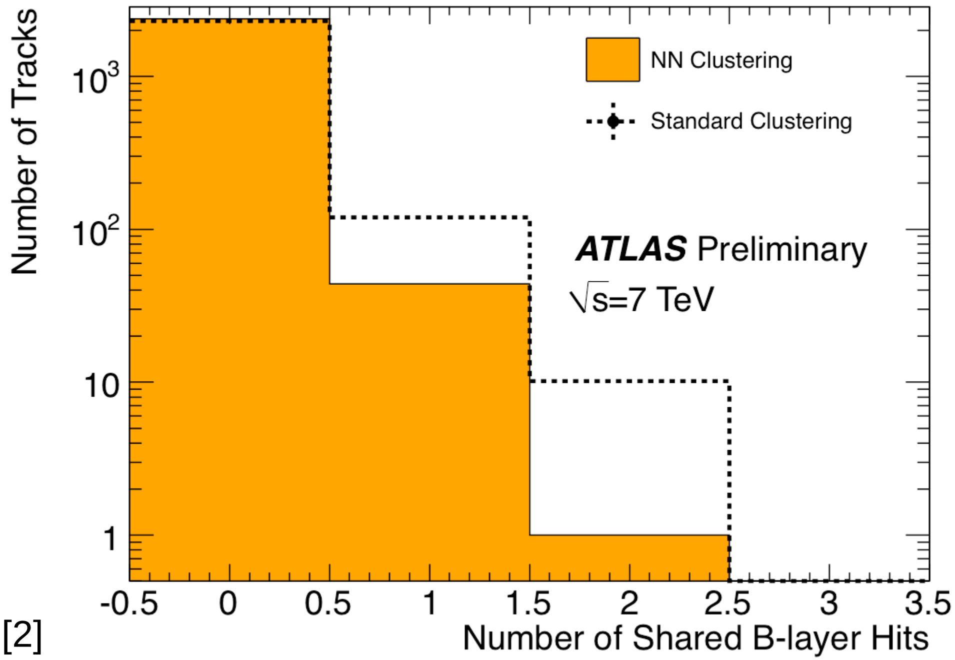
- Compare old (left) to new (right) clustering:
  - Reconstructed objects (black dots at Φ=-1) give intermediate and therefore worse input for track reconstruction
  - New approach separates the hits and with this increases resolution for track reconstruction



[2]

Cluster Cluster splitting performance (II)

- Shared measurements in the innermost pixel detector layer
  - Dashed line shows old clustering
  - Orange filled area NN approach
- Clearly the tracks share less hits
  - Results in better track resolution
  - Less tracks will be rejected



[2]

Preliminary results from GPU implementation

- Implementation of the NN that estimates the number of particles (3 outputs nodes)
- Serial CPU only vs. CUDA
- CUDA implementation lacks static input nodes (bias)
- Physical results **not** checked, yet

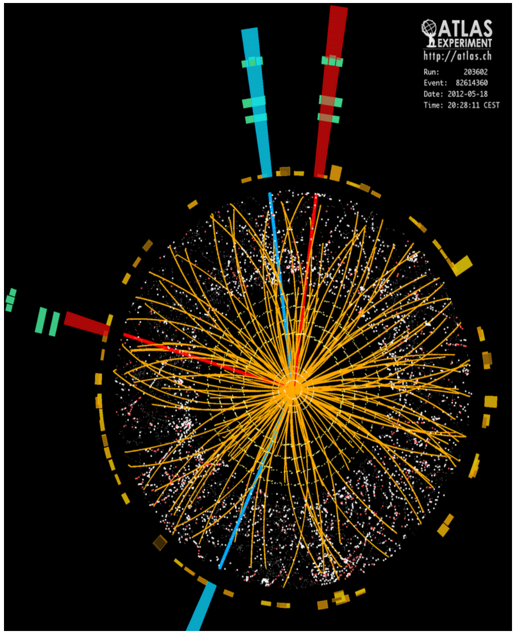
	GeForce GTX 570	Intel Core i7-2600s
Clock rate [GHz]	1.464	2.8 (3.8)
Memory Bandwidth [GB/sec]	152	21
Cores [int]	480	4 (HT)

- Large net: 95604 events
  - 61 input variables
  - 64 neurons and 4160 synapses
  - 3 output nodes
- Small net: 6k events
  - 4 input variables
  - 19 neurons and 45 synapses
  - 1 output node

Training time for different nets

	Large net	Small net
MLP	**14.97 h**	14.3 s
MLP (CUDA)	**24.3 m**	87.2 s

2^{nd} **Project:** Track fitting issues



- Number of tracks O(1000) per event
  - Current framework fits tracks one by one
- Combinatorics in measurement assignment
  - Finding **track seeds**: see talk of J. Mattmann (here I deal only with **fitting of tracks**)
  - Ambiguity solving during fit (shared measurements, duplicate/ghost seeds)
- Track fitting takes around 18 % CPU time of the reconstruction process
  - Needs precise geometry and magnetic field map
  - High fraction for solving ambiguities
  - Standard fitter (`GlobalChi2Fitter`) deals with matrices of O(50x50)

[3]

Update track fitting algorithms

- Goals
  - reduce computing time
  - More efficient usage of current and future hardware
  - Improve physics performance in dense track environments (boosted jets)
- Ideas
  - Fit many tracks at once
  - Algorithm that solve measurement ambiguities intrinsically (see next slides)
  - Use vector architectures
- Boundary Conditions
  - Track fitting needs precise detector geometry and magnetic field information
  - No port of whole reconstruction framework (lack of manpower)

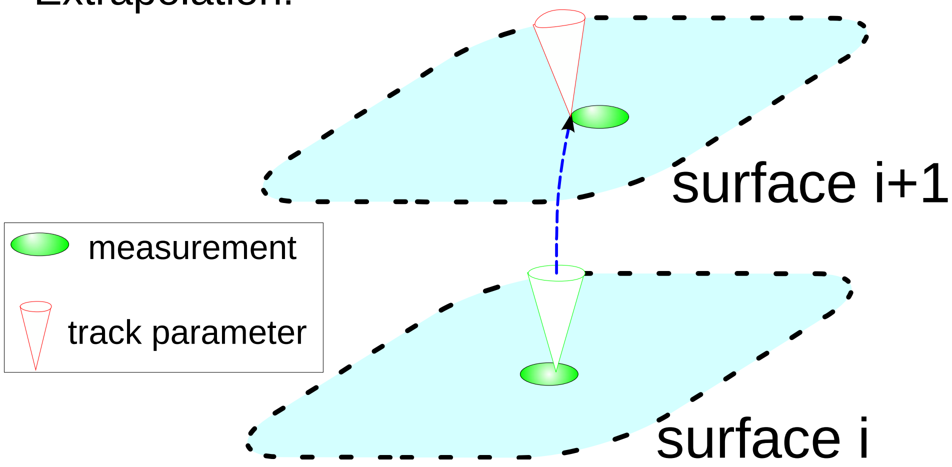
Manuel Neumann
Workshop on GPUs
15. - 16. April 2013

BERGISCHE UNIVERSITÄT WUPPERTAL

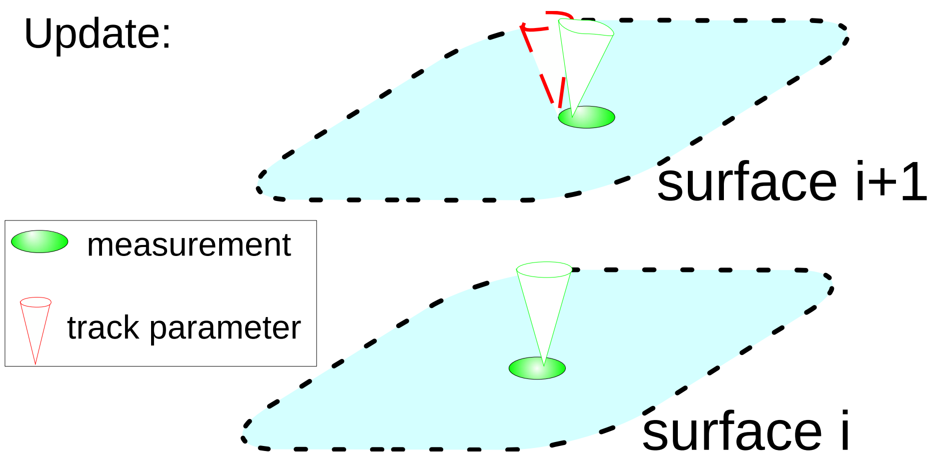
Basic tool: Kalman Filter

- Procedure
  - Extrapolate track parameters to next surface
  - Combine track parameters with corresponding measurement
  - Extrapolate updated track parameters to next layers
- Can be done inside out (fine to coarse detector) and vice versa
- Iterative approach does not need full geometry information in every step

Extrapolation:



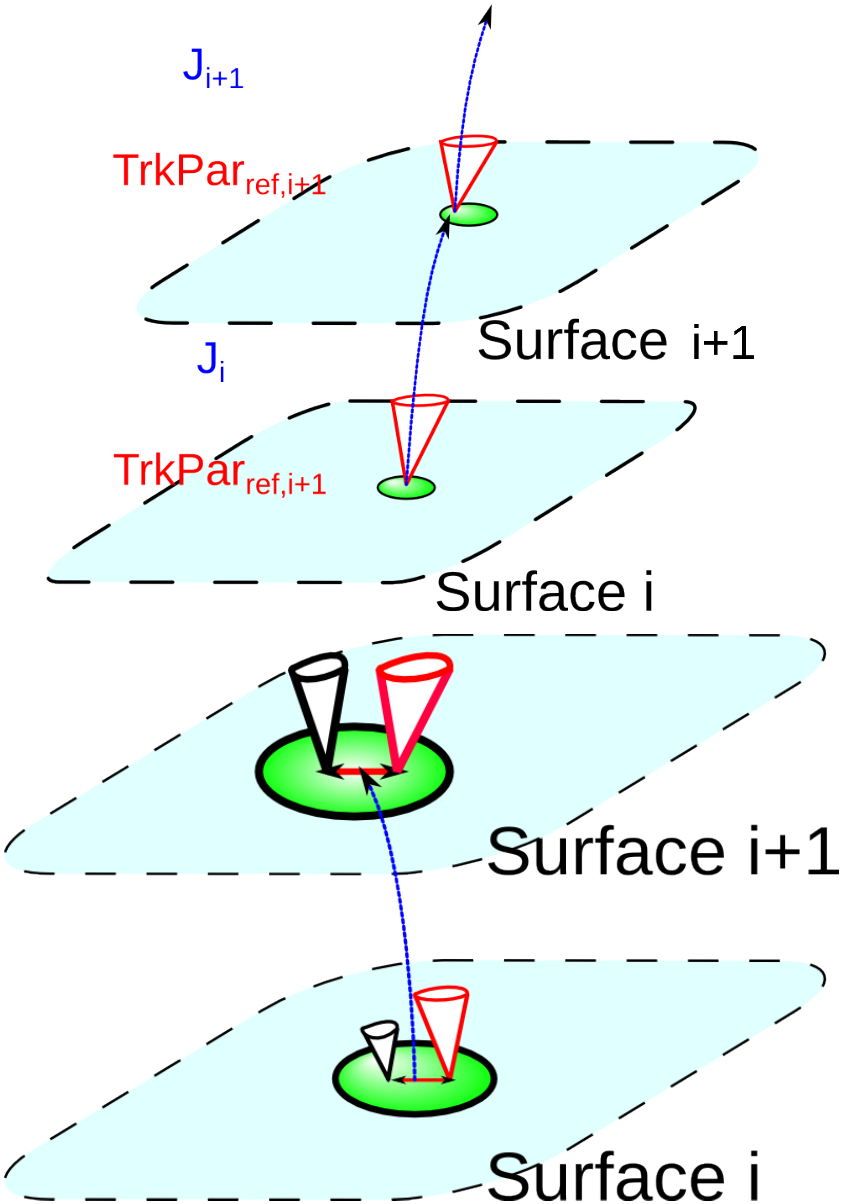
Update:



Reference fit method

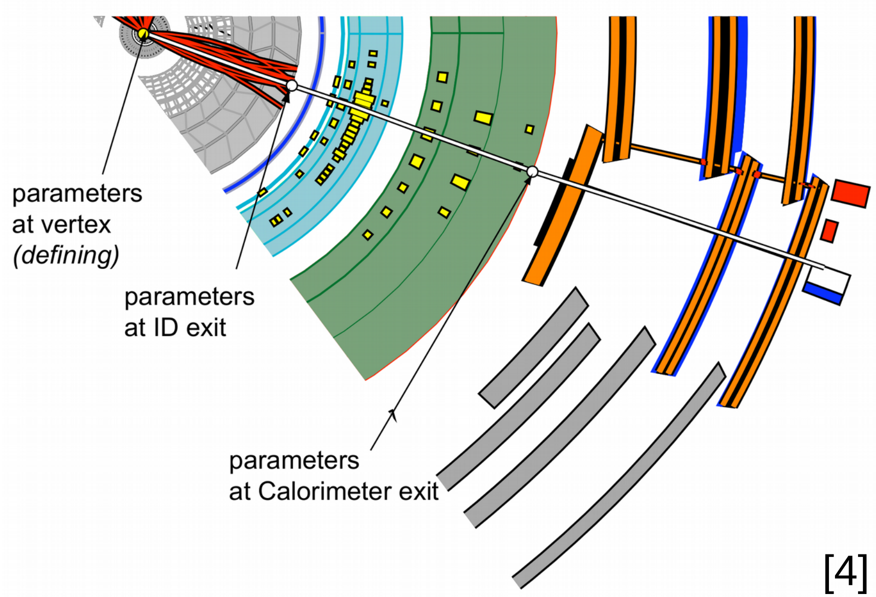
- Extension of the Kalman filter
- Linearisation around seed

- Initialisation:
  - Do a full Kalman fit without including the measurement parameters
  - Store transport Jacobian J matrices and track parameters $TrkPar_{ref}$ of this fit
- Fit:
  - Propagate difference of track parameters to measurement with the transport Jacobians
    $\Delta x_{i+1}^{pred} = J_i \Delta x_i$
  - Correct prediction with measurement parameters



Reasons for the reference fit method

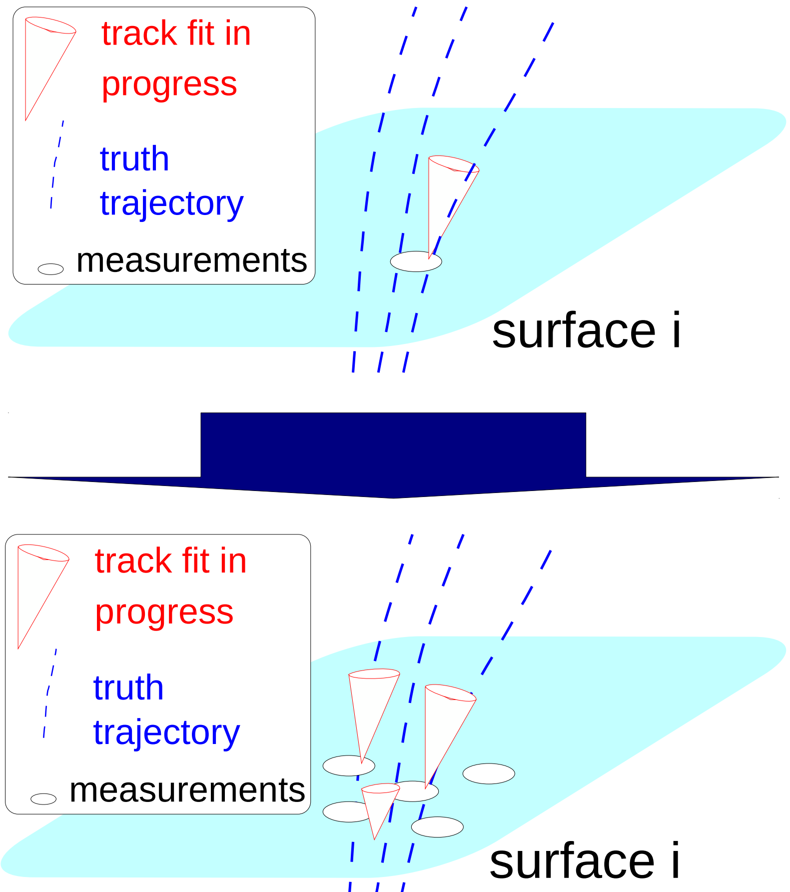
- Higher stability related to numerics and material effects
  - Special w.r.t. detector geometry and features
    - precise (ID) and coarse subsystems (muon spectrometer)
    - magnetic field
- no detector geometry after the first fit iteration needed
  - No database lookup in every iteration
- Matrix operations should be done fast on GPUs and SIMD units
- This allows data parallel (multiple tracks at once) processing

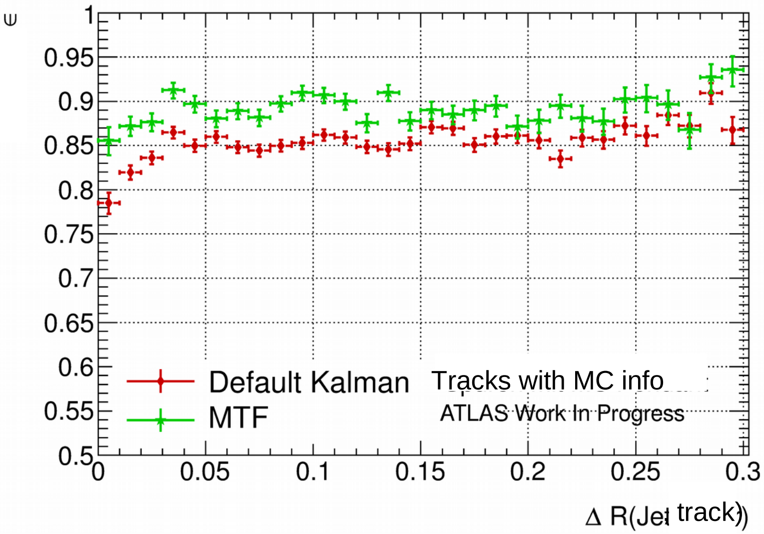


[4]

Multi Track Fitter

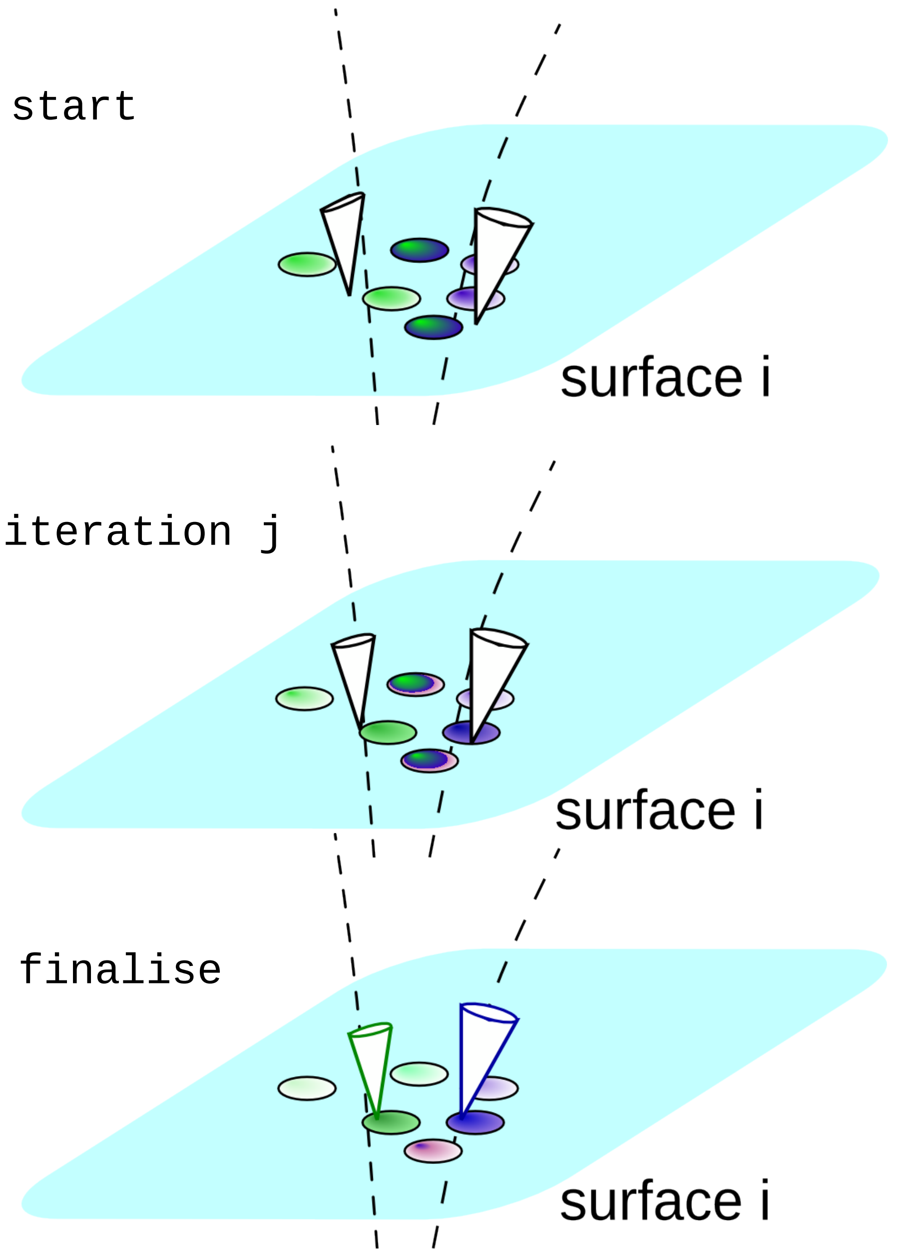
- Based on “adaptive multi track fitting” by A. Strandlie, R. Frühwirth [5]
- Idea:
  - Combine information of multiple track candidates
  - Loose assignment of measurements at the beginning of fit process
  - Inherent ambiguity processing of measurement to track assignment
  - Improve tracking performance inside jets





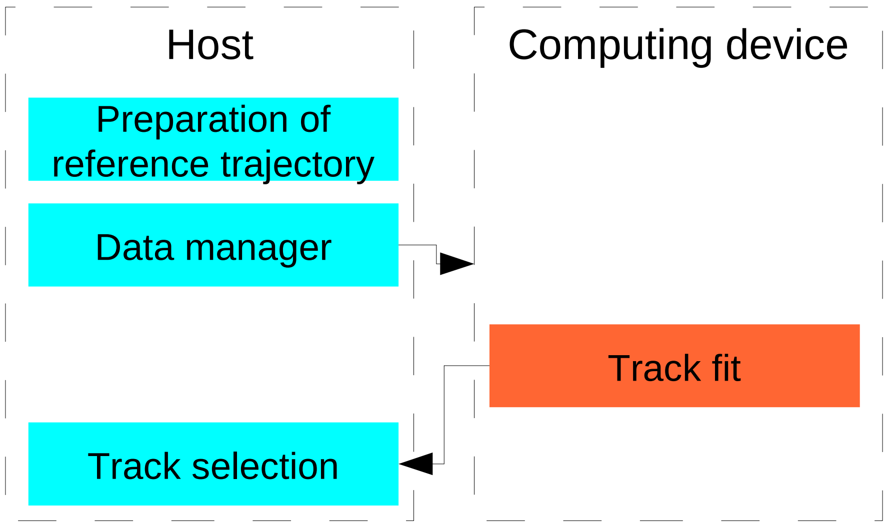
Multi Track Fitter - Principle

- Extension of the Kalman Filter:
  - Assign a weight to every measurement for every considered track
  - In every iteration of the MTF:
    - Decrease a temperature parameter that goes into the calculation of the weight (Low values for temperature cause a hard cut off)
      - Proceed with the Kalman filter
  - Selection: In the last iteration a single measurement will be assigned to a track



Multi Track Fitter – the whole package

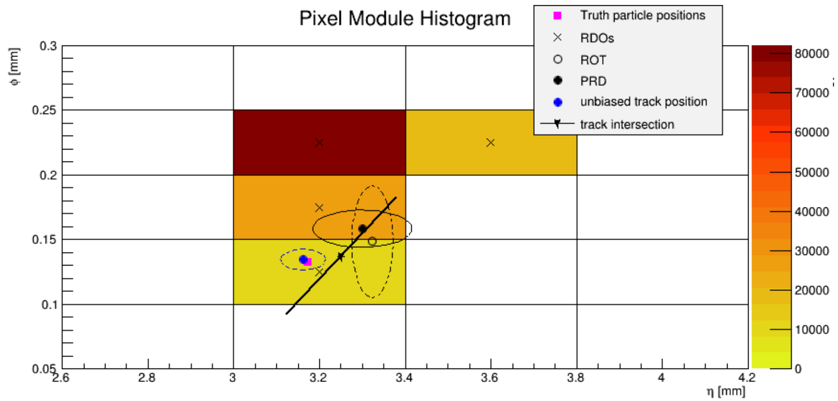
- Use Kalman filter with reference method
- Fill reference trajectory and assignment weights at the beginning
- Transfer trajectories and track seeds to computing device
- Fit predictions to measurements
  - only updates present information
  - Mostly matrix/vector operations (suitable for SIMD/GPU computing units
- Transfer back to host and do track selection

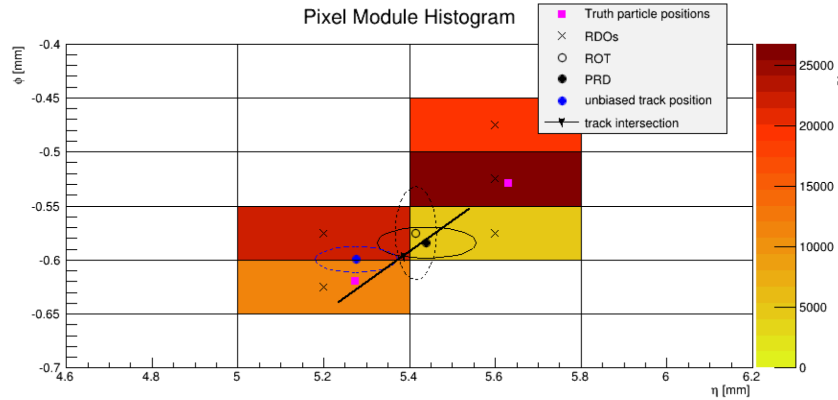


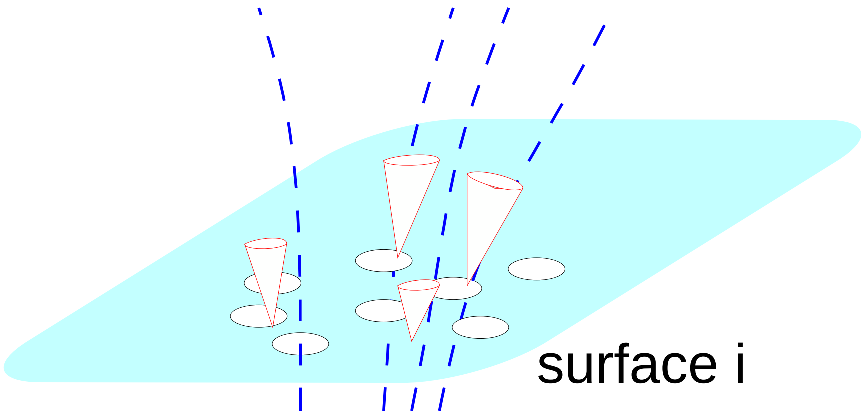
BERGISCHE UNIVERSITÄT WUPPERTAL

Summary

- Projects related to track reconstruction ongoing:
  - Neural networks to improve measurement information
  - Track fitter to gain from current (SIMD) and future (GPU) hardware
    - Will speed up fitting
  - Collaboration with FH Niederrhein/Muenster and Wuppertal's department of electrical engineering (CUDA Research Center)
  - Funding for GPU cluster (48 NVIDIA Tesla M2090 Modules) granted, but main user will be dept. of EE







Bibliography

[0]	Acceleration of multivariate analysis techniques in TMVA using GPUs A. Hoecker, H. McKendrick, J. Therhaag, A. Washbrook
[1]	Neural network based cluster creation in the ATLAS silicon Pixel Detector, Andreazza, A., ATL-PHYS-SLIDE-2013-155, 2013
[2]	Neural network based cluster creation in the ATLAS Pixel Detector, Andreazza, A., ATL-PHYS-PROC-2012-240, 2012
[3]	Track Reconstruction in the ATLAS Experiment – the Deterministic Annealing Filter, Fleischmann, S., 2007
[4]	Artemis School on Calibration and performance of ATLAS detectors / ID reconstruction, Salzburger, A., 2008
[5]	Adaptive multitrack fitting, A. Strandlie, R. Frühwirth, Computer Physics Communications, Volume 133, Issue 1, p. 34-42.

Backu
p

Kalman filter

- Prediction:

$$x_{i+1}^{pred} = F_i x_i$$

$$C_{i+1}^{pred} = F_i C_i F_i^T + Q_i$$

C: covariance,
x: track parameters,
F: model description
Q: material effects

- Update:

$$x_{i+1}^{upd} = x_i C_{i+1}^{upd} \left(\left(C_{i+1}^{pred} \right)^{-1} x_{i+1}^{pred} + H_{i+1}^T V_{i+1}^{-1} m_{i+1} \right)$$

$$\left(C_{i+1}^{upd} \right)^{-1} = \left(C_{i+1}^{pred} \right)^{-1} + H_{i+1}^T V_{i+1}^{-1} H_{i+1}$$

$V = C^{-1}$,
m: measurement parameters,
H: translation between trk/measurment space

Reference method

- Prediction

$$\Delta x_{i+1}^{pred} = J_i \Delta x_i$$

$$\Delta x = x_{trkpar} - x_{refpar}$$

$$C_{i+1}^{pred} = J_i C_i^{ref} J_i^T + Q_i$$

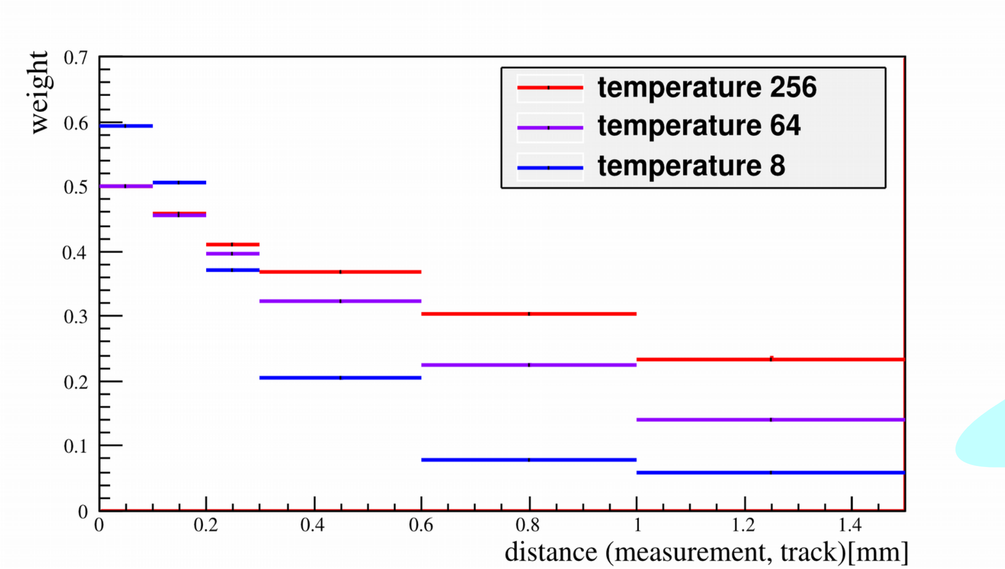
Optimisation

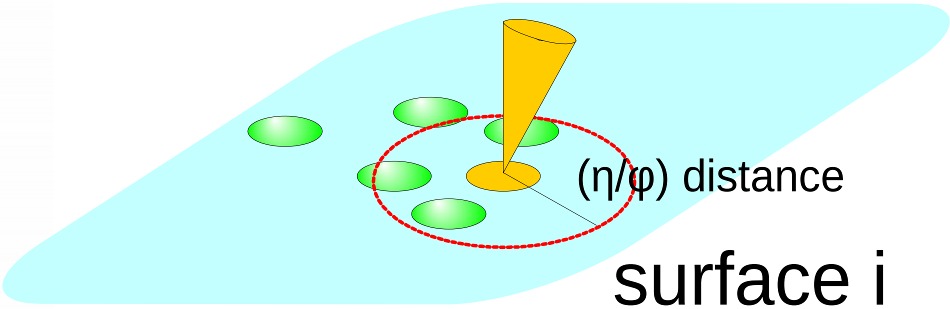
- Goal: Reduce fake rate while hold reconstruction efficiency high
- Problematic parameters of the MTF:
  - Temperature scheme
  - Road width of additional measurements (given by local distance between track and measurements)
  - max(η/φ) is the largest value for (η/φ)-distance

$$[T_0, T_1, \ldots, T_N]$$

$$distance_\eta := \frac{|Trk::loc\,\eta - cluster::loc\,\eta|}{\sigma_\eta}$$

$$distance_\varphi := \frac{|Trk::loc\,\varphi - cluster::loc\,\varphi|}{\sigma_\varphi}$$





BERGISCHE UNIVERSITÄT WUPPERTAL

Proposed competing ROT mean values in the SCT

- New compROT mean:

$$m'_i = \left(p_a \frac{m_{a,x}}{\lambda_{a,1}} + p_b \frac{m_{b,x}}{\lambda_{b,1}}\right)\left(\frac{p_a}{\lambda_{a,1}} + \frac{p_b}{\lambda_{b,1}}\right)^{-1} \qquad i \in x, y$$

$$V' = R V'' R^T$$

$$V'' = \begin{pmatrix} \frac{1}{p_a/\lambda_{a,1} + p_b/\lambda_{b,1}} & 0 \\ 0 & \frac{1}{p_a/\lambda_{a,2} + p_b/\lambda_{b,2}} \end{pmatrix}$$

$$R = \begin{pmatrix} \cos(\theta') & -\sin(\theta') \\ \sin(\theta') & \cos(\theta) \end{pmatrix}$$

  - θ is the angle between the system of the covariance and the SCT module system

$$\theta = \frac{1}{2}\arctan\left(\frac{2\sigma_{xy}}{\sigma_x^2 - \sigma_y^2}\right)$$

- λ_j are the eigenvalues of the covariance matrices:

$$\lambda_{a,j} = \frac{Tr(V)}{2} \pm \sqrt{\frac{Tr(V)^2}{4} - \left(\sigma_x^2 \sigma_y^2 - \sigma_{xy}^2\right)}$$