Connecting The Dots 2016

22-24 February 2016
HEPHY Vienna
Europe/Vienna timezone

This is a workshop on track reconstruction and other problems in pattern recognition in sparsely sampled data. The workshop is intended to be inclusive across other disciplines wherever similar problems arise. The main focus will be on pattern recognition and machine learning problems that arise e.g. in the reconstruction of particle tracks or jets in high energy physics experiments. Both hardware and software aspects will be addressed.

Scientific Programme

- **Algorithms and theoretical analysis**
  - Mathematical evaluation of pattern recognition problems, fitting, effect of noise, treatment of multiple scattering, theoretical limits, etc.

- **Parallel and/or discrete pattern recognition**
  - Includes Hough transform approaches, look-up tables, associative memory.
- **Neural networks, machine learning, and neuromorphic approaches**
  - Includes both software/firmware implementations and exploration of neuromorphic hardware
- **Applications and performance evaluation**
  - Examples of implemented pattern recognition problems and solutions with emphasis on new challenges and limits of scaling existing approaches.

Participation & content

- 74 participants
- Most of the talks' content also in previous CHEP, ACAT, Vertex, VCI
- More intimate venue and relaxed atmosphere allowed fruitful discussion

- Today will not cover
  - Machine Learning Challenge
  - Common Tracking Software Forum

Student Session

Eta correction for silicon sensors	*Manfred VALENTAN*	
Seminarraum 1,2,3, HEPHY Vienna		16:20 - 16:35
Tracking in ASACUSA	*Bernadette KOLBINGER*	
Seminarraum 1,2,3, HEPHY Vienna		16:35 - 16:50
Track reconstruction in the InGrid TPC for ILC	*Amir SHIRAZI*	
Seminarraum 1,2,3, HEPHY Vienna		16:50 - 17:05
Expected performance of the ATLAS Inner Tracker	*Simon VIEL*	
Seminarraum 1,2,3, HEPHY Vienna		17:05 - 17:20
Machine learning assisted track finding in the Belle II SVD	*Thomas MADLENER*	
Seminarraum 1,2,3, HEPHY Vienna		17:20 - 17:35
Tracking in MAPT	*Michael MILDE*	
Seminarraum 1,2,3, HEPHY Vienna		17:35 - 17:50

AUTOMATION & CONTROL INSTITUTE
INSTITUT FÜR AUTOMATISIERUNGS- & REGELUNGSTECHNIK

TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

Robot and Computer Vision

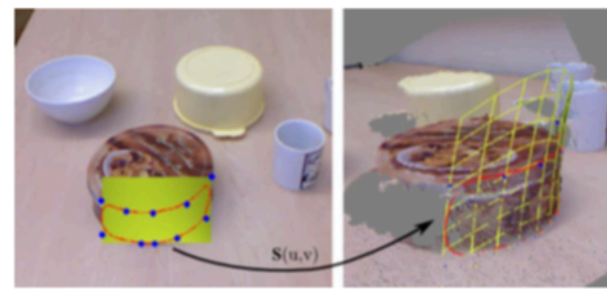
Markus Vincze

Institut für Automatisierungs- und Regelungstechnik
Technische Universität Wien
vincze@acin.tuwien.ac.at

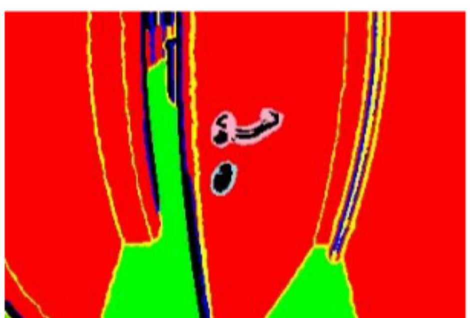
Connecting the Dots, 22. Feb. 2016, HEPHY Vienna

Vision for Robotics & Automation

- Vision: „We make robots see.“
- Form and function of objects
  - Robot navigation and grasping
  - Object function, shape and pose estimation
  - Learning novel objekts and object classes



- Industrial and service robots



HOBBIT – The Mutual Care Robot Fall Prevention and Acceptance

Demographic challenge

- Increasing age, highest risk: fall
- 50% hospital visit persons over 65
- 175M€ operations; 6% health costs

Robot for fall prevention/detection

- Clean up floor, free paths at home
- Socially connected, activity, entertainment

49 test persons in A, S, GR

- 70-88 Jahre, living alone, moderate impairments
- Very high acceptance 87%
- Rent for their home 77%

3-weeks Study in flats of older persons

[Haus der Barmherzigkeit]

Markus Vincze

Conclusion

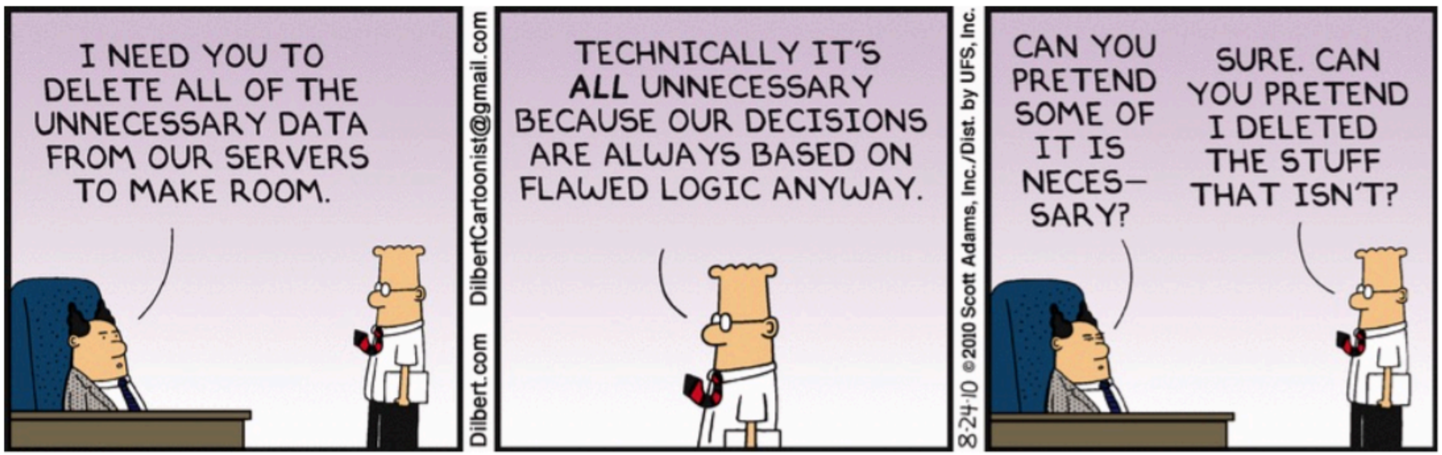
- Model-based methods for finding geometric features
- Idea: generate hypotheses and then check in data (hypothesise and verify)
- Can be applied if there are only a few percent of „good“ data
- Information over type of data may significantly accelerate search

Big-Data in Astronomy and Astrophysics

Extracting Meaning from Big-Data

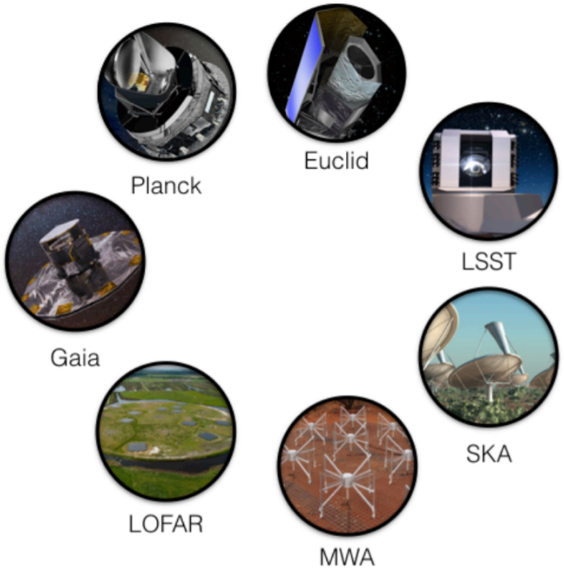
Jason McEwen
www.jasonmcewen.org
@jasonmcewen

Mullard Space Science Laboratory (MSSL)
University College London (UCL)



Connecting the Dots
Institute of High Energy Physics, Vienna, February 2016

What is big-data in astronomy and astrophysics?



Wide and deep data and observations

What is big-data in astronomy and astrophysics?

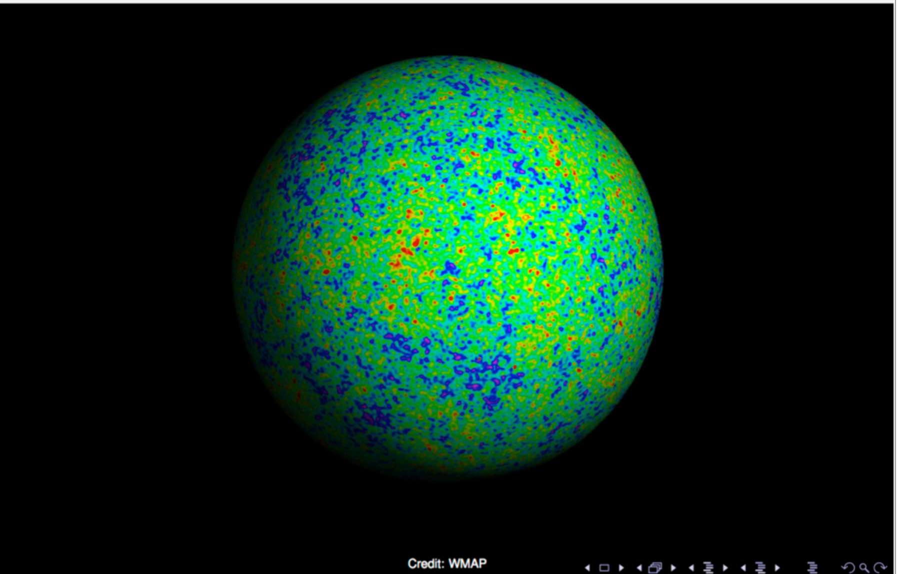
- Big machines
  - experiments, physical hardware, computing
- Big theory and simulations for forward modelling
  - cosmological evolution of linear perturbations, N-body simulations, non-linear scales (astrophysics + cosmology), radiative transfer, semi-numerical methods
- Big parameter space
- Big algorithms
- Big collaborations
- Big engagement
  - *e.g.* outreach, industry

Challenges of big-data

Analysis challenges (Fan et al. 2014):

1. Heterogeneity, e.g. sub-populations, different data sources, tension between data
2. Error accumulation, e.g. high-dimensional parameter spaces, bias
3. Spurious correlations, e.g. correlation vs causation, data dredging
4. Incident endogeneity, e.g. chance correlation between signal of interest and error

Observations of the cosmic microwave background (CMB)



Bianchi VII$_h$ cosmologies

Simulations

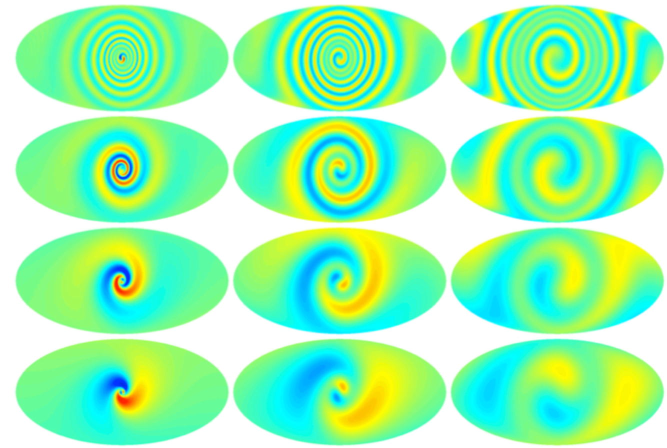


Figure: Simulated CMB contributions in Bianchi VII$_h$ cosmologies for varying parameters.

Bianchi VII$_h$ cosmologies

Best-fit Bianchi component (flat-decoupled-Bianchi model)

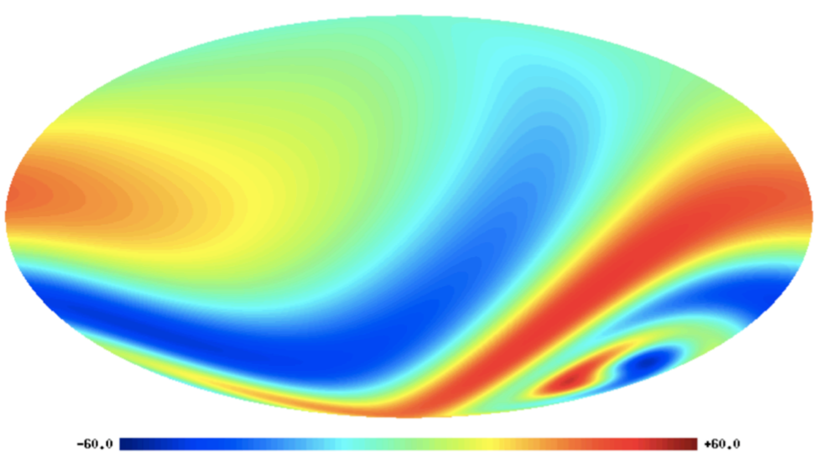


Figure: Best-fit template of flat-decoupled-Bianchi VII$_h$ model.

Bianchi VII$_h$ cosmologies

Planck results

BUT parameter estimates are not consistent with concordance cosmology.

- Follow up with Planck 2015 polarisation data, rules our flat-Bianchi-decoupled model.
- Find no evidence for Bianchi VII$_h$ cosmologies and constrain vorticity to (Planck Collaboration XVIII 2015):

$$\boxed{(\omega/H)_0 < 7.6 \times 10^{-10}}$$

95% confidence level

Concluding remarks

- Increasingly inter-disciplinary, drawing on statistics, applied mathematics, computer science, information engineering, . . .
- Increasingly intra-disciplinary (*e.g.* Planck, Euclid, LSST, SKA, . . .)
- Many methodological synergies

Concluding remarks

How can we exploit synergies?

1. Open (unencumbered) data and open code
2. Develop best practices (*e.g.* code development, general codes, reproducible/replicable research, blinded analysis)
3. Explore HPC synergies (*e.g.* Dirac, Archer, Hartree, Google, Amazon, . . .)
4. Go beyond individual techniques to understand properties of classes of approach
5. Develop common language
6. Promote inter- and intra-disciplinary collaboration and communication, *e.g.* Alan Turing Institute (ATI), workshops (*e.g.* BASP conference), Hackathons, . . .
7. . . .

First prototype of an "Artificial Retina" Processor for Track Reconstruction

Riccardo Cenci
SCUOLA NORMALE SUPERIORE &
INFN - PISA, ITALY

On behalf of INFN-Retina Collaboration

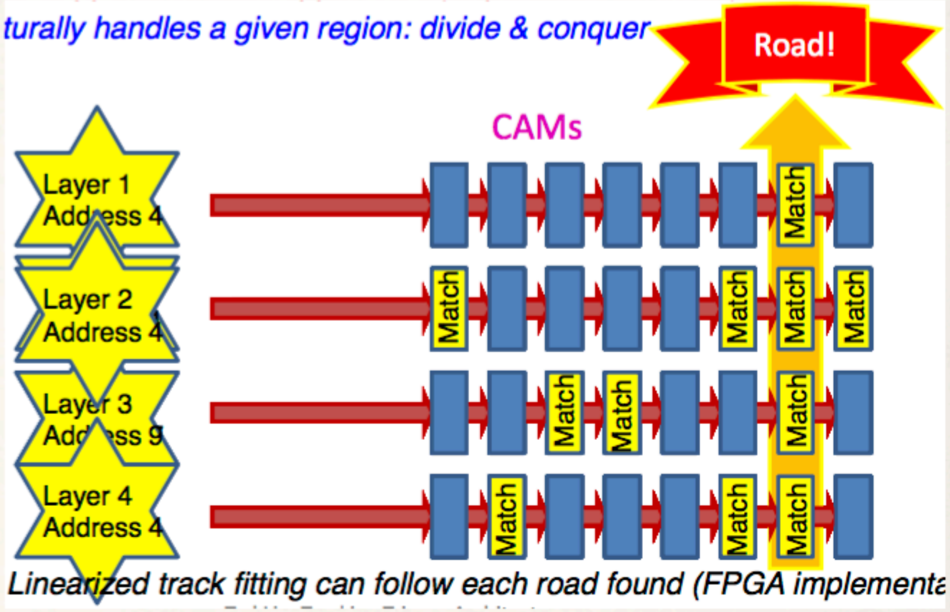
The "Retina" Project

- Three-year R&D program started in 2015 and supported by INFN-CNS5 (Technological Research Division)
- Goals:
  - Demonstrate the **feasibility, at a reasonable cost**, of a system based on the "artificial retina" algorithm **using FPGA devices**
  - Evaluate its performance in **HL-LHC** environment
- Our plan is to build two prototypes:
  - **Prototype 1**, to test the logic functionality of the full system when applied to a simple tracker
  - **Prototype 2**, to test the speed/latency for the basic components when implemented on modern high-speed devices

- INFN-Pisa: F. Bedeschi, F. Spinella, J. Waslh
- Scuola Normale Superiore and INFN-Pisa: R. Cenci, P. Marino, M. J. Morello
- Università degli Studi and INFN, Pisa: D. Ninci, A. Piucci, G. Punzi, S. Stracka
- Fermilab: L. Ristori

Pattern recognition

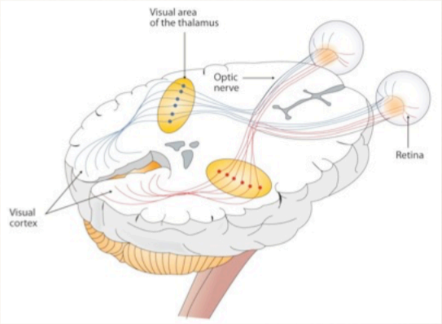
- The fastest approach to tracking implemented in a real experiment is direct matching to a bank of stored templates: **Associative Memory** (SVT@CDF)
  - No combinatorics, comparison in parallel, but patterns are still sequential in AM cell
  - Same approach will be used for Atlas L2 trigger (FTK) and CMS Phase-2
- But requirements for L0 at **HL-LHC** are not matched by a **factor ~80, is it impossible then?**



Name	Technology	Experiment	Year	Event Rate	Clock	Cycles/event	Latency
XFT	FPGA	CDF-L0	2000	2.5 MHz	200 MHz	80	<4μs
SVT	AM	CDF-L2	2000	30 kHz	40 MHz	~1600	<20μs
FTK	AM	ATLAS-L2	2015	100 kHz	~200 MHz	~2000	O(10μs)
?	**?**	**<LHC>-L0**	**~2020**	**40 MHz**	**~1GHz**	**~25**	**few μs**

Inspiration from human brain: the vision

- Early stages of vision (**edge detection**) has a lot in common with track reconstruction:
  - Lots of complex data/combinatorics
  - Little time available
  - Pressure to make accurate decisions
  - Strongly constrained computing resources



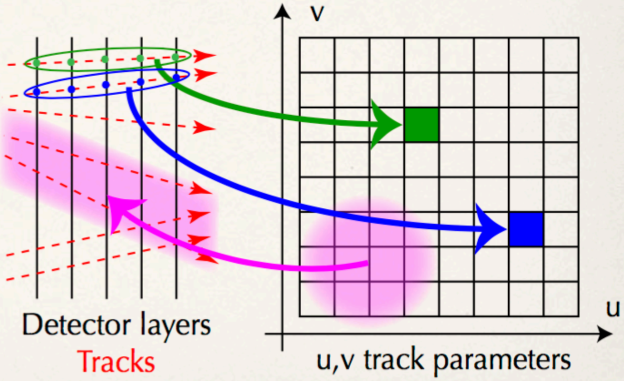
- Analog responses from retinas is delivered only to limited subset of neurons (=patterns)
- First stage of visual cortex (V1) is able to produce a sketch in less than 100 ms, working at a frequency of 30-40 Hz

ke" tracking algorithm (1)

uciano Ristori [NIM A453 (2000) 425-429] inspired to visual apparatus of mammals (from here the name **Artificial Retina**). Similarities with:

- Hough transform until 2D, but computationally simpler with more dimensions
- Associative memories for pattern matching, but analog responses using cells interpolation, implying similar or better resolution with lower number of stored patterns

- **Configuration phase** (common PC):
  1. Discretize space of track parameters (**cells**)
  2. Mapping 1: generate track intersections with detector planes (**receptors**) and connect them to cells
  3. Mapping 2: assuming contiguous cells corresponding to slightly different tracks, we connect cluster of cells to areas of detector readout



Conclusions

- Computing and storage available for **future experiments at HL-LHC** will not be able to cope with the increase of data rate, so more processing will have to be performed "online" to **reduce event rate and size**

- Current methods may not scale well. Alternative advanced solutions should be explored, like the **"Artificial Retina" algorithm**, that exploits higher degrees of parallelization and provides analog response

- The "Retina Project" aims to demonstrate the **feasibility (at reasonable cost)** of a real system based on this algorithm **able to reconstruct tracks at rates expected for LHC Run3**: we are completing a functional prototype and assembling another one for speed test

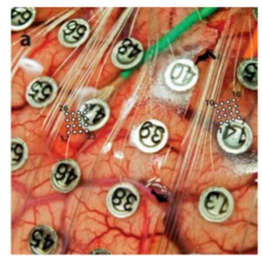
- Further developments and synergies with fast and smart tracking detector may lead to future experiments with **detector-embedded data reconstruction**

For more references see here: **link**

Exploring the boundaries of low-energy, real-time tracking with Neuromorphic Computing

K.E. Bouchard, **P. Calafiura**, R. Carney, D. Clark, D. D'Onofrio, M. Garcia-Sciveres, J.A. Livezey, C.E. Tull
LBNL

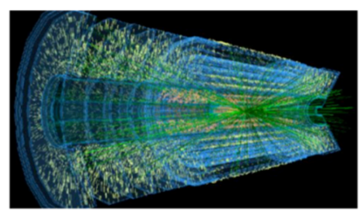
Our Goals

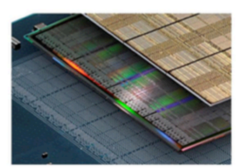


(source UCSF)

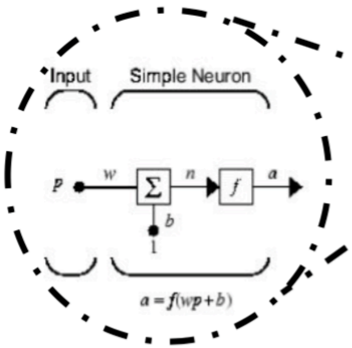
Understand role of neuromorphic computing in

- Embedded data processing
  - Portable sensors, difficult environments
- Real-time, massively data-parallel processing
  - HL-LHC TDAQ
- On board HPC co-processing
  - power-optimized alternative to FPGAs, GPUs for neural network algorithms





What runs on Neuromorphic Hardware?



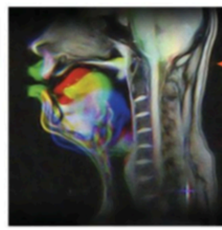
Simple computing elements…

by themselves, limited functional repertoire.

Brain
Machine
Interfaces

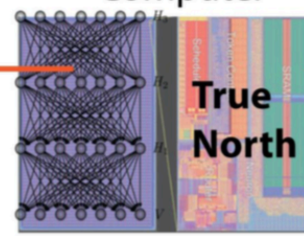
Speak your mind

Prosthetic



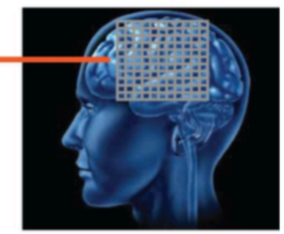
Control Signal

Computer

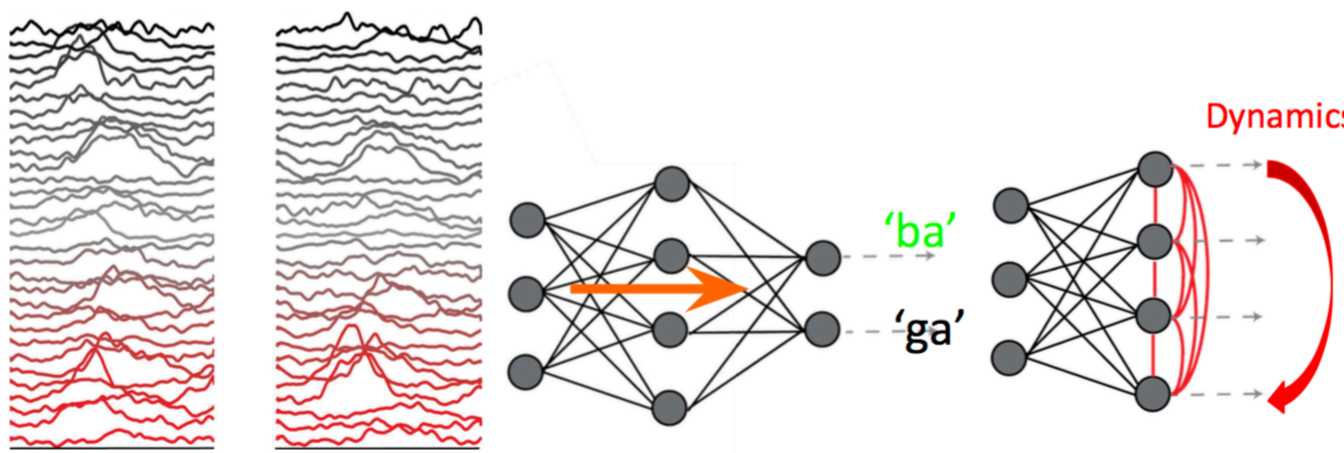


Brain Signals

Brain

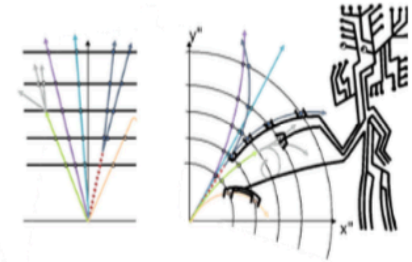


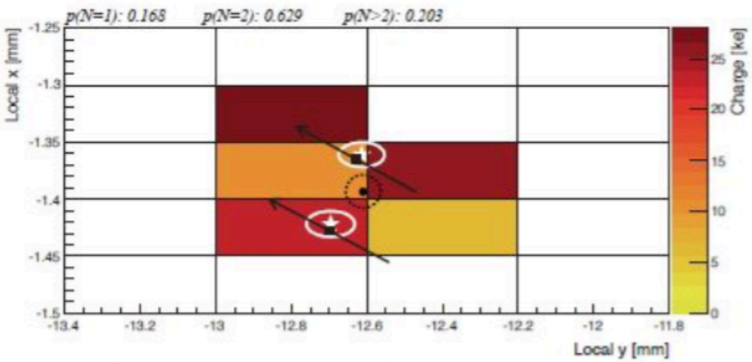
"Mind-reading with artificial brains"



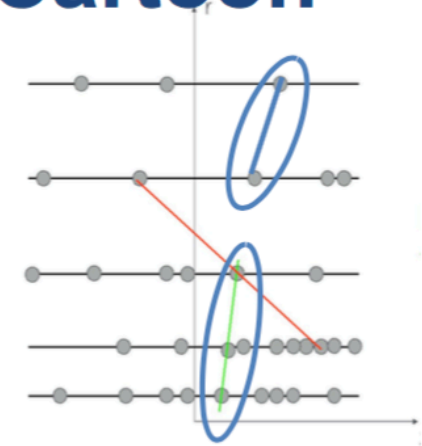
Feature Extraction **Classification** **Kalman Filter**

ML & Tracking: a Cartoon

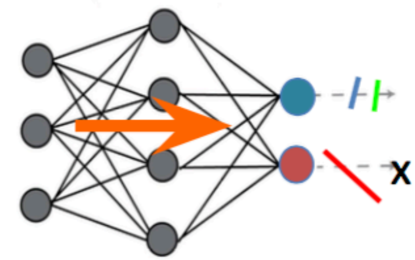




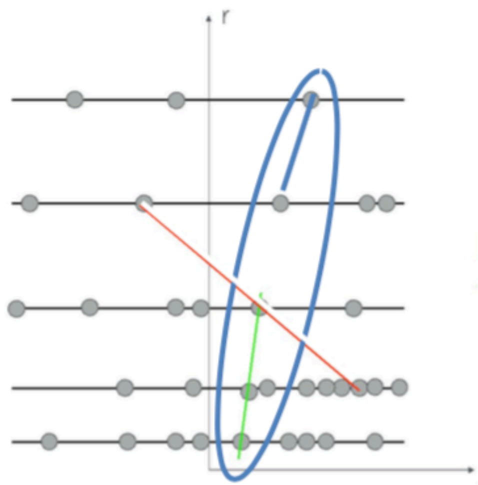
Hits Formation



Seed Creation

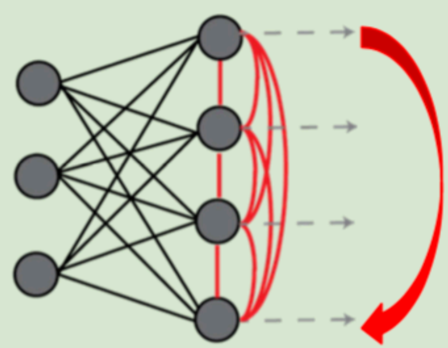


Seed Classification



Track Formation

Kalman Filter



Track Classification

+ possibly:

- Ambiguity Resolution
- Track Fitting
- Vertexing

Kalman Filter Tracking on Parallel Architectures

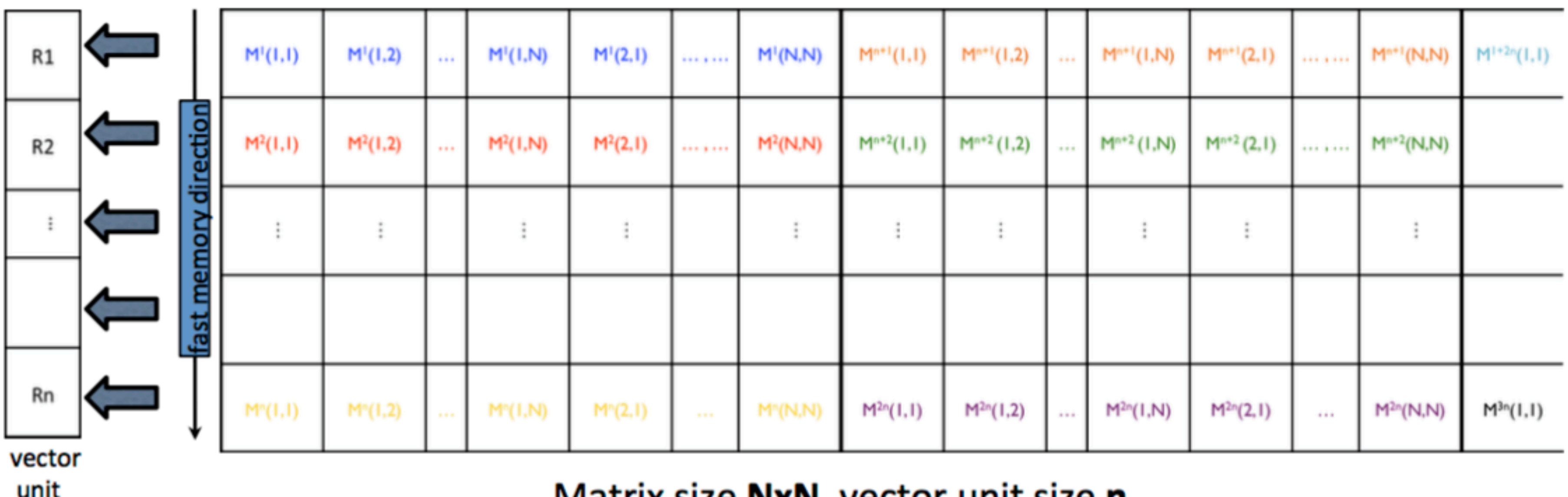
Connecting The Dots 2016: February 22, 2016

UC San Diego, Cornell University, PRINCETON UNIVERSITY

G. Cerati[1], P. Elmer[3], S. Krutelyov[1], S. Lantz[2], M. Lefebvre[3], K. McDermott[2], D. Riley[2], M. Tadel[1], P. Wittich[2], F. Würthwein[1], A. Yagil[1]

Matriplex

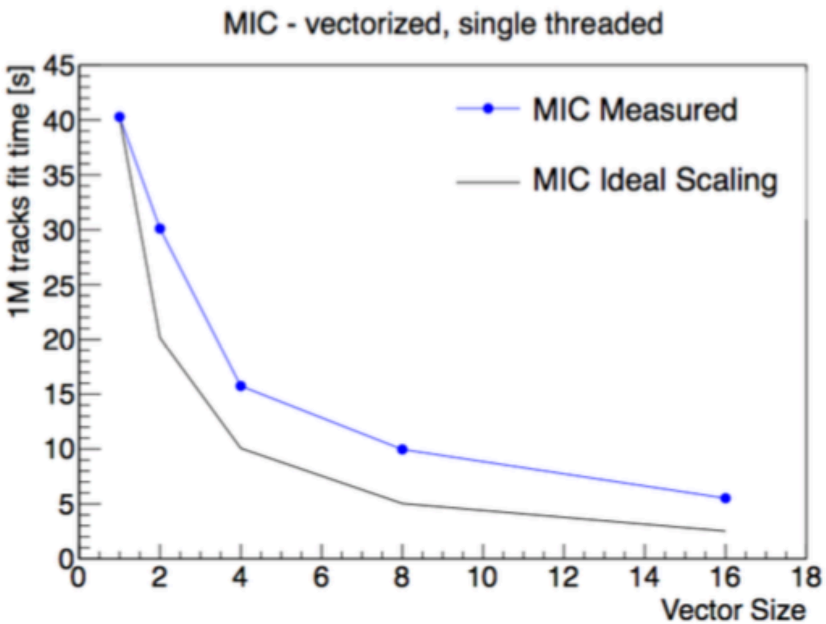
- Matrix operations of KF **ideal for vectorized processing:** however, requires **synchronization** of operations
- Arrange data in such a way that it can loaded into the vector units of Xeon and Xeon Phi with ***Matriplex***
  - Fill vector units with the same matrix element from different matrices: **n matrices working in synch on same operation**

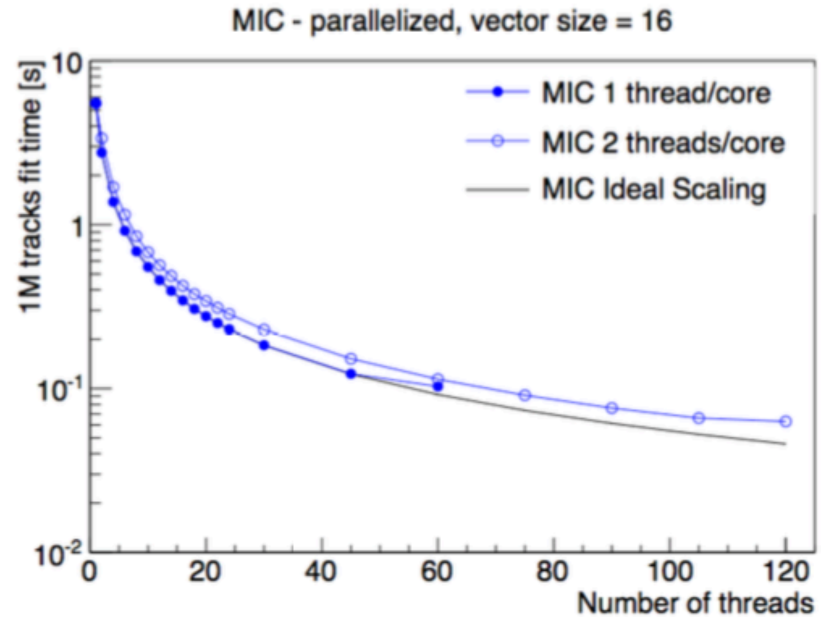


Matrix size **NxN**, vector unit size **n**

Fitting time results

arXiv: 1409.8213

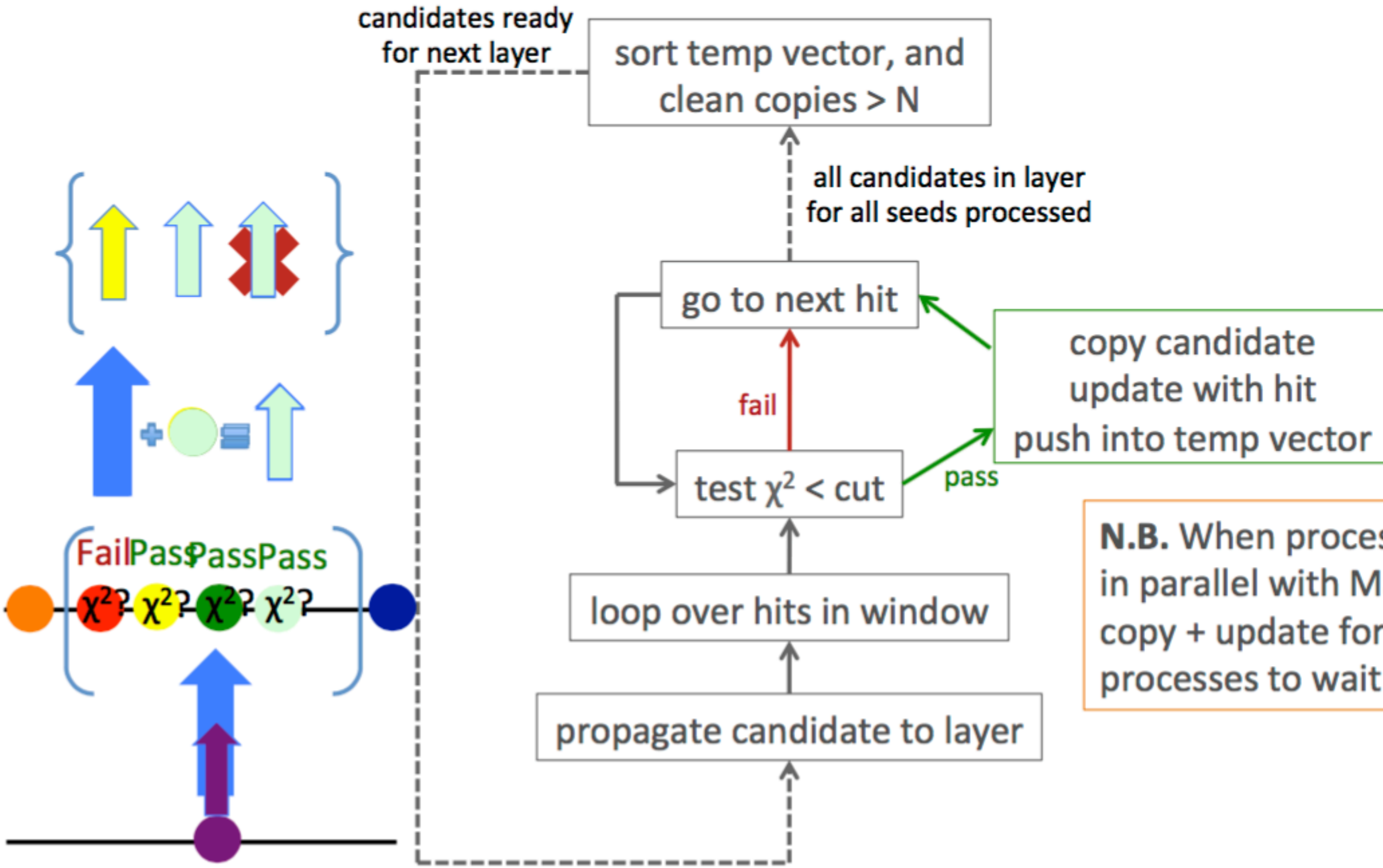




- Significant **speed-up** is observed for both **vectorization** and **parallelization**
  - Similar features on both Xeon and Xeon Phi
  - **Vector utilization is roughly 50%**
  - Parallelization near ideal for 1 thread/core, overhead observed in 2 threads/core
- Loss of vectorization and overhead related to L1 cache issues

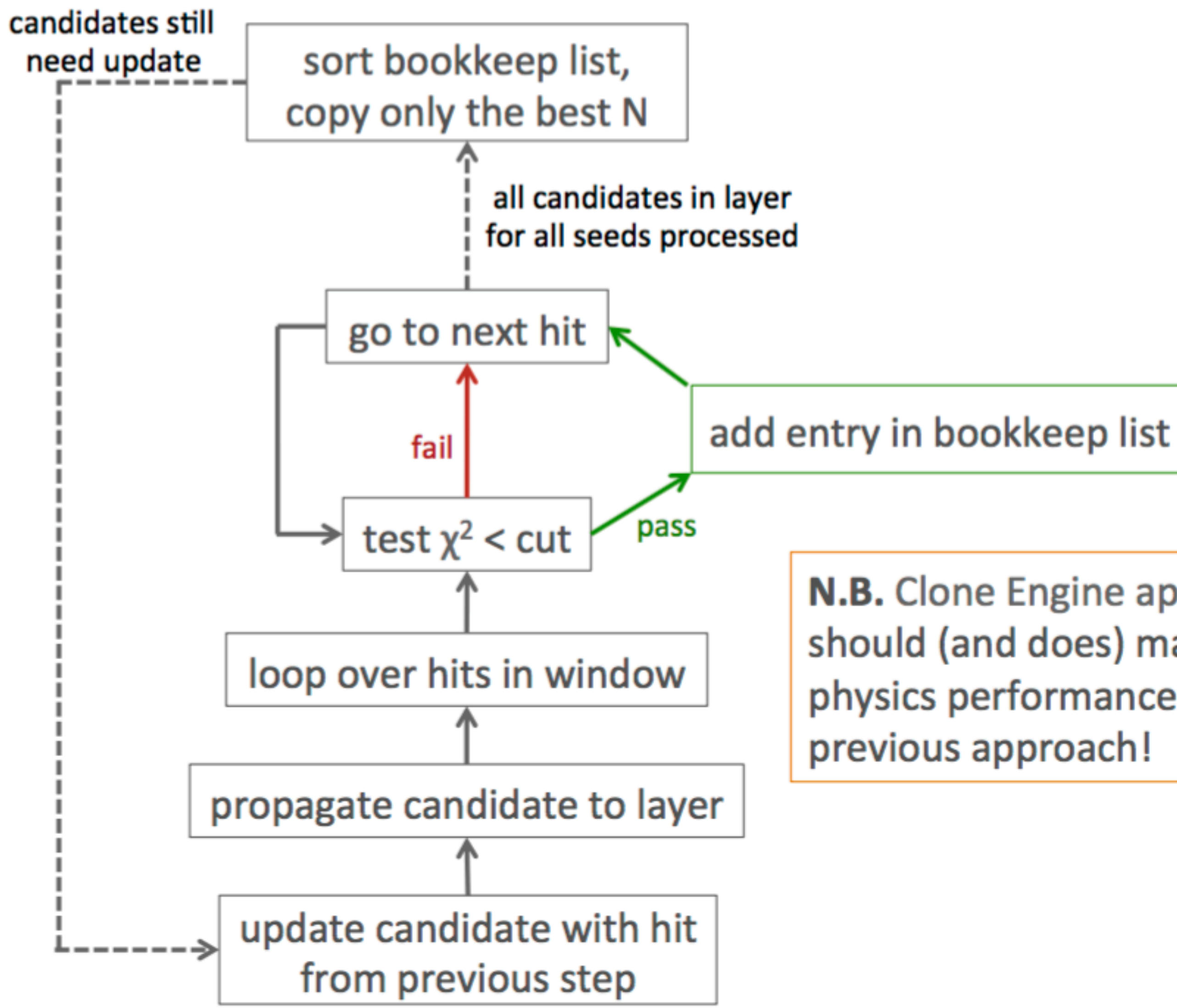
Demonstration of feasibility on fitting, move to track building

Handling multiple track candidates: first approach



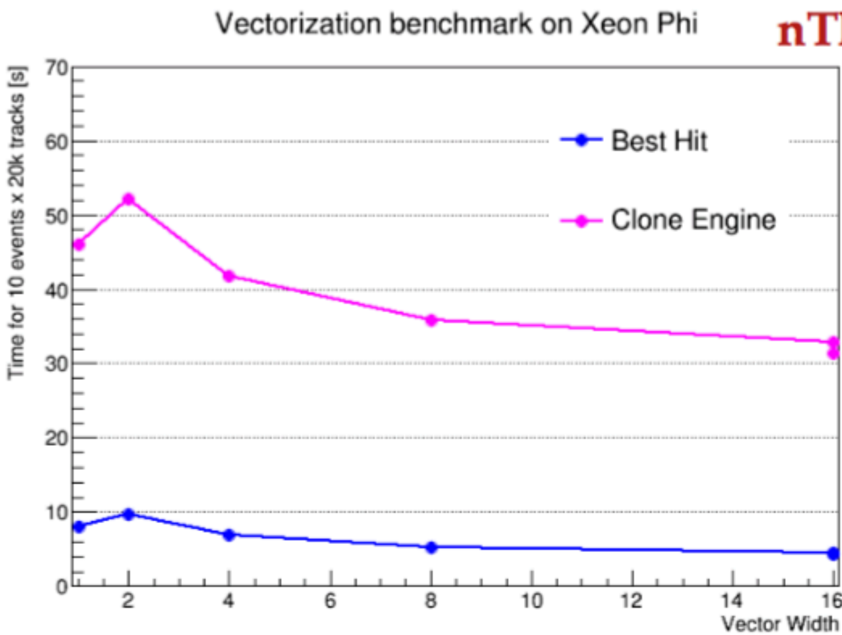
N.B. When processing tracks in parallel with Matriplex, copy + update forces other processes to wait!

Optimized handling of multiple candidates: "Clone Engine"

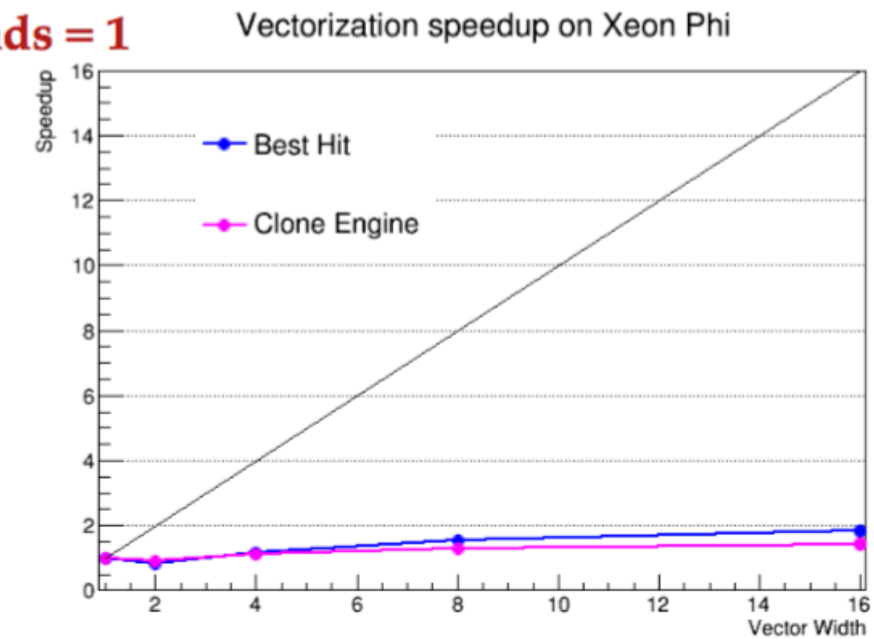


N.B. Clone Engine approach should (and does) match physics performance of previous approach!

Building: Xeon Phi Vectorization

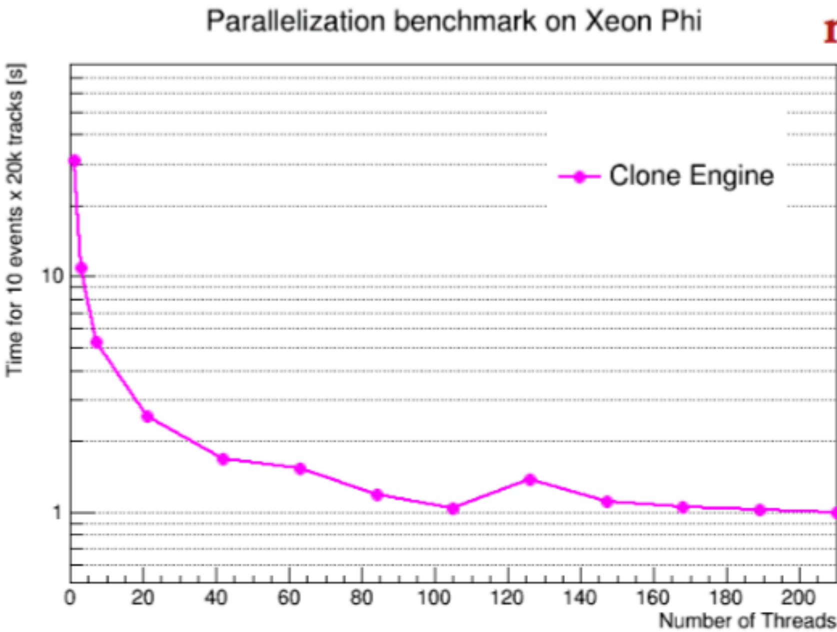


nThreads = 1

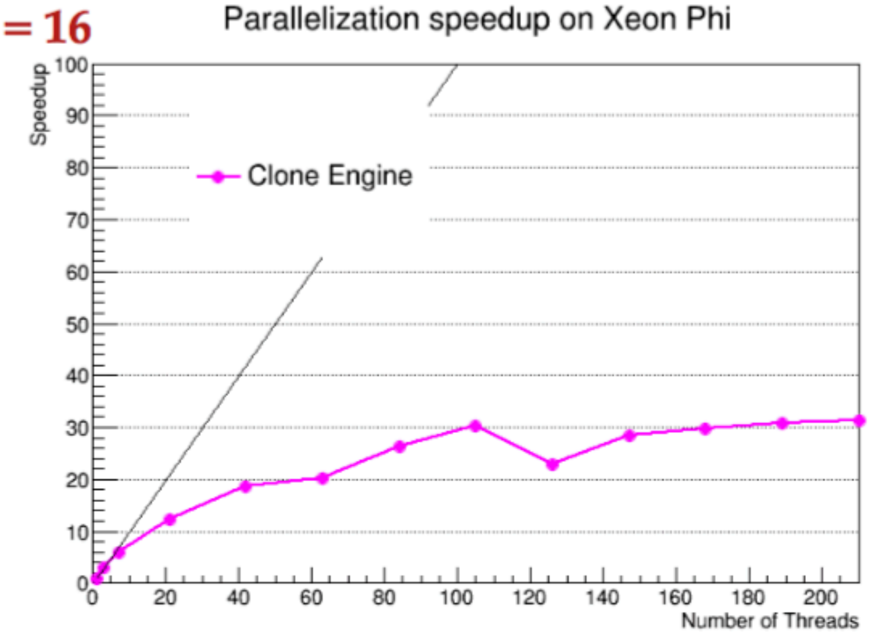


- Mostly same features seen on Xeon Phi
- Notable exception is VW=2, most likely **overhead** in beginning to fill vector units
- Once all registers are filled, achieve speedup of about **2x** for **Best Hit** and **Clone Engine**

Building: Xeon Phi Parallelization



nVU = 16

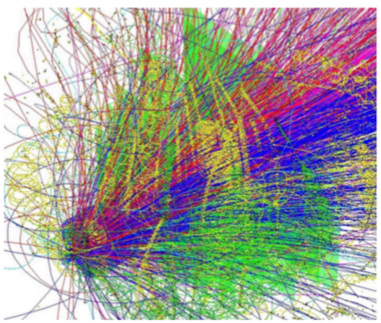


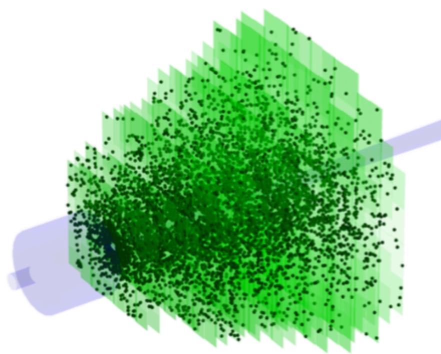
- Features at nThreads = 63, 126: overhead from 2nd thread per physical core, hyper-threading, respectively
- Eventually **recover** speedup from bumps: total of **30x speedup on top of vectorization**

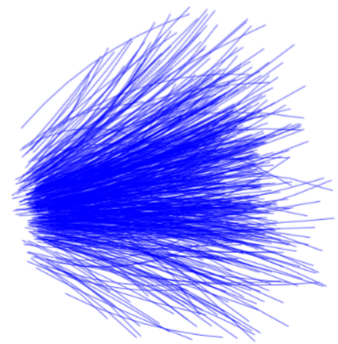
4D Cellular Automaton Track Finder in the CBM Experiment

Ivan Kisel
for the CBM Collaboration

Reconstruction Challenge in CBM at FAIR/GSI





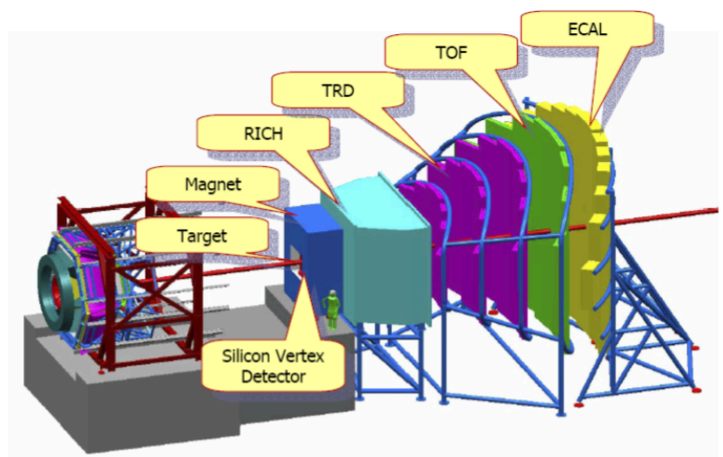


- Future fixed-target heavy-ion experiment
- 10^7 Au+Au collisions/sec
- ~ 1000 charged particles/collision
- Non-homogeneous magnetic field
- Double-sided strip detectors (85% fake space-points)

Full event reconstruction will be done
on-line at the First-Level Event Selection (FLES) and
off-line using the same FLES reconstruction package.

Cellular Automaton (CA) Track Finder
Kalman Filter (KF) Track Fitter
KF short-lived Particle Finder

All reconstruction algorithms are vectorized and parallelized.



Kalman Filter (KF) Track Fit Library

Kalman Filter Methods

Kalman Filter Tools:
- KF Track Fitter
- KF Track Smoother
- Deterministic Annealing Filter

Kalman Filter Approaches:
- Conventional DP KF
- Conventional SP KF
- Square-Root SP KF
- UD-Filter SP
- Gaussian Sum Filter

Track Propagation:
- Runge-Kutta
- Analytic Formula

Implementations

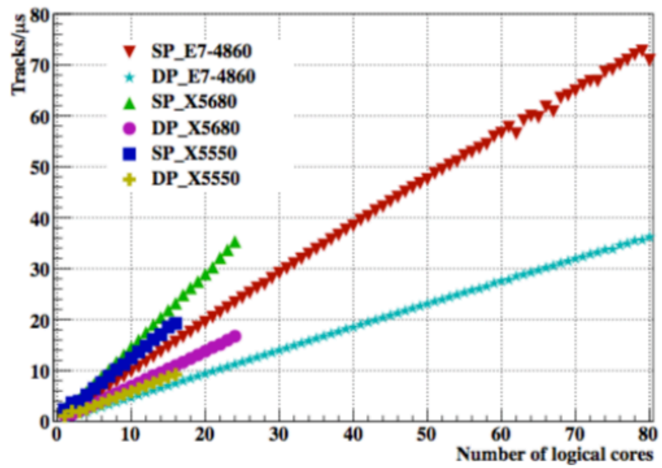
Vectorization (SIMD):
- Header Files
- Vc Vector Classes
- ArBB Array Building Blocks
- OpenCL

Parallelization (many-cores):
- Open MP
- ITBB
- ArBB
- OpenCL

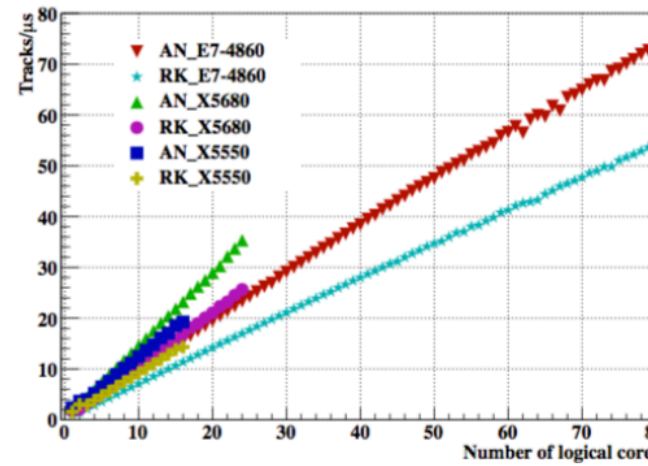
Precision:
- single precision SP
- double precision DP

Comp. Phys. Comm. 178 (2008) 374-383

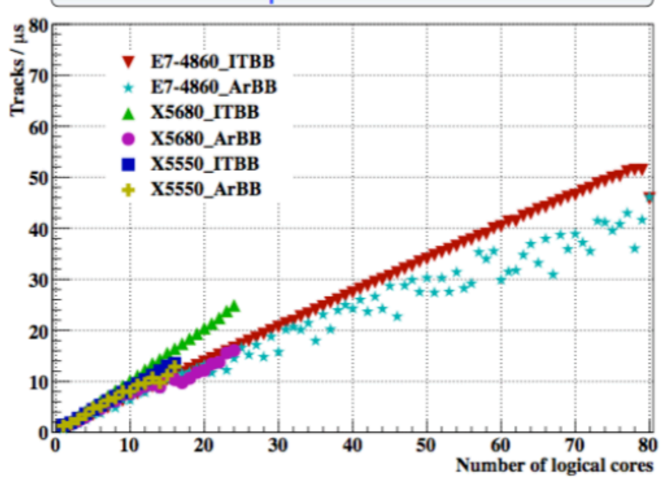
Conventional KF DP vs. SP



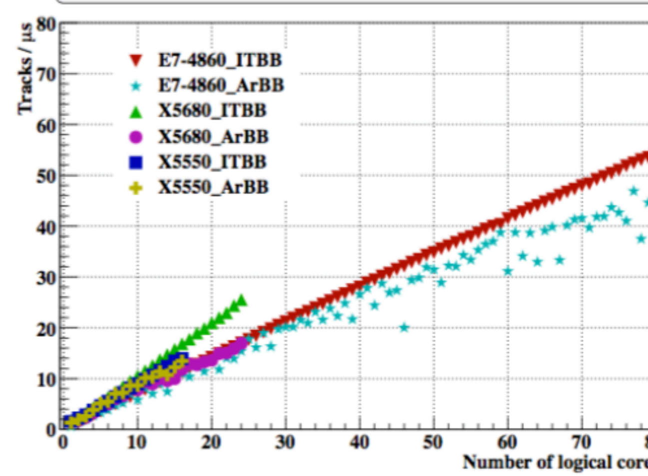
Conventional KF RK4 vs. Analytical



Square-Root KF



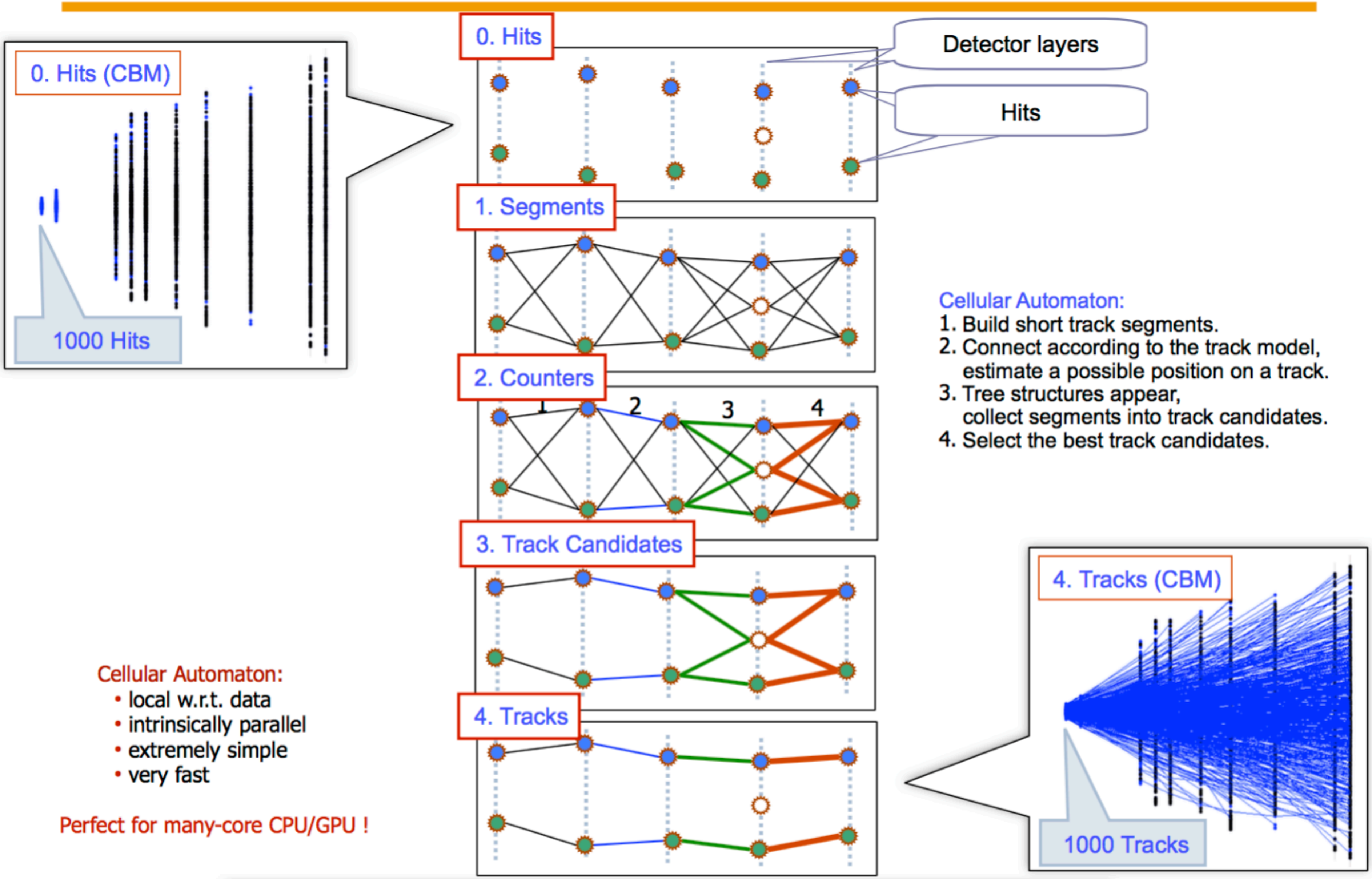
UD KF



Strong many-core scalability of the Kalman filter library

with I. Kulakov, H. Pabst* and M. Zyzak (*Intel)

Cellular Automaton (CA) Track Finder



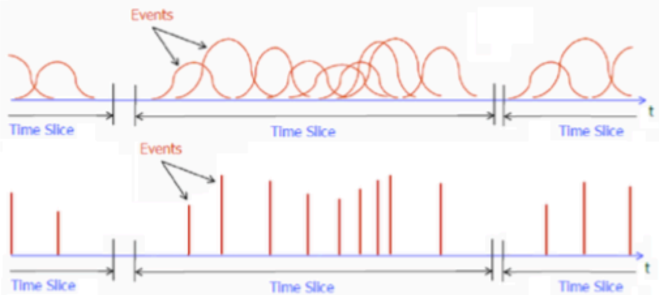
Cellular Automaton:
1. Build short track segments.
2. Connect according to the track model, estimate a possible position on a track.
3. Tree structures appear, collect segments into track candidates.
4. Select the best track candidates.

Cellular Automaton:
- local w.r.t. data
- intrinsically parallel
- extremely simple
- very fast

Perfect for many-core CPU/GPU !

Useful for complicated event topologies with large combinatorics and for parallel hardware

Time-based (4D) Track Reconstruction with CA Track Finder



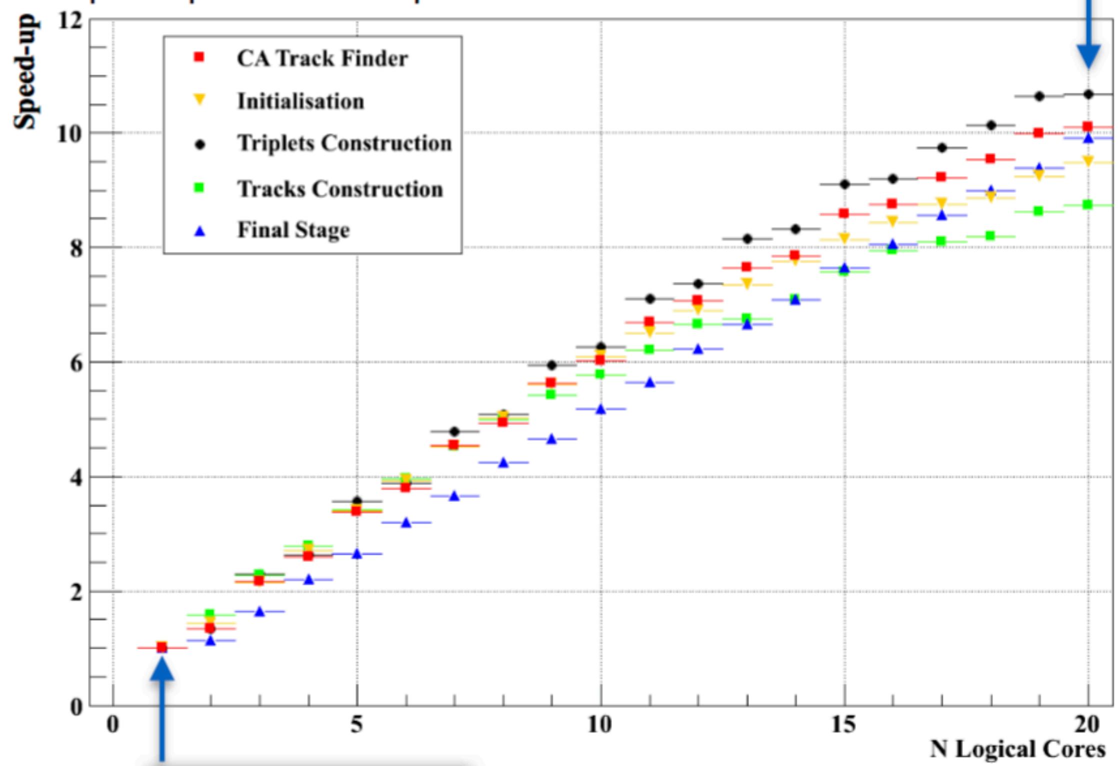
- The beam in the CBM will have no bunch structure, but continuous.
- Measurements in this case will be 4D (x, y, z, t).
- Significant overlapping of events in the detector system.
- Reconstruction of time slices rather than events is needed.

Stage of the algorithm	% of total execution time
Initialisation	8
Triplets construction	64
Tracks construction	15
Final cleaning	13

Efficiency, %	3D	3+1 D	4D
All tracks	83.8	80.4	83.0
Primary high-p	96.1	94.3	92.8
Primary low-p	79.8	76.2	83.1
Secondary high-p	76.6	65.1	73.2
Secondary low-p	40.9	34.9	36.8
Clone level	0.4	2.5	1.7
Ghost level	0.1	8.2	0.3
Time/event/core, ms	8.2	31.5	8.5

Total CA time = 84 ms

Speed-up factor due to parallelization within the time-slice



Total CA time = 849 ms

100 mbias events in a time-slice

4D event building is scalable with the speed-up factor of 10.1; 3D reconstruction time 8.2 ms/event is recovered in 4D case

The Neuro-Z-Vertex Trigger of the Belle II Experiment

Sebastian Skambraks

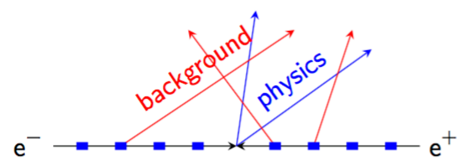
Technische Universität München

Neuro Team

F. Abudinen (LMU), Y. Chen (TUM), M. Feindt (KIT), R. Frühwirth (HEPHY), M. Heck (KIT), C. Kiesling (MPI), A. Knoll (TUM), S. Neuhaus (TUM), S. Paul (TUM), J. Schieck (HEPHY), S. Skambraks (TUM)

Belle II Background

Beam Background Tracks

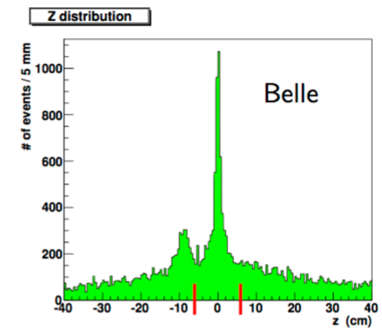


- increase with Luminosity
- tracks from the beamline with displaced z vertices
- main processes:
  - Touschek Effect
  - Radiative Bhabha
  - Beam Gas

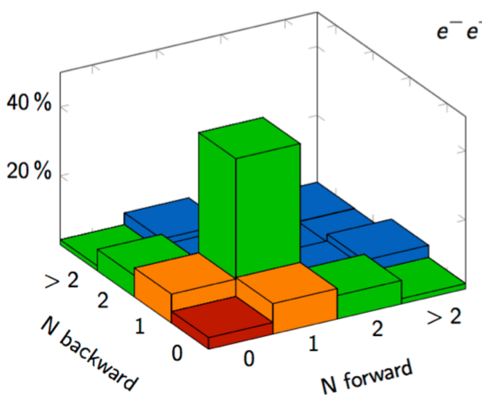
$\Rightarrow$ need z vertex reconstruction at 1st trigger level

NeuroTrigger Goals

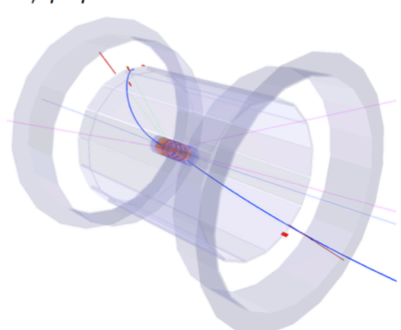
- suppress machine background
- reject tracks from $z \neq 0$ cm
- single track z-vertex resolution < 2 cm
- time window $< 1\,\mu$s



Benefits of a z-Vertex Trigger



$e^- e^+ \longrightarrow \tau^- \tau^+$



- without z trigger: 3 tracks required (≥ 1 in each hemisphere)
- with z trigger: only 2 tracks required

- rescue low multiplicity events
- potential efficiency increase by factor 3.9

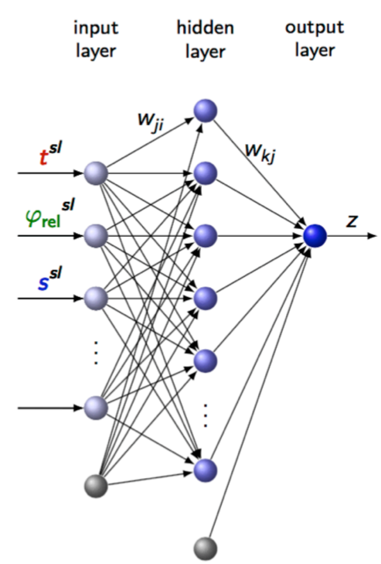
NeuroTrigger - Multi Layer Perceptron

Properties

- supervised machine learning
- function approximation
- short deterministic runtime
- one neuron:
  $y = \tanh(\sum_i w_i \cdot x_i + w_0)$

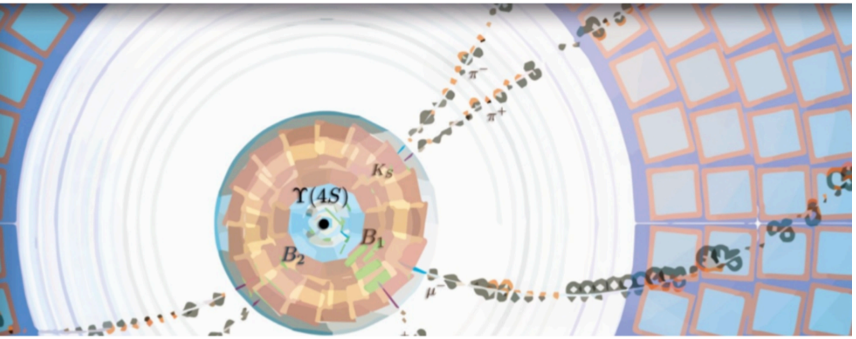
input one TS Hit per SL per track (positions: φ_{rel}, s and drift times: t)

output z estimate

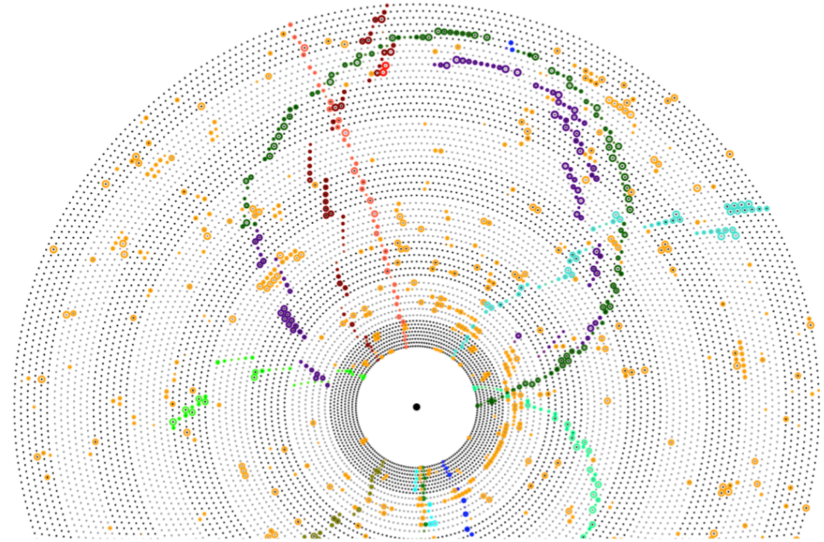


Tracking in the Belle II Drift Chamber.

Vienna - Connecting the Dots 2016



Oliver Frost on behalf of the Belle II collaboration
Deutsches Elektronen-Synchrotron (DESY)
2016-02-22



Hough searches

Hough algorithm

Discretised maximum likelihood optimisation over

$$L(n\,|\{x_i\}) = \sum_i \int \mathrm{d}n\ \delta(d(n, x_i))$$

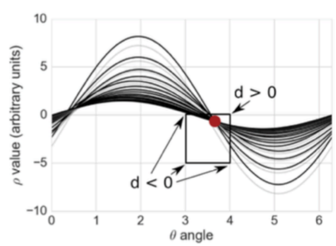
where d is the distance measure of track to hit.
Typically carried out as

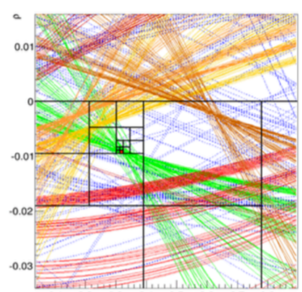
- grid search
- *Fast Hough* bisecting each dimension

over small volumes dn of the parameter space evaluating only the signs of d on the edges.

Refinements

- Weighting of hits versus tracks e.g. on distance d or prior distributions
- Priorisation of search areas
- Overlapping volumes





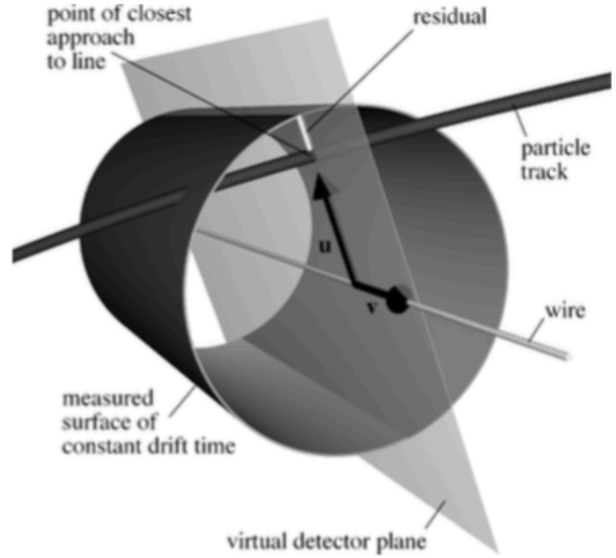
Characteristics

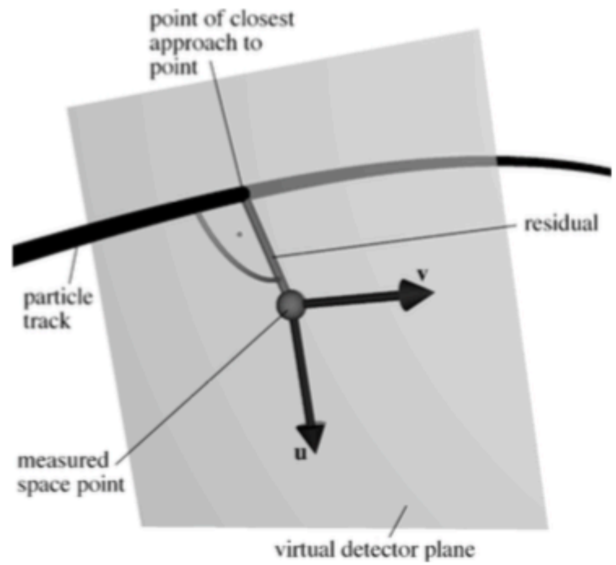
- Templated C++ for all aspects
- Dynamically expanding tree (Quadtree, 2ˆn-Tree) managing node memory
- Weighting of the hits in the tree nodes
- Abitrary dimensional e.g.
  - **Base line xy model:** θ **and** ρ
  - Cosmics base line xy model: d_0, θ, and ρ
  - *Experimental* z inclusion: θ, ρ and $\tan\lambda$
  - **Base line sz :** $\tan\lambda$ **and** z_0
  - *Experimental* full helix d_0, θ, ρ, $\tan\lambda$ and z_0
- Flexible division schemes
  - **Division factors** other than **2** individually for each dimension. 3 or 4 seem feasable.
  - **Overlapping division boundaries**
  - Pre-Sectorisation: Starting with finer binning in the top node to step to specific region of the detector
  - **Alinear divisions** (e.g. to allow finer binning in low curvature regions)
  - Allow abitrary division shapes (circles, spheres, remember that the ordinary hough peaks have butterfly shape)
- Single best and **all nodes higher than threshold weight**

GENFIT2 - Kalman and deterministic annealing filter fitting

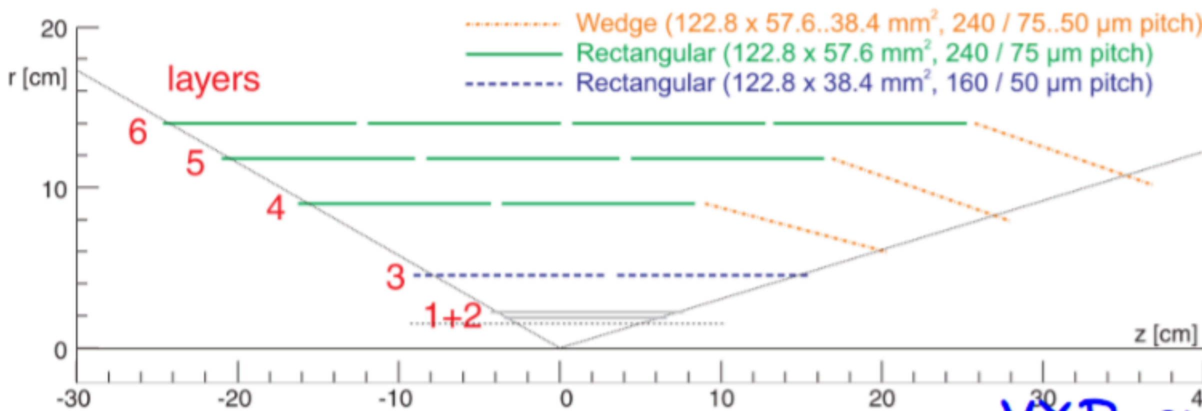
GENFIT2 package

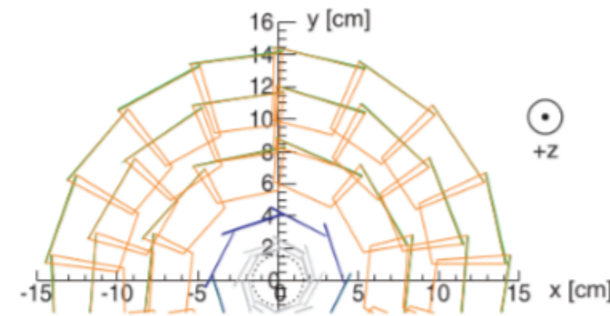
- Proper Kalman fit including
  - material effects
  - magnetic field inhomogenities
  - time of flight corrections
- Deterministic annealing mode for
  - hit cleaning
  - ambiguity resolving
- Specialisable for all kinds of detectors
  - wire chambers
  - planar detectors
  - ...
- *Seeding needed by fast fits*
- https://github.com/TobiSchluter/genfit





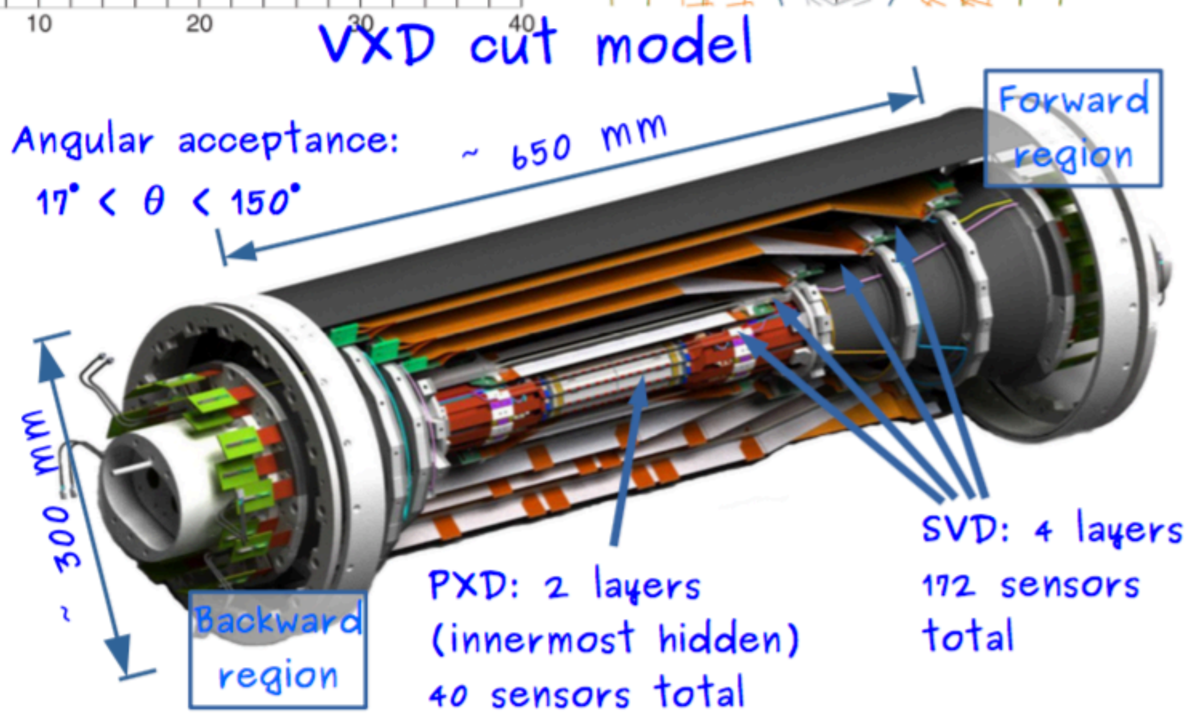
Belle II VXD



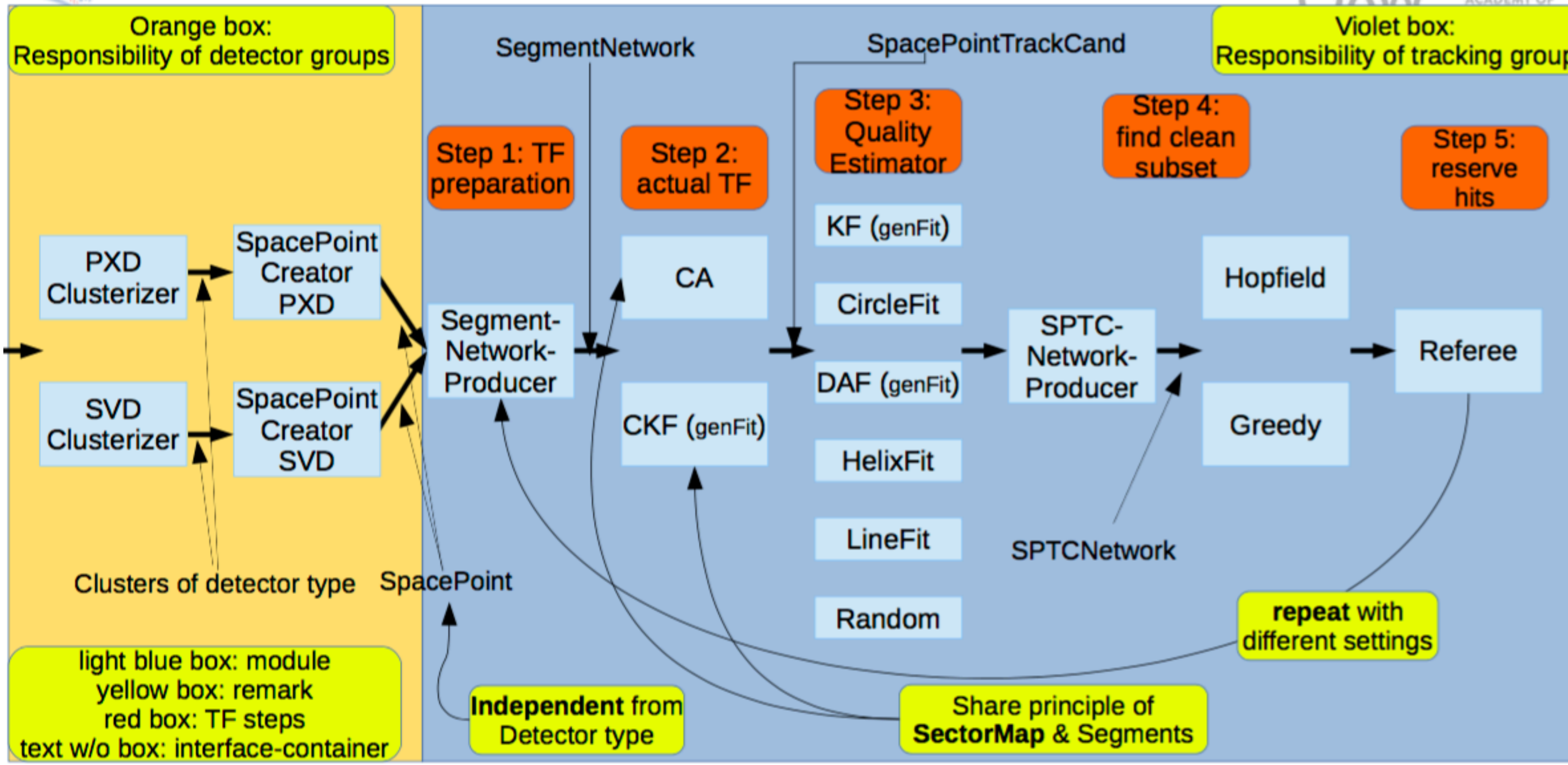


Vertex Detector (VXD) consists of:

- 2 layers of DEPFET Pixels (PXD),
  @ radii: 1.4, 2.2 cm,
  # of pixels ~ 8,000,000,
  thickness of sensitive areas: **75 µm**
- 4 layers of double sided silicon strip (DSSD) sensors (SVD),
  @ radii: 3.9, 8, 11.5, 13.5 cm,
  # of channels: ~ 226,000,
  low material budget:
  X/X0: ~ 0.55% / Layer



Planned structure for the VXDTF (event-part)



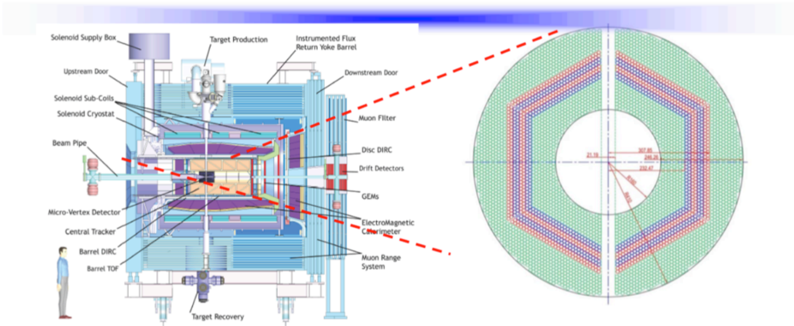
- CA: Cellular Automaton
- KF: Kalman Filter
- CKF: Combinatorial KF
- DAF: Deterministic Annealing Filter
- Hopfield: a neural network of Hopfield type
- SPTC: SpacePointTrackCandidate

Online and offline Pattern Recognition in PANDA

Gianluigi Boca

Universita' di Pavia and INFN, Italy

Straw Tube Tracker system (STT)

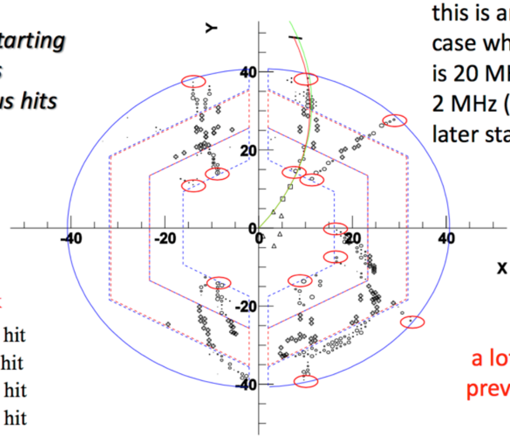


- 4636 Straw tubes
- 23-27 planar layers
  - 15-19 axial layers (green) in beam direction
  - 4 stereo double-layers for 3D reconstruction, with ±2.89 skew angle (blue/red)

7

Road Finding method in the Central Tracker

- *tracklets found starting from red circled hits collecting contiguous hits*



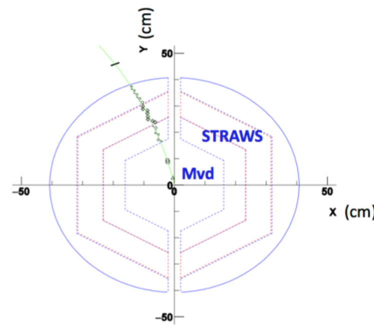
this is an event in the extreme case when the interaction rate is 20 MHz. In PANDA the rate is 2 MHz (20 MHz possibly at a later stage)

MC truth
reco track
△ Mvd Pixel hit
□ Mvd Strip hit
○ Axial STT hit
Skew STT hit
SciTil hit

a lot of pile up hits from previous and subsequent events !

Pattern Recognition on GPUs

Hough transform algorithm

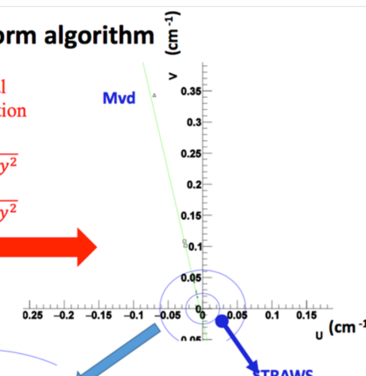


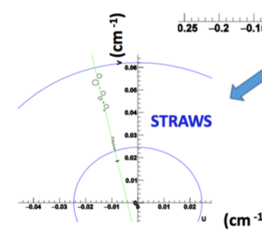
Green line = MC truth Track
△ Mvd Pixel
□ Mvd Strip
○ Stt Axial

Conformal Transformation

$$u \equiv \frac{x}{x^2+y^2}$$

$$v \equiv \frac{y}{x^2+y^2}$$





genfit2 in PandaRoot

- *genfit2:* experiment-independent track fitting tool. External package
- *genfit2* is announced to be a general tool, for every B field.
  Revision checked in PandaRoot: *genfit2-v1826*; old revision: *genfit-v400*.
- In $\overline{\text{P}}$ANDA different field maps:
  solenoid (2T)
  dipole (2Tm)
  “twister” *TransMap*
- *genfit* (rev 400) and *genfit2* (rev 1826) are both available in PandaRoot:
  the current PandaRoot trunk-rev 28747 provides a switch to run both.
- Int
  Ge

genfit2 design

- Measurements
  - PlanarMeasurement
  - WireMeasurement
  - SpacePointMeasurement
  - Virtual detector plane, measurement coordinates, covariance
- Track Representations
  - Track parameterization
  - Extrapolation through material and magnetic field
  - Propagate time of flight
  - Particle hypothesis
- Track fitting algorithms
  - Use measurements and TrackReps to calculate fit results
  - ition

6

- Experiments using genfit2: Belle II, $\overline{\text{P}}$ANDA, GEM –TPC, FOPI, SHip, AFIS,...)

The family is growing....

genfit2 details

- Track fitting in genfit2 based on:
  1) Measurements
  2) Track representation
  3) Fitting algorithms: Kalman fitters linearizing the transport around the state prediction;
     Kalman filter linearizing around the reference track;
     DAF;
     GBL.

- Measurements: objects containing measured coordinates from a detector;
  provide functions to construct virtual plane;
  provide measurement coordinates and covariance in that plane.
- Track representation: combine track parameterization and track extrapolation code

$\left.\right\}$ TRACK

- Fitting algorithms: use measurements and track representation to calculate fit results;
  start value for fit needed, e.g. from pattern recognition

Why shall we use genfit2?

- General implementation of the Kalman fitter
- Track representation included
- Alignment studies: GBL interfaced
- Vertexing: RAVE interfaced
- Many parameters for fit convergence user-adjustable
- Independent on detector geometry
- Valid tool for every B field
- Suited to track low momentum particles: $\overline{\text{P}}$ANDA and BELLE II: p>50 MeV/c

How difficult is to interface genfit2 with another framework?

It depends...
My experience with PandaRoot:
~3 months to get the *GenfitTool* interface running inside /development/brunch/;
~3 months for debugging (PidCorrelator, memory leak, ...);
>3 months to perform generalized tests with all mass hypotheses and different p_{beam}

- Documentation: common paper with Belle II and $\overline{\text{P}}$ANDA planned.

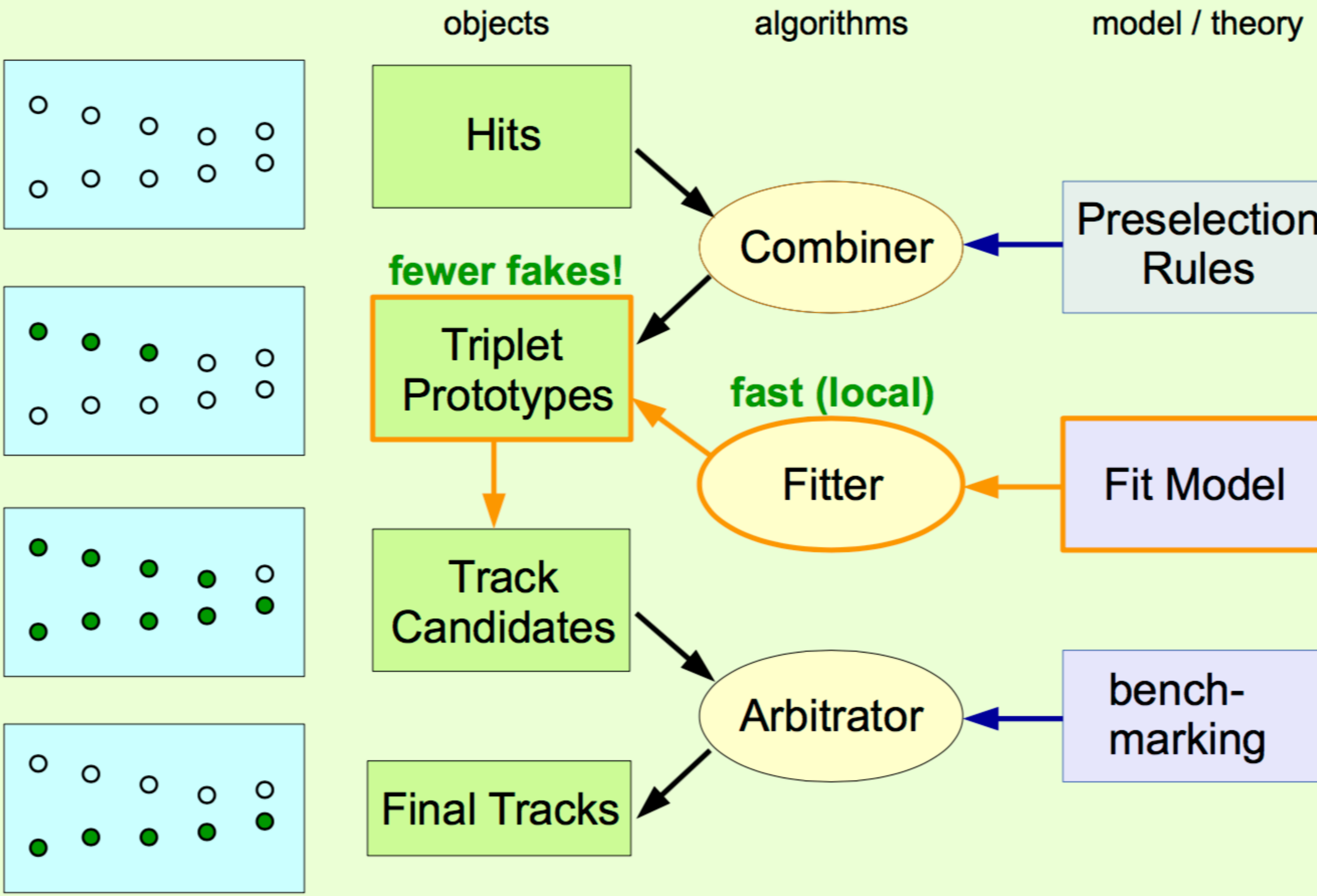
A New Track Reconstruction Algorithm based on Hit Triplets and Broken Lines

André Schöning
Universität Heidelberg
Physikalisches Institut

with contributions from N.Berger, M.Kiehn, A.Kozlinskiy

Connecting the Dots 2016
HEPHY Vienna
22.February 2016

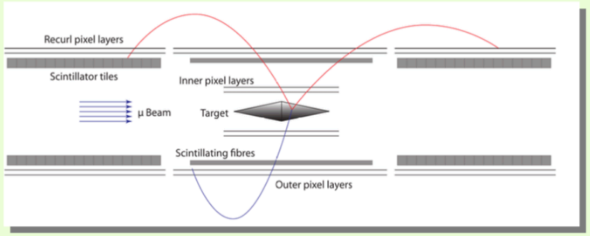
New Triplet Tracking Concept

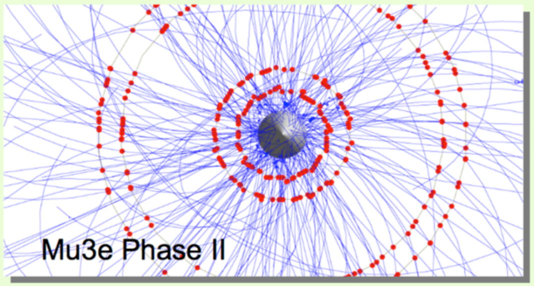


Example: Mu3e Experiment

Search for
$\mu^+ \rightarrow e^+ e^+ e^-$

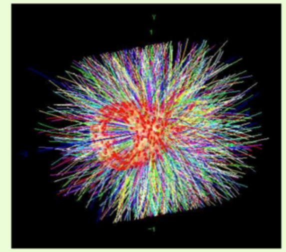
particle momenta:
$p < 53$ MeV/c





→ uncertainties dominated by **multiple scattering**!

Typical LHC Experiment

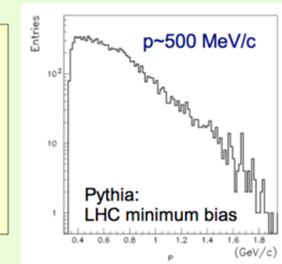


- O(10000) charged tracks at HL-LHC
- material budget ~ 2-3% / layer
- 10-12 layers per experiment for R≤1m

Uncertainties:

- hit resolution ~ 15 μm → $\sigma_\Theta \approx$ 0.15 mrad
- scattering: $\sigma_\Theta \propto \frac{1}{p}\sqrt{X/X_0} \rightarrow$ **p_{crit} = 15 GeV/c**

p≤10 GeV/c	**p≥10 GeV/c**
• **multiple scattering** uncertainty dominates	• **hit uncertainty** dominates
• **~99% of particles**	• ~1% of particles



$$\chi^2 = \underbrace{\frac{\Theta_{MS}^2}{\sigma_\theta^2} + \frac{\Phi_{MS}^2}{\sigma_\phi^2}}_{\text{multiple scattering}} + \underbrace{\cancel{\sum_{jk}(x_j-\xi_j)V_{jk}^{-1}(x_k-\xi_k)}}_{\text{hit uncertainties}}$$

neglected for the moment

- solution: (if triplet is not too small → discussion later)

$$R_{3D} = -\frac{\kappa\,\tilde{\Phi}_C\,\sin^2\theta + \beta\,\tilde{\Theta}_C}{\kappa^2\,\sin^2\theta + \beta^2}$$

$\kappa,\ \tilde{\Phi}_C,\ \beta,\ \tilde{\Theta}_C$ are parameters calculated from three hit coordinates

- uncertainty + fit quality:

$$\sigma(R_{3D}) = \sigma_{MS}\sqrt{\frac{1}{\kappa^2\,\sin^2\theta + \beta^2}} \rightarrow \chi^2_{min} = \frac{1}{\sigma_{MS}^2}\,\frac{(\beta\,\tilde{\Phi}_C - \kappa\,\tilde{\Theta}_C)^2}{\kappa^2 + \beta^2/\sin^2\theta_C}$$

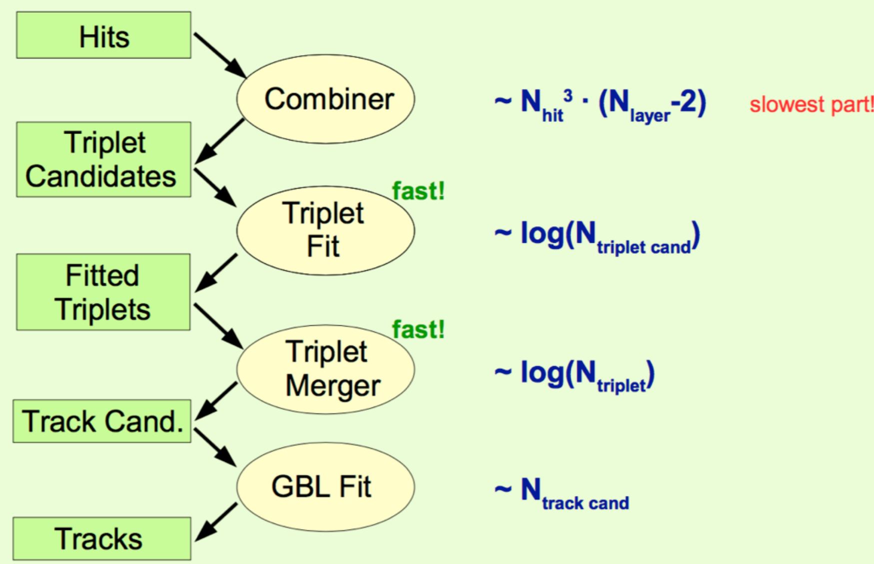
- multiple scattering uncertainty (calculated from above obtained parameters):

$$\sigma_{MS} = \frac{b}{R_{3D}}$$

the **scattering parameter *b*** is given by the effective material thickness and the magnetic field strength

$$\boxed{b} \approx \frac{4.5\ \mathrm{cm\ T}}{B}\sqrt{X/X_0}$$

Overview of Algorithm



Wire Cell Reconstruction Method and Software Library
for Liquid Argon Time Projection Chambers

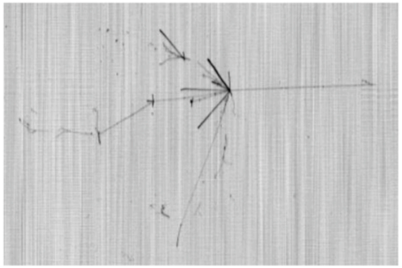
Brett Viren
for the BNL Wire Cell Group

Physics Department

BROOKHAVEN
NATIONAL LABORATORY

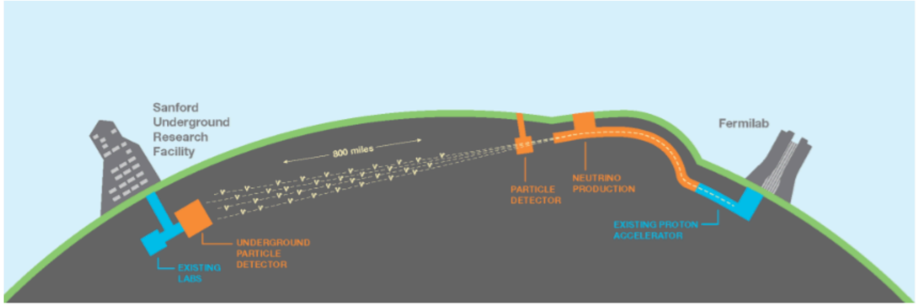
The origin of LArTPC technology for Neutrinos: C. Rubbia, 1977 led to **ICARUS, the first, large-scale LArTPC**.

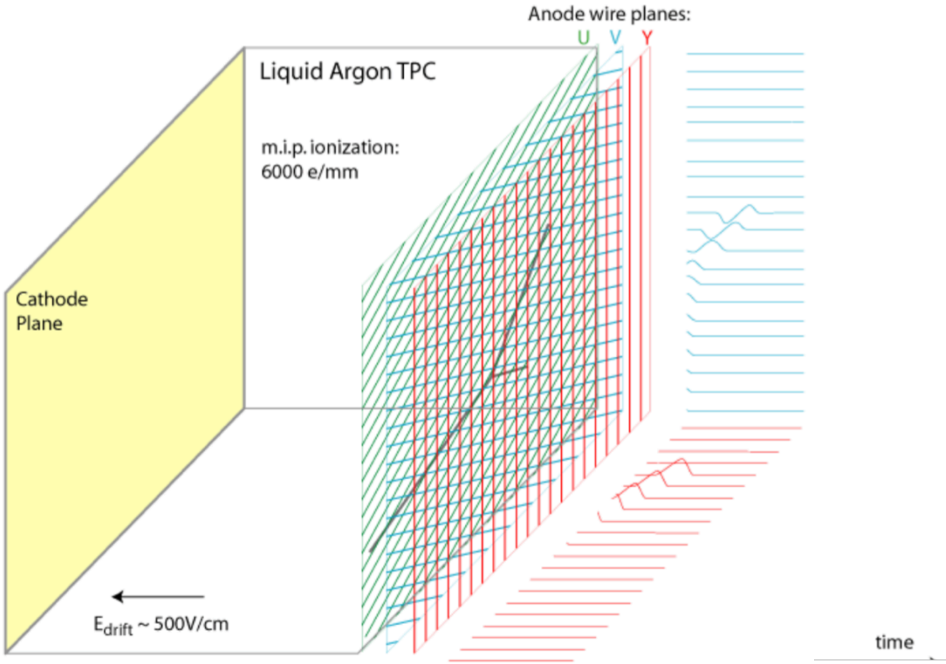
- $2\times$ 300 t modules.
- Took data in the Gran Sasso tunnel, Italy from CERN neutrino beam.
- Moving to Fermilab as part of the **Short-Baseline Neutrino** Program.

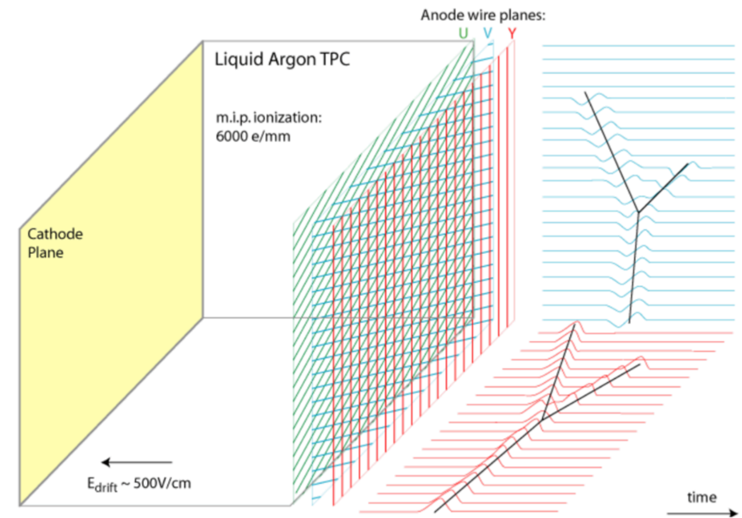


LArTPC Experiments - DUNE DEEP UNDERGROUND NEUTRINO EXPERIMENT

"International **mega-science** project"



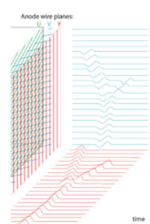




LArTPC Data

LArTPC can produce **huge quantities** of **high-resolution** data from **large detector volumes**:

- 10^4 – 10^6 channels
- 2MHz @ 12 bit waveform digitization
- each "event" spans several milliseconds



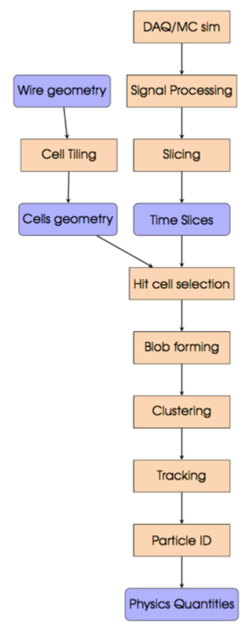
Two general DAQ readout strategies:

Full Stream: read out entire waveform (**MicroBooNE**)

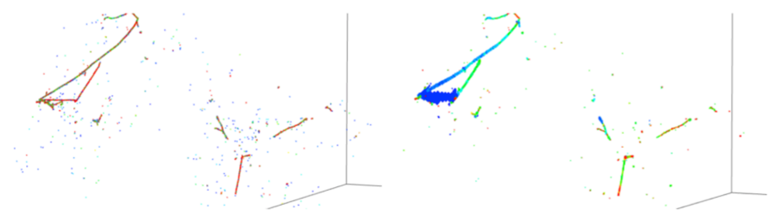
- **30GB/s in 120 MB "events".**
- DUNE at FS would produce 5 TB/s in 25 GB "events"!

Zero Supression: only save waveform parts with significant activity (**DUNE**)

- Threshold chosen based on noise (E_{thesh} ~0.1 MeV/wire)
- 2.5 MB/event → **100's TB/year**
- requires rejection of natural ^{39}Ar decay @ **50 PB/year**



The Payoff: imaged 3 GeV ν_e interaction



True energy depositions. Wire Cell Imaging.

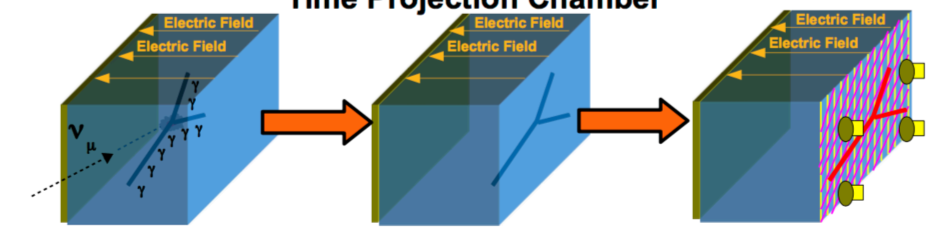
- **Excellent imaging** of major features and isolated activity.
  - → a static 2D view doesn't do it justice! Follow link to view it online.
- **Residual ambiguity** seen as wide blue patches.
  - → **Inherent problem of tomography using low number of viewing angles**
  - → Will pursue an **iterative** approach: constrain ambiguous regions after reconstructing the good parts to the kinematics-level.

Deep Learning Event Reconstruction In LArTPC

Amir Farbin

LArTPC

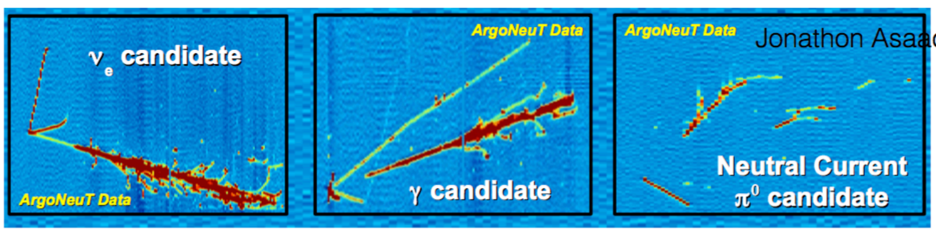
Time Projection Chamber



Neutrino interaction in LAr produces ionization and scintillation light

Drift the ionization charge in a uniform electric field

Read out charge and light produced using precision wires and PMT's

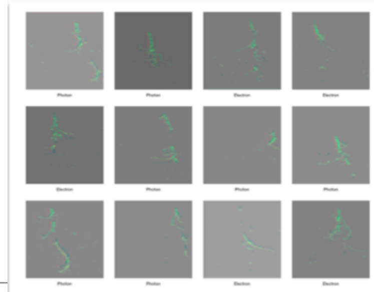


Jonathon Asaadi

Tracking, Calorimetry, and Particle ID in same detector.
Goal ~80% Neutrino Efficiency.

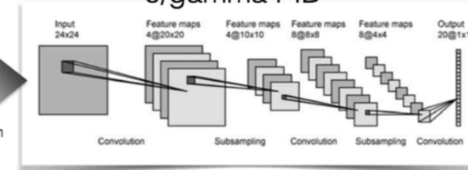
Deep Learning

e/gamma PID



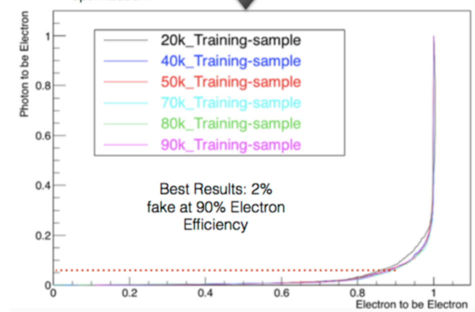
Raw Data: Wire ADC vs Time x Planes
(LArIAT Simulation)

Train



Deep Convolutional Neutral Network
(GoogLeNet)

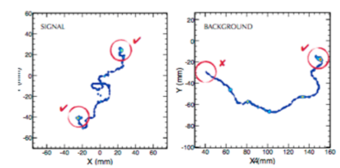
Out of the box *Feasibility Study* with No attempt at optimization.

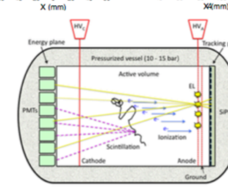


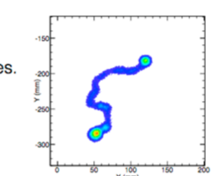
Best Results: 2% fake at 90% Electron Efficiency

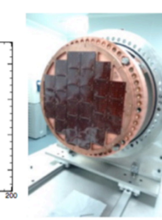
NEXT Experiment

- Neutrinoless Double Beta Decay using Gas TPC
- SiPM readout give 3D images. Best use 3D Convolution.
- PMTs measure energy, low spacial resolution:
  - Source moved around volume to calibrate response.
- Signal: 2 Electrons. Bkg: 1 Electron.
  - Hard to separate, because of high multiple scattering.
- First DL Study: what is the ultimate performance? Are we limited by the physics?
- Fast simulation of energy deposits with a few effects put in. (100k of sig/bkg each)
- 1 mm effective resolution. Real detector planning 1 cm.
- Projected 3-D into 3 2D planes and put into 3 color intensities. Use GoogleNet.
- 99.96% Signal Efficiency for 0.2% Background.
- Next step is to reduce resolution to optimize detector.









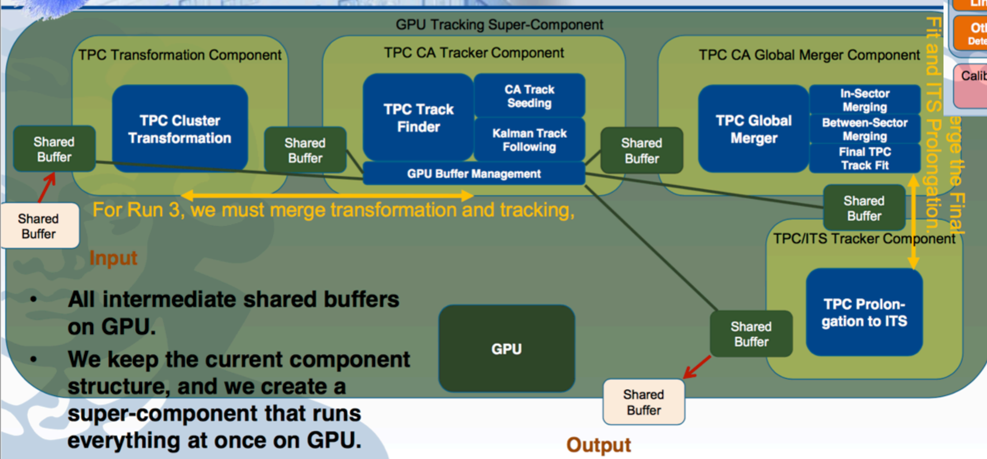
Online reconstruction and calibration with feed back loop in the ALICE High Level Trigger

David Rohr, drohr@cern.ch
Frankfurt Institute for Advanced Studies

Challenges for Online Calibration

- Calibration involves long-running tasks, which cannot run in the HLT in an event-synchronous way. (Problem A)
  → **Asynchronous tasks**
- HLT is loop-free, calibration is created at the end of the chain, and must be used at the beginning. (Problem B)
  → **Zero-MQ sidechannel, that feeds back calibration asynchronously.**
- Calibration requires TPC and ITS tracks.
  - We need fast online tracking algorithms. (Problem C)
    → **GPU TPC Tracking**
  - There is a chicken and egg problem: calibration needs reconstruction and vice versa. (Problem D)
    → **Fast standalone (simple) ITS tracking to prepare calibration.**
- Calibration must process in the order of 5000 events (in Pb-Pb), which takes some time.
  - We cannot cache events that long. (Problem E)
    → **Apply the calibration after some delay as long as it is stable.**
    (The first events of a fill are processed online without calibration.
    For offline, the full calibration is available.)

Next developments in tracking

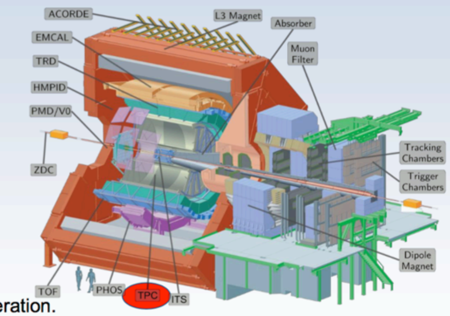


- **All intermediate shared buffers on GPU.**
- **We keep the current component structure, and we create a super-component that runs everything at once on GPU.**

Challenges in this talk

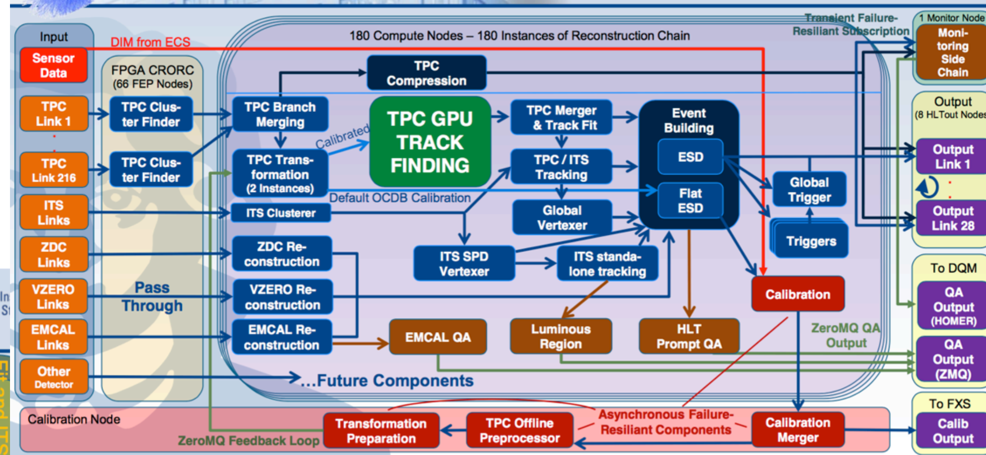
- **Reconstruction of particle trajectories in the TPC is computationally very expensive:**
  - Several thousand tracks per event.
  - High combinatorial complexity.
- **As a gas-based detector, the TPC is sensitive to calibration.**
  - Environment variables such as temperature and pressure affect the calibration.
  - The conditions change during a run.

→ **Challenging tasks for the HLT:**
  - Needs fast reconstruction algorithm for online operation.
  - Detectors must be continuously calibrated online.



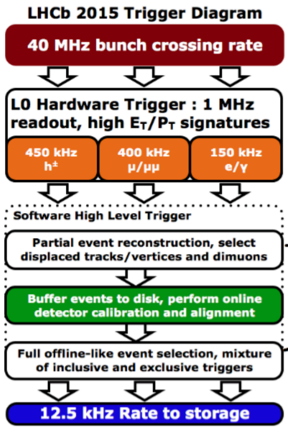
- **Online calibration improves online reconstruction quality.**
- **Online calibration can save offline compute resources by replacing offline calibration passes.**
- **Future experiments (ALICE in Run 3, FAIR at GSI) rely on online processing and thus online calibration.**

Overview of current HLT components



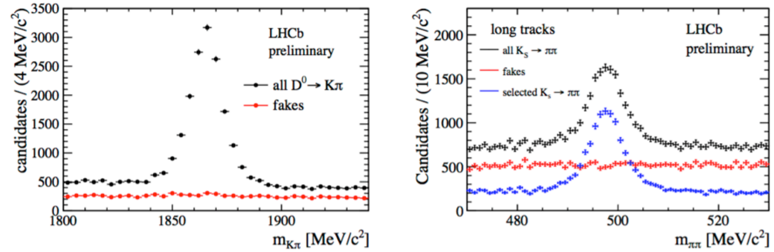
Machine learning and parallelism in the reconstruction of LHCb and its upgrade

Michel De Cian, University of Heidelberg
on behalf of the LHCb collaboration



stage	time/event
HLT1 (150kHz)	≈ 40 ms
HLT2 (12.5Hz)	≈ 800 ms

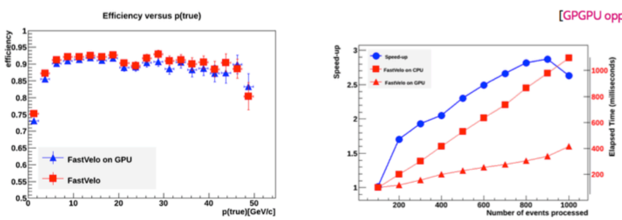
Fake track rejection (II)



- General overhaul of "ghost probability" for Run II, improved timing by factor $\mathcal{O}(90)$
  - Using less than 0.5 ms per event.
  - Customized neural net by improved timing of the activation function (see backup for details).
- Use a combination of a cut on the track-χ^2 and the output of the neural net to reject fake tracks after the Kalman filter.
- Large reduction of fakes without signal loss (remaining $\approx$ 14.0%).
  - Reduces combinatorics in HLT2 for trigger selections by 40%.

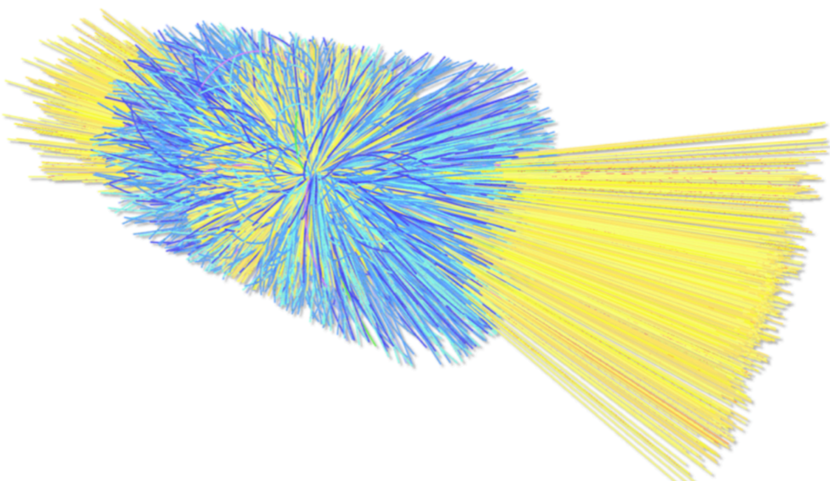
Velo tracking with GPUs

[GPGPU opportunities at the LHCb trigger]



- To learn about GPU systems: Implement the Velo tracking on a GPU (using CUDA) and run it in "parasitic" mode in Run II.
  - Run it in the monitoring farm of the HLT.
- Efficiencies are very similar, but using "event level parallelisation", the GPU obviously gains.
- The question is not only: What has the best performance, but also: what has the best performance/cost.

- Thanks to machine learning, LHCb managed to reduce its rate of fake tracks by about 40% in the trigger while maintaining the efficiency.
- The time-consumption of the reconstruction was reduced by a factor of 2, thanks to the help of parallelization / SIMD in hot-spots of the software.
- The upgrade of LHCb will use a purely software-based trigger.
  - This poses severe restrictions on the timing-budget.
- Many ideas are explored for massively parallelizing parts of the track reconstruction. Next months will lead to a decision.
- Parallelization will be crucial for track reconstruction in the future.



Track Reconstruction Challenges for FCC-(hh) and HL-LHC

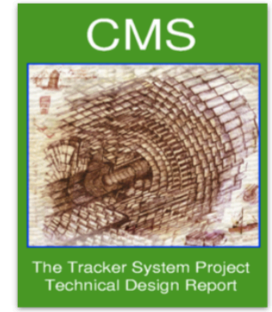
A. Salzburger (CERN)

From TDR to data taking

- TDRs for ATLAS/CMS Tracker TDRs written in late 1990s
  - track reconstruction software strongly inspired by LEP experiments

1997

- ATRECON framework (FORTRAN)
- full move to Gaudi-Athena around 2002/2003



1999

- ORCA framework (2000/2001)
- move the CMSSW in 2006

Step 2

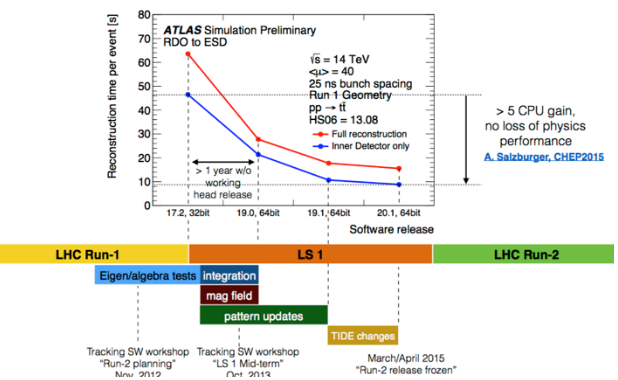
- the journey begins

- LEP & LHC preparation
- LHC Run-1
- LHC LS-1
- LHC Run-2/3
- LHC Upgrade
- HL-LHC Run
- FCC(-hh)

time line

Pimp your engine

- CPU performance driven SW campaign to optimise the ID tracking



> 5 CPU gain, no loss of physics performance

A. Salzburger, CHEP2015

Software/Algorithms - ways to speed up

approximate reality, simplify your models — π ≈ 3

optimise your code — see talk **J. Hrdinka**

1 €	2 DM
2 €	4 DM

prepare your work, use look-up tables

ctrl C / V — take shortcuts, or simply cheat

zzzz — don't do anything, work on demand — on off

use new technologies, increase your work force

Conclusion

- LHC Run-1 was a great success for track reconstruction
  - wonderful results with outstanding performance
- LHC experiments have largely "updated" the track reconstruction for Run-2 and Run-3
  - this will most likely work (just)
- HL-LHC will be a shift in paradigm
  - instantaneous pile-up of up to 200 interactions expected
  - not sustainable with current approaches/software

    needs R&D not only on the detectors, but also on the algorithms, the SW
- FCC-hh
  - weill, let's see … however, let's not forget that we want to do precision physics

Performance requirements for the Phase-2 Tracker Upgrades for ATLAS and CMS

Duccio Abbaneo

Requirements from the experiment as a whole

The Trigger is much more challenging at HL-LHC

Higher luminosity requires higher first-level trigger rate(*) and/or more effective event selection

Selection algorithms become less effective in high pileup!

Solution: higher first-level trigger rate AND longer latency

ATLAS: 100 kHz → 1000 kHz
2.5 μs → 6.0 μs

CMS: 100 kHz → 750 kHz
3.2 μs → 12.8 μs

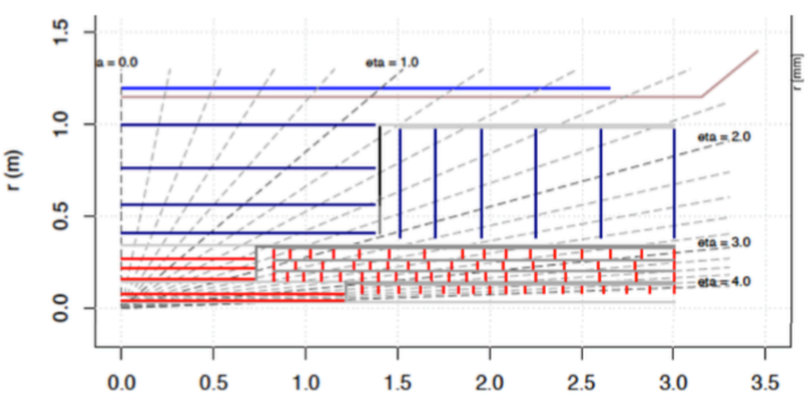
In addition *in CMS*:

The Outer Tracker contributes to the L1 trigger decision!

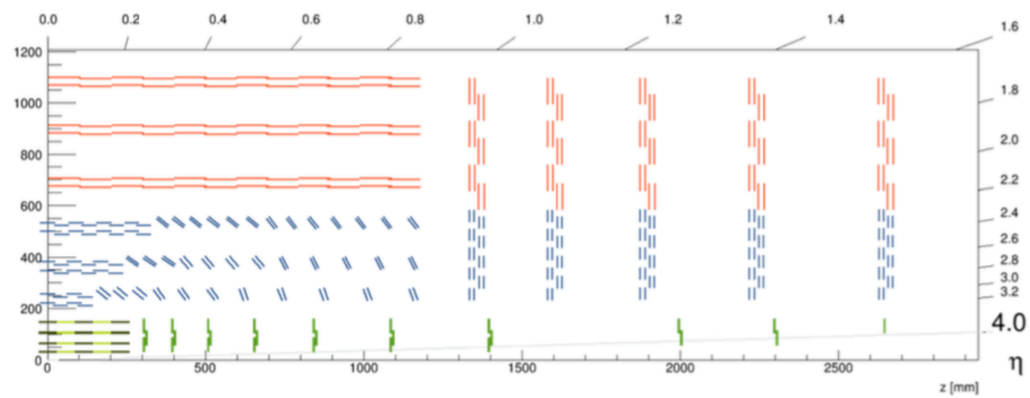
This choice drives several differences between the two tracking detectors
To some extent it is motivated by other differences between ATLAS and CMS
(ATLAS has higher-granularity information from the calorimeters, CMS has a stronger B field)

(*) *To confuse the reader, the first-level trigger is called Level-0 in ATLAS and Level-1 in CMS*

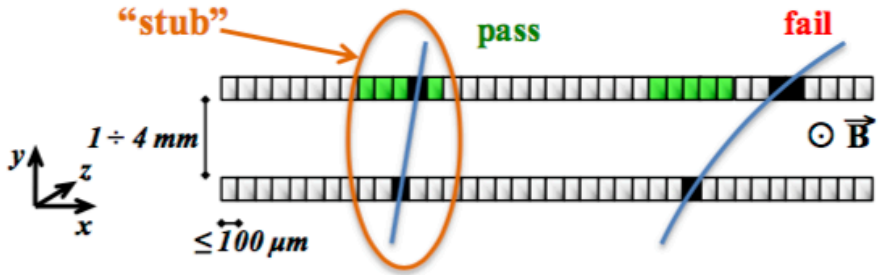
ATLAS

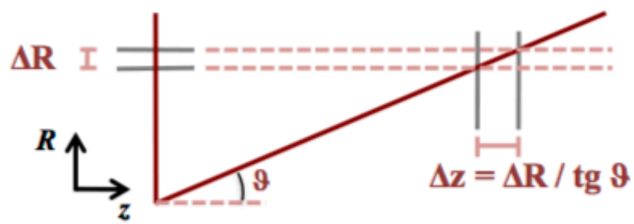


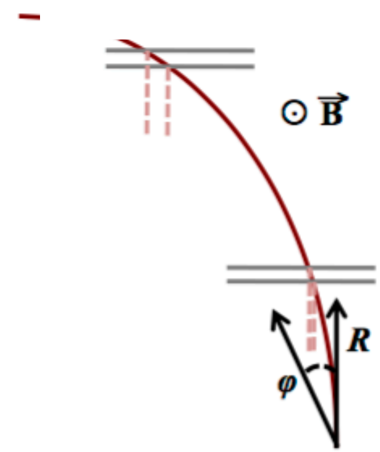
CMS



Working principle of p_T modules

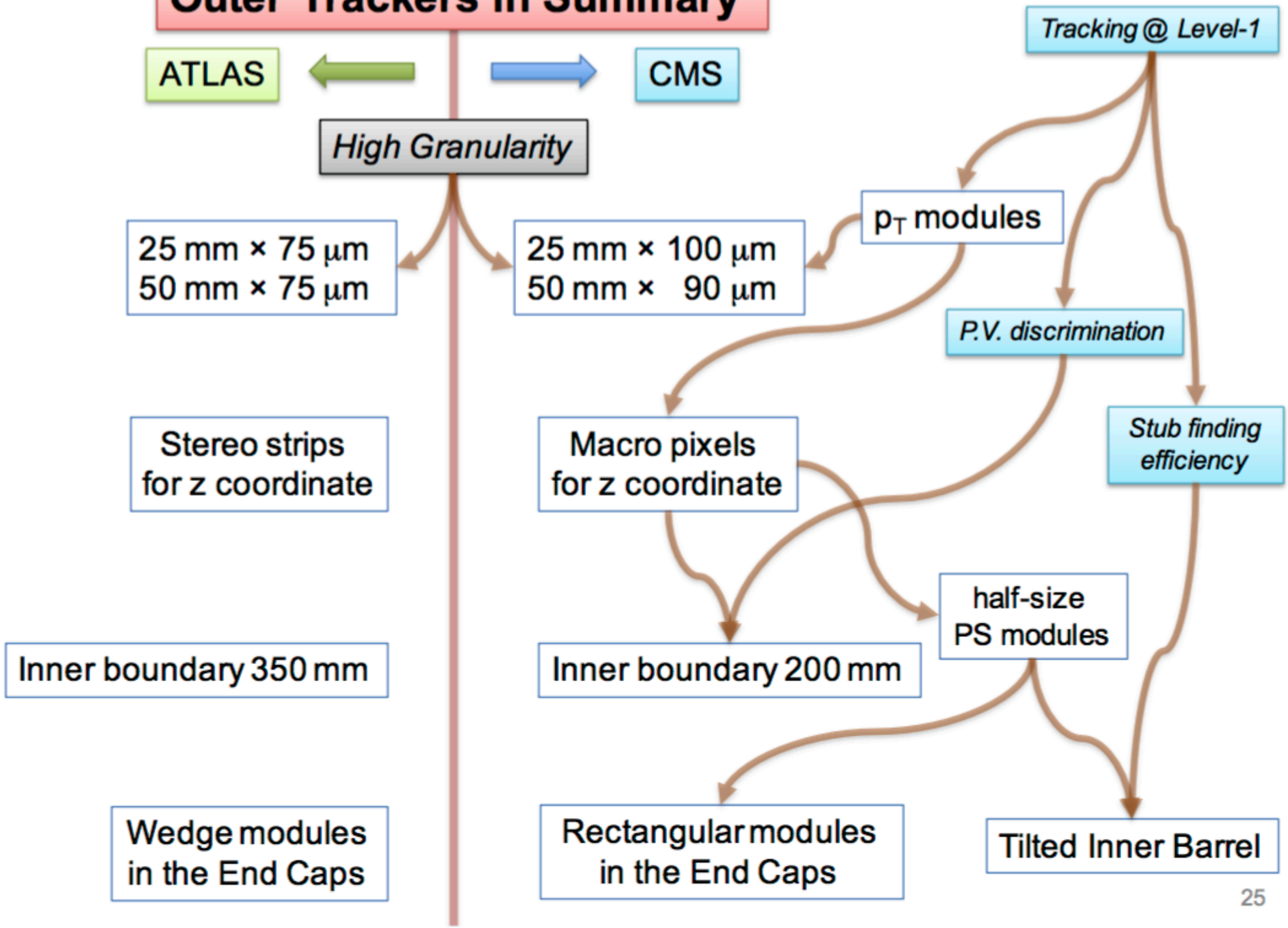




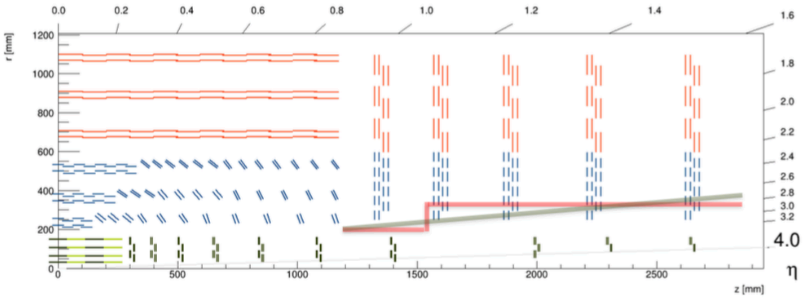


- Sensitivity to p_T from measurement of $\Delta(R\varphi)$ over a given ΔR
  - For a given p_T, $\Delta(R\varphi)$ increases with R
  - In the barrel, ΔR is given directly by the sensors spacing
  - In the end-cap, it depends on the location of the detector ($tg\vartheta$)
    - End-cap configuration typically requires wider spacing, and yields worse discrimination
- Optimize selection window and/or sensors spacing
  - To obtain, as much as possible, consistent p_T selection through the tracking volume
- The concept works down to a certain radius
  - 20÷25 cm with the CMS magnetic field and a realistic 100 μm pitch
- **No room for stereo strips!!**

Outer Trackers in Summary



Pixel Detector Layout: CMS



Started from a fairly "conventional" layout

- Barrel geometry inspired by "phase-1" detector
- End Cap geometry inspired by Outer Tracker Double-Disks
  - Different options for module size under consideration
  - Large pixels (×4 surface) could be used in the outermost layers/rings, to save power

BUT:

Installation of the central section around the beam pipe requires [...] opening in the forward part

The detector slides in with an inclined angle!

The OT/Pixel boundary must be at larger radius in the [...] part

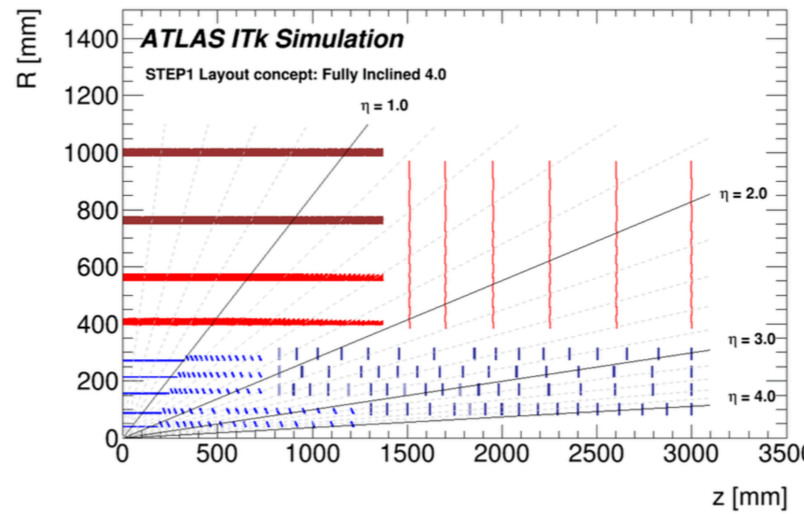
A step? Where? How large?

A conical boundary? (… watch complication...)

Work in progress

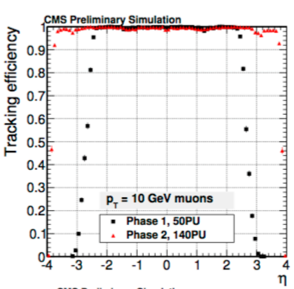
Pixel Detector Layout: ATLAS

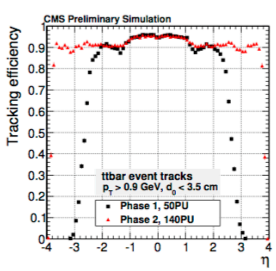
…. to be combined with a creative end-cap layout…



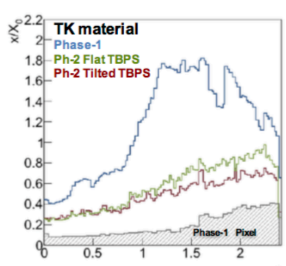
Performance: CMS

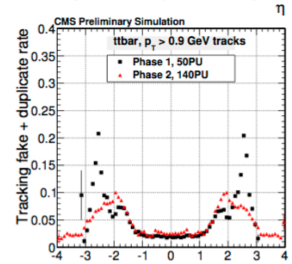
- Compare Phase-1 @ 50 PU with Phase-2 @ 140 PU

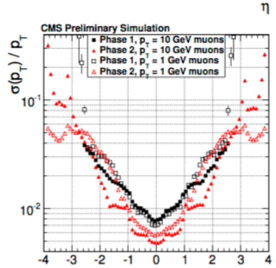




Phase-1 tracking software adapted to phase-2 geometry





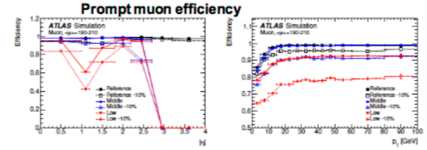


Expect substantial improvement also in z_0 resolution and b-tagging

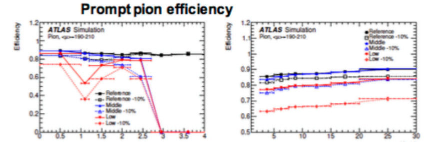
Too early to give quantitative estimates

Performance: ATLAS

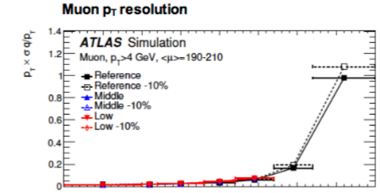
Prompt muon efficiency



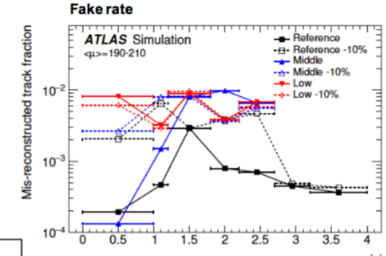
Prompt pion efficiency



Muon p_T resolution



Fake rate



Black markers are for full coverage

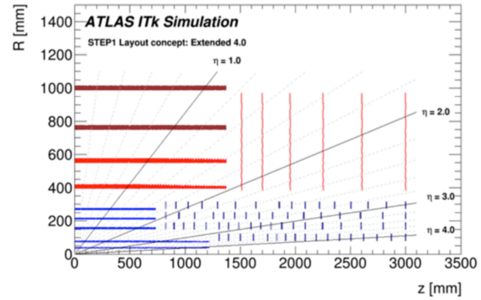
Blue and red markers are cost-saving options with reduced coverage

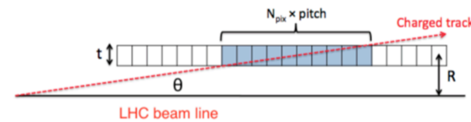
Use Of Pixel Cluster Information In Pattern Recognition

Sasha Pranko

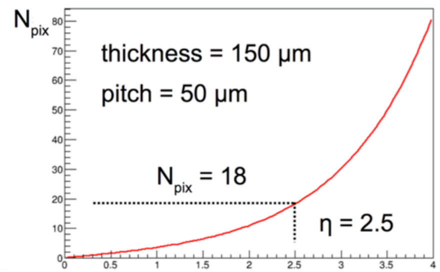
(LBNL)

"Extended" Layout Option For Pixel Barrel Detector





$$\tan\theta = \frac{t}{(N_{pix} - \delta) \times p}, \quad \delta \approx 1$$



- **Main idea:** long clusters = "tracklets", providing initial precise estimates of θ and z_0
  - Seed pattern recognition
  - Potential to reduce fake rate
  - Potential to reduce CPU time

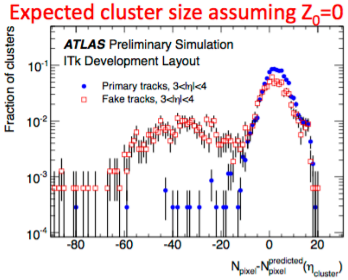
Basic information about sensors:

Barrel Layer-0,1 & inner end-cap ring: 50×50×100 μm³

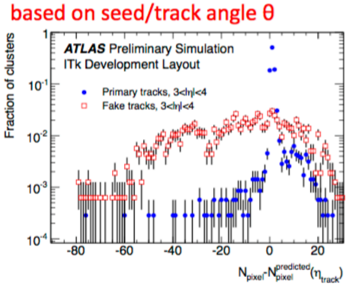
Barrel Layer-2,3,4 & end-cap: 50×50×150 μm³

How Can Cluster Size Information Be Used?

Expected cluster size assuming Z_0=0



Refined expected cluster size based on seed/track angle θ

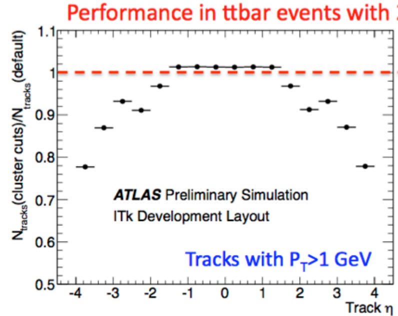


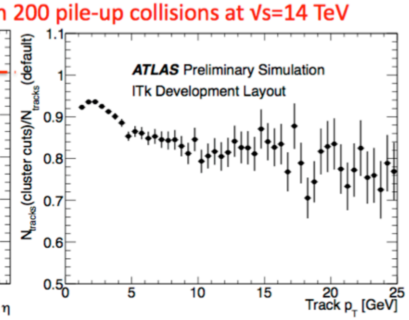
- STEP-1: pre-processing
  - Many pixel clusters are not even used in seed and track finding
  - Safely get rid of as many spurious clusters as possible to reduce the number of space points to be considered in $O(N^2)$-loops at STEP-2
- STEP-2: find track seeds made of 3 space points (next slide)
  - Strategy-1: reject seeds where pixel cluster size is incompatible with θ_{seed}
  - Strategy-2: search for clusters in small cone determined by cluster size in inner layers
- STEP-3: combinatorial track finder
  - Attach cluster only if cluster size is compatible with $\theta_{candidate}$
- STEP-4: ambiguity solution
  - Can use estimate of $\theta \approx \text{atan}(t/(p*N_{pixel}))$ as an additional parameter in the track fit

Seed Finding Based On Cluster Size: Reduction Of Fake Tracks

- **Default pattern recognition:** large fraction of the reconstructed tracks in the very forward region (|η|>3) are fakes
- **New pattern recognition:** large reduction in the number of fake tracks in the forward region with minimal impact on tracks from hard scattering and pile-up interactions (see previous slide)
  - **Preliminary results; optimization is still in progress; performance depends on layout**

Performance in ttbar events with 200 pile-up collisions at √s=14 TeV





Tracking for Triggering Purposes in ATLAS

John Baines on behalf of the ATLAS Collaboration

Trigger Requirements, Challenges & Solutions

Requirements:

- High Efficiency; Low fake rate
- Excellent track parameter resolution

Challenges:

- **Event complexity**: many superimposed collisions
  - 45 (Run 1) to 69 (Run 3) to 200 (HL-LHC)
- **High rate:**
  - 100 kHz Run 2&3 to 400 kHz (1MHz) HL-LHC
- **Short Time:**
  - finite HLT farm size => ~300ms/event for ALL Reco.
  - ~factor 50 faster than offline
- **Huge number of hit combinations**

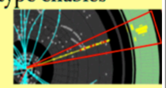
for current luminosities (~30 interactions):

10^5 Space-points → 10^9 Triplets → 10^4 Seeds → 10^3 Tracks

Space-point = Pixel cluster or SCT cluster-pair (ϕ+stereo)

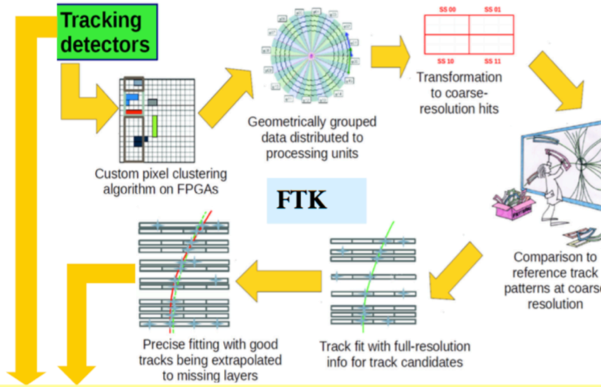
Solutions for the Trigger:

- **Reconstruction in Regions of Interest**
  - Reduced detector volume reconstructed
  - Knowledge of L1 trigger type enables optimised reconstruction



- **Two stage tracking:**
  - **Fast Tracking:** Initial loose trigger selection using reduced resolution tracks
  - **Precision Tracking:** full precision tracks for final trigger selection
- **Limit hit combinations:**
  - Geometrical constraints using RoI, beamspot and possibly primary vertex info
- **Hardware Trigger: FTK (Run 2 & 3)**
  - Pattern matching using custom ASIC
- **Acceleration (Future)**
  - Exploring use of GPGPUs

Fast TracKer FTK



- FTK :
  - **Gets SCT & Pixel data** direct from detector front-end
  - **Performs pattern matching & track fitting** at up to full Level 1 rate (100 kHz)
  - **Sends tracks to the HLT**: perigee parameters and hits-on-track information (cluster centres)
- HLT has access to:
  - FTK Tracks in whole event
  - HLT Refit of FTK tracks
  - HLT tracks in RoI

HLT

FTK Tracks

- Primary vertex reconstruction
- Initial track-based selections in whole event or large RoI incl.
  - hadronic Tau-lepton decays
  - B decays including hadrons
  - Lepton isolation; Track-based jets

HLT Refit of FTK tracks in RoI

- Selections requiring higher parameter precision incl.
- Primary & secondary vertex for b-jet tagging
- Impact parameter and invariant mass for B-physics

HLT tracking in RoI

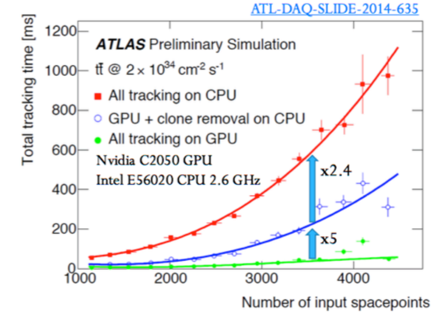
- Single leptons: e, μ, leptonic τ decays
- Final selections requiring ultimate efficiency & precision

GPU Performance & Outlook

ATL-DAQ-SLIDE-2014-635

Code Speed-up:

- Average factor 12 speedup for Tracking on GPU c.f. 1 CPU core



System Performance:

- Question: What increase in throughput comparing CPU system with CPU+GPU
- Measurements in progress:
  - Updated hardware: K80 GPU, Intel E5-2695V3 CPU
  - Updated software: Run-2 Tracking algorithm
- Initial measurements with Seed-Maker suggest factor of two increase in system throughput could be obtained by adding GPU.
- Work in progress to add GPU track following
- Prototype also includes Calorimeter & Muon reconstruction