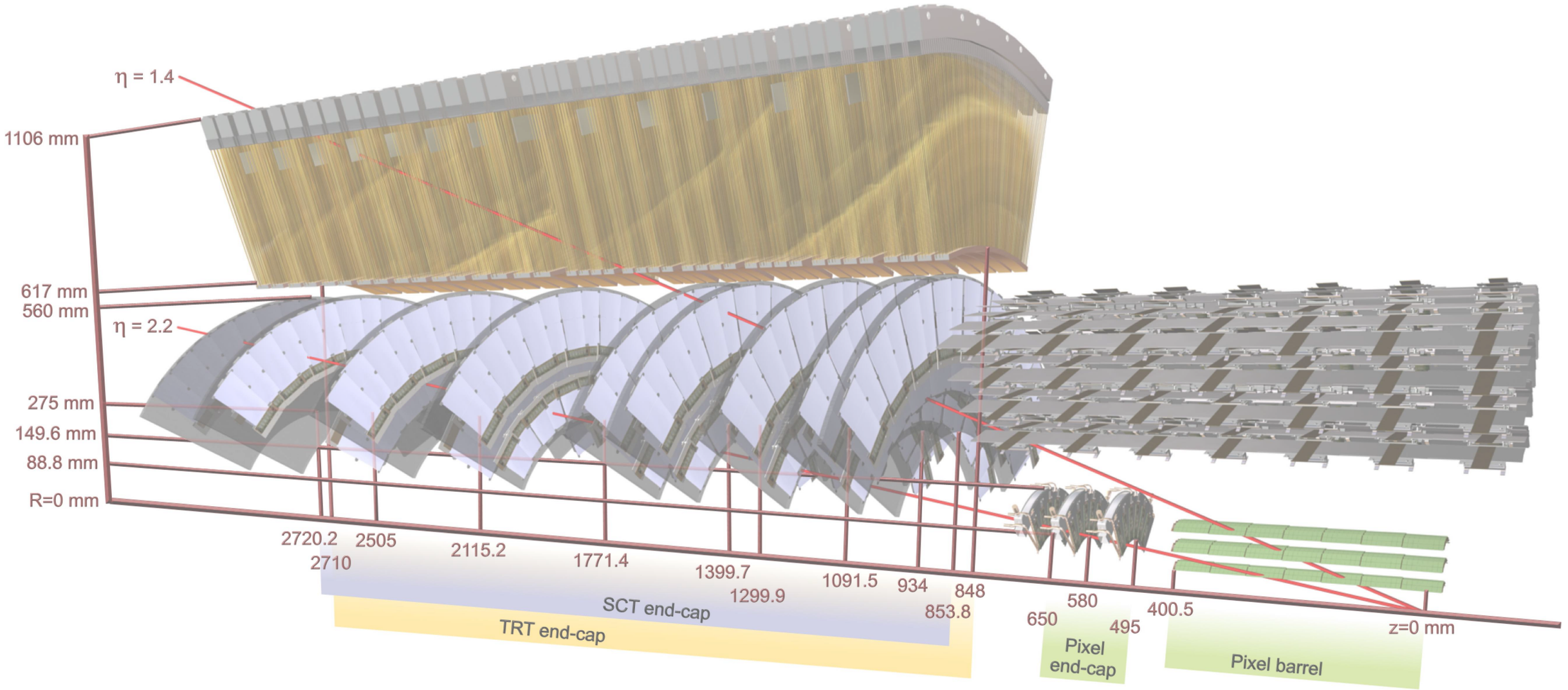
21st International Conference on Computing in High Energy and Nuclear Physics **CHEP2015** Okinawa Japan: April 13 - 17, 2015

Optimisation of the ATLAS track reconstruction software for Run-2

A. Salzburger, CERN

ATLAS Inner Detector (ID) track reconstruction

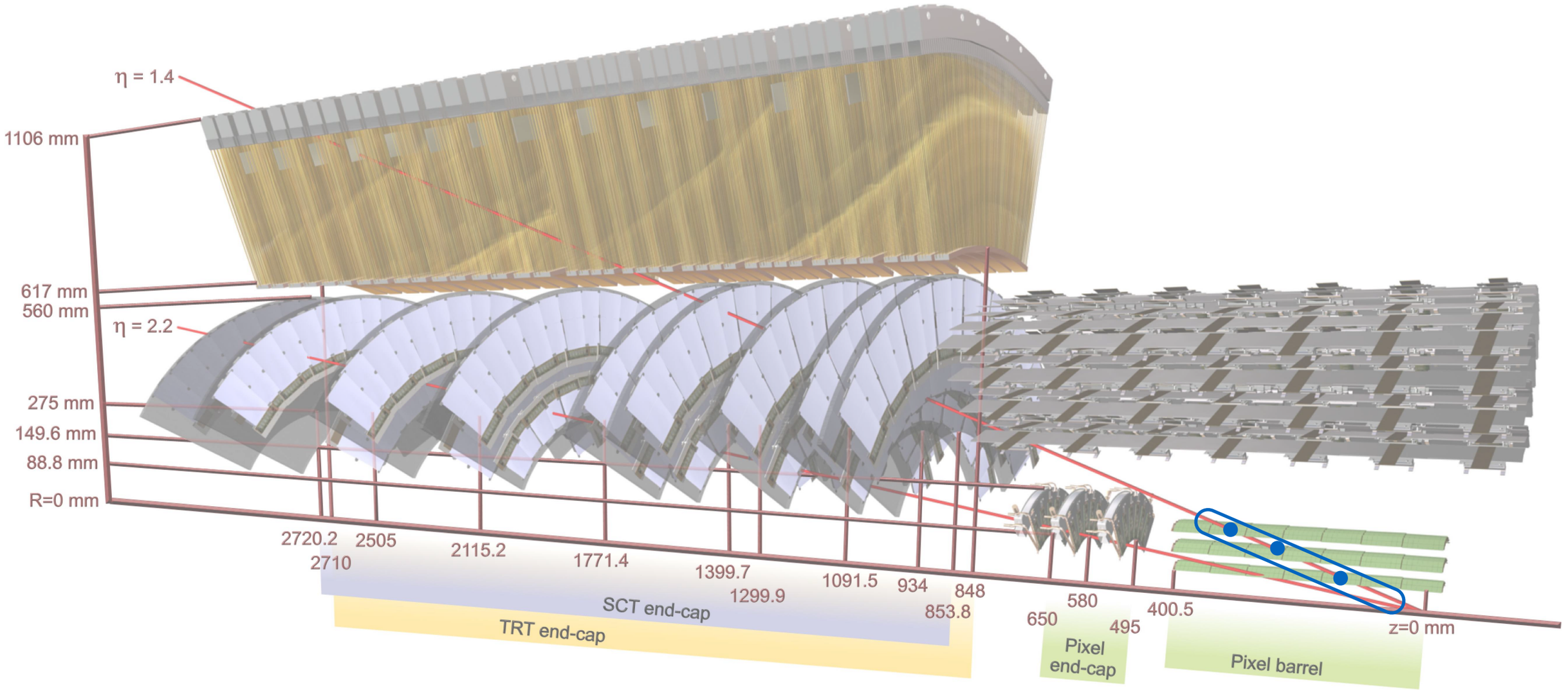
- Track reconstruction is the most challenging step in event reconstruction
  - classical pattern recognition problem in steps



ATLAS Inner Detector (ID) track reconstruction

- Track reconstruction is the most challenging step in event reconstruction
  - classical pattern recognition problem in steps

    (1) track seeding using groups of 3D space points

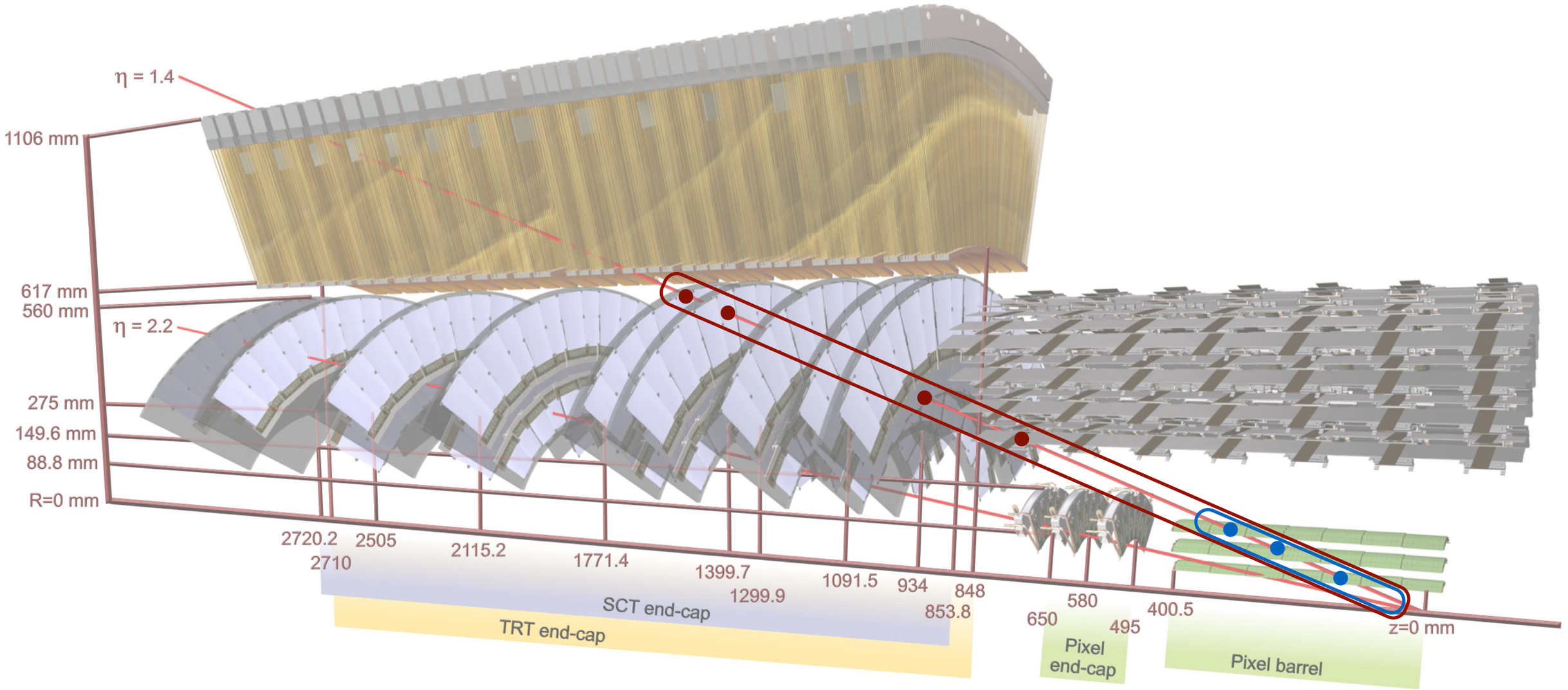


ATLAS Inner Detector (ID) track reconstruction

- Track reconstruction is the most challenging step in event reconstruction
  - classical pattern recognition problem in steps

    (1) track seeding using groups of 3D space points

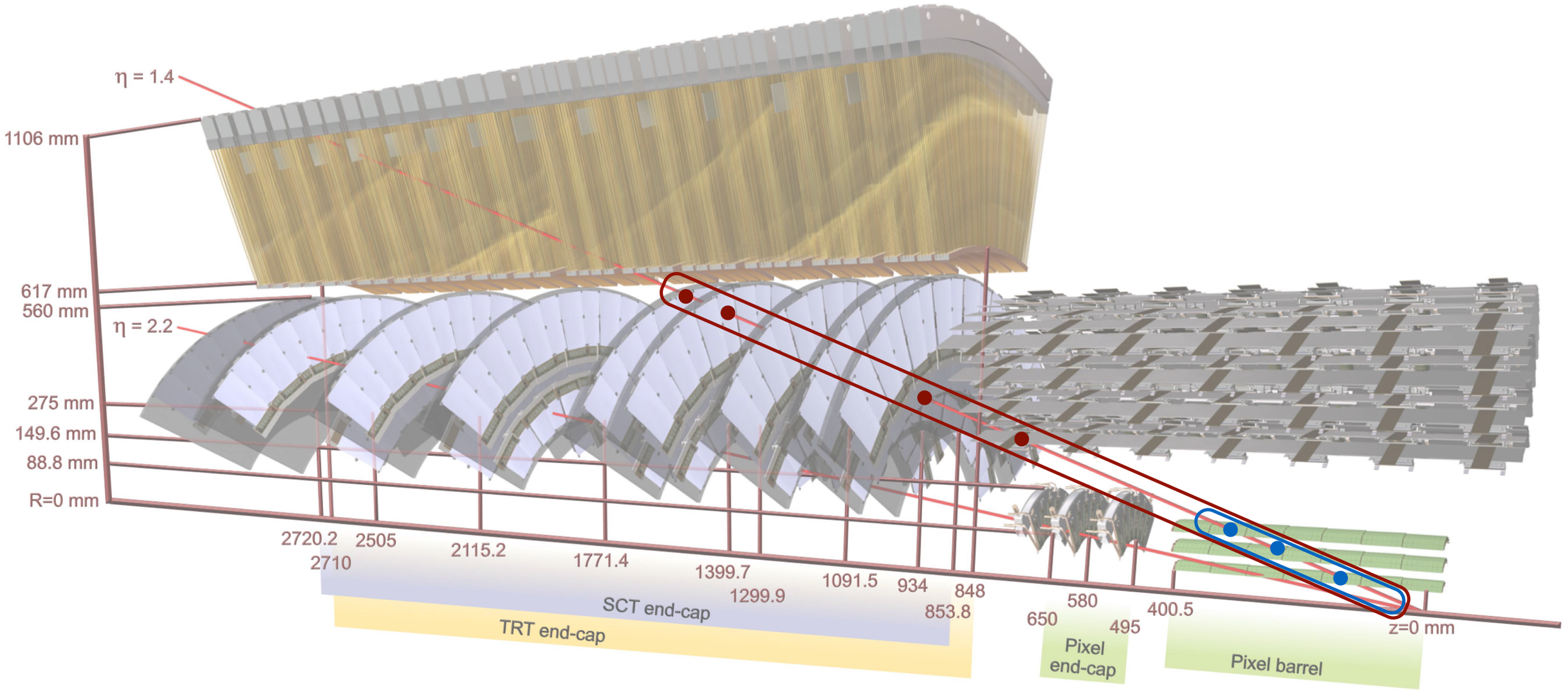
    (2) track candidate building using a combinatorial filter



ATLAS Inner Detector (ID) track reconstruction

- Track reconstruction is the most challenging step in event reconstruction
  - classical pattern recognition problem in steps

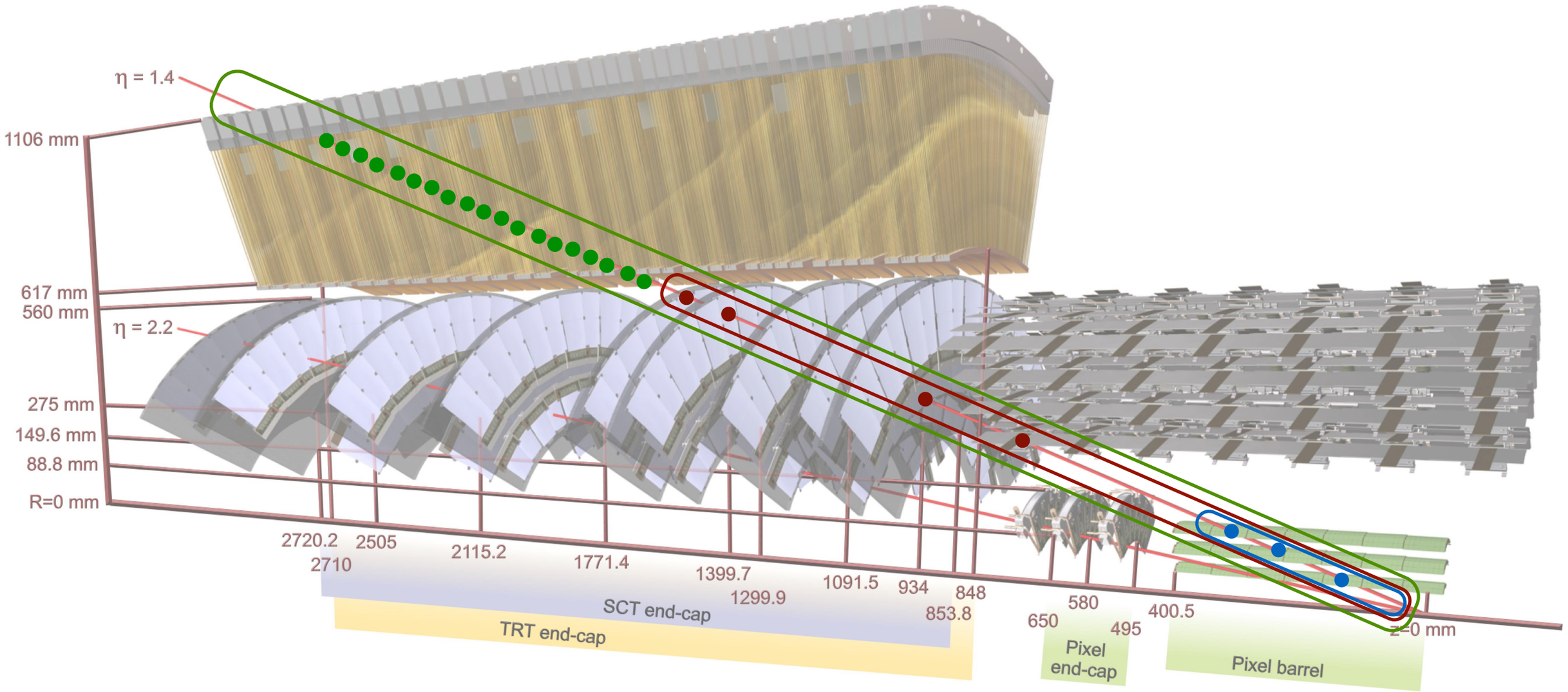
(1) track seeding using groups of 3D space points
(2) track candidate building using a combinatorial filter
(3) ambiguity solving of tracks in the silicon tracker



ATLAS Inner Detector (ID) track reconstruction

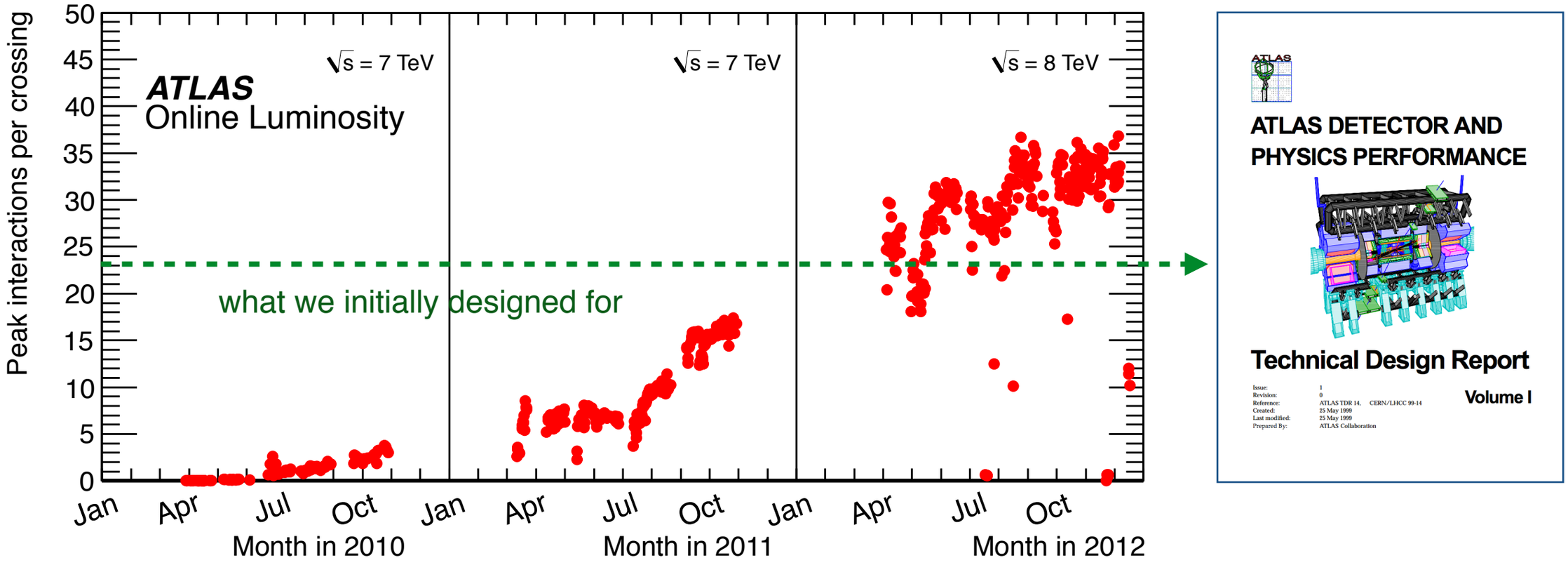
- Track reconstruction is the most challenging step in event reconstruction
  - classical pattern recognition problem in steps

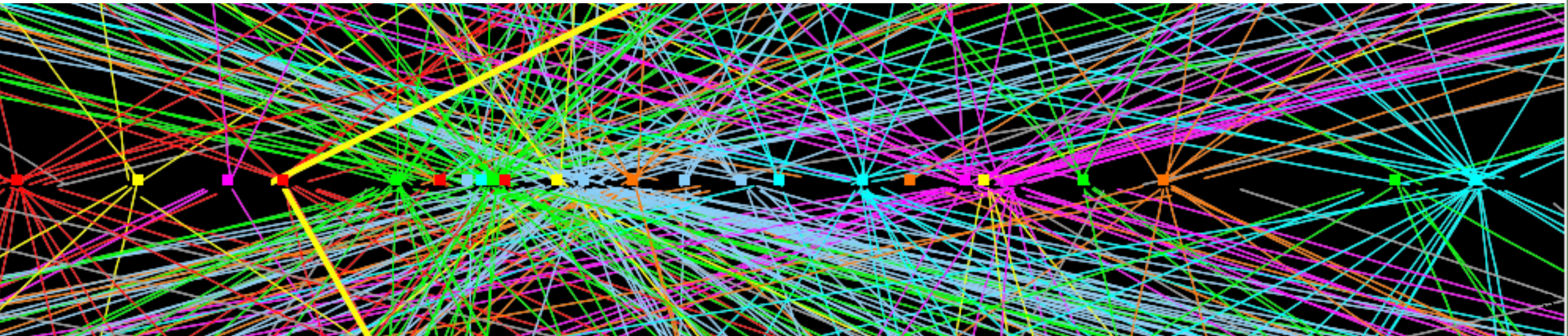
(1) track seeding using groups of 3D space points
(2) track candidate building using a combinatorial filter
(3) ambiguity solving of tracks in the silicon tracker
(4) track extension into transition radiation tracker (TRT)



The Run-1 data taking period

- The LHC was performing outstandingly well - came with a price
  - event pile-up, i.e. instantaneous collisions per bunch crossing



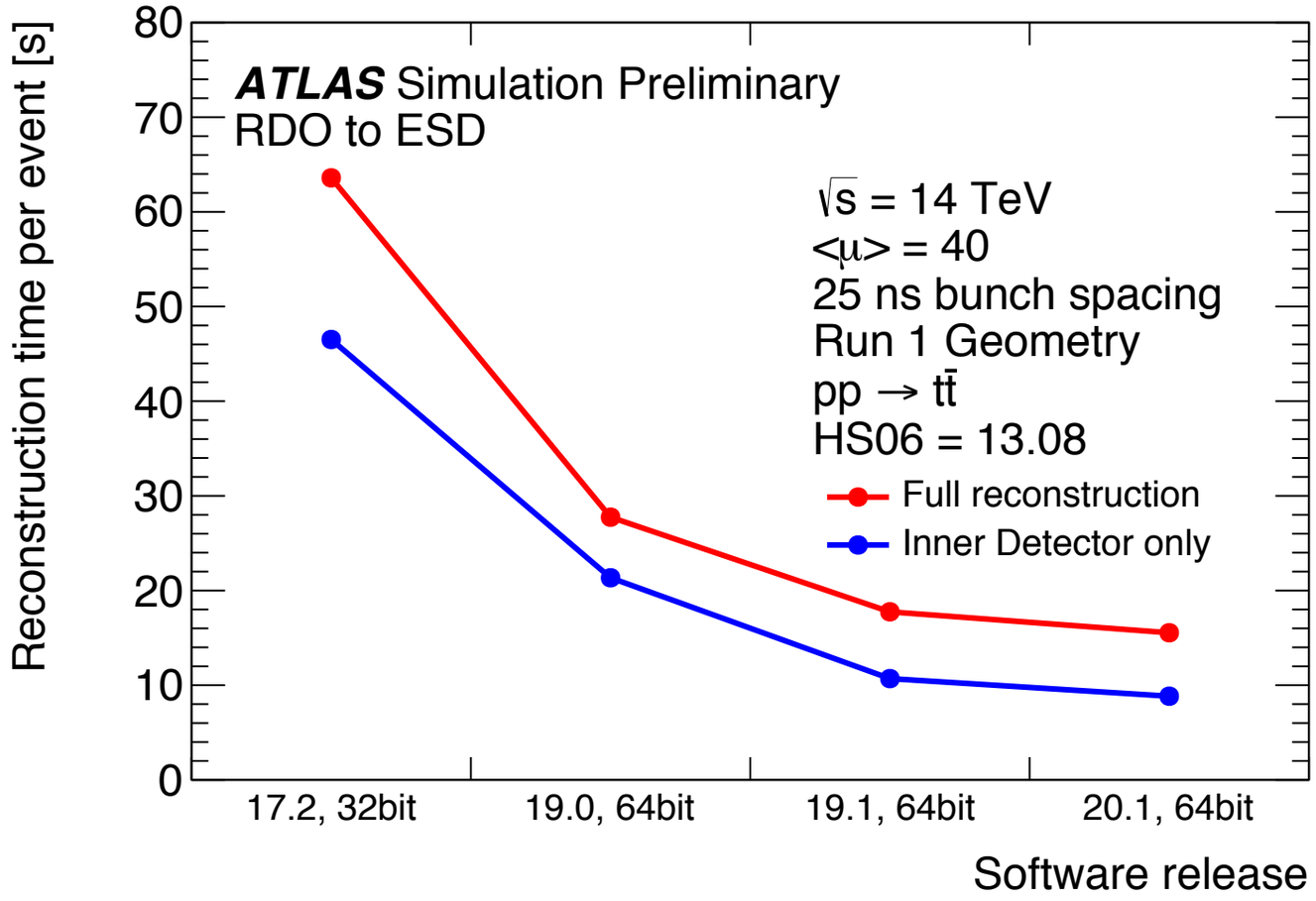


Boundaries and projections - 境界と予測

- Run-2 start up brings excitement of new challenges
  - increase to 13 TeV (from 7 TeV) : more particles per collision
  - increase of HLT rate to 1kHz (from 400 Hz): more collisions to process/time
  - increase of pile-up to <μ>~40 (from ~20) : more collisions per bunch crossing

- Funding profile is likely to stay flat at best
  - extrapolation: 3 speed-up for event reconstruction needed
  - ID track reconstruction is the dominant part

- **Achieved a factor 4**
  - reduced relative fraction of ID to total reconstruction time



Changing the algebra and math libraries

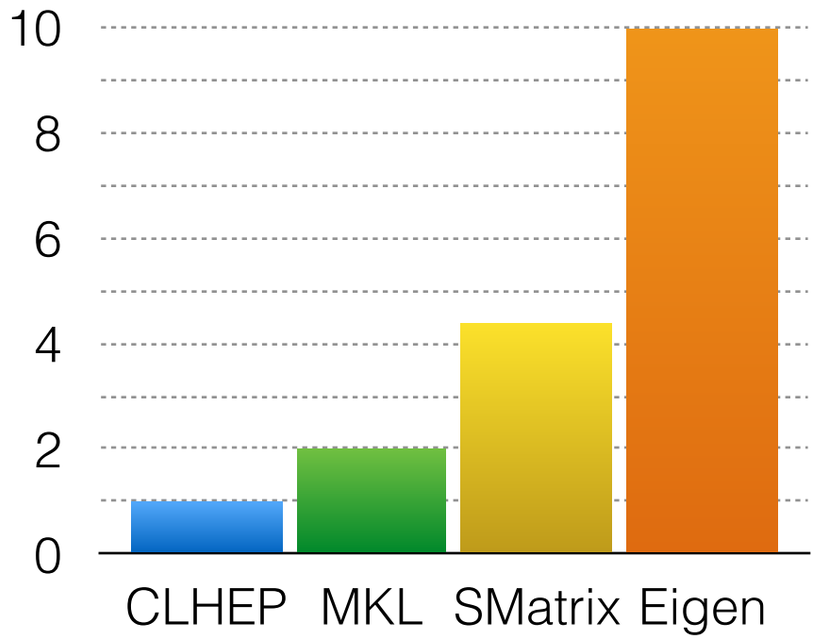
- Event Data Model (EDM) and algorithmic code was based on CLHEP

- Track reconstruction makes heavy use of matrix manipulations
  - usually N x M (with N,M in [1,5]) and inversions

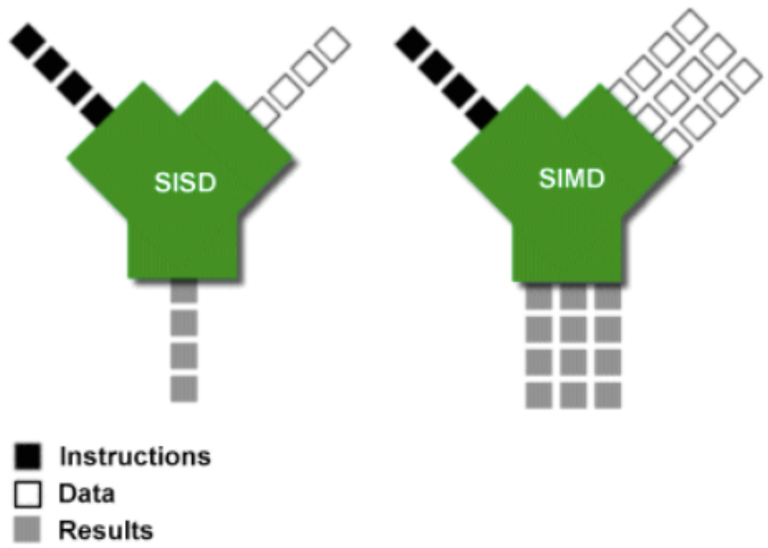
- Identified that CLHEP was one of the bottlenecks in our software
  - simple testbeds implemented to mimic Kalman filterting or Jacobian transport
  - Eigen algebra library chosen (supports SIMD instructions)

- Massive reworking of entire ATLAS code
  - more than 1000 packages changed
  - Eigen/ATLAS interface via typedefs and plugins

  http://eigen.tuxfamily.org/



Achieved speed-up w.r.t. CLHEP in 5x5 matrix multiplication testbed



Cleaning up the Event Data Model (EDM)

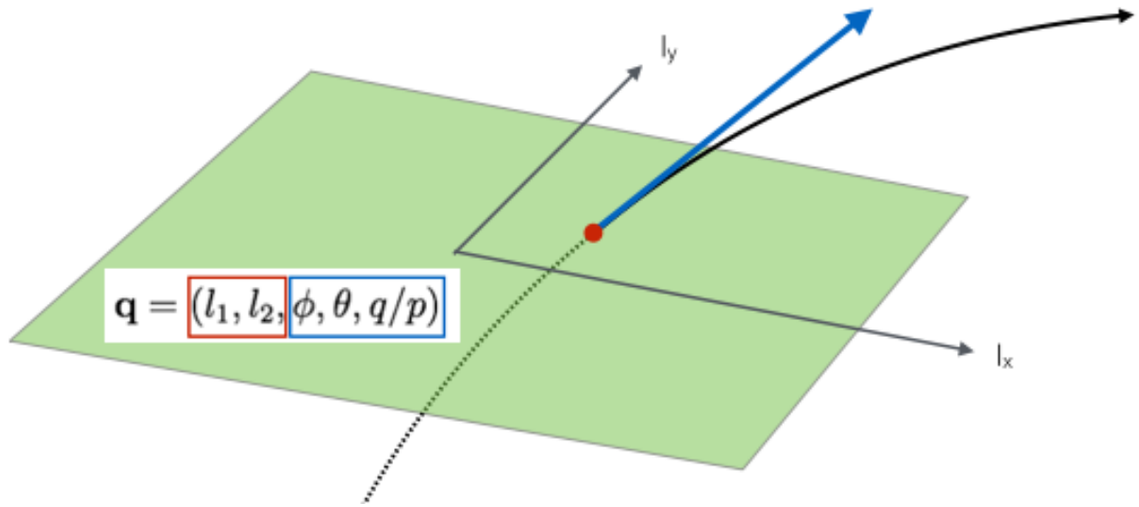
- Flattening the structures of the track reconstruction EDM

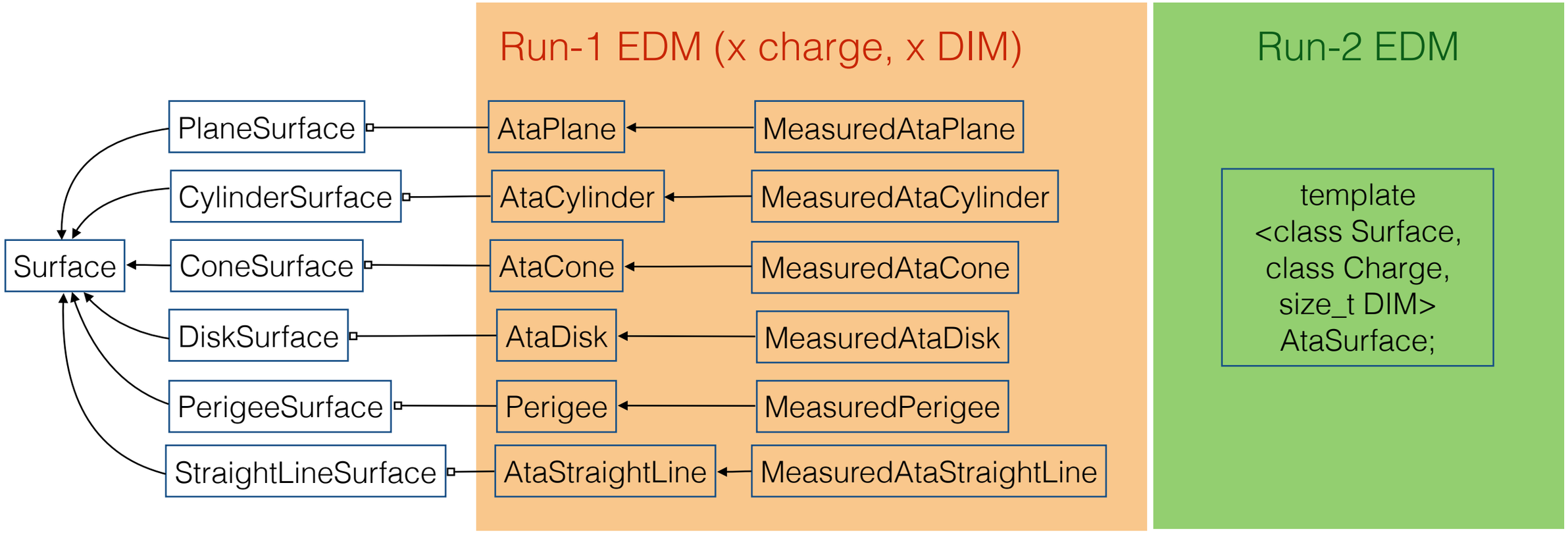
  trajectory needs to be expressed on different surfaces of the detector

  exist as charged / neutral representation

  may exist with covariance or without

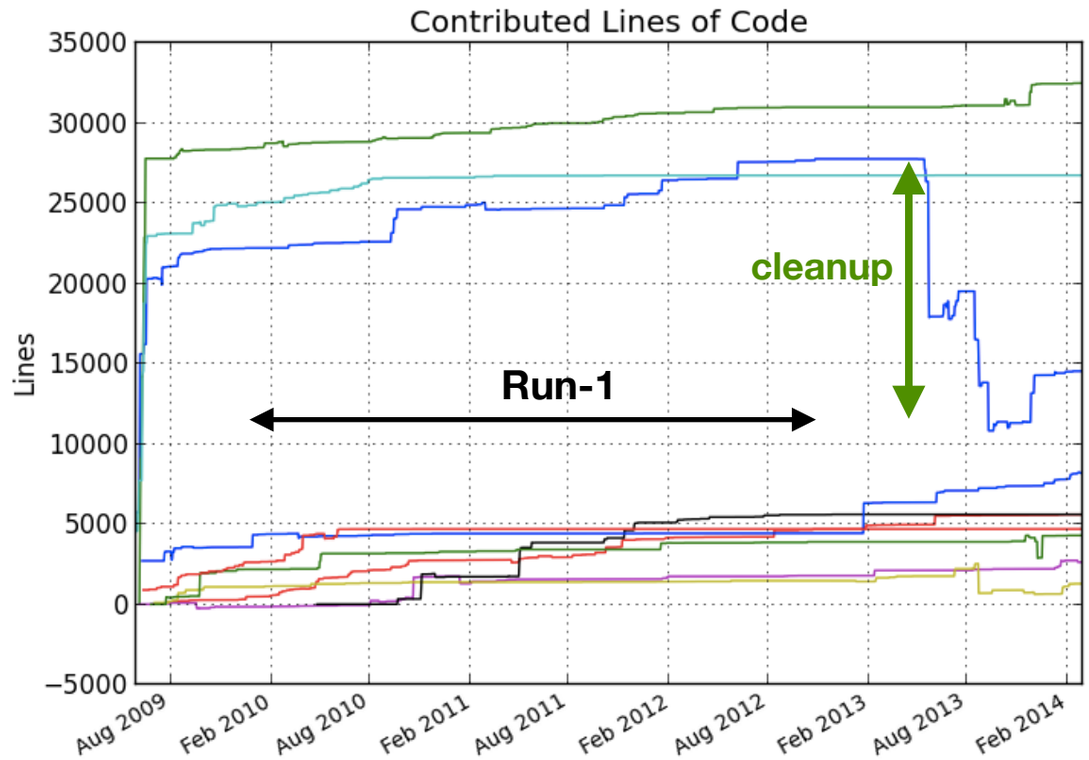
  may be 5-dimensional or 6-dimensional representation (when adding mass)





- Enormous reduction of code lines in tracking EDM
  - while even extending the functionality

Package	C++	C/C++ Header	C++	C/C++ Header
TrkParameterBase	63	561	11	214
TrkParameters	1715	602	0	52
TrkNeutralParameters	1425	663	0	48
ExtendedTrkParameterBase	0	295	0	0
ExtendedTrkParameters	1412	514	0	0
ExtendedTrkNeutralParameters	1416	514	0	0
Total	**6031**	**3149**	**11**	**266**
		Run-1		Run-2



  - nice consequence:

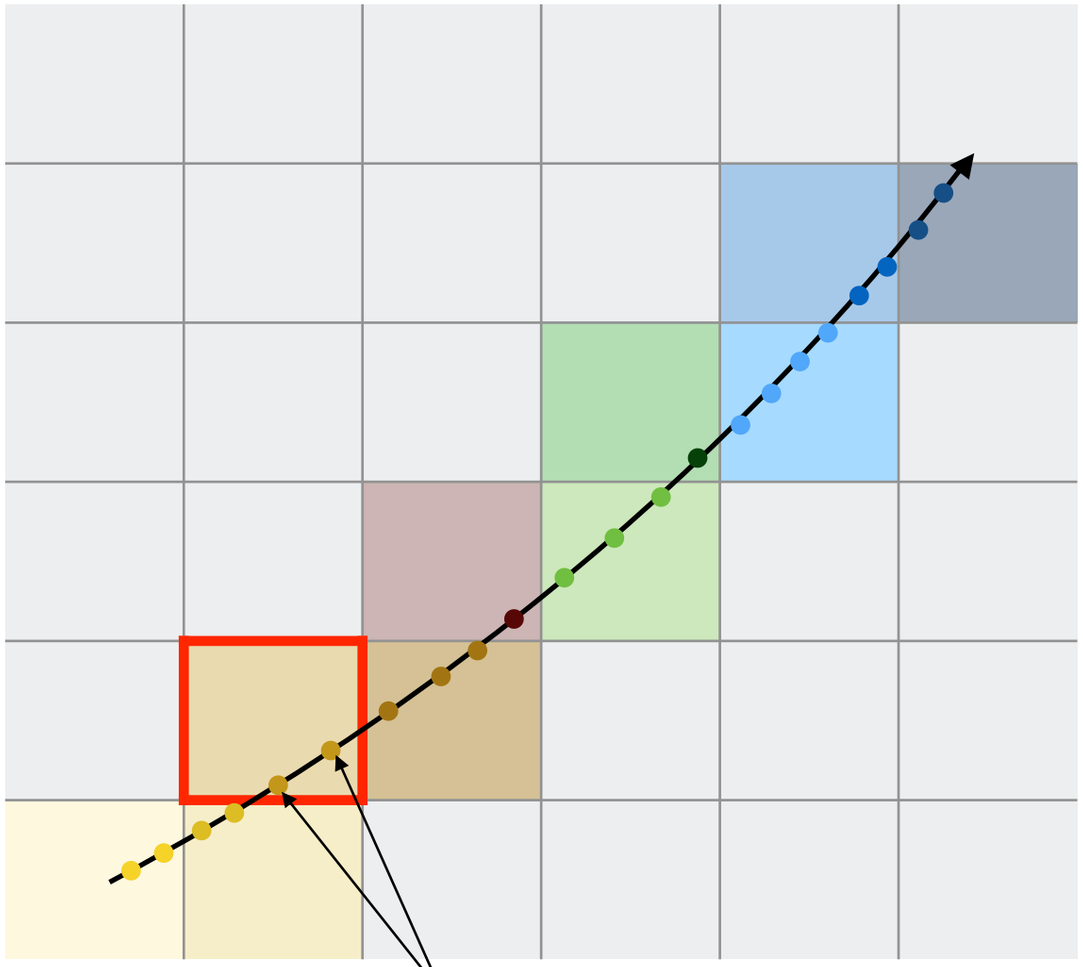
    simplification of object persistency service (only one converter needed)

- Additional campaigns
  - removed lazy initialisation and dynamic memory allocation where possible (led to memory fragmentation)
  - implemented type identifiers to avoid dynamic_cast testing

Optimising the software - ソフトウェアを最適化

- Example: magnetic field access
  - numerical (Runge-Kutta) field integration is one of the big CPU consumers
  - ATLAS adaptive Runge-Kutta propagator has been highly optimised
    dedicated version was back-ported into Geant4
  - field access was not yet optimised
    deep caller chain
    field data needed conversion
    was written in FORTRAN90
  - new field service implemented
    simplified caller chain
    use native units
    use cell caching to store value of field
    -> minimised cache misses
    *speed-up of **20%** in simulation,*
    ***few %** in reconstruction*

Magnetic field map in memory as 3D grid



Field look up in Runge-Kutta integration

Centralise tasks - タスクを一元化

- Track to calorimeter cluster association is a frequent process in event reconstruction:
  - clients throughout the combined reconstruction, e.g.:

    particle flow, photon reconstruction,

    muon/tau/electron reconstruction,

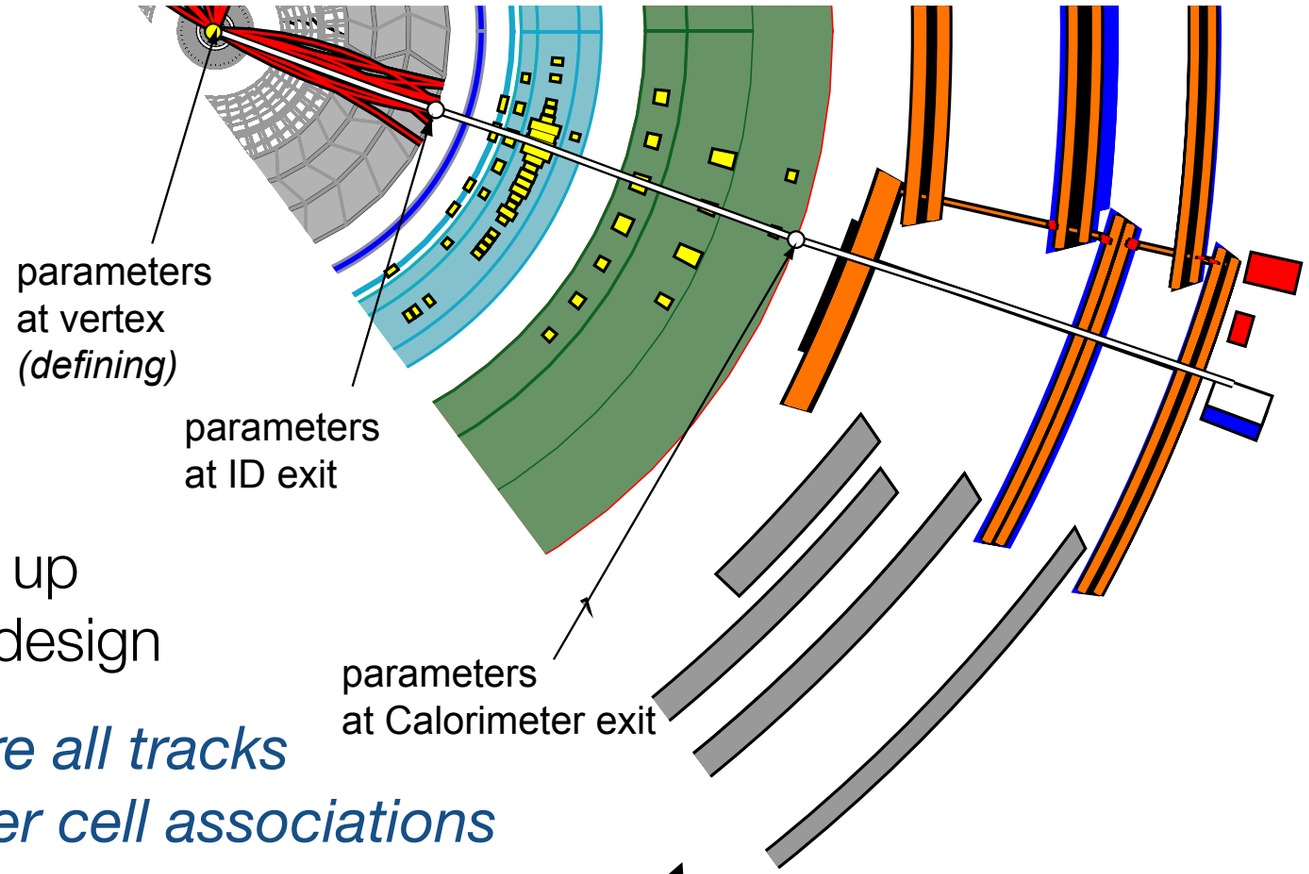
    missing track Et, etc.

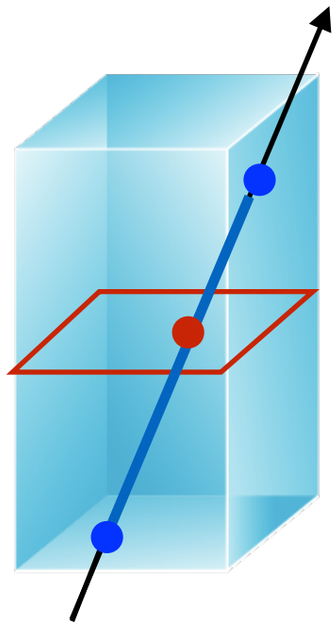
  - analysis showed that this was done up to six times per track in our factory design

    switched to a service design where all tracks are "dressed" with their calorimeter cell associations

    time saving from multiple calls allows free'd CPU cycles to be invested into a more precise job

  - neatly tie in with the new ATLAS analysis event data format (xAOD)





Track prediction through an example calorimeter cell

Run-1/2: intersection with cell center

Run-2: additionally entry/exit position and path length in cell

Being smarter - 賢く

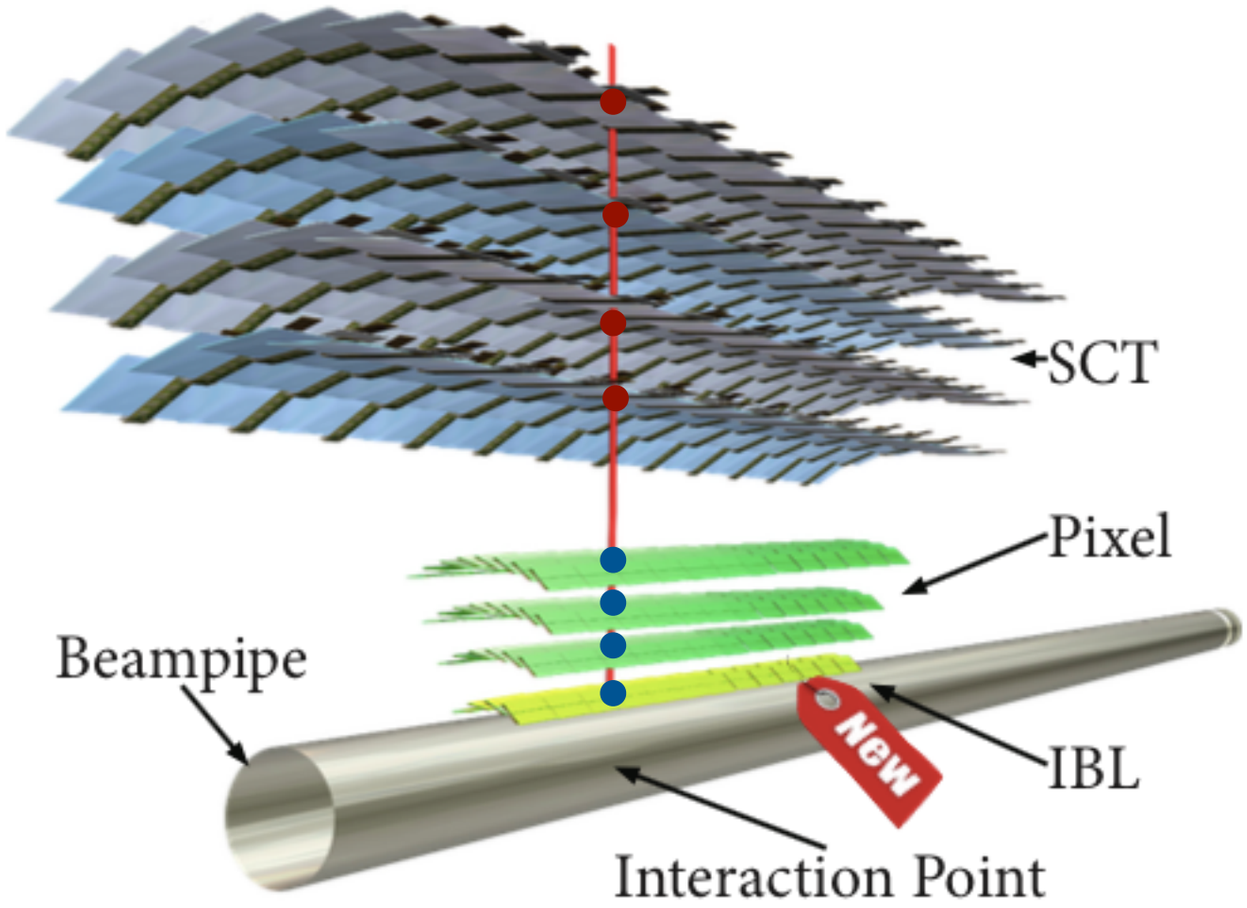
- Track reconstruction software for Run-1 was designed with a lot of redundancy and safety margin

- Run-1 performance convinced us that our system was understood
  - and extremely well modelled by MC

- Re-investigation of track seeding
  - taking high purity seeds from strip
  - make optimal use of new innermost Pixel layer (IBL)

- Greatly improved the seed purity

*managed to be **more efficient***
in less time: ~25 % saving



triple seeds can we built as:
- pixel space points only (PPP | ●●●)
- strip space points only (SSS | ●●●)
- a combination of both, e. g. (PSS | ●●●)

Being smarter - 賢く

- Track reconstruction software for Run-1 was designed with a lot of redundancy and safety margin

- Run-1 performance convinced us that our system was understood
  - and extremely well modelled by MC

- Re-investigation of track seeding
  - taking high purity seeds from strip
  - make optimal use of new innermost Pixel layer (IBL)

- Greatly improved the seed purity

*managed to be **more efficient** in less time: ~25 % saving*

Efficiency of a seed with 3 space points resulting in a successful track candidate

<μ>	PPP	PPS	PSS	SSS
0	57%	26%	29%	66%
40	17%	6%	5%	35%

When requiring confirmation by another space point (Run-2 strategy)

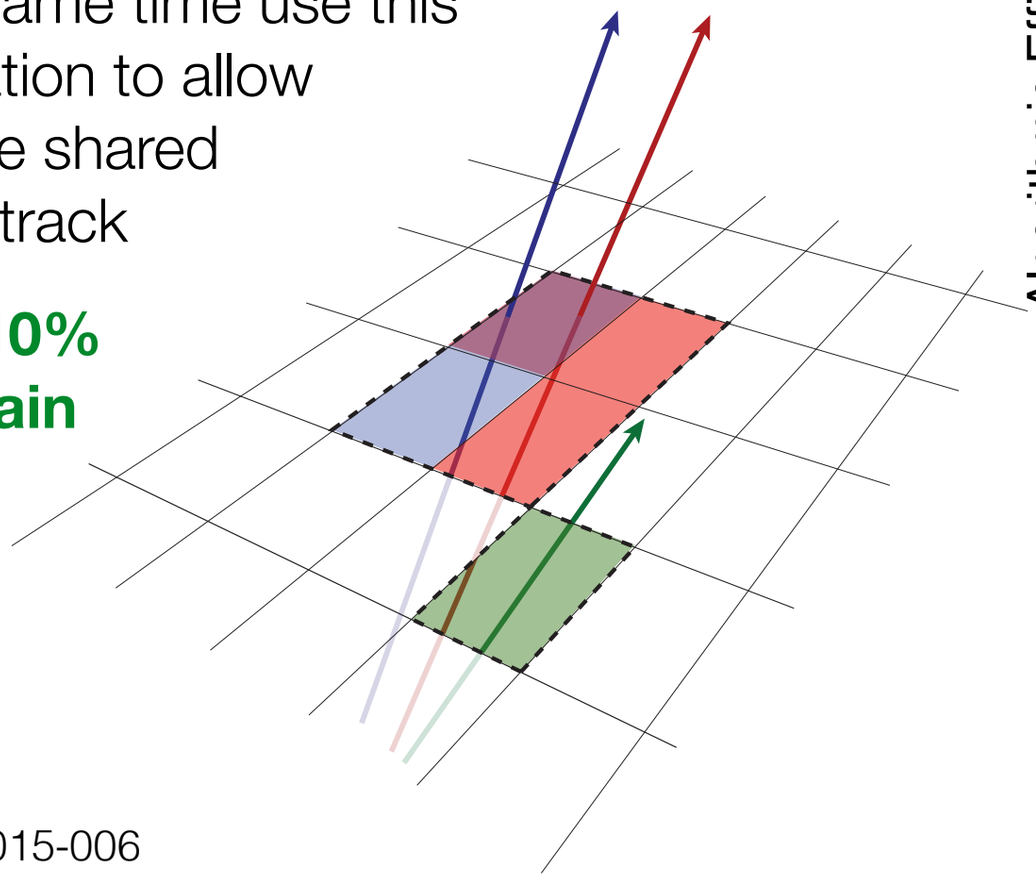
<μ>	PPP+I	PPS+I	PSS+I	SSS+I
0	79%	53%	52%	86%
40	39%	8%	16%	70%
	Step 2			Step 1

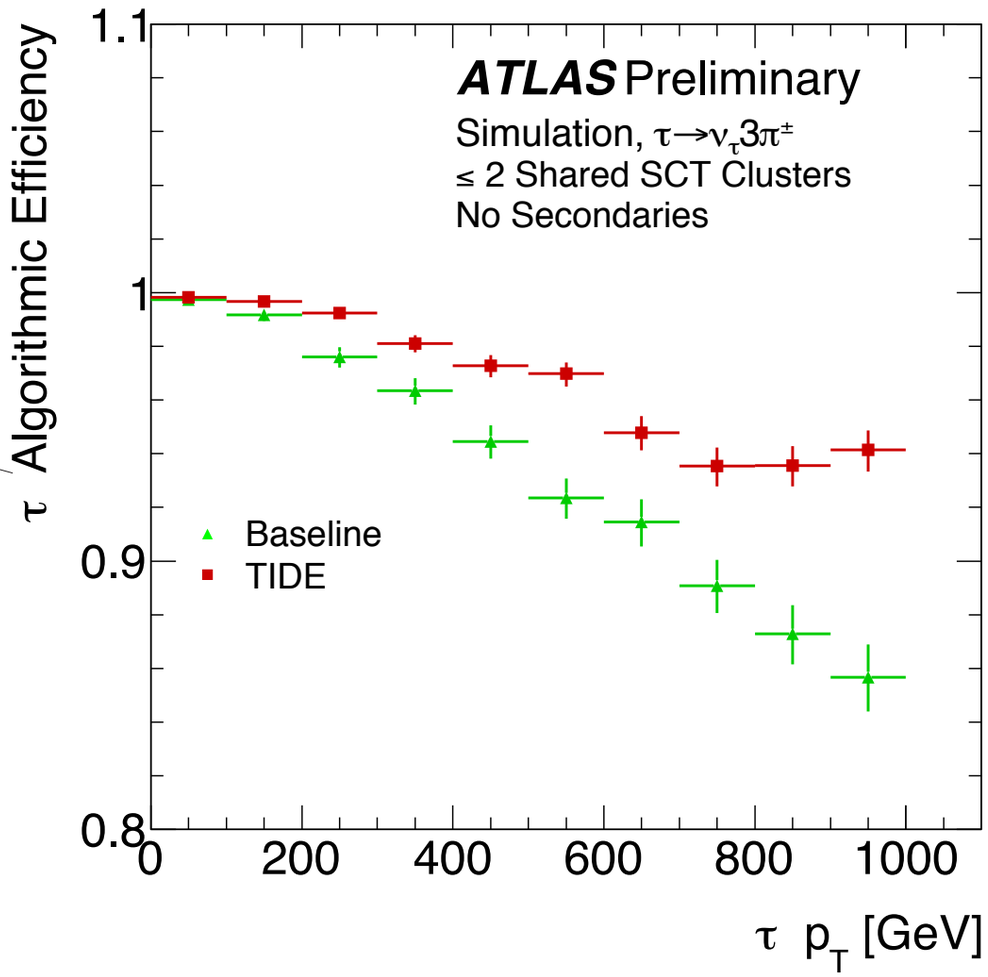
remove space points from Step 1

Doing better - より良いやって

- During Run-1 we developed a neural network based cluster splitting
  - aimed at identifying merged clusters stemming from multiple particles
  - was run as default before the pattern (ran over all clusters as default)

- Second iteration - update of the ambiguity solving method
  - only clusters on track candidates are further tested for splitting (less hits into the pattern)
  - at the same time use this information to allow for more shared hits on track
  - about **10% CPU gain**





ATL-PHYS-PUB-2015-006

Free lunch - フリーランチ

- Some CPU saving came for free by updates
  - new kernel in Scientific Linux 6 gave approximately 10% saving for total event reconstruction
  - switching from 32bit to 64bit architecture

    did increase memory footprint slightly
  - newer compiler version

- some vectorisation benefits that came into place via the Eigen library

  nota bene: track reconstruction is mainly operating on

  local coordinate systems: DIM 1,2

  global coordinate systems: DIM 3

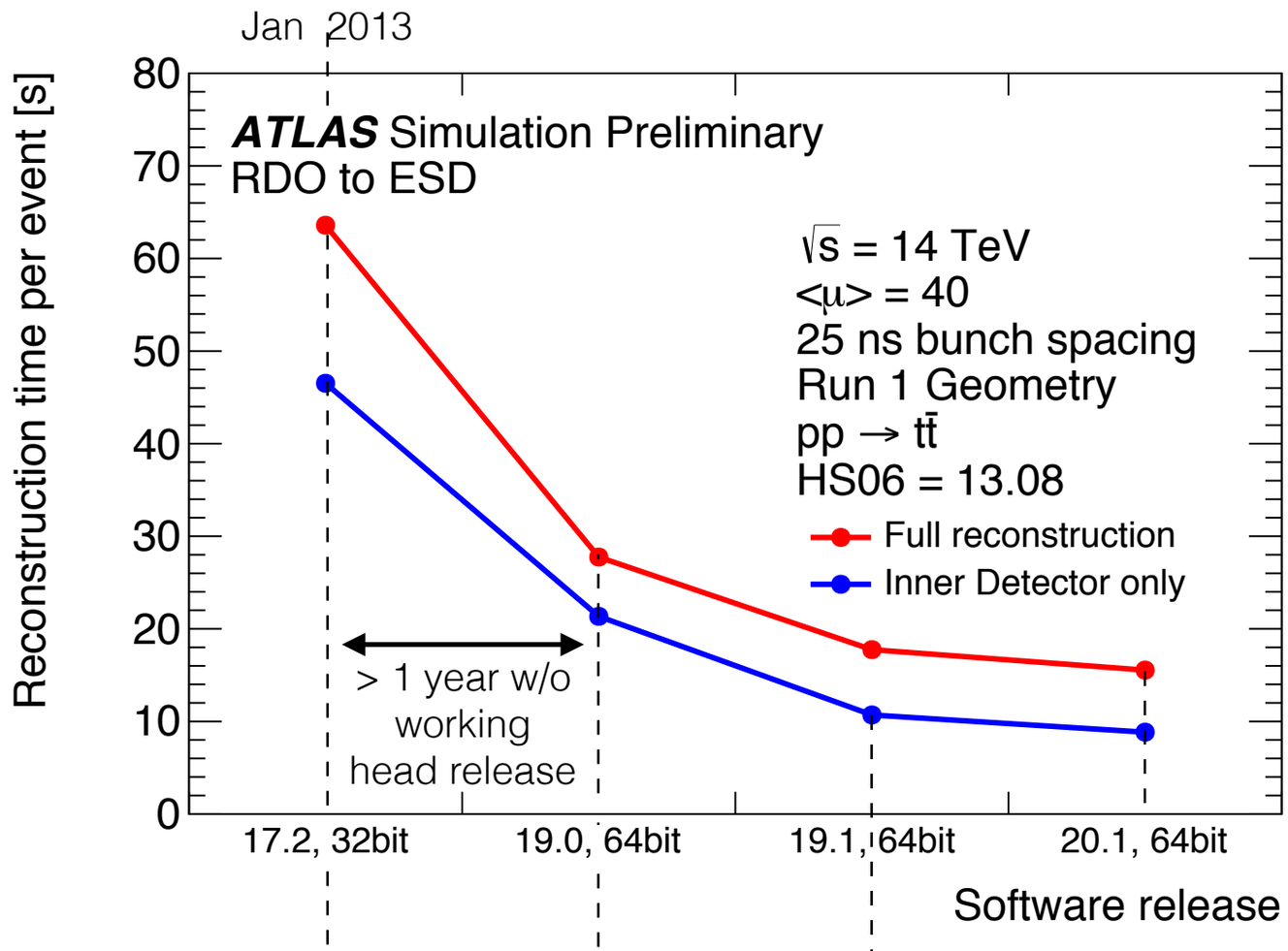
  full track representation: DIM 5

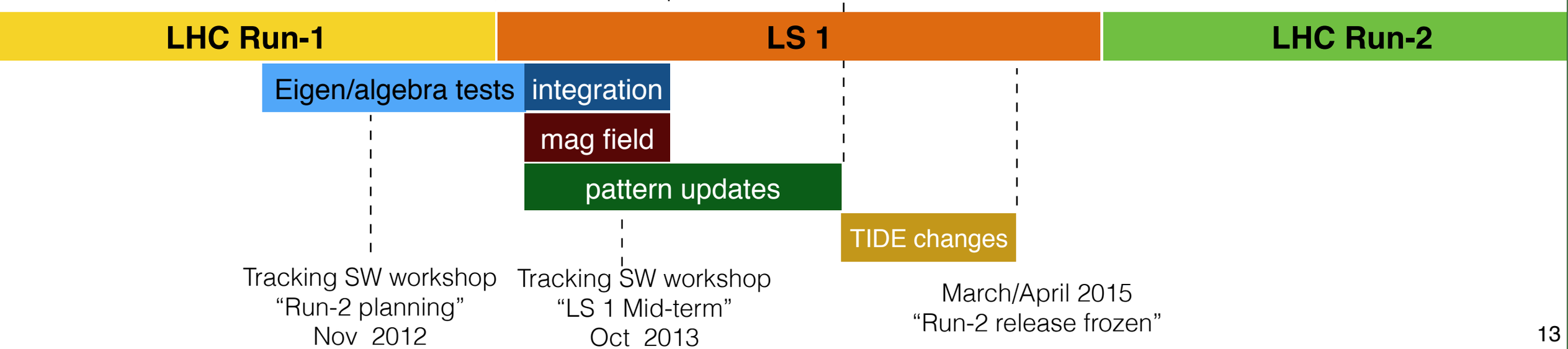
  this is not optimal for a DIM 4 based vector register
  [are currently revisiting a potential DIM 4 based Runge-Kutta method]

- switch to Intel math library (pre-loaded as a plug-in)

The factor 4 - 因子4

Planning & Deployment





Today and Tomorrow - 今日と明日

- LS1 gave a unique opportunity to clean up the ATLAS track reconstruction software
  - massive campaign with a rework of almost the entire repository

    mixture of technology improvements, algorithmic improvements and simply code cleanup
  - disentangling the impact of the different projects is almost impossible, due to time constraint of LS-1 we had to develop and deploy in parallel

- We achieved a factor 4 speedup of the overall event reconstruction time
  - at the same time improving physics performance on all ends
  - mainly achieved by the Inner Detector track reconstruction
  - ready for Run-2 data taking

- Currently in review of the Tracking software for future ATLAS framework
  - anticipate extensive use of concurrency

ご静聴ありがとうございました。