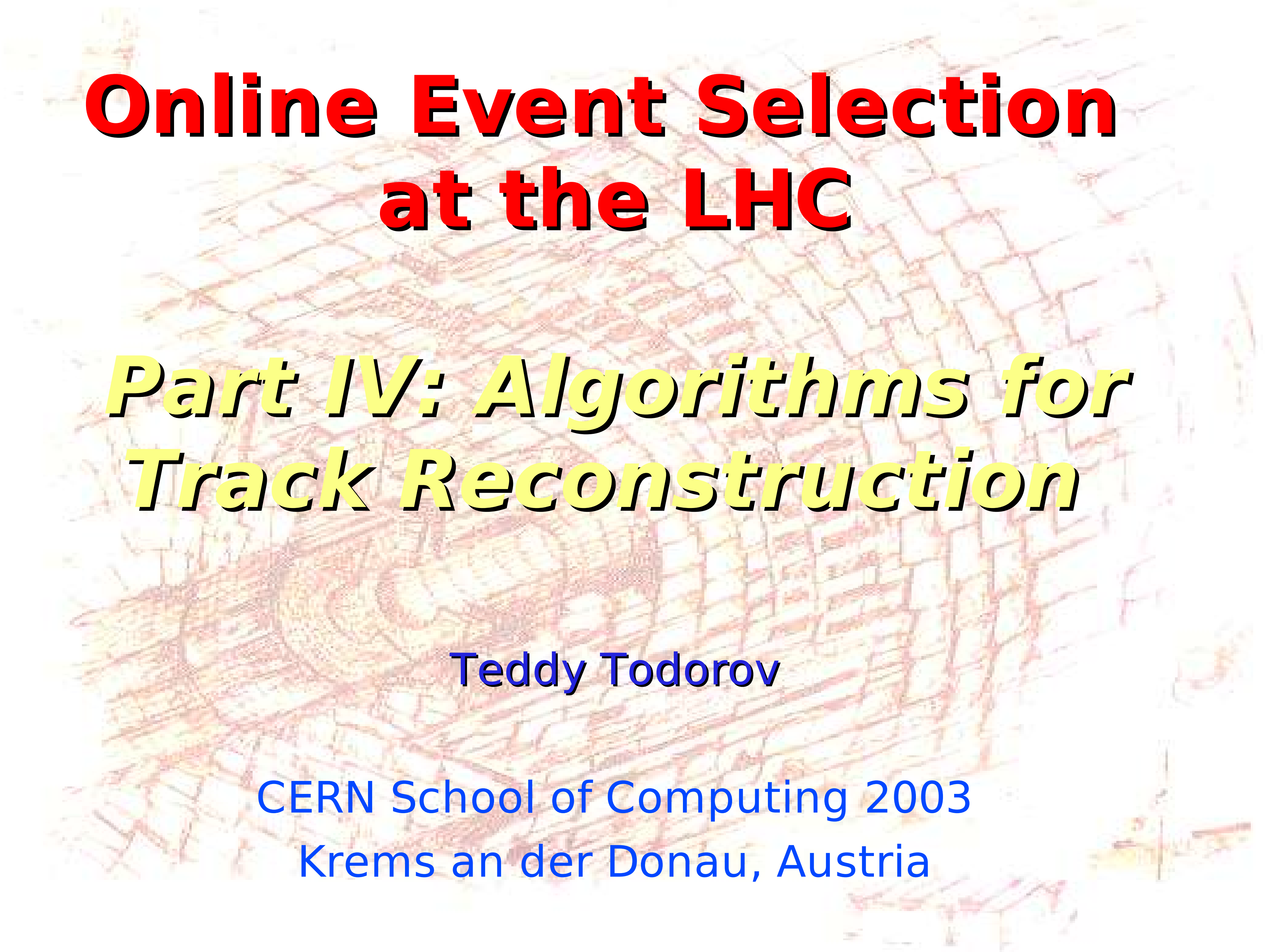


Online Event Selection at the LHC

Part IV: Algorithms for Track Reconstruction

Teddy Todorov

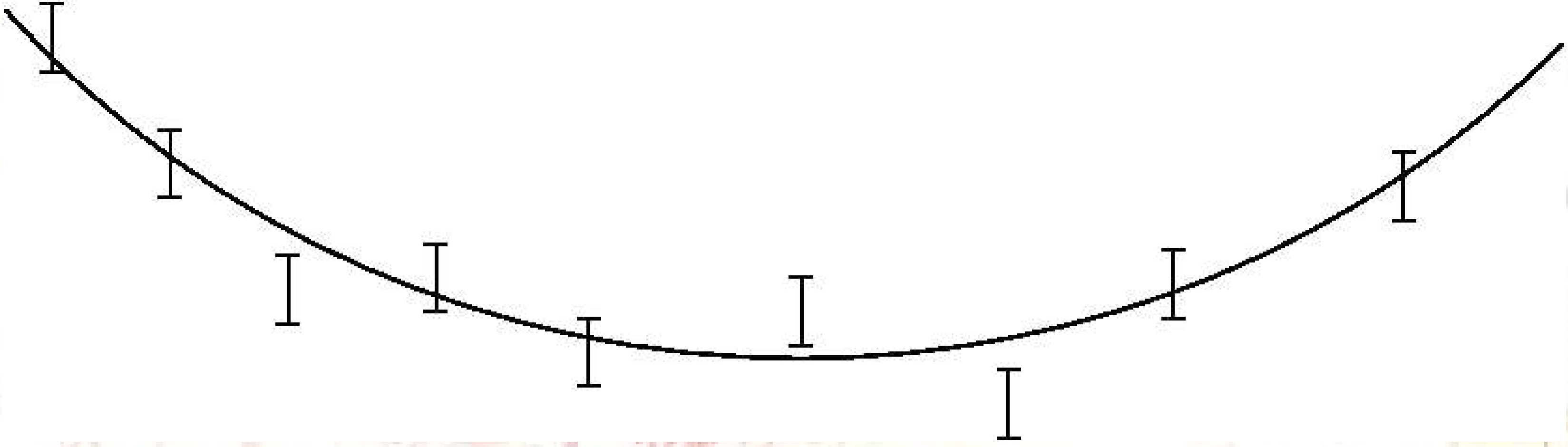
CERN School of Computing 2003
Krems an der Donau, Austria

Track reconstruction

- Track reconstruction covers
  - Track finding, or “pattern recognition”: the attribution of hits to tracks
  - Track fitting, or the determination of the track parameters from a given set of hits

Global track fit

- Standard minimization problem for parametric function
  - We need a parametrization for the track, a “track model”

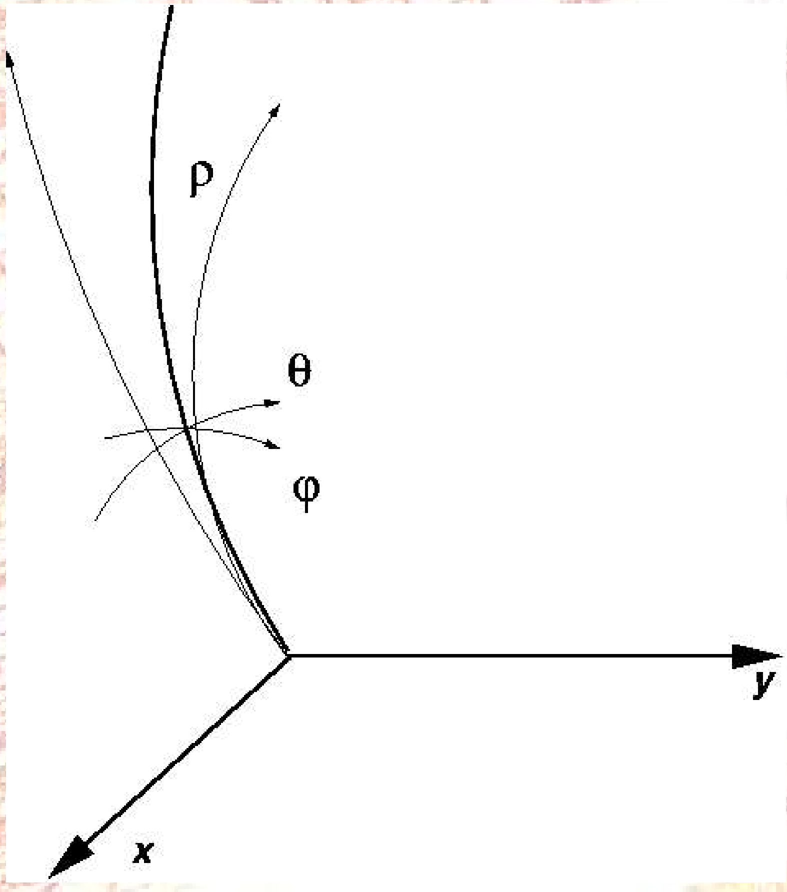


Track model

- In a uniform magnetic field, and in the absence of material, the solution to the equation of motion of a charged particle is a helix.
- Locally magnetic fields “are” uniform, and material effects vanish, so a helix is also a local approximation, or linearization, of the general case

Helix dimension

- A helix is a 5D object:
  - Two positions
  - Two angles
  - Curvature
- The particular choice of parametrisation, while important, is beyond the scope of the lecture



Track state representation

- A track state can be represented as a point in 5D linear space
- Not the whole story: a track is a measured (fitted) object, and has uncertainties (errors) on it's parameters
- A track state is fully described by 5 parameters and a 5x5 symmetric error matrix
  - Simply called “track state” or “trajectory state” from now on
  - This is what a global track fit gives as a result

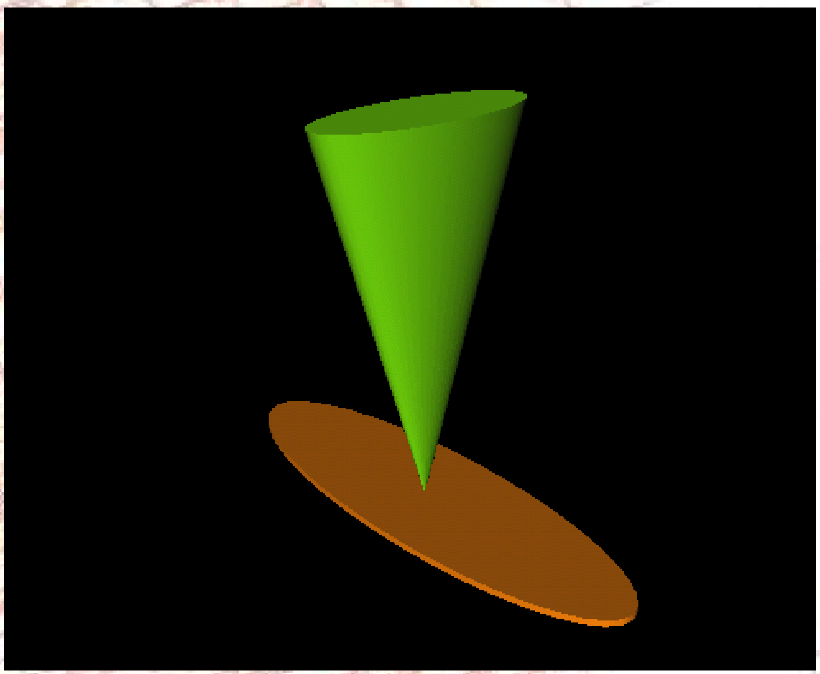
Track state propagation

- A track state can be "propagated" from one place to another, e.g. From one measurement surface to the next.
- Propagation has a purely geometrical part, which is conceptually straightforward, but technically challenging
  - Propagation of track parameters is computation of crossing point of a helix with a surface (e.g. Plane)
  - Propagation of track errors involves 5D jacobians
- Physics effects, like energy loss and multiple scattering, can be added during propagation

NewState = propagate(SomewhereState, Surface)

Track state visualization

- Is difficult.
- On the right is an attempt to visualize 4 parameters with their uncertainties
  - Not enough: off-diagonal (correlation) terms are essential, but I don't know how to draw them!



Kalman filter

- Since a trajectory state is a local thing, and so is a measurement (hit), is there a way to “update” a track state with a hit locally?
  - Yes! (found in 1984…). The operation is called “Kalman update”

  NewState = update(PredictedState, Hit)

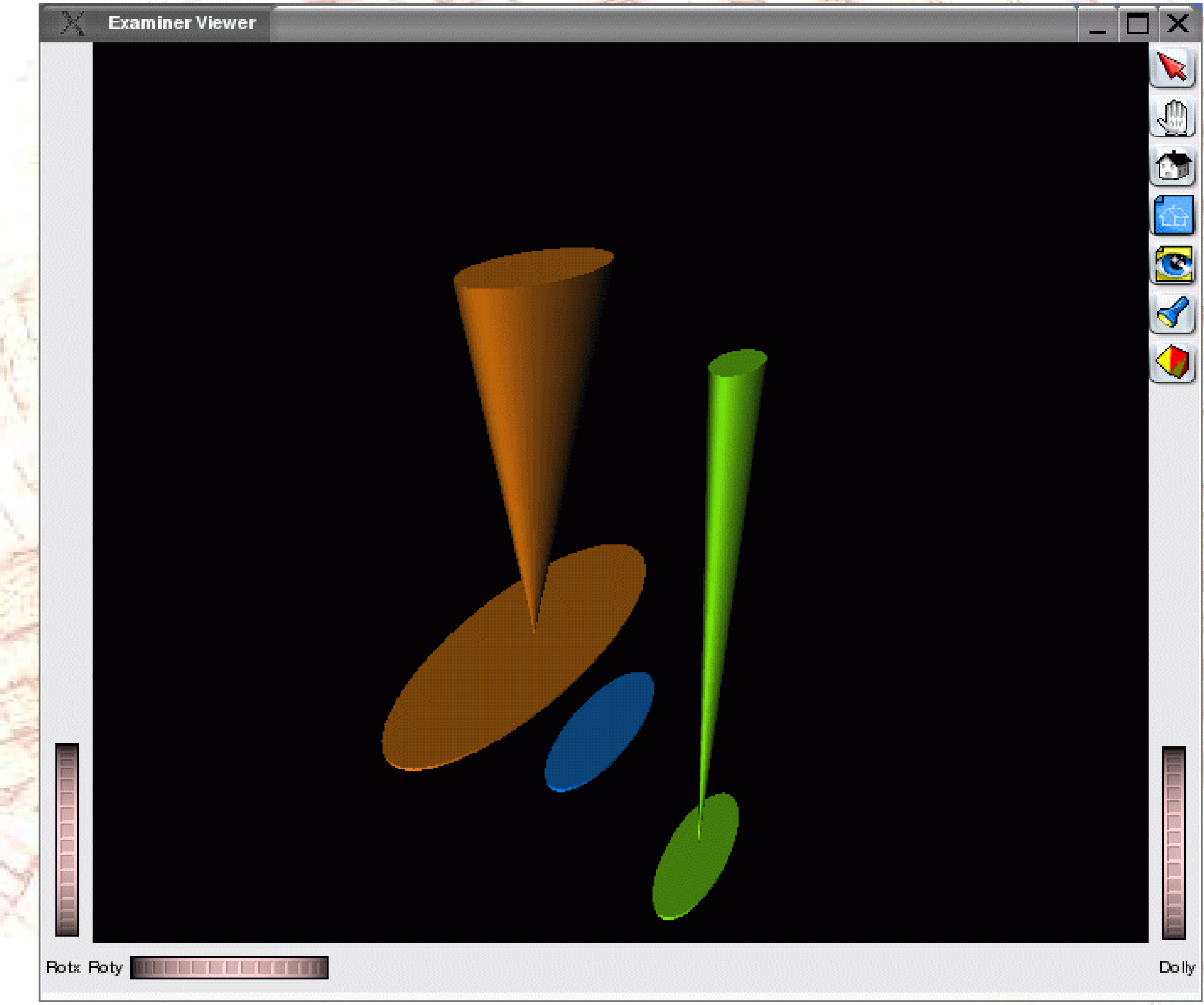
  The PredictedState must be on the same surface as the hit.

  Essentially a weighted mean of the measurement and the projection of the predicted state, but affecting the whole state, not just the projection

  Components not measured in the hit get updated via the correlation terms in the covariance matrix

    these terms appear during propagation

Artistic view of a Kalman update



Kalman track fit

- Given some starting state, the track fit is just a sequence of propagations to measurement surfaces *in the order in which they are crossed by the track*, and updates
- After each update the track is fully fitted with all the hits used so far.
  - Only the last updated state contains the full information. Previous states contain partial information.
- All the hits need not be known in advance, since they are used one at a time
  - This property is at the basis of Kalman filter track finding

Is Kalman filter the last word?

- The Kalman filter is an optimal estimator of track parameters in case of
  - Unbiased measurements with Gaussian errors
  - Gaussian process noise (multiple scattering etc.)
  - No outliers (hits that don't belong to the track)
- For the non-Gaussian generalisation see Are Strandlie's lectures on adaptive algorithms next week.
  - Non-gaussian probability density functions (PDFs) of the hit positions don't hurt too much
  - Non-gaussian noise (energy loss) can degrade the fit seriously (GSF)
  - Ambiguous situation require more advanced outlier treatment (DAF)
- For HLT the Kalman filter is more than sufficient

Kalman track finder

- Given a starting state, hits can be found one at a time!
  - After using (updating with) each hit, the track parameter accuracy improves, and the compatibility window for the next hits gets smaller
- The Kalman filter is a track finder!

Seeding the Kalman filter

- The Kalman filter requires a “starting state”
  - With “infinite” errors, not to bias the fit
  - With parameters close to the fitted ones, to work in the “linear regime”
- Starting the search for compatible hits from “zero knowledge” would be a waist of CPU, since by def. All hits are compatible.
  - A seed should constrain (at least roughly) all 5 parameters

Seed generators

- A track seed can be
  - internal to the tracker (e.g. A pair of hits and a beam spot constraint)
  - External (e.g. From calorimetric cluster)
- For internal seeds, all hits of the tracker need not be used
  - Usually a small number of “seeding layers” is chosen

Choice of seeding layers

An obvious choice would be the outermost layers, since the occupancy is lowest there.

But in Atlas and CMS

- About 10% of the 1 GeV pions interact before crossing 8 layers
- The outer layers don’t have stereo information
- The innermost layers are “pixel”, with very low channel occupancy and excellent 2D resolution
- The region of interest is usually defined at the origin

Therefore the pixel layers are the favored seeding layers

  - Except for reconstruction of secondary interaction results, like electron-positron pairs from photon conversions

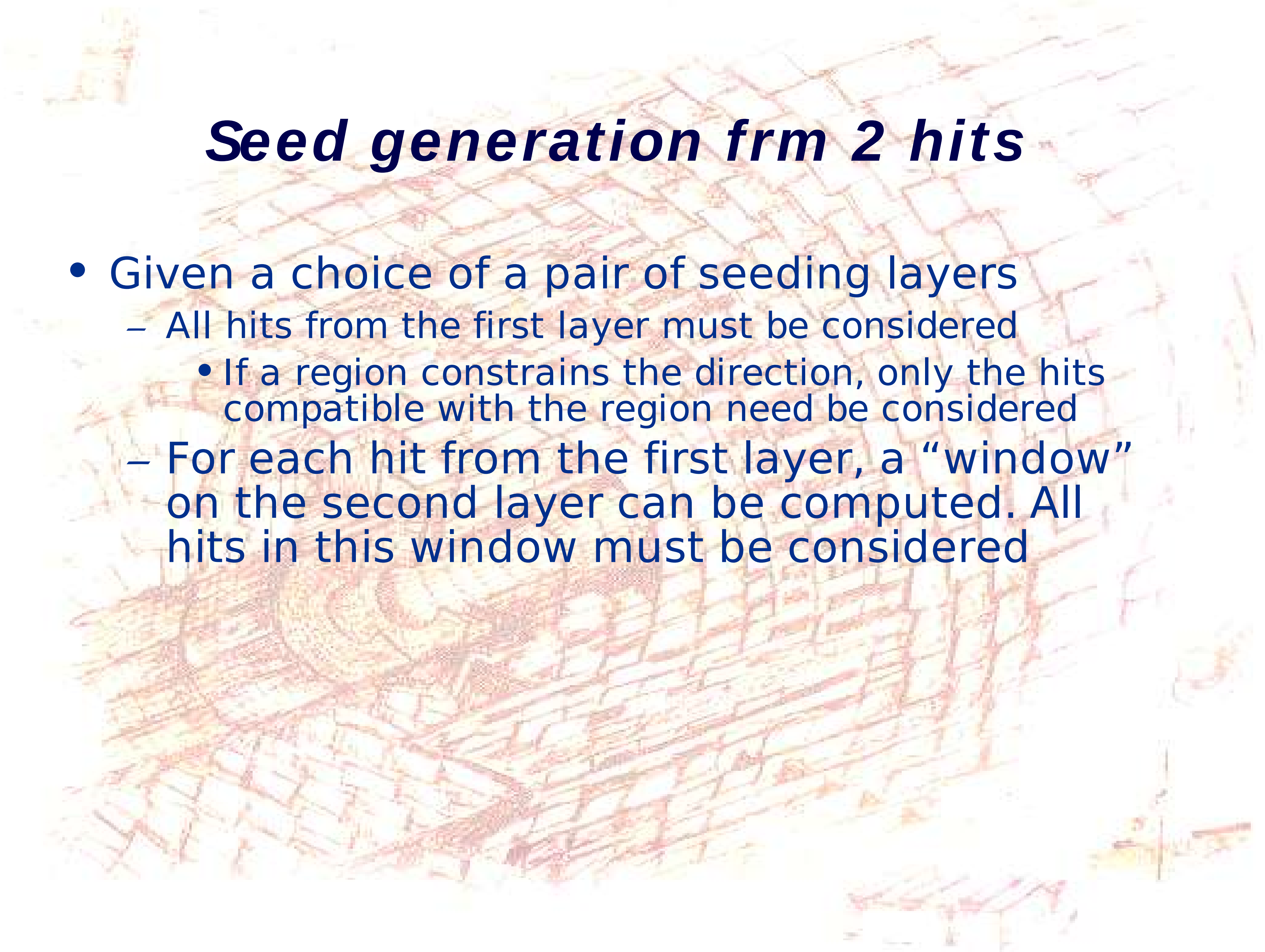
Global vs. regional seeding

- Seeding is never truly global, not even at LEP!
  - Typically limited to tracks compatible with some “interaction region”, and above some minimal momenta or Pt
  - Regional seeding in addition imposes limits on the direction of the tracks
  - So “global” means restricted in 3 dimensions and “local” means restricted in all 5 dimensions

Tracking region

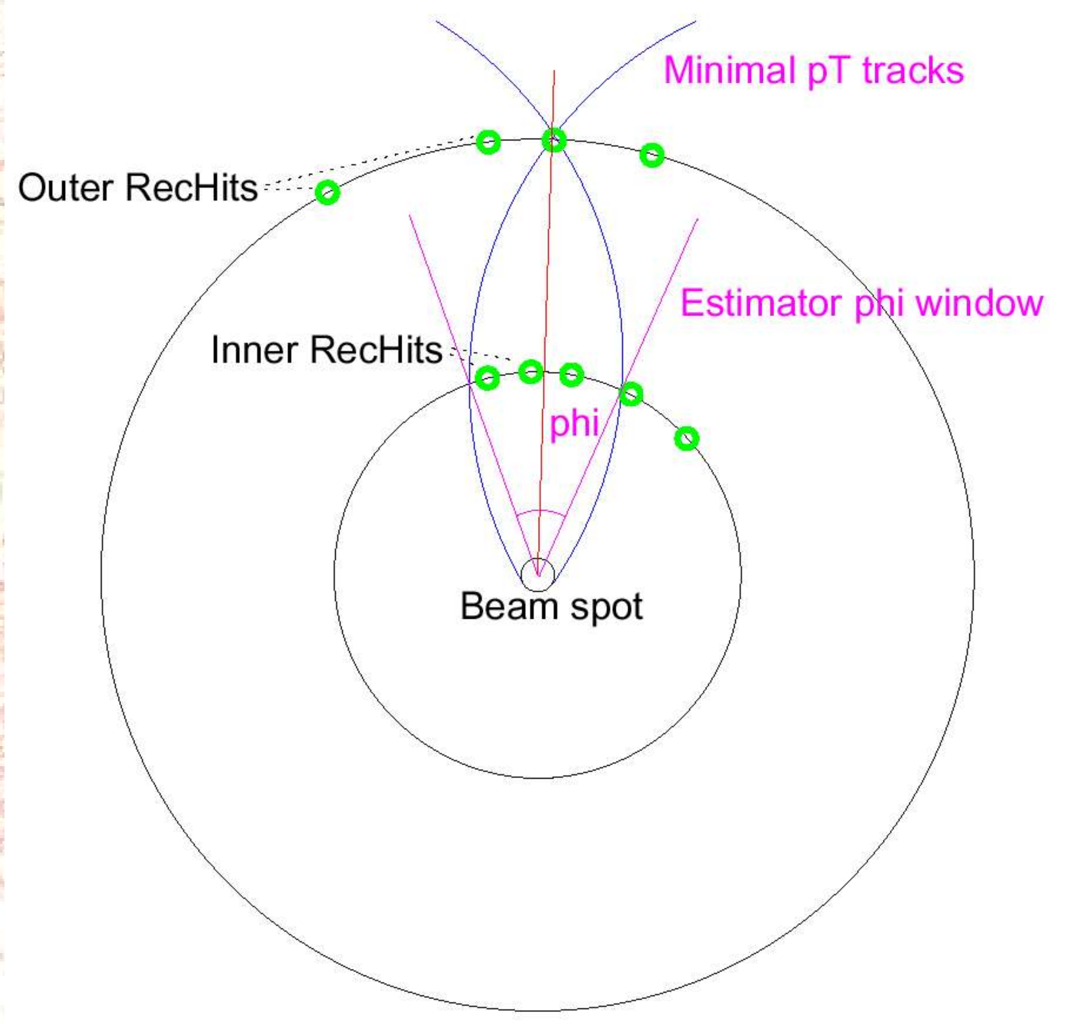
- Now we can define the tracking region more formally:
  - Since a helix has dimension 5, the tracking region is a volume in 5D space
  - In a collider it makes sense to define the positional part of the region around the “beam spot”
    - Since all tracks of interest come from there, and therefore cross it
  - The size of the positional part and the minimal Pt are determined by physics
    - e.g. 50 microns for prompt muons, 2 mm for muons from B decays

Seed generation frm 2 hits



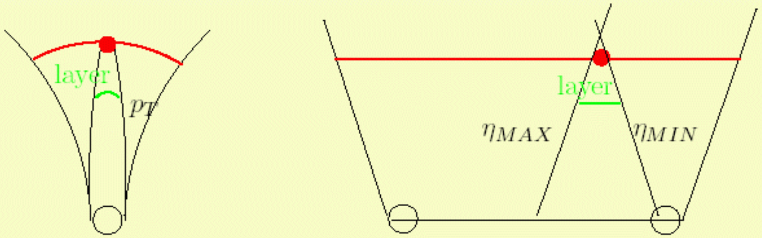
- Given a choice of a pair of seeding layers
  - All hits from the first layer must be considered
    - If a region constrains the direction, only the hits compatible with the region need be considered
  - For each hit from the first layer, a “window” on the second layer can be computed. All hits in this window must be considered

Internal seeding



Second hit details

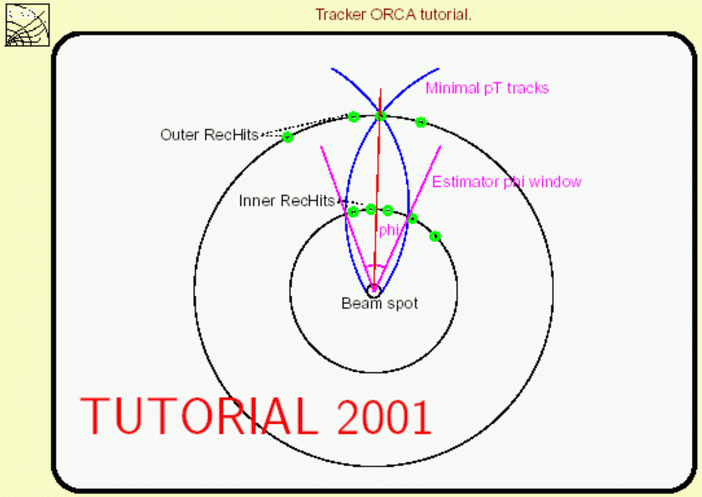
The hits from layers are accessed more than once. Since the φ constraint is more predictive than r/z the hits are kept in φ-sorted cache.



- using the analytical prediction for φ the STL binary search is used to find hits compatible in phi. No direction constraints are used.
- Each hit is tested against r/z.
  rz constraint from region is used.

Key points:

- caching,
- optimal sorting
- fast searching



Regional and partial seeding

- Seed generation from hit pairs uses all 5 constraints of the region
  - Tracks reconstructed from these seeds will be mostly compatible with the region
  - Except for the Pt constraint, for which the accuracy from 2 hits +IR is usually not enough
- Seeding is ususally not partial (at least at present...)

Trajectory building

- Given a seed, Kalman filter track finding proceeds like this, in a loop until the end of the tracker
  - Search for next compatible hits (detector and implementation specific, most time consuming)
  - Propagation to the surface of the next hit
    - May already be done in previous step for the purpose of computing compatibility with the hits
  - Update with the next hit
- In case there are more than one compatible hits, more than one candidate track should be followed

Combinatorial explosion

- At every layer there is a possibility of multiplying the existing candidates by the number of compatile hits. If left unchecked, this leads to “combinatorial explosion”
  - One rare event is enough to spoil things
- Different strategies exist to limit the growth

Example of hit combinatorics

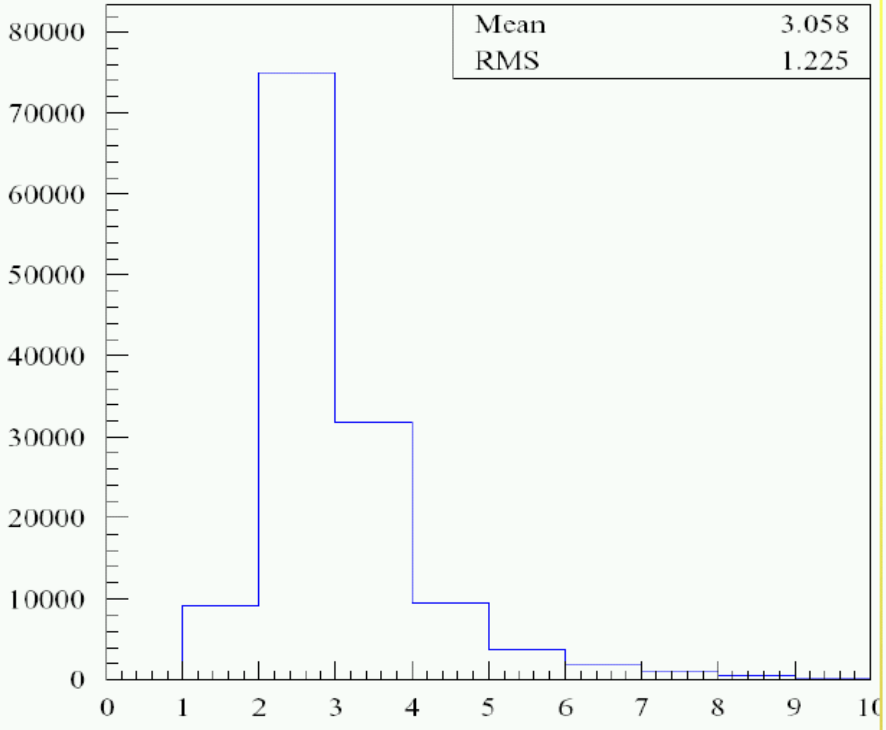


Figure 14: *Number of compatible hits found on TIB layer 1 for each trajectory candidate when leaving Barrel Pixel layer 3 for 100 GeV b jets without pile up.*

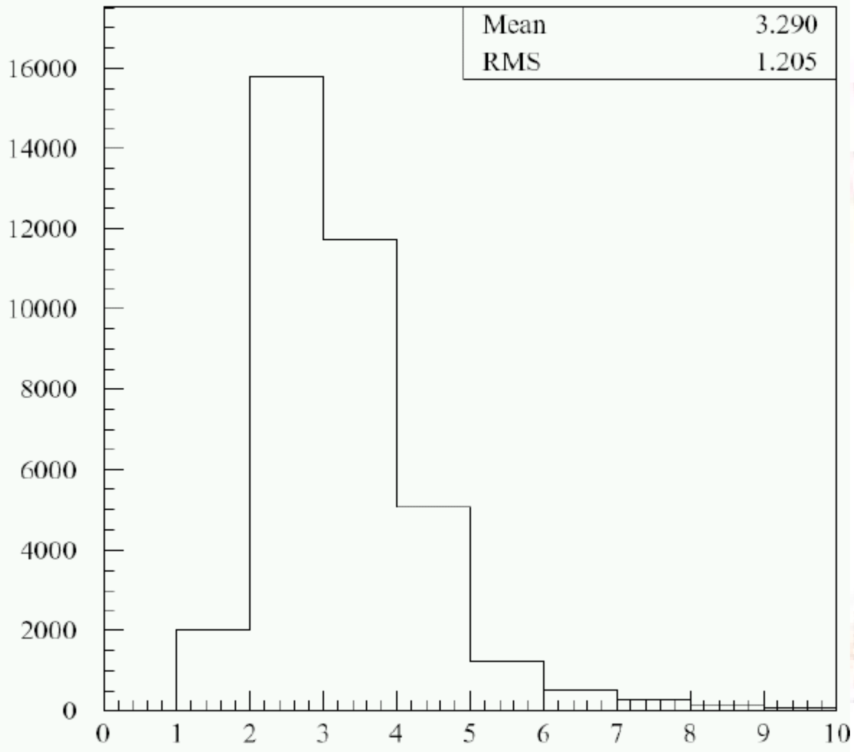


Figure 21: *Number of compatible hits for each trajectory candidate when leaving the Forward Pixel disk 2 (100 GeV b jets without pile up).*

Trimming of combinatorics

Very democratic, equal opportunity, etc.

- All track candidates advance in parallel from layer to layer
- The candidates are trimmed after each layer
  - Only the N best ones survive (N=5 is 0.1% below full algorithmic efficiency)
  - Candidate quality defined as chi^2, with a penalty for missing hits
- Intermediate cleaning:
  - If two candidates differ by only one hit in the middle, only the better one is kept

Resolution of ambiguities

- A single seed typically produces ether no tracks at all or several track candidates
- These candidates are “mutually exclusive” in the sense that they share many hits
- The ambiguity resolution can be very simple, just based on the fraction of shared hits (the “best” candidate survives), or quite complex

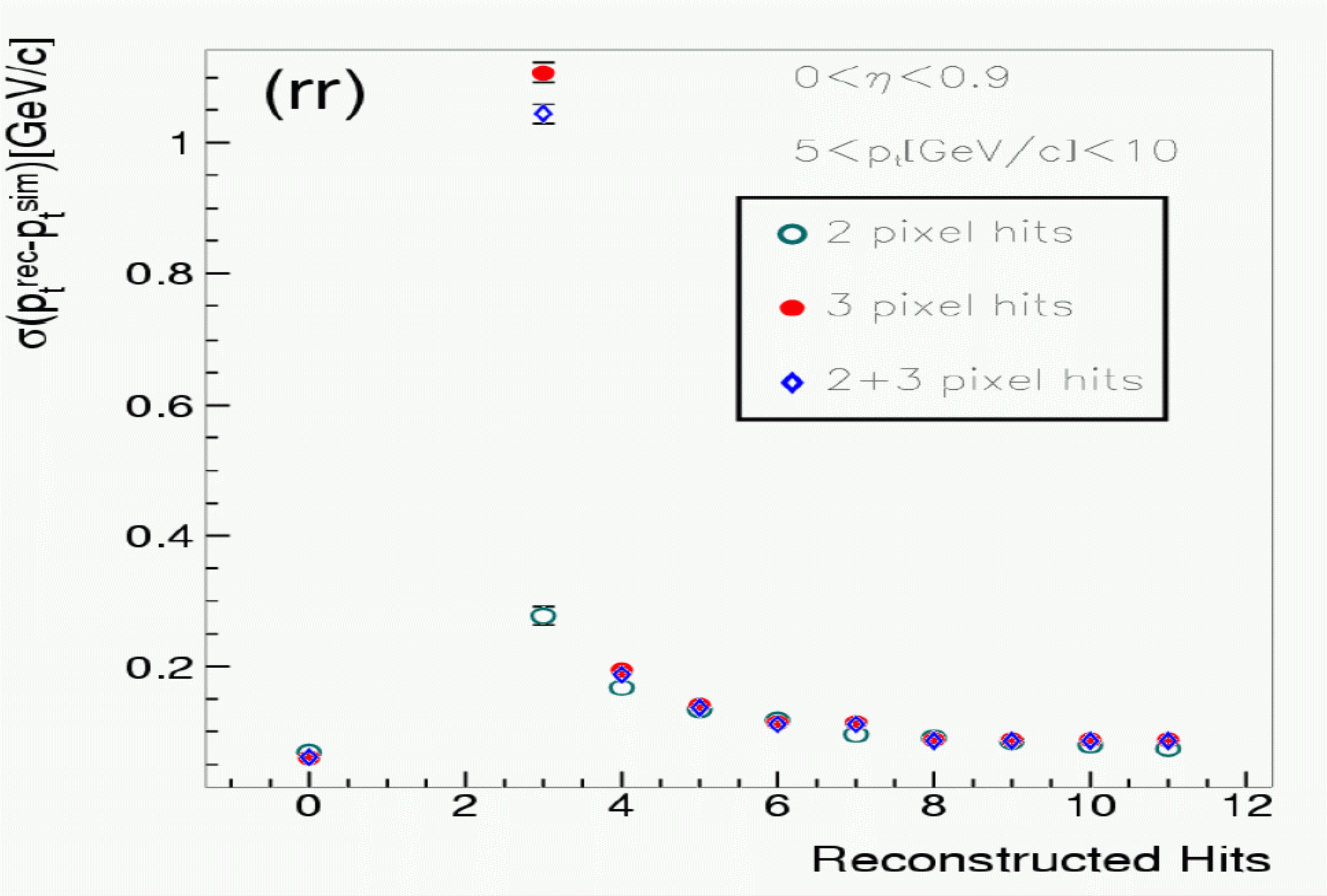
Smoothing

- At the end of the “forward” fit, the track parameters are known precisely at the exit of the tracker, but completely unknown at the origin (infinite errors…)
- We can perform a “backward fit”, using only the hits from the forward fit (no pattern recognition) to find the parameters at origin
  - But we lose them at the other end
- A procedure, called *smoothing*, allows to combine the forward and backward fits in such a way that the parameters are optimally known at every measurement

Partial track finding

- In order to satisfy the event selection, the accuracy of the full track may not be required
  - Ex. if the selection criteria is “a track above 20 GeV”, then track building for a 1 GeV track can be stopped after only 3 or 4 hits, without any loss of efficiency or accuracy on the 20 GeV tracks

Track accuracy as a function of number of hits



Regional and partial trajectory building

- If the seed is in the region, the track will likely be in the region too, for position and direction
  - It is expensive to check for impact parameter and direction at origin during track building, since this requires a full backward fit after every hit
- The region check on Pt translates in partial tracking
  - Pt is easy to check after every update, since it does not change significantly (except for electrons), and the value at the “end” of the track is OK

Stopping conditions

- The stopping of track building “as soon as momentum is for sure below threshold” is a “stopping condition”
  - Other useful SC:
    - stop after N good hits
    - Stop as soon as the momentum is definitely ***above*** the threshold
- Can be an abstract component of track building, allowing any user SC

Example: Tracker L2 muon trigger

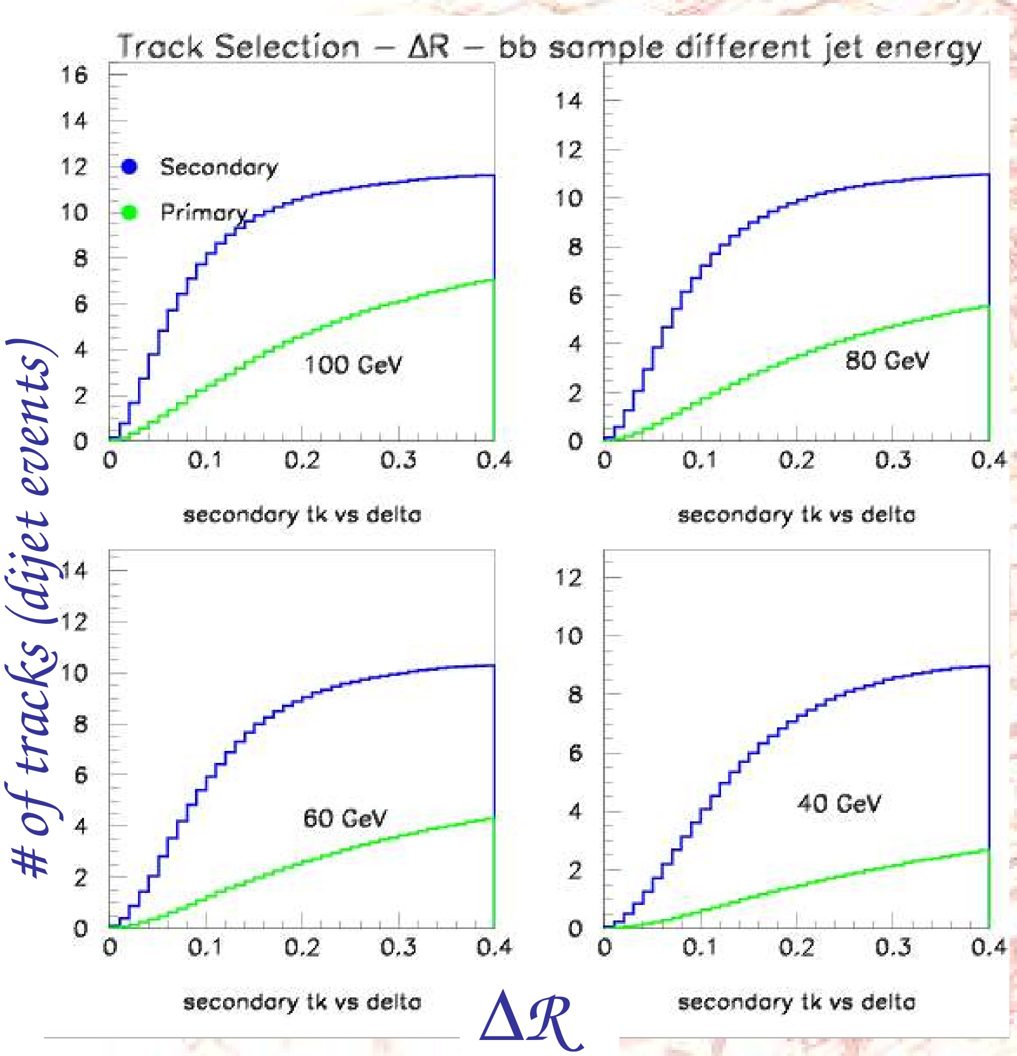
- Conditions:
  - High Pt threshold – around 15 GeV
  - Primary muon: transverse impact parameter below 30 microns
  - Direction known from L1 with poor accuracy
- Tracker information needed:
  - confirm existence of track with the selection criteria above
  - Check isolation (no other tracks with Pt above 0.8 GeV in a cone of some size around the muon)

- There are two track reconstructions involved: of the muon and of the tracks in the isolation cone.
  - For the muon, the z position of the vertex is not known. The region is very long is Z (30 cm), but very constrained in Pt and in tranverse size
  - For the isolation tracks, they should come from the same interaction, and have the same primary vertex
    - The muon Z impact parameter defines the vertex
    - The extent around the vertex in Z and R is larger (about 1 mm) to include tracks from B decays
- The result of the first reconstruction defines the region for the second one!

B tagging a jet

- A jet from a b quark has a secondary vertex typically a few mm from the primary, and therefore tracks with large impact parameters
- In HLT the direction of the jet is given by the calorimeter
- The size of the tracking region is defined by the properties of B-jets (next slide)
- Partial reconstruction is defined by limiting the number of hits
  - Impact parameters expensive to check during track building

Region of Interest

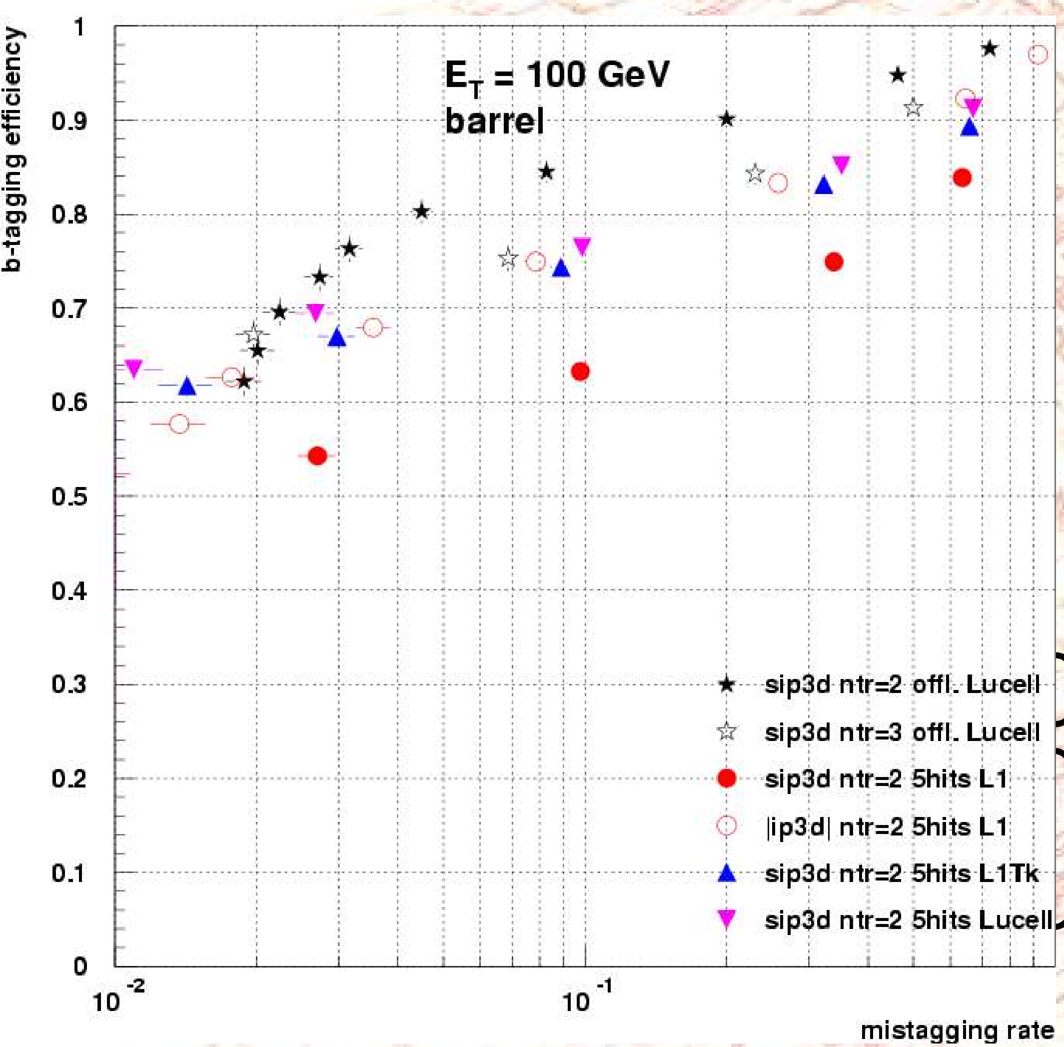


Average number of tracks
100 GeV sample
[PYTHIA/Lucell]

?R cut	*All*	**0.4**	*0.15*
Primary	*15*	**7**	*3.5*
Secondary	*12*	**12**	*10*

Best Region of Interest
ΔR<0.4

L1+Tracks B-tag (2)



} OFFLINE

} HLT

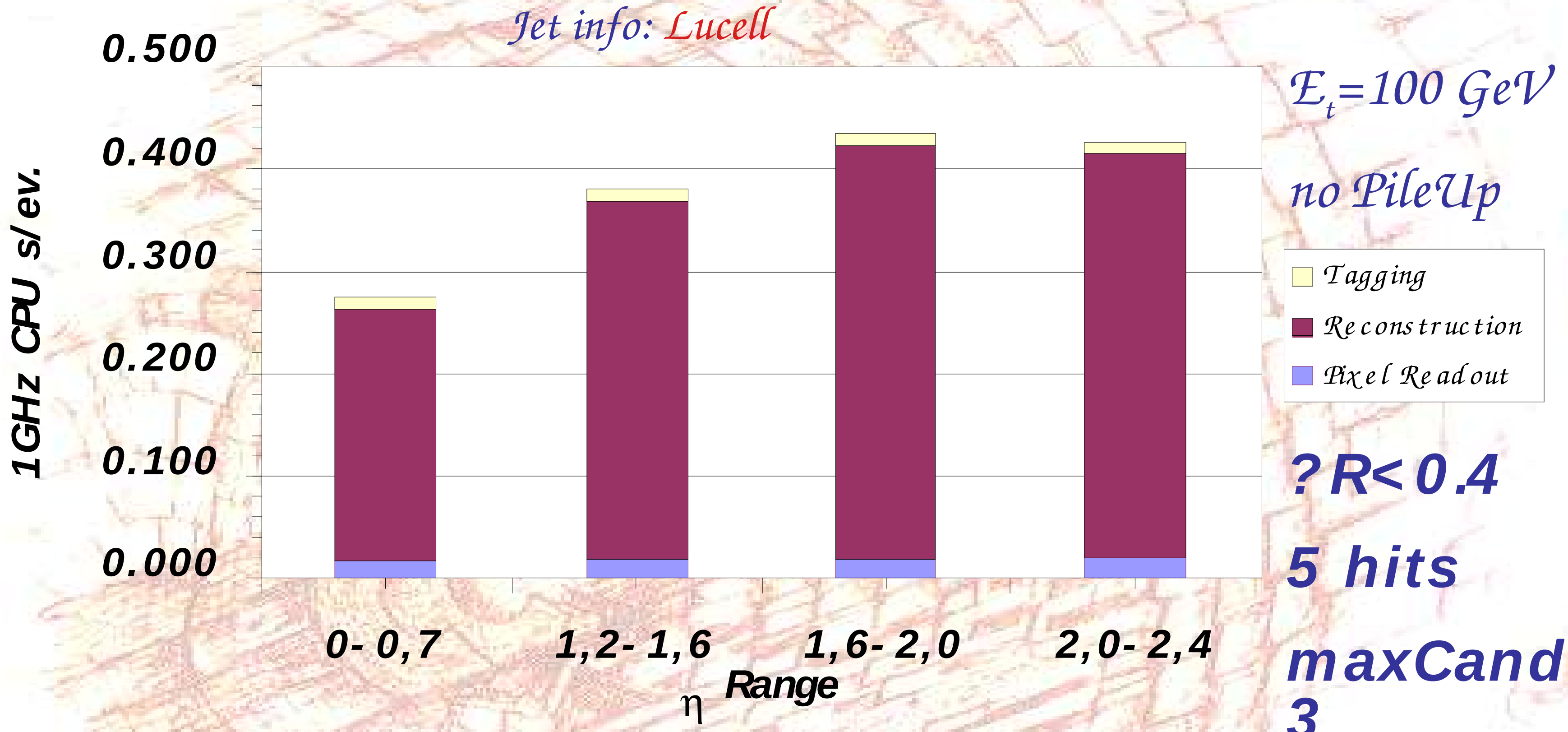
E$_t$=100 GeV jets

barrel 0.<|η|<0.7

Better **b** jets efficiency with 3d IP

Jet-tag: 2 tracks with S_{IP}>0.5,1.,1.5,2.,2.5,3.,3.5,4.

Timing bb jets



E_t=100 GeV

no PileUp

?R<0.4

5 hits

maxCand=3

Increasing of reco time towards forward regions

Tagging algorithm: <10 ms/ev !!!

Tracker use in jets

In addition to B tagging, the tracker can improve
- Jet energy measurement
- Jet direction estimate
- Separate jets from trigger and pile-up events

All this comes at a (CPU) price. So no matter how fast the track reconstruction, it will never be fast enough

(some even consider global reconstruction of all tracks at HLT to improve missing Et resolution)

Conclusions

Using the same track reconstruction framework and algorithms it is possible to achieve both

- offline requirements on reconstruction efficiency and accuracy and
- HLT requirements on CPU speed and rejection power

The key points are

- Regional reconstruction
- Partial reconstruction
- Action on demand