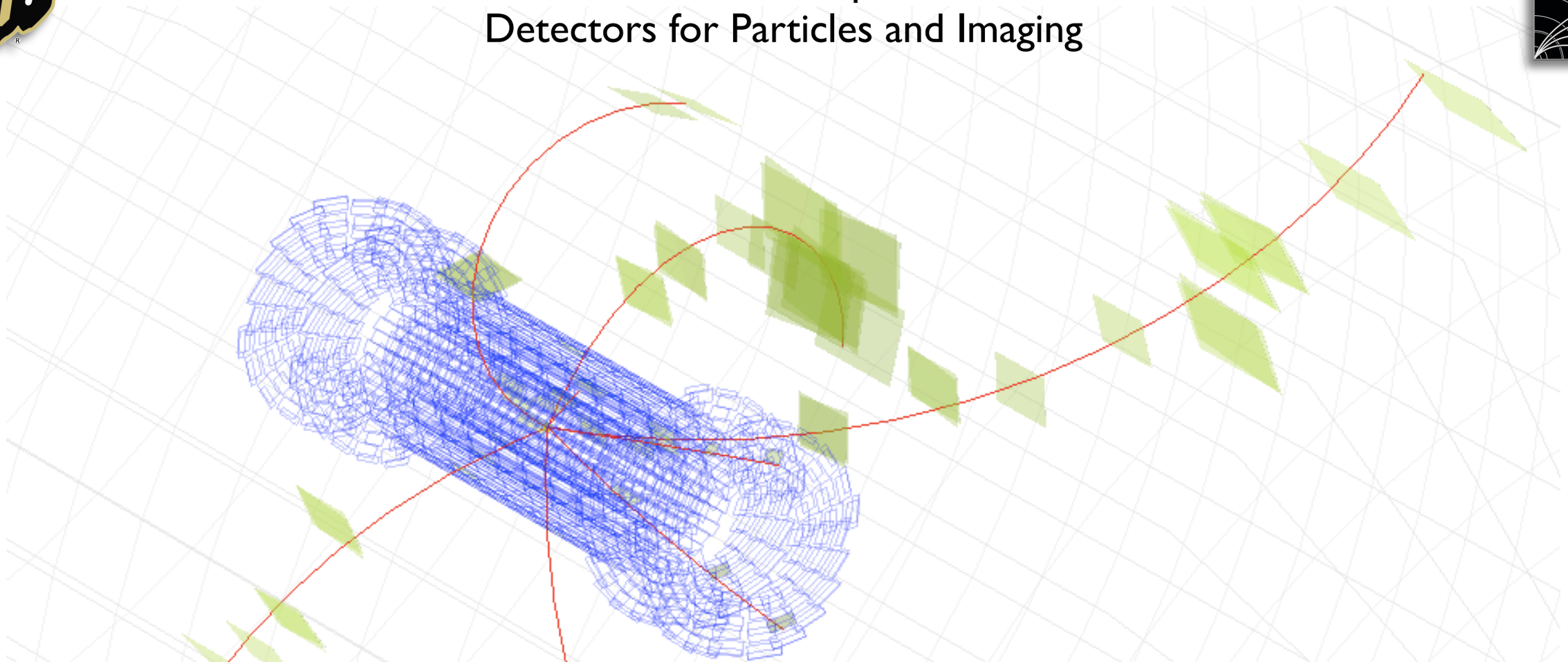
Pixel 2010: International Workshop on Semiconductor Pixel Detectors for Particles and Imaging



Tracking and Vertexing Capabilities of the CMS Tracking Detector with the First LHC Data

Mauro Dinardo
University of Colorado - Boulder, USA
On Behalf of the CMS Collaboration

06/09/2010

Outline

1. The CMS Tracking Detector Layout

2. The CMS Tracking Algorithm
   - Efficiency
   - Momentum Resolution
   - Impact Parameter Resolution

3. The CMS Primary-Vertexing Algorithm
   - Efficiency
   - Resolution

4. b-Jet Identification
   - Efficiency

5. An Application of the Pixel Stand-Alone Tracking: *Beam-Spot Monitor*

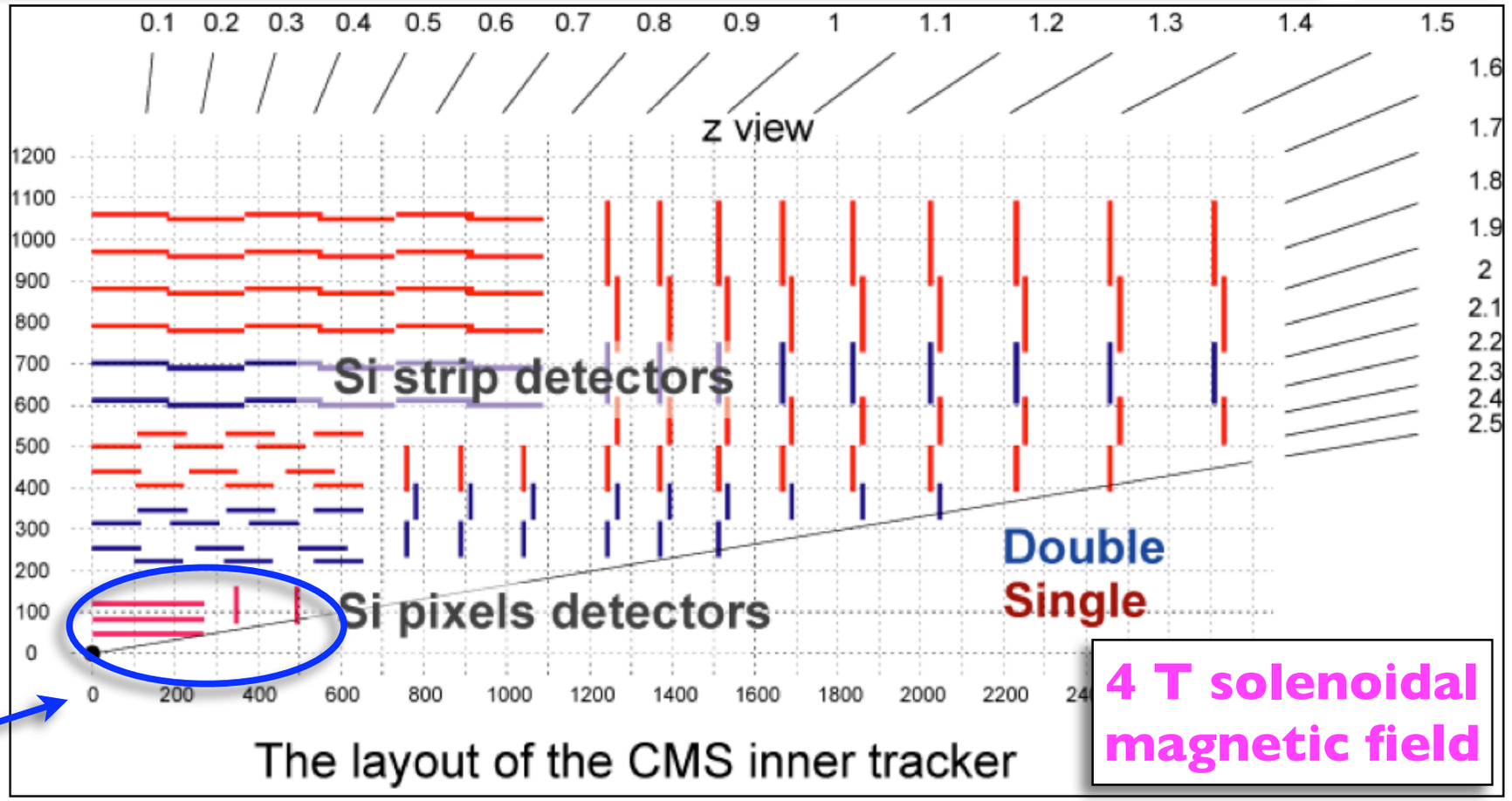
6. Conclusions

(all shown measurements are performed at $\sqrt{s}$ = 7 TeV)

The CMS Tracking Detector Layout

All-Silicon based tracker detector

Strips
- 9.3 M channels
- ~200 m^2 sensor area
- 10 barrel layers
- 24 endcap disks (12 per side)



1/4 of the whole tracking system

Pixels
- 66 M channels
- ~1.1 m^2 sensor area
- 3 barrel layers
- 4 endcap disks (2 per side)
- innermost layer at r = 4.3 cm

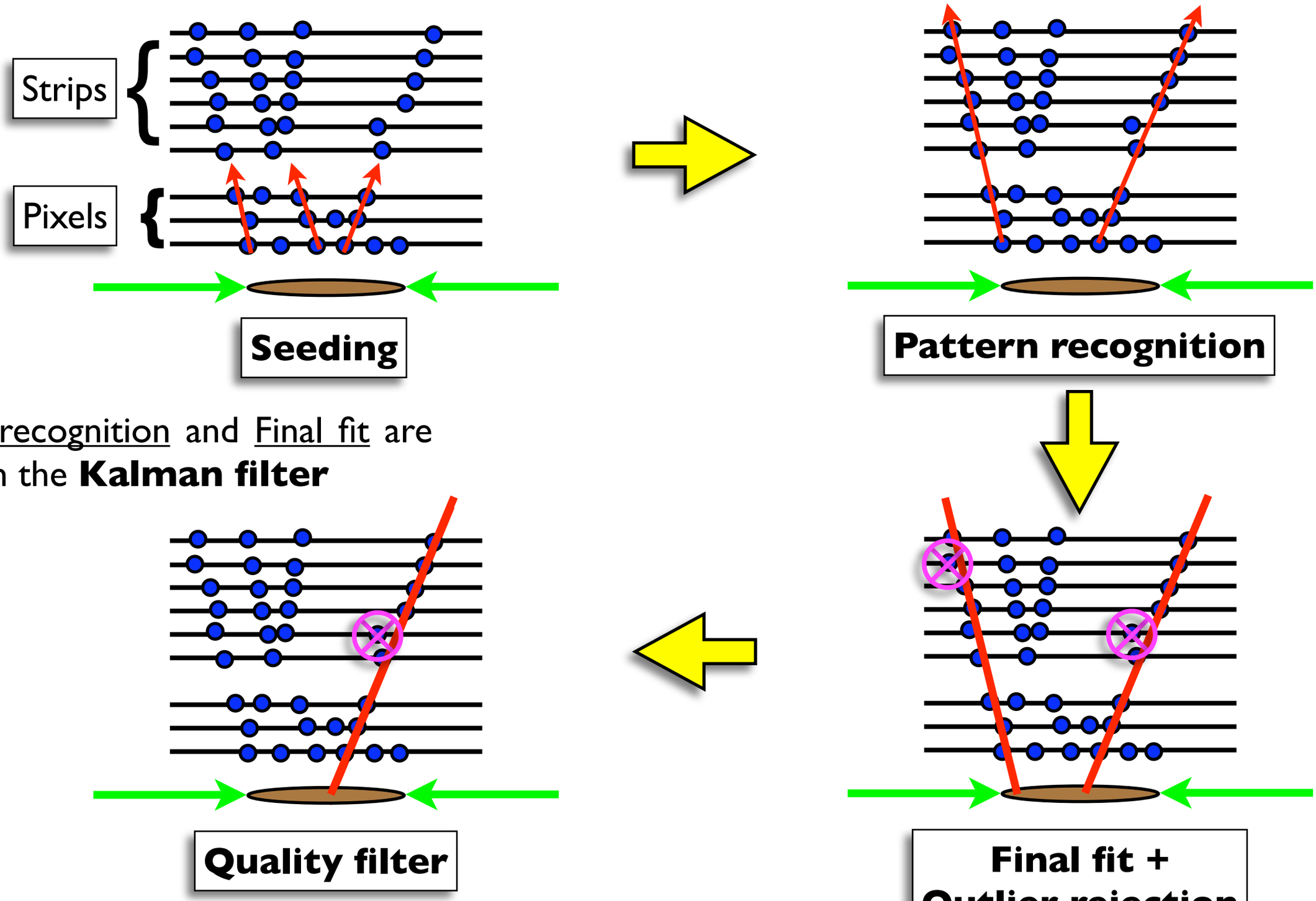
Operational fraction:
- Strips: 98.1%
- Pixels: 98.3%

Full coverage up to $|\eta| \leq 2.5$

The CMS tracking detector allows for a full reconstruction of events up to the LHC design luminosity: 10^{34} cm^{-2} s^{-1} (i.e. events with ~1000 charged tracks)

The CMS Tracking Algorithm

The CMS tracking algorithm is divided in 4 stages:

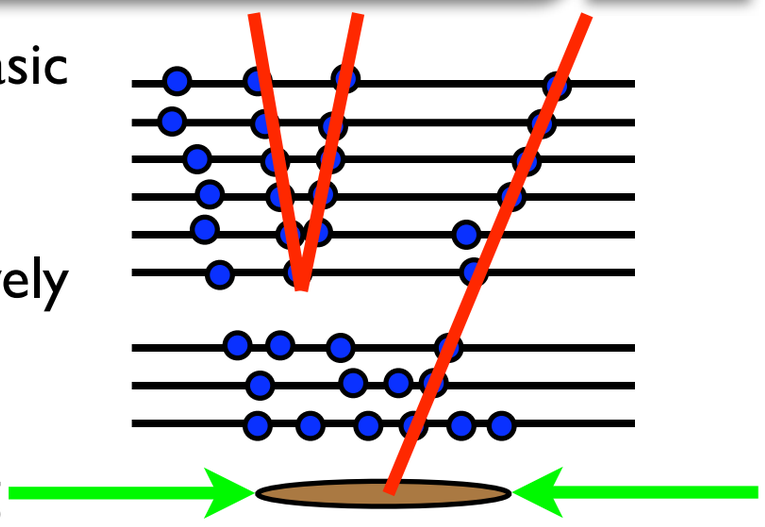


Pattern recognition and Final fit are based on the **Kalman filter**

The CMS Tracking Algorithm

The final collection of tracks is obtained by repeating 6 times the 4 basic stages: **iterative tracking**

- **Steps 1-2:** based on pixel seeds, allow to find tracks with relatively high momentum
- **Step 3:** allows to find low momentum (short) tracks
- **Steps 4-6:** allow to find tracks which are not found by pixel seeding

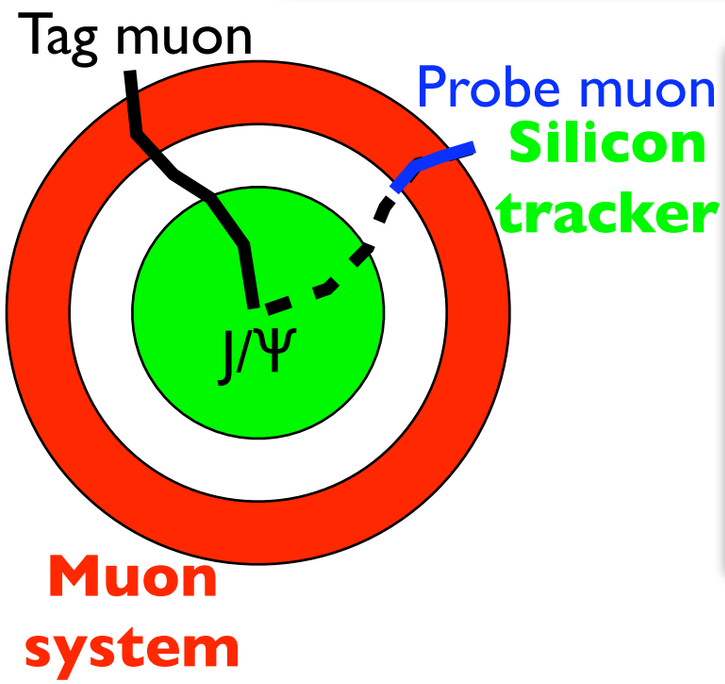


Final retained trajectories

Iterative Steps	Seed Type	Seeding Sub-Detector	Min p_T
1	Triplets of Hits	Pixel	0.8 GeV/c
2	Pairs of Hits	Pixel + Strip (endcap)	0.9 GeV/c
3	Triplets of Hits	Pixel	0.075 GeV/c
4	Pairs of Hits	Pixel + Strip (endcap)	0.35 GeV/c
5	Pairs of Hits	Strip (inner barrel + endcap)	0.5 GeV/c
6	Pairs of Hits	Strip (outer barrel + endcap)	0.8 GeV/c

- The momentum of charged particles can be measured across five order of magnitude range: 100 MeV/c - 1 TeV/c
- The fake-track rate is 2-3% (5%) for ttbar events reconstructed in the barrel (endcap)

Tracking Efficiency for Muons (using J/Ψ)



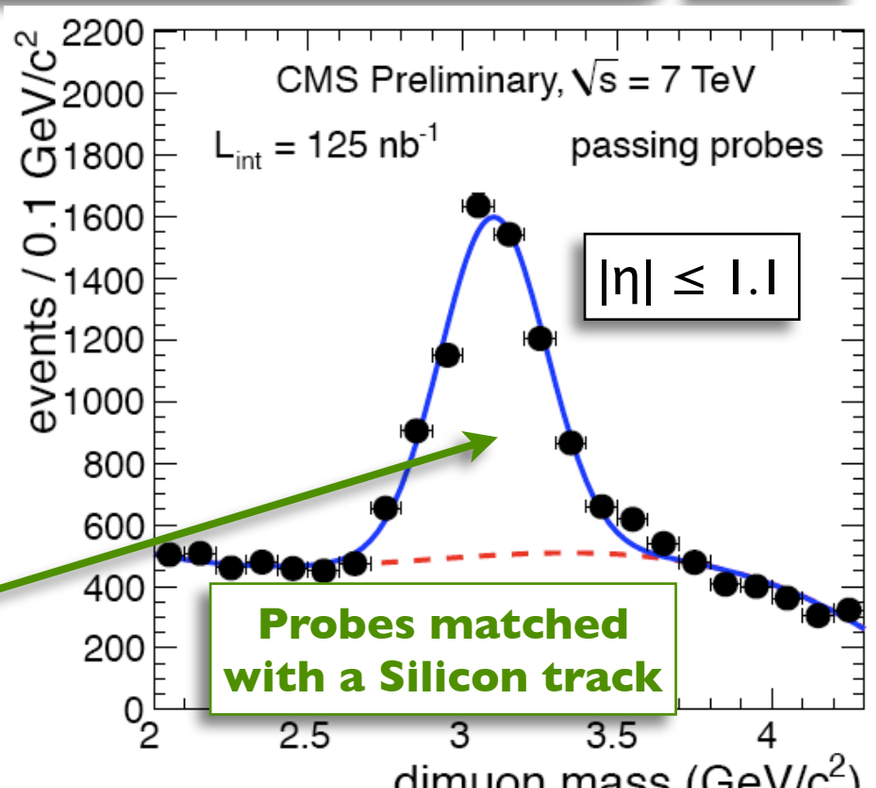
The tracking efficiency is estimated from **J/ψ→μμ** invariant mass spectrum

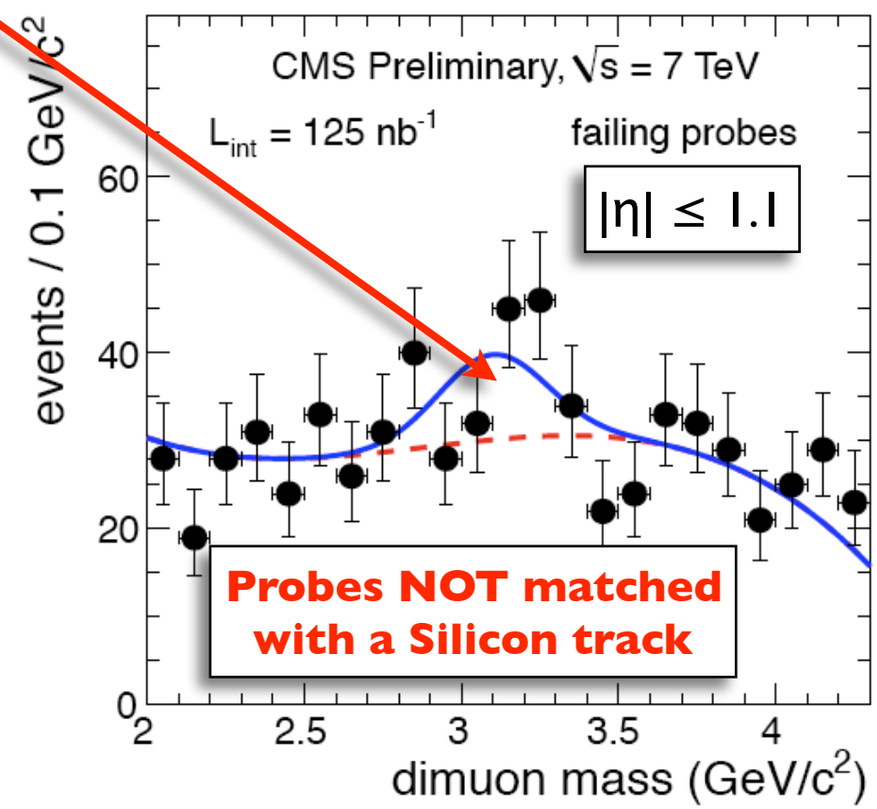
μ: tag muon from stringent track quality requirements

μ: probe muon which might or might not be matched with a Silicon track

$$\varepsilon = \varepsilon_T \varepsilon_M + (1 - \varepsilon_T \varepsilon_M)\varepsilon_F$$

- ε: Measured efficiency
  - Match/(Match+~~Match~~)
- **ε_T: Tracking efficiency**
- ε_M: Matching efficiency
- ε_F: Random matching





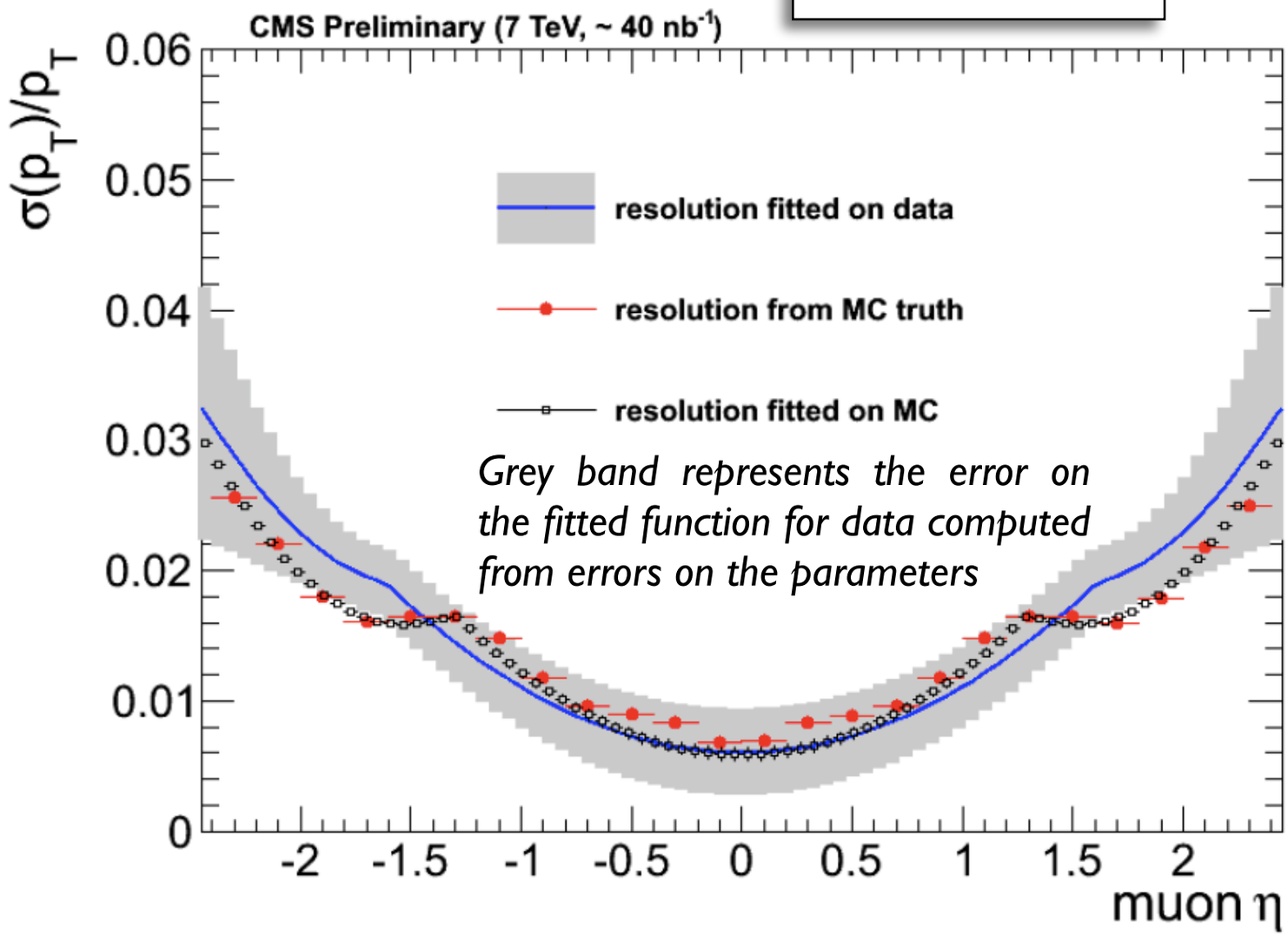
Region	Data Eff. (%)	Sim Eff. (%)	Data/Sim
$0.0 \le \lvert\eta\rvert < 1.1$	$100.0^{+0.0}_{-0.3}$	$100.0^{+0.0}_{-0.1}$	$1.000^{+0.001}_{-0.003}$
$1.1 \le \lvert\eta\rvert < 1.6$	$99.2^{+0.8}_{-1.0}$	$99.8^{+0.1}_{-0.1}$	$0.994^{+0.009}_{-0.010}$
$1.6 \le \lvert\eta\rvert < 2.1$	$97.6^{+0.9}_{-1.0}$	$99.3^{+0.1}_{-0.1}$	$0.983^{+0.009}_{-0.010}$
$2.1 \le \lvert\eta\rvert < 2.4$	$98.5^{+1.5}_{-1.6}$	$97.6^{+0.2}_{-0.2}$	$1.010^{+0.015}_{-0.016}$
Combined	$98.8^{+0.5}_{-0.5}$	$99.2^{+0.1}_{-0.1}$	$0.996^{+0.005}_{-0.005}$

Track Momentum Resolution (using J/Ψ)

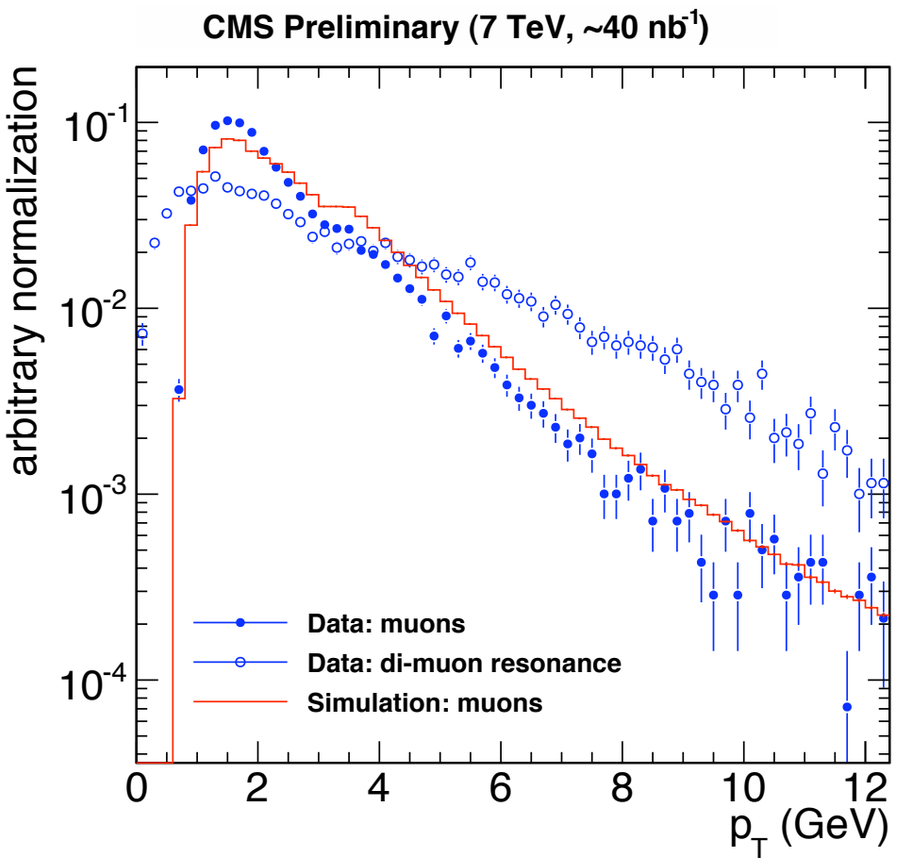
The p_T resolution is estimated from the width of the J/ψ→μμ invariant mass peak
(i.e. the J/ψ width is expressed as a function of the kinematics of the two tracks)

M(J/ψ) ~3097 MeV, Γ(J/ψ) ~90 keV

Resolution



Muon's momentum spectrum



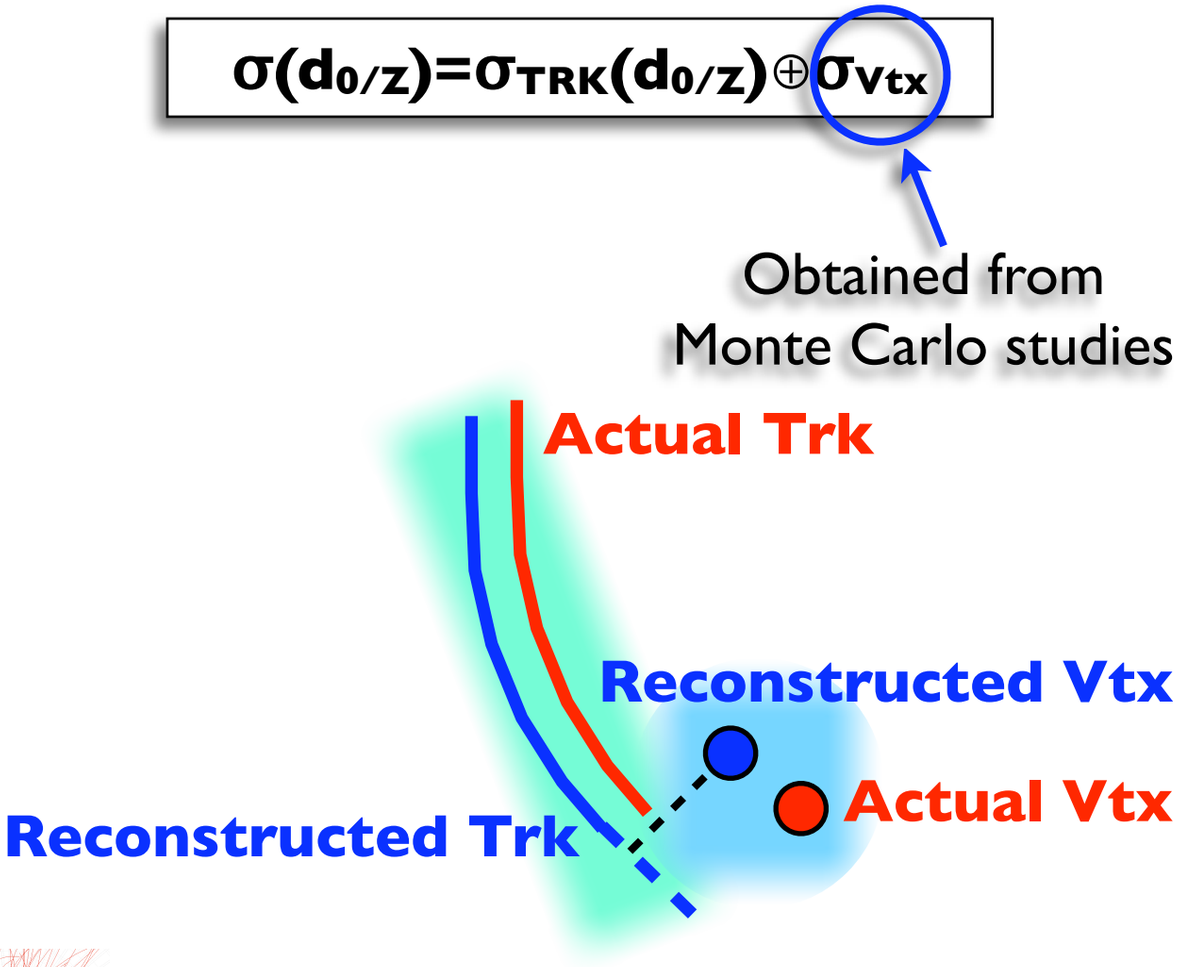
Track Impact Parameter Resolution (1/2)

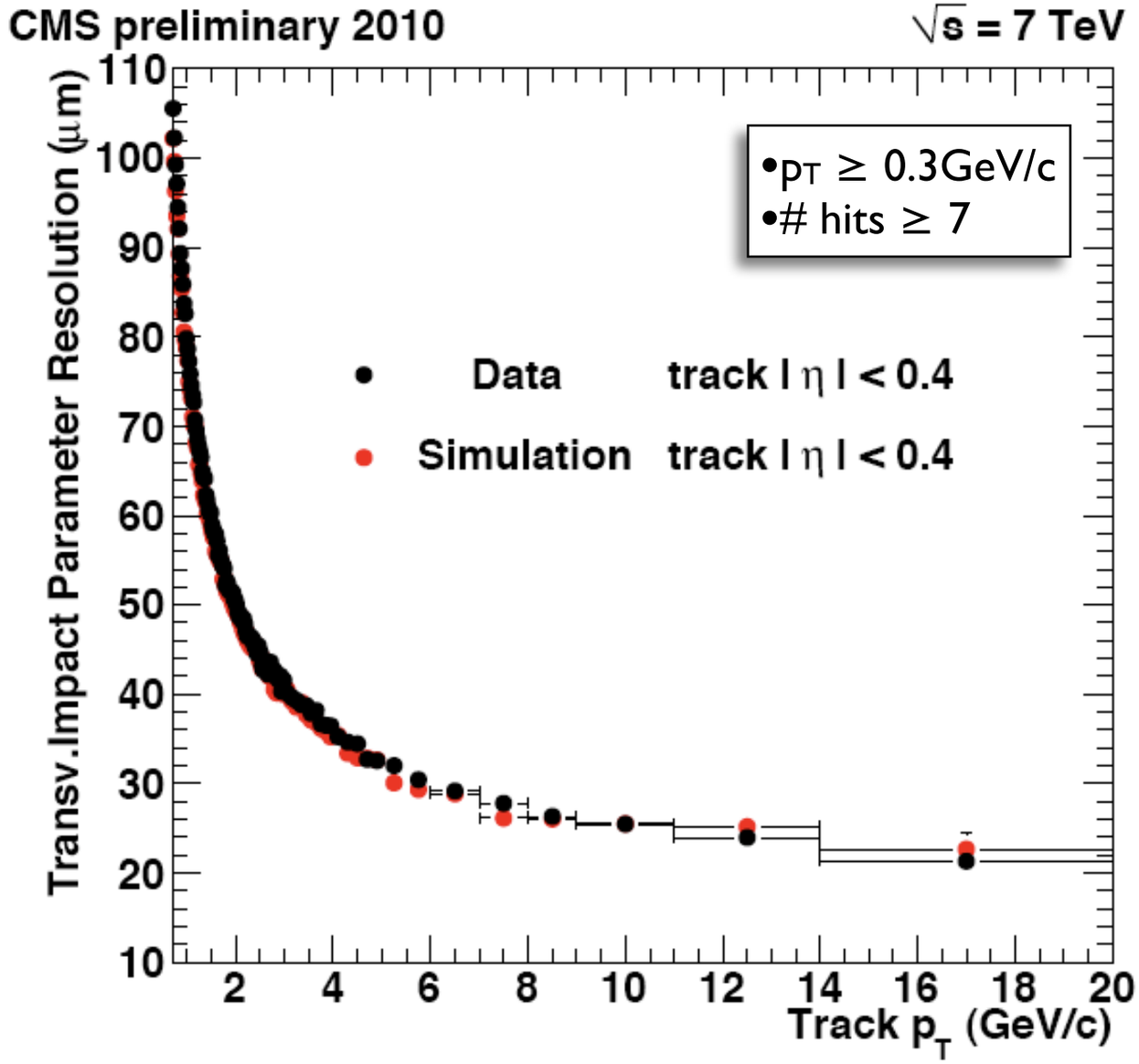
The resolution on the track impact parameters is extracted from the Data evaluating the impact parameter with respect to the primary vertex position

The width of the distribution of the track impact parameter (i.e. $\sigma(d_0)$ or $\sigma(d_Z)$) depends on the track impact parameter resolution $\sigma_{TRK}(d_{0/Z})$ and on the vertex position resolution σ_{Vtx}

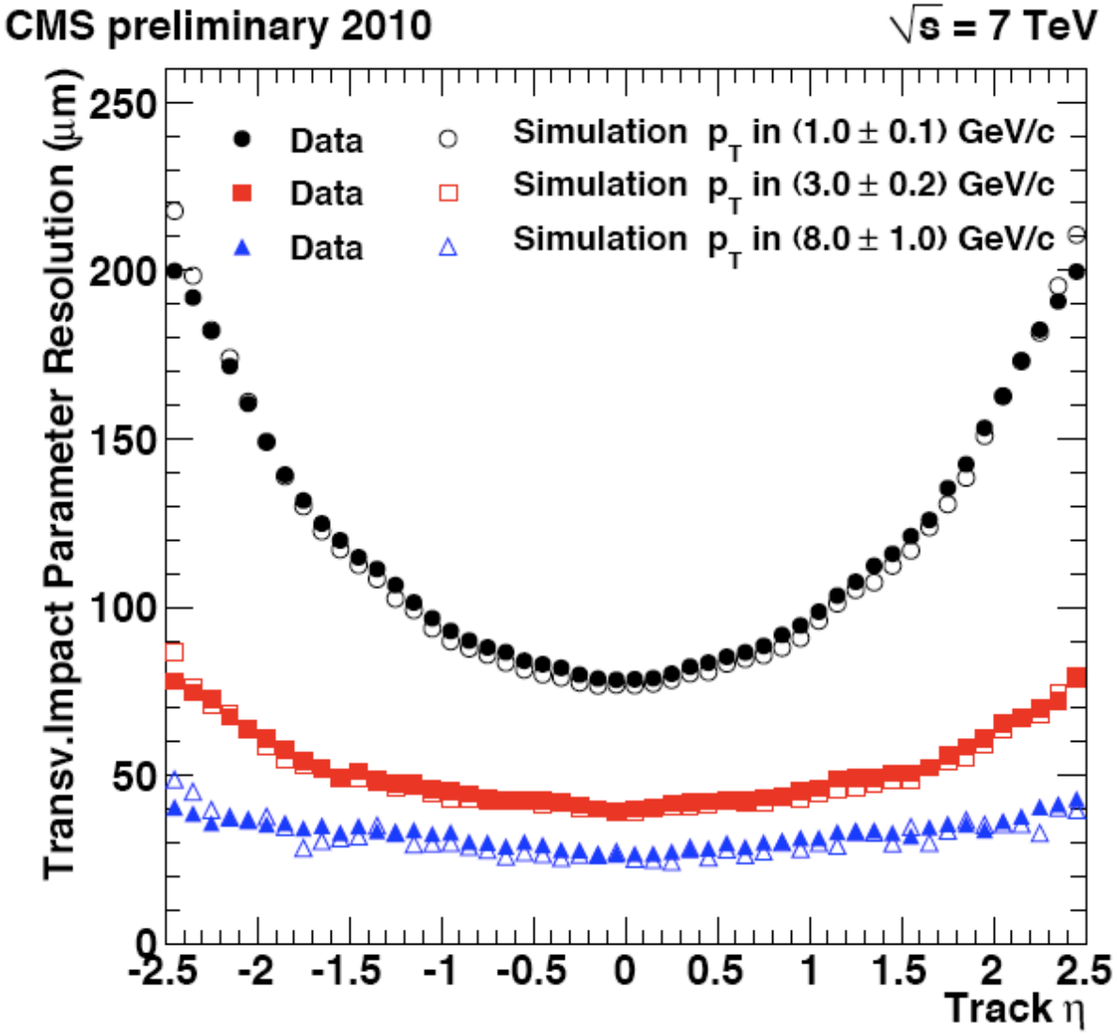
$$\sigma(d_{0/Z}) = \sigma_{TRK}(d_{0/Z}) \oplus \sigma_{Vtx}$$

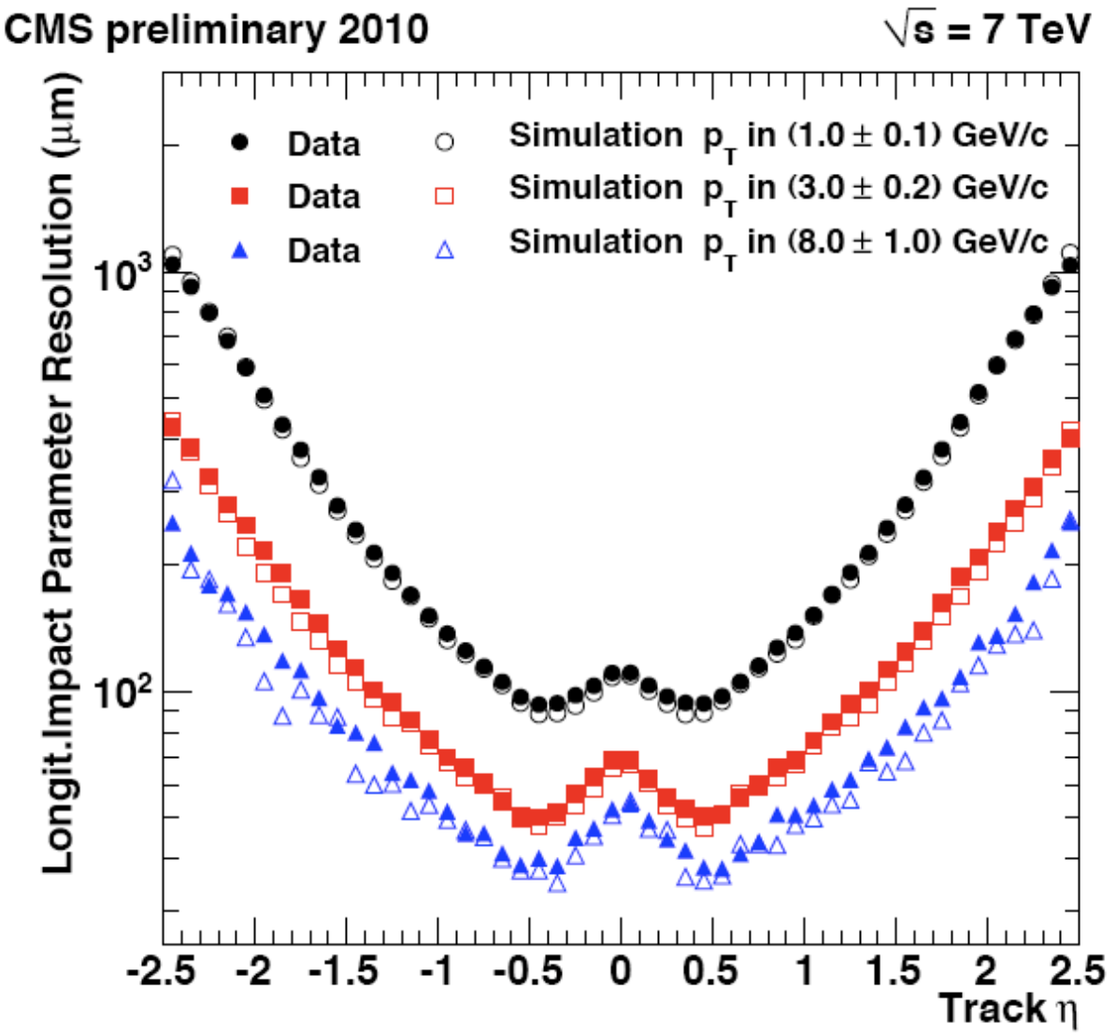
Obtained from Monte Carlo studies





Track Impact Parameter Resolution (2/2)



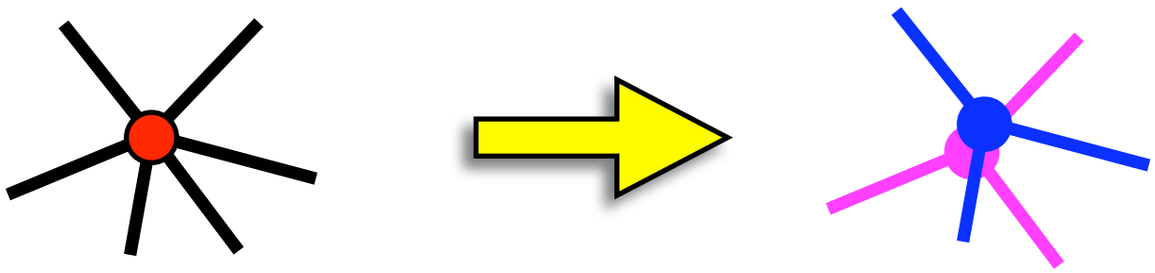


Good agreement between Data and Simulation for a wide range of track p_T and η

Primary-Vertex Efficiency

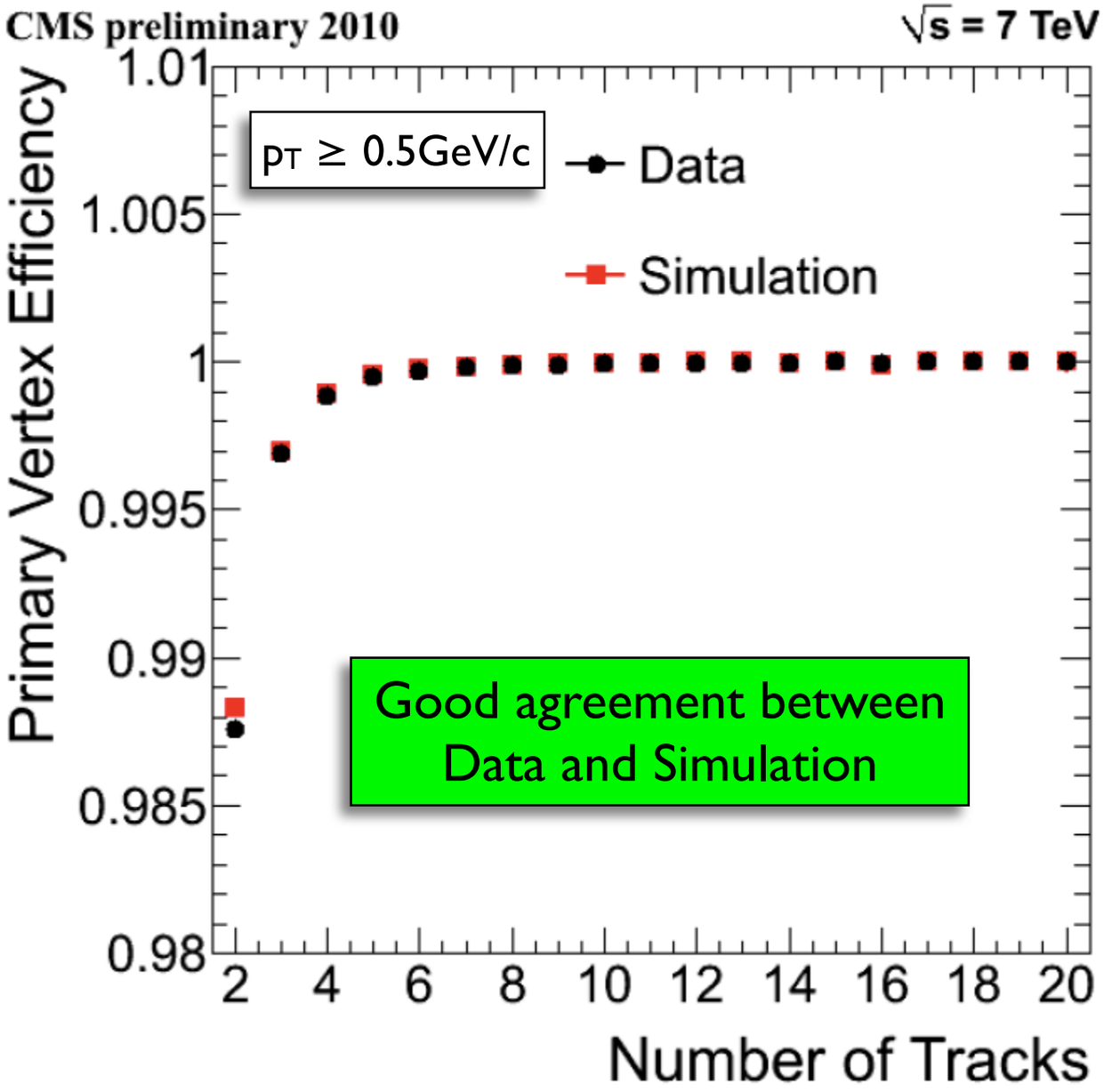
The primary-vertexing algorithm is based on an adaptive Kalman filter: after a first coarse approximation of the vertex location, the Kalman filter updates the position track-by-track (tracks are weighed according to their longitudinal distance to the vertex, *see backup slides*)

The primary-vertex efficiency and position resolution are measured with the data-driven method based on vertex-splitting

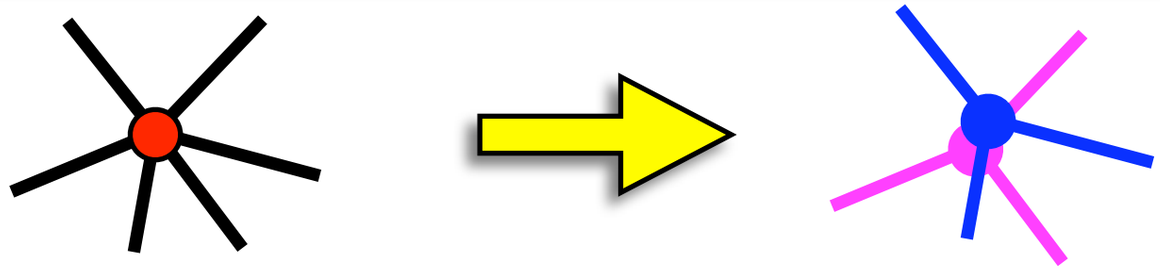


Primary vertex is split into two sub-sets: **TAG** and **PROBE**

Efficiency = # PROBES / # TAGS



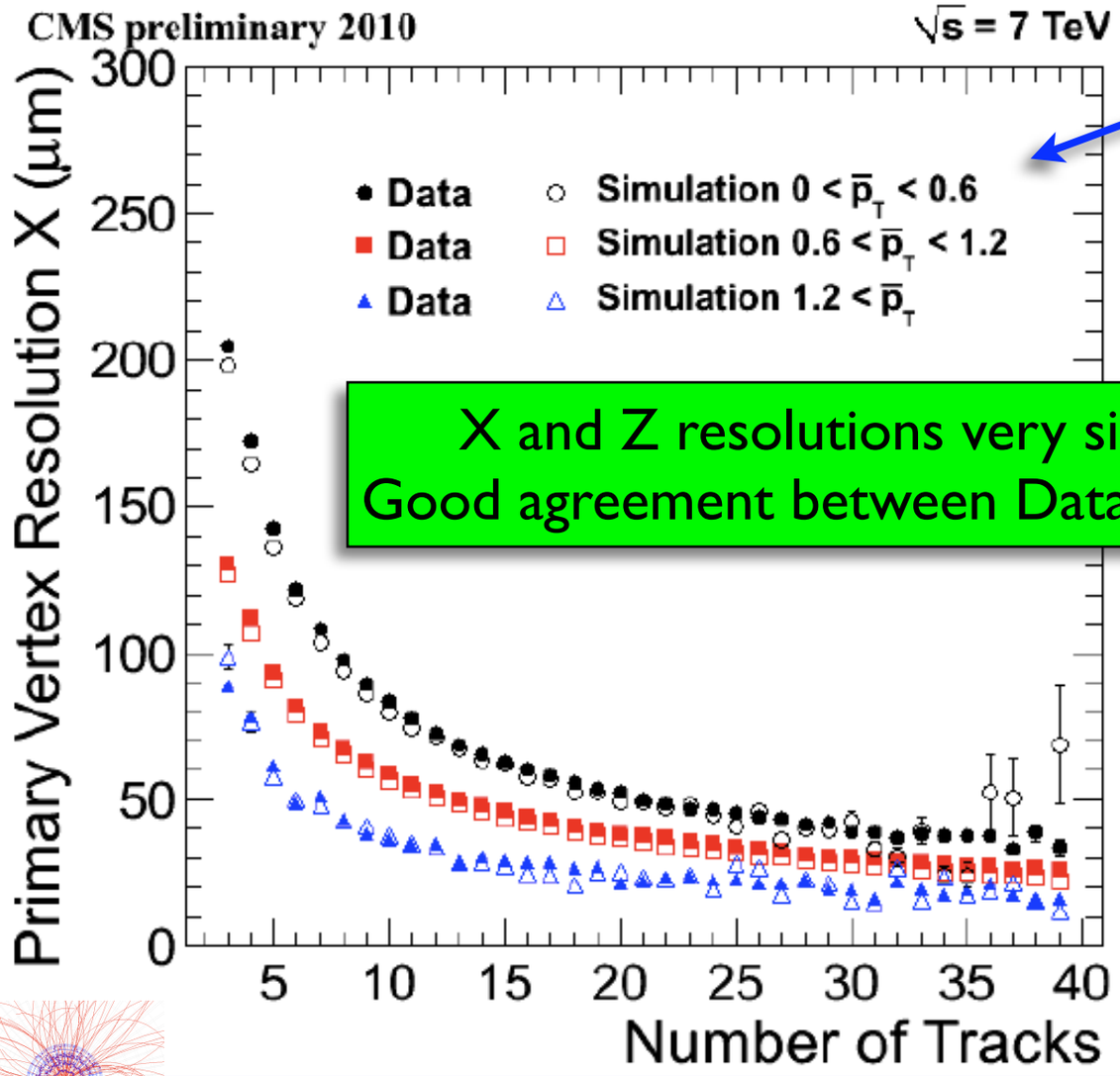
Primary-Vertex Resolution

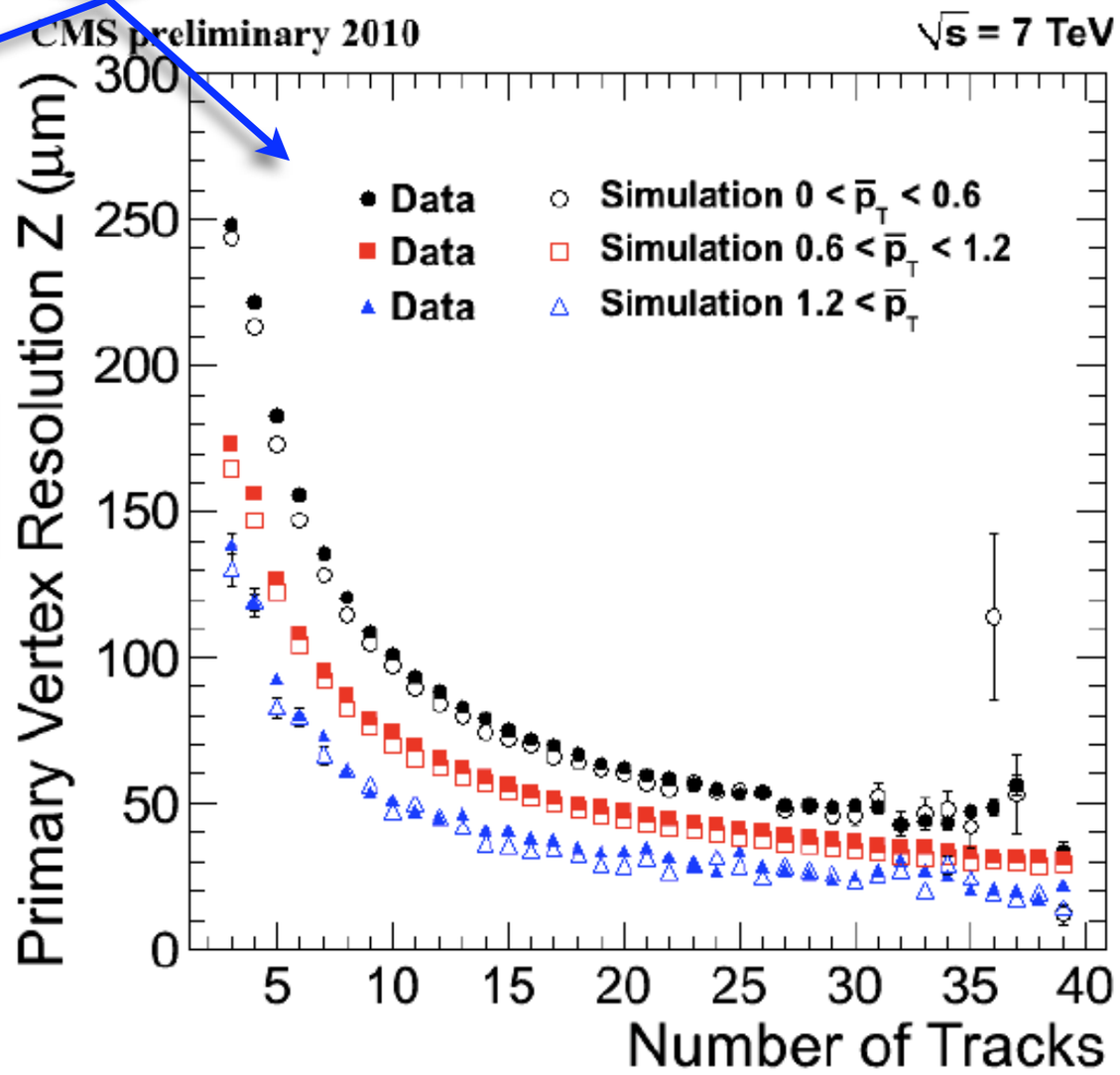


Primary vertex is split into two nearly equally populated sub-sets

The position of **one Vertex** is compared to the position of the **other Vertex**

$$\sigma(x_1 - x_2) = \sqrt{2} \cdot \sigma_{vtx}$$

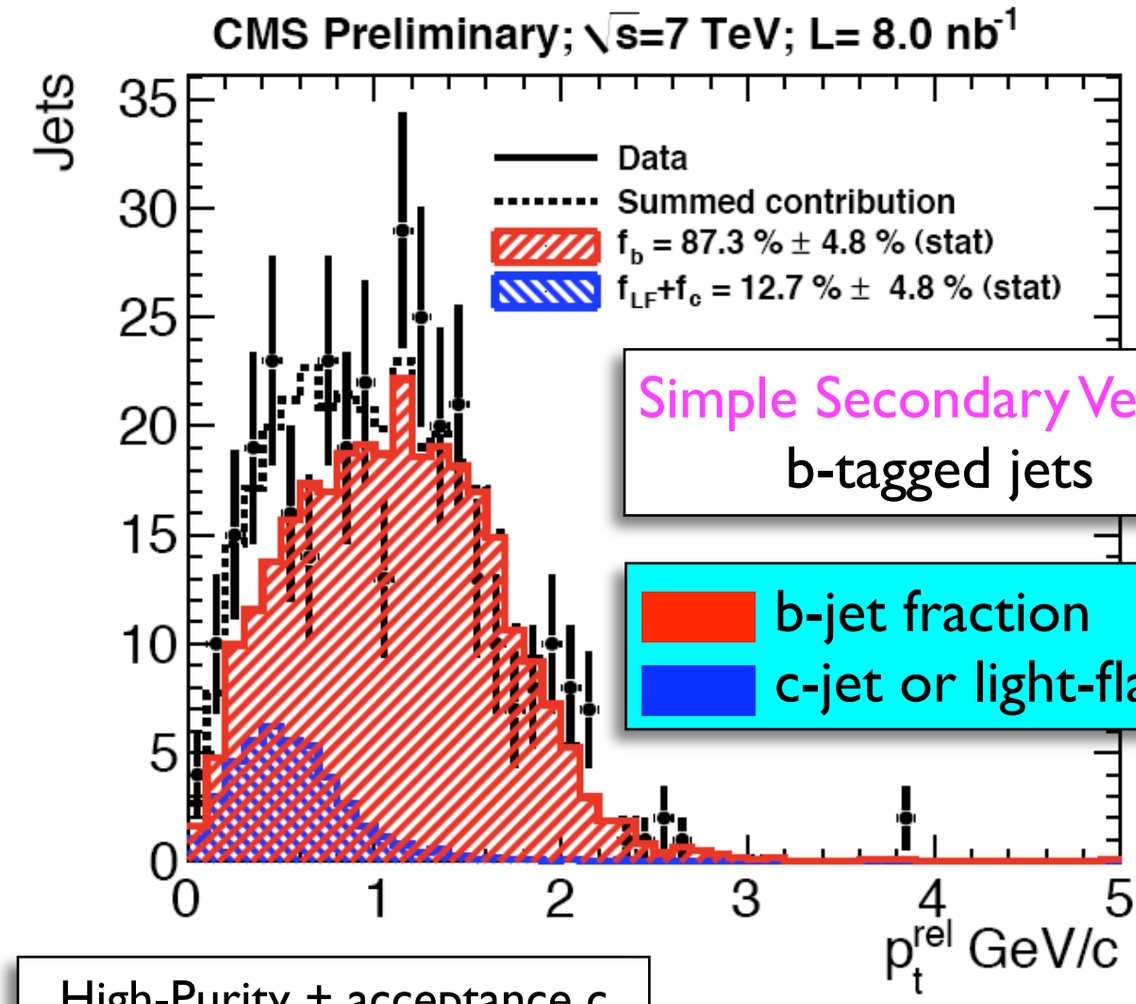


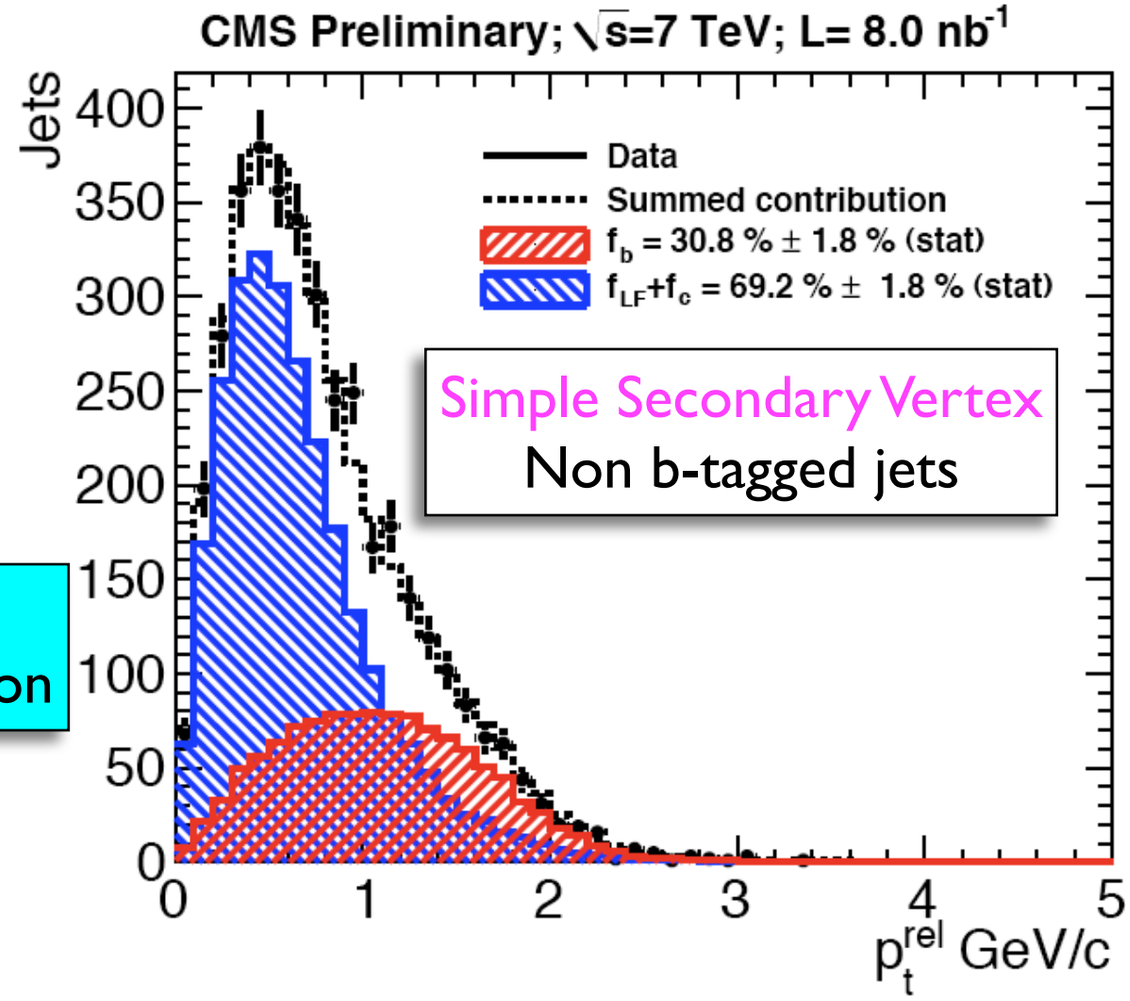


X and Z resolutions very similar
Good agreement between Data and MC

b-Jet Identification Efficiency

- b-jet identification efficiency is estimated from Data by fitting the p_T^{rel} (p_T with respect to the jet axes) distribution of muons in the semi-leptonic jets (p_T > 20 GeV/c; $|\eta|$ < 2.4)
- b-jet fraction is extracted from the fit using distribution templates based on Monte Carlo





High-Purity + acceptance c and light-flavored jets ≤ 0.1%

- $|\eta| \leq 2$
- $\langle p_T \rangle$ = 31GeV/c

Algorithm	ε_b(Data)	ε_b(MC)	ε_b(Data) / ε_b(MC)
Simple Secondary Vertex	0.203±0.015	0.207±0.002	0.98±0.08±0.18
Track Counting	0.233±0.014	0.244±0.002	0.95±0.06±0.19

The Pixel Stand-Alone Tracking

Up to now I presented the performance of CMS tracking system as a whole

On the other hand, the **pixel detector on itself is already able to provide very good information on tracks and vertices which are extremely useful to elaborate a fast high-level trigger (b-tagging, τ-reconstruction, etc...)**, indeed pixel stand-alone tracking is simpler and faster than the general tracking

Preliminary results on the studies of the pixel stand-alone tracking and vertexing **are very promising**, for now I'll present the performance of the Beam-Spot monitor application which is entirely based on the pixel information

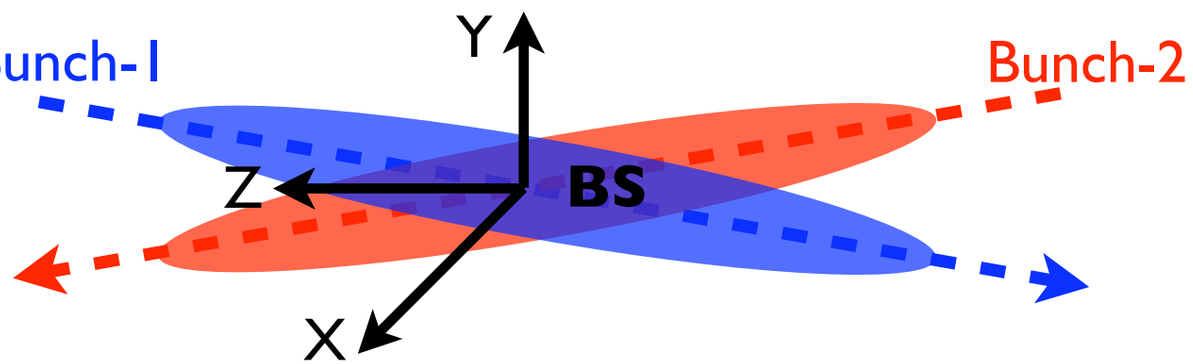
Pixel tracks are made from triplets of pixel hits:

- The track parameters are obtained: by a fast circle fit with the conformal mapping method and by solving analytically the straight line equation in the Z-azimuthal angle plane
- Pixel vertices are made of pixel tracks using the adaptive Kalman filter algorithm

Beam-Spot (BS) monitor method: log-likelihood estimator based on the unbinned 3D Gaussian fit to the pixel vertex positions

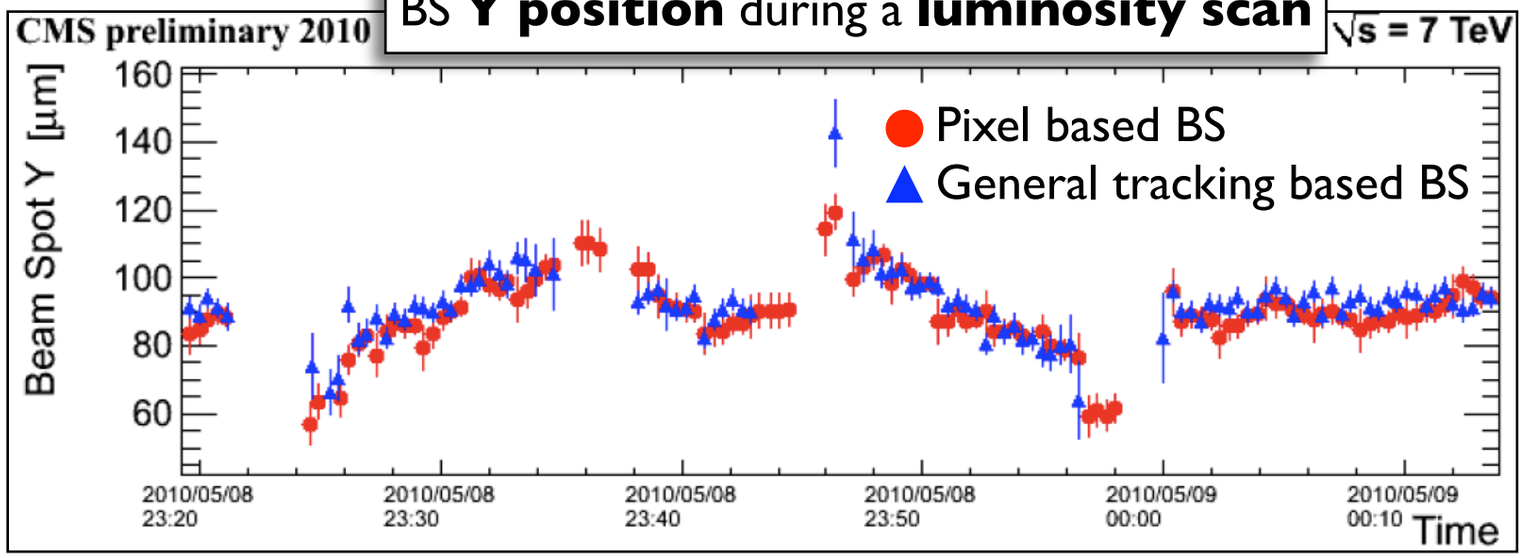
BS measured quantities:

- 3 coordinate position
- 3 widths
- 2 tilt angles *(in the X-Z and Y-Z planes)*

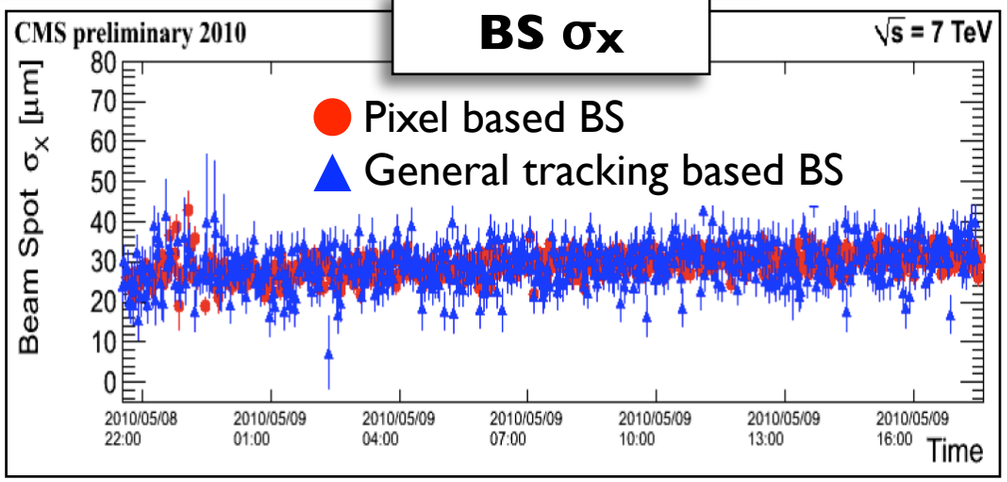


An Application of the Pixel Stand-Alone Tracking: **Beam-Spot Monitor**

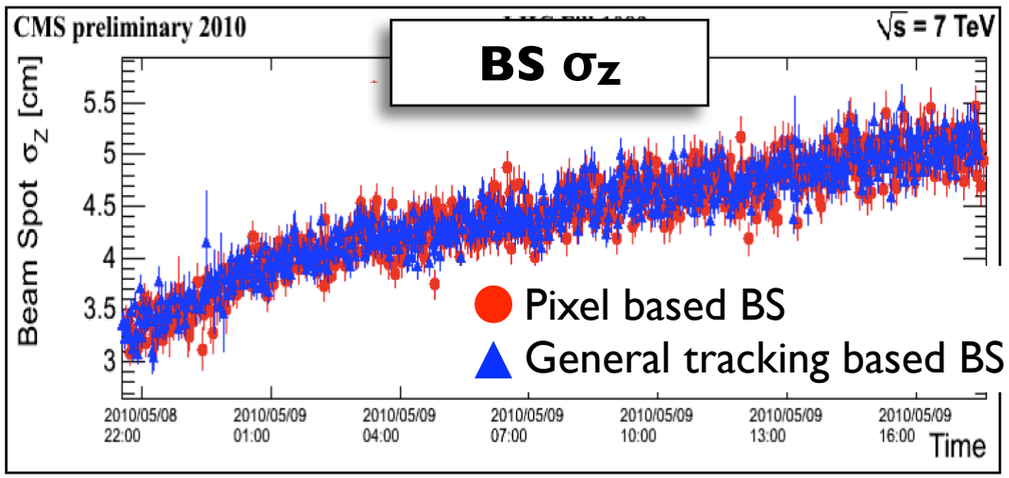
BS **Y position** during a **luminosity scan**



BS σx



BS σz



Comparison between the pixel based and the general tracking based methods: both give consistent results and are able to track well the movements of the beam

Resolutions (~200 vertices):

- X and Y position error: ~4 μm
- Z position error: ~3 mm
- X and Y width error: ~6 μm
- Z width error: ~2 mm
- tilt angles error: ~10^{-4} rad

Timing performance (min.bias √s = 2.3 TeV):

- High-Level Trigger menu without BS: ⟨Time⟩ = 23.62 ms
- pixel stand-alone BS monitor adds just ~0.54 ms *(in the worst case scenario where all the events are L1 confirmed it adds ~3.61 ms)*

Pixel-based BS monitor could be used at HLT

Conclusions

- After collecting about 100 nb^{-1}, we have a good understanding of tracking efficiency, momentum and impact parameter resolutions and vertex reconstruction performance

- The performance of b-jet identification has been analyzed on data and compared to simulation

- The great b-jet identification performance is strictly related to the great pixel detector tracking capabilities, which are exploited in many other contexts: seeding, Beam-Spot monitor, etc...

See related talks:

- *"CMS pixel detector status", G.Bolla*
- *"Calibration, Operation and Performance of the CMS Pixel Detector", B.Kreis*
- *"Offline Calibrations and Performance of the CMS Pixel Detector, U.Langenegger*
- *"Impact of beam induced backgrounds for the CMS Pixel and other inner radii detectors", S.Mueller*
- *"The Alignment of the CMS Silicon Strip and Pixel Tracker", F.Meier*

- As the integrated luminosity collected by CMS increases, tracking performance is estimated from data with increasing detail

References

Most recent Physics Analysis Summaries (PAS) on tracking, vertexing and b-tagging performance:

•CMS Collaboration, "Tracking and Vertexing Results from First Collisions," CMS-PAS-TRK-10-001

•CMS Collaboration, "Measurement of Tracking Efficiency," CMS-PAS-TRK-10-002

•CMS Collaboration, "Measurement of momentum scale and resolution using low-mass resonances and cosmic ray muons," CMS-PAS-TRK-10-004

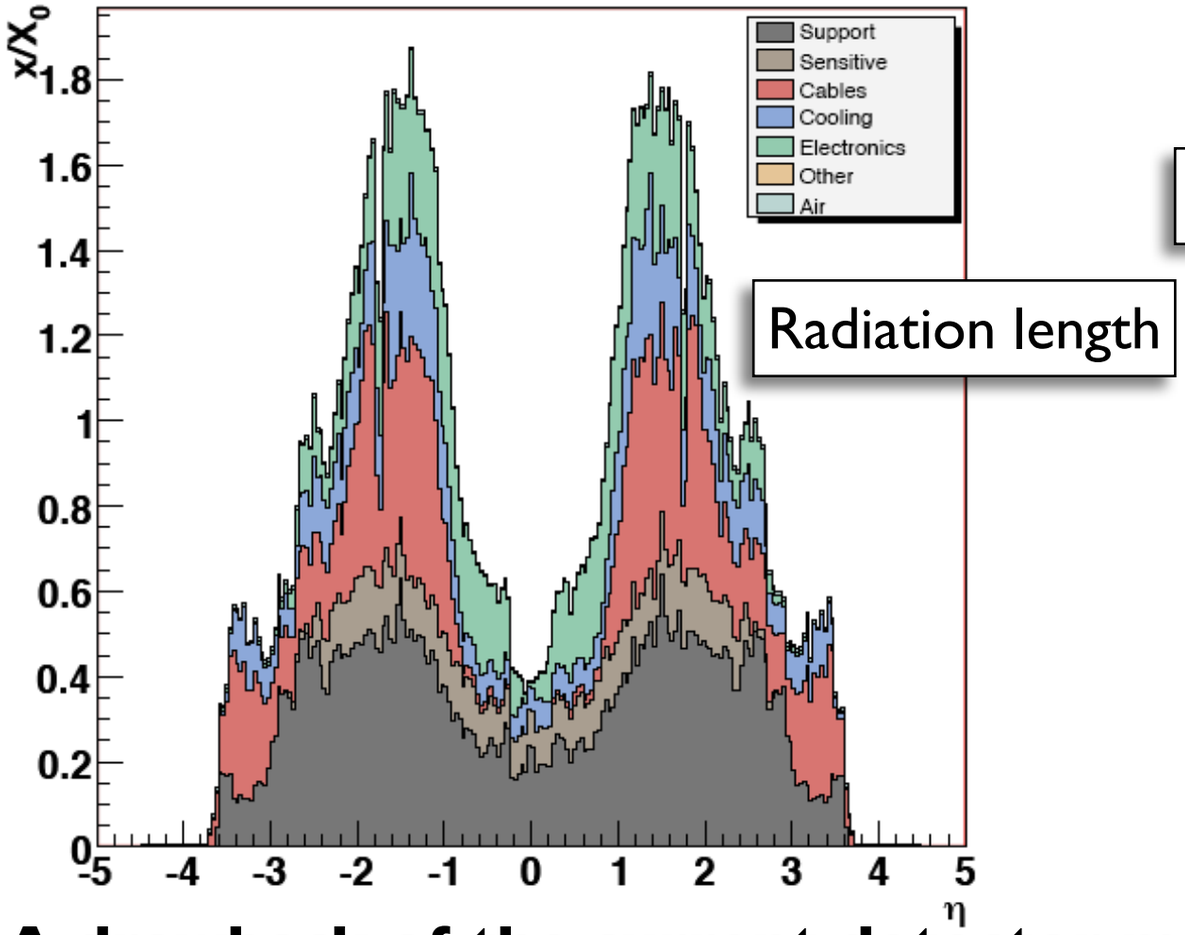
•CMS Collaboration, "Tracking and Primary Vertex Results in First 7 TeV Collisions," CMS-PAS-TRK-10-005

•CMS Collaboration, "Commissioning of b-jet identification with pp collisions at sqrt(s) = 7 TeV," CMS-PAS-BTV-10-001

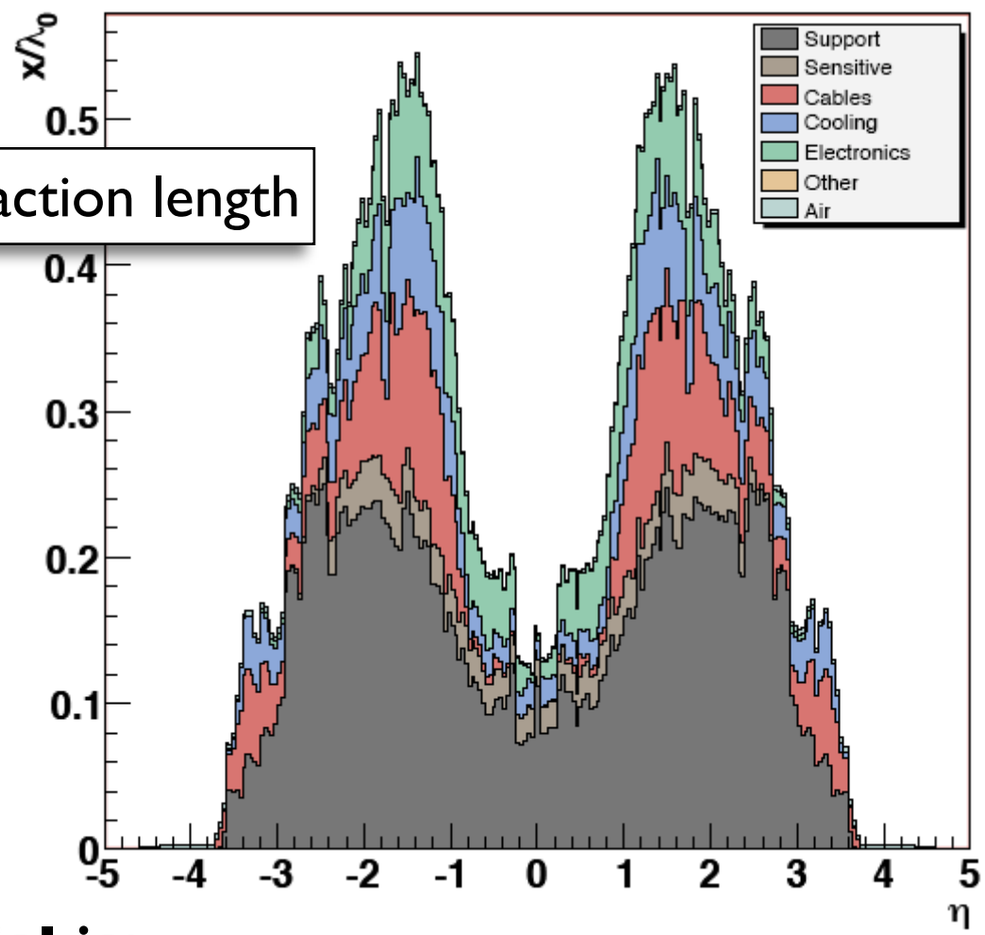
Backup Slides

The CMS Tracking Algorithm

The CMS tracking detector allows for a full reconstruction of events up to the LHC design luminosity: 10^{34} cm^{-2} s^{-1} (i.e. events with ~1000 charged tracks)



Radiation length



Interaction length

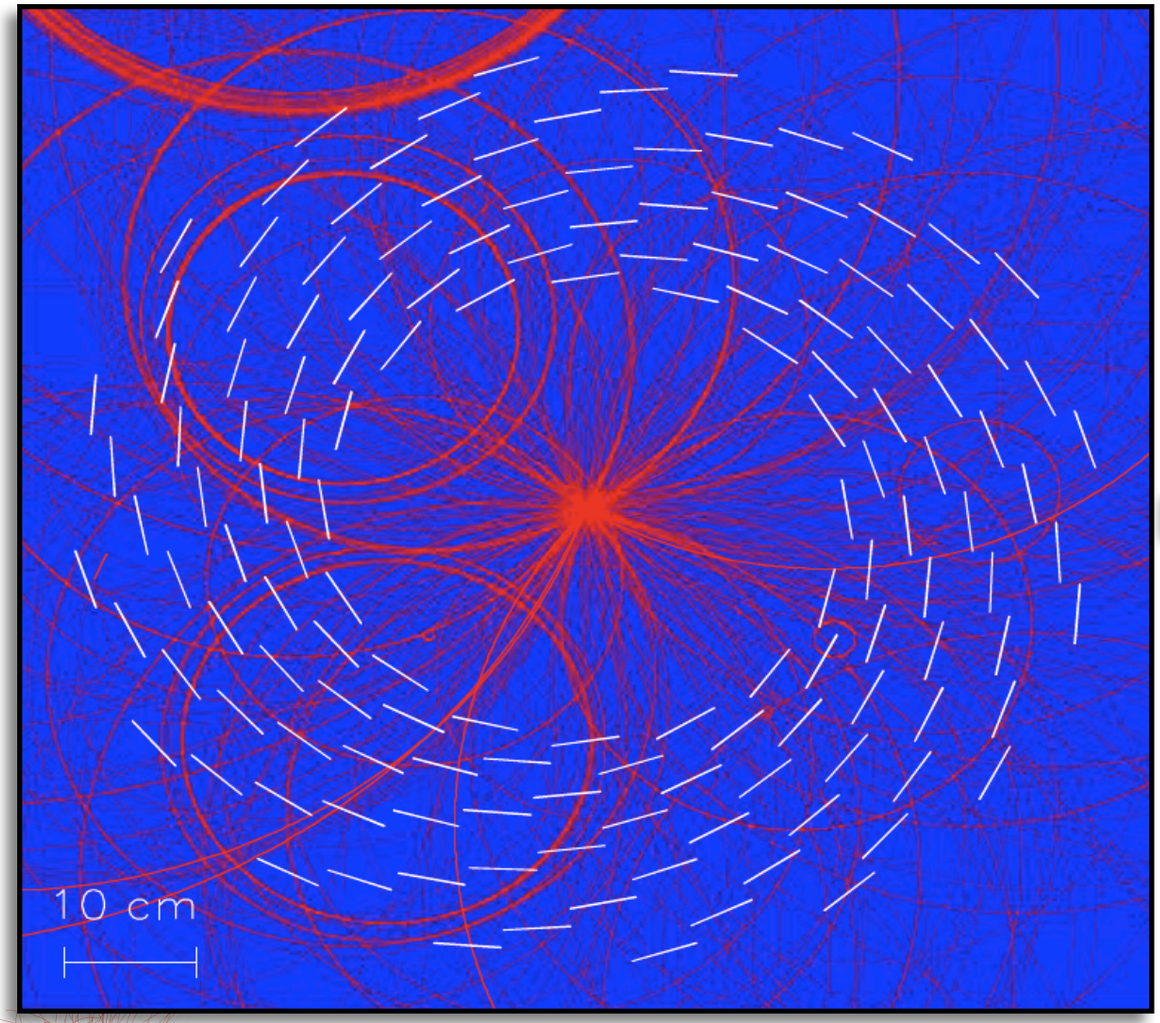
A drawback of the current detector material is:

- Electrons lose energy from hard Bremsstrahlung radiation
- Charged hadrons suffer elastic and inelastic nuclear interactions with the tracker material: up to 10% of charged pions experience destructive inelastic interactions before crossing the minimum number of sensitive layers necessary to measure the curvature

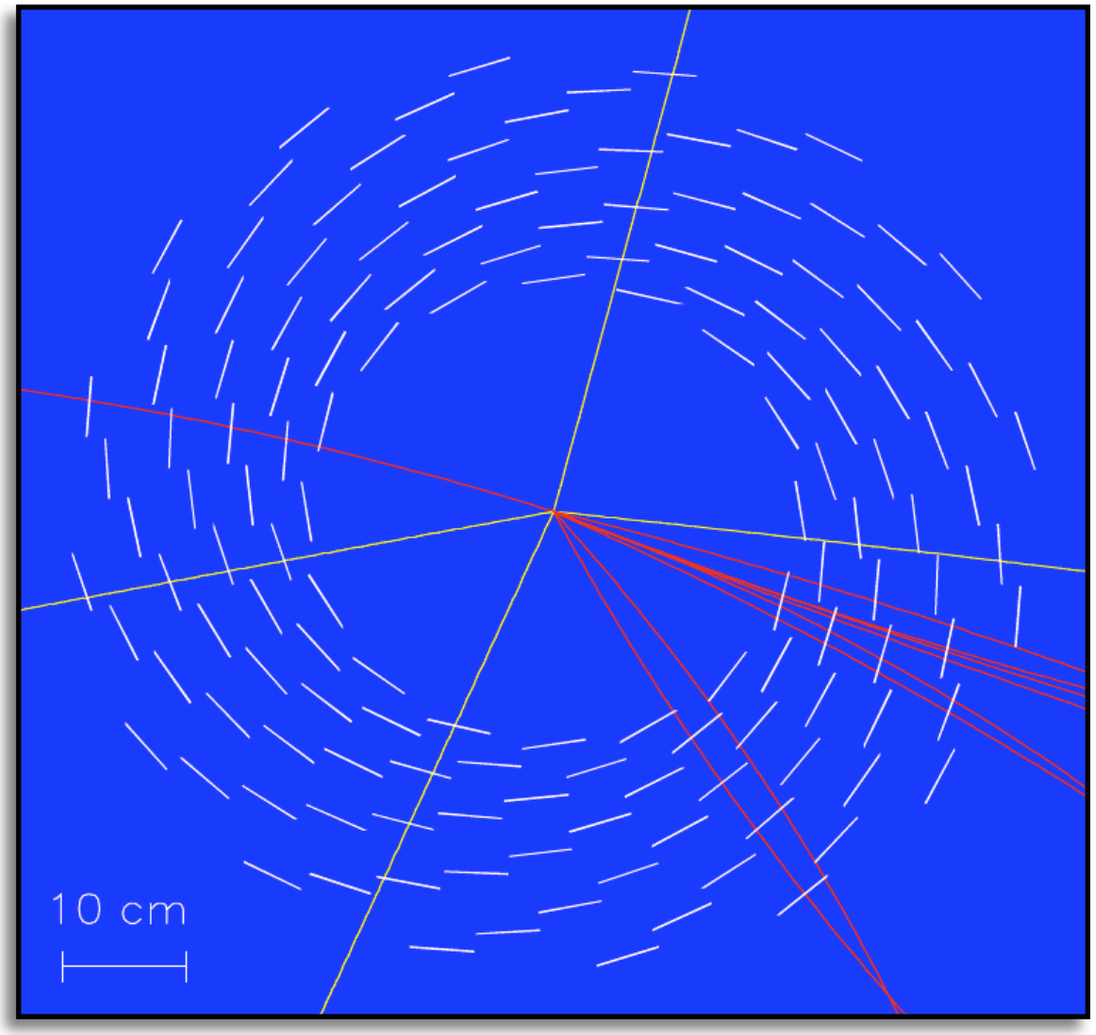
Very challenging task for pattern recognition and track reconstruction !

The CMS Tracking Algorithm

$H \rightarrow 4\mu$ dispersed in 1000 charged tracks at LHC design luminosity 10^{34} cm^{-2} s^{-1}



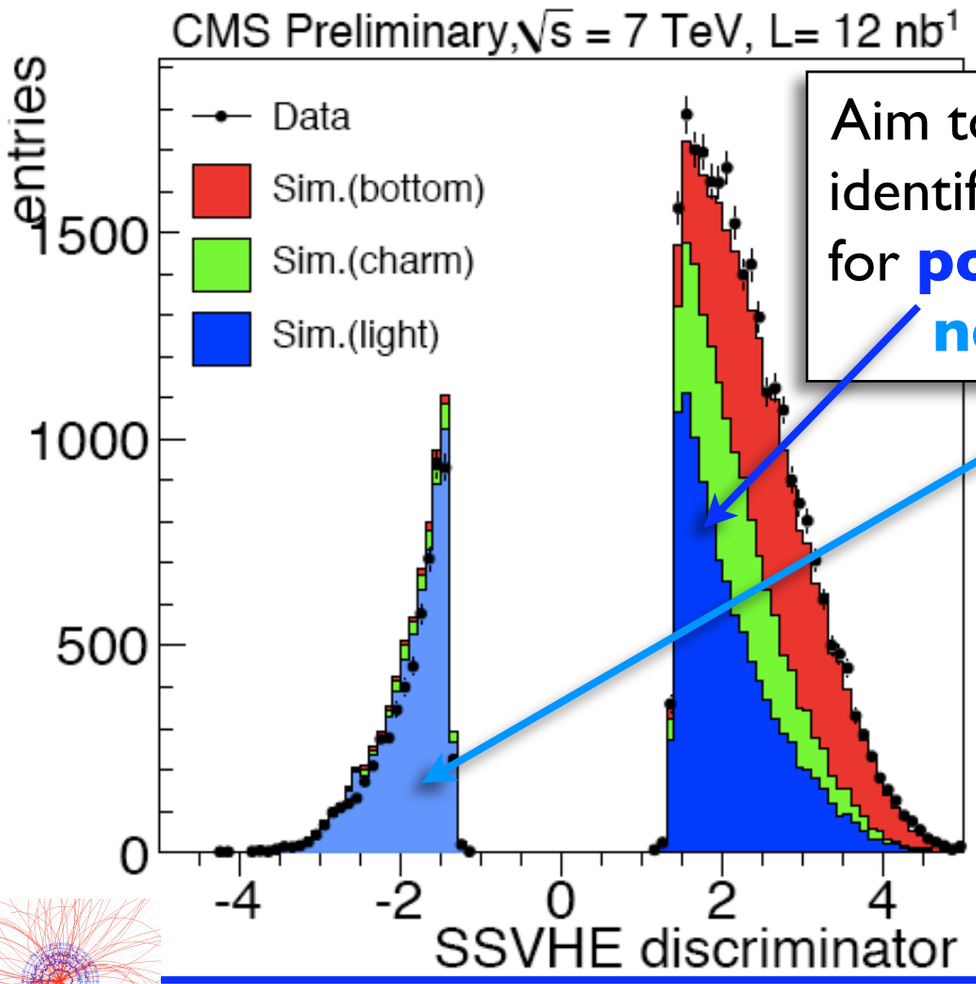
Same event requiring $p_t > 2$ GeV/c



b-Jet Mis-Identification Rate From Negative Tags

Track Counting algo.: requires 2 or 3 tracks with an impact parameter significance larger than a given cut. Negative tagging: the 2 or 3 tracks have a negative scalar product between the impact parameter and the jet axes

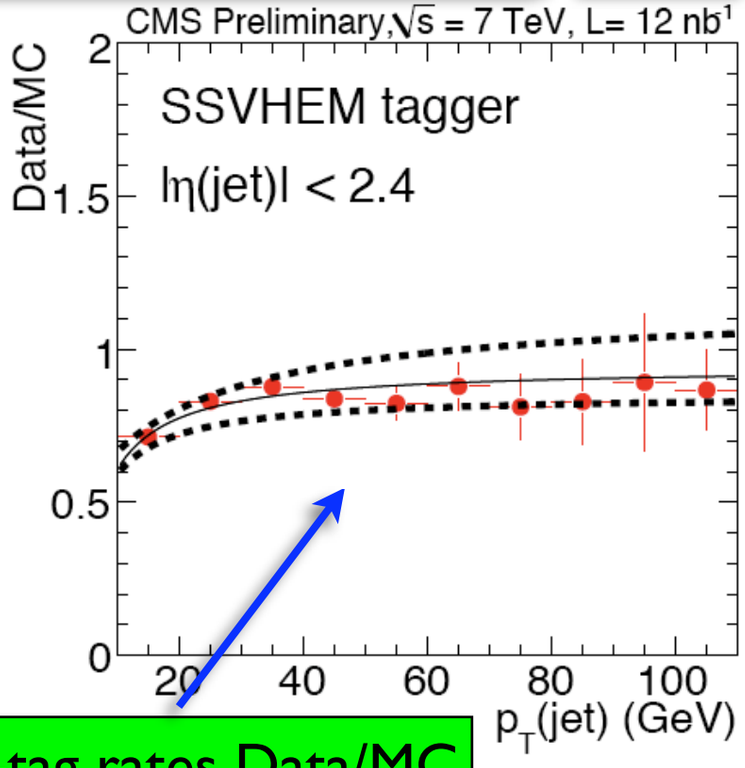
Simple Secondary Vertex algo.: requires a significance of the secondary vertex decay length larger than a given cut. Negative tagging: the secondary vertex is in the opposite direction with respect to the jet



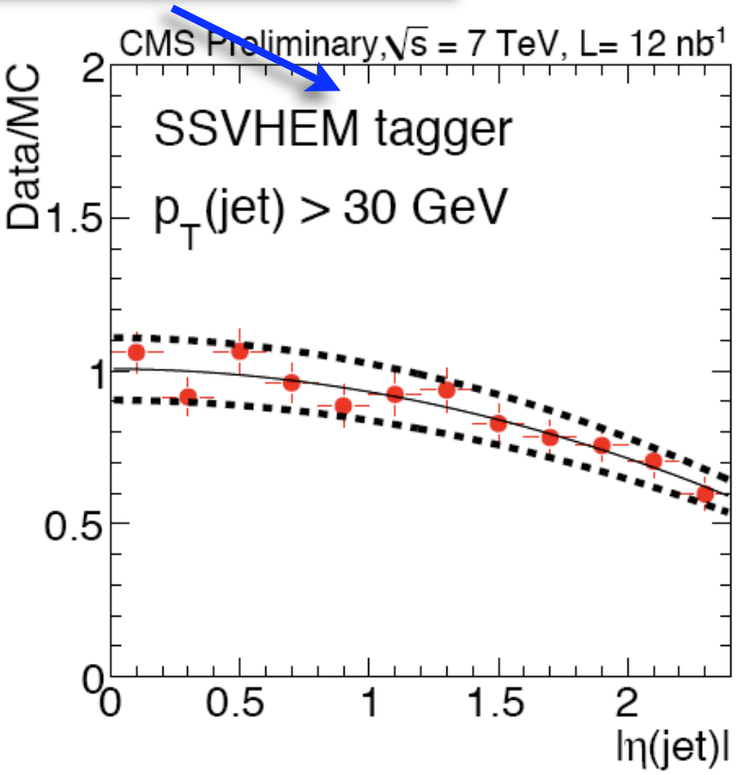
Aim to estimate the mis-identification distribution for **positive tags** using **negative tags**

$$\varepsilon_{data}^{mistag} = \varepsilon_{data}^{-} \frac{\varepsilon_{MC}^{mistag}}{\varepsilon_{MC}^{-}}$$

Corrects for asymmetry between positive and negative tags



Mis-tag rates Data/MC



The CMS Primary-Vertexing Algorithm

The primary-vertexing algorithm is based on an adaptive Kalman filter:

1. Find a coarse approximation of the vertex position
   - (i) For each pair of tracks compute the "crossing point" (i.e. mean of two points of closest approach of two tracks)
   - (ii) Assign a weight to each crossing point proportional to the distance of the two tracks
   - (iii) Find the mode of the crossing points

2. Weight the tracks according to their standardized (χ^2) distance to the vertex:

$$w(\chi^2) = \frac{1}{1 + e^{-\frac{\chi^2_{\text{cutoff}} - \chi^2}{2T}}}$$

3. Apply the Kalman filter algorithm (update the vertex position track-by-track)

4. Restart from (2) with a smaller parameter *T*

The algorithm stops either when the difference between the new computed transverse vertex position and the previous one is "small", or when a maximum number of iterations is reached

Linearization Point Finder

Linearization Pint Finder algorithm finds a coarse approximation of the vertex position:

1.For each pair of tracks compute the "crossing point" (i.e. mean of two points of closest approach of two tracks, or in other words it's the pair of points of two tracks which have the smallest distance between each other; the crossing points are computer for each pair of tracks)

2.Assign a weight to each crossing point proportionally to the distance of the two tracks

3.Find the mode of the crossing points in each of the three spatial coordinates separately (in one dimension it finds the shortest interval containing points with a sum of weights exceeding a fixed fraction of the sum of all weights (0.4 by default); the procedure is then repeated on the found interval, until at most two points remain; the mode is finally the average of the remaining points)