The ZEUS Global Tracking Trigger Barrel Algorithm

Mark Sutton
University College London

30th September 2004

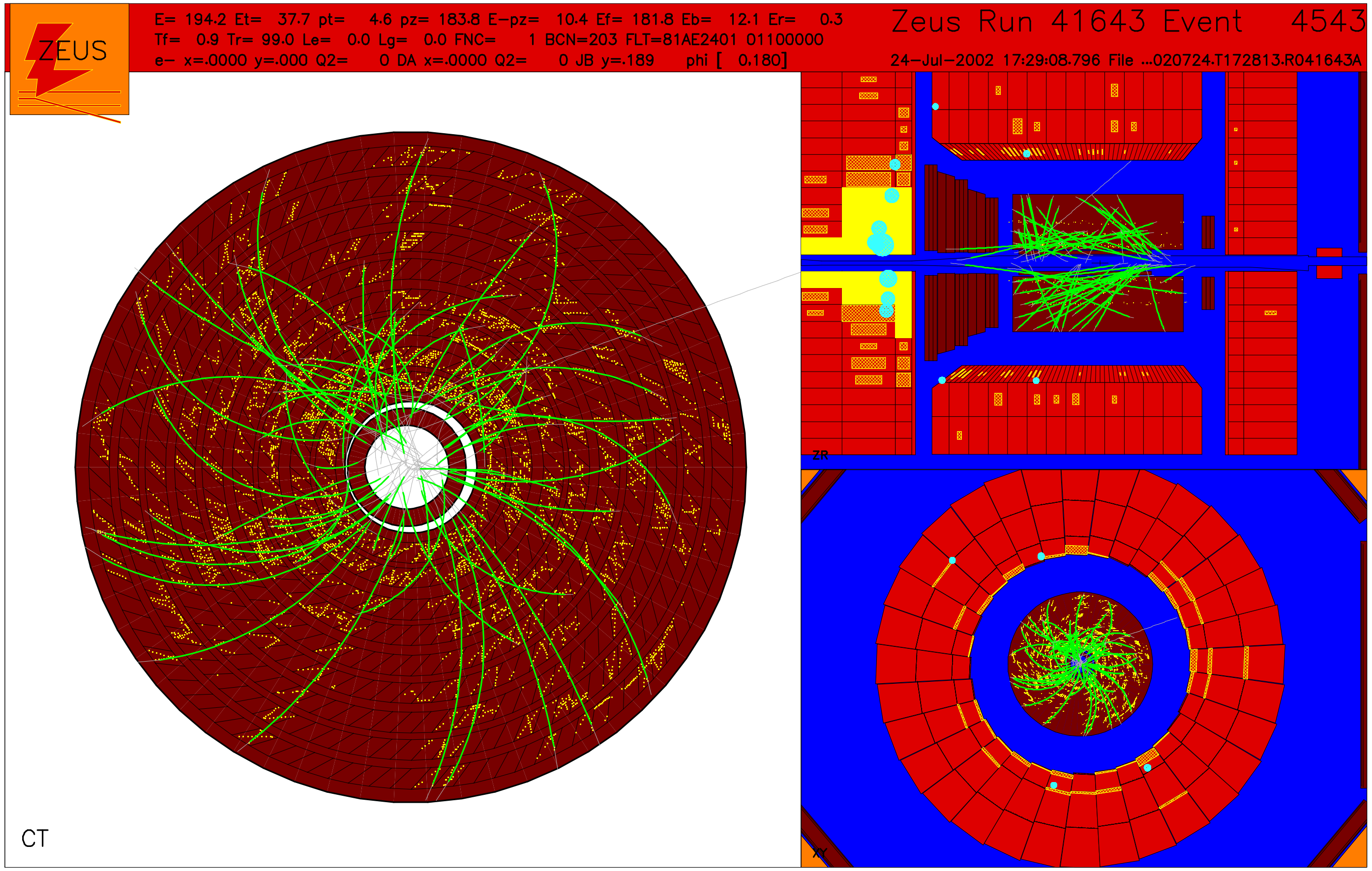
On behalf of the ZEUS GTT group:

DESY-Hamburg, University of Glasgow, MEPHI, University of Oxford
University College London, Yale

Introduction to the GTT

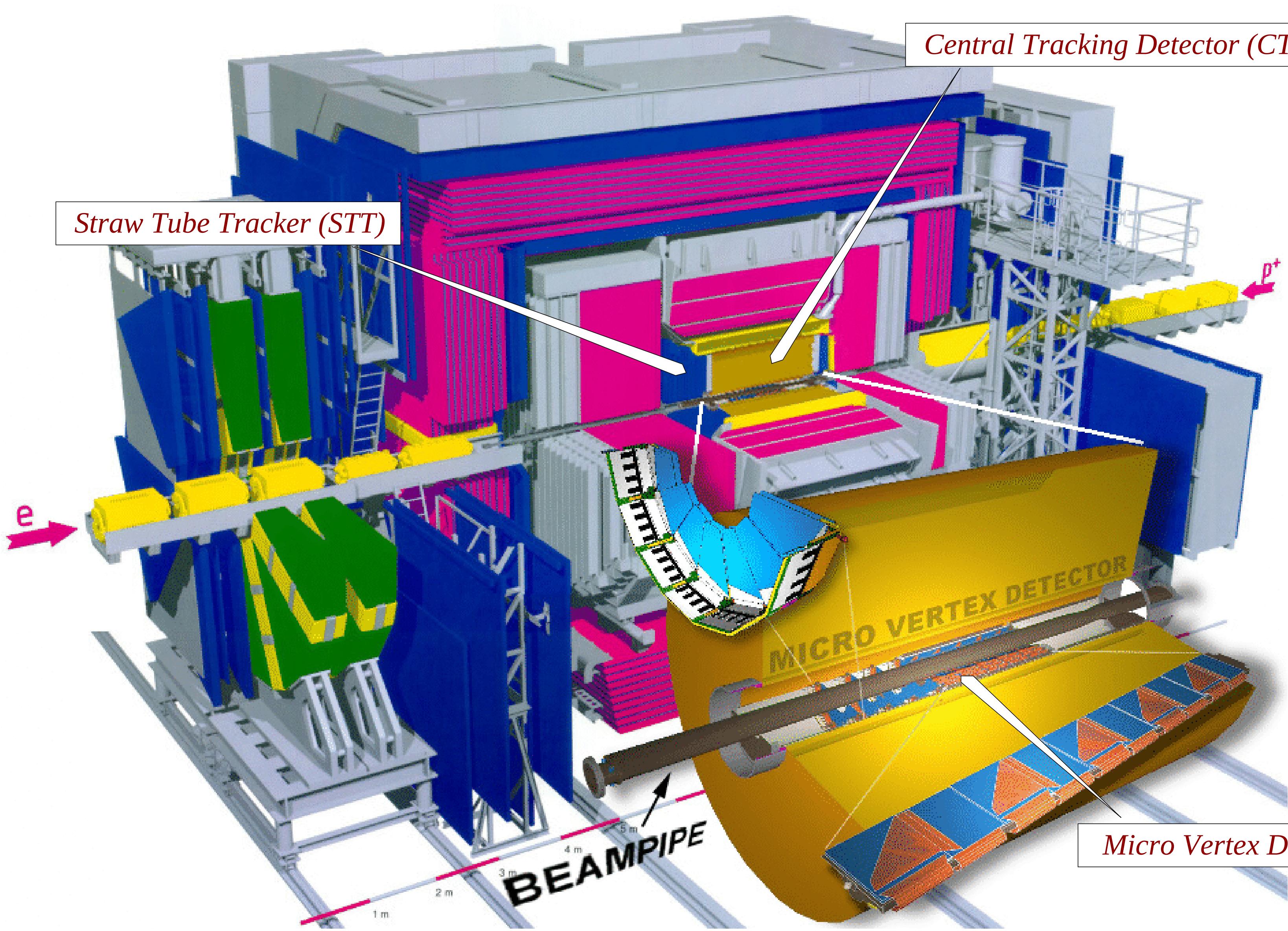
- HERA luminosity Upgrade ⇛ ZEUS added MVD and forward STT.
- Before upgrade, at Second Level Trigger, tracking only with CTD
  - ▷ Vertex resolution 9cm.
  - ▷ Transputer technology more than 10 years old, not easily adaptable.
- Must deliver trigger result within ∼ 15 ms.
- Improved CPU and network technology make better, faster tracking possible,
  ⇛ combined tracking with CTD, MVD and STT data
  - ▷ Extend to forward region.
  - ▷ Improve vertex resolution, ⇛ improved event selection and heavy flavour tagging online in the Second Level Trigger.

Tracking Conditions at the Second Level Trigger Stage

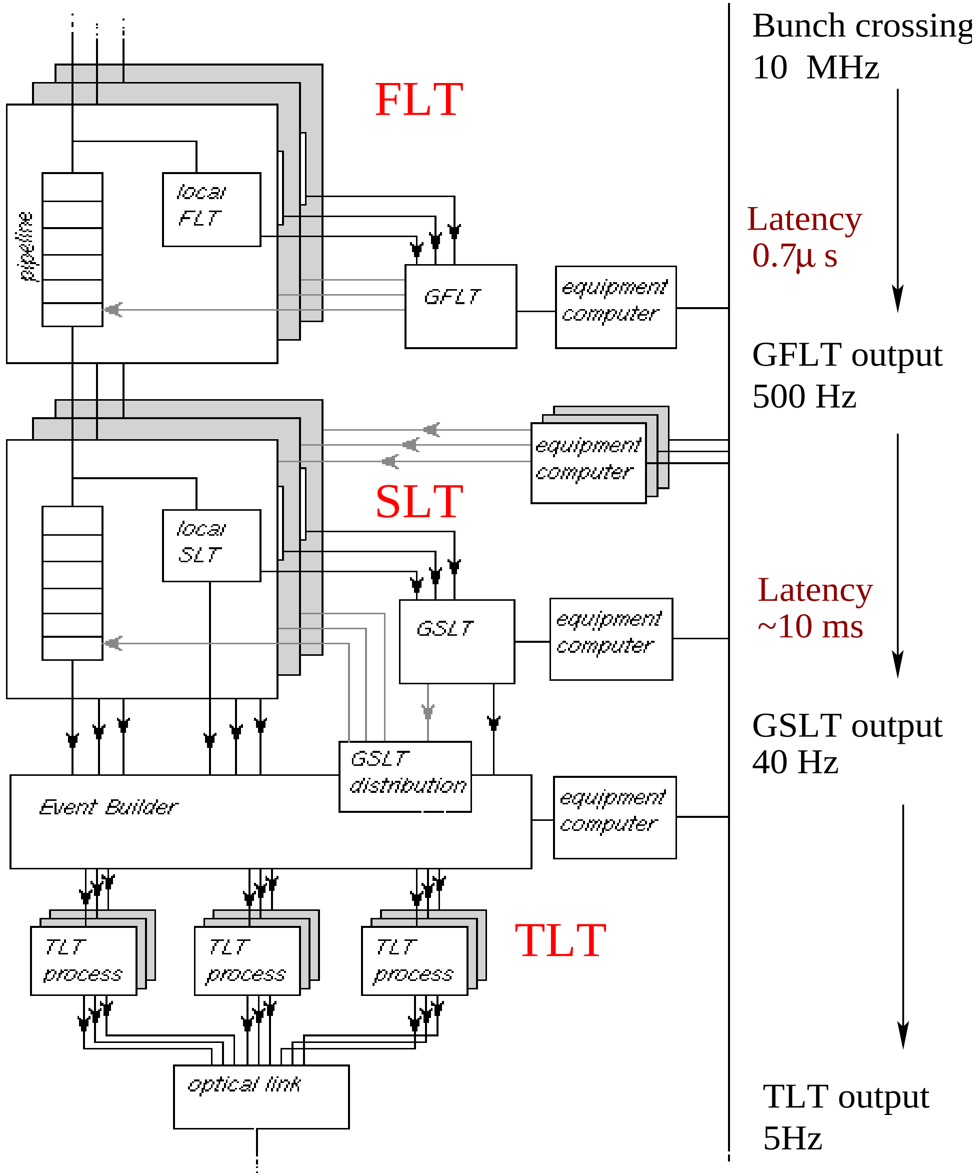


- Very hostile environment for detailed tracking ⇒ limited available processing time, large contribution from high rate, high occupancy beam gas events.
- Data readout/transfer and event processing must be complete within $\sim$ 15 ms.
- Don't have space points, must reconstruct complete event from raw detector data.

The ZEUS Detector



The ZEUS Trigger



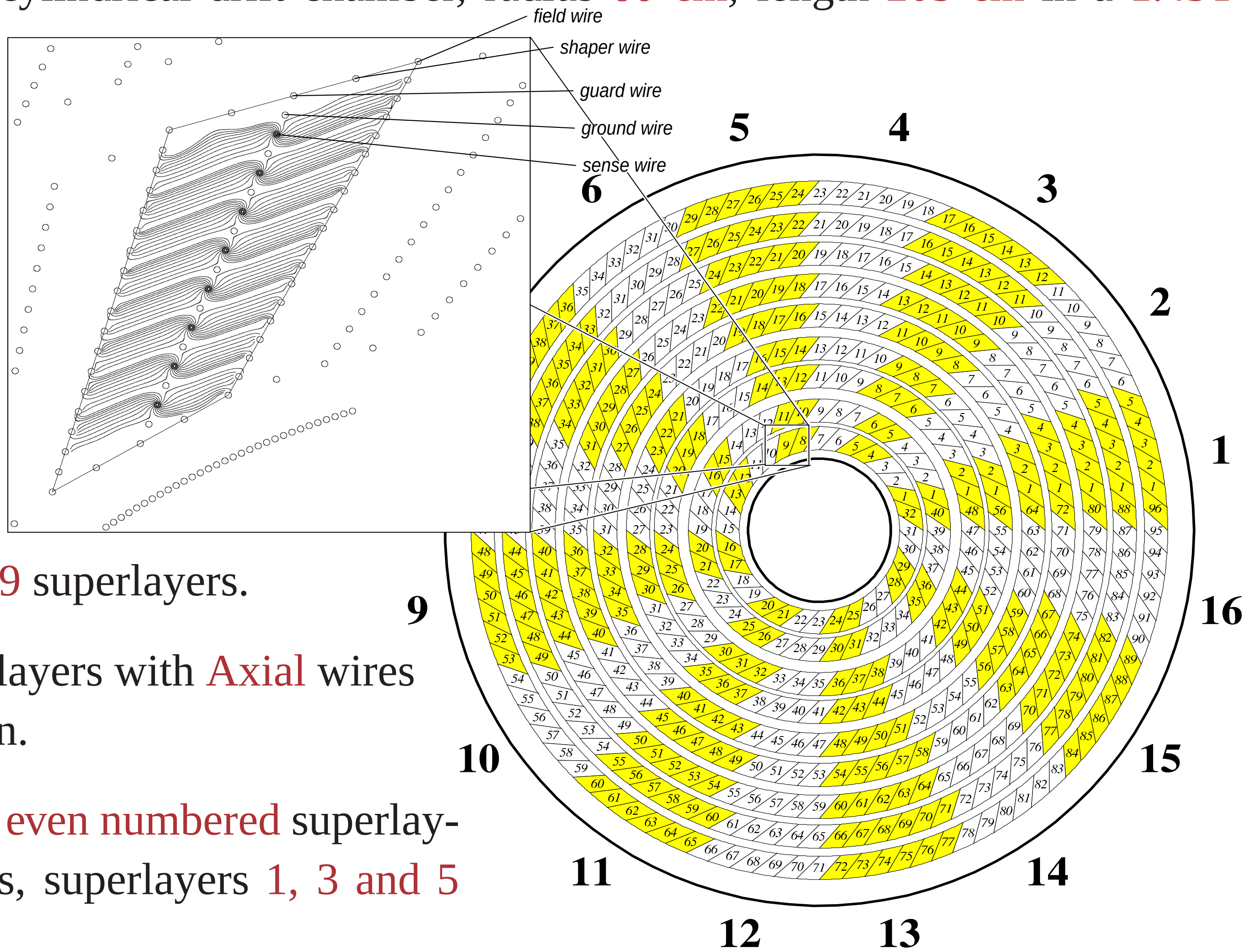
- Flexible three level trigger.
- Built in 1992 ⇒ First High Rate, pipelined trigger – 96 ns bunch crossing.
- First Level Trigger (FLT) dedicated pipelined hardware trigger, deadtime free.
- Second Level Trigger (SLT) based on INMOS transputers.

Overview of Algorithm Design Considerations

- Modular Algorithm Design
  - ▷ Multi threaded event processing.
  - ▷ Two concurrent algorithms
    ➡ Barrel and Forward.
  - ▷ Complete processing of one event per Algorithm processor.
- Development and testing possible with full "playback" system
  - ▷ Data injected at component interfaces.
- Don't have spacepoints ➡ must fully process raw detector information before we can get an idea of the event topology.
- Multiple pattern recognition ambiguities, algorithm must be fast
  ➡ all ambiguities must be broken as soon as possible with the minimum of processing, no time to processes all combinations.
- All data structures and output from each stage of processing stored in lookup tables.

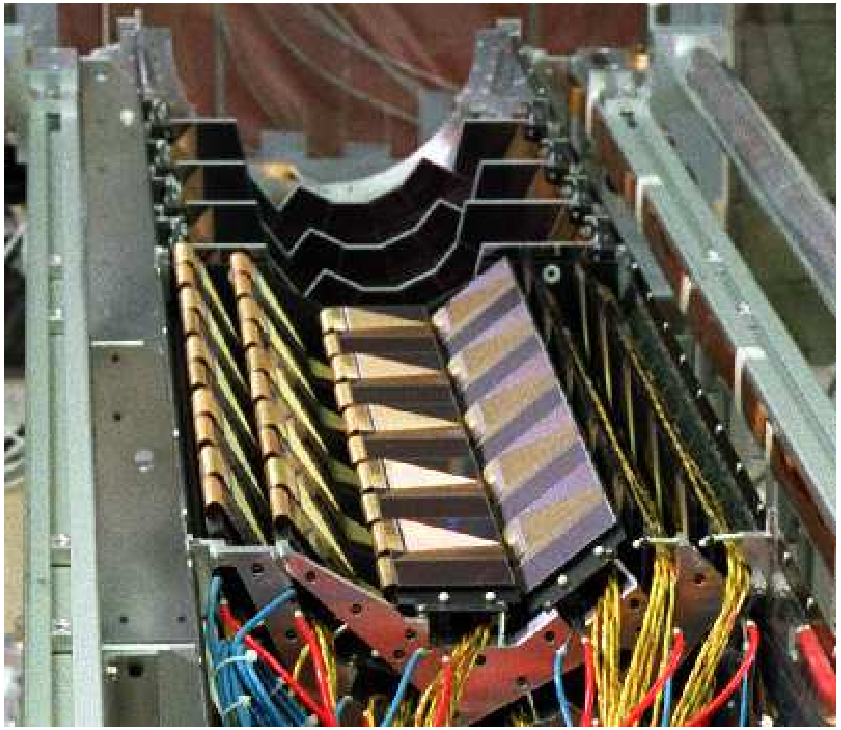
The ZEUS Central Tracking Detector

- The CTD is a large cylindrical drift chamber, radius 85 cm, length 205 cm in a 1.43T solenoidal field.

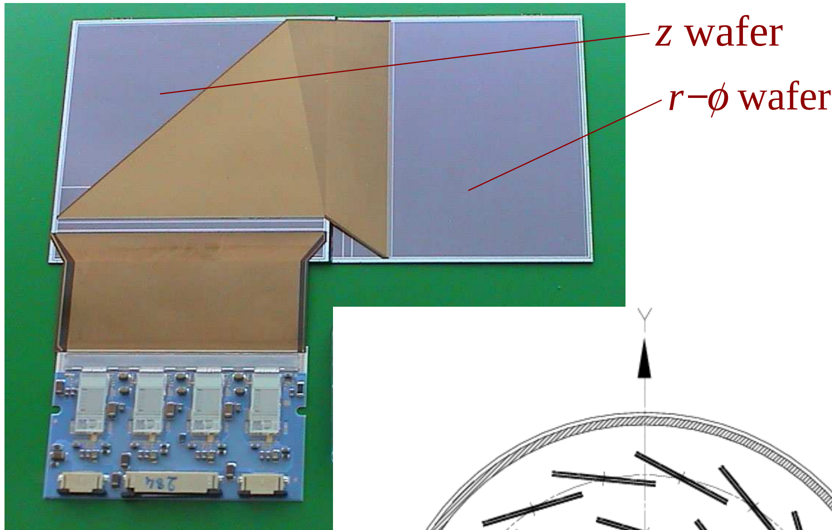


- 4608 Sense Wires in 9 superlayers.
- Odd numbered superlayers with Axial wires for r-ϕ reconstruction.
- For z reconstruction, even numbered superlayers have Stereo wires, superlayers 1, 3 and 5 have z-by-timing.
- Hit resolution, 200μm. Drift Cells at 45° to radii to aid pattern recognition.

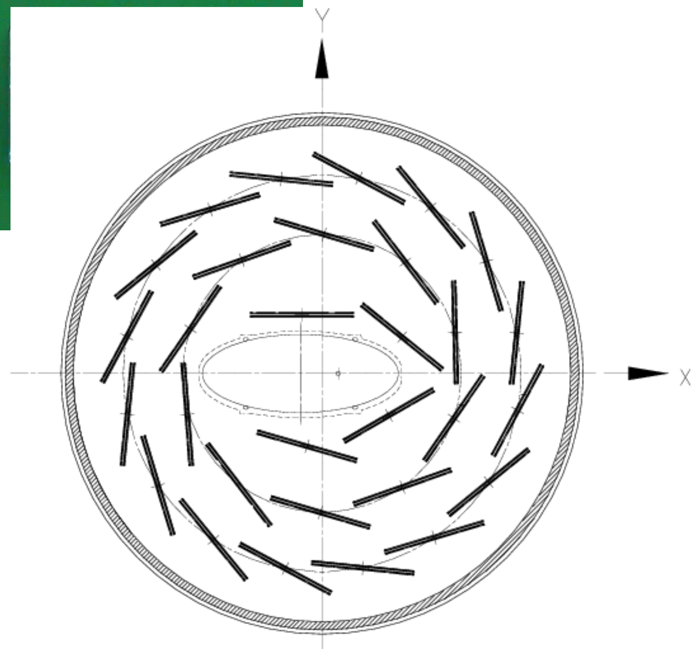
The ZEUS Micro-Vertex Detector



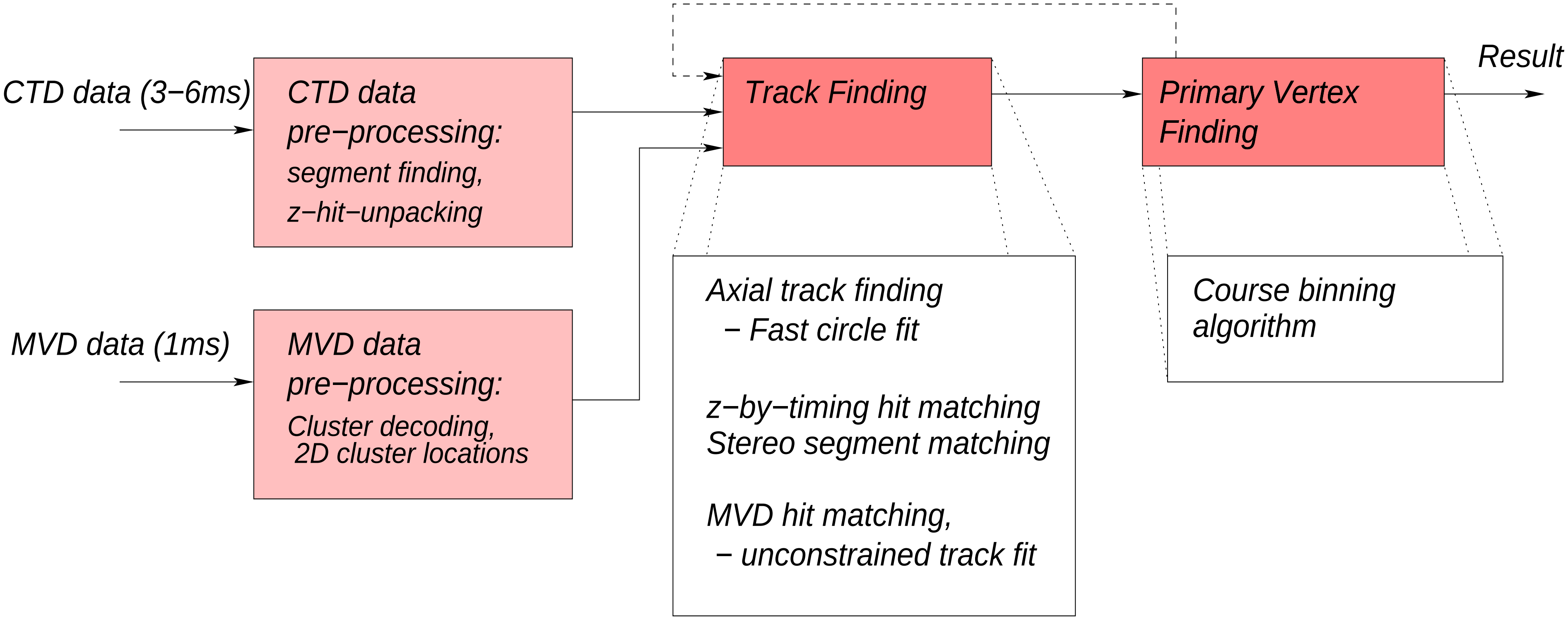
- MVD: 50μm hit resolution Si-diodes in 4 forward wheels and 3 barrel layers. Acceptance 7-170°.



- MVD Barrel: 5 modules per ladder.
- Each module, has two z-wafers and two r-ϕ-wafers. Pairs of z- and r-ϕ-wafers multiplexed.



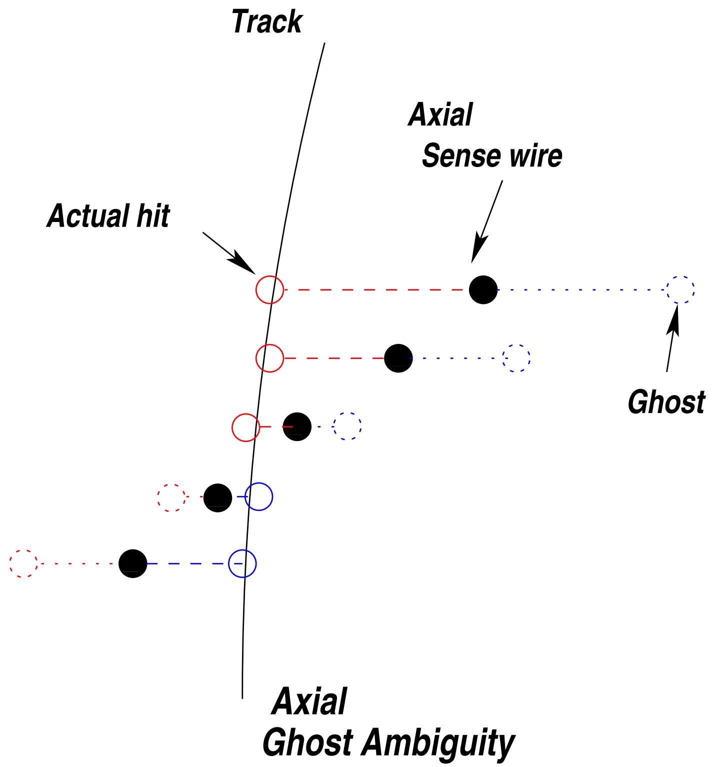
Barrel Algorithm Overview



- Fast, multi-stage algorithm...
  - ▷ Multi-threaded pre-processing stage.
  - ▷ Data decoded in place, stored in lookup tables for fast access.
- After primary vertex fit, perform track and vertex refit to increase efficiency: potentially slow ➡ reuse information already computed.

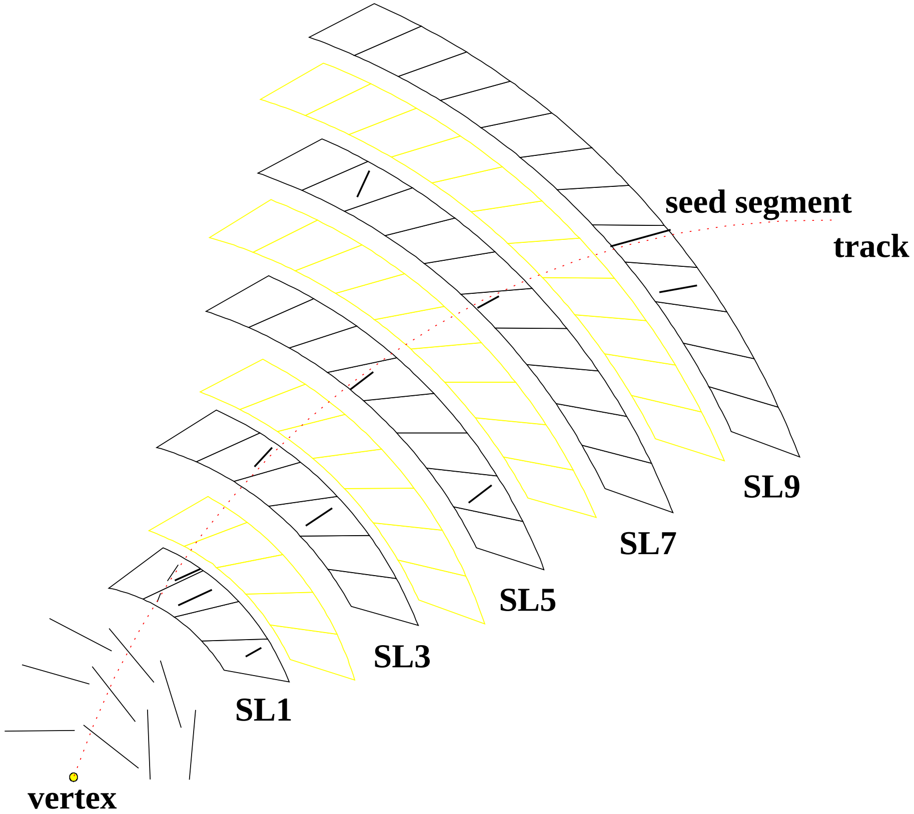
CTD Segment Finding

- Hit positions in the CTD given by drift time of charge to wire.
- Drift time scalar quantity ⇒ left-right ghost ambiguity.
- Look for linear segments of hits in local coordinate system of each cell.
- Cells oriented at 45° to the CTD radii.
  - ▷ Break ambiguity, taking segment candidate that points to the beam line.
- Fully process all CTD cells (Axial and Stereo), allow up to 4 segments per cell.



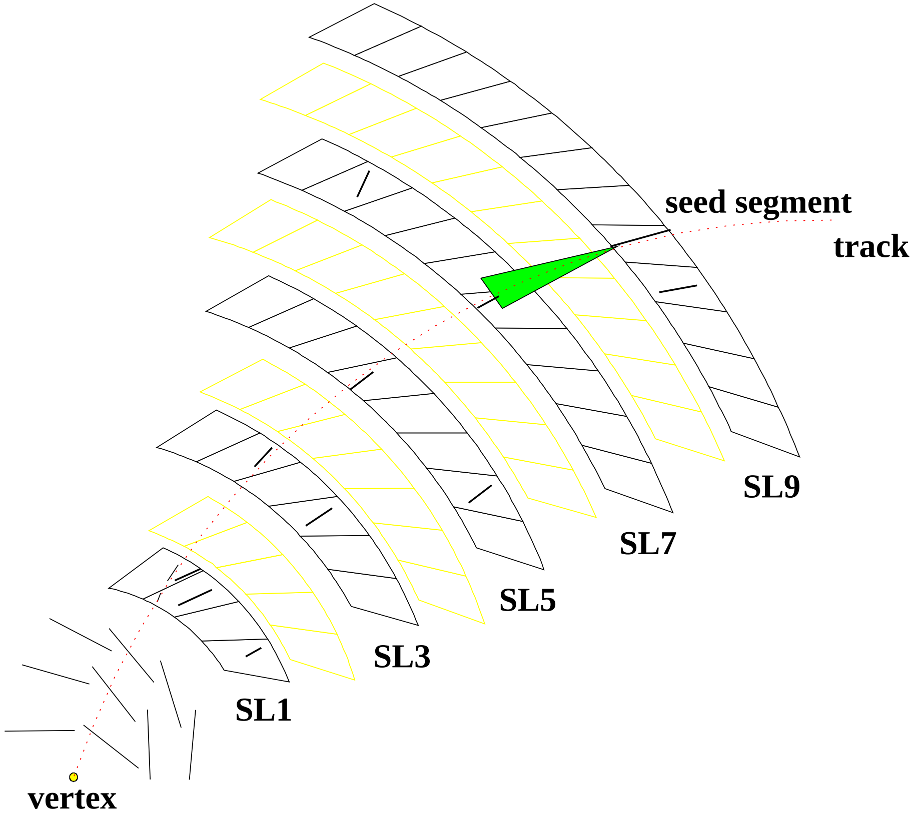
- Transform Axial segments to CTD coordinates (Only wire numbers and drift times needed for Stereo segments).
- Store found segments in lookup table.

Axial Track Finding



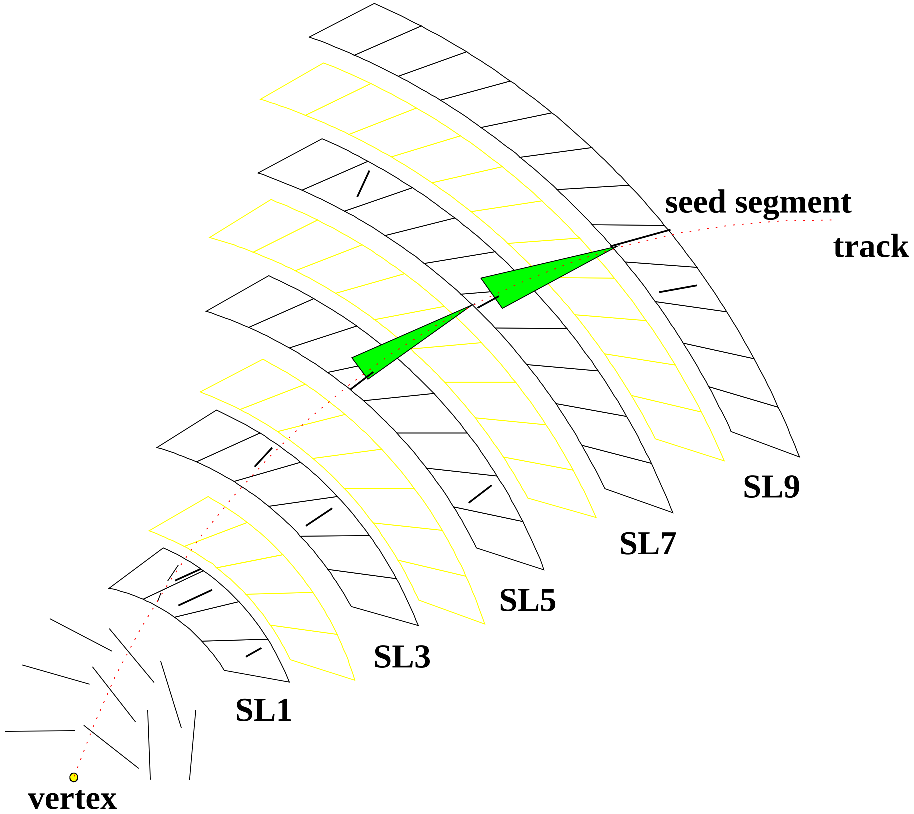
- Fit CTD track first...
  - ▷ Use segments found in Axial layers of CTD.
  - ▷ Seed segment from outer superlayer.
  - ▷ Use beam line constraint to match inner segments.
  - ▷ Fit track with fast circle fit for r-ϕ tracks.
- Match MVD r-ϕ hits to track.
  - ▷ Refit r-ϕ track including MVD r-ϕ hits which are given a larger weight in the fit.
- Constrained or unconstrained fit?
  - ▷ Pattern recognition better with constrained tracks.
  - ▷ Secondary vertices require unconstrained tracks ⇨ use unconstrained fit for tracks after MVD hits have been matched.

Axial Track Finding



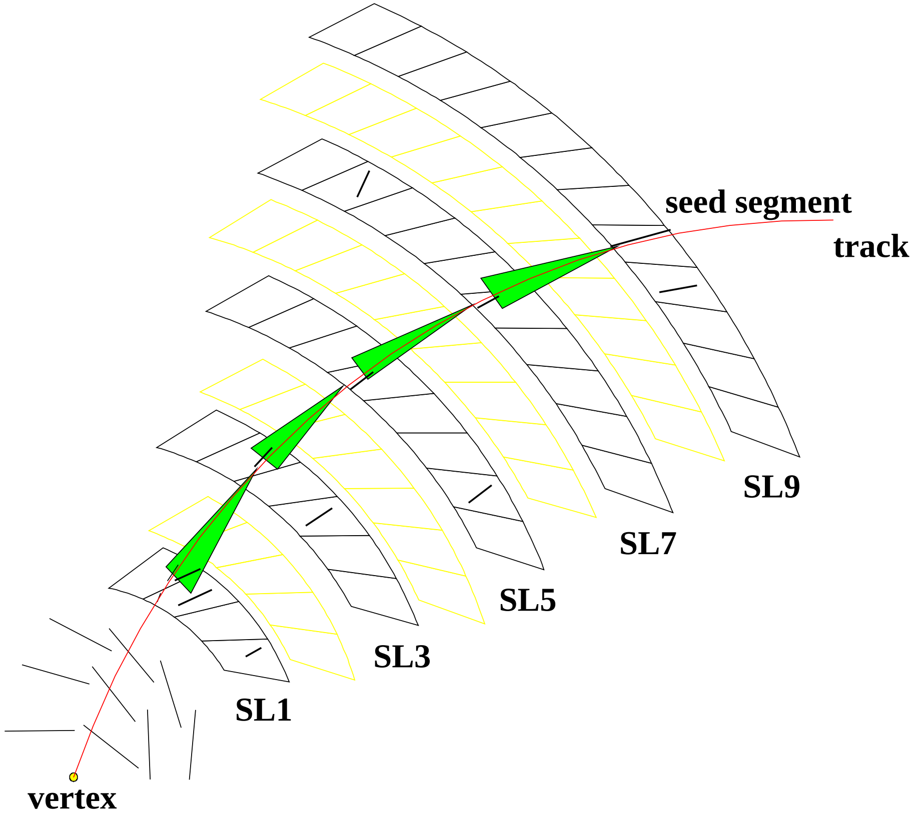
- Fit CTD track first...
  - ▷ Use segments found in Axial layers of CTD.
  - ▷ Seed segment from outer superlayer.
  - ▷ Use beam line constraint to match inner segments.
  - ▷ Fit track with fast circle fit for r-ϕ tracks.
- Match MVD r-ϕ hits to track.
  - ▷ Refit r-ϕ track including MVD r-ϕ hits which are given a larger weight in the fit.
- Constrained or unconstrained fit?
  - ▷ Pattern recognition better with constrained tracks.
  - ▷ Secondary vertices require unconstrained tracks ⇒ use unconstrained fit for tracks after MVD hits have been matched.

Axial Track Finding



- Fit CTD track first...
  - ▷ Use segments found in Axial layers of CTD.
  - ▷ Seed segment from outer superlayer.
  - ▷ Use beam line constraint to match inner segments.
  - ▷ Fit track with fast circle fit for r-ϕ tracks.
- Match MVD r-ϕ hits to track.
  - ▷ Refit r-ϕ track including MVD r-ϕ hits which are given a larger weight in the fit.
- Constrained or unconstrained fit?
  - ▷ Pattern recognition better with constrained tracks.
  - ▷ Secondary vertices require unconstrained tracks ⇒ use unconstrained fit for tracks after MVD hits have been matched.

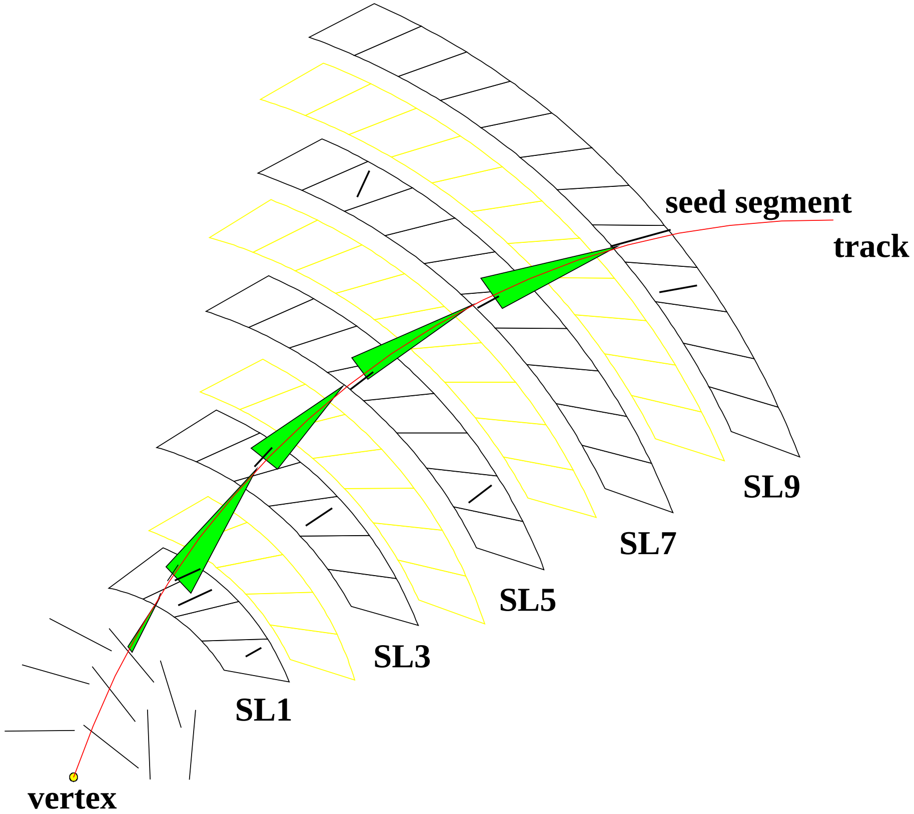
Axial Track Finding



- Fit CTD track first...
  - ▷ Use segments found in Axial layers of CTD.
  - ▷ Seed segment from outer superlayer.
  - ▷ Use beam line constraint to match inner segments.
  - ▷ Fit track with fast circle fit for r-ϕ tracks.
- Match MVD r-ϕ hits to track.
  - ▷ Refit r-ϕ track including MVD r-ϕ hits which are given a larger weight in the fit.

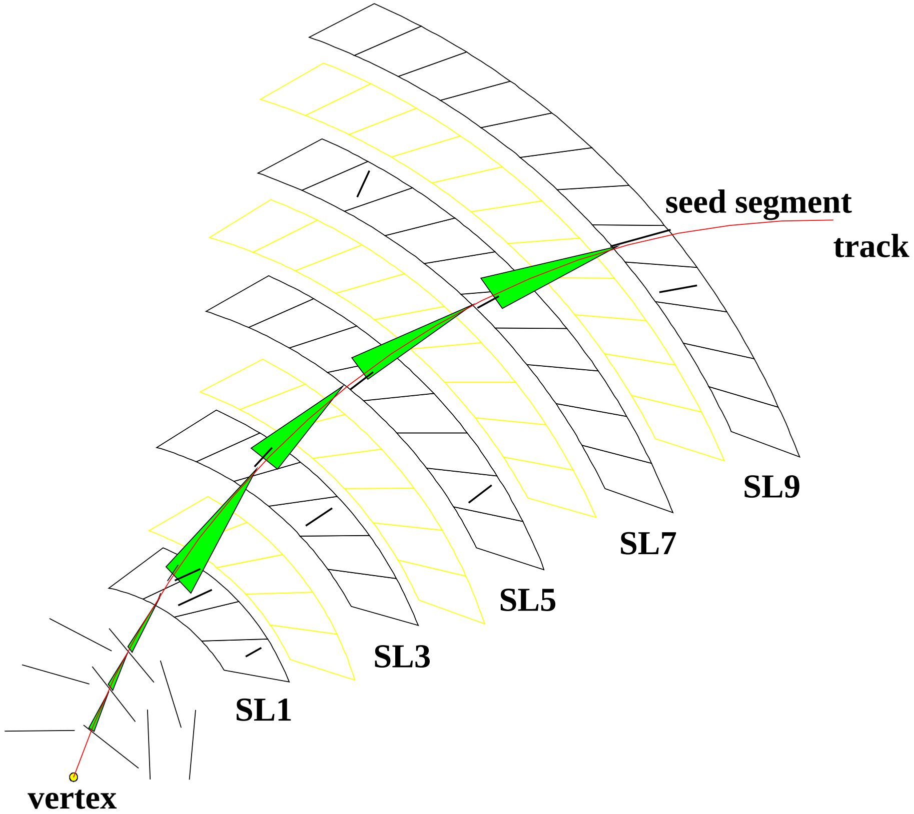
- Constrained or unconstrained fit?
  - ▷ Pattern recognition better with constrained tracks.
  - ▷ Secondary vertices require unconstrained tracks ⇒ use unconstrained fit for tracks after MVD hits have been matched.

Axial Track Finding



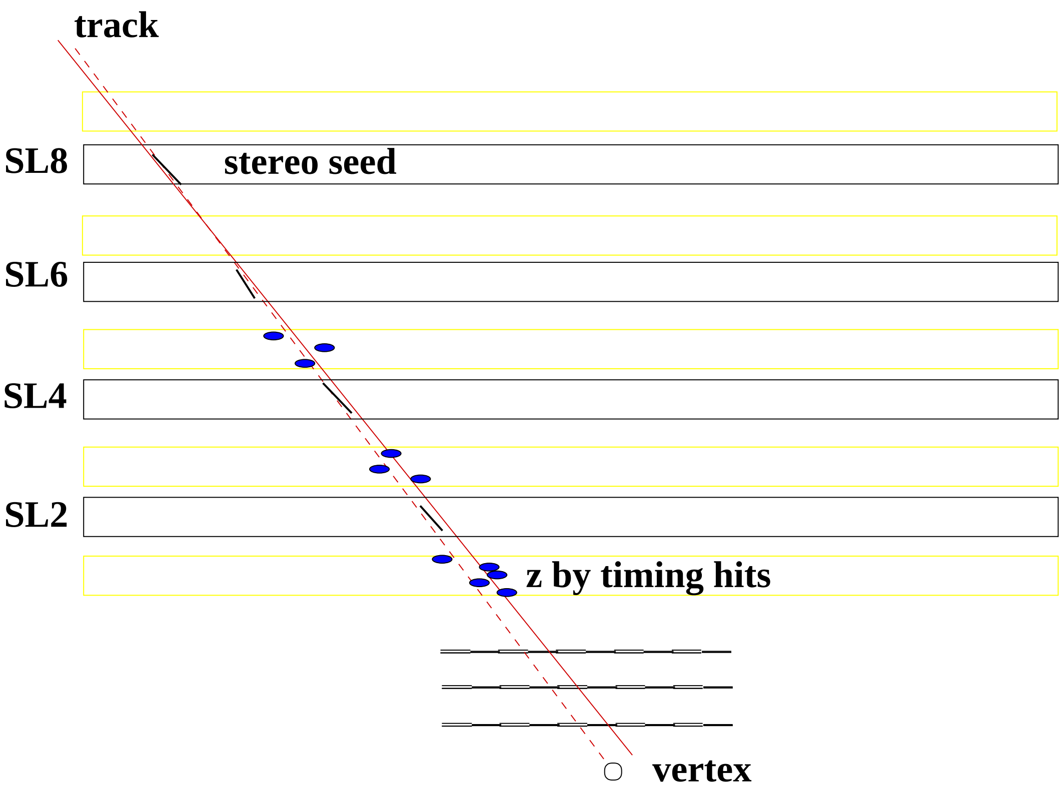
- Fit CTD track first...
  - ▷ Use segments found in Axial layers of CTD.
  - ▷ Seed segment from outer superlayer.
  - ▷ Use beam line constraint to match inner segments.
  - ▷ Fit track with fast circle fit for r-ϕ tracks.
- Match MVD r-ϕ hits to track.
  - ▷ Refit r-ϕ track including MVD r-ϕ hits which are given a larger weight in the fit.
- Constrained or unconstrained fit?
  - ▷ Pattern recognition better with constrained tracks.
  - ▷ Secondary vertices require unconstrained tracks ⇒ use unconstrained fit for tracks after MVD hits have been matched.

Axial Track Finding



- Fit CTD track first...
  - ▷ Use segments found in Axial layers of CTD.
  - ▷ Seed segment from outer superlayer.
  - ▷ Use beam line constraint to match inner segments.
  - ▷ Fit track with fast circle fit for r-ϕ tracks.
- Match MVD r-ϕ hits to track.
  - ▷ Refit r-ϕ track including MVD r-ϕ hits which are given a larger weight in the fit.
- Constrained or unconstrained fit?
  - ▷ Pattern recognition better with constrained tracks.
  - ▷ Secondary vertices require unconstrained tracks ⇒ use unconstrained fit for tracks after MVD hits have been matched.

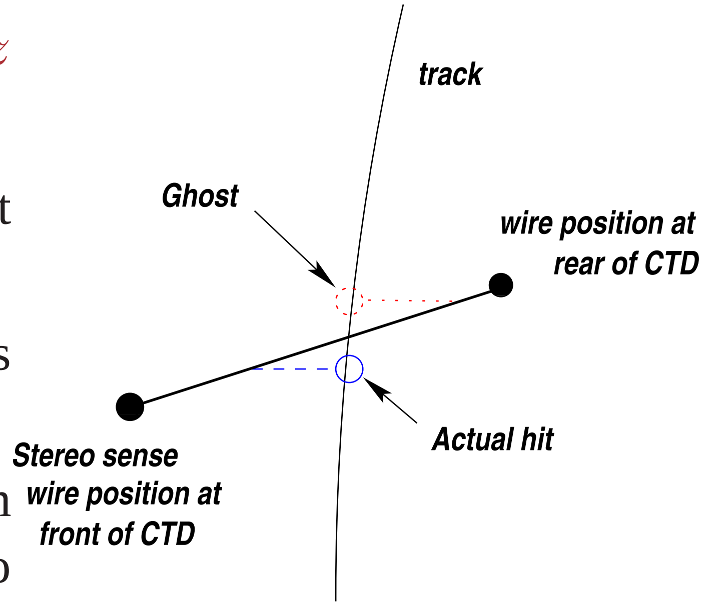
Three-Dimensional Track Finding



- After finding finding 2D tracks, match hits on wires in superlayers 1, 3 and 5 with z-by-timing hits to get course z positions (z hit resolution $\sim$ 6cm)
  - ▷ All 8 wire layers in superlayer 1,
  - ▷ Even numbered wire layers in superlayers 3 and 5.

Three-Dimensional Track Finding

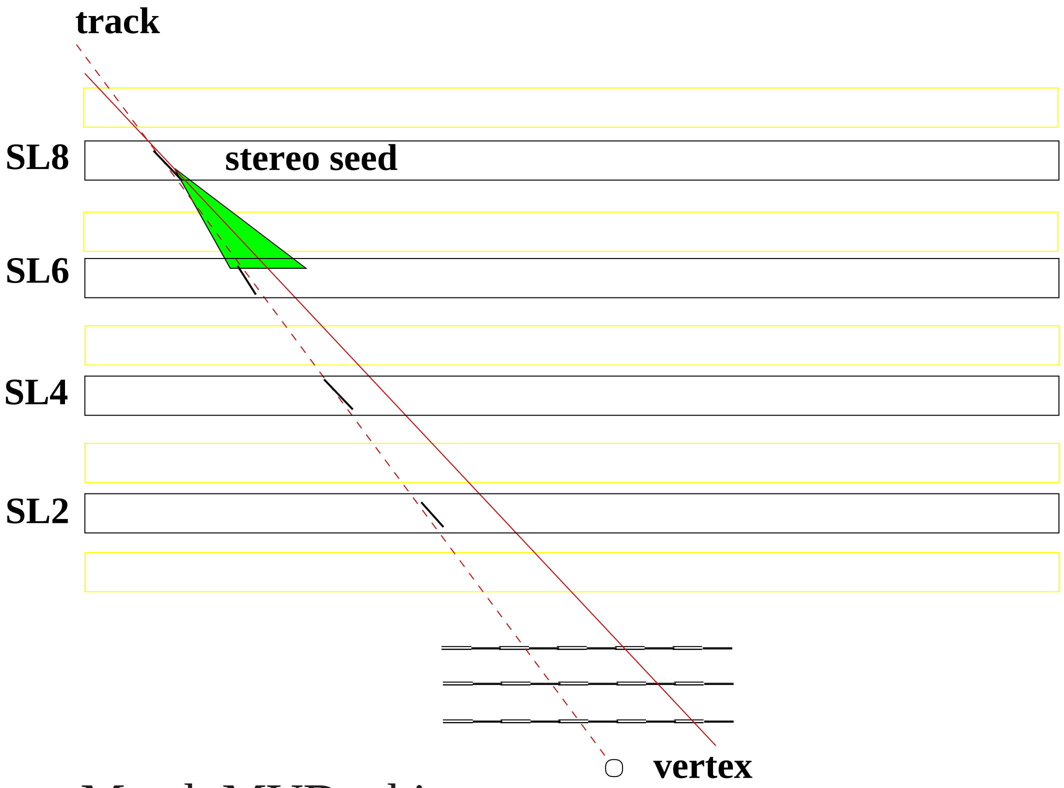
- In stereo superlayers, no r-ϕ information ➡ z position given by position of hit along wire.
  - ▷ Calculate which stereo cells might intersect track in r-ϕ – up to 4 cells per superlayer.
  - ▷ Calculate z positions along wires so that hits lies on the track.
  - ▷ Ghost ambiguity resolved using hits from segment candidate most closely pointing to beamline.



Stereo ghost ambiguity

- Each hit position must be calculated seperately for each track.
- Non-analytical, iterative fit ➡ Time consuming, use only segment end points, matched segments stored in lookup table, indexed by both track and cell number.

Three-Dimensional Track Finding

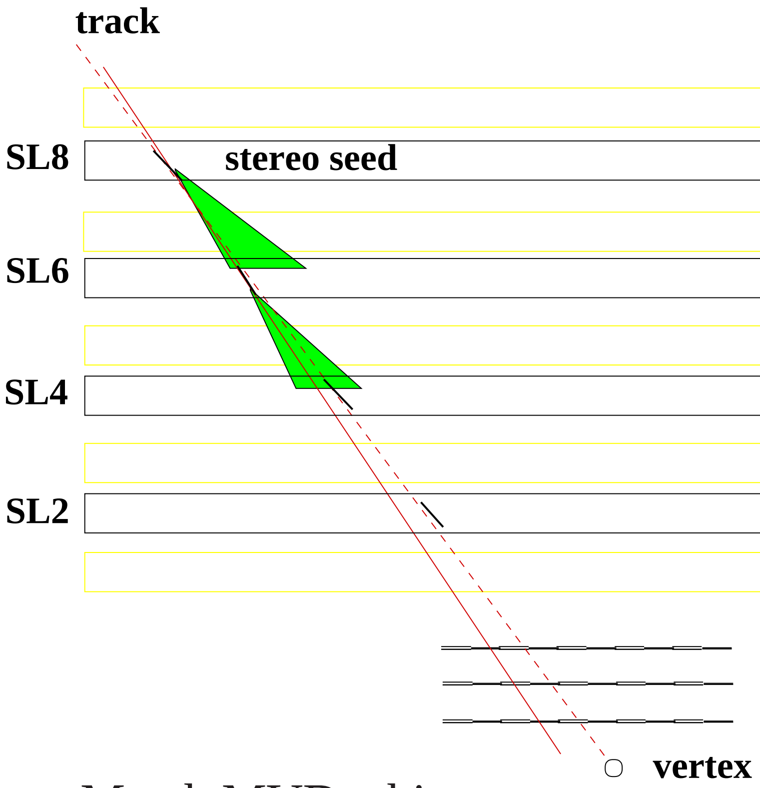


- After finding 2D tracks in r-ϕ, look for 3D tracks in z and the axial track length, s.
- Match stereo segments to track in r-ϕ to get z positions for z-s fit.
  - ▷ Extrapolation to inner CTD layers.
  - ▷ Use z-by-timing hits ($\pm$6cm resolution) in fit to guide extrapolation.
  - ▷ Refit track.

- Match MVD z hits...
  - ▷ Refit z-s track including MVD z-hits again given a larger weight in the fit to reflect the higher MVD spatial resolution.

- The track is assigned a weight, w_{track}, based on the number of stereo segments and matched MVD hits.

Three-Dimensional Track Finding

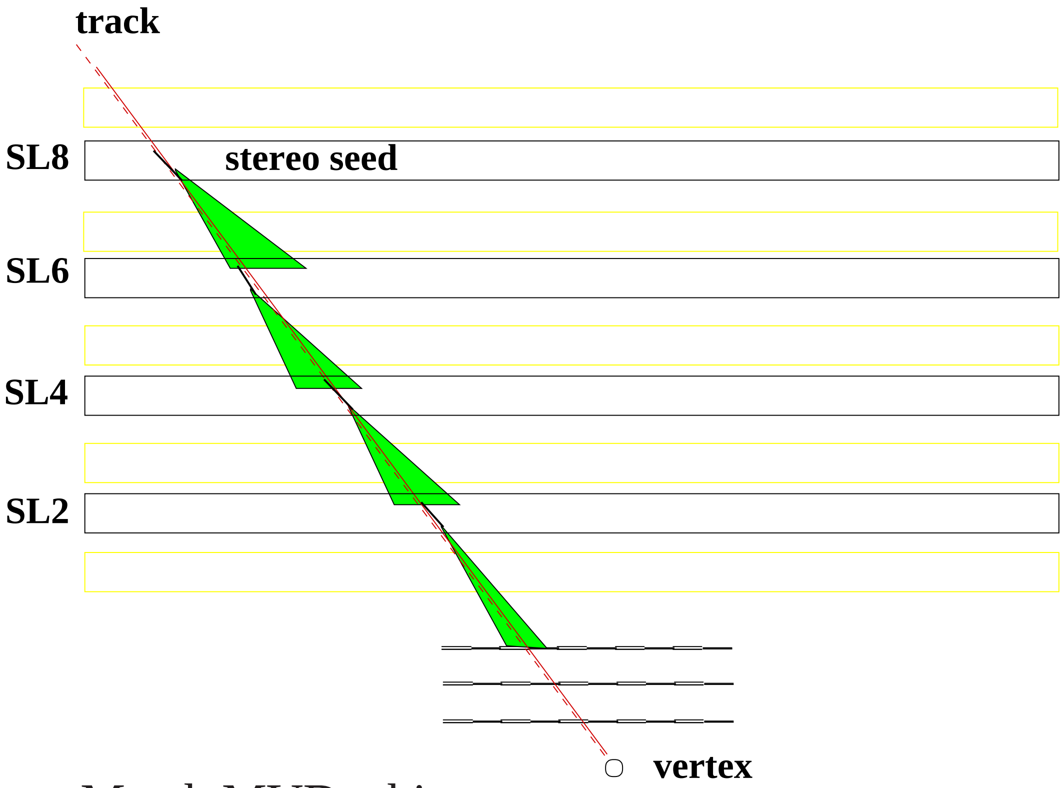


- After finding 2D tracks in r-ϕ, look for 3D tracks in z and the axial track length, s.
- Match stereo segments to track in r-ϕ to get z positions for z-s fit.
  - ▷ Extrapolation to inner CTD layers.
  - ▷ Use z-by-timing hits ($\pm$6cm resolution) in fit to guide extrapolation.
  - ▷ Refit track.

- Match MVD z hits...
  - ▷ Refit z-s track including MVD z-hits again given a larger weight in the fit to reflect the higher MVD spatial resolution.

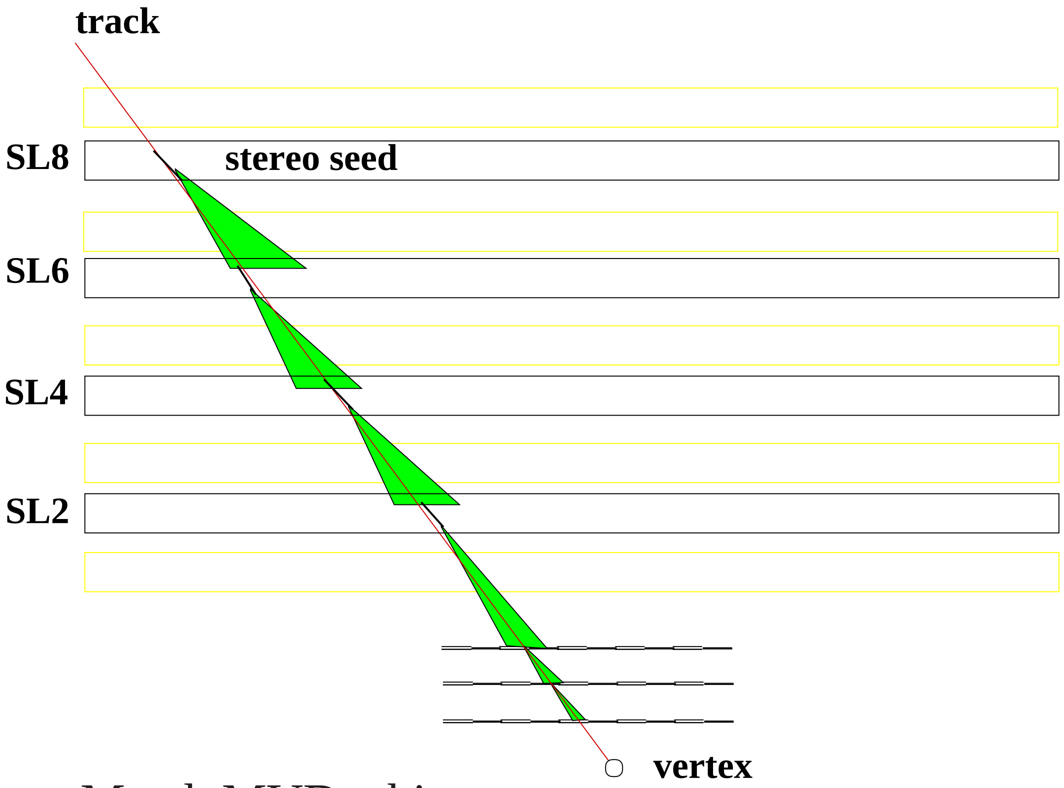
- The track is assigned a weight, w_{track}, based on the number of stereo segments and matched MVD hits.

Three-Dimensional Track Finding



- After finding 2D tracks in r-ϕ, look for 3D tracks in z and the axial track length, s.
- Match stereo segments to track in r-ϕ to get z positions for z-s fit.
  - ▷ Extrapolation to inner CTD layers.
  - ▷ Use z-by-timing hits ($\pm$6cm resolution) in fit to guide extrapolation.
  - ▷ Refit track.
- Match MVD z hits...
  - ▷ Refit z-s track including MVD z-hits again given a larger weight in the fit to reflect the higher MVD spatial resolution.
- The track is assigned a weight, w_{track}, based on the number of stereo segments and matched MVD hits.

Three-Dimensional Track Finding

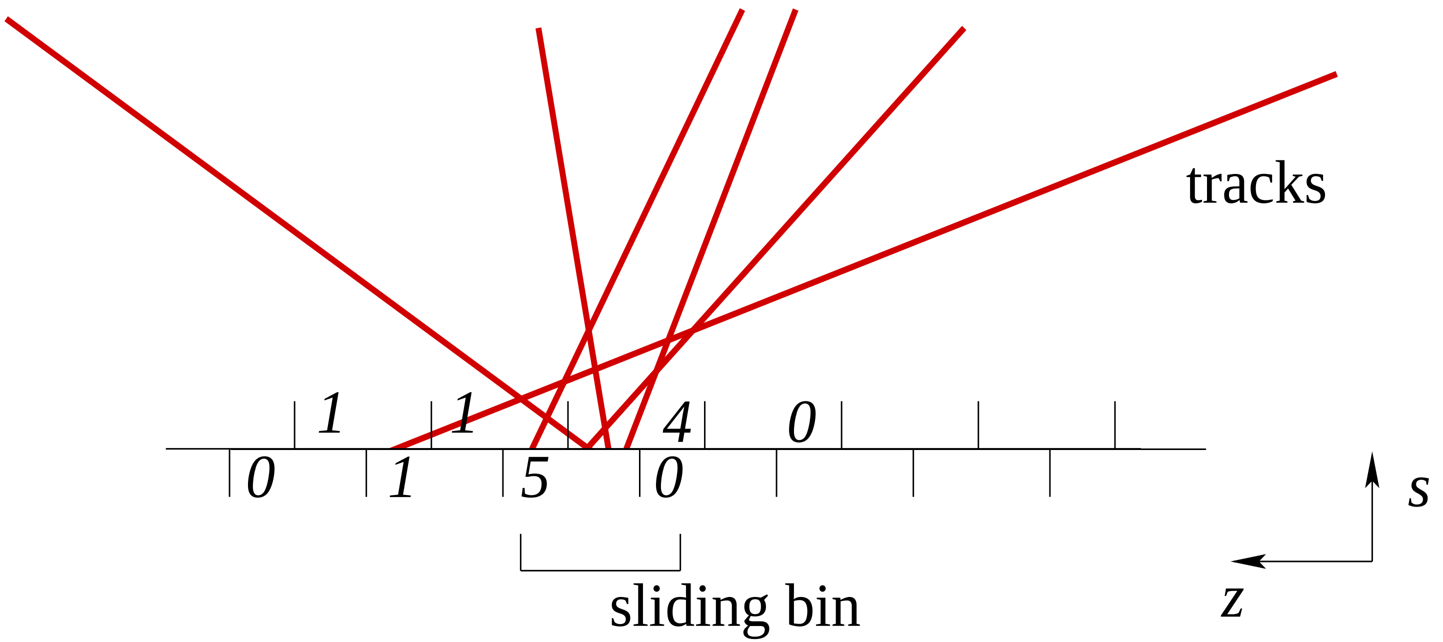


- After finding 2D tracks in r-ϕ, look for 3D tracks in z and the axial track length, s.
- Match stereo segments to track in r-ϕ to get z positions for z-s fit.
  - ▷ Extrapolation to inner CTD layers.
  - ▷ Use z-by-timing hits ($\pm$6cm resolution) in fit to guide extrapolation.
  - ▷ Refit track.

- Match MVD z hits...
  - ▷ Refit z-s track including MVD z-hits again given a larger weight in the fit to reflect the higher MVD spatial resolution.

- The track is assigned a weight, w_{track}, based on the number of stereo segments and matched MVD hits.

Primary Vertex Algorithm



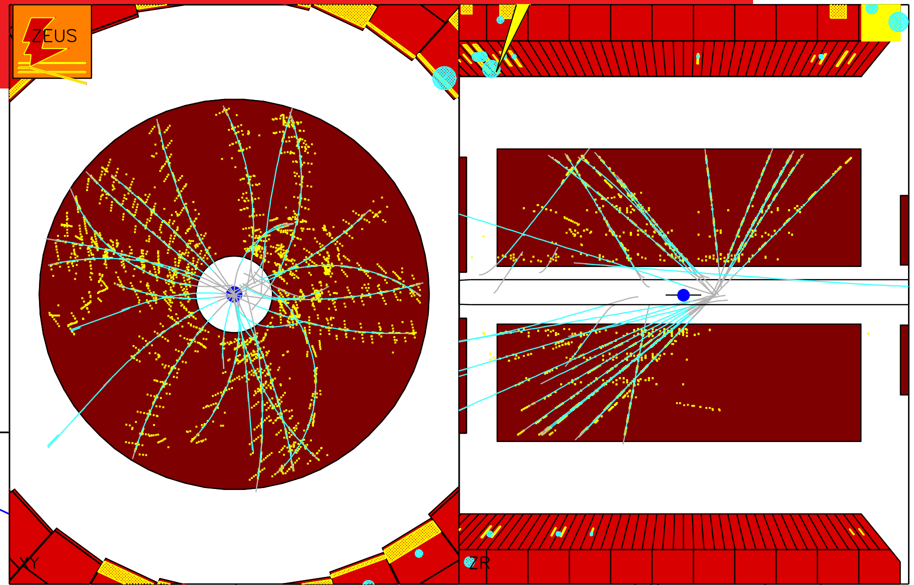
- The track-vertex, z_{track}, for each track is binned in overlapping bins of z with weight, w^2_{track}.
- An initial mean vertex position is calculated using tracks in the most probable bin (MPB) with the largest number of weights,

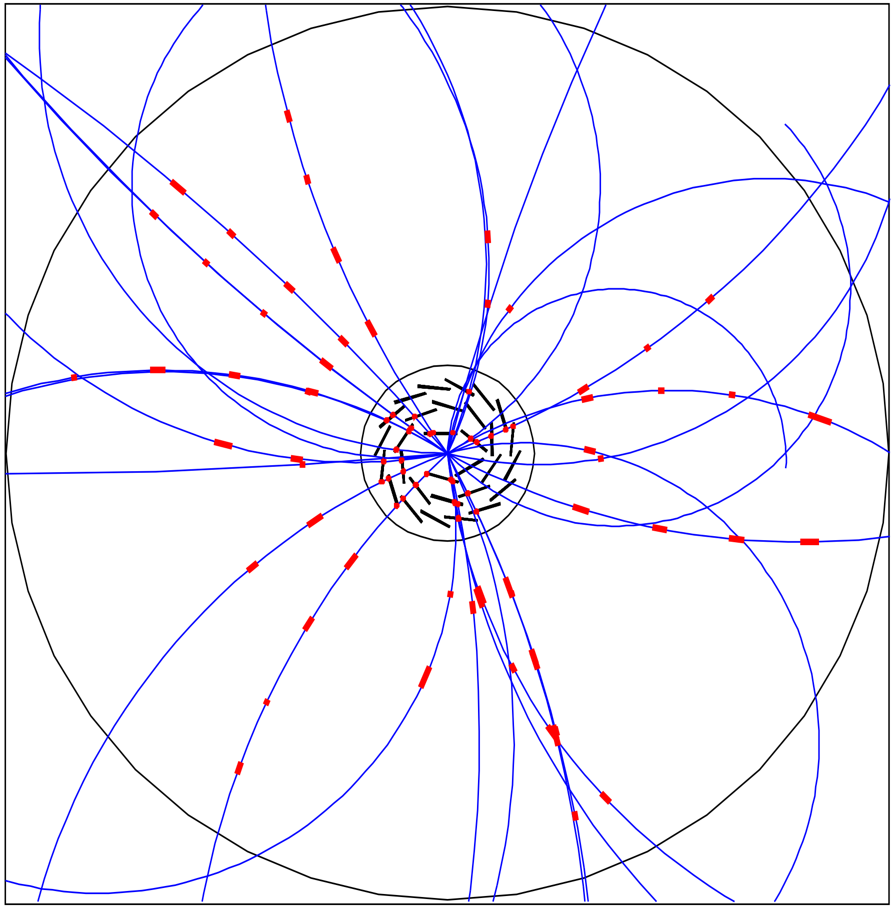
$$z_{\text{initial}} = \sum z_{\text{track}} w^2_{\text{track}} / \sum w^2_{\text{track}}$$

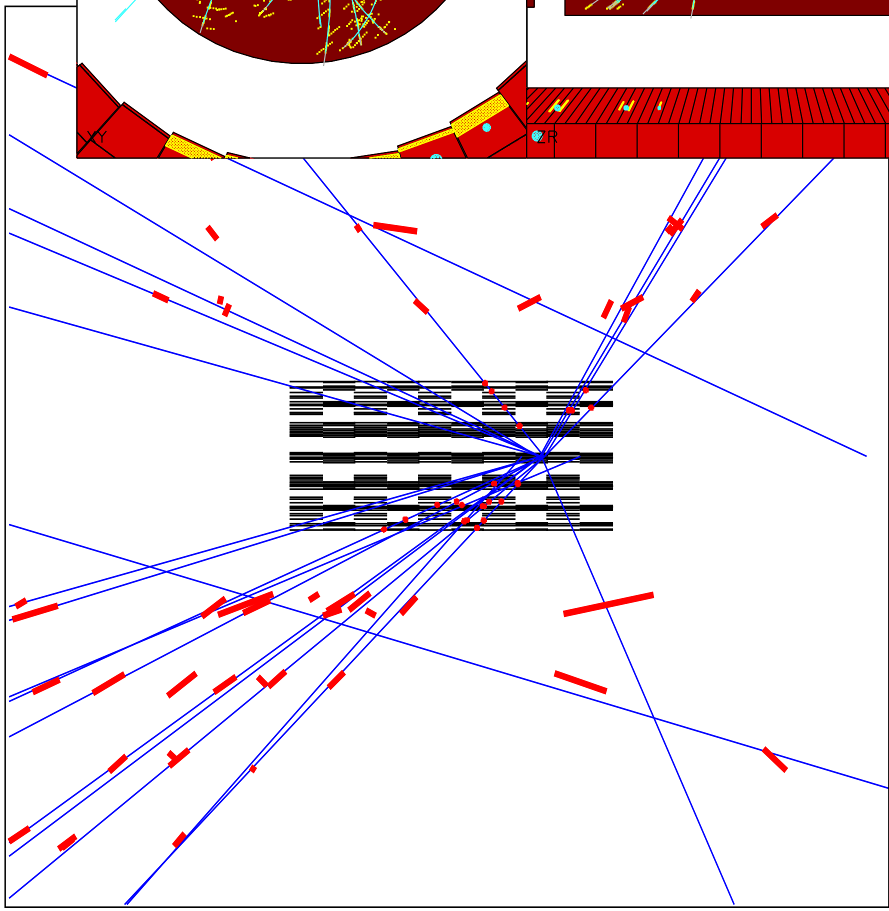
- The event vertex is calculated using all tracks within ± 9 cm of z_{initial}.

Event Topology – Monte Carlo

- Combined CTD+Barrel MVD reconstruction.
- Algorithm reconstructs complex event topology.

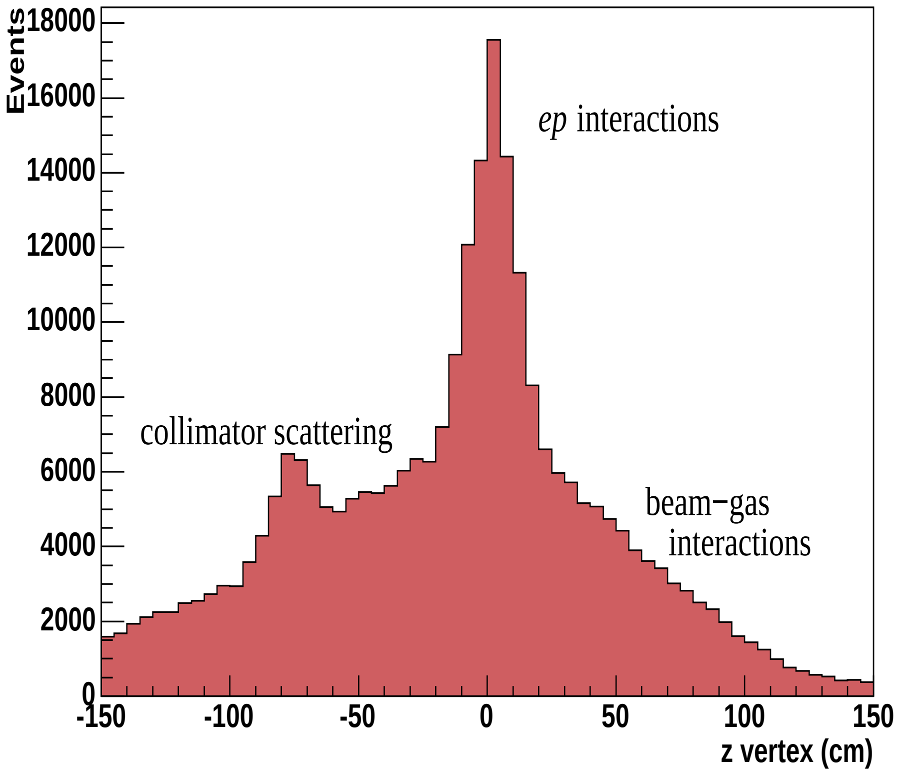






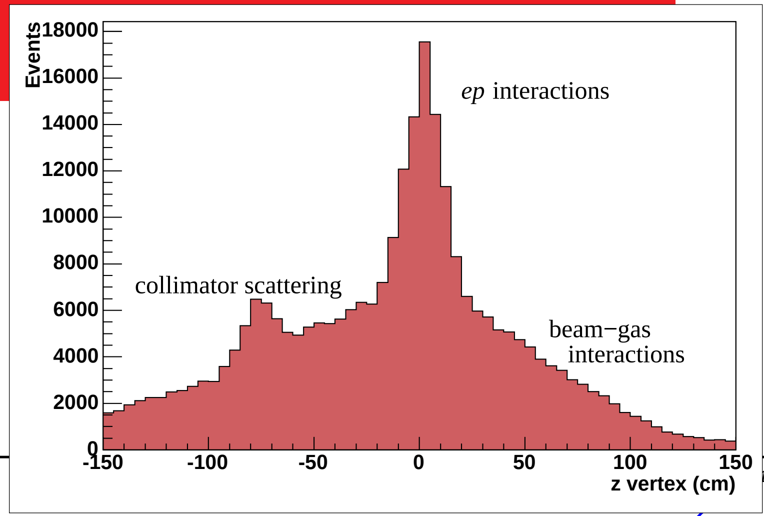
GTT Operation during Data Taking

- Routine operation of GTT during data taking, providing full tracking
- Greater than 40 pb^{-1} on tape.
- HERA beamgas related background significantly larger than pre-upgrade running.
- Current running considerations...
  - ▷ MVD hits from low momentum tracks in busy events bias reconstruction. ➡ run GTT in CTD only mode.

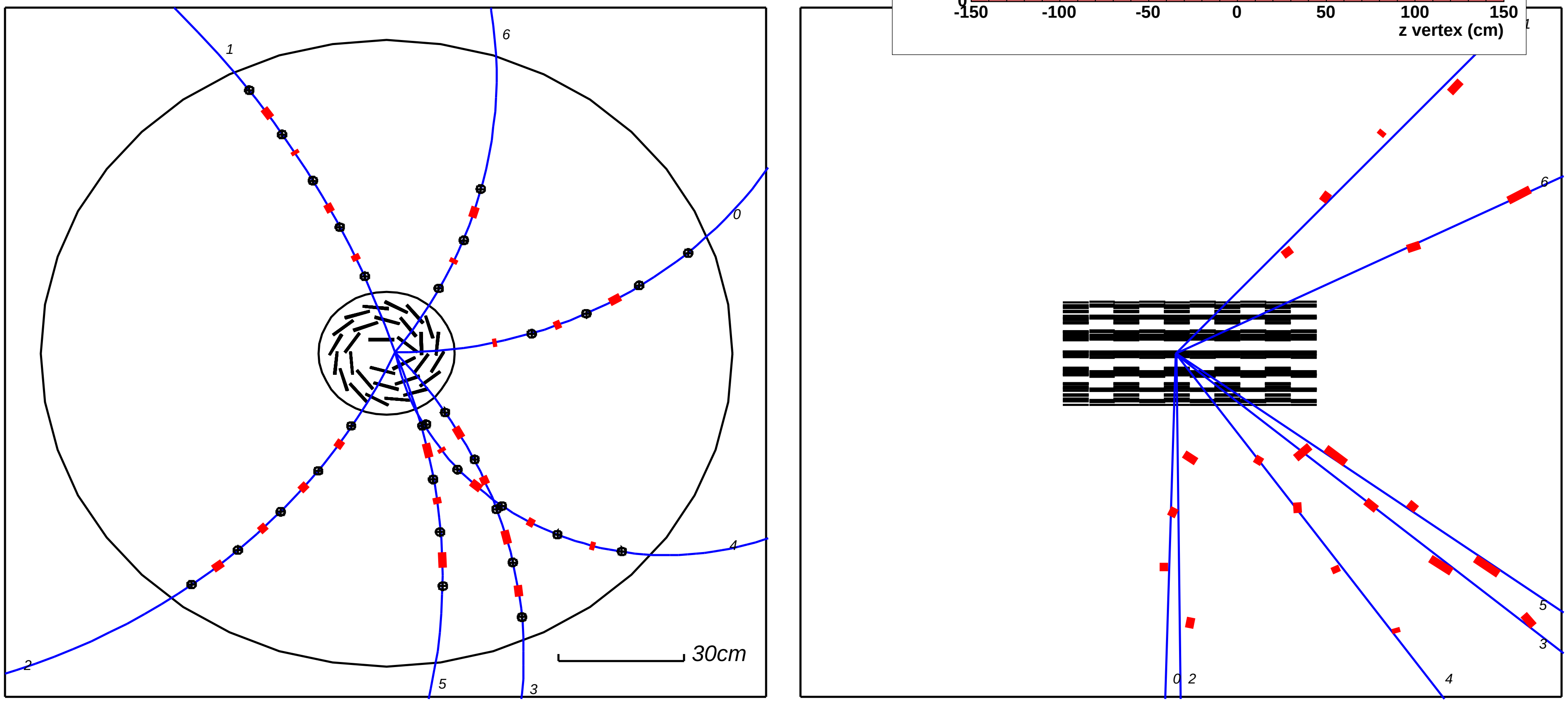


- Fully integrated into ZEUS DAQ and Trigger system ➡ Used in physics filters to select events.

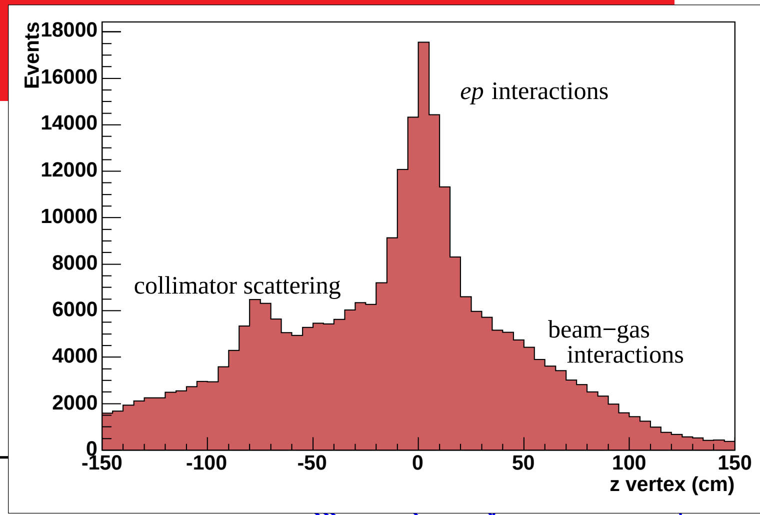
Event Topology – Real Data



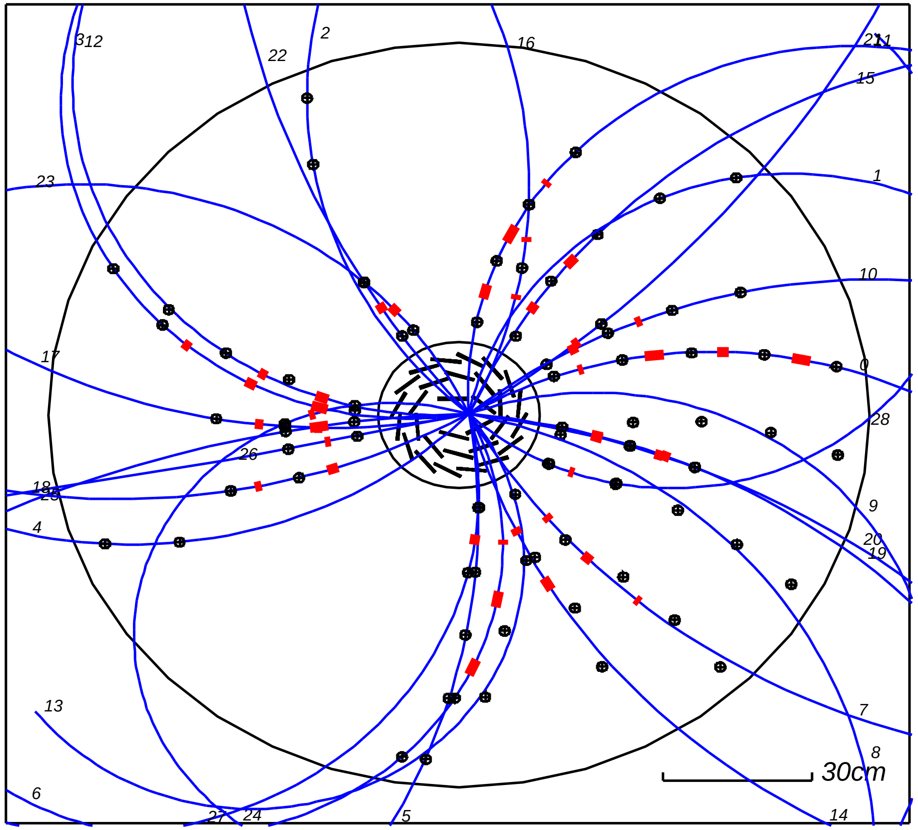
- Clean *ep* interaction event.

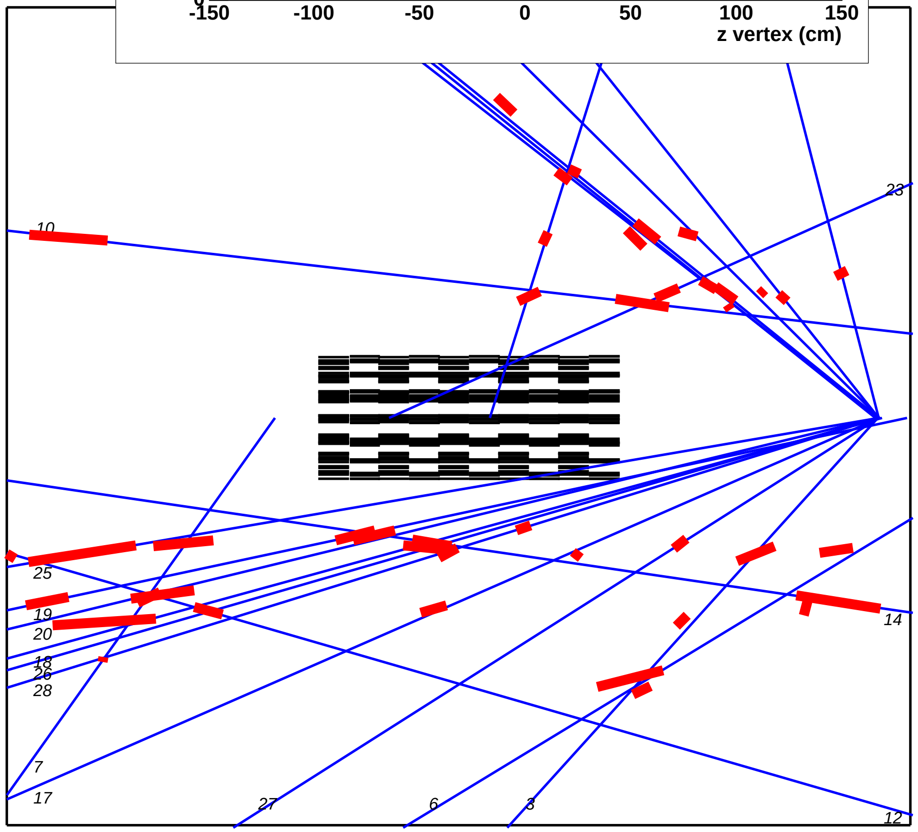


Event Topology – Real Data (contd)

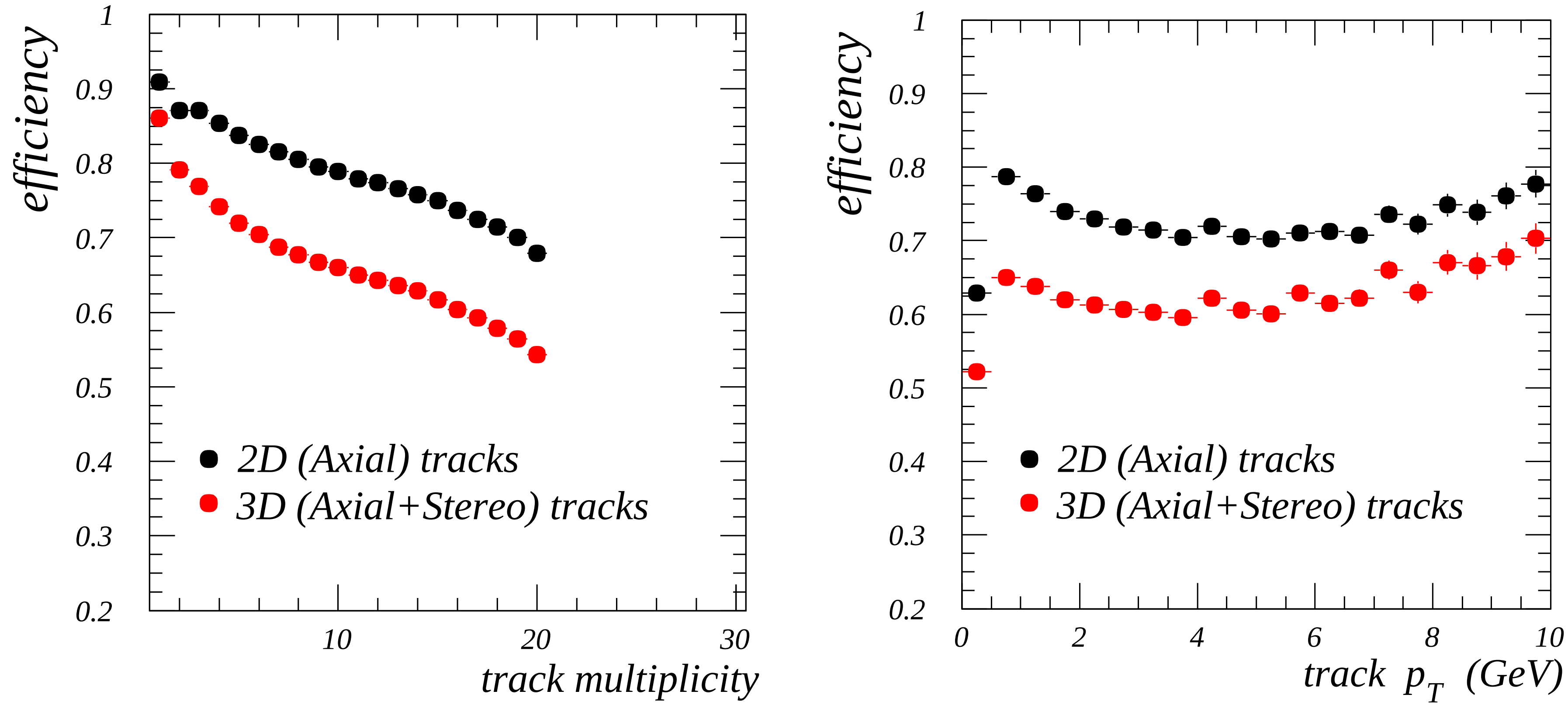


- Beamgas scattering from -80cm collimator.



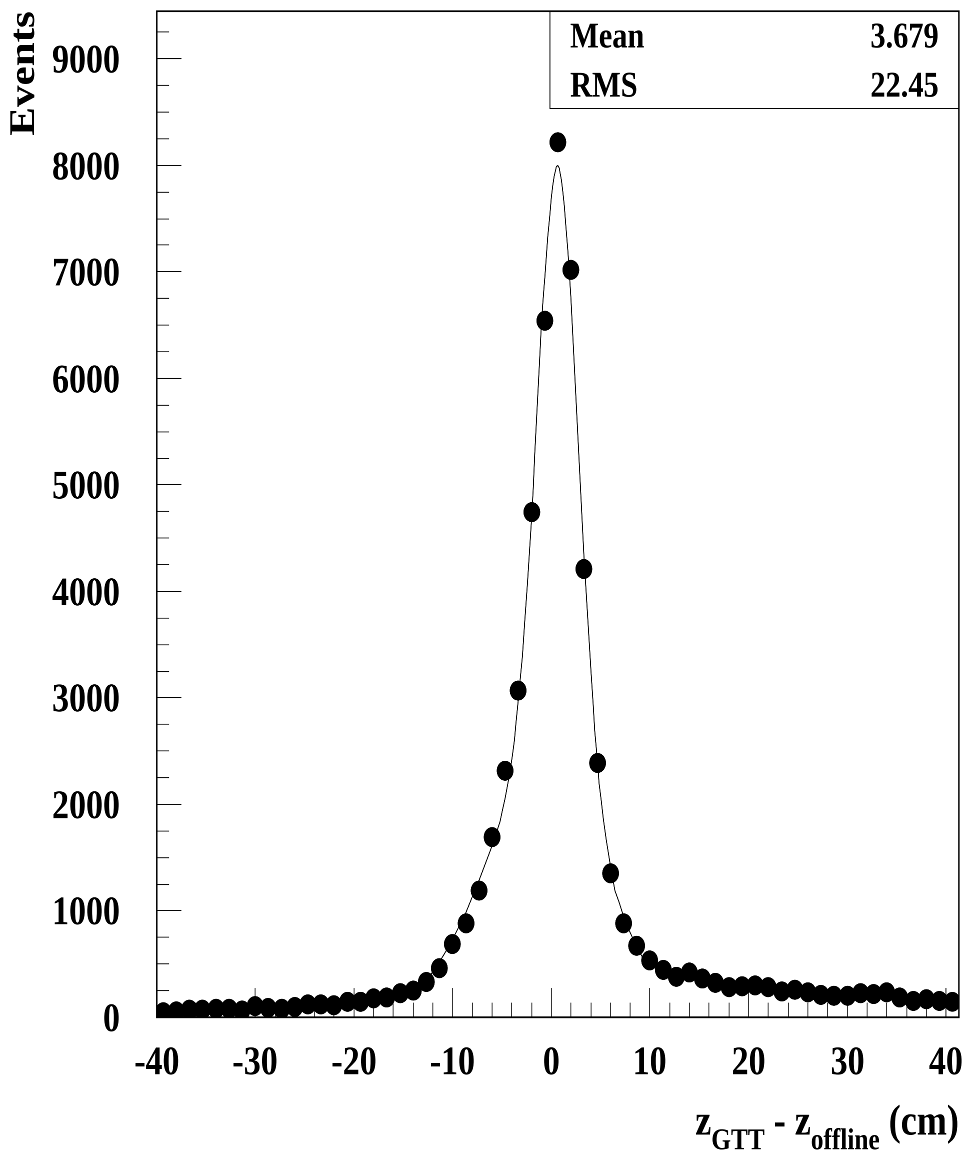


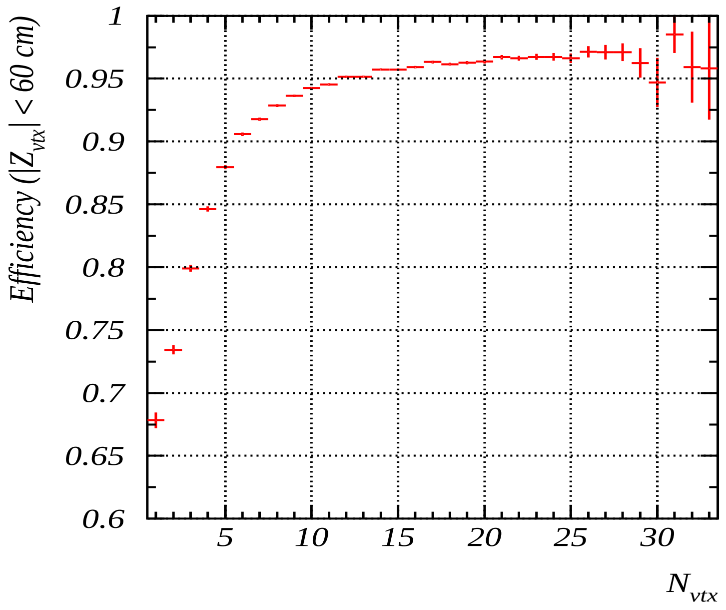
Performance Studies



- Detailed performance studies: comparing online with offline tracking.
  - ▷ Resolutions: for full length tracks,
    $\sigma(1/p_T) \sim 0.07\,\mathrm{GeV}^{-1}$, $\sigma(\phi) \sim 12\,\mathrm{mrad}$, $\sigma(\eta) \sim 0.05$
- Complex pattern recognition at high multiplicity ⇒ efficiency falls steeply.

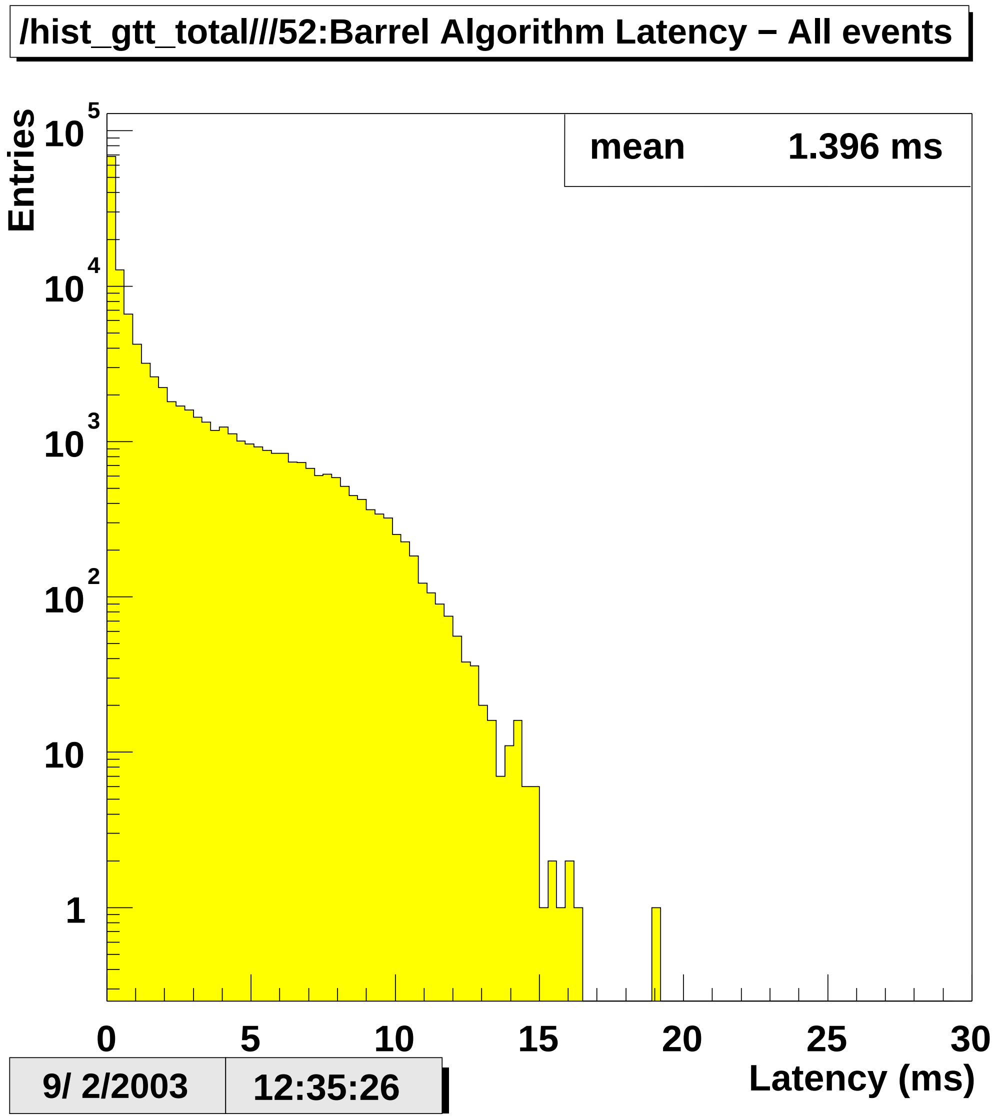
Vertex Algorithm Performance – Data

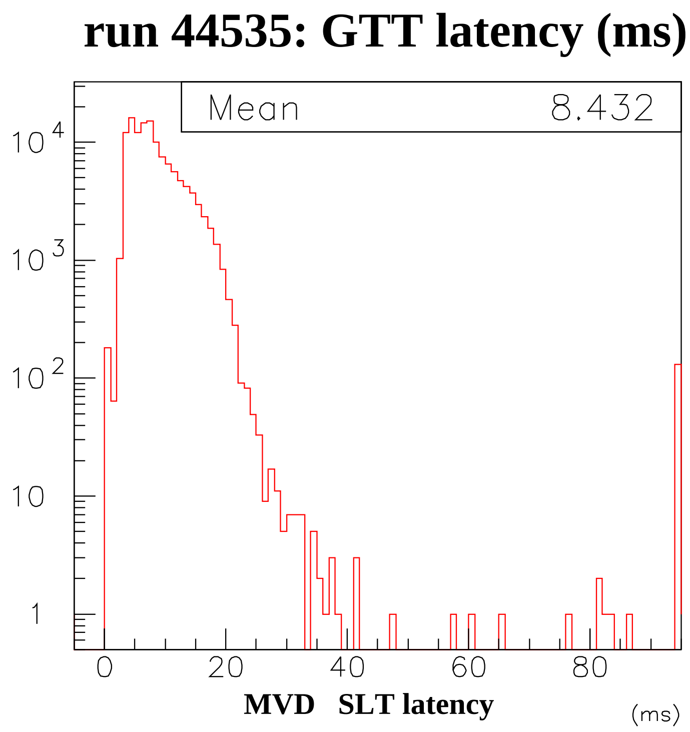




- Residual with respect to offline vertex (dijet photoproduction selection).
- Double Gaussian fit, widths $\sigma_Z \sim 2$ cm and $\sigma_Z \sim 6$ cm.
- Efficiency for vertex within ± 60 cm of nominal interaction region, $> 90\%$ for events with greater than 5 vertex tracks.

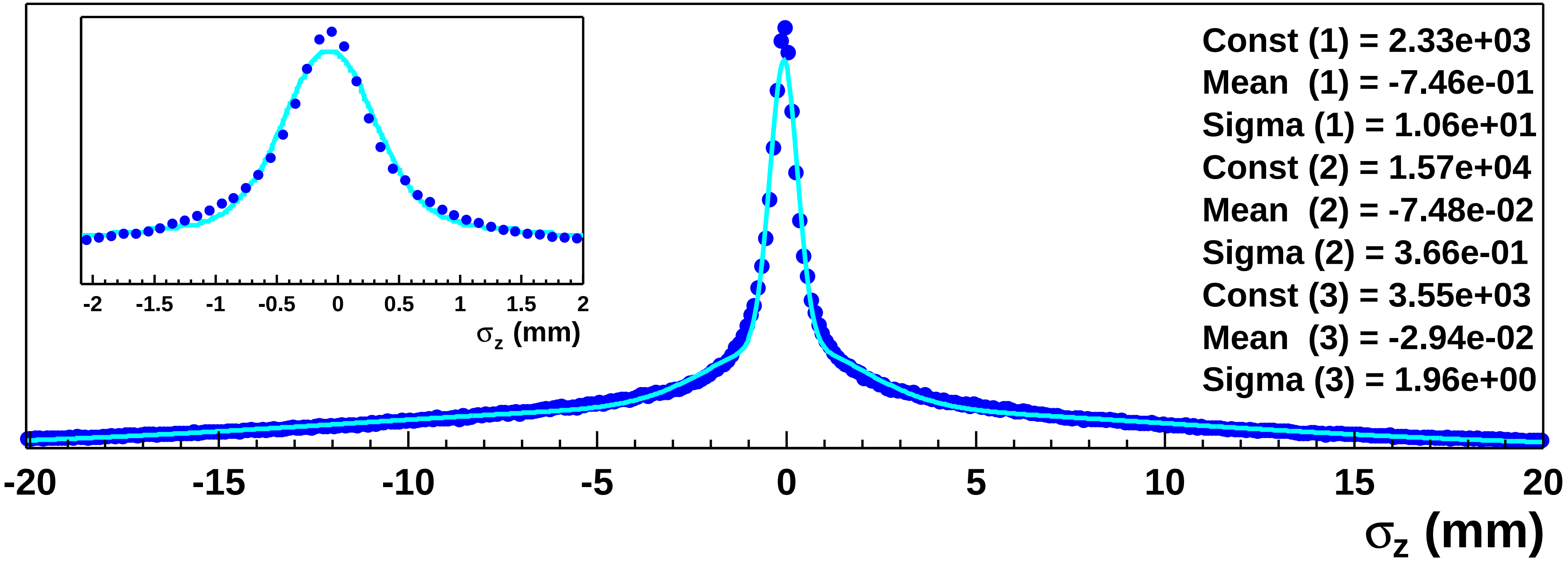
Latency





- Run 44535, mean latency at the GSLT 8.4ms.
- Mean algorithm latency, 1.4ms ⇒ latency dominated by transfer time of large data volumes.

Vertex Algorithm Performance – Monte Carlo

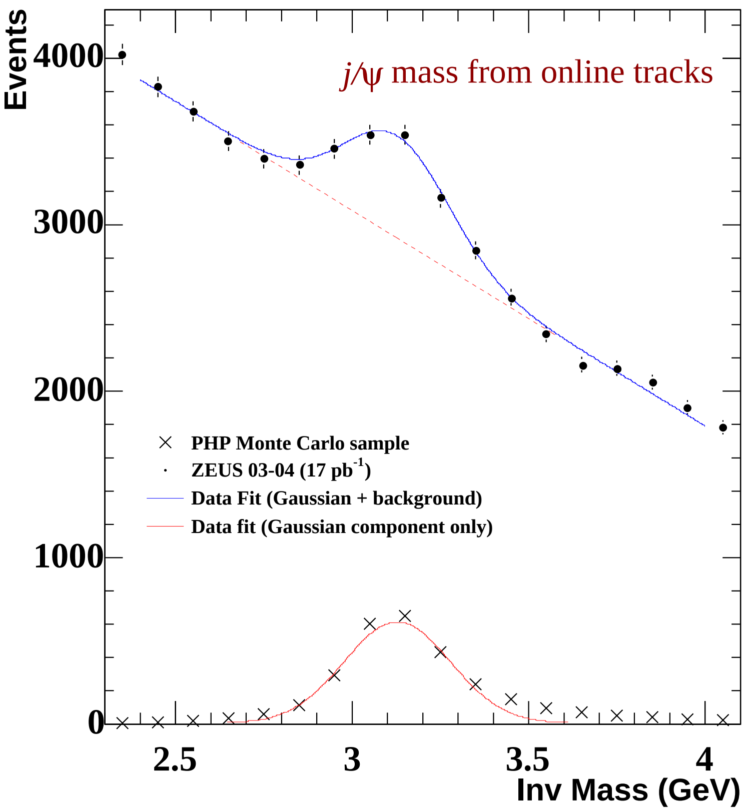


- Broad interaction region at HERA, width $\sigma_Z \sim 11\text{cm}$ from proton bunch length.
  - ▷ CTD-SLT event z-vertex resolution $\sim$9cm.
- GTT z-vertex resolution from dijet photoproduction Monte Carlo including MVD barrel information, $\sim 400\mu\text{m}$
- With ideal conditions, approaches 100% efficiency for vertices within $\pm 25\text{cm}$ for events with ≥ 5 tracks.

Summary and Outlook

- The GTT Barrel Algorithm running very stable and reliable throughout 2003-04 ⇒ over 40 pb^{-1} on tape.
- Fully integrated in ZEUS DAQ and Trigger system, used in physics filters to select events online.

- For the HERA startup (October 2004)
  - ▷ Include MVD hits in standard algorithm ⇒ greatly improve vertex resolution.
- Starting to address improved heavy flavour selection...



- Looking forward to more high luminosity data taking with an improved Barrel Algorithm when HERA restarts in October.