# Status of W-DHCAL Analysis: Data Quality and Calibration

Christian Grefe (CERN)

on behalf of the CALICE collaboration and the CLIC physics and detector study

14. November 2013
Linear Collider Workshop, Tokyo

# Outline

1 Introduction

2 Data Quality

3 Calibration

4 Digitization (RPCSim)

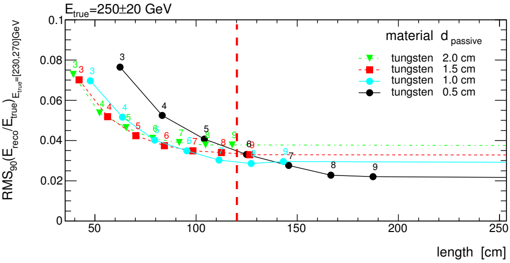
# Why Tungsten?

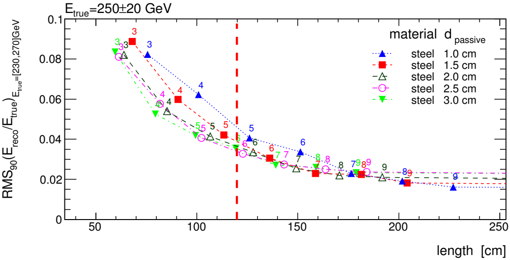
- Calorimeters have to be inside coil to avoid large dead areas
- Coil size limited by cost and feasibility
- Make optimal use of the available space
- Sampling calorimeter with reasonably small gap for active layer
  - Assume 5 mm plastic scintillator + 1.5 mm for readout
- Test various (simplified) calorimeter geometries
  - Absorber materials: steel, tungsten
  - Absorber thickness: 5–30 mm

| Material | $X_0$ [cm] | $\lambda_I$ [cm] |
|---|---|---|
| Steel | 1.73 | 16.9 |
| Tungsten | 0.37 | 10.2 |

P. Speckmayer, C.G. - LCD-Note-2010-001

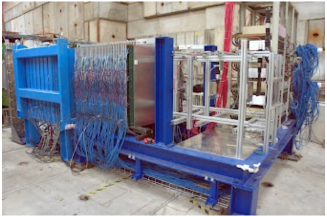
- Fit resolution to determine sampling and constant term
  $\sigma(E)/E = \frac{s}{\sqrt{E}} \oplus c$
- Steel has better intrinsic resolution
- Steel leakage dominated up to 1.5 m HCal depth
- For limited calorimeter depth ($\sim 1.2\,\mathrm{m}$) tungsten performs better
- Optimal sampling with tungsten absorber thickness around 10 mm
- Corresponds to a total depth of 7.5 $\lambda_I$

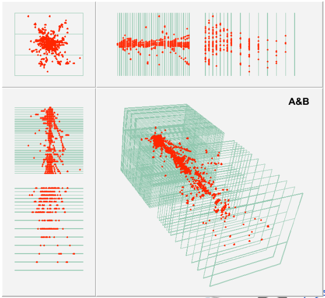


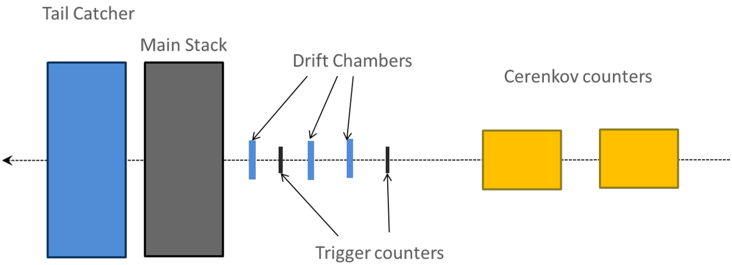
# Data Taking at CERN (2012)

- 54 RPC layers:
  39 with tungsten absorber (main stack),
  15 with steel absorber (tail catcher)
- Each layer instrumented with $96 \times 96$ $1 \times 1\,\mathrm{cm}^2$ pads $\Rightarrow \sim 500000$ channels
- PS (1–10 GeV): 1 run period of 2 weeks
- SPS (10–300 GeV): 2 + 1 + 1 weeks
- Dedicated μ and high rate runs
- In total $\sim$ 30 million events recorded





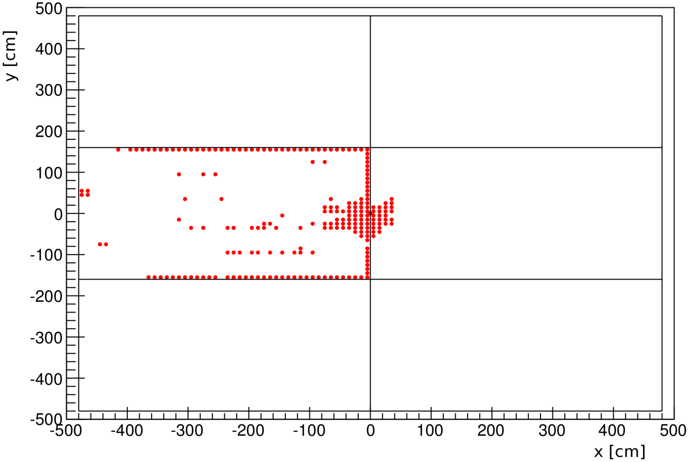
# Data Taking at CERN (2012)




- 39 layers W-DHCAL + 15 layers Fe-DHCAL
- $10 \times 10\,\mathrm{cm}^2$ scintillator triggers ($30 \times 30\,\mathrm{cm}^2$ for dedicated muon runs)
- Three wire chambers $\Rightarrow$ beam profile
- Two Cerenkov counters $\Rightarrow$ particle identification

# Data Quality Issues

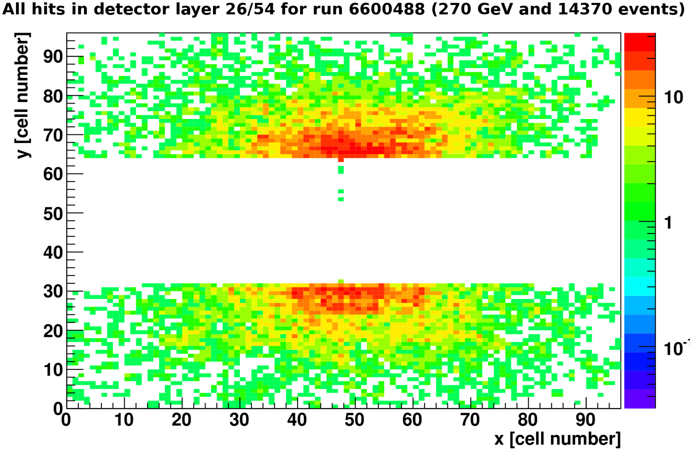
- Box events
- Dead RPC modules
- Dead and oversensitive chips
- Oversensitive cells
- Dead cells




Helga Holmestad (CERN, University of Oslo)

# Data Quality Issues

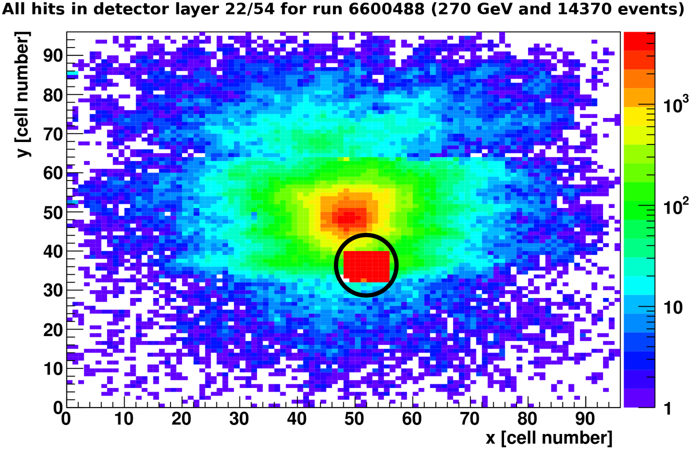
- Box events
- Dead RPC modules
- Dead and oversensitive chips
- Oversensitive cells
- Dead cells




Helga Holmestad (CERN, University of Oslo)

# Data Quality Issues

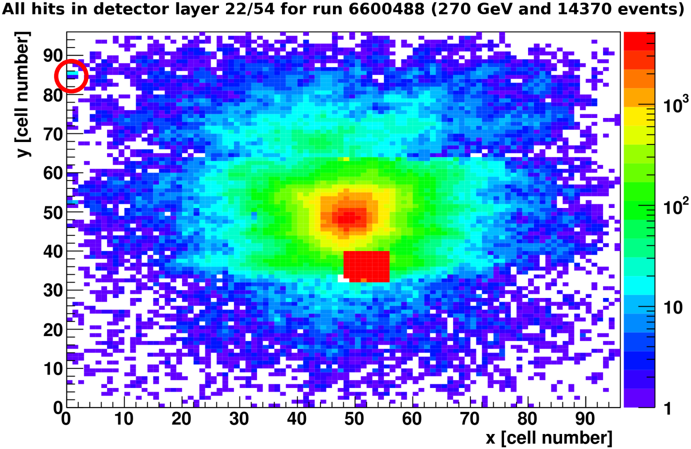
- Box events
- Dead RPC modules
- Dead and oversensitive chips
- Oversensitive cells
- Dead cells




Helga Holmestad (CERN, University of Oslo)

# Data Quality Issues

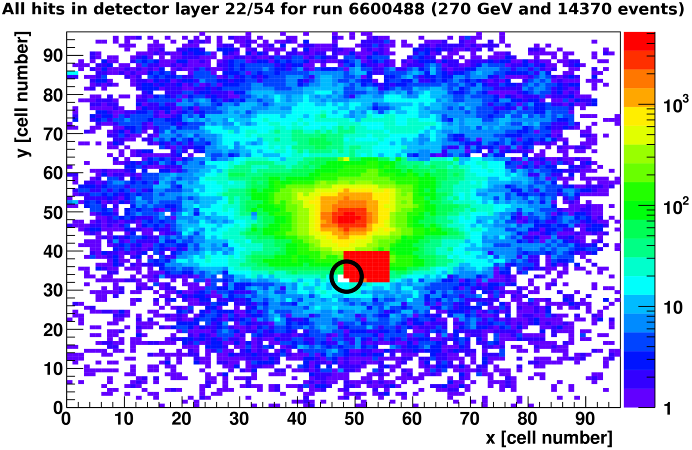
- Box events
- Dead RPC modules
- Dead and oversensitive chips
- Oversensitive cells
- Dead cells




Helga Holmestad (CERN, University of Oslo)

# Data Quality Issues

- Box events
- Dead RPC modules
- Dead and oversensitive chips
- Oversensitive cells
- Dead cells

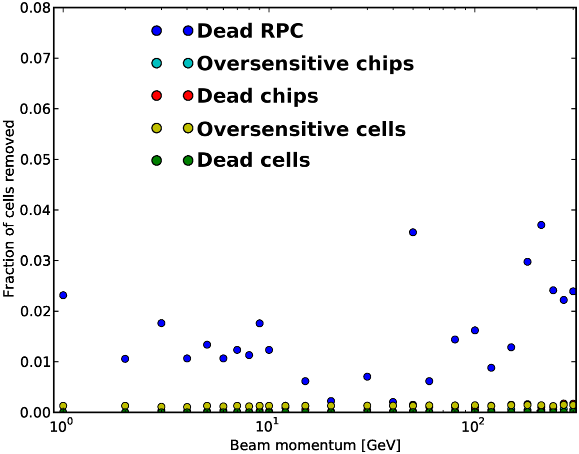



Helga Holmestad (CERN, University of Oslo)

# Cleaning Procedure

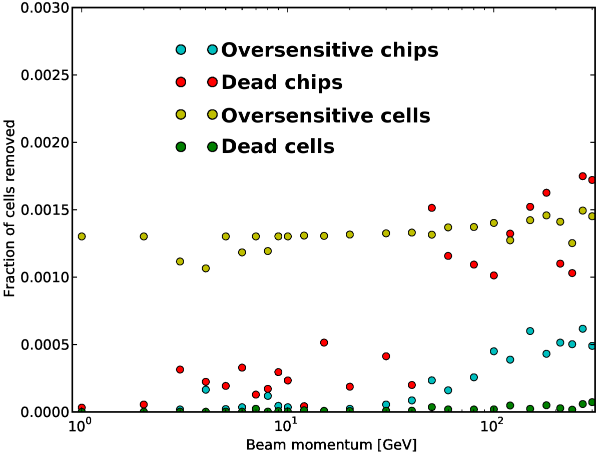
- Ignore duplicate hits → event based
- Remove out of time hits: only accept -1900 ns to -1700 ns → event based
- Algorithm to identify dead and noisy regions by looking for large steps (loose) → run based
- Identify and reject box events by looking for patterns along module boundaries → event based
- Re-run algorithm to identify dead and noisy regions (tight) → run based

# Fraction of Cells Removed




Helga Holmestad (CERN, University of Oslo)

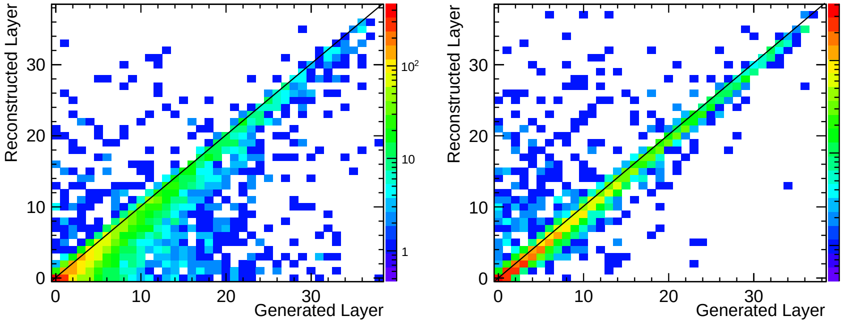
# Fraction of Cells Removed




Helga Holmestad (CERN, University of Oslo)

# Interaction Layer Definition

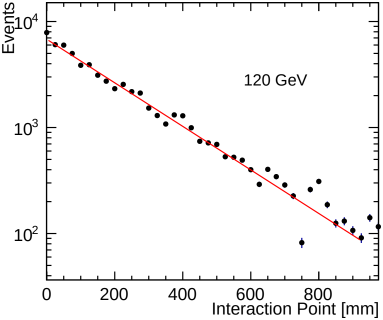
- Old definition: minimum of 3 hits in two consecutive layers
- New definition of interaction layer based on a three layer hit average
- Require increase of factor 2 and minimum average of 4
- Assume 3 hits in each "layer before stack" to allow identification in first layer

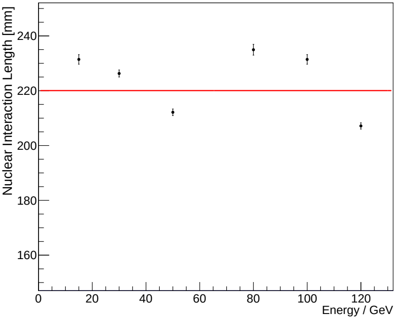



Christian Reichelt (Technical University of Denmark)

# Interaction Layer Definition

- Verified in data and MC that the interaction layer follows exponential drop
- Extract interaction length from exponential fit
  $\Rightarrow$ matches expected interaction length from material budget

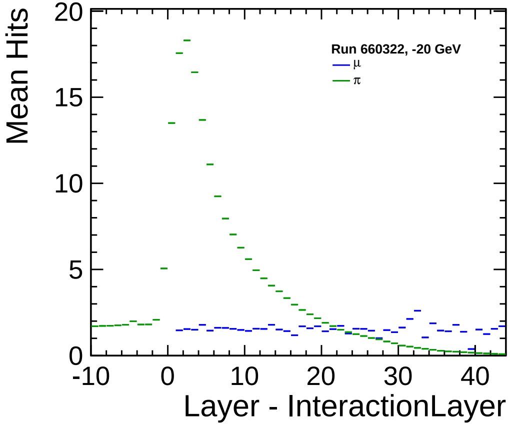



Christian Reichelt (Technical University of Denmark)

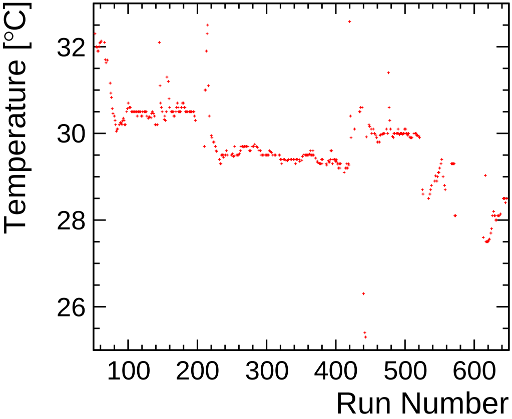
# Why Calibrate?

- DHCAL only measures number of hits
- Multiplicity $\mu$ and efficiency $\epsilon$ depend on many factors
  - Temperature
  - Pressure
  - Voltage
  - ...
- Temperature stabilized by tent and AC



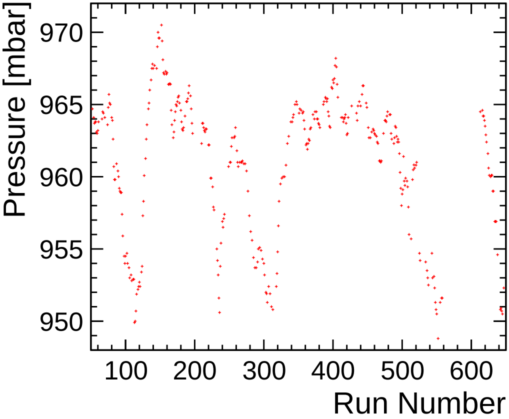
# Why Calibrate?

- DHCAL only measures number of hits
- Multiplicity $\mu$ and efficiency $\epsilon$ depend on many factors
  - Temperature
  - Pressure
  - Voltage
  - ...
- Temperature stabilized by tent and AC



# Why Calibrate?

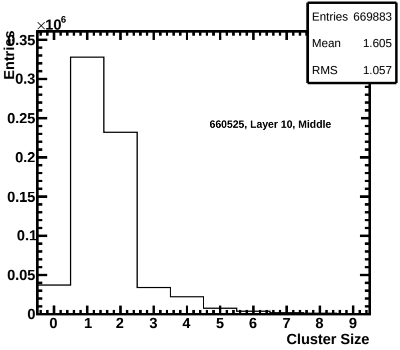
- DHCAL only measures number of hits
- Multiplicity $\mu$ and efficiency $\epsilon$ depend on many factors
  - Temperature
  - Pressure
  - Voltage
  - ...
- Temperature stabilized by tent and AC

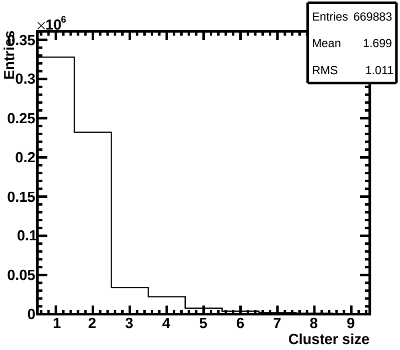


# Determination of Efficiency and Multiplicity

- Lose pre-selection for muon events based on number of active layers ($> 30$) and total number of hits ($< 150$)
- For each layer finds mip stub candidate in neighboring layers ($\pm 3$ layers, min 4 valid clusters)
- Only use clusters with 3 or less hits
- Straight line fit to verify mip stub and identify intersection with layer of interest
- Determine if nearby cluster exists in layer of interest
- Efficiency $\epsilon$: fraction of events with cluster found
- Multiplicity $\mu$: mean cluster size for events with cluster found
- Ignore if intersection is a module border or has been identified as dead or noisy

# Example Histogram



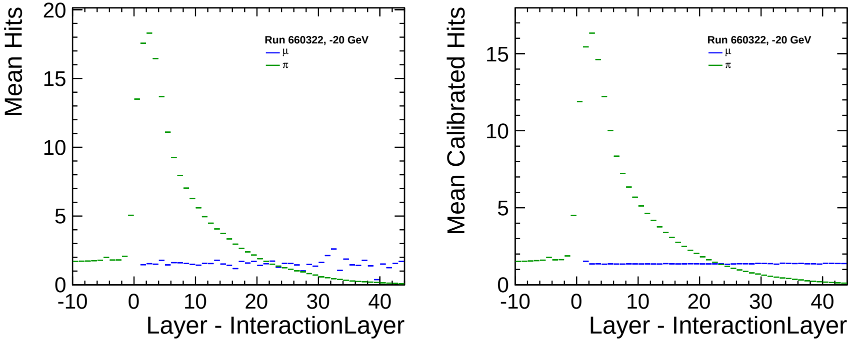




- Extract efficiency as $(N_{\text{total}} - N_0)/N_{\text{total}}$
- Extract multiplicity as mean excluding bin 0
- Determined for each module in each layer

# Calibration Procedure

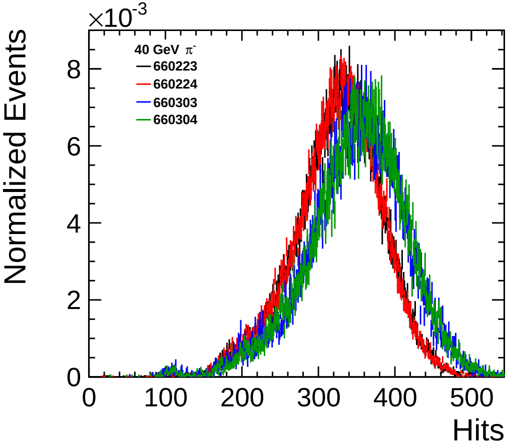
- Correct each hit for its local efficiency and multiplicity to nominal values:
$$N_{\text{hits}}^{\text{calibrated}} = \alpha \sum_i^N \frac{\mu_0 \ \epsilon_0}{\mu_i \ \epsilon_i}$$
- $\mu_i$ and $\epsilon_i$ are determined for each module in each layer from muons within the run: works well only for central module
- Could use temperature and pressure dependence to correct for run conditions and use single calibration set $\Rightarrow$ need to remove voltage dependence
- $\mu_0$ and $\epsilon_0$ are the nominal values, determined as average from all modules and layers in all dedicated muon runs

# Longitudinal Shower Profiles (20 GeV)



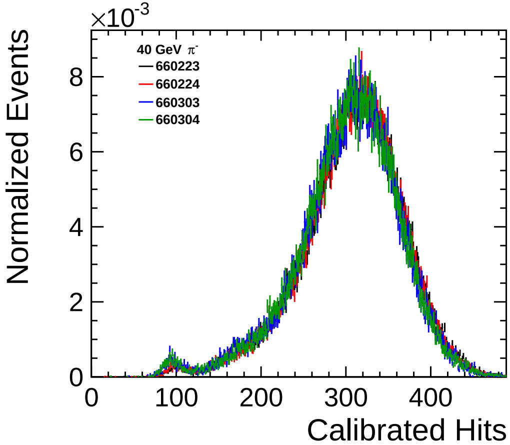

- Layer-to-layer fluctuations are effectively removed

# Total Number of Hits (40 GeV Pions)



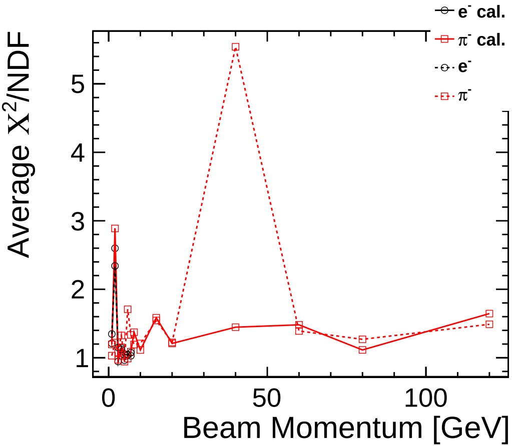

- Runs taken at different time for same beam momentum can show large differences in response
- Remove run-to-run fluctuations by applying calibration

# Total Number of Hits (40 GeV Pions)



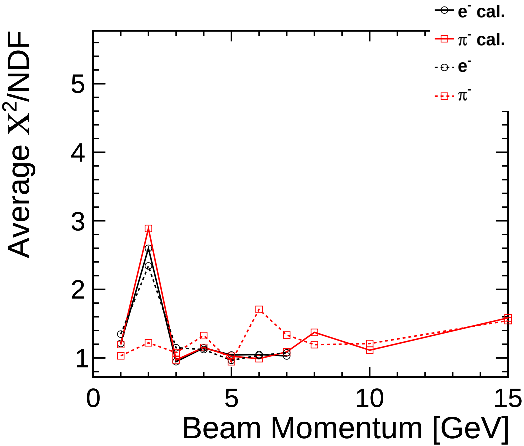

- Runs taken at different time for same beam momentum can show large differences in response
- Remove run-to-run fluctuations by applying calibration

# Impact on Total Number of Hits



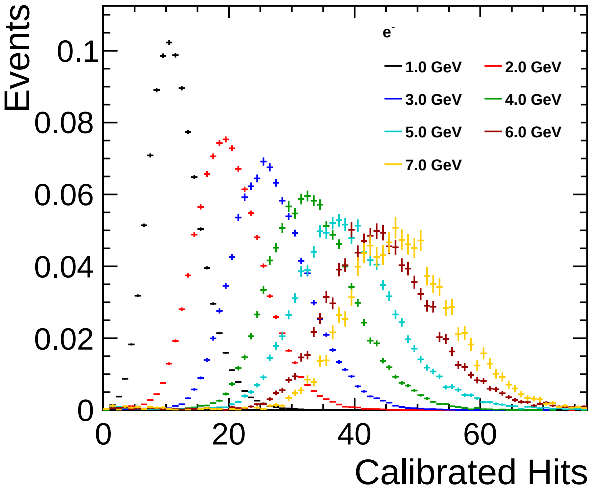

- Calculate average $\chi^2$/NDF (comparing each bin) between the first run and all other runs to estimate impact of calibration

# Impact on Total Number of Hits

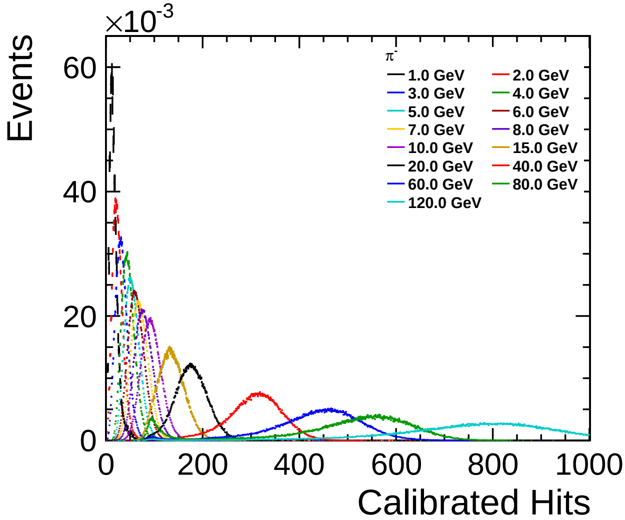



- Calculate average $\chi^2/\mathrm{NDF}$ (comparing each bin) between the first run and all other runs to estimate impact of calibration

# Total Number of Hits (Electrons)



# Total Number of Hits (Pions)

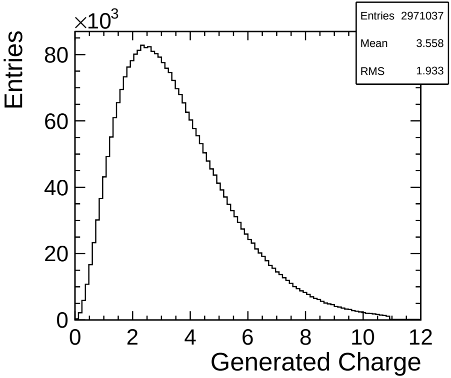


# RPCSim Overview

- Re-implementation of RPCSim as a Marlin processor
- Start from SimCalorimeterHits generated by GEANT4: range cut of 5 µm, store all energy deposits ($\gg 1$ per hit)
- Sort all hits by their layer and treat each layer individually
  - Collect all energy deposits of all SimCalorimeterHits
  - Apply distance cut ($d_{\mathrm{cut}}$): double loop over all combinations of deposits $\Rightarrow$ remove second deposit if the distance is below $d_{\mathrm{cut}}$ (removed from loop)
  - Randomly generate total charge for each remaining deposit (according to data from RPC with analog readout)
  - Distribute charge around deposit position according to charge spread model and add to corresponding pad
  - Create a CalorimeterHit for each pad with charge above threshold
  - Create relation between each digitized hit and all contributing SimCalorimeterHits
- Store CalorimeterHit and LCRelation collection

# RPCSim Charge Generation

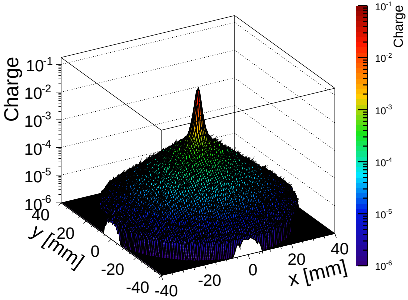
- Charge generation parametrized from data taken with analog RPC
- Total charge generated should correlate with distance of energy deposit from anode plane
- Instead the charge is randomly generated from distribution
- On average this also models the varying charge $z$-position
- Only correct for MIP-like deposits traversing the gas perpendicular to gap



# RPCSim Charge Distribution

- Charge distributed according to charge spread model:
  RPCSim3: $f(r) = Re^{-r/S_1} + (1-R)e^{-r/S_2}$
  Total charge: $Q = \int_0^{2\pi} \int_0^{r_{\text{max}}} \left(Re^{-r/S_1} + (1-R)e^{-r/S_2}\right) d\phi dr$

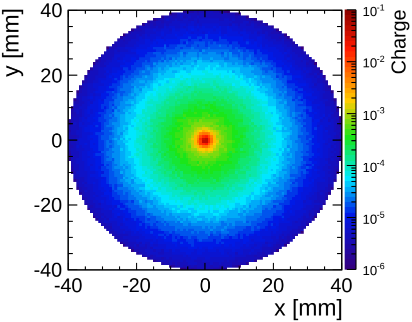
- Need to determine charge integral for each individual pad (Cartesian coordinates)
- No analytical solution
  ⇒ use Monte Carlo integration
- Calculate $N$ (10k–100k) charge fractions for random positions in $r_{\text{max}}$ and add to corresponding pad
- Calculated for each charge deposit
  ⇒ very slow!



# RPCSim Charge Distribution

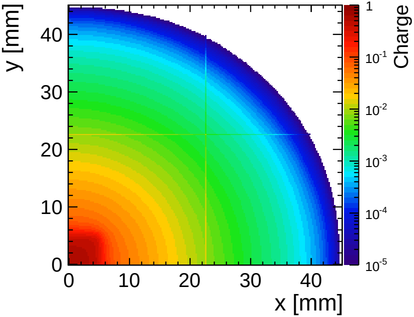
- Charge distributed according to charge spread model:
  RPCSim3: $f(r) = Re^{-r/S_1} + (1-R)e^{-r/S_2}$
  Total charge: $Q = \int_0^{2\pi} \int_0^{r_{\max}} \left(Re^{-r/S_1} + (1-R)e^{-r/S_2}\right) d\phi dr$

- Need to determine charge integral for each individual pad (Cartesian coordinates)
- No analytical solution
  ⇒ use Monte Carlo integration
- Calculate $N$ (10k–100k) charge fractions for random positions in $r_{\max}$ and add to corresponding pad
- Calculated for each charge deposit
  ⇒ very slow!



# RPCSim Charge Distribution

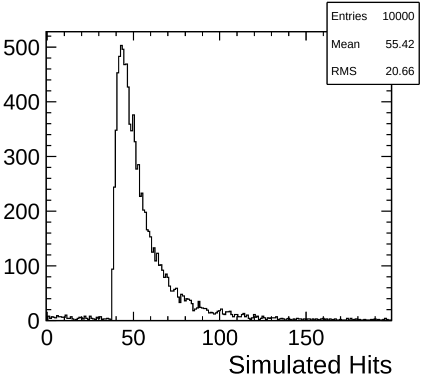
- Do the Monte-Carlo integration once
  $\Rightarrow$ allows much higher values for $N$
- Stores normalized charge integral for a grid of possible relative cell positions ($\vec{p}_{\text{pad}} - \vec{p}_{\text{deposit}}$)
- Take into account symmetry: store only one quadrant of relative positions ($0 \leq x \leq r_{\text{max}} + 0.5$ CellSize)
- Using 2D histogram with $M \times M$ bins
  $\Rightarrow$ look-up using bilinear interpolation for actual relative position

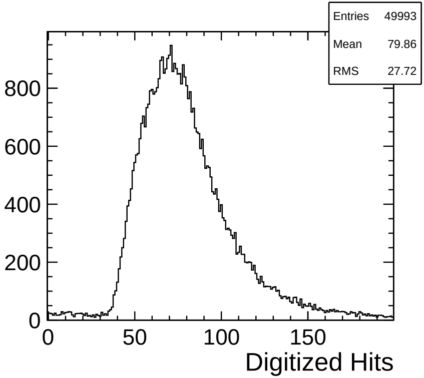



- Using $M = 200$ (225 µm steps)
- Using look up for charge integrals gives a speed-up of more than a factor 10
  From 0.16 s/event to 0.012 s/event
- Once parameters are fixed store histogram in ROOT file

# Result

- Number of digitized hits in muon events




Simulation

Digitization

# Summary and Outlook

- Procedures to clean data dead and noisy cells
  → some effects still need to be understood
- Hit multiplicity and efficiency depend on temperature, pressure and voltage
- Layer calibration to eliminate these fluctuations
  → needs further improvement to properly treat inefficient regions
- Re-implementation of RPCSim as Marlin processor
- Replaced Monte Carlo integration with look-up table
  → improve time by more than factor 10

## Next Steps

- Finalize Mokka model including beam line instrumentation
- Include beam profile and particle angles from wire chamber data in simulation
- Tune digitization model with muon and electron data → prediction for pions
- Add local density based calibration
  → need to treat MIPs different from showers
- Improve particle identification using Monte Carlo