Development of a Two-Dimensional Event Display for Tracking Mode of Q_{weak}

Derek Jones, The George Washington University

A SEARCH FOR NEW PHYSICS — Qweak

Jefferson Lab — Thomas Jefferson National Accelerator Facility

THE GEORGE WASHINGTON UNIVERSITY — WASHINGTON DC

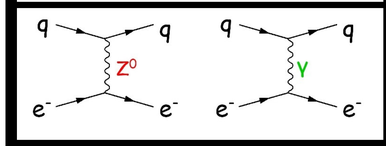
Abstract

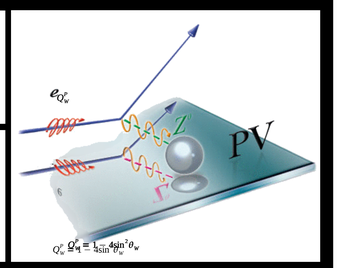
In nuclear physics research, several different types of apparatus are used to detect scattered particles. The computer programs that are often developed to analyze data taken from the detectors are frequently very complex and use so much computing time that the acquisition of new data is reduced or even lost. The Q_{weak} Collaboration at Jefferson Lab uses gas electron multipliers, horizontal drift chambers, vertical drift chambers, and trigger scintillators for calibration. The purpose of this project was to develop a graphical user interface that displays Q_{weak} tracking mode data in analyzable projections with scaled geometry to show hit patterns in real time for each event. By plotting the triggered elements in the tracking hardware for each event, collaborators may be able to derive a viable interpolation of the particle track(s) for each event by using the drawn patterns. The new macro was created with C++ primarily using the ROOT library. Geant4 Monte Carlo simulated data and raw experimental data were used to troubleshoot the program. Gathering data for beam production analysis was controlled by the CEBAF Online Data Acquisition system and stored into ROOT files to be read by the event display. Orthographic projections and tracking data were organized to move between events and beam-line regions to depict hit patterns of the particles for each event. Through effective collaboration and testing, hit patterns were successfully displayed with practical features in a user-friendly environment. The macro may also be used by collaborators to provide hardware diagnostics during production. It may be run independently or within the larger Q_{weak} Data Analysis Graphical User Interface.

Q_{weak} Overview

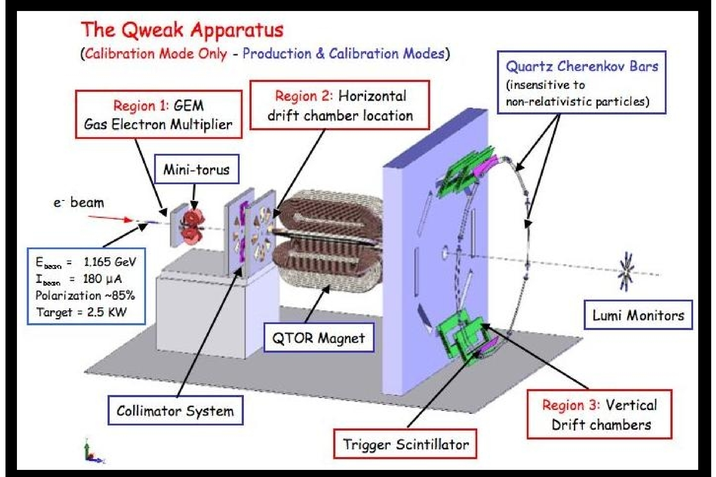
- Conduct very precise study to challenge the Standard Model
- Measure sensitivity of parity violation to weak interaction
- Measure weak charge of proton Q_W^P with ~4% total error
- Determine weak mixing angle $\sin^2\theta_W$ to ~0.3%

$$Q_W^P = 1 - 4\sin^2\theta_W$$

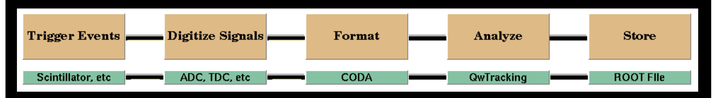




Tracking Systems



DAQ Systems



- CEBAF Online Data Acquisition (CODA)
  - Software toolkit that standardizes and transports events
  - Minimalize commercial software dependance
  - Provides specialized support for some JLab detector systems
- QwTracking Analyzer
  - Process run files from CODA into readable events

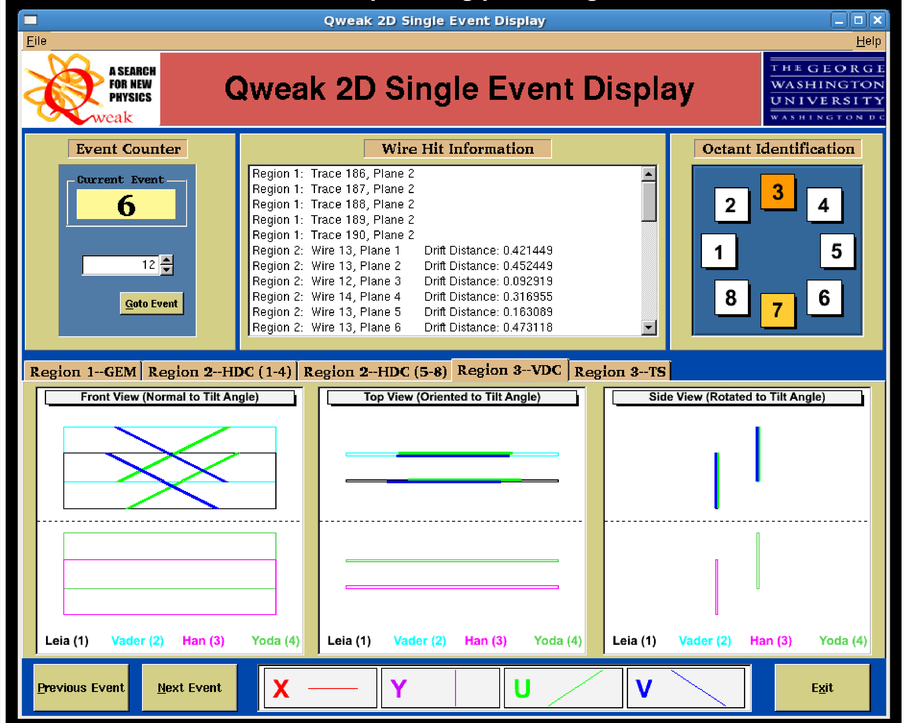
Event Display Features

Viewing Features

- EVENT BOXES: Event Counter, Wire Hit Information, Octant Identification
- REGION BOXES: Orthographic projections of Regions 1-3 in scaled geometry

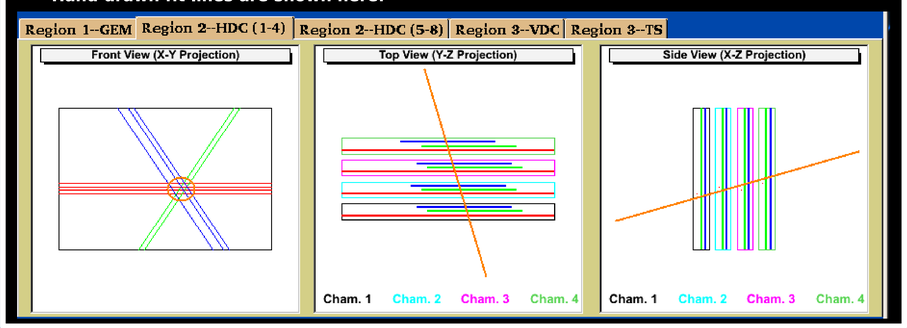
Control Features

- GOTO EVENT: Number entry and button skip to a desired event
- REGION TABS: Move between region views quickly
- PREVIOUS/NEXT EVENT: Move to preceding/proceeding event number

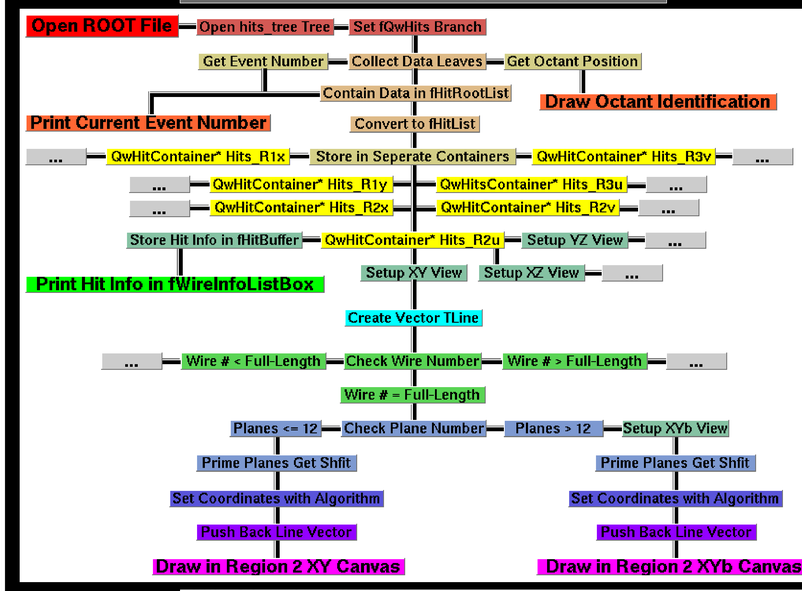


Event Display Analysis

- Hit patterns from the triggered elements allow collaborators to interpolate tracks
  - Hand drawn fit lines are shown here:



Programming Logic



- Code snip of the algorithm for the XY projection of Region 2 hits:

```
fPShift = .5*R2_UVDIST*R2_CM; // Half drift cell distance
if (fWire == R2_FULLWIRE1 || fWire == R2_FULLWIRE2)
  fShiftTrim = 1;
break;
}
if (fPlane > 12) // If on second arm, draw in appropriate tab
  fRegion2bXY->GetCanvas()->cd();
if (fWire < R2_FULLWIRE1){
  Line.SetX1( .5 + (R2_WIDTH*R2_CM*.5) - (R2_UVDIST*R2_CM*fWire) - fPShift);
  Line.SetY1( .5 - (R2_LENGTH*R2_CM*.5));
  Line.SetX2( .5 + (R2_WIDTH*R2_CM*.5));
  Line.SetY2( .5 - (R2_LENGTH*R2_CM*.5) + (((R2_UVDIST*R2_CM*(fWire - fShiftTrim)) + fPShift)*tan(R2_ANGLE*TMath::DegToRad())));
}
else if (fWire < R2_FULLWIRE2){
  Line.SetX1( .5 + (R2_WIDTH*R2_CM*.5) - (R2_UVDIST*R2_CM*fWire) - fPShift);
  Line.SetY1( .5 - (R2_LENGTH*R2_CM*.5));
  Line.SetX2( .5 + (R2_WIDTH*R2_CM*.5) - (R2_UVDIST*R2_CM*(fWire - R2_FULLWIRE1)) - fPShift);
  Line.SetY2( .5 + (R2_LENGTH*R2_CM*.5));
}
else{
  Line.SetX1( .5 - (R2_WIDTH*R2_CM*.5));
  Line.SetY1( .5 + (R2_LENGTH*R2_CM*.5) - (((R2_WIDTH*R2_CM) - (R2_UVDIST*R2_CM*(fWire - R2_FULLWIRE1)) - fPShift)*tan(R2_ANGLE*TMath::DegToRad())));
  Line.SetX2( .5 + (R2_WIDTH*R2_CM*.5) - (R2_UVDIST*R2_CM*(fWire - R2_FULLWIRE1)) - fPShift);
  Line.SetY2( .5 + (R2_LENGTH*R2_CM*.5));
}
```

Conclusions

- Successful reconstruction of hit patterns with many useful control features
- May also be used as a diagnostic tool for tracking hardware
- Further program features could include computer drawn fit lines for tracks
- Compliments a 3D event display (Juan Carlos Cornejo, William & Mary)
- Included as a valuable subsystem in the Q-weak Data Analysis GUI

Acknowledgments

This project was conducted at the Thomas Jefferson National Accelerator Facility. The members of the Q-weak Collaboration, especially the analysis team, were instrumental in the completion of this project. Strong contributions were also made by Marcus Hendricks (original GUI design), Ramesh Subedi (technical support), and Wouter Deconinck (software development).

This project was supported by a grant from the National Science Foundation.