Raster Calibration

Melanie Rehfuss and Burcu Duran

January 10, 2018

Objectives

- ✓ Determine raster constants (adc_per_cm and adc_zero_offsets) from data obtained with 2.218 GeV beam energy

- ✓ Make raster positions consistent with EPICS coordinate system (left-handed coordinate system: +x points beam right)
  - Currently, raster X ~ BPM Y and raster Y ~ -BPM X
  - Confirmed through series of carbon hole runs

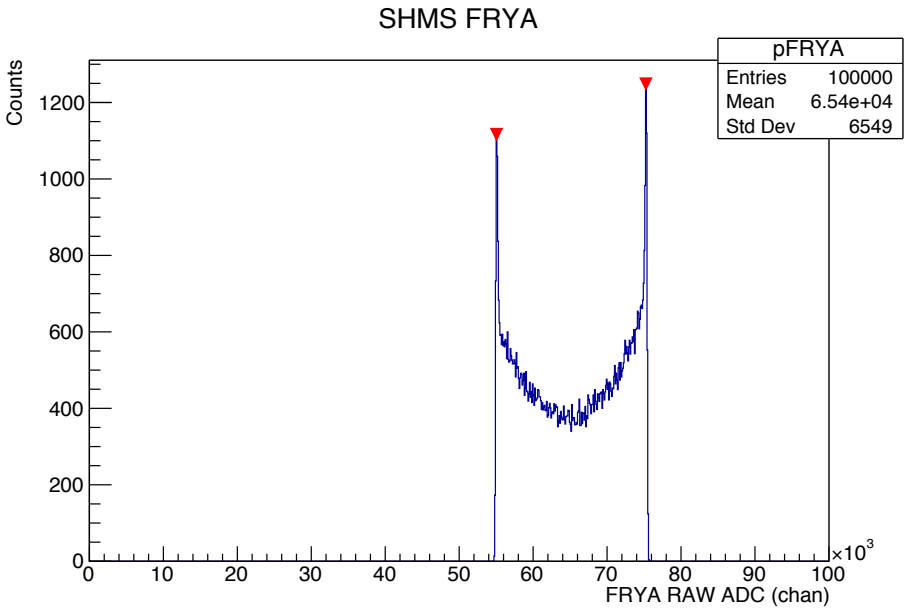
- ✓ Make beam positions calculated through a right-handed coordinate system (+x points beam left)

Used SHMS 1565 run to find new raster constants

- Updated beam energy to 2.218 GeV and used the same formulas as for the KPP calibration:
  - ADC_zero_offset = (lower_edge + upper_edge) / 2
  - ADC_per_cm = (upper_edge - lower_edge) / 0.3 (cm)

```
#include "TSpectrum.h"

{

TFile *f = new TFile("~/Desktop/HallC/hallc_replay/ROOTfiles/
shms_replay_production_1565_100000.root");

h1 = (TH1F*)f->Get("pFRYA");
h1->Draw();

TSpectrum *sp1 = new TSpectrum();
sp1->Search(h1,1.1,"",0.05);
Int_t npeaks = sp1->GetNPeaks();
cout << "Peaks found " << npeaks << endl ;

Double_t *peaks ;
peaks = sp1->GetPositionX();
for (int i=0 ; i<npeaks ; i++)
cout << "Peak at = " << peaks[i] << endl;

}
```

```
Info in <TCanvas::MakeDefCanvas>:  created default TCanvas with name c1
Peaks found 2
Peak at = 55050
Peak at = 75250
Info in <TCanvas::Print>: pdf file /Users/melanierehfuss/Desktop/FRYA.pdf has been created
```



Used SHMS 1565 run to find new raster constants

- New constants added to PARAM/GEN/gbeam.param file

```
15
16 ;            Fast Raster calibration constants
17 ;            ================================
18   gusefr              = 1
19 ; flag if 1 fast raster data used with average beam pos and angles in
20 calculating beam position
21 ; if 0 then only use average beam pos and angles in calculating beam position
22 ; Various fast raster quantities: gUse* are flags
23
24
25  ; gfr_cal_mom  = 6.4  ; = beam momnetum during calibration run
26    gfr_cal_mom  = 2.218  ; = beam momentum during Winter '17 run
27
28  ; gfrxa_adcpercm = 97666.7;
29  ; gfrxb_adcpercm = 97333.3;
30  ; gfrya_adcpercm = 113667;
31  ; gfryb_adcpercm = 114333;
32
33  ; gfrxa_adc_zero_offset = 67800;
34  ; gfrxb_adc_zero_offset = 69050;
35  ; gfrya_adc_zero_offset = 67700;
36  ; gfryb_adc_zero_offset = 67000;
37
38   ; raster constants from Fall '17 Commissioning shms run 1565
39
40    gfrya_adcpercm = 56333.3;
41    gfryb_adcpercm = 56000;
42    gfrxa_adcpercm = 67000.0;
43    gfrxb_adcpercm = 62666.6;
44
45    gfrya_adc_zero_offset = 67200;
46    gfryb_adc_zero_offset = 65650;
47    gfrxa_adc_zero_offset = 65200;
48    gfrxb_adc_zero_offset = 67050;
49
50 ; positions of FR magnets relative to target
51   gfrx_dist   = 1375    ; cm
52   gfry_dist   = 1337    ; cm
53
54
```

Switched X and Y Rasters to be consistent with EPICS BPM coordinate system

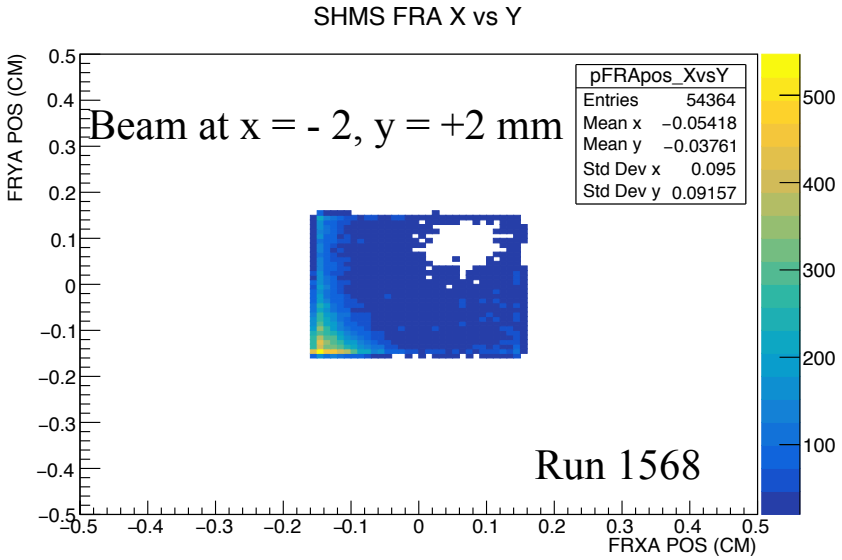
- Modified THcRaster::Decode() method in THcRaster.cxx

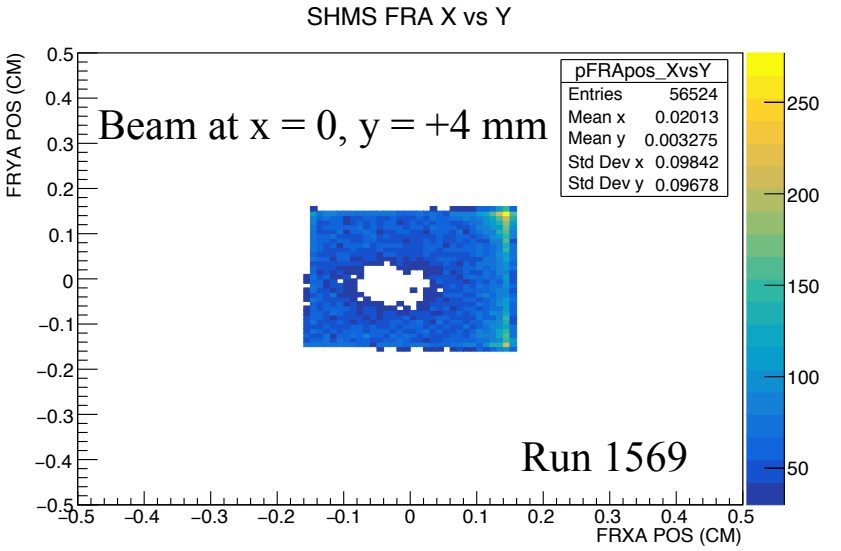
```
for(Int_t ielem = 0; ielem < frPosAdcPulseIntRaw->GetEntries(); ielem++) {
    Int_t    nraster        = ((THcSignalHit*) frPosAdcPulseIntRaw->ConstructedAt(ielem))->GetPaddleNumber() - 1;
     Double_t pulseIntRaw  = ((THcSignalHit*) frPosAdcPulseIntRaw->ConstructedAt(ielem))->GetData();
     if (nraster ==0) FRYA_rawadc = pulseIntRaw;
     if (nraster ==1) FRXA_rawadc = pulseIntRaw;
     if (nraster ==2) FRYB_rawadc = pulseIntRaw;
     if (nraster ==3) FRXB_rawadc = pulseIntRaw;
}
```

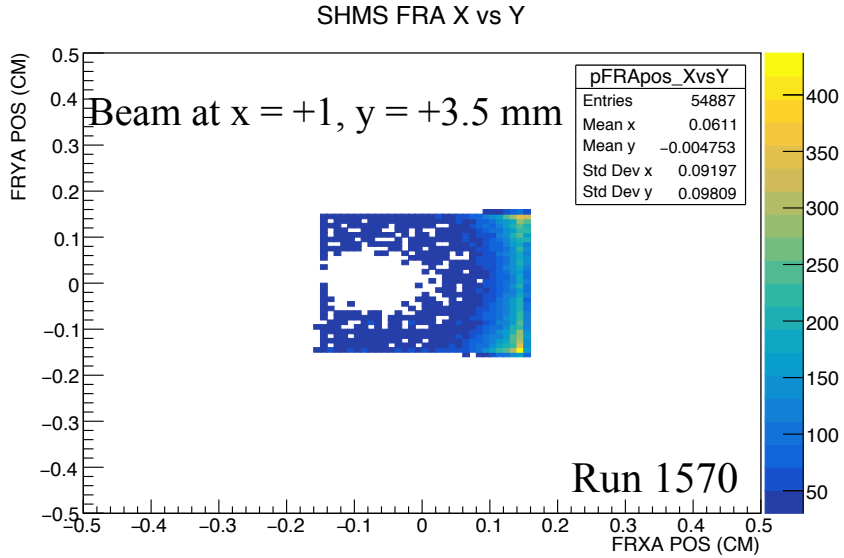
```
#*************
# Fast Raster
#*************

# Assume FADC range is set to 1V and the integration is 25 channels
# Offset in the FADC is about 0.122V or 500 channels.

formula FRXAraw_V P.rb.raster.frxaRawAdc*(1./4096.)/25.
formula FRXBraw_V P.rb.raster.frxbRawAdc*(1./4096.)/25.
formula FRYAraw_V P.rb.raster.fryaRawAdc*(1./4096.)/25.
formula FRYBraw_V P.rb.raster.frybRawAdc*(1./4096.)/25.

formula FRXApos P.rb.raster.fr_xa
formula FRXBpos P.rb.raster.fr_xb
formula FRYApos P.rb.raster.fr_ya
formula FRYBpos P.rb.raster.fr_yb

TH1F pFRXAraw 'SHMS FRXA Raw; FRXA RAW ADC (chan); Counts' P.rb.raster.frxaRawAdc 1000 0 100000
TH1F pFRXAraw_V 'SHMS FRXA Raw; FRXA RAW ADC (Volts); Counts' FRXAraw_V 1000 0 1.0
TH1F pFRXBraw 'SHMS FRXB Raw; FRXB RAW ADC (chan); Counts' P.rb.raster.frxbRawAdc 1000 0 100000
TH1F pFRXBraw_V 'SHMS FRXB Raw; FRXB RAW ADC (Volts); Counts' FRXBraw_V 1000 0 1.0
TH1F pFRYAraw 'SHMS FRYA Raw; FRYA RAW ADC (chan); Counts' P.rb.raster.fryaRawAdc 1000 0 100000
TH1F pFRYAraw_V 'SHMS FRYA Raw; FRYA RAW ADC (Volts); Counts' FRYAraw_V 1000 0 1.0
TH1F pFRYBraw 'SHMS FRYB Raw; FRYB RAW ADC (chan); Counts' P.rb.raster.frybRawAdc 1000 0 100000
TH1F pFRYBraw_V 'SHMS FRYB Raw; FRYB RAW ADC (Volts); Counts' FRYBraw_V 1000 0 1.0

TH1F pFRXA 'SHMS FRXA; FRXA ADC (chan); Counts' P.rb.raster.frxa_adc 1000 -50000 50000
TH1F pFRXB 'SHMS FRXB; FRXB ADC (chan); Counts' P.rb.raster.frxb_adc 1000 -50000 50000
TH1F pFRYA 'SHMS FRYA; FRYA ADC (chan); Counts' P.rb.raster.frya_adc 1000 -50000 50000
TH1F pFRYB 'SHMS FRYB; FRYB ADC (chan); Counts' P.rb.raster.fryb_adc 1000 -50000 50000

TH1F pFRXApos 'SHMS FRXApos; FRXA POS (CM); Counts' P.rb.raster.fr_xa 1000 -1.0 1.0
TH1F pFRXBpos 'SHMS FRXBpos; FRXB POS (CM); Counts' P.rb.raster.fr_xb 1000 -1.0 1.0
TH1F pFRYApos 'SHMS FRYApos; FRYA POS (CM); Counts' P.rb.raster.fr_ya 1000 -1.0 1.0
TH1F pFRYBpos 'SHMS FRYBpos; FRYB POS (CM); Counts' P.rb.raster.fr_yb 1000 -1.0 1.0

TH2F pFRAraw_XvsY 'SHMS FRA Raw X vs Y; FRXA RAW ADC (Volts); FRYA RAW ADC (Volts)' FRXAraw_V FRYAraw_V 1000 0 1.0 1000 0 1.0
TH2F pFRBraw_XvsY 'SHMS FRB Raw X vs Y; FRXB RAW ADC (Volts); FRYB RAW ADC (Volts)' FRXBraw_V FRYBraw_V 1000 0 1.0 1000 0 1.0

TH2F pFRApos_XvsY 'SHMS FRA X vs Y; FRXA POS (CM); FRYA POS (CM)'
P.rb.raster.fr_xa P.rb.raster.fr_ya 1000 -1 1 1000 -1.0 1.0
TH2F pFRBpos_XvsY 'SHMS FRB X vs Y; FRXB POS (CM); FRYB POS (CM)'
P.rb.raster.fr_xb P.rb.raster.fr_yb 1000 -1 1 1000 -1.0 1.0
```

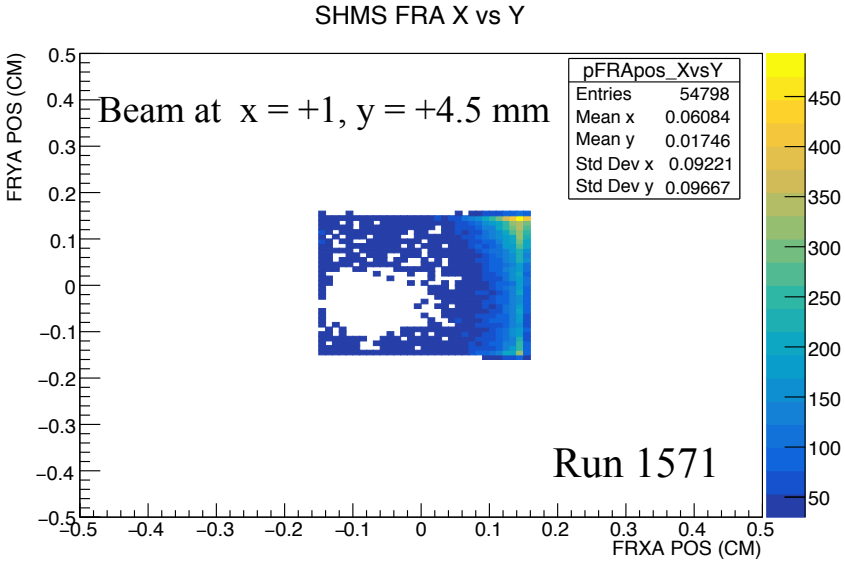
- Added ADC (adc_raw – offset) and 2D position histograms to DEF-files/SHMS/PRODUCTION/RASTER/praster.histos.def

SHMS Carbon Hole Runs

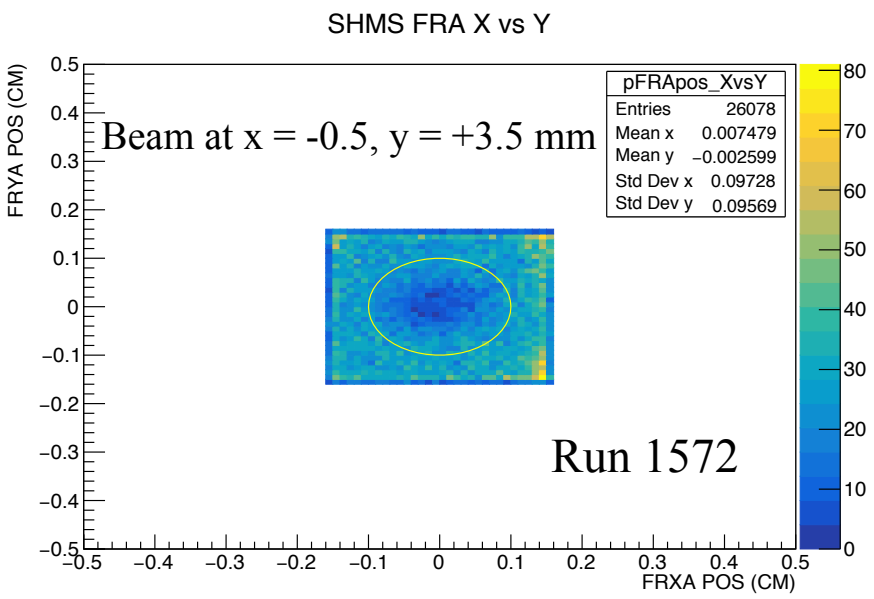


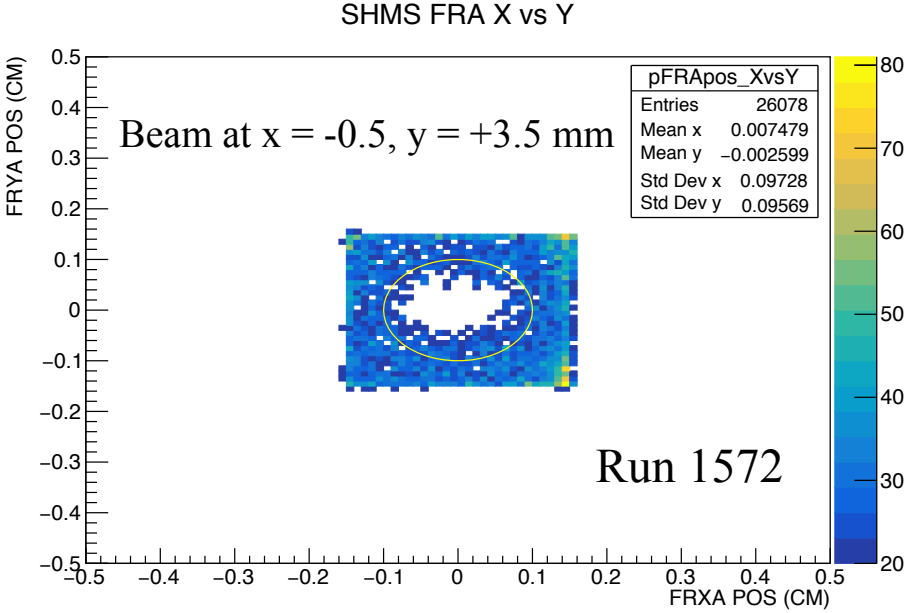






Carbon Hole Centered with SHMS Run 1572





Beam Position Made Right-Handed

- Modified THcRaster::Process() method in THcRaster.cxx so that +x points beam left

```
  Double_t tt;
  Double_t tp;
  if(fgusefr != 0) {
    fPosition[1].SetXYZ((-1)*(fXA_pos+fgbeam_xoff), fYA_pos+fgbeam_yoff, 0.0);
    tt = (-1)*(fXA_pos/fgfrx_dist+fgbeam_xpoff);
    tp = fYA_pos/fgfry_dist+fgbeam_ypoff;

  } else {      // Just use fixed beam position and angle
    fPosition[0].SetXYZ((-1)*fgbeam_xoff, fgbeam_yoff, 0.0);
    tt = (-1)*fgbeam_xpoff;
    tp = fgbeam_ypoff;
  }
  fDirection.SetXYZ(tt, tp ,1.0); // Set arbitrarily to avoid run time warnings
  fDirection *= 1.0/TMath::Sqrt(1.0+tt*tt+tp*tp);

  return 0;
}
```