FNAL beam test data analysis
A Hands-on session at CMS Upgrade School

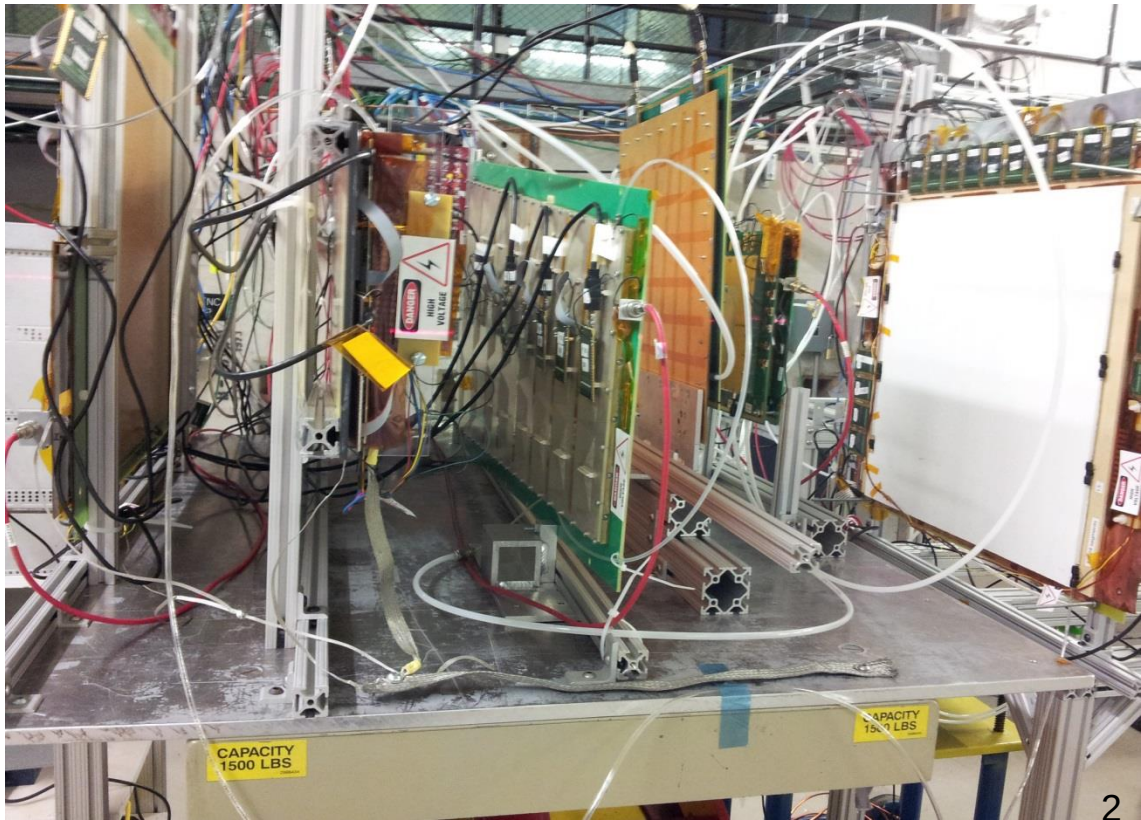
Aiwu Zhang

Florida Institute of Technology

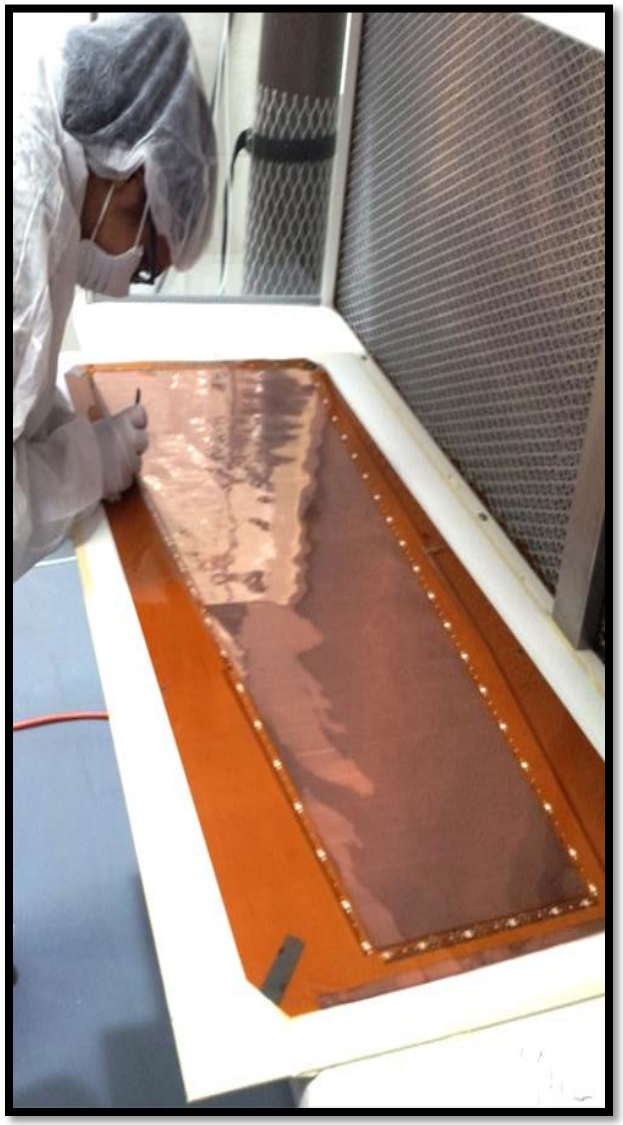
On behalf of the CMS GEM collaboration

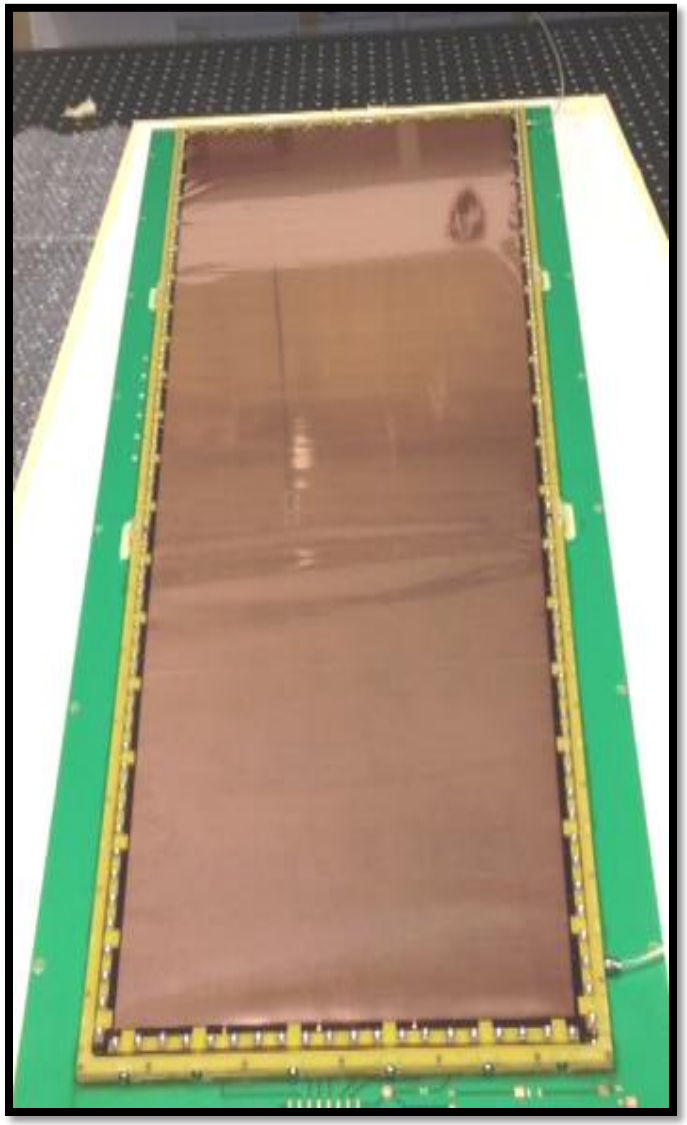
FNAL beam test information

- FIT & U.Va operated 10 triple-GEM detectors in Oct. 2013 at Fermilab Test Beam Facility (FTBF), 4 of them worked as reference detectors (trackers). All detectors worked in Ar:CO_2(70/30) gas.
- All data were taken with 32GeV/c mixed hadrons (p,π,K).
- One CMS GE1/1-III GEM detector was tested in this beam test. 8 APVs were read out one time (1APV/sector).
- For the CMS GEM detector, We did HV scan in the middle-sector 5, and position scan for the entire chamber (except sector 1) (we scanned along three lines: the lower, middle and upper lines).

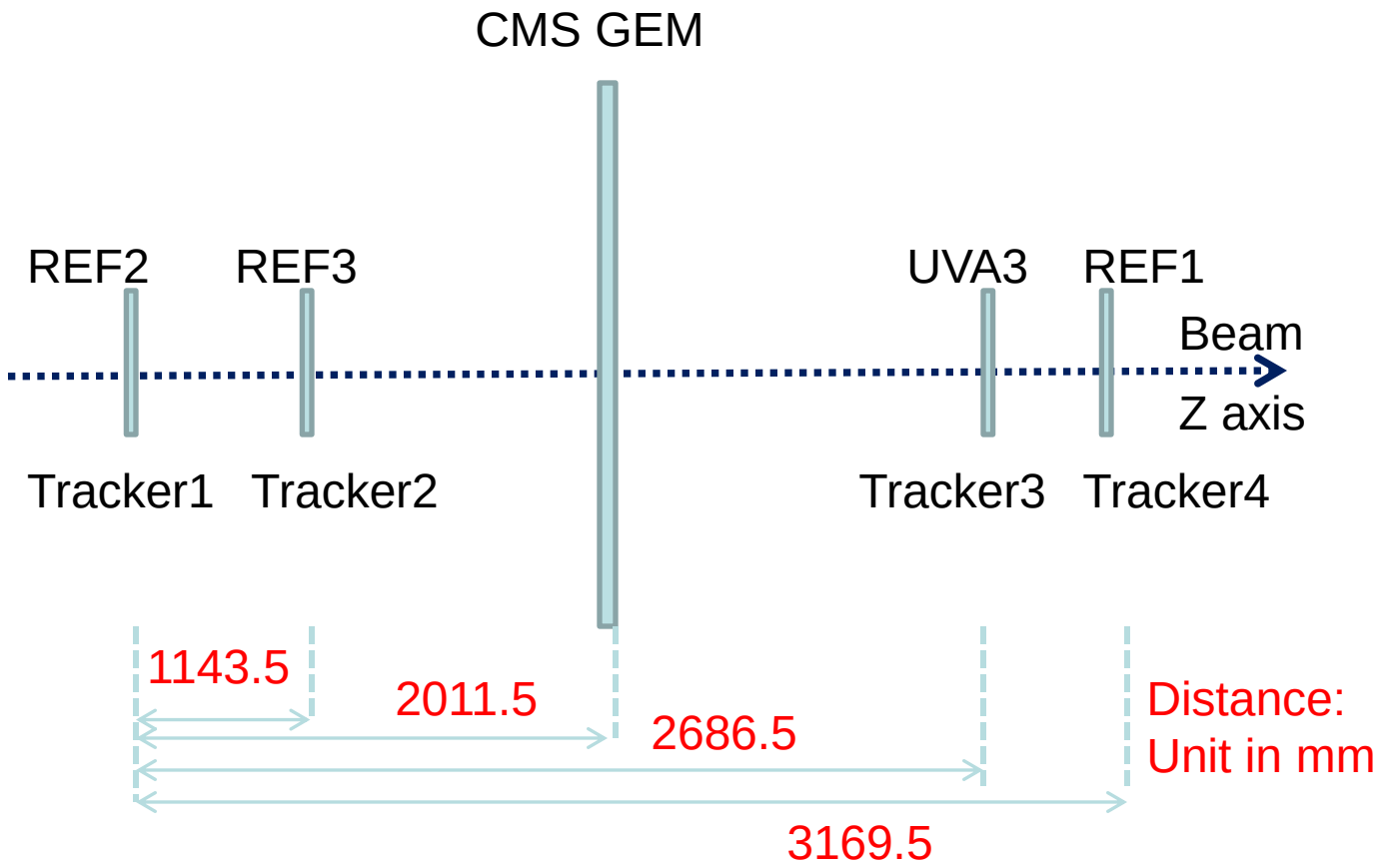


A quick look at the GEM foils & the chamber



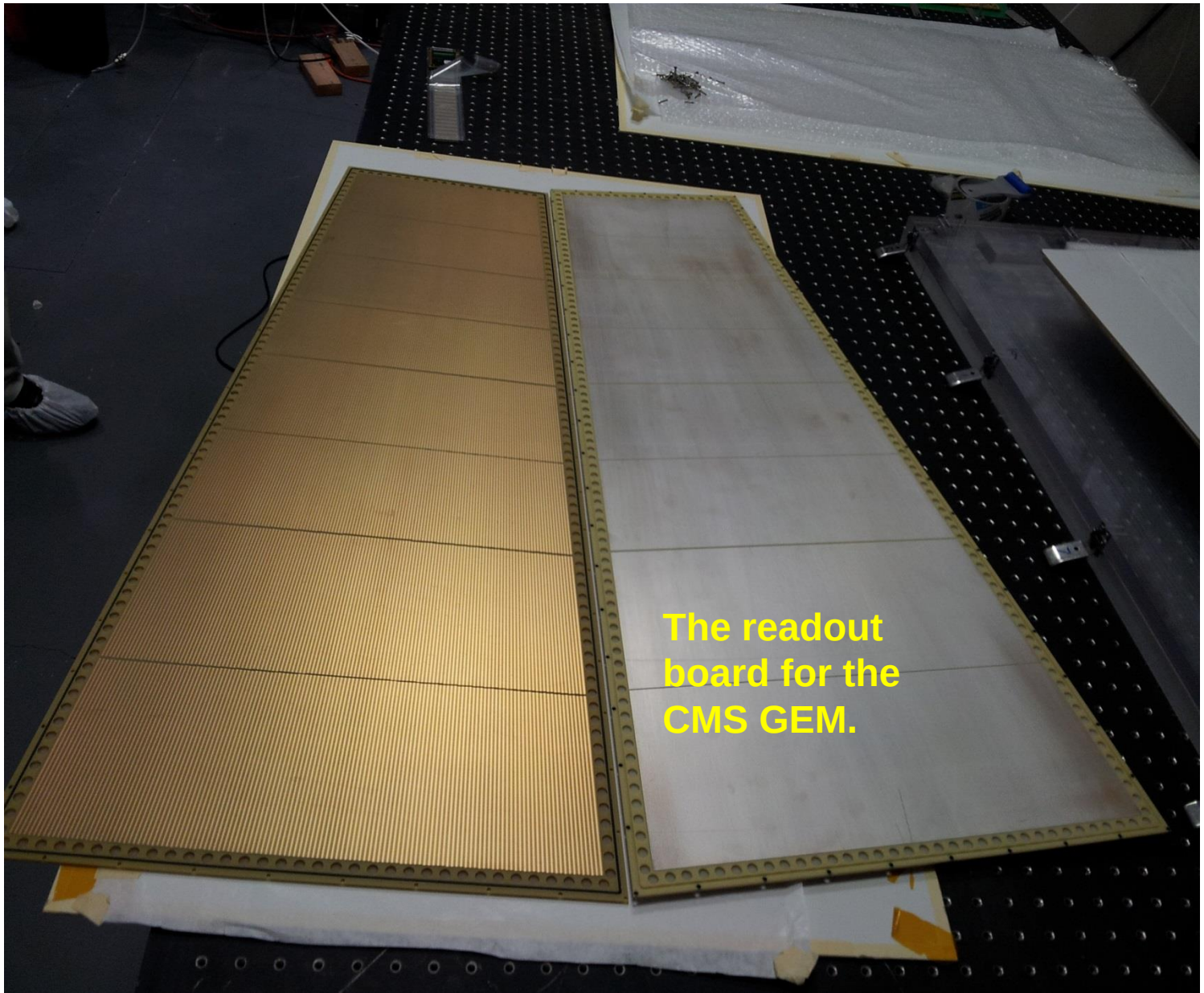


Detector arrangement in the beam test



One more thing to be in mind:
-> the CMS GEM detector has a trapezoidal shape, the (1 dimensional) readout strips run in radial direction and can measure polar phi-coordinate. We measure hit (we often say cluster) positions in polar coordinates, so we'll transfer Cartesian to polar system for the trackers in some of the analysis, for instance, resolution studies.

A quick look at readout strips



A quick look at the raw data

```
EventNb 3 ⟶ Event number
ETA05 2 370.706 42.3435 105.264 106 97.85 105 272.856
REF1X 3 519.867 -18.5821  81.0447 80  106.392 81  283.868 82  129.608
REF1Y 3 1226.25 5.14787 140.37  139 79.0594 140 614.82  141 532.367
REF2X 3 305.978 3.37365 135.934 135 99.4558 136 127.224 137 79.2983
REF2Y 3 283.757 9.39132 150.978 150 75.5257 151 138.86  152 69.3718
REF3X 2 227.502 -1.25998  124.35  124 147.864 125 79.6377
REF3Y 3 320.561 4.65757 139.144 138 44.327  139 185.77  140 90.4638
VERTI 2 507.993 0 14.757  14  123.421 15  384.572
UVA1X 3 748.372 -33.289 479.278 478 62.5527 479 415.508 480 270.311
UVA1Y 3 432.49  -7.83315  667.917 667 125.291 668 217.767 669 89.4319
UVA3X 3 1547.99 -9.32205  104.195 103 208.174 104 830.001 105 509.814
UVA3Y 3 1348.69 22.1233 182.808 182 466.61  183 674.216 184 207.862
ZZ01M 3 1342.62 5.99074 26.3743 27  575.017 26  695.116 25  72.4907
ZZ02S 1 416.201 5.21063 26  26  416.201
EventNb 4
ETA05 2 409.129 29.28 91.1656 92  67.7716 91  341.357
```

The contents in each line (except the event number line) are:
(1)detector plane name: “ETA05” means sector 5 in the CMS GEM;
(2)number of fired strips in this plane;
(3)cluster charge (in ADC counts);
(4)cluster position in mm;
(5)cluster position in strip number;
(6)Groups of strip number *i* and charge on it, *i=1,2,…,n*, *n* is the number in (2).

- Notes: (a) in one event not all detectors have signal due to inefficiency (and some other reasons like fluctuation, noise, …); (b) please focus on these planes: REF1X(Y), REF2X(Y), REF3X(Y), UVA3X(Y) and “ETA0*i*”(*i=0,1,…,7*).

Select “valid” events

- “Valid” event means an event that creates hits in both trackers and the CMS GEM detector.

- The “*Script_ValidEventSelection.C*” script does this work.

-> This script reads in the raw data file, outputs a new text file which contains the “valid” events that can be studied in further analysis.

-> The cut is basically on the cluster sizes in the detector planes. Cluster size means number of strips that are fired in an event.

-> Total 9 columns in the new output text file, the first 8 columns are clusters positions in trackers (both X and Y planes), the last column is the cluster position in the CMS GEM detector in phi-coordinate (radian).

We’ll go through the script briefly …

Script_ValidEventSelection.C

- To run the script:

(1)“*root -l*” in a terminal; -> open ROOT

(2)“*.L Script_ValidEventSelection.C*”; -> load the script

(3)“*main()*”. -> execute the function

```
20  using namespace std;
21  int main(){
22    const int NN = 15000; //number of total raw events
23    const Double_t planeSize = 102; // planeSize for tra
```

Number of raw events in file

```
28    //-- Raw Text File Name to be processed
29    fstream fin("./FNAL_HVscanData_CMSGEM_textFormat/Cluster_run010_HVScan_3800V_32GeV_20131017_0744am.txt",ios::in);
30    //-- Output text file name for selected events
31    //-- file names can be changed by yourselves.
32    fstream fout("Position_run001_HVScan_UVa3800_25GeV_20131014_344am_Allstrip.txt",ios::out); // All cluster size > 0
33    //fstream fout2("Position_run001_HVScan_UVa3800_25GeV_20131014_344am_2strip.txt",ios::out); // I want only cluster
34    //fstream fout3("Position_run001_HVScan_UVa3800_25GeV_20131014_344am_3strip.txt",ios::out); // I want only cluster
35    //-- ROOT file to save some distributions, 2D histograms, etc.
36    TString rootfile = Form("Histograms_run001_HVScan.root");
```

File names

Selection cuts on cluster sizes

```
170     // -- set number of hits cut conditions
171     Bool_t cutNHitsREF2=kFALSE; if( (NHitsREF2X[i]>0) && (NHitsREF2Y[i]>0) ) {cutNHitsREF2=kTRUE;}//cout<<"REF2 true"<<endl;}
172     Bool_t cutNHitsREF3=kFALSE; if( (NHitsREF3X[i]>0) && (NHitsREF3Y[i]>0) ) {cutNHitsREF3=kTRUE;}//cout<<"REF3 true"<<endl;}
173     Bool_t cutNHitsUVA3=kFALSE; if( (NHitsUVA3X[i]>0) && (NHitsUVA3Y[i]>0) ) {cutNHitsUVA3=kTRUE;}//cout<<"UVA3 true"<<endl;}
174     Bool_t cutNHitsREF1=kFALSE; if( (NHitsREF1X[i]>0) && (NHitsREF1Y[i]>0) ) {cutNHitsREF1=kTRUE;}//cout<<"REF1 true"<<endl;}
175     Bool_t cutNHitsCMS=kFALSE;if( NHitsCMSEta5[i]>0 ){cutNHitsCMS=kTRUE;}
176
177     // -- A valid event should leave hits on all tracker detector planes
178     Bool_t trigger=kFALSE; if(cutNHitsREF2==kTRUE && cutNHitsREF3==kTRUE && cutNHitsUVA3==kTRUE && cutNHitsREF1==kTRUE ) trigger=kTRUE;
179     if(trigger){
180
181         if(NHitsCMSEta5[i]>=1){
182            totalEvents++;
183            if(totalEvents%100==0) cout<<"total valide events: "<<totalEvents<<endl;
184
185            fout<<pREF2X[i]<<"\t"<<pREF2Y[i]<<"\t"<<pREF3X[i]<<"\t"<<pREF3Y[i]<<"\t"<<pUVA3X[i]<<"\t"<<pUVA3Y[i]<<"\t"
186                  <<pREF1X[i]<<"\t"<<pREF1Y[i]<<"\t"<<pCMSEta5[i]<<endl;
```

Spatial resolution study

- Another "*Script_GetResiduals.C*" reads in the new text file, fits tracks and histograms residuals for trackers and the CMS GEM. We'll focus on residual distributions for the CMS GEM.
- The detectors need to be aligned before residuals can be calculated. We ignore the alignment procedure, the final alignment parameters are set in this script. (For HV scan, since detectors were not moved, one group of alignment parameters is OK; for position scan we'll have to do alignment for different positions.)

- In track fitting, we do exclusive and inclusive fits. Exclusive (Inclusive) means hit position in CMS GEM is excluded (included) when fitting a track. So we'll get both exclusive and inclusive residual distributions. The residual distributions are fitted with a double Gaussian function ("*Script_doubleGausFit.C*"), and the primary sigma (width) in each fit will be taken as exclusive and inclusive resolutions ($\sigma_{ex}\ and\ \sigma_{in}$).
- Finally we calculate resolution from the geometric mean of the widths: $\sigma = \sqrt{\sigma_{ex} \times \sigma_{in}}$, and error can be simply propagated.

Script_GetResiduals.C

Same way to run this script!

```
250 int main(){
251
252   string name[1]={
253     "Position_run001_HVScan_UVa3800_25GeV_20131014_344am_Allstrip"
254   };
255   const int evtNb[1]={3283};
256
257   for(int iterNbX=0;iterNbX<1;iterNbX++)
258     {
259      tracking(name[iterNbX],evtNb[iterNbX]);
260     }
261   return 0;
262 }
```

Set file name(s), and event number(s).

Change *"_Residuals_Inclusive.root"* to *"_Residuals_Exclusive.root"* If calculate *ex*clusive residual,

```
138   string outputrootname=thestring+"_Residuals_Inclusive.root";
139   TFile* f = new TFile(outputrootname.c_str(),"recreate");
```

```
196       //fill track in phi direction, exclusive
197       TGraph* g2 = new TGraph();
198       g2->SetPoint(0,0,     phiREF2);
199       g2->SetPoint(1,1143.5,phiREF3);
200       g2->SetPoint(2,2011.5,phiEta5);
201       g2->SetPoint(3,2686.5,phiUVA3);
202       g2->SetPoint(4,3169.5,phiREF1);
203       g2->GetXaxis()->SetRangeUser(-1000,3300);
204       TF1* f2 = new TF1("line2","pol1",0,3200);
```

→

```
196       //fill track in phi direction, exclusive
197       TGraph* g2 = new TGraph();
198       g2->SetPoint(0,0,     phiREF2);
199       g2->SetPoint(1,1143.5,phiREF3);
200       //g2->SetPoint(2,2011.5,phiEta5);
201       g2->SetPoint(2,2686.5,phiUVA3);
202       g2->SetPoint(3,3169.5,phiREF1);
```

For *Inclusive track fit*, please make sure the lines 196—204 are the same as on the left, for *exclusive track fit*, please change those lines accordingly to the right.

Practice with the ValidEventSelection script

-> Within the HV scan data (10 files), use any one file;
-> Select events that has size >= 1 in all tracker planes, and also >=1 in the CMS GEM;
-> Note that we can also study the clusters that have a certain size, eg., 2-strip clusters.

-> Check the output text file;
-> Also a root file is output, some histograms are stored in it. It'll also be interesting to check them.

Practice with the GetResiduals script

-> Use one newly output text files, run the script with “Exclusive” mode and “Inclusive” mode,
-> then read residual widths of both, and calculate geometric mean for the resolution

-> do the same thing for all HV points.

Let’s go ……

Summary

- Resolutions at different HV points can be analyzed with the two scripts.

- The first script is actually can be used to do some other analyses: (1) cluster charge distribution; (2) detection efficiencies; (3) cluster sizes characters; etc… please implement them if you are interested!

Thanks!

Run	HV(V)	Exclusive (μrad)	Ex_err	Inclusive (μrad)	In_err	Geo. Mean	err
10	2900	167.8	2.1	133.8	1.7	149.8	2.7
11	2950	170.5	2	135.8	1.6	152.2	2.5
12	3000	166.3	1.8	132	1	148.2	1.9
13	3050	156.8	1.6	125.1	1.3	140.1	2.0
14	3100	152.1	1.5	120.2	1.2	135.2	1.9
16	3200	140.9	1.3	112.2	1.0	125.7	1.6
17	3250	137.9	1.3	110.7	1.0	123.6	1.6
18	3300	144	1	114.6	1.1	128.5	1.5
19	3350	142.6	1.4	114	1	127.5	1.7