Diamond- and Silicon Strip Detectors
Test Beam Analysis and Results

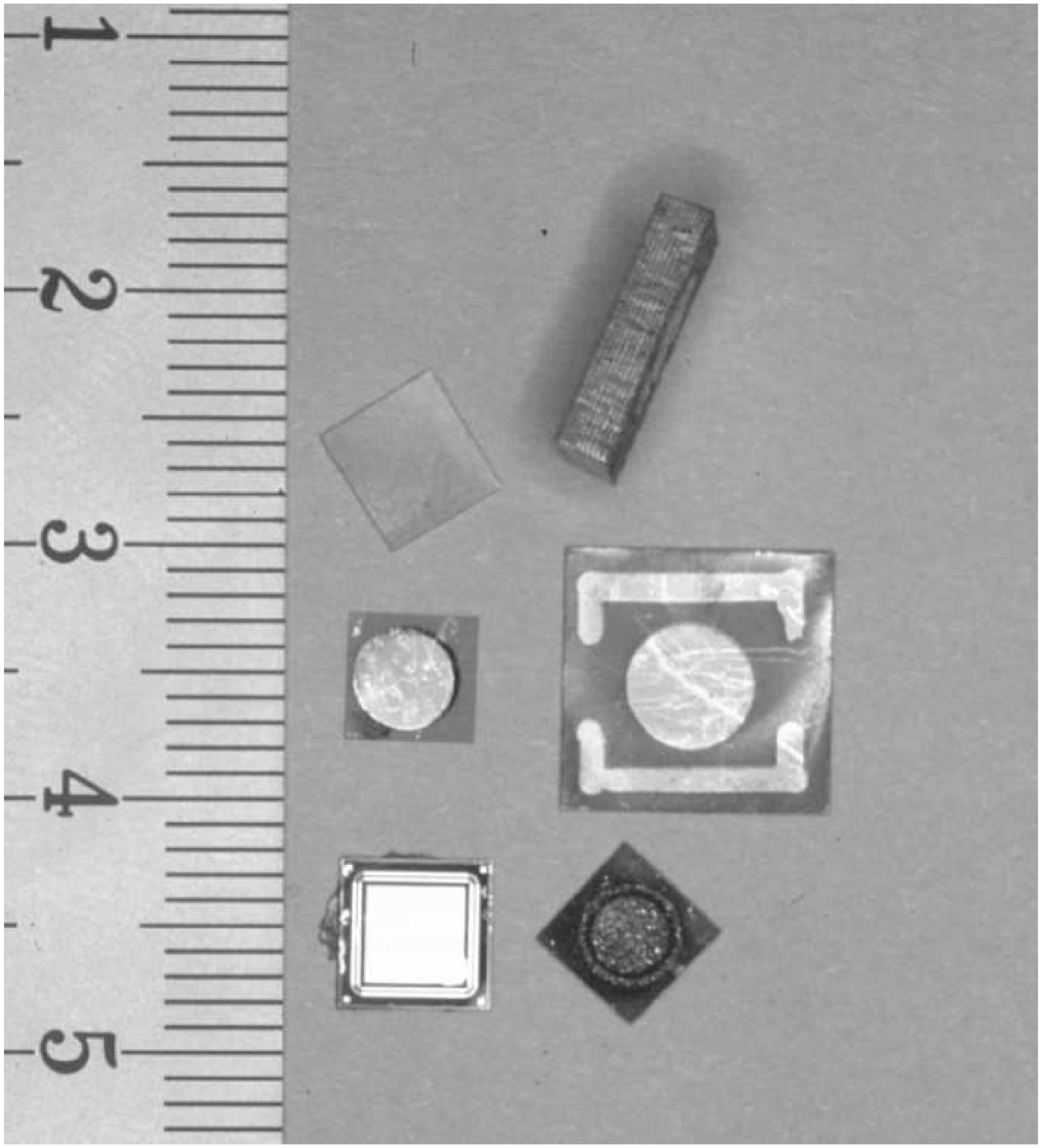
Seminar, Strasbourg, May 6. 1998
Dirk.Meier@cern.ch
with RD42 Diamond-Detector-Collaboration

Contents

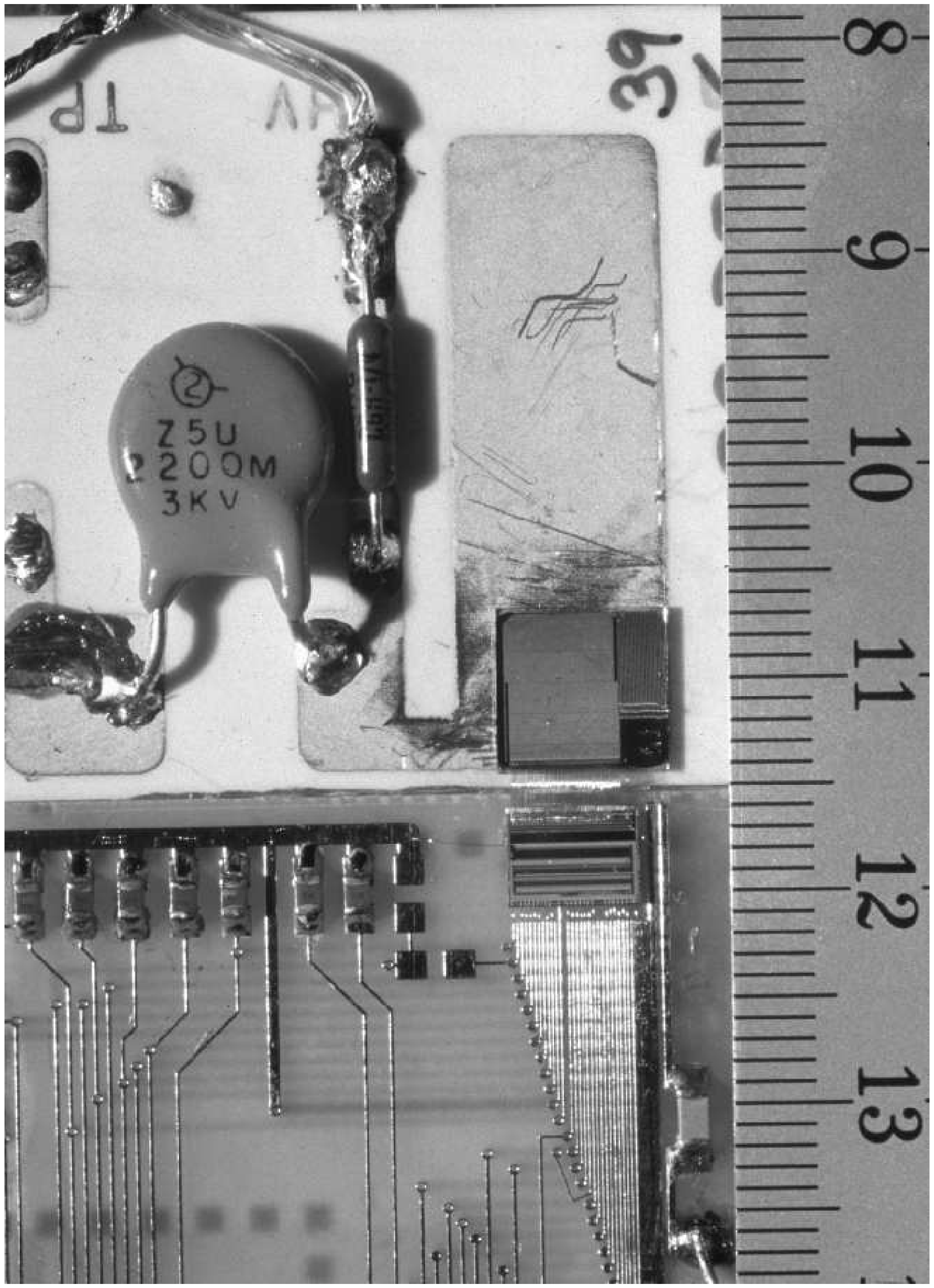
Abstract

Silicon and diamond strip detectors were tested by the RD42 collaboration using a 100 GeV/c pion beam from the SPS at CERN. A silicon beam telescope was used to predict pion tracks and their intersections with detectors under test. The detectors under test were then characterized with regard to their efficiency, charge collection and spatial resolution. Results on beam test analysis will be presented. Emphasis will be on tracking software using the ROOT/C++ framework. The basic aspects of telescope and detector alignement, cluster- and transparent analysis will be given with examples.

Diamond Sensors with Dot Contact

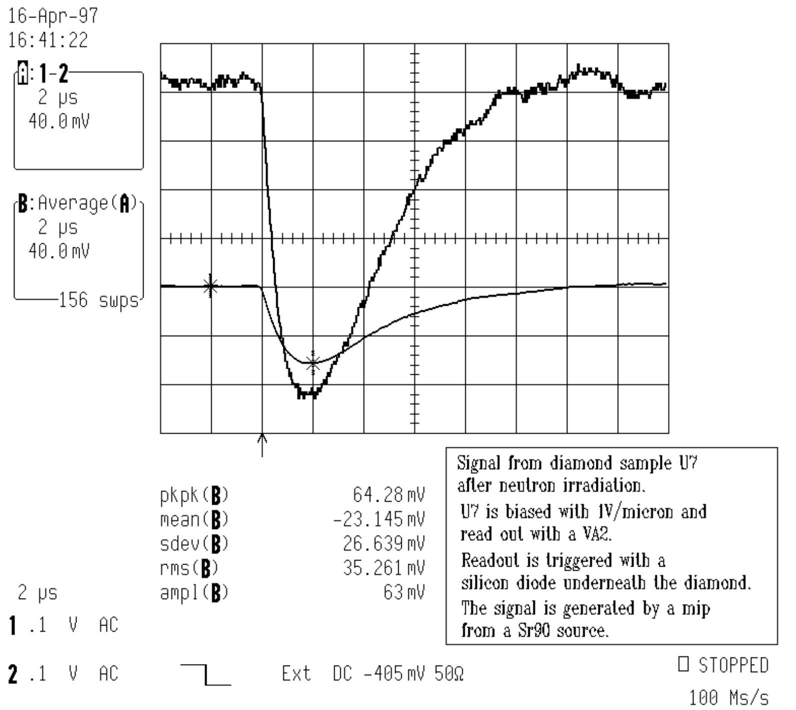


Diamond Strip Detector

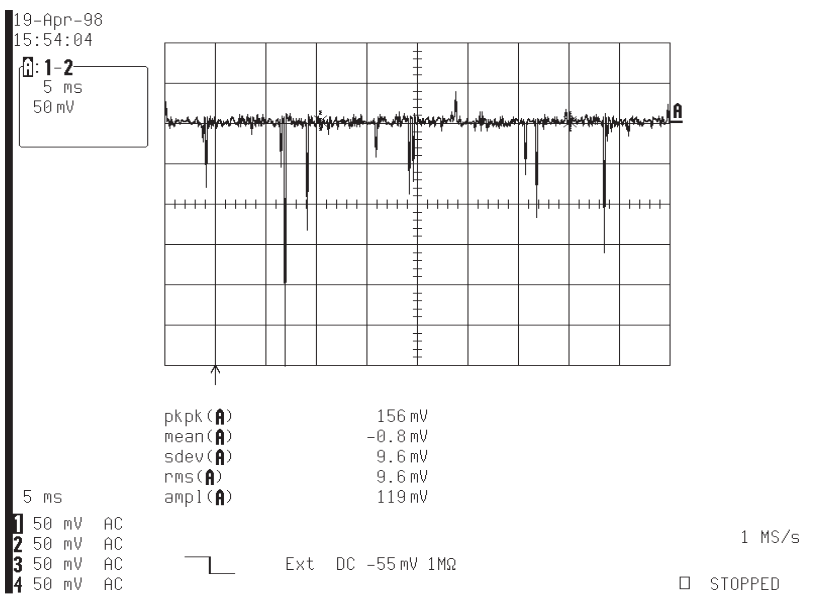


Signals on CVD Diamond Sensor under ^{90}Sr

Diamond Signal, VA2 readout amplifier: 128 channels, charge integration, signal shaping: $\tau_{\text{peak}} = 2\ \mu$s, low noise (measured): ENC$\approx 82e+14\ e$/pF

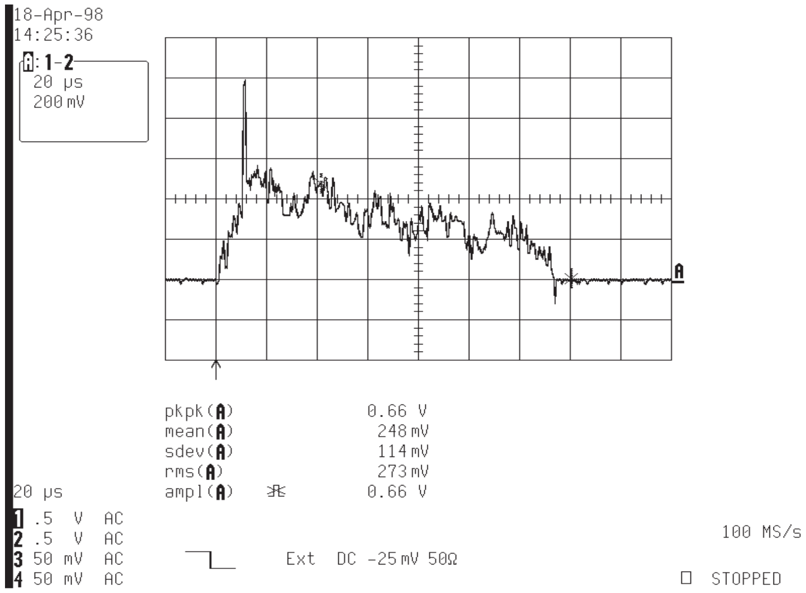


Diamond Tracker, signals: qualitative check under ^{90}Sr source

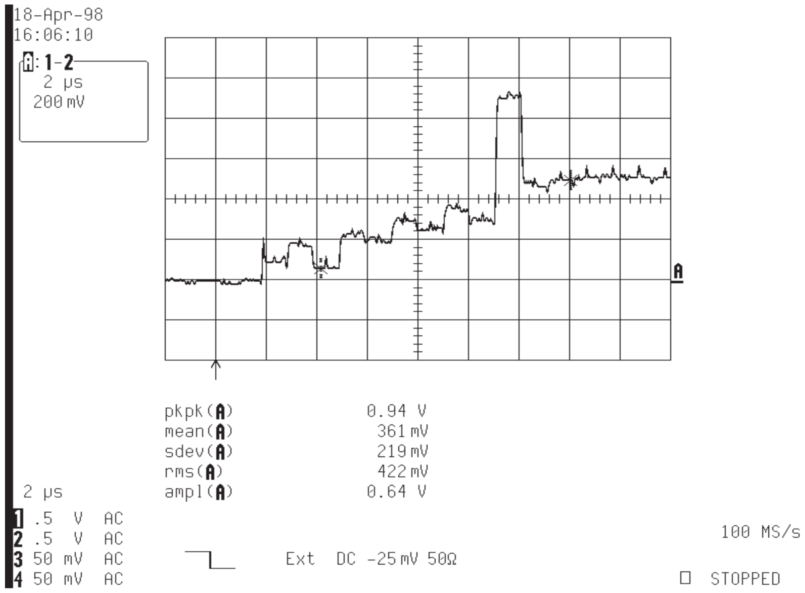


Signal Readout of a Strip Detector

128 channels readout sequence from VA2's multiplexer

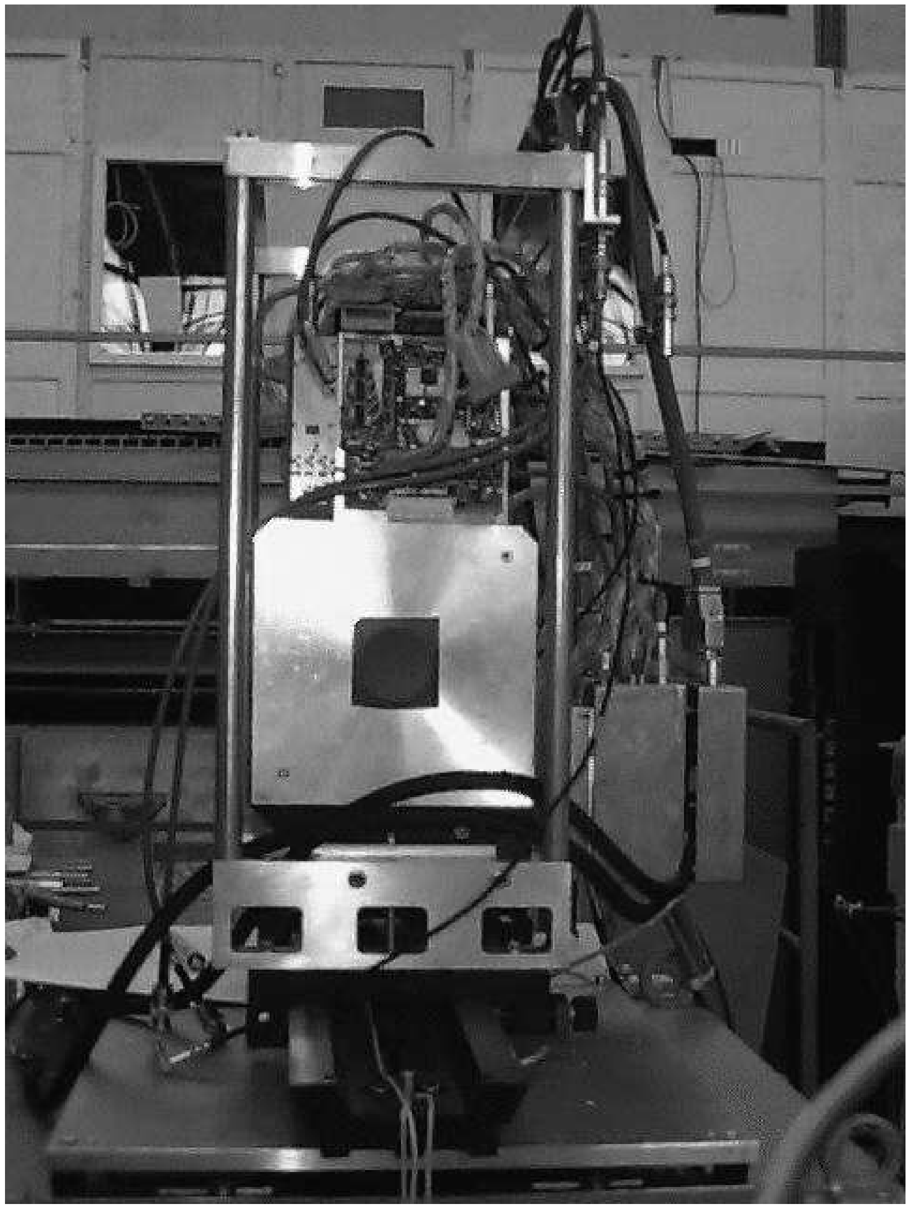


first ≈ 16 channels

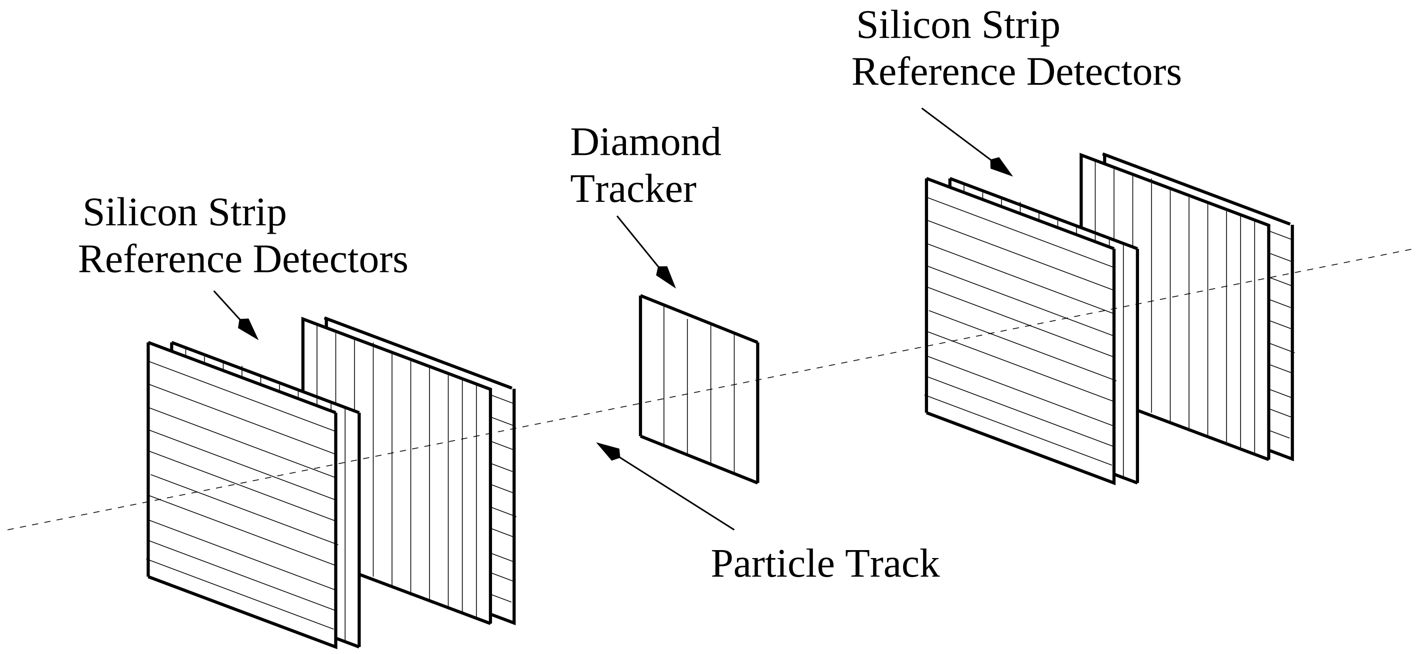


Particle's View

- Front view onto the 'LEPSI' Silicon Beam Telescope



Silicon Telescope



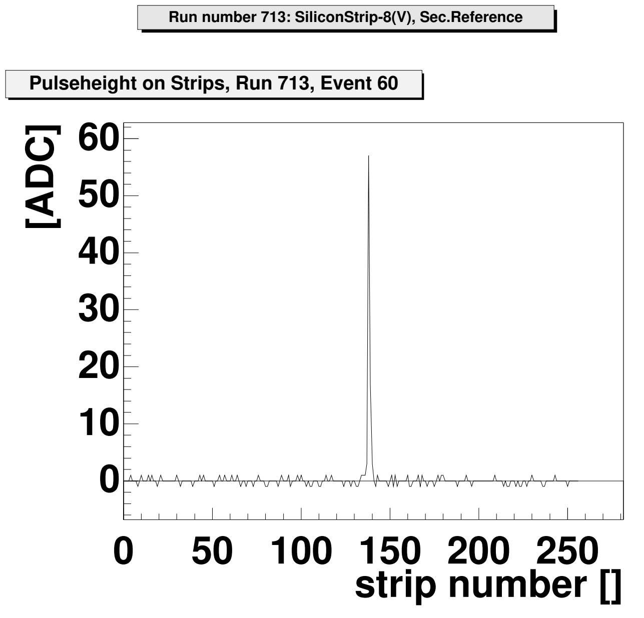
Bits on Tape and Raw Data Dump

- analysis code reads raw data from tape
- hex raw data dump to terminal

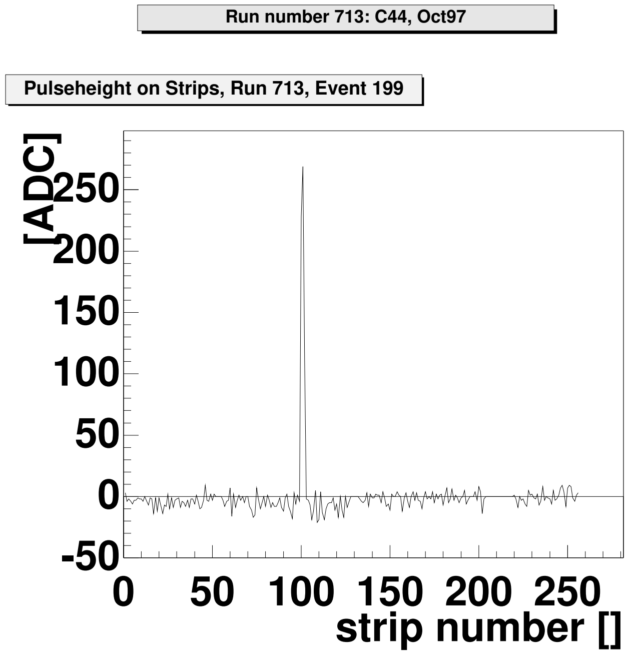
```
->>>> DSession, Event 34798, track -1
->>>> DSession, Event 34799, track -1
Run#: 826, Event#: 34800, Date: 980421, Time[s]: 13040, Length: 1587
 > Sirocco, Id: 1, CpS: 512, CpS': 512, Inputs: 4
 > Sirocco, Id: 2, CpS: 128, CpS': 128, Inputs: 2
 > Sirocco, Id: 3, CpS: 128, CpS': 128, Inputs: 2
 > Sirocco, Id: -1, CpS: 128, CpS': -1, Inputs: -1
Sirocco-A D1 I1 (Raw): c1: 3d,49,47,.. c129: 44,46,43,..
.......................c257: 2c,2f,2e,.. c385: 3e,3a,3b,..
.......................c513: ff,ff,ff,.. c641: ff,ff,ff,..
Sirocco-A D1 I2 (Raw): c1: 37,40,45,.. c129: 35,2f,32,..
.......................c257: 47,41,43,.. c385: 39,33,33,..
.......................c513: ff,ff,ff,.. c641: ff,ff,ff,..
Sirocco-A D1 I3 (Raw): c1: 32,35,3a,.. c129: 30,30,2d,..
.......................c257: 38,34,36,.. c385: 30,35,36,..
.......................c513: ff,ff,ff,.. c641: ff,ff,ff,..
Sirocco-A D1 I4 (Raw): c1: 38,41,42,.. c129: 41,3e,3b,..
.......................c257: 36,2f,33,.. c385: 36,2f,2f,..
.......................c513: ff,ff,ff,.. c641: ff,ff,ff,..
Sirocco-B D1 I1 (Raw): c1: 4b1,5c4,5b4,.. c129: ffff,ffff,ffff,..
Sirocco-B D1 I2 (Raw): c1: 18f,2a5,1fe,.. c129: ffff,ffff,ffff,..
Sirocco-B D2 I1 (Raw): c1: 440,3f6,455,.. c129: ffff,ffff,ffff,..
Sirocco-B D2 I2 (Raw): c1: 494,496,490,.. c129: ffff,ffff,ffff,..
Sirocco-B D3 I1 (Raw): c1: ffff,ffff,ffff,.. c129: ffff,ffff,ffff,..
Sirocco-B D3 I2 (Raw): c1: ffff,ffff,ffff,.. c129: ffff,ffff,ffff,..
DSession, Event 34800, no track
->>>> DSession, Event 34800, track 118539
```

Online Display: Pulsheights on Planes

Hit in a
Silicon Plane

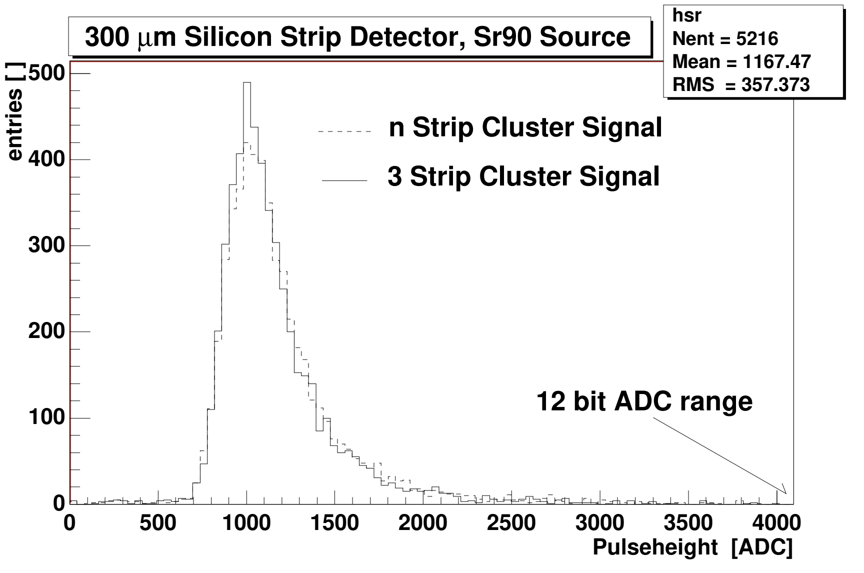


Hit in a
Diamond Plane

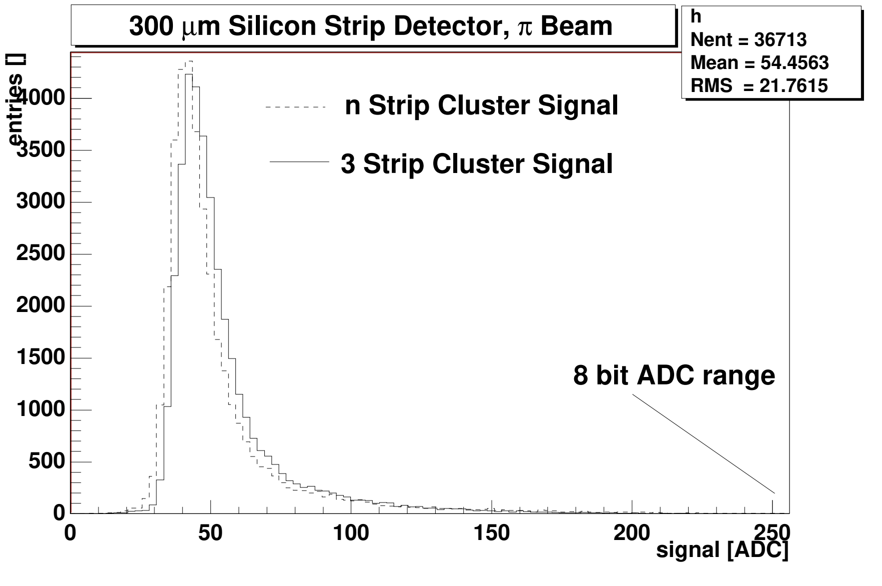


Cluster Signals from ^{90}Sr and π-Beam

Silicon Cluster Signal from ^{90}Sr β-Source



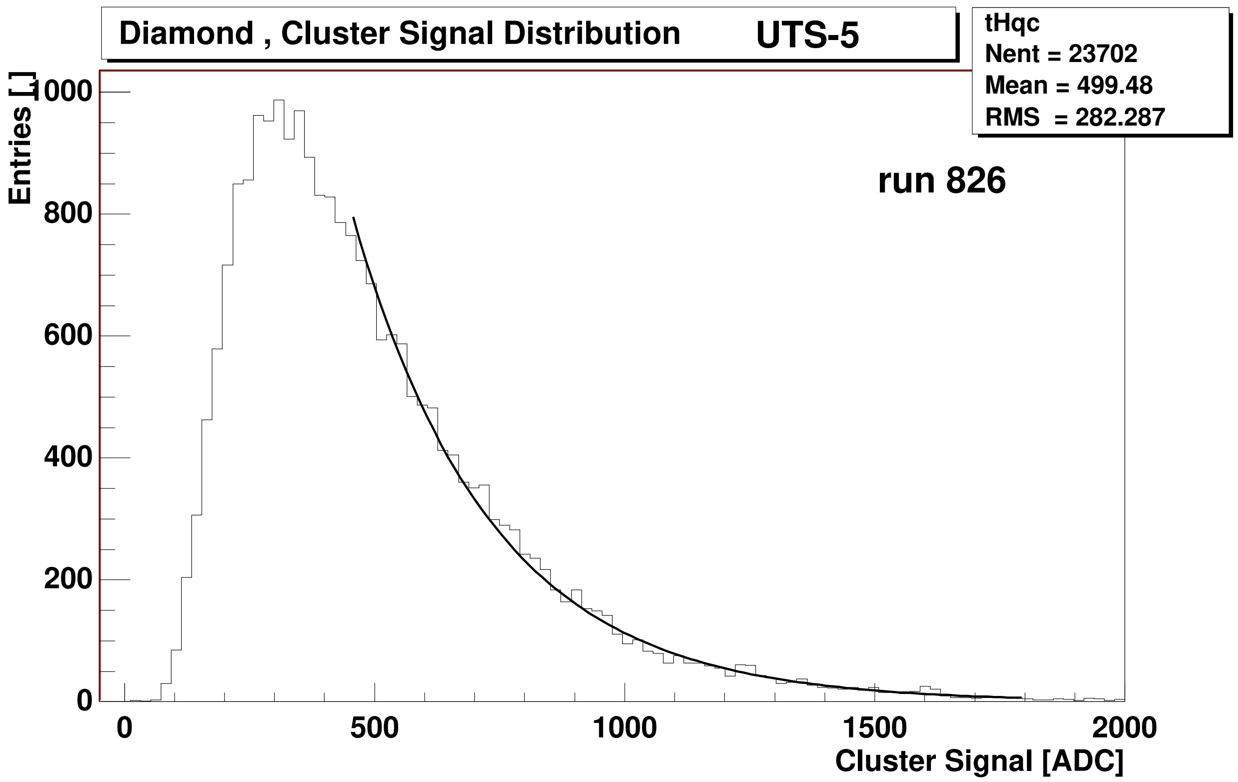
Silicon Cluster Signal in 100 GeV/c π-Beam



- $\mathrm{FWHM}/S_{\mathrm{mp}} \approx 0.45$

Cluster Signal Distribution π-Beam

- Hit Cluster: search hightes pulse on strips above $S/N \geq 6$
- add pulseheights on adjacent strips with $S/N \geq 2$

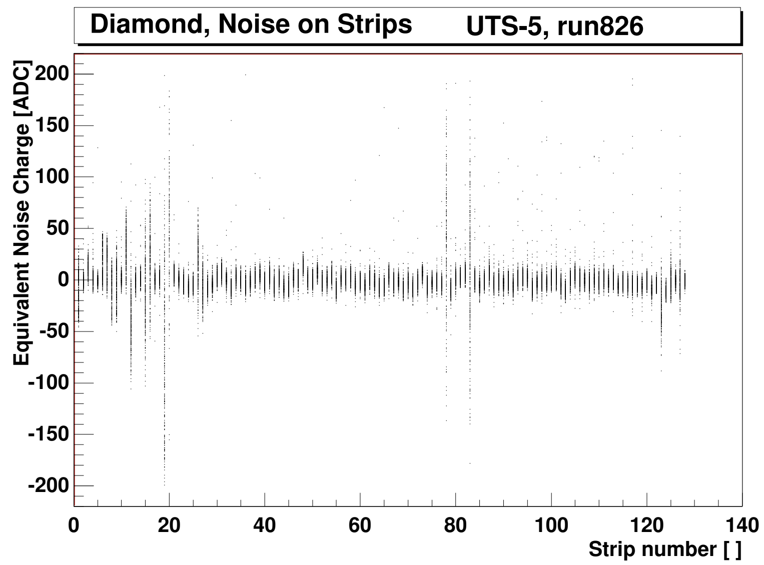


- mean signal $\approx$ 500 ADC, most probable $\approx$ 320 ADC
- gain $\approx$ 16 e/ADC $\Longrightarrow$ mean signal 8000 e
- $\frac{S_{\rm mp}}{\rm ENC} \approx 46$
- $\mathrm{FWHM}/S_{\rm mp} \approx 1.3$

Noise on Diamond Tracker

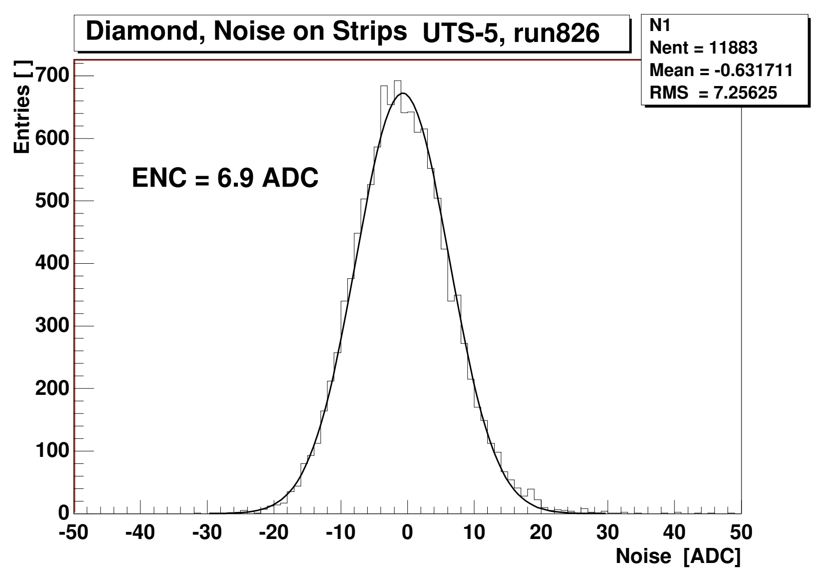
Noise on Strips

- few noisy strips

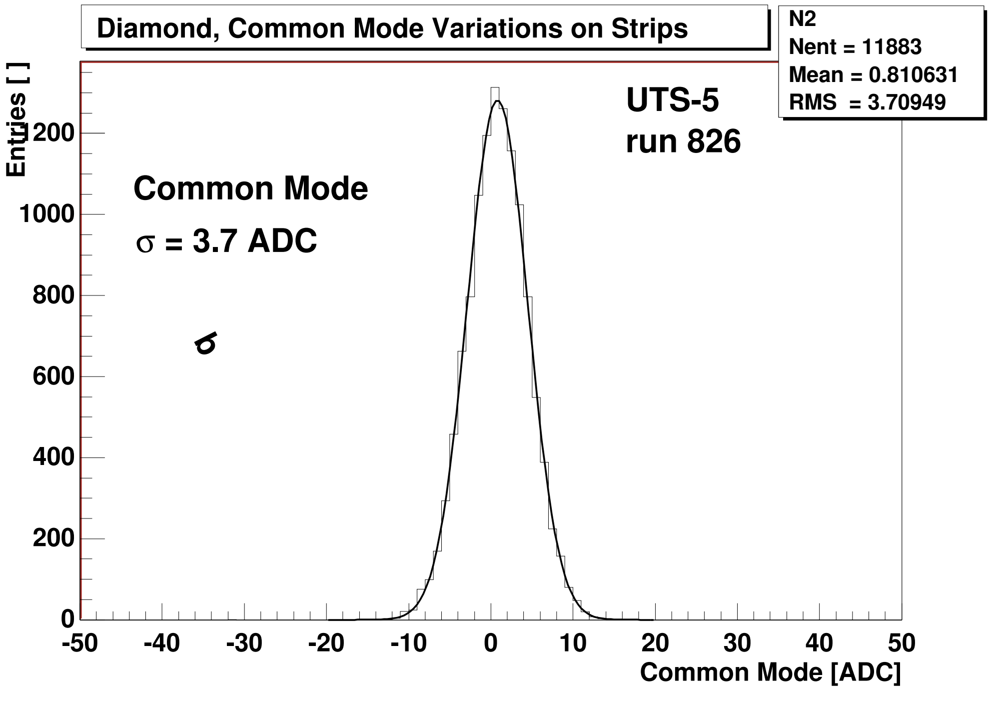


Noise Distribution

- ENC=6.9 ADC
- corresponds to ≈ 120 e

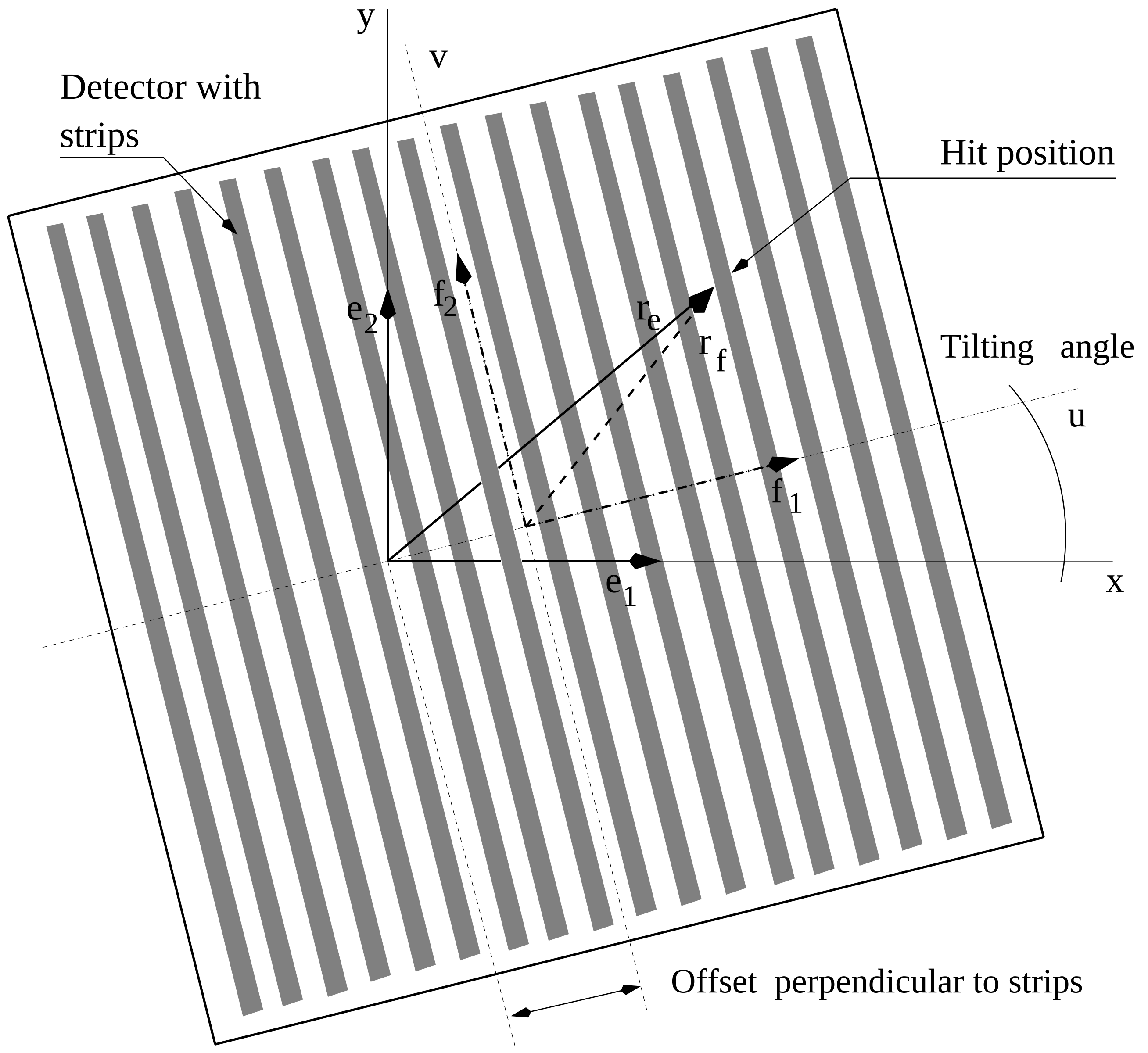


Common Mode Variations

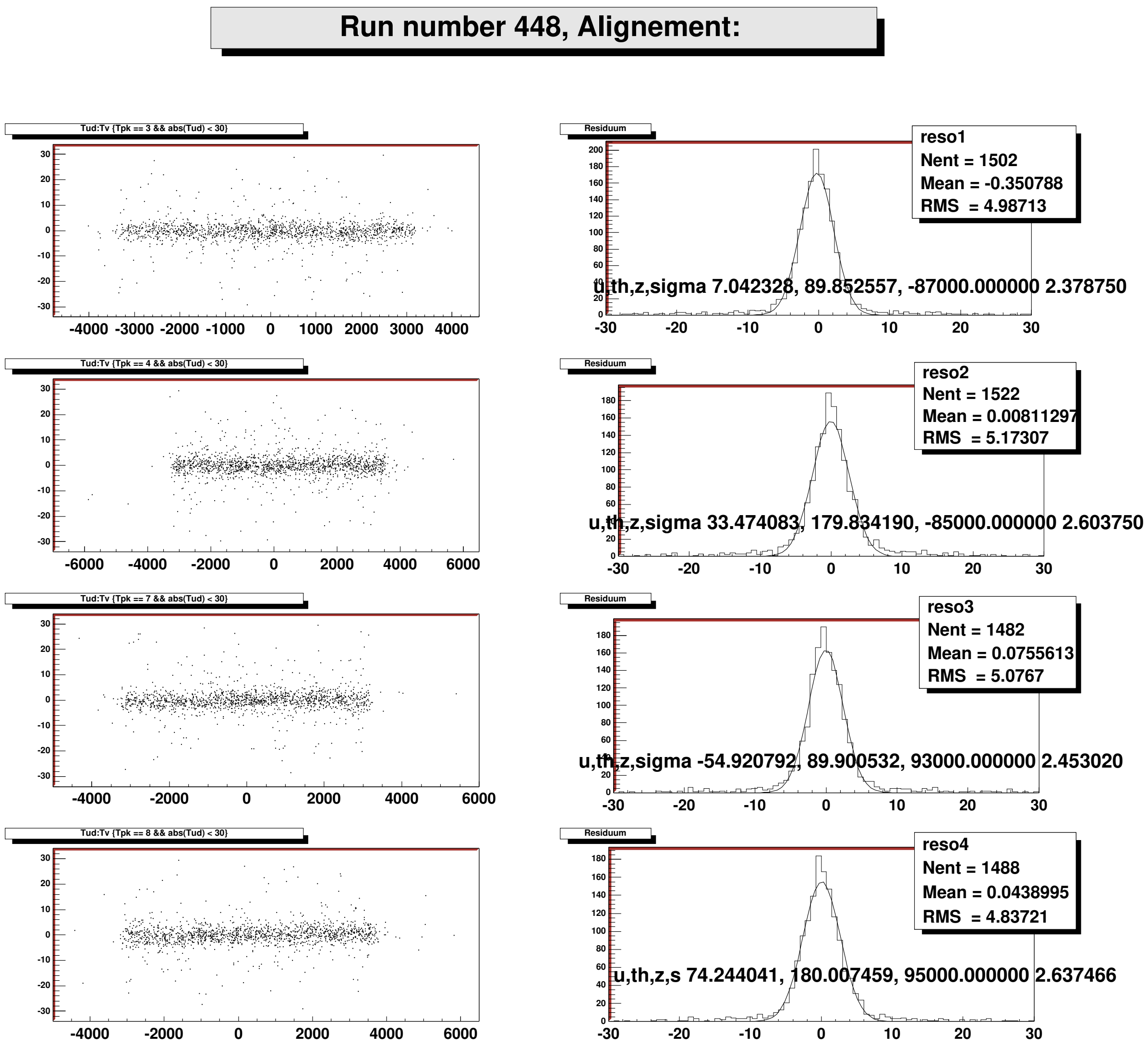


- total noise = (strip noise)^2 + (common mode)^2

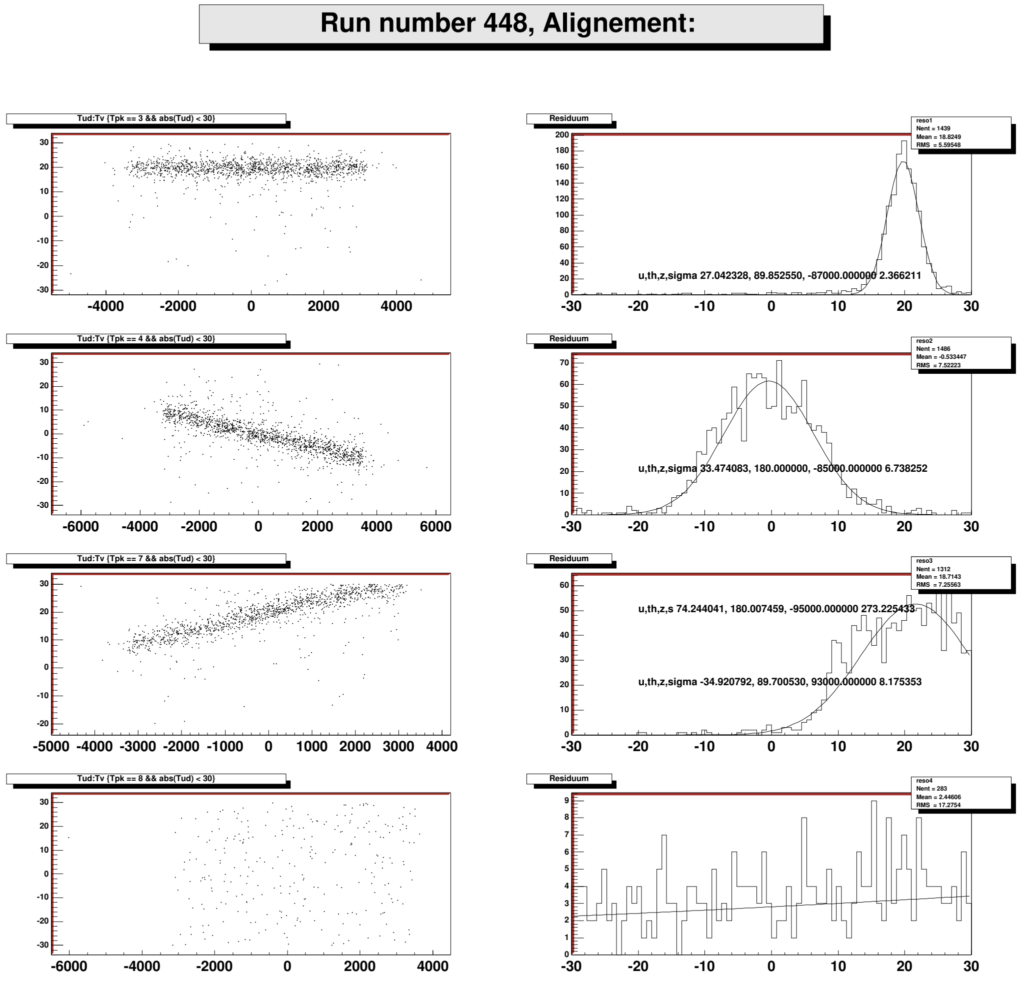
Coordinate Systems



Four Plane Alignement

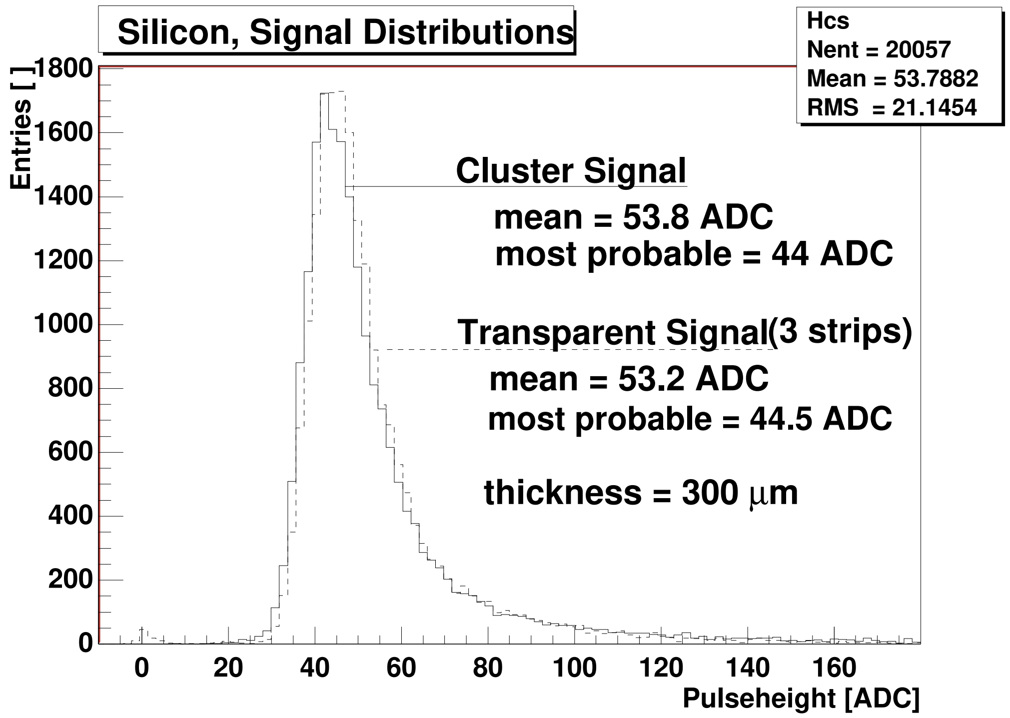


Mis-Alignment



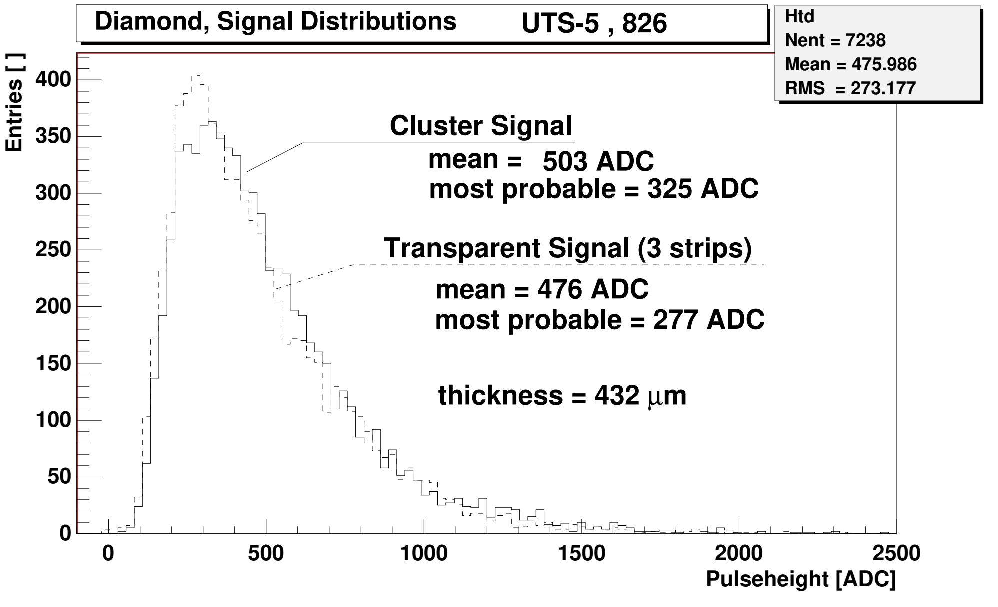
- $\{u_{\text{hit}}\}$ out of beam tracks
- detector tilted around beam axis
- wrong position along beam axis

Silicon, Transparent- and Cluster Signal



- transparent and cluster signal overlay
- $\frac{S_{\mathrm{mean}}}{S_{\mathrm{mp}}} = \frac{53.5}{44} = 1.215$
- $\frac{S_{\mathrm{mp}}}{\mathrm{ENC}} \approx 80$
- signal separates from 0, start at $0.7 \times S_{\mathrm{mp}}$

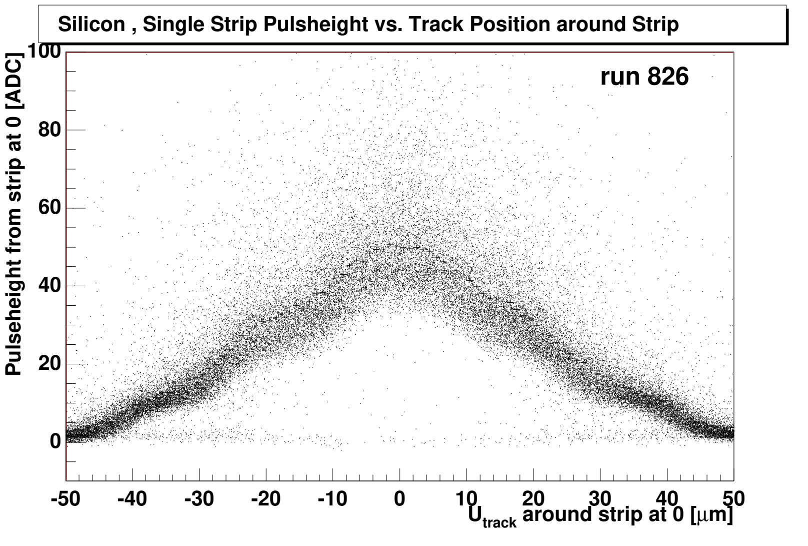
Diamond, Transparent- and Cluster Signal



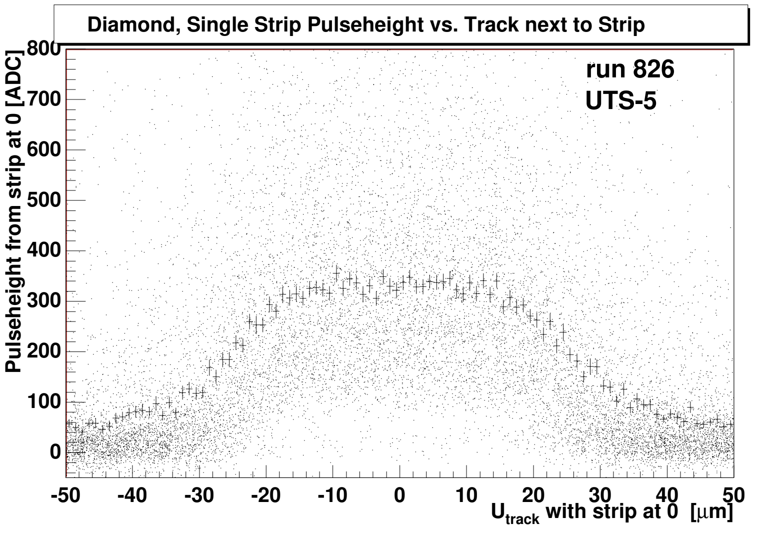
- transparent and clustersignal overlay
- $\frac{S_{\text{mean}}}{S_{\text{mp}}} \approx \frac{476}{277} \approx 1.7$
- $\frac{S_{\text{mp}}}{\text{ENC}} \approx 40$ (based on transparent signal)
- signal separates from 0, start at $0.3 \times S_{\text{mp}}$

Charge Collection on Single Strip

Silicon Plane



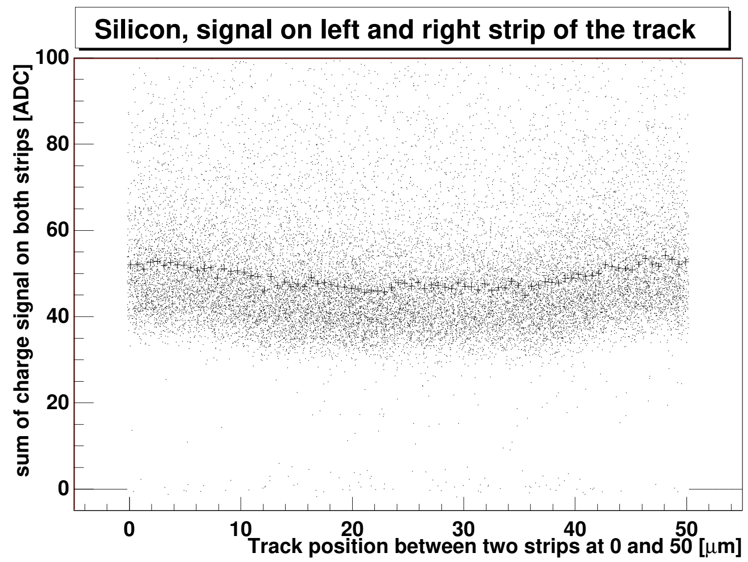
Diamond Plane



Charge Collection on Two Strips

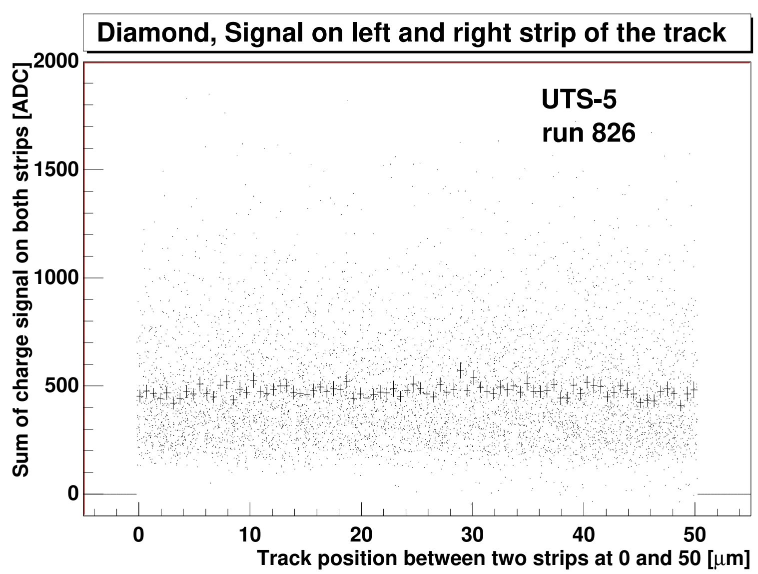
Silicon Plane,

- small loss between strips
- charge collected on adjacent strip



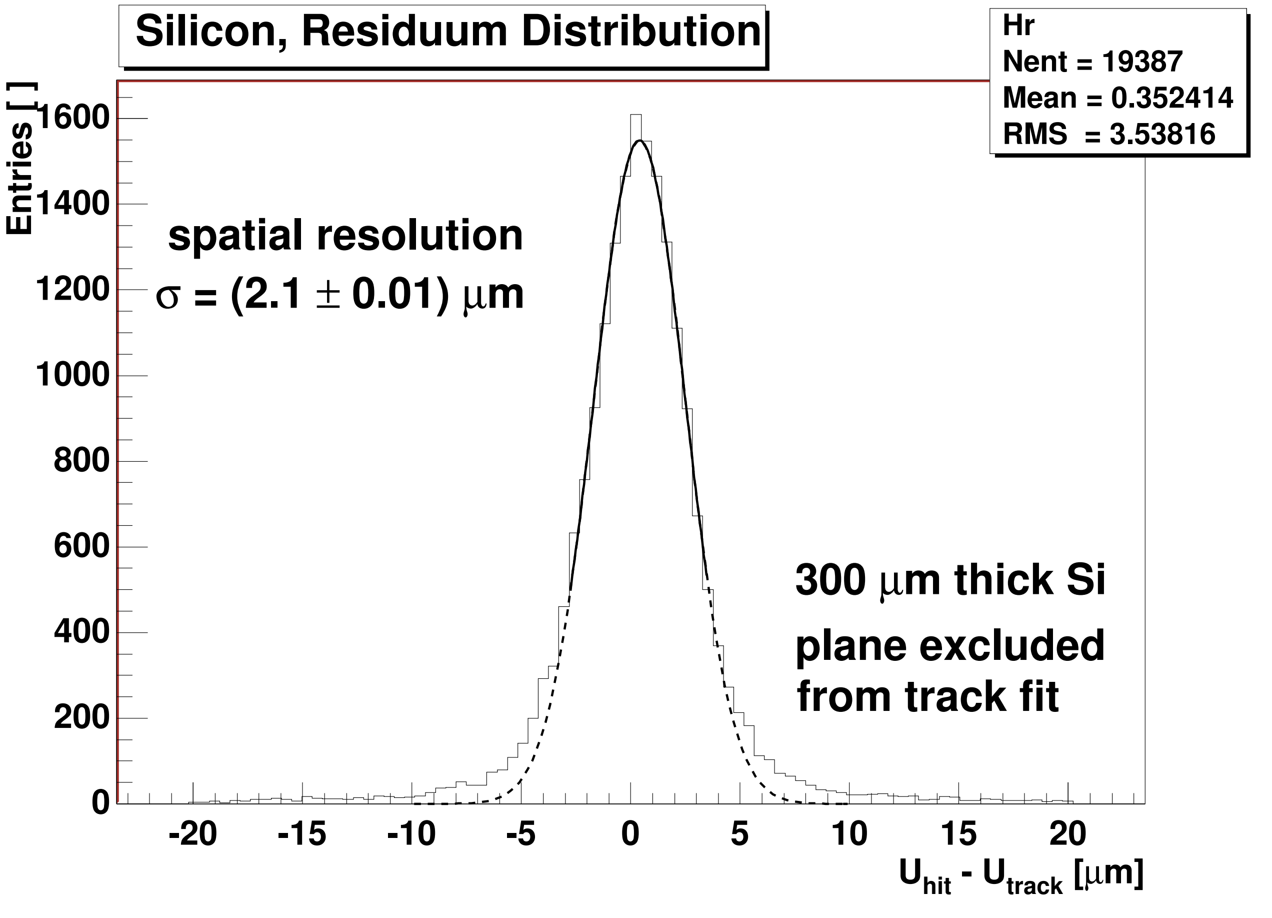
Diamond Plane,

- no loss between strips



Silicon, Spatial Resolution

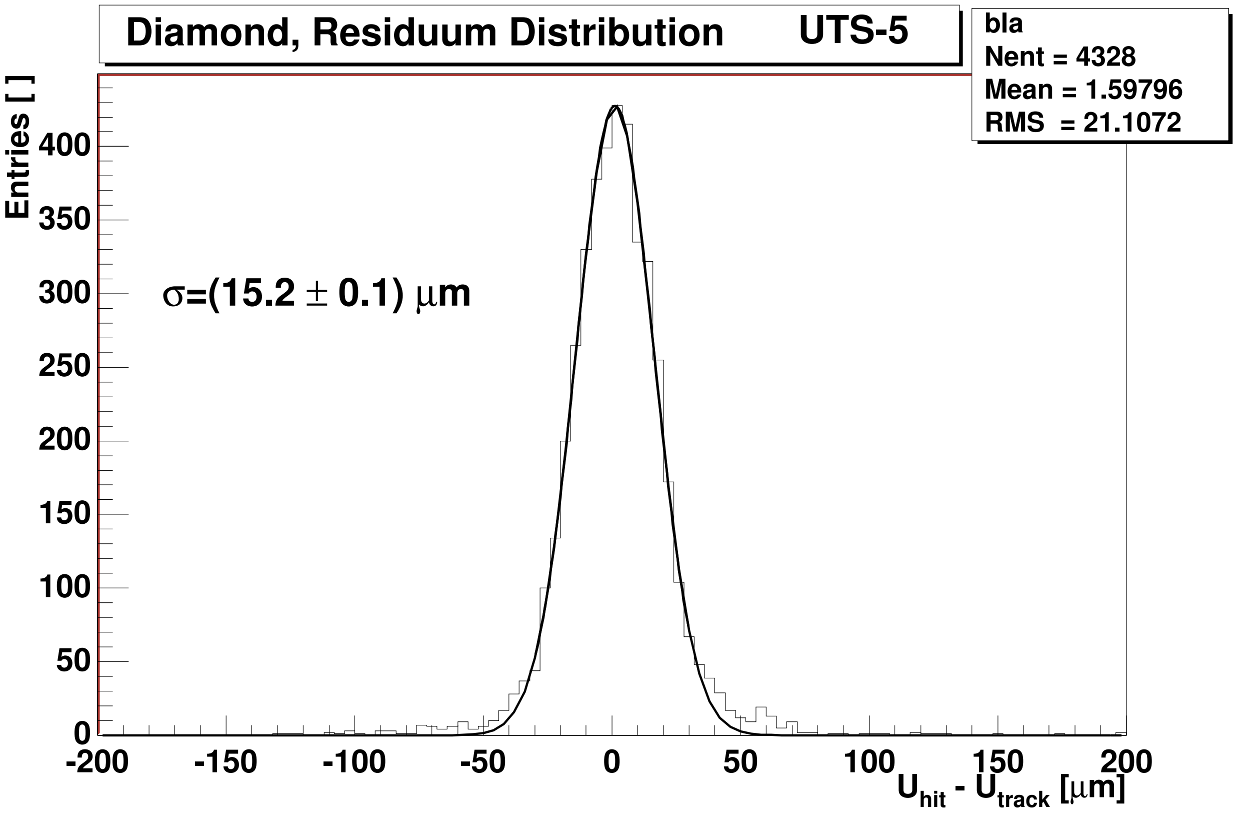
- **Residuum**: Difference between hit and track position
- Hit finding using η-Algorithm



- standard deviation = spatial resolution of 2.1 μm
- **intrinsic spatial resolution** 1.5 μm

Diamond, Spatial Resolution

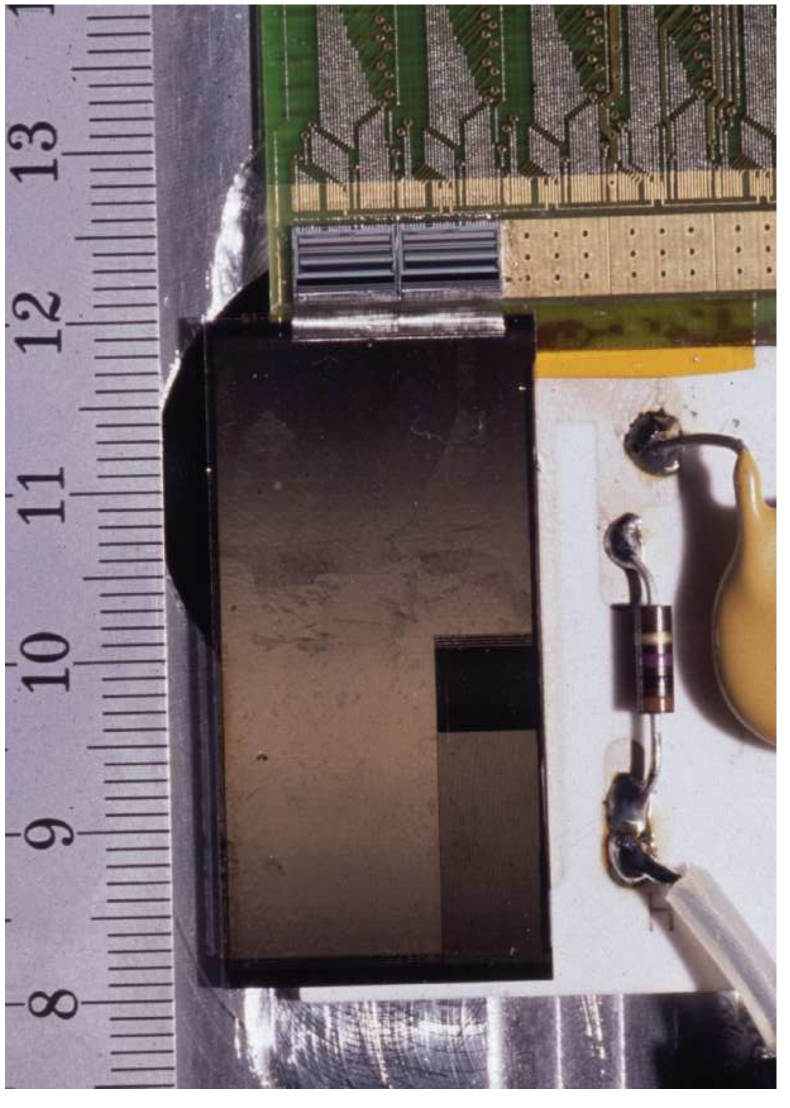
- Residuum with Hit Center of Gravity Method



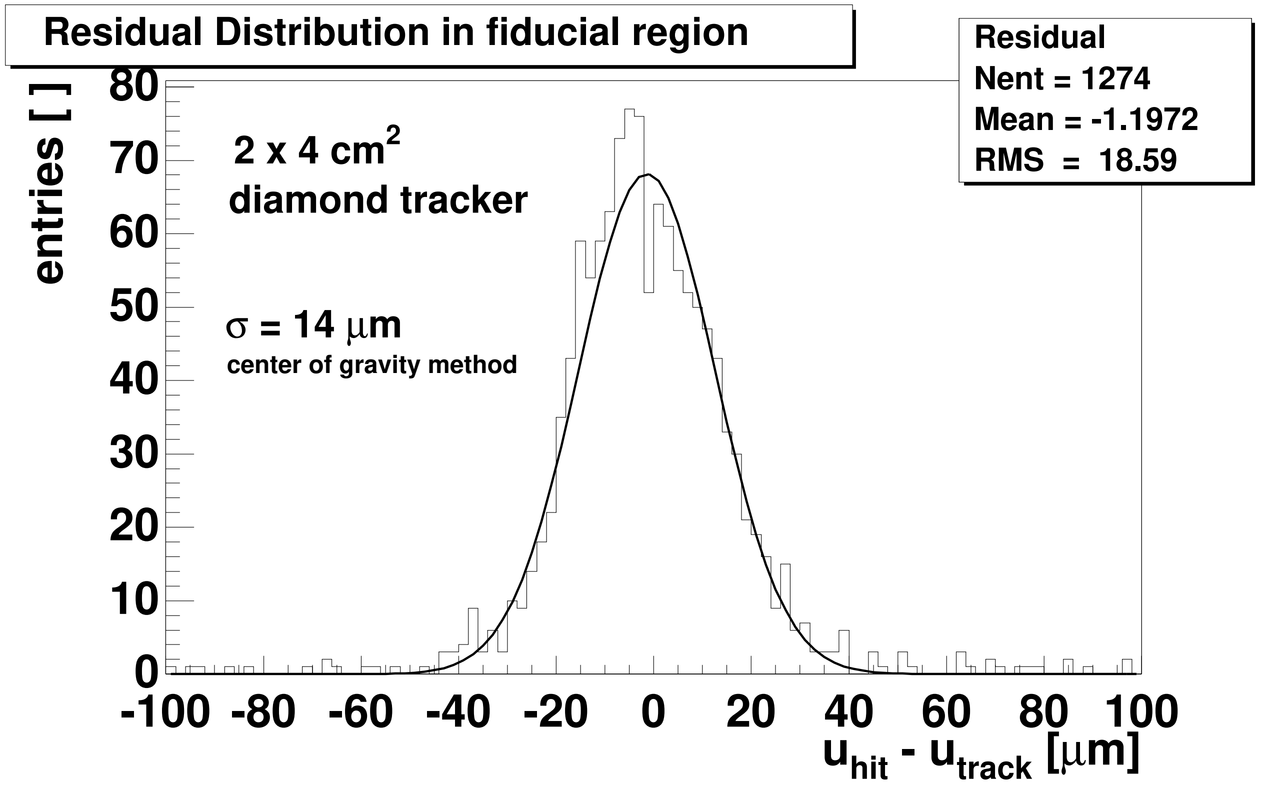
- gaussian distribution, $\sigma = 15\ \mu$m
- .. = digital resolution for 50 μm pitch
- other hit finding algorithm under study

2×4 cm^2 **Diamond**

- Diamond Tracker C44: first 2×4 cm^2 size
- in 100 GeV/c π-Beam in Oct.97, X5A, SPS, CERN
- strips: 50 μm pitch, 25 μm width, length 1.8 cm and 3.8 cm
- $2\times$ VA2 readout, 2 μs shaping, 256 channels



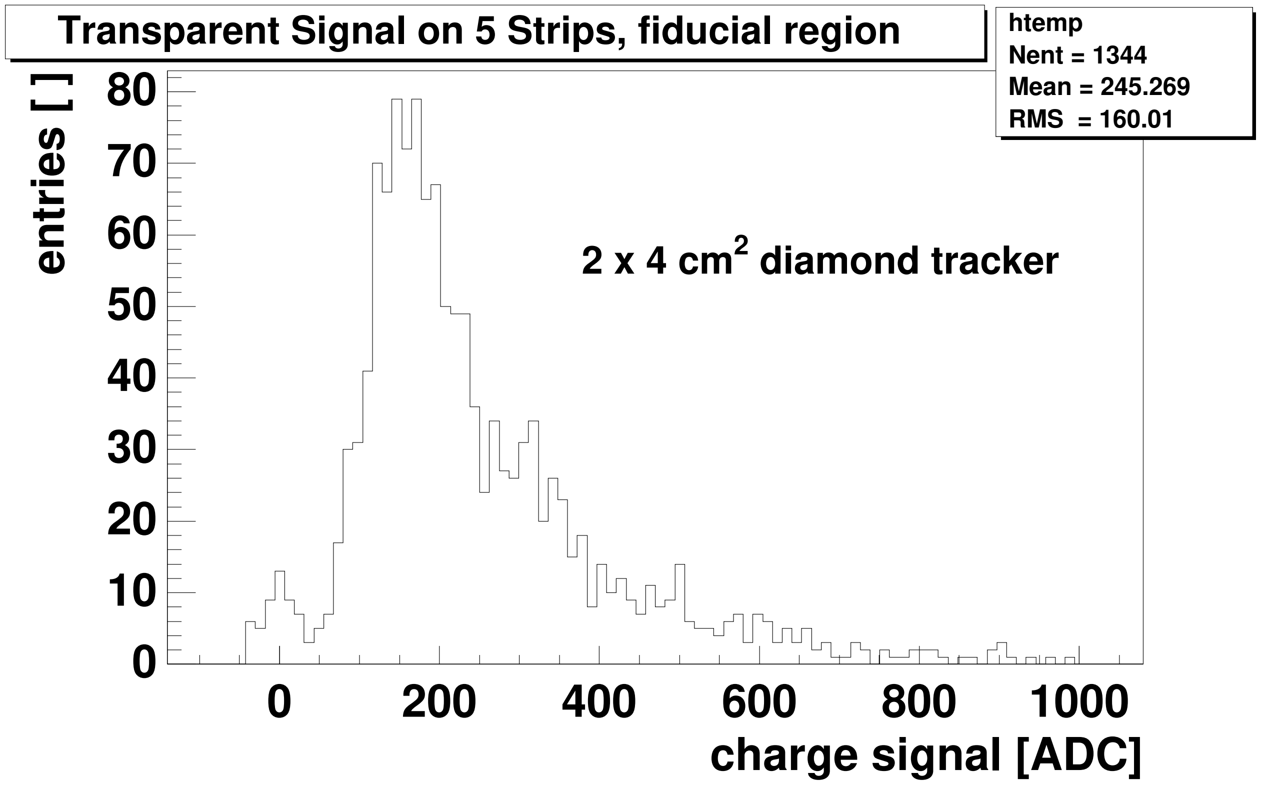
Diamond, Spatial Resolution



- Center-of-Gravity Method: spatial resolution ≈ 14 μm
- .. = digital resolution for 50 μm pitch detector

Diamond, Charge Collection

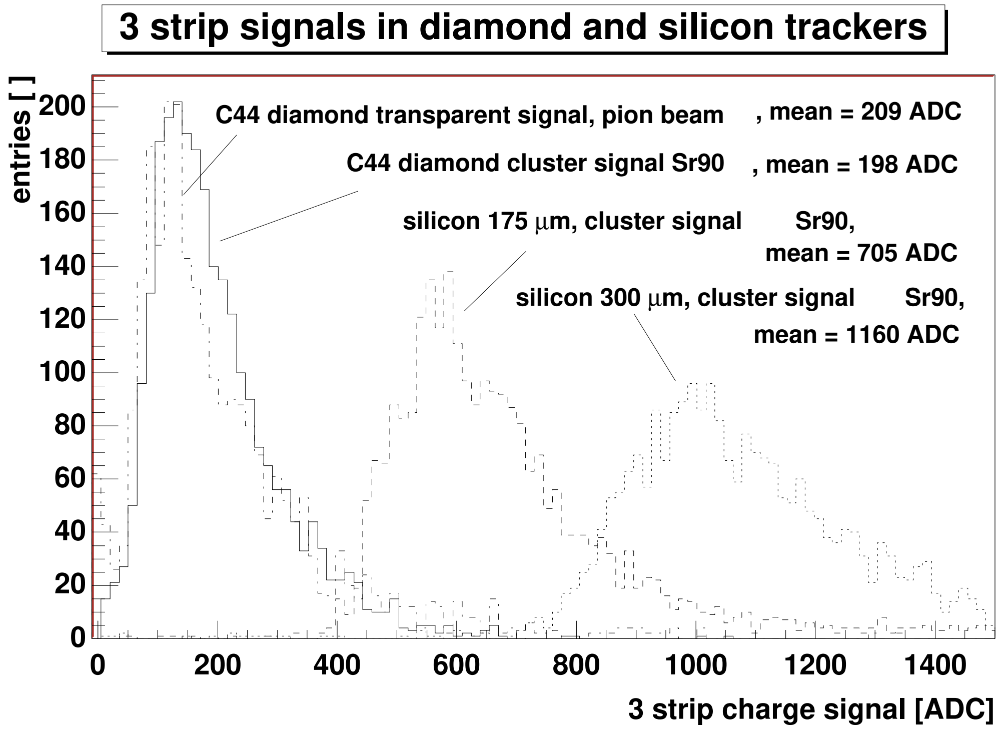
- sum pulseheights on 5 strips, next to track



- **mean** '5-strip' transparent signal $\approx$ 258 ADC
- ratio mean/most probable = 1.5
- Noise $\approx$ 7 ADC ENC $\quad\Longrightarrow S_{\mathrm{mp}}/N \approx$ 23-to-1

Diamond, Charge Collection

- C44 charge signal measured with ^{90}Sr by W. Dulinski, Jan.98 (pumped)
- .. and silicon strip detectors measured using ^{90}Sr for comparison

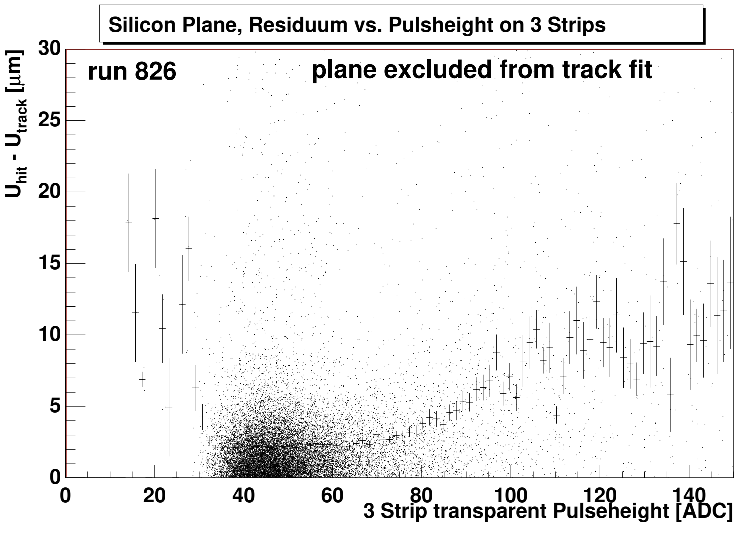


- Mean charge in 300 μm thick silicon 22500 $e \cdot 1.23 = 27700\ e$
- '175er' Silicon 22.9 e/ADC, '300er' silicon 23.8 e/ADC
- C44 **charge collection distance** $4600/36\ \mu\text{m} \approx 128\ \mu\text{m}$

Residuum versus Pulseheight

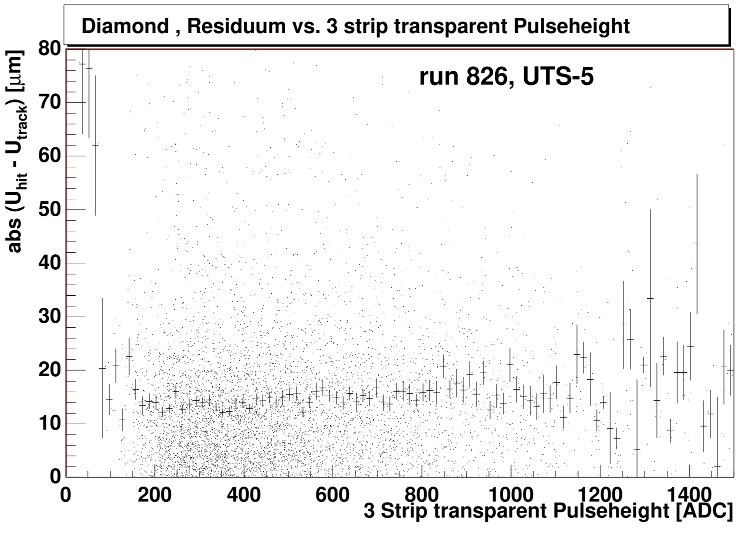
Silicon Plane

- resolution decreases with pulseheight
- δ-electron effect



Diamond Plane

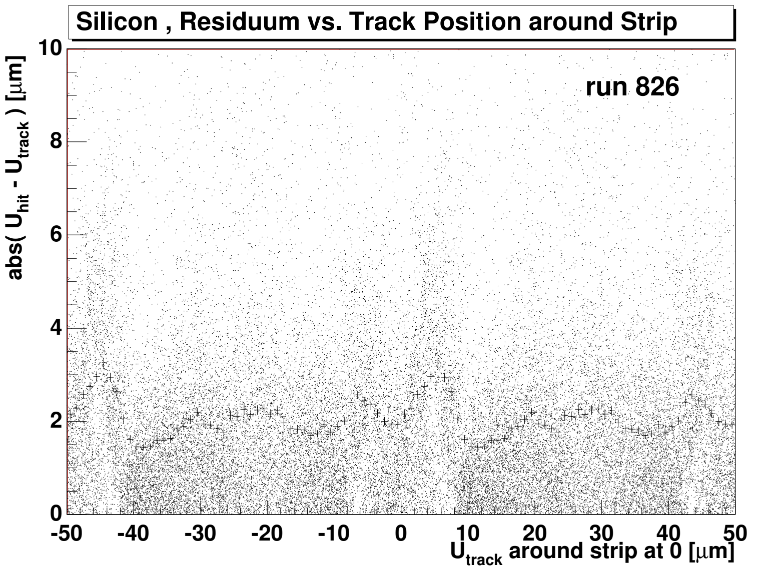
- resolution constant with pulseheight
- CG-Method not sensitive to δ's



Residuum versus Track Position

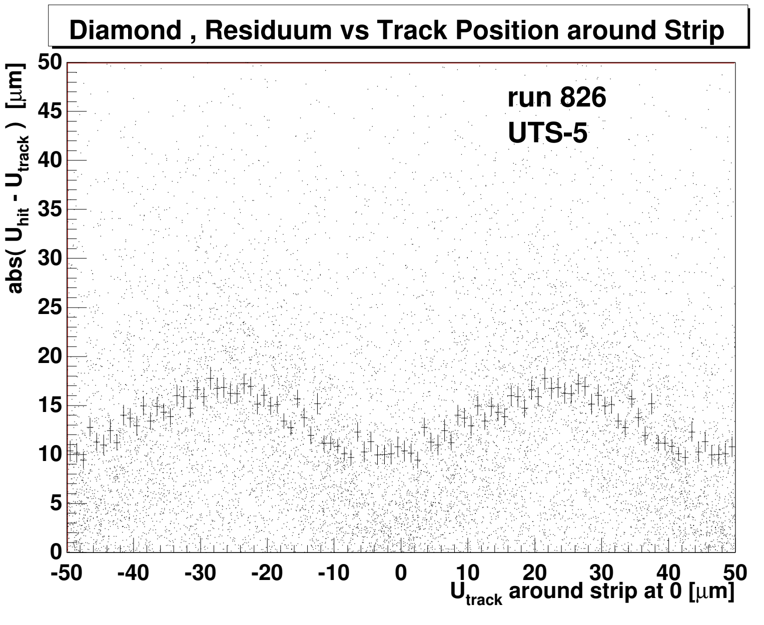
Silicon Plane

- mean varies slightly
- .. due to η-hit finding



Diamond Plane

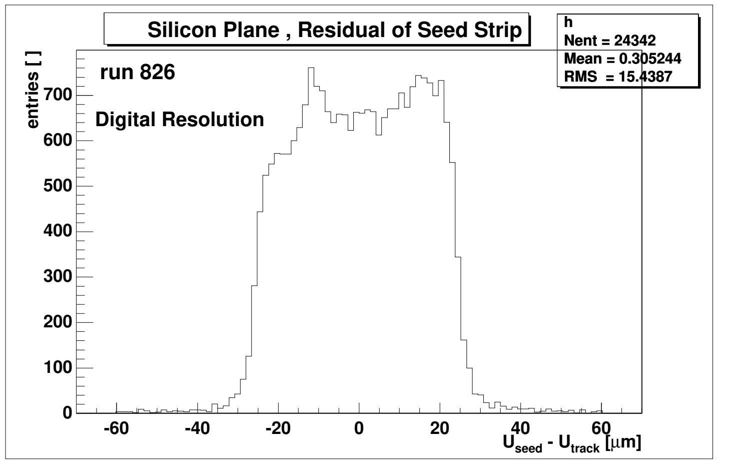
- residuum oscillates
- best at strip position



Digital Resolution (Exercise)

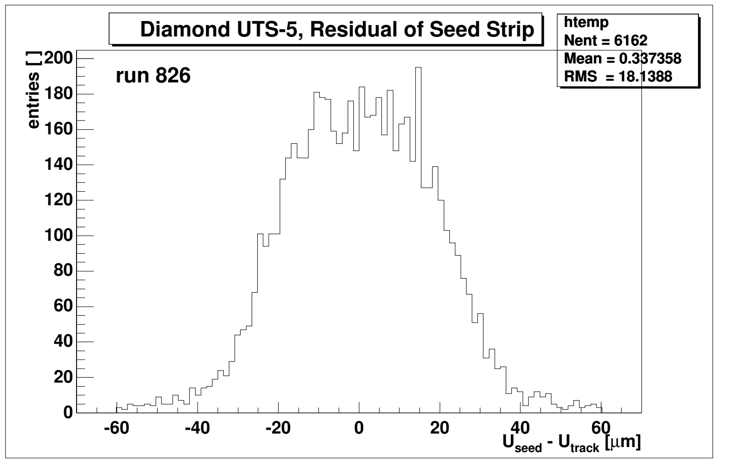
Silicon Plane

- hit taken at seed strip
- $\sigma \approx 50\ \mu\text{m}/\sqrt{12}$



Diamond Plane

- $\sigma \approx 50\ \mu\text{m}/\sqrt{12}$

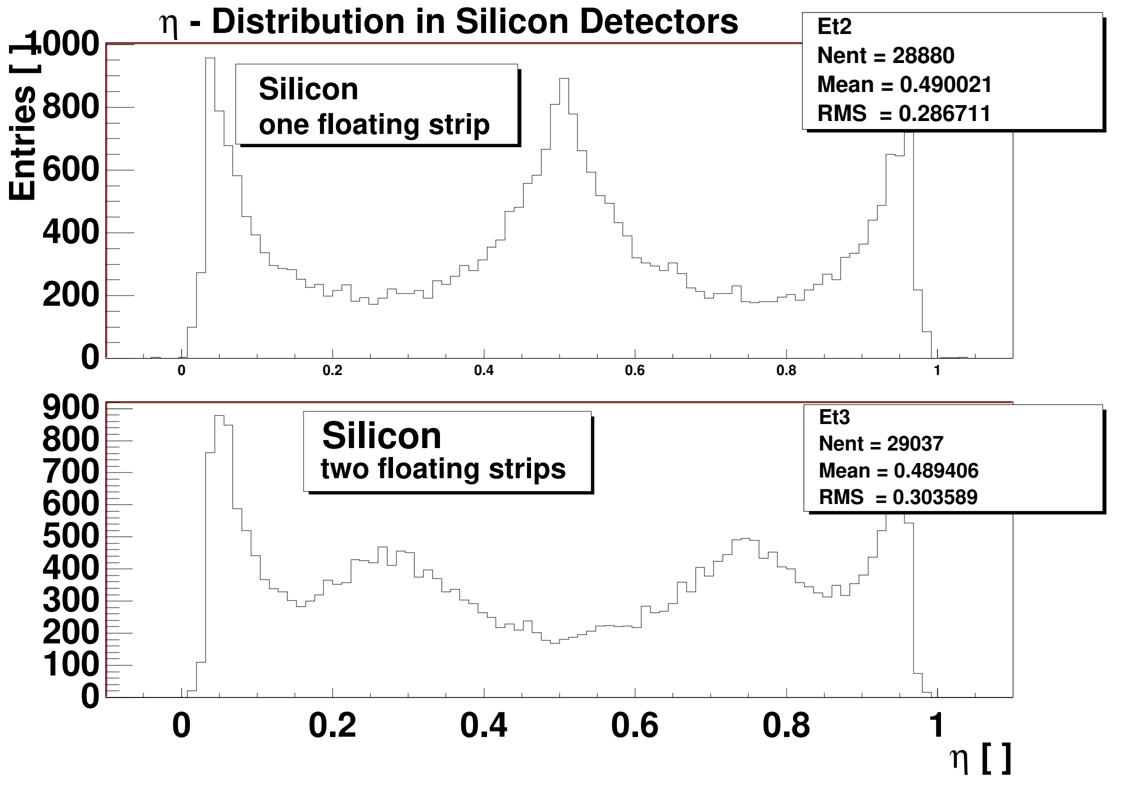


- strip pitch P, strip centered in intervall $[-P/2, +P/2)$
- probability $p = 1$ independent of u_{track}
- residuum distribution has rectangular shape
- spatial resolution σ_{digital} given by

$$\sigma_{\text{digital}} \equiv \sqrt{\langle u^2 \rangle} = \int_{-P/2}^{P/2} p u^2 du = P/\sqrt{12}. \qquad (1)$$

Silicon, η-Distribution

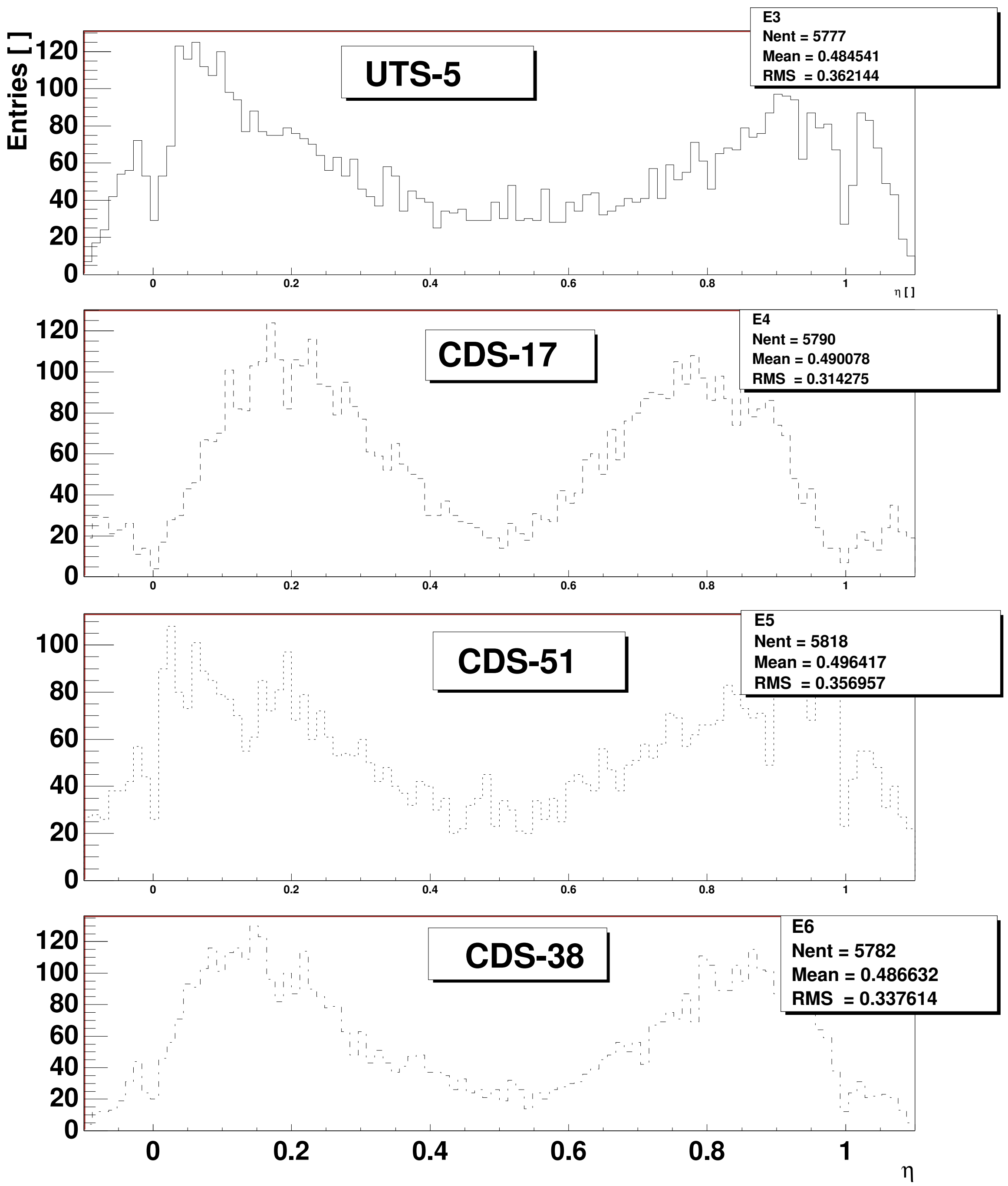
- biggest and 2nd biggest pulse: one **right**, one **left** strip
- Definition $\eta := \frac{\text{Pulseheight}_{\text{rightstrip}}}{\text{Pulseheight}_{\text{rightstrip}}+\text{Pulseheight}_{\text{leftstrip}}}$



- η within intervall [0..1]
- maxima at strips (implants)
- one or two intermediate floating strips

Hit Position Finding

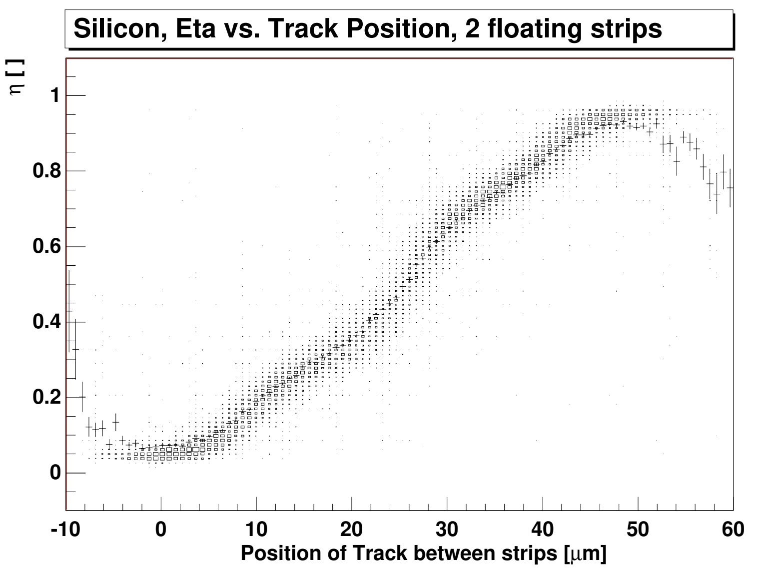
Diamond, η-Distribution



Track with η Correlations

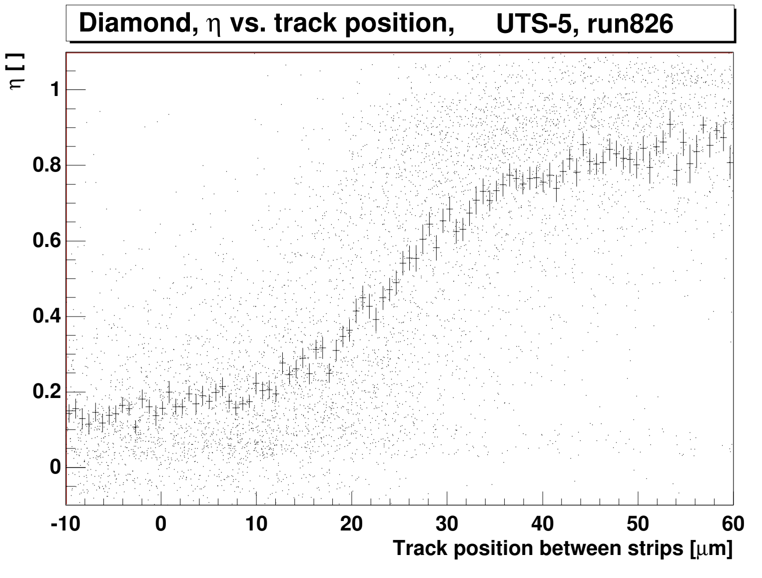
Silicon Plane,

- η correlates with track
- possibilty to improve hit findig



Diamond Plane,

- η correlates with track



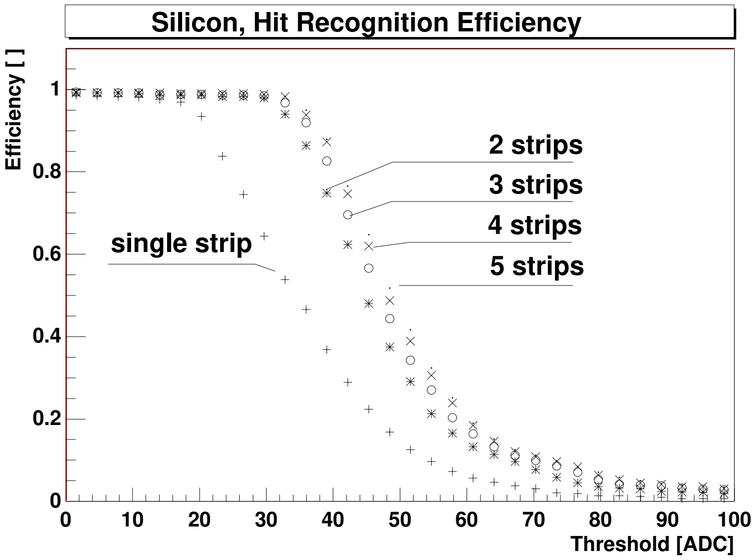
$$x_{\text{hit}} = x_{\text{leftstrip}} + \int_0^{\eta} f(\eta')d\eta' \tag{2}$$

Silicon, Hit Recognition Efficiency

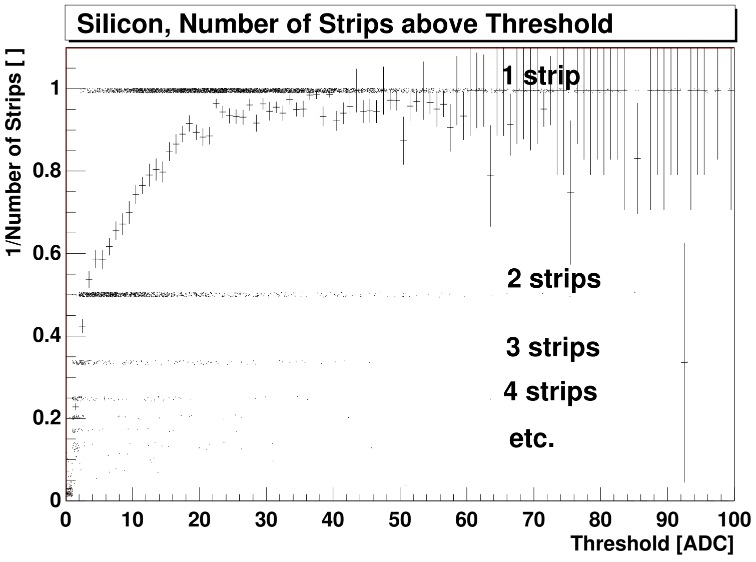
- Definition of **efficiency**

$$\text{efficiency} = \frac{\text{number of hits with PH above threshold}}{\text{all tracks passing detector}}$$

Hit recognition efficiency

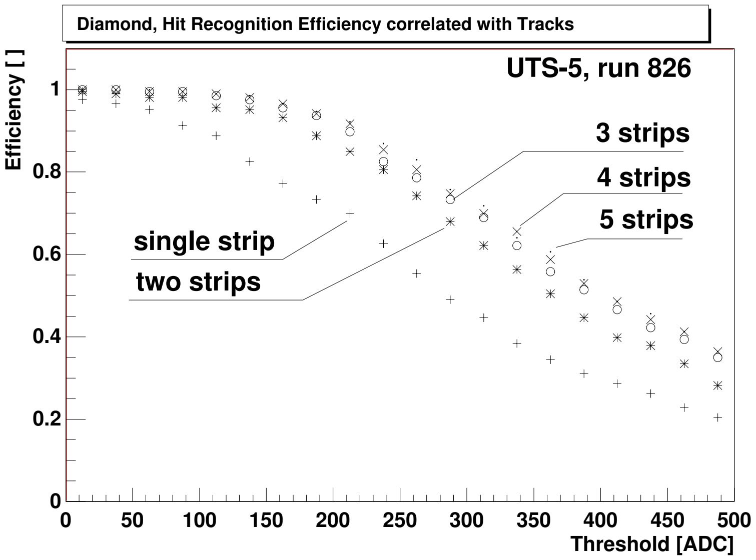


Occupancy

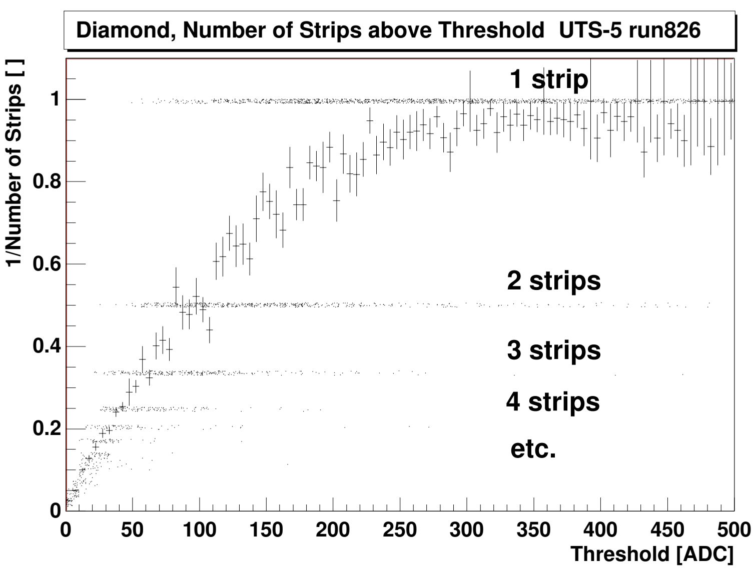


Diamond, Hit Recognition Efficiency

Hit recognition efficiency



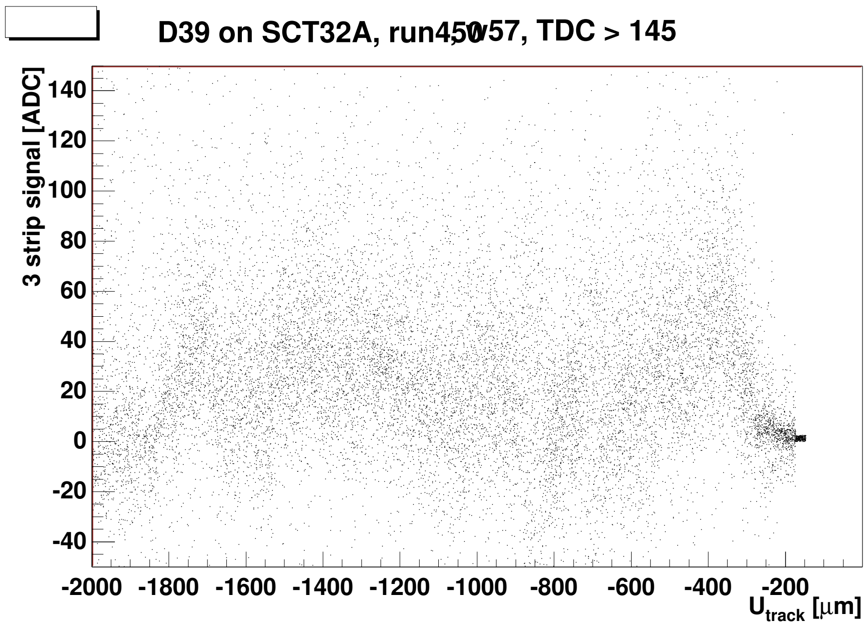
Occupancy



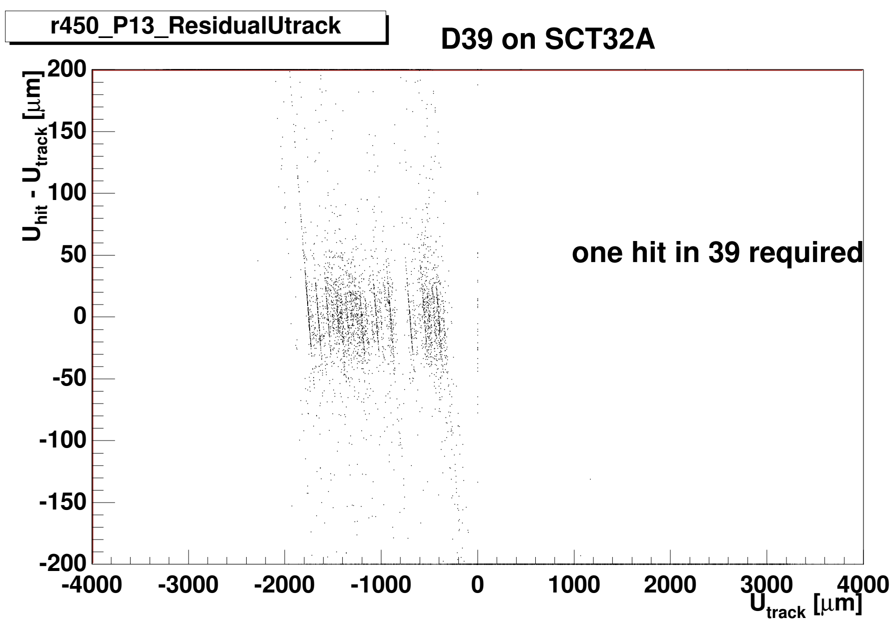
- optimal threshold between 100 ADC and 150 ADC

Fiducialization

Transparent Signal vs. U_{track}

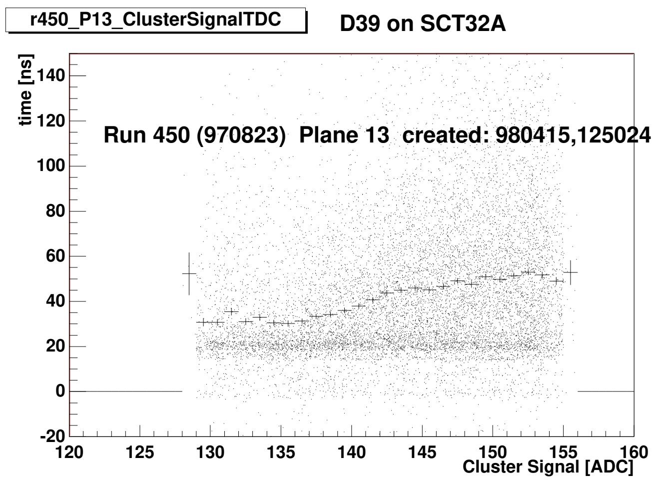


Residuum vs U_{track}

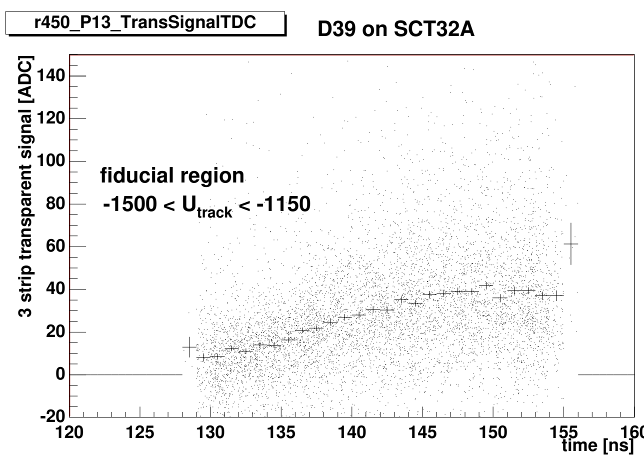


Time Window Selection

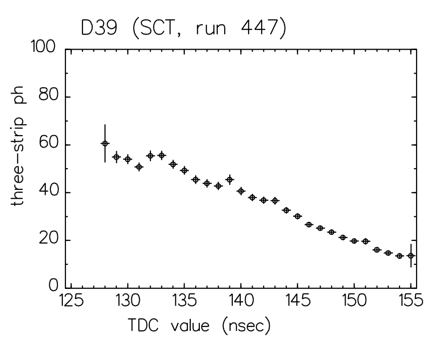
Cluster signal vs TDC



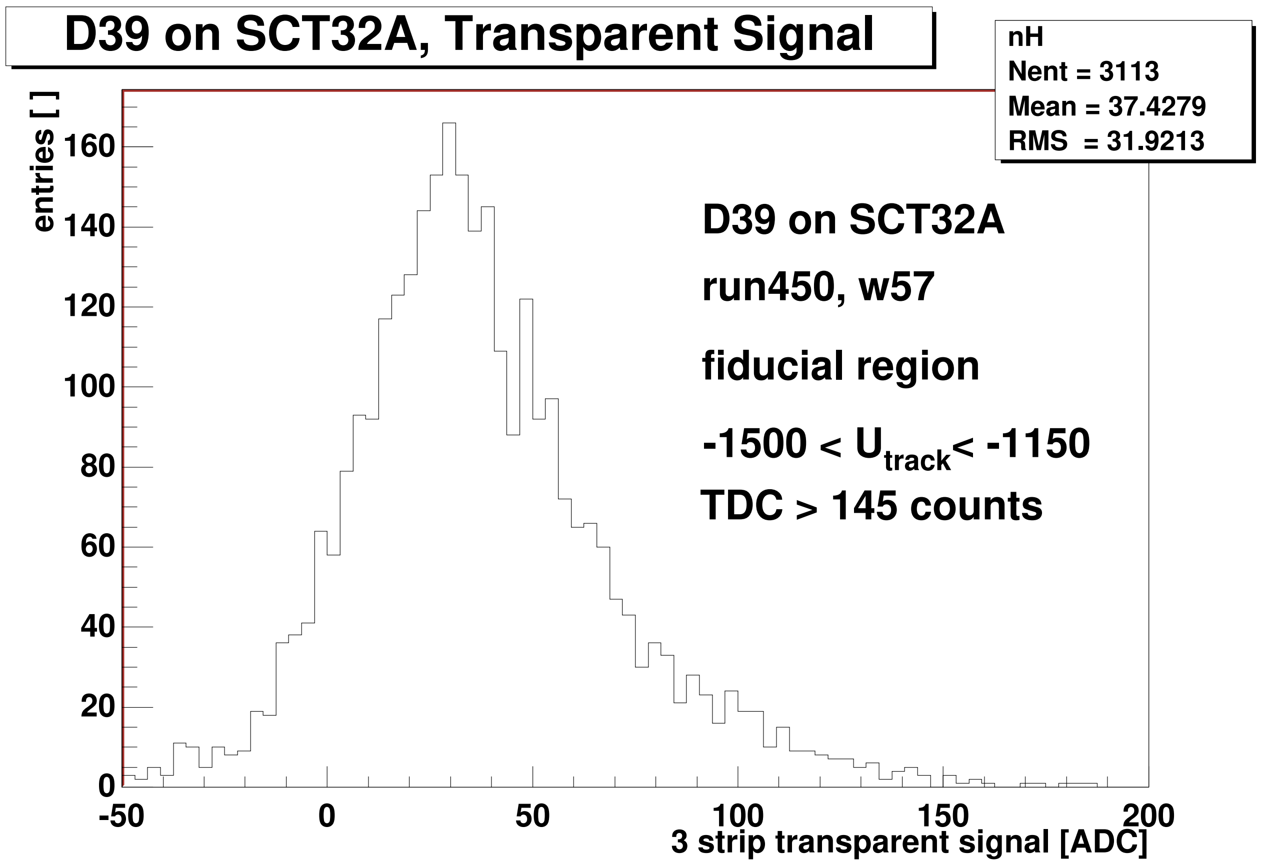
Transparent signal vs TDC,
w57 bucket



Signal vs TDC
(J. Conway, RD42 Notes),
run447, w58 bucket

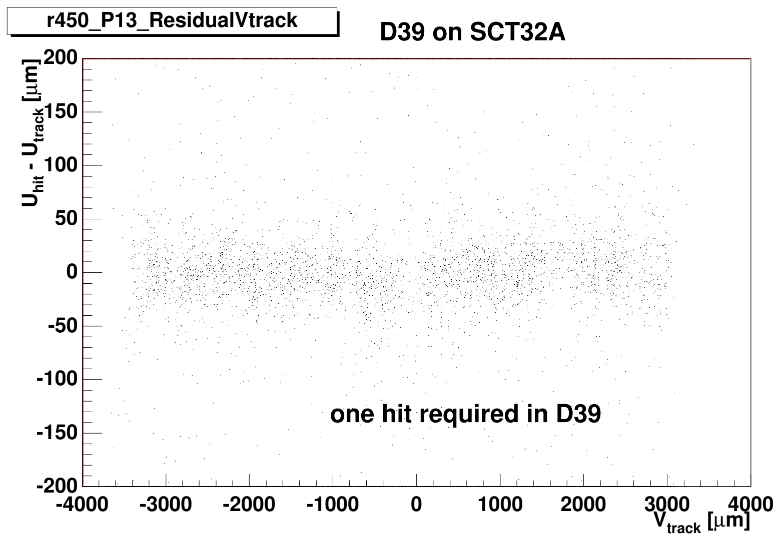


Transparent Signal Distribution

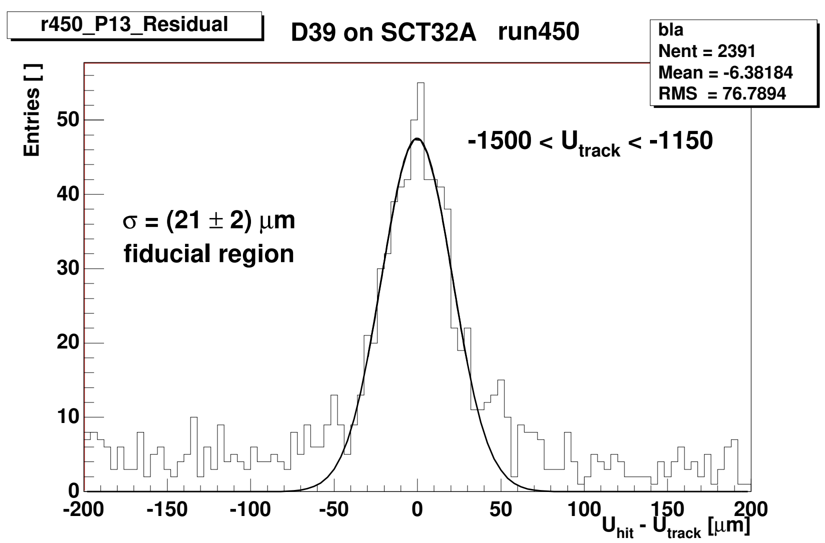


Alignement and Resolution

Residual vs v_{track}



Residual distribution gives spatial resolution of 21 μm



ROOT and OO

- ROOT frame = PAW + objectivity + DataBase Concept

Motivation to use ROOT

- C++ (efficient language, *e.g.* dynamic memory allocation)
- Objects (e.g. events or graphs) put in TTree 's
- Data storage concept: fast, easy, compressed
- ROOT's C++ interpreter
- PAW like: visualization, histogramming, fitting
- good user support, increasing documentation

ROOT Macro, read&dump Raw Data, print Residuum

```
//
//  Example how to use DiamondTracking program
//

{ // start the macro here

  DSession     *tSession = new DSession();                    // define the Session

  tSession->SetRunNumber(826);                                // set a Run Number
  tSession->SetConfigurationPath("./config");                 // set configuration
  tSession->InitSession();                                    // some initialization
  tSession->SetEvents(500);                                   // number of events

  DTracker *tTracker = &tSession->GetTracker();               // declare the Telescope
  DAcq     *tDaq     = &tSession->GetDataAcquisition();       // .. and Acquisition
  DPlane   *tPlane   = &tTracker->GetPlane(9);                // .. and a Plane
  DHit     *aHit;                                             // .. and a Hit
  DTrack   *aTrack;                                           // .. and a Track
  DR3       tTrackPosition;                                   // intersection plane/track
  Float_t   tResiduum;                                        // the residuum
```

ROOT Macro, read&dump Raw Data, print Residuum

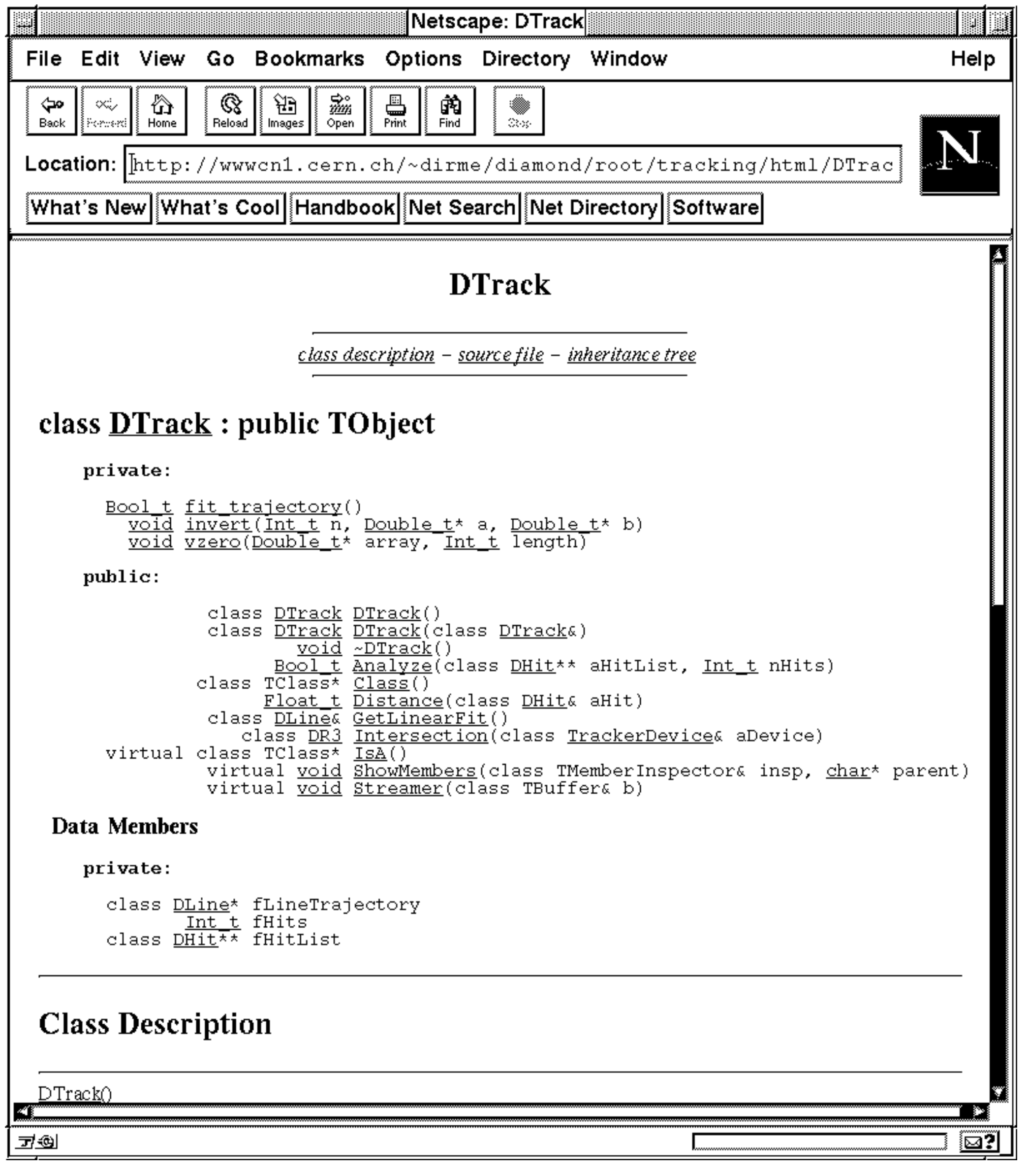
```
  while(tSession->NextRawEvent() == kTRUE) {                    // event loop, get event
    tSession->Update();                                         // update the session
    tDaq->PrintDataToTerm();                                    // print data to terminal
    aTrack         = tTracker->GetPrincipalTrack();             // get the track
    tTrackPosition = aTrack->Intersection(*tPlane);             // intersect plane/track
    aHit           = tPlane->GetPrincipalHit();                 // get the hit in plane
    tResiduum = tTrackPosition(0) - aHit->GetPositionUhit();// calculate the residuum
    tTrackPosition.Print();                                     // print the track intersection
    cout << tResiduum << " micron" << endl;                     // print the residuum
  }     // end the eventloop
}       // end the macro
```

ROOT Macro, read&dump Raw Data, print Residuum

```
//
//  Example how to use DiamondTracking program
//

{ // start the macro here

  DSession     *tSession = new DSession();                  // define the Session

  tSession->SetRunNumber(826);                              // set a Run Number
  tSession->SetConfigurationPath("./config");               // set configuration
  tSession->InitSession();                                  // some initialization
  tSession->SetEvents(500);                                 // number of events

  DTracker *tTracker = &tSession->GetTracker();             // declare the Telescope
  DAcq     *tDaq     = &tSession->GetDataAcquisition();     // .. and Acquisition
  DPlane   *tPlane   = &tTracker->GetPlane(9);              // .. and a Plane
  DHit     *aHit;                                           // .. and a Hit
  DTrack   *aTrack;                                         // .. and a Track
  DR3       tTrackPosition;                                 // intersection plane/track
  Float_t   tResiduum;                                      // the residuum

  while(tSession->NextRawEvent() == kTRUE) {                // event loop, get event

    tSession->Update();                                     // update the session

    tDaq->PrintDataToTerm();                                // print data to terminal

    aTrack         = tTracker->GetPrincipalTrack();         // get the track

    tTrackPosition = aTrack->Intersection(*tPlane);         // intersect plane/track

    aHit           = tPlane->GetPrincipalHit();             // get the hit in plane

    tResiduum = tTrackPosition(0) - aHit->GetPositionUhit();// calculate the residuum

    tTrackPosition.Print();                                 // print the track intersection

    cout << tResiduum << " micron" << endl;                 // print the residuum

  }     // end the eventloop

}       // end the macro
```

Automated HTML-Documentation

- C++ source code and comments convertable to HTML
- … each Class on WWW, e.g. `class DTrack` (which describes a track)



Summary

- **Diamond** Sensors

  Mean signal $\approx 8000\ e$ on *mip*'s, $S_{\mathrm{mp}}/N \approx$ 40-to-1

  Efficiency 100% if threshold below $\approx 2000\ e$

  Large area diamond sensors feasible, $S_{\mathrm{mp}}/N \approx$ 23-to-1

  Digital spatial resolution with Center-of-Gravity Method

  η-algorithm might improve resolution

  Fast DMILL/SCT readout, $S_{\mathrm{mp}}/N \approx$ 3-to-1 in this test

- **Silicon** Telescope

  $\langle S \rangle /N =$ 100-to-1

  intrinsic spatial resolution 1.5 μm

- **Tracking Software**

  ROOT framework

  additional classes linked to it