Hit Finding with PRF and Error Handling in MarlinTPC

Alain Bellerive

Peter Hayman

March 26, 2012

Outline

- Intro: Signal Pulse
  - Electronic Amplitude and time (A,t) for a pad
  - Pad Response Function (PRF) to define a hit
- Background Stand Alone Code
  - Brief code history
  - Code structure: Improvements and limitations
- MarlinTPCP Development
  - Hit Finding with PRF [now]
  - Handling Error from (A,t) $\rightarrow$ (PRF,t) $\rightarrow$ (x,y,z) for a Hit to unbiased track estimators
  - Calibration PRF Module and Simulation [long term]

Charge dispersion

- A high resistivity film bonded to a readout plane with an insulating spacer

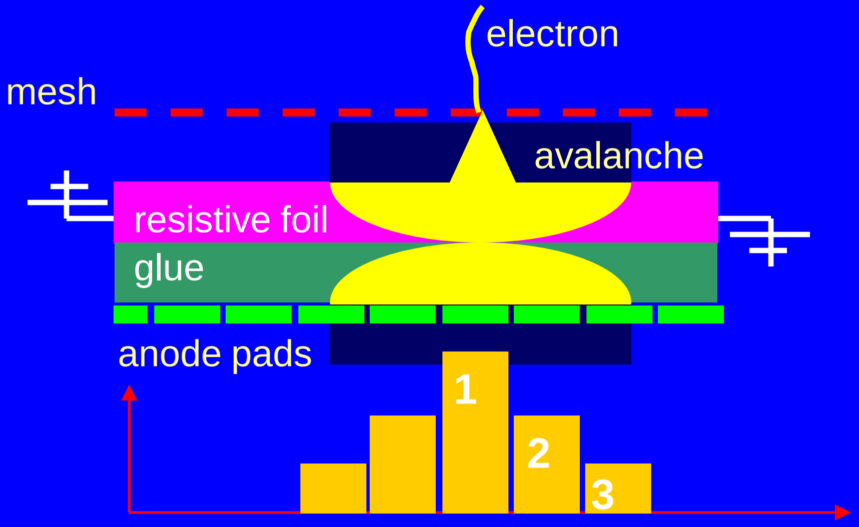
- 2D continuous RC network defined by material properties and geometry.

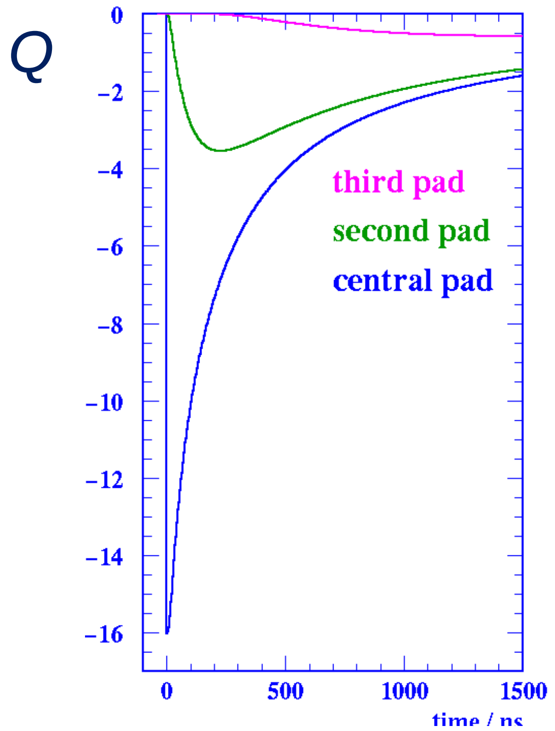
- point charge at r = 0 & t = 0 disperses with time.

$$\frac{\partial \rho}{\partial t} = \frac{1}{RC}\left[\frac{\partial^2 \rho}{\partial r^2} + \frac{1}{r}\frac{\partial \rho}{\partial r}\right]$$

$$\Rightarrow \rho(r,t) = \frac{RC}{2t} e^{\frac{-r^2 RC}{4t}}$$

Micromegas + resistive anode





Amplitude = ∫ Q dt

Time = mid point (c.p.)

Pulse shape origin

Transverse diffusion

$$T(x) = \frac{1}{\sigma_x\sqrt{2\pi}}\exp\left(\frac{-x^2}{2\sigma_x^2}\right)$$

Longitudinal diffusion

$$L(t) = \frac{1}{\sigma_t\sqrt{2\pi}}\exp\left(\frac{-t^2}{2\sigma_t^2}\right)$$

Induction gap

$$R(t) = \frac{t}{T_{rise}} \qquad 0 < t < T_{rise}$$

$$= 1 \qquad t > T_{rise}$$

$$= 0 \qquad t < 0$$

Electronic Response

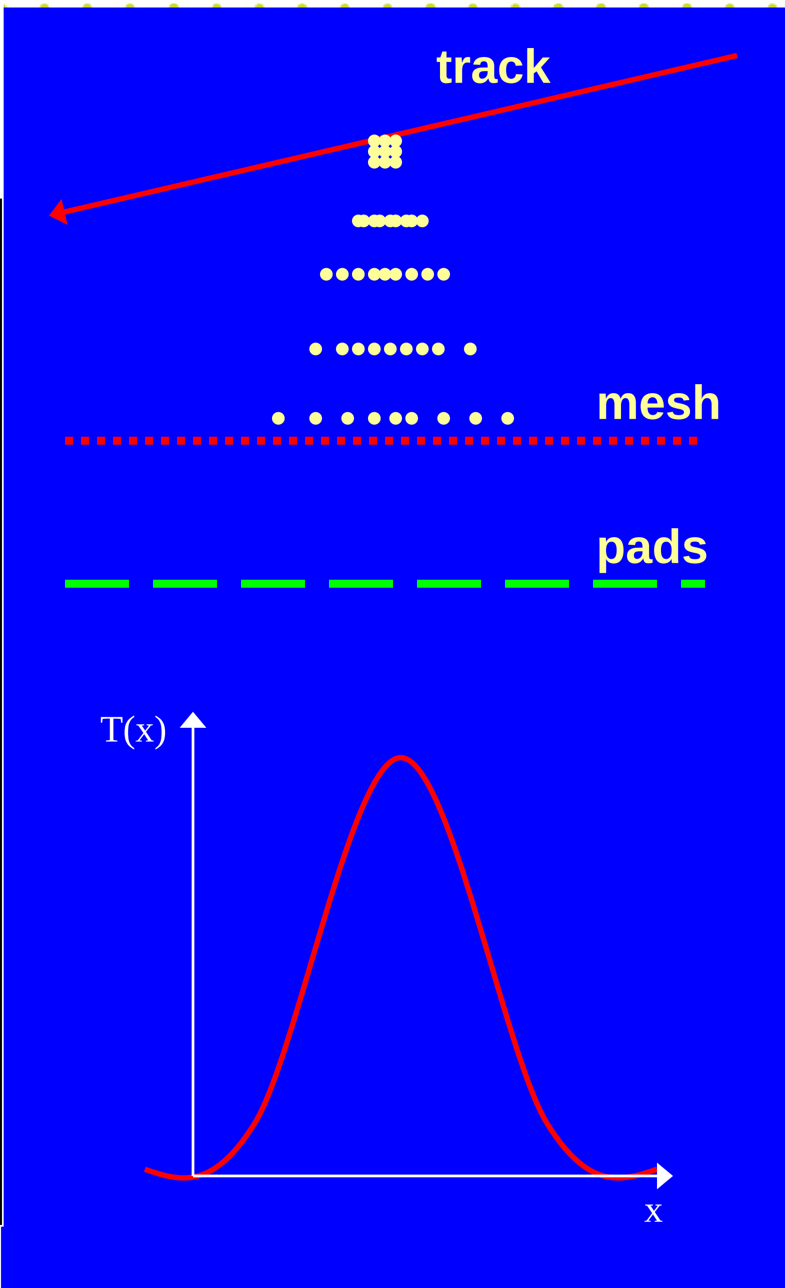
$$A(t) = \exp\left(-\frac{t}{t_f}\right)\left(1-\exp\left(\frac{t}{t_r}\right)\right) \qquad t > 0$$

$$= 0 \qquad t < 0$$

Resistive foil + glue

$$\rho(x,y,t) = \left(\frac{1}{\sigma_t\sqrt{\pi t h}}\right)^2 \exp\left(\frac{-(x^2+y^2)}{4th}\right)$$

$$h = 1/RC$$



Pulse shape origin

Transverse diffusion

$$T(x) = \frac{1}{\sigma_x\sqrt{2\pi}}\exp(\frac{-x^2}{2\sigma_x^2})$$

Longitudinal diffusion

$$L(t) = \frac{1}{\sigma_t\sqrt{2\pi}}\exp(\frac{-t^2}{2\sigma_t^2})$$

Induction gap

$$\begin{aligned} R(t) &= \frac{t}{T_{rise}} && 0 < t < T_{rise} \\ &= 1 && t > T_{rise} \\ &= 0 && t < 0 \end{aligned}$$

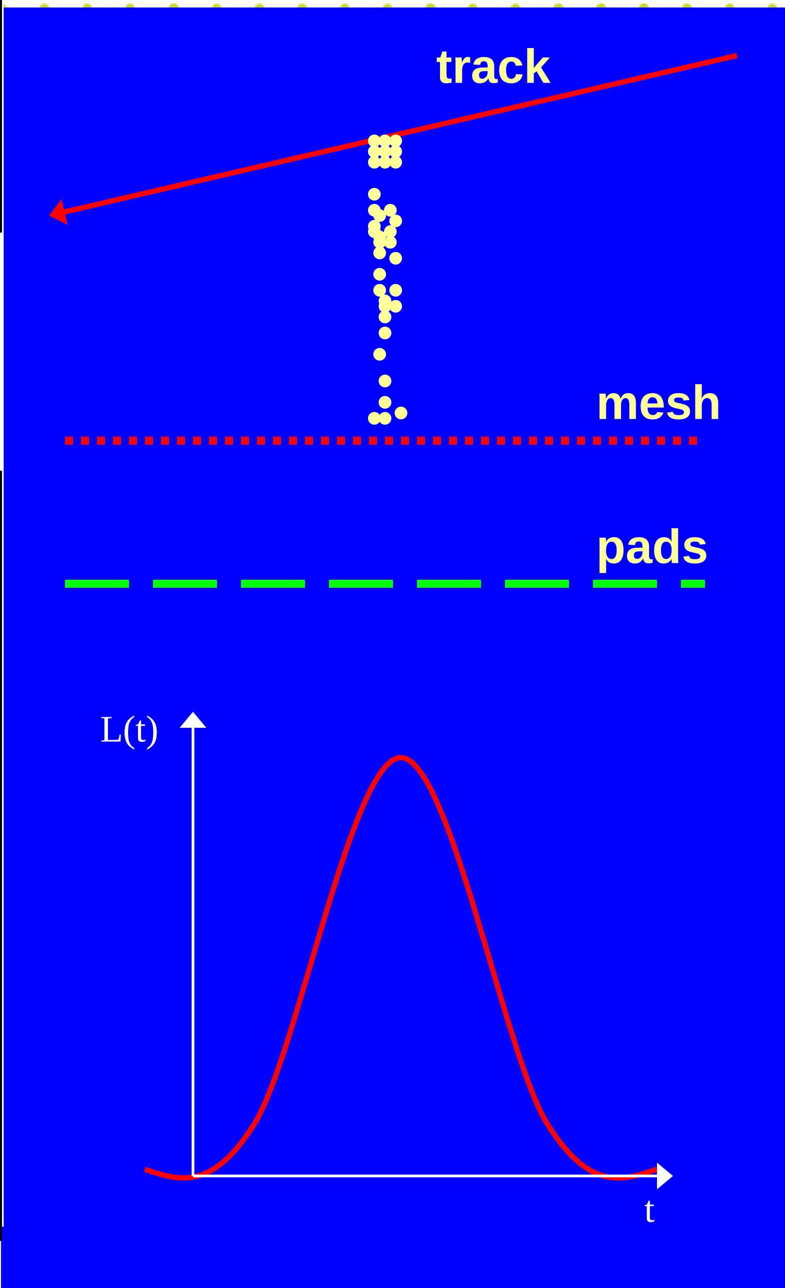
Electronic Response

$$\begin{aligned} A(t) &= \exp\left(-\frac{t}{t_f}\right)\left(1-\exp\left(\frac{t}{t_r}\right)\right) && t > 0 \\ &= 0 && t < 0 \end{aligned}$$

Resistive foil + glue

$$\rho(x,y,t) = \left(\frac{1}{\sigma_t\sqrt{\pi t h}}\right)^2 \exp\left(\frac{-(x^2+y^2)}{4th}\right)$$

$$h = 1/RC$$



Pulse shape origin

Transverse diffusion

$$T(x) = \frac{1}{\sigma_x \sqrt{2\pi}} \exp\left(\frac{-x^2}{2\sigma_x^2}\right)$$

Longitudinal diffusion

$$L(t) = \frac{1}{\sigma_t \sqrt{2\pi}} \exp\left(\frac{-t^2}{2\sigma_t^2}\right)$$

Induction gap

$$R(t) = \frac{t}{T_{rise}} \qquad 0 < t < T_{rise}$$
$$= 1 \qquad t > T_{rise}$$
$$= 0 \qquad t < 0$$

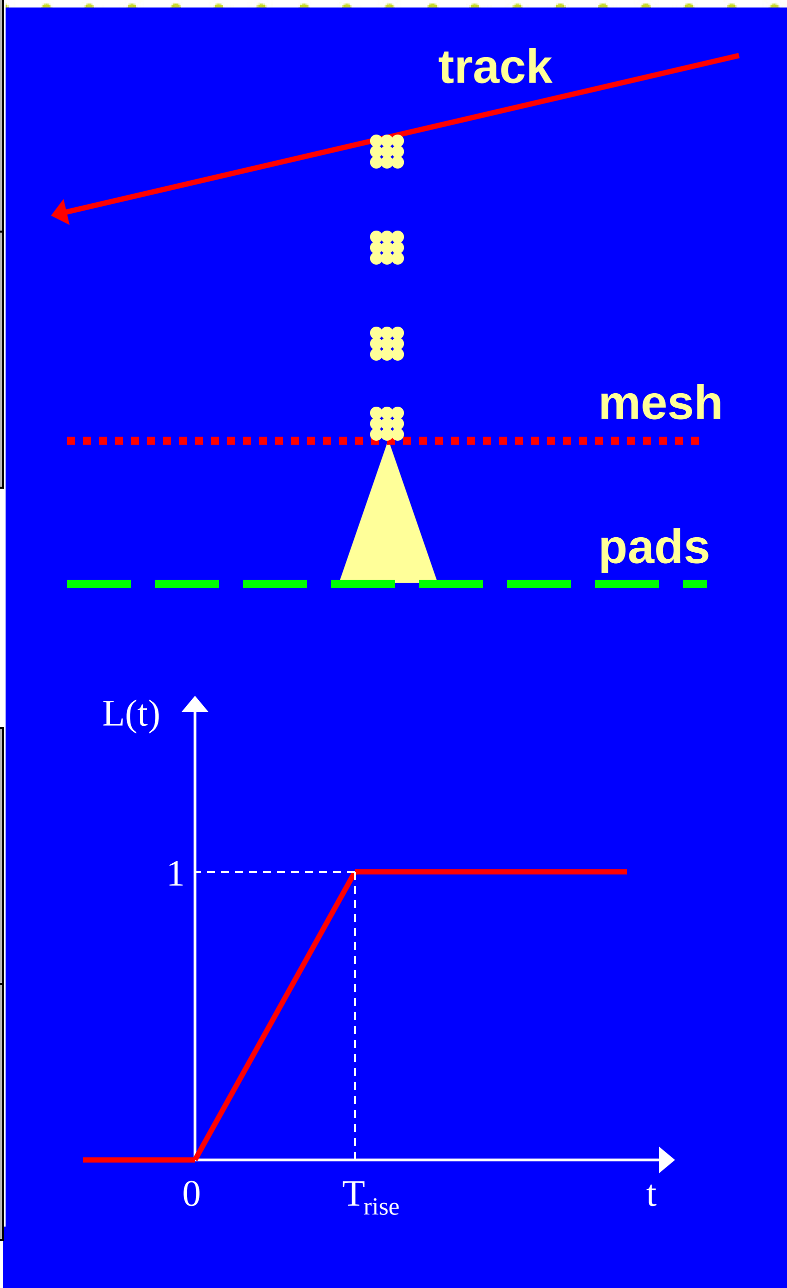
Electronic Response

$$A(t) = \exp\left(-\frac{t}{t_f}\right)\left(1 - \exp\left(\frac{t}{t_r}\right)\right) \qquad t > 0$$
$$= 0 \qquad t < 0$$

Resistive foil + glue

$$\rho(x, y, t) = \left(\frac{1}{\sigma_t \sqrt{\pi t h}}\right)^2 \exp\left(\frac{-(x^2 + y^2)}{4th}\right)$$

$$h = 1/RC$$



Pulse shape origin

Transverse diffusion

$$T(x) = \frac{1}{\sigma_x\sqrt{2\pi}}\exp(\frac{-x^2}{2\sigma_x^2})$$

Longitudinal diffusion

$$L(t) = \frac{1}{\sigma_t\sqrt{2\pi}}\exp(\frac{-t^2}{2\sigma_t^2})$$

Induction gap

$$\begin{aligned} R(t) &= \frac{t}{T_{rise}} && 0 < t < T_{rise} \\ &= 1 && t > T_{rise} \\ &= 0 && t < 0 \end{aligned}$$

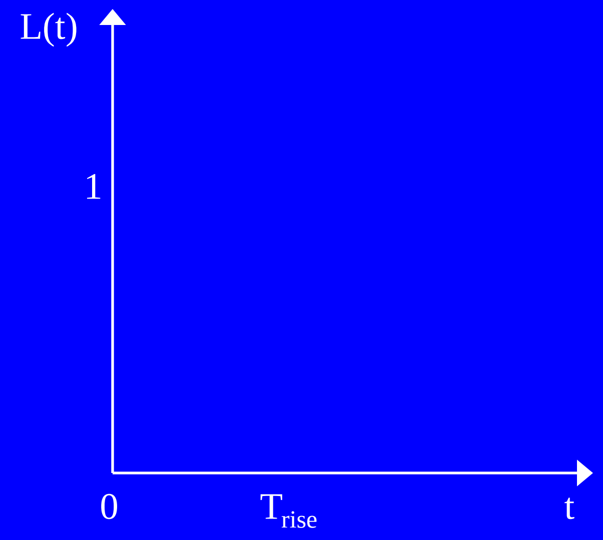
Electronic Response

$$\begin{aligned} A(t) &= \exp\left(-\frac{t}{t_f}\right)\left(1-\exp\left(\frac{t}{t_r}\right)\right) && t > 0 \\ &= 0 && t < 0 \end{aligned}$$

Resistive foil + glue

$$\rho(x,y,t) = \left(\frac{1}{\sigma_t\sqrt{\pi t h}}\right)^2 \exp\left(\frac{-(x^2+y^2)}{4th}\right)$$

$$h = 1/RC$$



Pulse shape origin

Transverse diffusion	$T(x)=\frac{1}{\sigma_x\sqrt{2\pi}}\exp(\frac{-x^2}{2\sigma_x^2})$
Longitudinal diffusion	$L(t)=\frac{1}{\sigma_t\sqrt{2\pi}}\exp(\frac{-t^2}{2\sigma_t^2})$
Induction gap	$R(t)=\frac{t}{T_{rise}} \quad 0<t<T_{rise}$; $=1 \quad t>T_{rise}$; $=0 \quad t<0$
Electronic Response	$A(t)=\exp\left(-\frac{t}{t_f}\right)\left(1-\exp\left(\frac{t}{t_r}\right)\right) \quad t>0$; $=0 \quad t<0$

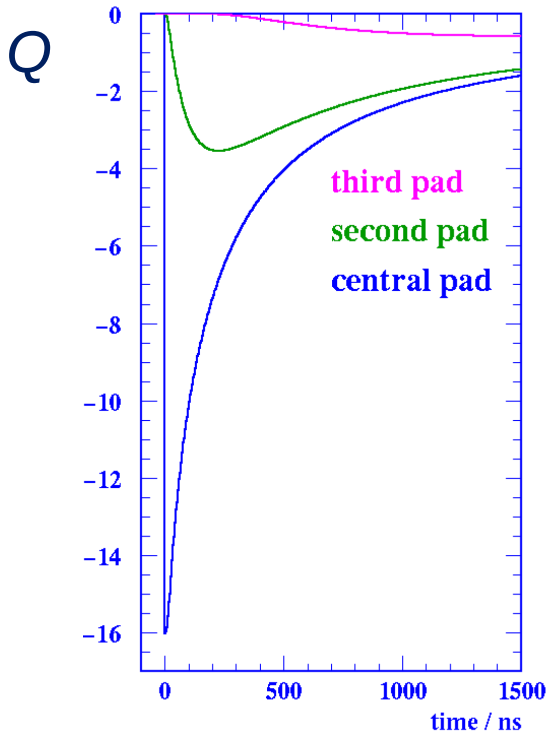
Resistive foil + glue $\rho(x,y,t)=\left(\frac{1}{\sigma_t\sqrt{\pi th}}\right)^2\exp\left(\frac{-(x^2+y^2)}{4th}\right)$

$h=1/RC$

Qi = ∫ ρ$_i$(r) dr

A$_i$ = Amplitude = ∫ Q$_i$(t) dt

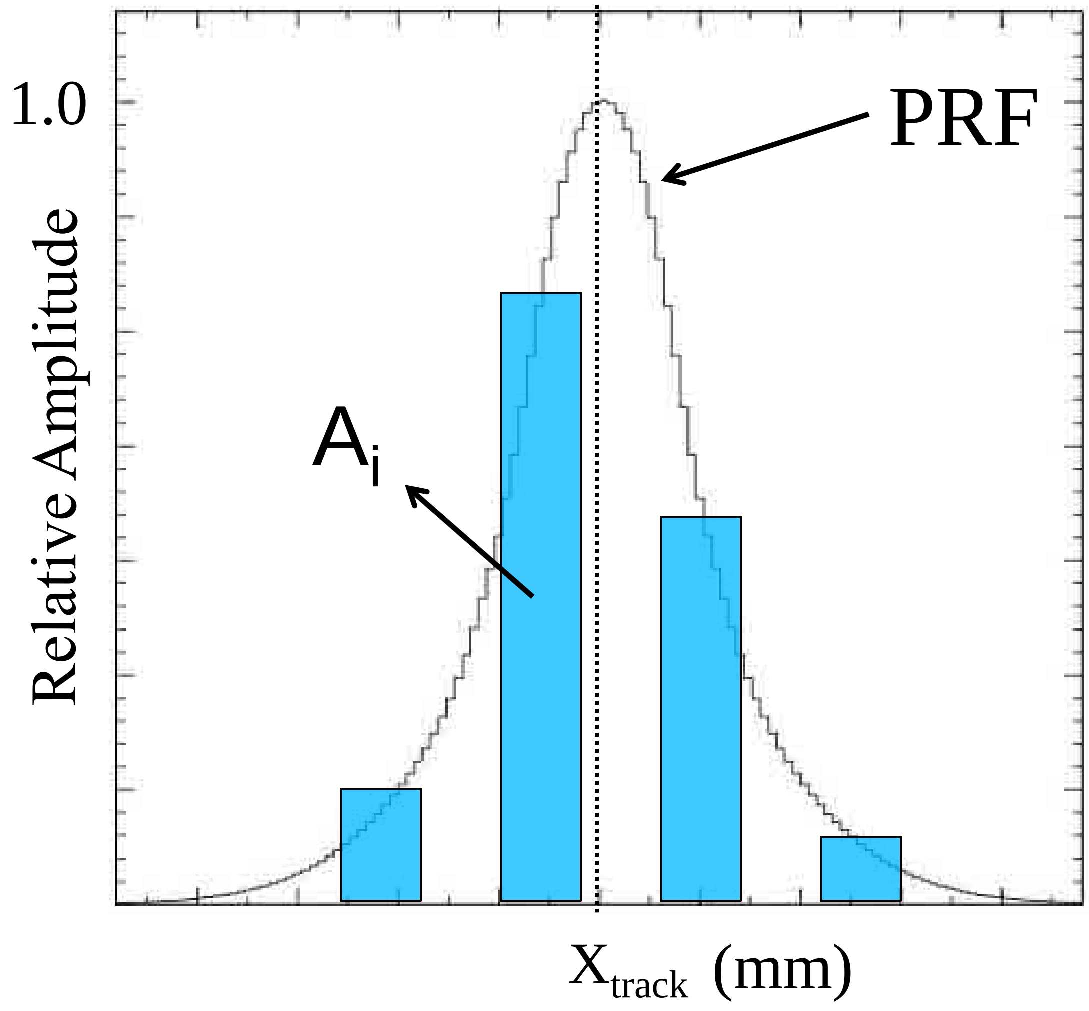
Time = mid point (c.p.)



M.S. Dixit & A. Rankin, NIM A566, 281 (2006)

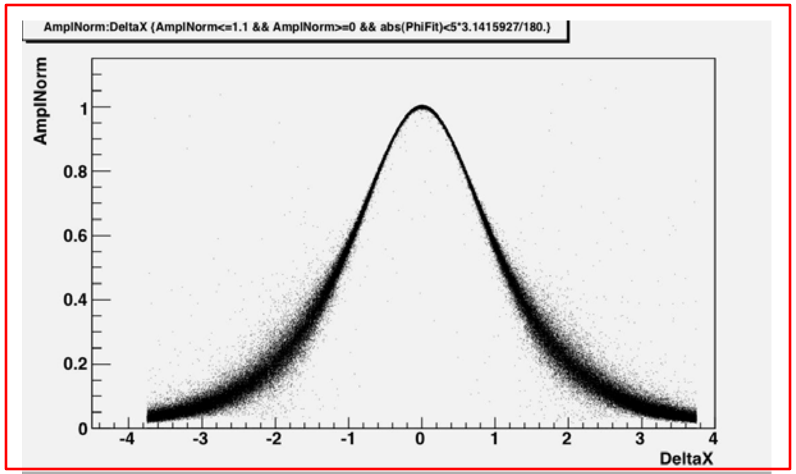
C++ code developed during summer 2010

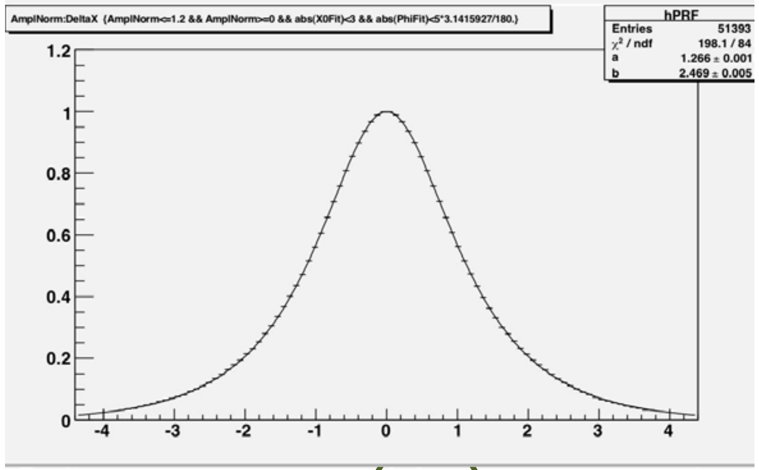
Pad Response Function (PRF)



For a given X_{track} (known position) the PRF is determined for each row

Pad Response Function





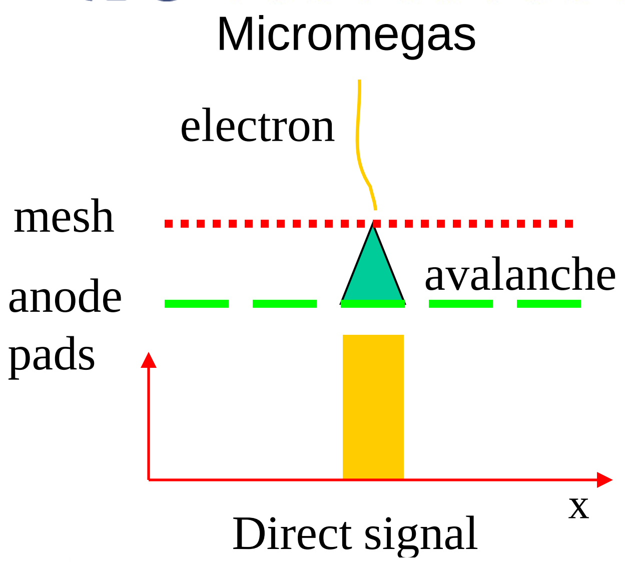
(mm)

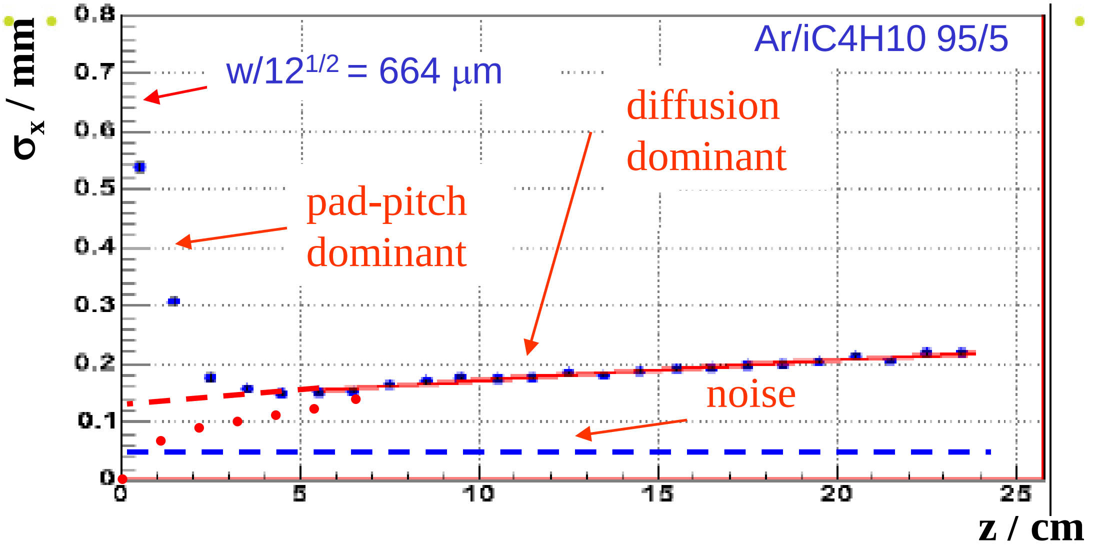
$$\mathrm{PRF}(x,\Gamma,\Delta,a,b)=\frac{1+a_2x^2+a_4x^4}{1+b_2x^2+b_4x^4}$$

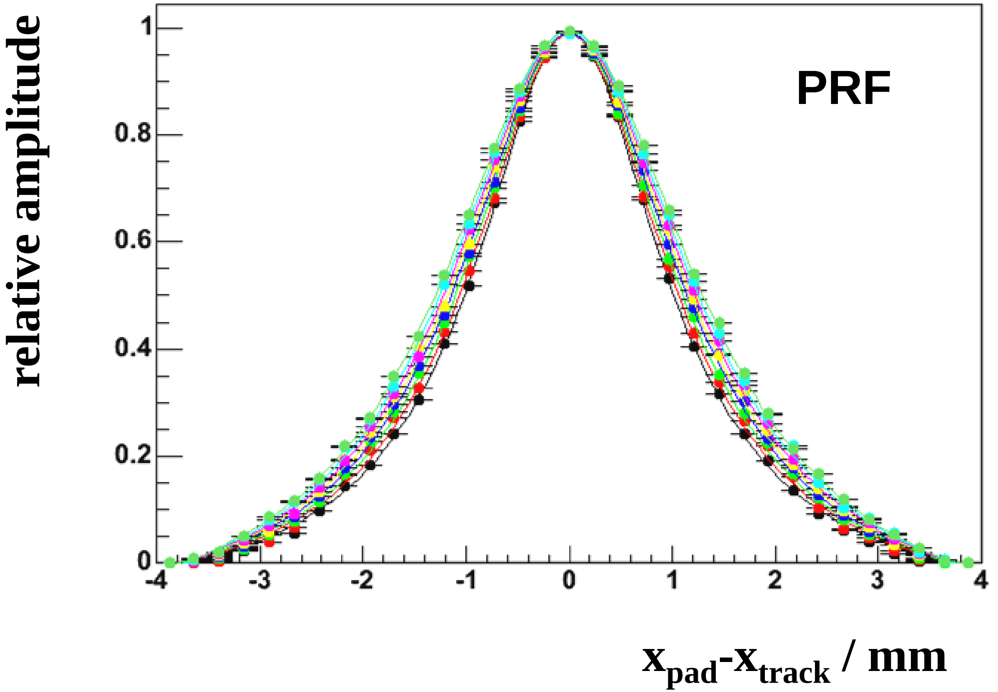
$$PRF(x,a,b)=b^2\,\frac{e^{\frac{-x^2}{2a^2}}}{x^2+b^2}$$

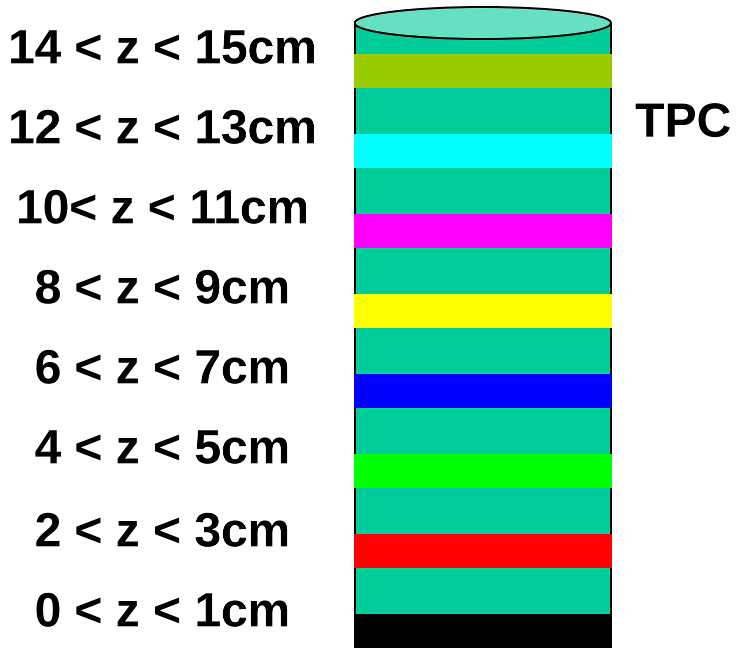
- Only two parameters (simpler model)
- Easier to work with
- Better fits to data
- Error on A_i or $P(x_i)$ [understood to first order... But important for optimal spatial resolution and minimum bias]

PRF versus Z

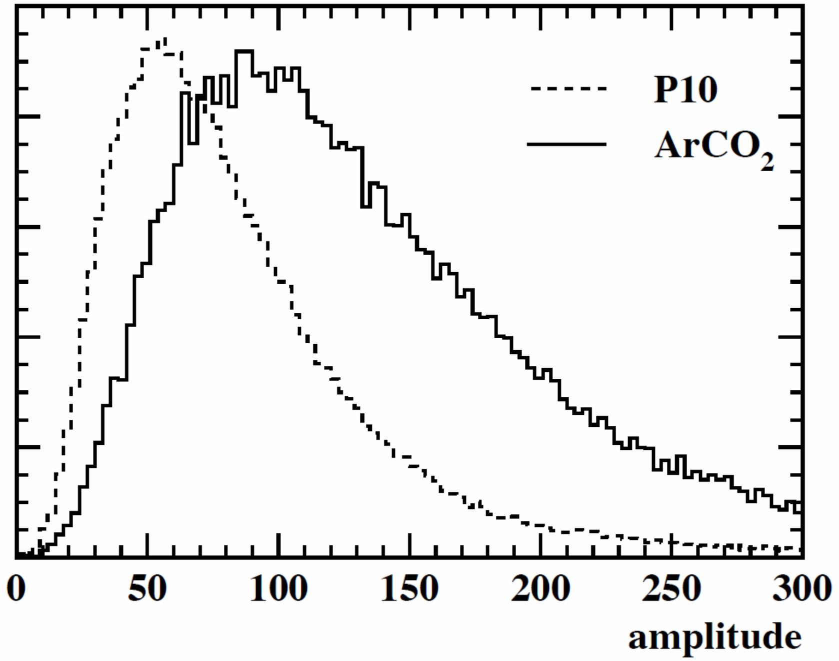

Micromegas
electron
mesh
avalanche
anode
pads
x
Direct signal

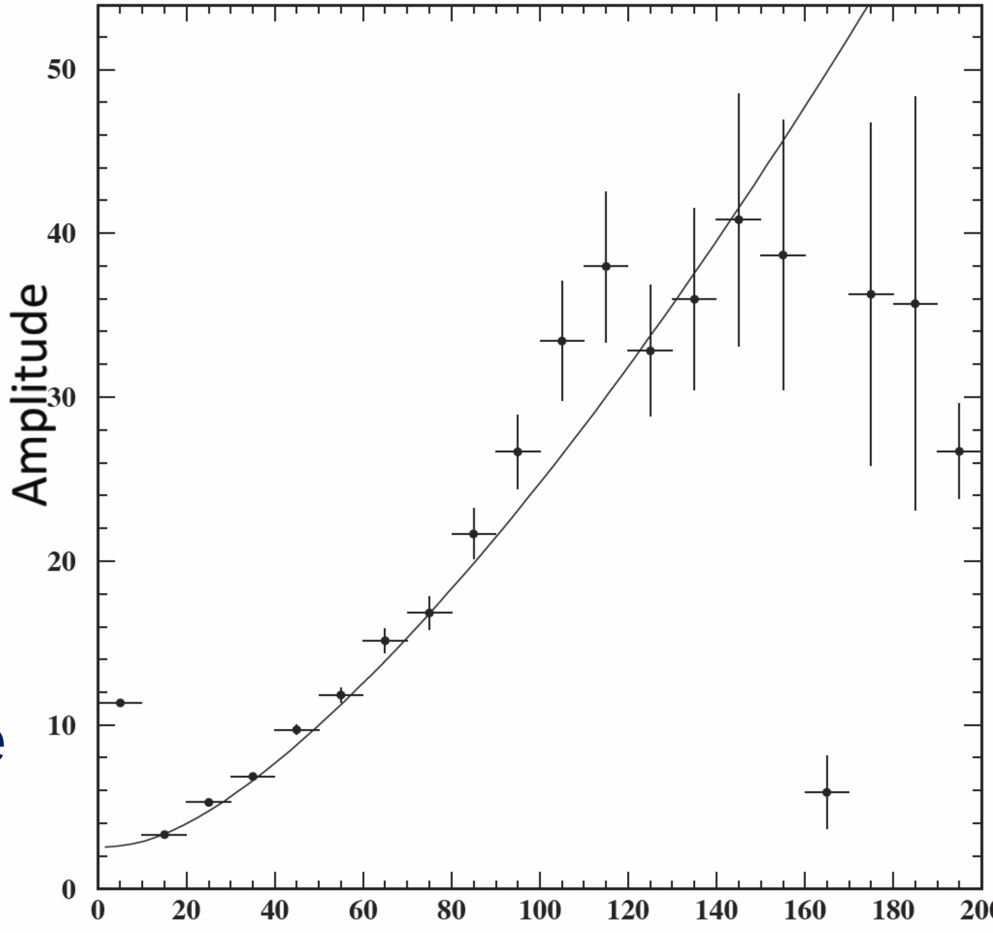

Ar/iC4H10 95/5
w/12^1/2 = 664 µm
diffusion dominant
pad-pitch dominant
noise
σx / mm
z / cm


PRF
relative amplitude
xpad-xtrack / mm


14 < z < 15cm
12 < z < 13cm
10< z < 11cm
8 < z < 9cm
6 < z < 7cm
4 < z < 5cm
2 < z < 3cm
0 < z < 1cm
TPC

Amplitude Errors





Many source of errors on the amplitude: pure statistical, diffusion, noise, track-angle, etc…[under study]

$\sigma_{A_i} = \text{RMS}$

Background - Stand Alone Code

- Our analysis code was developed (mostly) independently at Carleton
- It began life as a FORTRAN95 program, but was eventually machine-code translated into C++
- This code was modified many times over the years, and was used to develop the analysis process from testbeam data (KEK, DESY and LP)
- It was successful for its purpose
- At the beginning of the summer 2010, the code consisted of several unique programs...

Carleton
UNIVERSITY

Background – Stan Alone Code

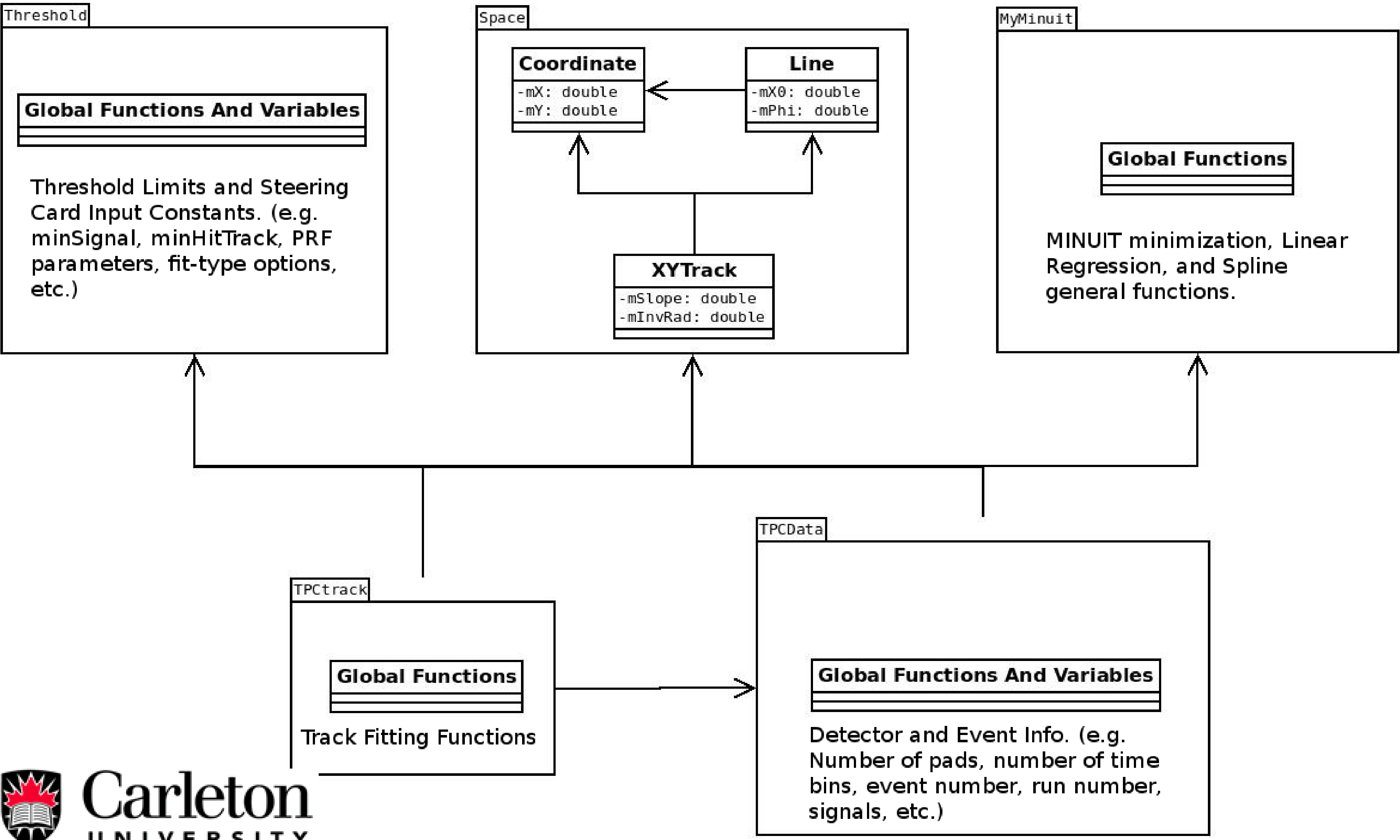
- **NativeToLCIO**
  - Converts data from the native file format of the detector hardware to the LCIO standard
- **Main Code**
  - DD: creates dense data files from LCIO
  - PRF: determines track fits based on pad response function (prf) supplied by user
  - BIAS: calculates and saves values used for bias and reso ROOT scripts
- **ROOT Scripts:**
  - PRF: used to determine goodness of fit of the prf with chosen parameters
  - BIAS: calculates and corrects for signal bias inherent to the detector
  - RESO: calculates the resolution

Improvements – Structure

- The structure of the operation of the code is mostly unchanged, with only small modifications to improve ease-of-use (*ex.* command-line arguments)
- Underlying source code was vastly changed – specifically the main code.
- Proper programming practice was implemented in 2011 (so it can now really be called C++)
- Biggest memory leaks have been plugged
- The new code do not affect the physics results

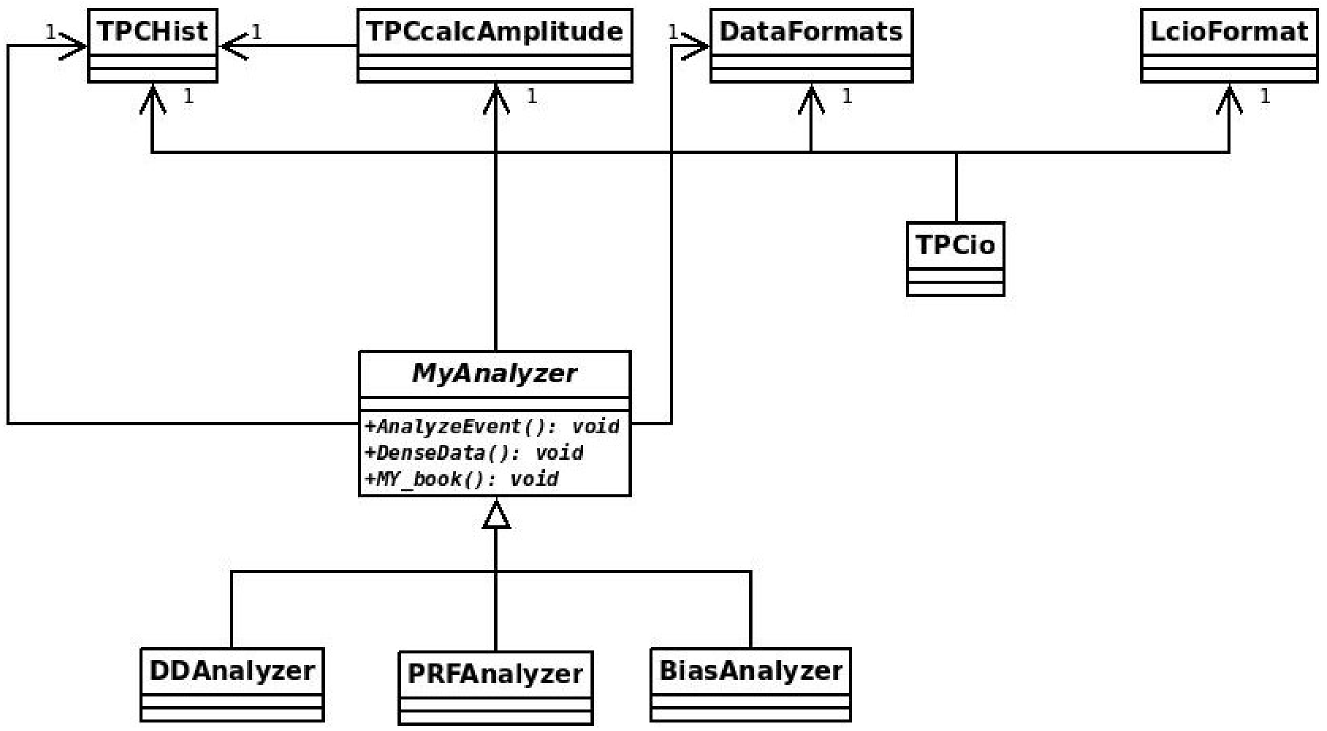
Improvements – Structure

Globals:


Threshold
Global Functions And Variables
Threshold Limits and Steering Card Input Constants. (e.g. minSignal, minHitTrack, PRF parameters, fit-type options, etc.)
Space
Coordinate
-mX: double
-mY: double
Line
-mX0: double
-mPhi: double
XYTrack
-mSlope: double
-mInvRad: double
MyMinuit
Global Functions
MINUIT minimization, Linear Regression, and Spline general functions.
TPCtrack
Global Functions
Track Fitting Functions
TPCData
Global Functions And Variables
Detector and Event Info. (e.g. Number of pads, number of time bins, event number, run number, signals, etc.)

Improvements – Structure

Classes:



Improvements – Limitations

- The new code was much more readable and far simpler to actually develop. However,
  - **It is still fairly slow**
  - **The ROOT scripts are virtually untouched and still very detached from the main code (which, while undesirable for a unified analysis code package, is potentially useful later on)**
  - **Lack of communication between the scripts and the code prevents implementation of some potentially very useful improvements (*e.g.*, reducing overflow rejections)**
  - **Still completely detached from the international effort**

MarlinTPC

- What was the next step for January 2012?
  - Could continue structural improvements and unification of different processes into one large analysis package, but...
  - Could also kill several small flying creatures with one stone and integrate our code with MarlinTPC.
  - A global analysis software needed for the 7-module runs coupled with global track fitting algorithm

Carleton UNIVERSITY

MarlinTPC

- MarlinTPC is the global effort to develop a single analysis code package for all the different prototype TPCs being developed.
- It is far from complete, but it has a solid foundation, and the MPGD TPC is the last prototype that is unrepresented
- Furthermore, now seems to be the optimal time to integrate our code, as demonstrated by the number of small flying animals this one stone will hit...

MarlinTPC

- Small Flying Animal #1:
  - **Modularity of ROOT scripts and code processes is ideally suited to the Marlin design, while Marlin contributes a unified analysis package**
- Small Flying Animal #2:
  - **Many basic elements of the procedure are already implemented in Marlin, such as detector layout (GEAR), constants (LCCD), and global containers (LCIO)**
- Small Flying Animal #3:
  - **The similar pad-based detectors already implemented in Marlin seem to be at about the same stage we are with respect to track fitting, but are lacking bias and residuals checks**

MarlinTPC

•The consequences of each of these points, are as follows:

- **Our pre-existing modularity means that the most difficulty and effort in integration will be due to learning the Marlin environment**
- **The pre-existing Marlin tools means that improved overflow checks should be trivial to implement, and having access to a polar coordinate system means accurate representation of the detector, and an easier extension to curved tracks**
- **The current stage of development on all sides means that our involvement will allow us to pool our resources and develop a strong trackfitting algorithm**

Immediate Plans

- Implement the PRF in Hit finding
  - **First goal is to use PRF parameters determined with stand alone code (FTPC) to find hits with Marlin**
  - **Hits will be found by fitting the PRF to associated groups of pulses (*i.e.*, pulse from the main pad, and the first, and second neighbors)**
  - **Hits will then be used in the Kalman Filter to find tracks**
- Current Progress
  - **A rough draft of the PRFBasedHitFinderProcessor has been written, and is debugging is underway**
  - **The PRF needs to be provided**

Row Hit Position Finding

- Fit: Row contributions using with PRF $P(x_i)$ *for z=constant (known)*:

$$\boxed{\chi^2 = \sum_i \frac{(A_i - A_{peak} P(x_i))^2}{\sigma^2_{A_i}}}$$

PRF parameterization needs to be known [critical]

- Marlin public *Minuit* to minimizes the χ^2 by adjusting parameters x of the PRF.
- Need to know error associated with A_i or $P(x_i)$!!! Default is σ_{A_i}=constant or $\sqrt{A}$ (under study).

$\Rightarrow$ Fit Result:
transverse position of the <u>hit</u> in row

Hit Resolution for Track Fitting

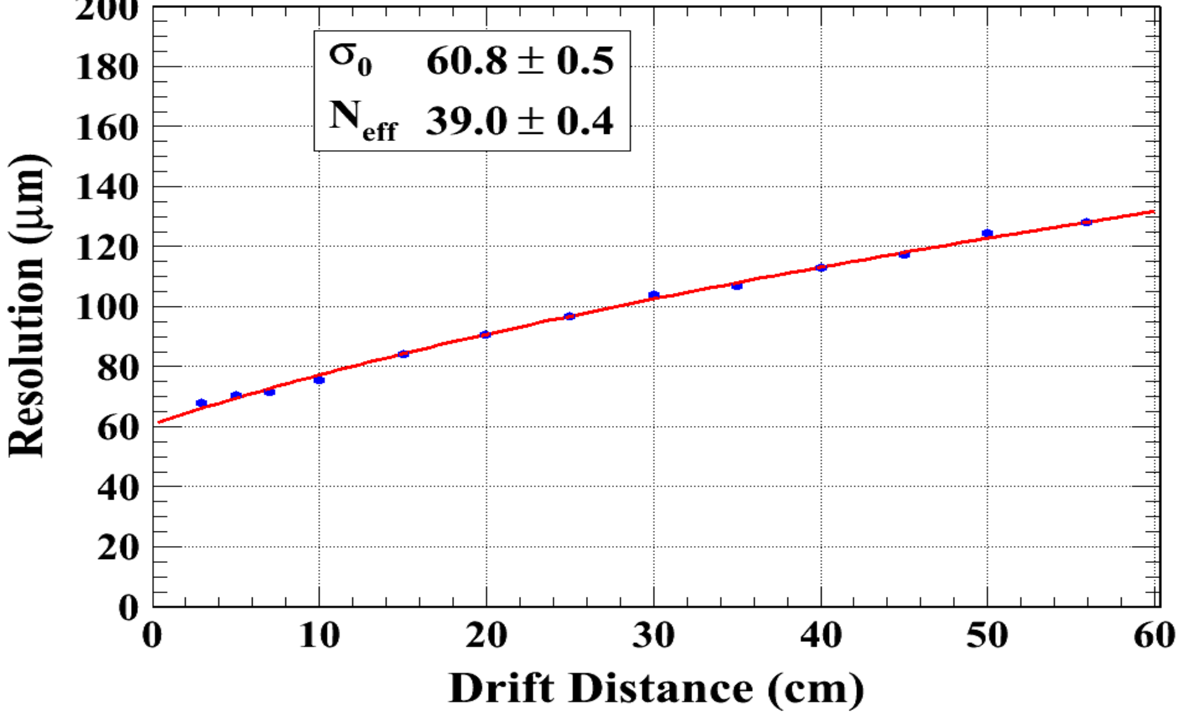
$$\text{Resolution} = \sigma = \sqrt{\sigma_0^{\,2} + \frac{C_d^2 \cdot z}{N_{eff}}}$$

σ_0 : the resolution at Z=0

N_{eff} : the effective number of electrons

Tranverse Plane (x-y)

B=1 T C_d = 94.2 μm/√cm



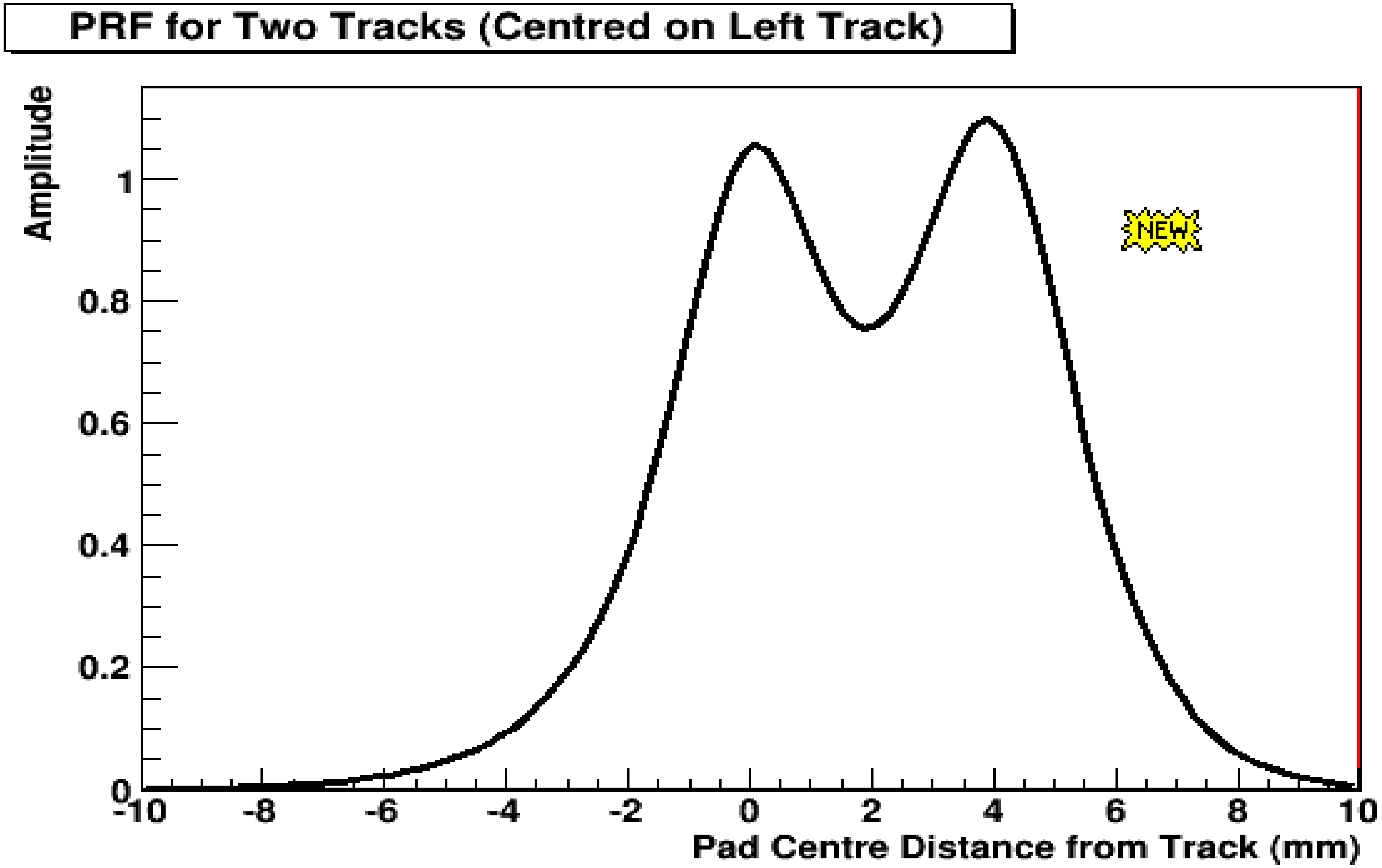
Definitions (recall):

- residual: x_{row} - x_{track}

- <u>resolution</u>: standard deviation of residuals

- we need PRF to find track, but we need track position to determine the PRF

Example: Two-track finding per row



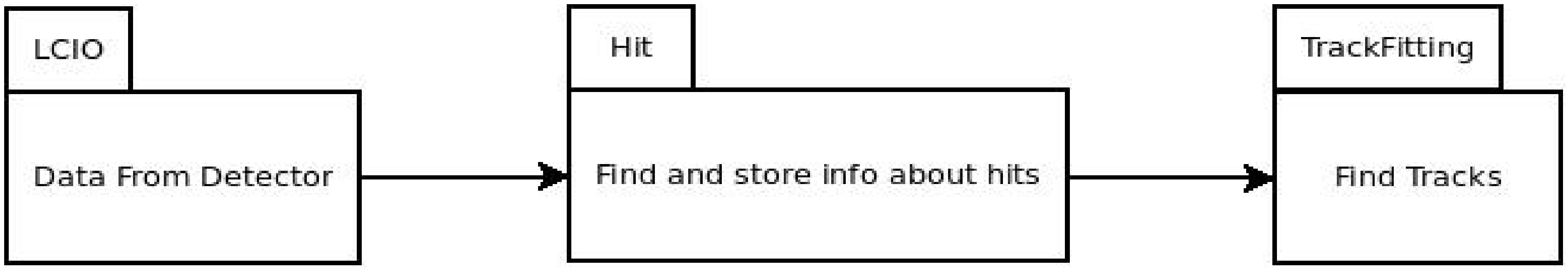
Future Plans

- Implement PRF Parameterization with Errors
  - **Reconstruction in 3D (x,y,z) and properly account for errors when calibrating the PRF, such that the PRF can be used to find 3D hits and their errors in Marlin**
- Implement PRF Calibration in Marlin
  - **Eventually, the calibration process that is being done in the FTPC code will be ported to Marlin. This will allow direct calibration with the 7-module prototype, which could potentially return significantly different parameterizations from previous prototypes**
- Simulation Signals: $N_{electron/ion} \rightarrow (A,t) \rightarrow PRF$
  - **Full understanding of ionization, transport, geometry, and electronics response for 3D tracks**

Future – Simulation

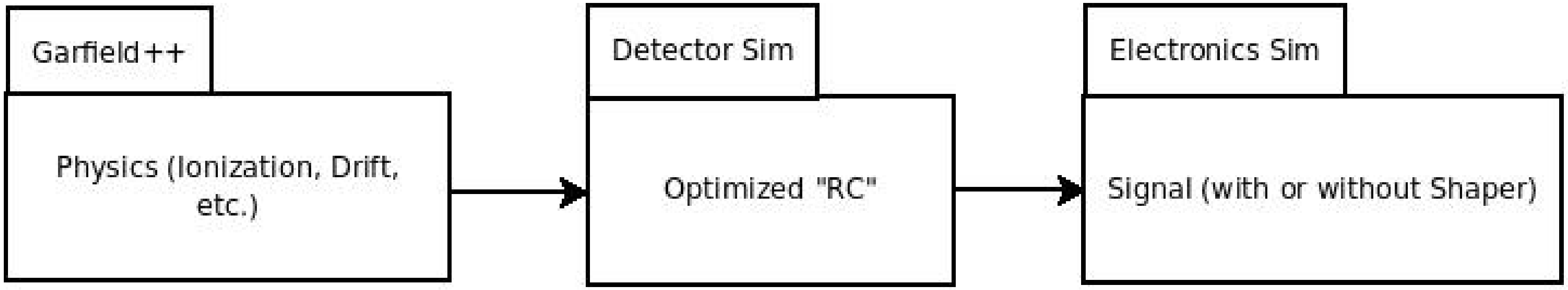
•Concurrently developing simulation of Micromegas detector

•The procedure for the analysis is, basically,



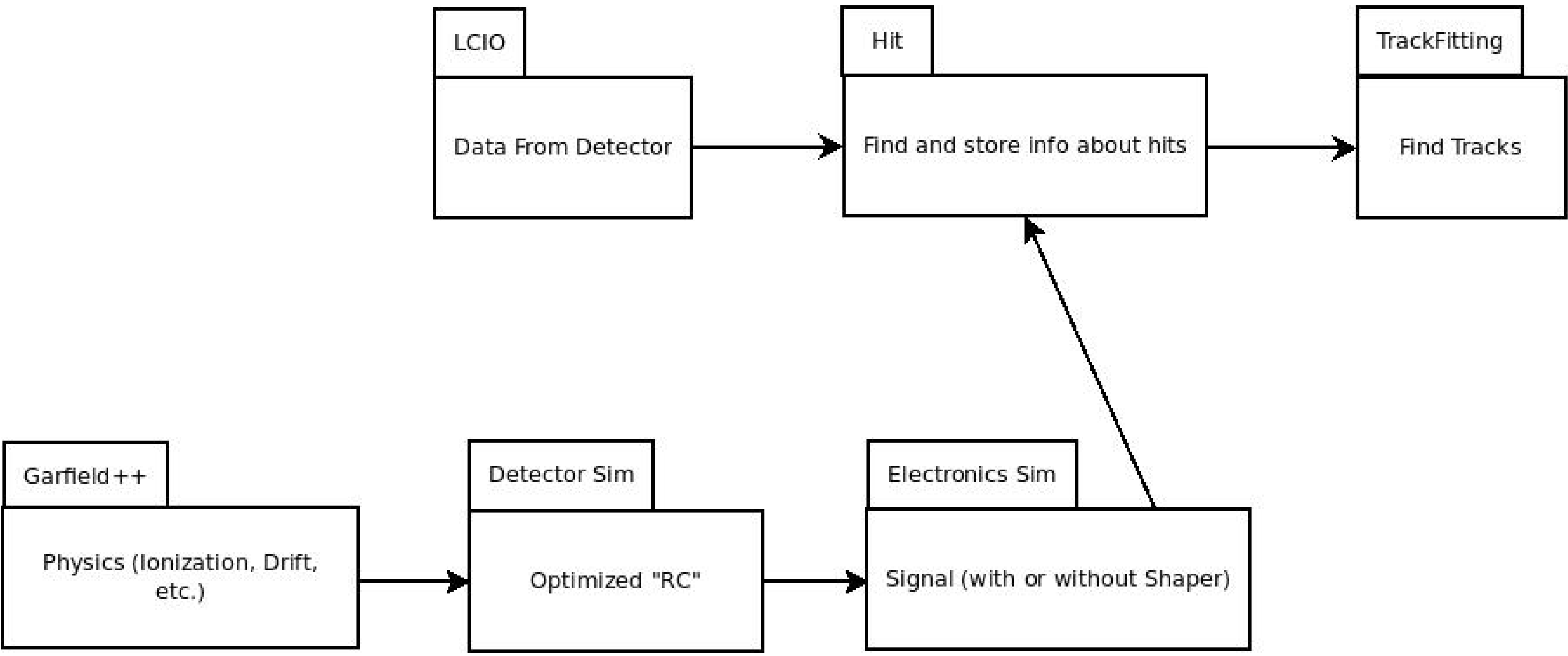
Simulation

The simulation will perform the following calculations,


Garfield++
Physics (Ionization, Drift, etc.)
Detector Sim
Optimized "RC"
Electronics Sim
Signal (with or without Shaper)

And this will fit in with the analysis work, by simply replacing the detector data with the simulated data in the analysis procedure.

Simulation


LCIO
Data From Detector
Hit
Find and store info about hits
TrackFitting
Find Tracks
Garfield++
Physics (Ionization, Drift, etc.)
Detector Sim
Optimized "RC"
Electronics Sim
Signal (with or without Shaper)

Conclusion

- Progress toward **PRFBasedHitFinderProcessor**
  - First C++ implementation in MarlinTPC done
  - Investigation of error on amplitude and time (A,t)
  - Pad Response Function (PRF) to define a hit in 3D
  - Up to now Z=constant (known)
  - Transverse resolution versus Z (σ_0 and N_{eff}) of a hit as well as longitudinal resolution (time resolution) to be used for later "track fitting" (PRF-track is chicken-egg)
- Long Term:
  - PRF determination in MarlinTPC (calibration)
  - Handling Error from (A,t) $\rightarrow$ (PRF,t) $\rightarrow$ (x,y,z) for a Hit to find unbiased track estimators and their uncertainties
  - Simulation of amplitude and time (A,t) to close the loop