“Compressed” Hit Finding

DUNE FD sim/reco meeting

Chris Backhouse

University College London

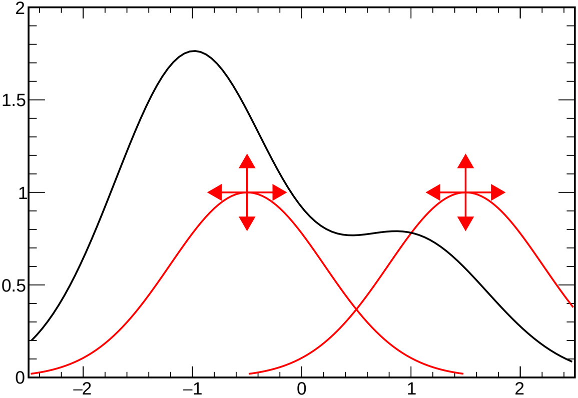
January 13, 2020

Overview

- Raw data from detector (or simulation) – `RawDigit`s in larsoft
- Processed – noise removal and deconvolution → `recob::Wire`
- Most downstream algorithms want to work with `recob::Hit`s

- GausHit algorithm in essense: peak finding ($\sim$ local maxima in trace) used to seed a MINUIT fit for sum of N gaussians
- I'm amazed this works as well as it does

GausHit cartoon



$$\chi^2 = \sum_{i}^{\text{bins}} \left(d_i - \sum_{j}^{\text{peaks}} \alpha_j \exp\left(-\frac{(t-\mu_j)^2}{\sigma_j^2} \right) \right)^2$$

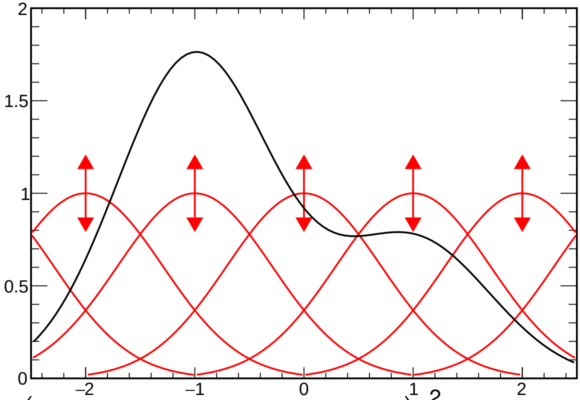
Alternate model

- Named after “compressed sensing” – maybe a bit of a misnomer, perhaps “basis pursuit” is more accurate
- Shares some similarities with the ideas in `SpacePointSolver`

- We’re looking for the simplest sum of gaussians model that explains the data
- In this case “simplest” $\equiv$ fewest gaussians

- Having a model that gives a quadratic expression for the χ^2 is an extremely nice property – guarantees you can always find the global minimum, and quickly
- So – we’re going to use a large number of candidate gaussians, but not allow them to move laterally

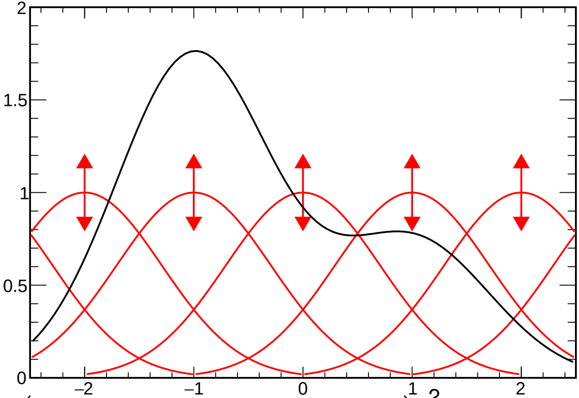
Compressed hit finder cartoon



$$\chi^2 = \sum_{i}^{\text{bins}} \left(d_i - \sum_j \alpha_j \exp\left(-\frac{(t-\mu_j)^2}{\sigma^2} \right) \right)^2 \qquad \alpha_j \geq 0$$

- Only degrees of freedom are the normalizations
- Positivity requirement avoids wild solutions

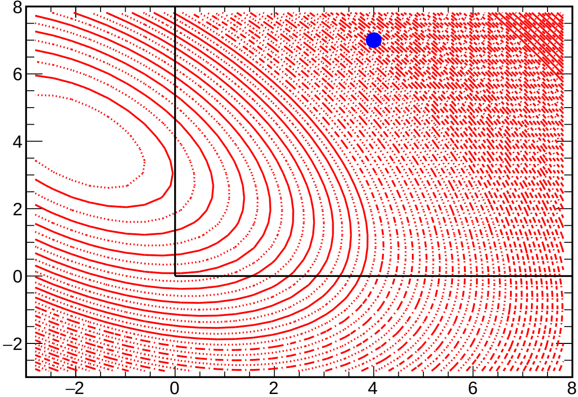
Compressed hit finder cartoon



$$\chi^2 = \sum_i^{\text{bins}} \left(d_i - \sum_j \alpha_j \exp\left(-\frac{(t-\mu_j)^2}{\sigma^2} \right) \right)^2 + \lambda \sum_j \alpha_j \qquad \alpha_j \geq 0$$

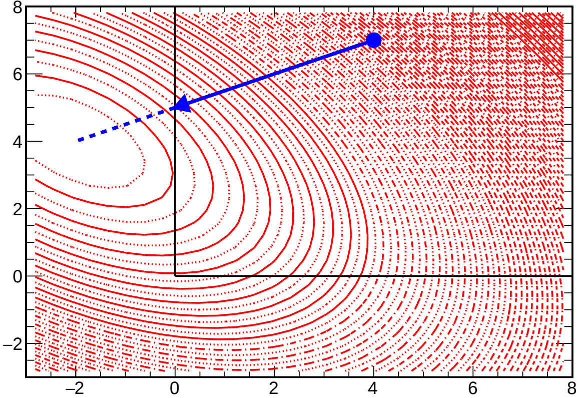
- Only degrees of freedom are the normalizations
- Positivity requirement avoids wild solutions
- Potentially include a regularization term

Finding the minimum



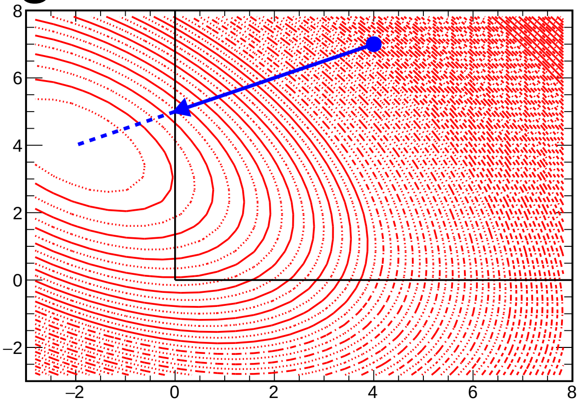
- Quadratic – but global minimum almost certainly has -ve values
- Algorithm goes something like this:
  - Analytically find the minimum for the current set of variables

Finding the minimum



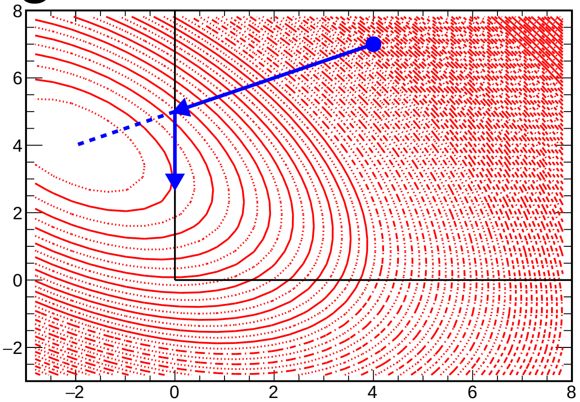
- Quadratic – but global minimum almost certainly has -ve values
- Algorithm goes something like this:
    - Analytically find the minimum for the current set of variables
    - Move as far as possible in that direction until a variable hits zero

Finding the minimum



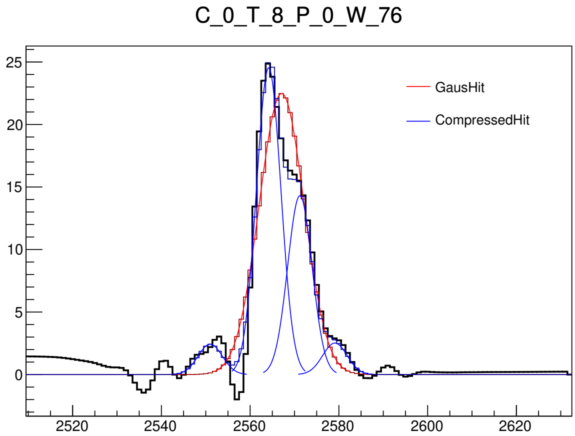
- Quadratic – but global minimum almost certainly has -ve values
- Algorithm goes something like this:
  - Analytically find the minimum for the current set of variables
  - Move as far as possible in that direction until a variable hits zero
  - The set of active variables becomes all variables not at the boundary, plus any for which the derivative points into the space

Finding the minimum



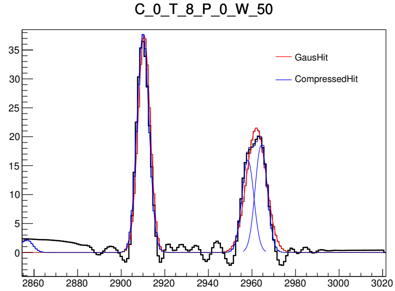
- Quadratic – but global minimum almost certainly has -ve values
- Algorithm goes something like this:
    - Analytically find the minimum for the current set of variables
    - Move as far as possible in that direction until a variable hits zero
    - The set of active variables becomes all variables not at the boundary, plus any for which the derivative points into the space
    - Repeat

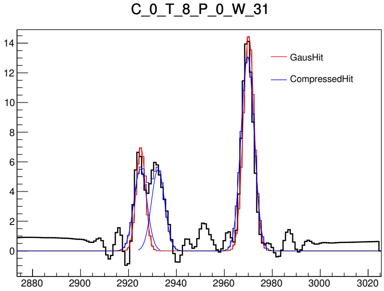
Example fits



- Looking at an FD MC MCC10 file
- Many traces are easy, but `CompressedHit` frequently detects wide hits are composite, where `GausHit` doesn't (no clear second peak)
- Aside: seeing these strange "ringing" artifacts in induction view...

Example fits

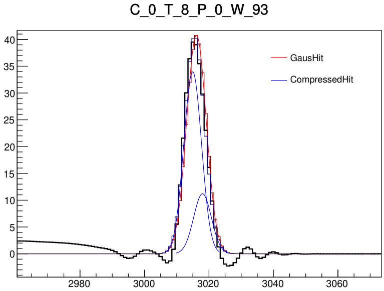


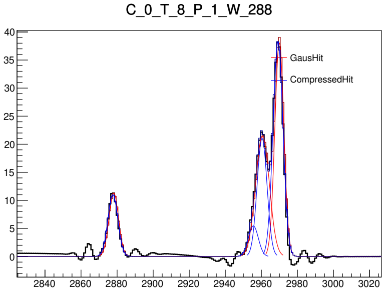


- Another wide hit, plus a phantom hit caused by the ringing

- A case where GausHit straight-up missed a hit

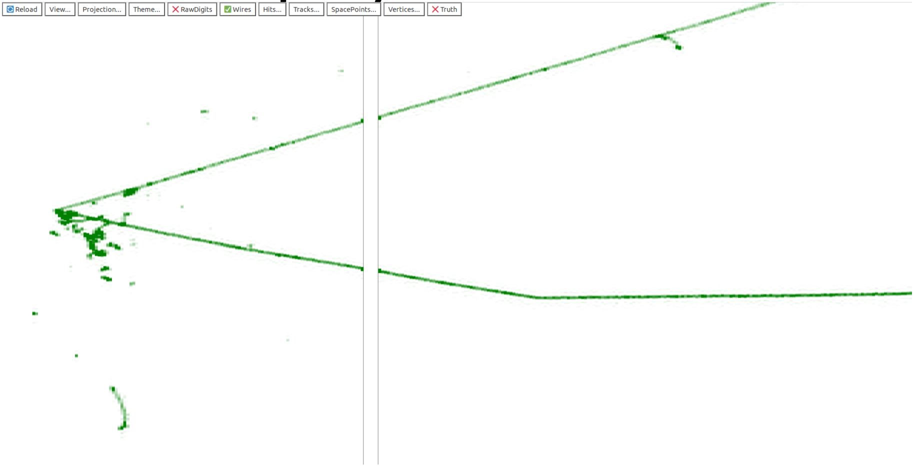
Example fits





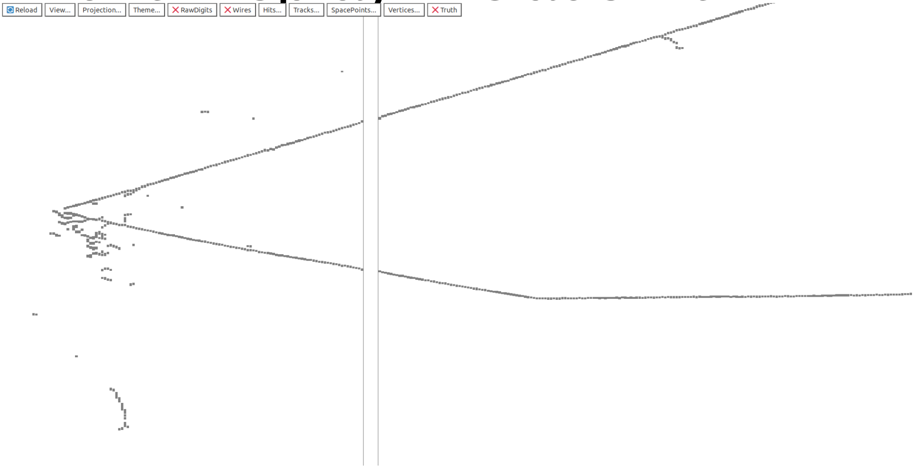
- Also plenty of cases where the merits of the extra hit aren’t clear
- Effect these would have depends on the specifics of the reconstruction

Event Display – Wires



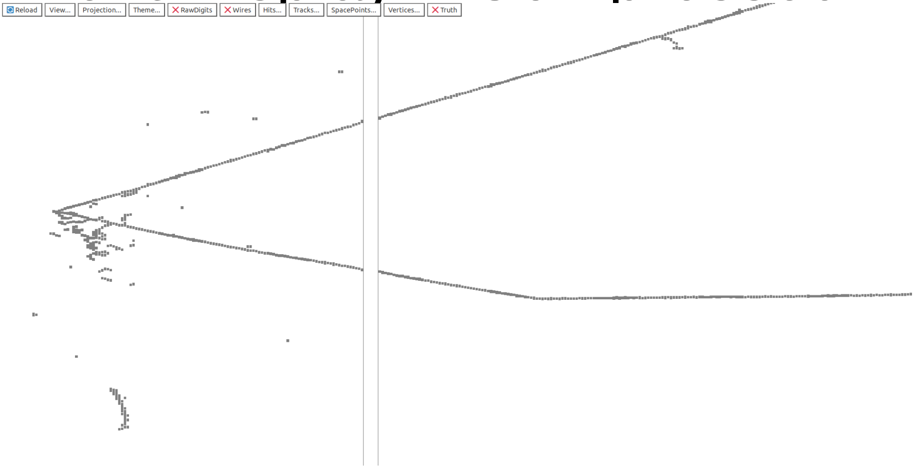
- The underlying event the hits are supposed to reflect

Event Display – GausHit



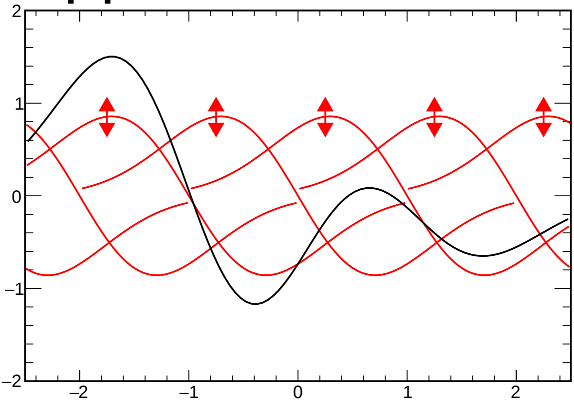
- What we are currently using

Event Display – Compressed



- CompressedHit makes more hits
- In the showery parts it's debatable
- But I think it's clear the vertex region is reconstructed better

Direct application to RawDigits



- Not constrained to any particular hit shape
- In principle can apply the same algorithm directly to RawDigits
- Coded up, but requires some tweaking
- I'm looking for a good model for the bipolar induction shape
- $-x\exp(-x^2)$ is close to what's in the simulation, but not identical
- May just record empirical template from data

Conclusion

- Approach shows good promise
- Problem statement very simple → no possibility of fit failure
- Seems better at detecting partially-overlapping hits
- Could *e.g.* help track particles all the way in to the vertex

- Code available in `larreco` branch `feature/bckhouse_compressedhitfinder`

- Any expert knowledge about induction wire hit shapes?
- Do we have any pre-existing metrics for hit finding performance (*e.g.* matching to IDEs in simulation) better than eyeballing it?

- Goal for collaboration meeting: excercise on real ProtoDUNE data