Reconstruction Framework Status

Christian Bora, Andrew Shultz, Ilya Kravchenko

University of Nebraska, Lincoln

Guy Nir

Weizmann Institute of Science, Israel

July 22, 2015

Introduction

- Object oriented data analysis framework based on ROOT
- Collection and combination of algorithms that process L0/Simulated data to L2 data into a single framework
- Intent is to expand the functionalities of the current AraVertex in AraROOT
- Primary advantage of the framework is to provide fast access and usage of all reconstruction algorithms
- Modular structure for users to easily add customized algorithms

A historical introduction

For UNL, the reconstruction framework grew from several related projects of the past few years,

- Experience of processing data with AraMassProcessing
- L2 re-processing of TestBed datasets, adding the most recent reconstruction techniques via simple function wrappers to the original Hagar's AraVertex

This was a temporary solution of simple wrappers around reconstruction packages taken as-is from individual collaborators, through 2014 until Fall, as the first pass, then moving from wrappers to the full re-write, as planned originally

Reference to previous presentations

Presentations shown in the past about the framework and where they can be found

- Guy's talk about MCMC and Scan: `http://ara.physics.wisc.edu/cgi-bin/DocDB/ShowDocument?docid=1194`
- Guy's updates : `http://ara.physics.wisc.edu/cgi-bin/DocDB/ShowDocument?docid=1168`
- Christian's slides, Jan 26, 2015: `http://ara.physics.wisc.edu/cgi-bin/DocDB/ShowDocument?docid=1160`
- Guy's reco talk, June 23, 2014: `http://ara.physics.wisc.edu/cgi-bin/DocDB/ShowDocument?docid=1063`

Modules

- InputOutput: reading input and writing output event structure to file
- TimeFinding: scanning the waveform to find Δt and σ for each channel
- Geometry: storing antenna position in 3-D space, conversion between coordinate systems and computing a depth dependent index of refraction
- VertexFinding: localizing the source point
- Filters: timing and geometric cuts

Output event structure

EventData is the class that holds the data for each output event and contains:

- The geometry object
- The ice object that contains the index refraction used and its variability bit
- Δt_i, w_i
- (R, θ, ϕ) and a confidence level(χ^2 mapped into [0-1]) for each polarization and each method

The vertex results with good confidence level can be averaged on demand

A framework and a library

- Already 40+ classes
- Analysis can be done by creating objects and calling a sequence of functions in a C++ script
- Alternative option to do analysis by using text based commands for the casual user
- Compiled into a shared library that can be loaded into ROOT for interactive control
- There is enough flexibility to add a generic algorithm by sub-classing one of the classes
- Base classes have most of the functionalities required

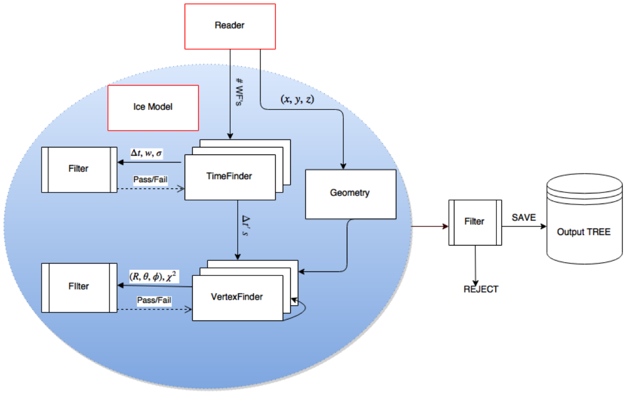
Reconstruction steps

The analysis loop for each event is executed in steps:

1. an input event from a FileReader that loads the waveforms
2. a processor scans waveforms to find time differences and allocates a weight[0-1] to each channel
3. channels with weights > 0 are used for vertex localization
4. an output event in the structure of EventData is written to file

Between any of these steps a filter can be applied to label a step as Pass/Fail.
Before the loop over events, the geometry and the ice model are initialized or loaded.

Reconstruction conceptual view



—: initialized by user ●: saved in the tree

Time finding overview

Module responsible for computing hit times or time differences for channels

- Needs waveforms from Readers, or can directly get truth times from simulation and add a Gaussian error
- Scan over the waveform and outputs Δt, a quality parameter(weight), and error
- The results are saved into a container and fed to the VertexFinding module

Multiple finders can be applied and last finder applied will be used for vertex finding

Time finding methods

These are the methods currently implemented in the framework

1. Coherent summed waveform(CSW)
2. Cross correlation
3. Gaussian smoothing
4. Simple(Maximum bin in WF)
5. Threshold based

Vertex finding overview

Module responsible for finding the vertex

- Channels with weights > 0 are used for vertex localization
- Output source location in both coordinate systems and a confidence level(χ^2 mapped into [0-1])
- Finders can be used in series or in parallel
- Finders can be run on a subset of channels or polarization VPOL/HPOL, combined
- User can decide to store average or best results if more than one finder is used

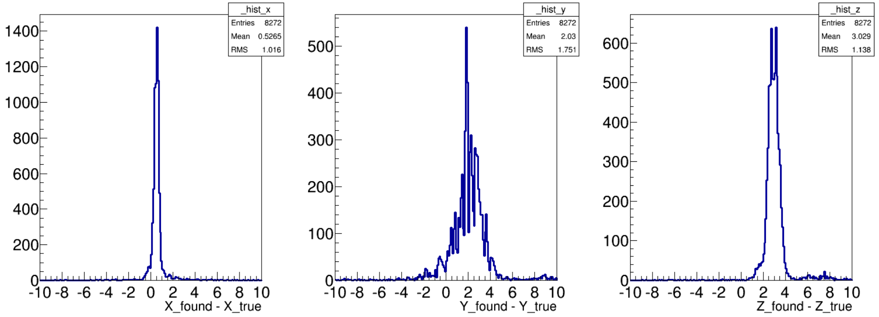
Vertex finding methods

These are the methods currently implemented in the framework

1. 3-dimensional scan
2. Analytic spherical method
3. Interferometry
4. Markov chain monte carlo
5. Minuit

Results from one method can be used as initial position for the next, this is very crucial for minimization methods

Preliminary plots



Preliminary results using L0 TestBed data as input. Analytic spherical method with Threshold time finder on Vpol channels.

Future work

- Complete the work on filters
- Bug fixes and validation against standalone executables of the finders
- Benchmark all finders to compare their runtime and accuracy
- Add shower profile to allow application of shower energy reconstruction
- Include helper functions into classes and writing user guides for casual user and programmer
- Plan to integrate this framework with AraROOT

Conclusion

- We can read data from a built-in EventGenerator/AraSim/L0 data
- Finders mentioned above have already been implemented and validation work against standalone remains to be done
- Framework is fully functional and analysis can be done with the missing component of filters
- The final goal is to integrate all reconstruction algorithms and integrate with AraROOT

Getting started

- Dependencies : ROOT, libROOTFftwWrapper.so, AraROOT, AraSim
- The framework is currently hosted on the ARA repository `https://svnmsn.icecube.wisc.edu/ara/Software/Reconstruct` `trunk/`
- Compile into shared library "libReco.so"
- Example scripts to get started in scripts/
- User guides can be found in docs/