



Flasher Analysis

REU Undergraduates at UWRF

Mason Austin¹, Madeleine Hanley², and Quang Phung³, Suruj Seunarine⁴

Dawn Williams⁵ provided invaluable help: scripts to process data, advice on how to start and proceed with projects, ...

¹Marquette University, ²Colorado School of Mines, ³Normandale Community College, ⁴University of Wisconsin-River Falls,

⁵University of Alabama



Projects

- Is the total charges per event when 6 horizontal LEDs are flashed == sum of charges when six LEDs are flashed individually? (Madeleine)
- Determine the azimuthal orientation of DOMs on strings that flashed in 2017 (Quang)
- Fix broken 2017 All Purpose Flasher Set wiki page (Mason)



Data Set

- 2017 All Purpose Flasher Data Set
 - Runs 129057, 129058, 129063, 129064, 129069, 129073, 129078, 129079, 129083, and 129084
 - All DOMs on Strings 36, 79, 80, 83-85 flashed
 - Single Horizontal LEDs (7-12) Flashed individually
 - All 6 horizontal LEDs flashed at the same time
 - Brightness=40
 - Width=20
 - Rate=10 Hz



Flasher Data Processing

- **I3WaveCalibrator**
 - Applies calibration constants to transform the contents of raw DOMLaunches (ADC counts) into calibrated waveforms (mV), while correcting for known effects of the electronics
- **WaveDeform**
 - A linear algebra-based algorithm for deconvolving the DOM response to photons. It deconvolves the response from both digitizers simultaneously, improving the accuracy of the fit and preventing edge effects at the boundary of the ATWD.
- **FlasherShiftedPulses**
 - There is a time shift made to the pulses based on observations of a mismatch between flash time and time of first light in receiving DOMs.
- **Charge** on Receiving DOMs is **sum of pulses** from **FlasherShiftedPulses**

```
for pulses in pulse_series:
    #print "Pulses: ", pulses
    pulse = dataclasses.I3RecoPulse()
    q = pulses.charge
    t = pulses.time - ft
    pulse.time = t
    pulse.charge = q
    vec.append(pulse)
shiftedpulses[om] = vec
```

DOM Orientation

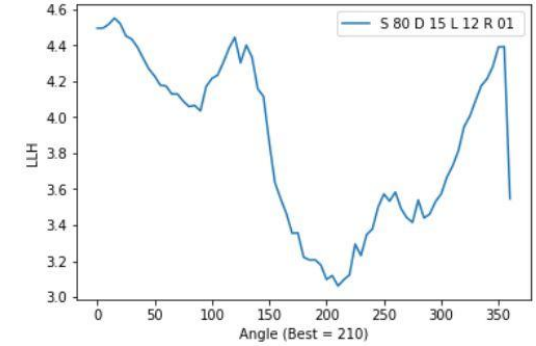
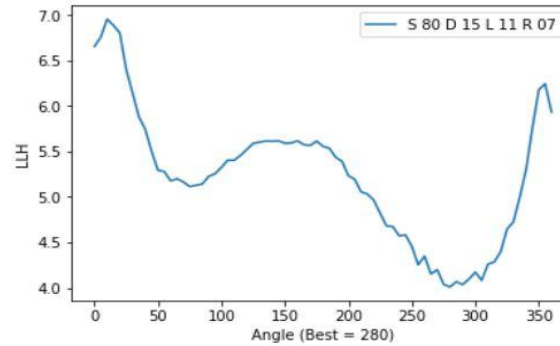
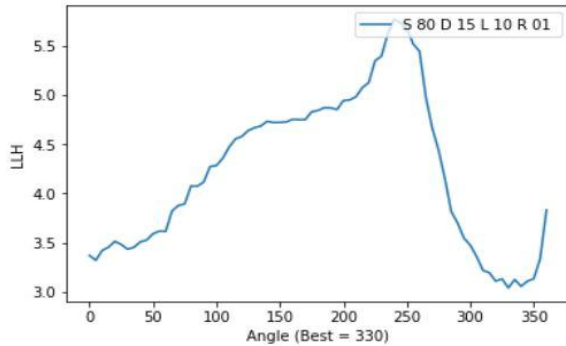
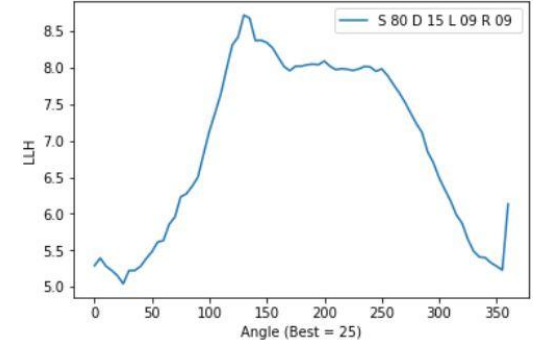
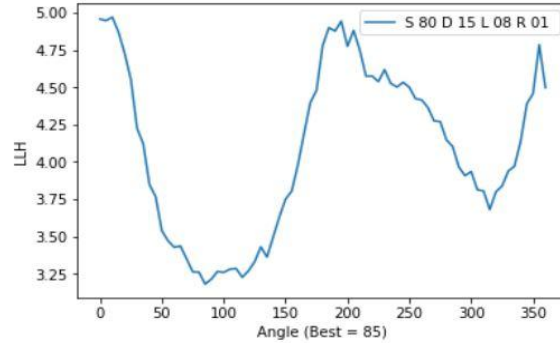
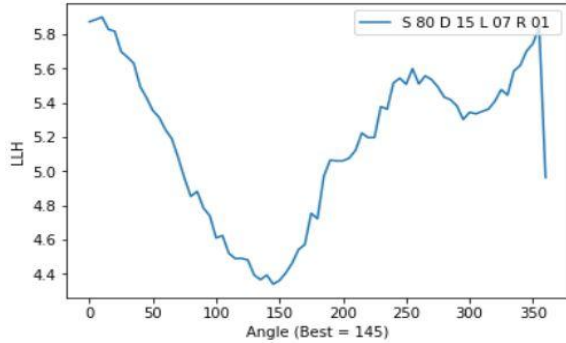
- Previous analysis:
 - 19, 24, 57, 62, 80, 81 orientation
 - https://wiki.icecube.wisc.edu/index.php/LED_Azimuthal_Orientation
 - Use FitAzimuth.py:
- This analysis:
 - Fit orientation of DOMs 36, 79, 80, 83-85
 - Use ppc and the llh from DirectFit method, <http://code.icecube.wisc.edu/svn/projects/ppc/trunk/private/ppc/llh/>
 - Select an orientation angle and calculate likelihood flasher points in that direction
- Compare new results with previous results for string 80 before processing all data.



The screenshot shows a web browser window displaying the 'LED Azimuthal Orientation' page on the IceCube wiki. The page includes a navigation menu, a table of contents, and an overview section. The overview section contains a code block for a Python script named 'fitAzimuth.py'.

```
#!/usr/bin/python
#
# fitAzimuth.py
#
# Created by Chris Wendt
##### temporary issues bypassed:
##### (1) amplitudes seem too large by a factor two, so divide that out
##### (2) data files lack the "number of waveforms averaged" column
#
# usage: fitAzimuth.py geometry.txt fadcData.txt atwdData.txt
#
```

String 80 - DOM 15 - LED 07 to 12: llh vs angle



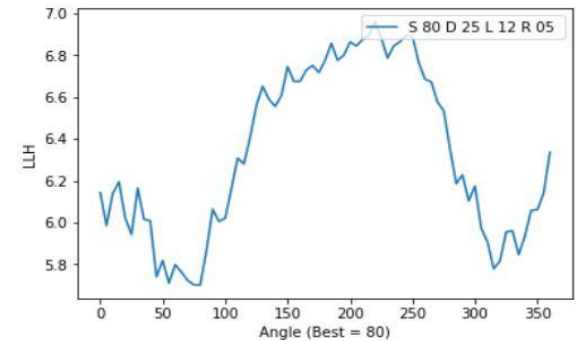
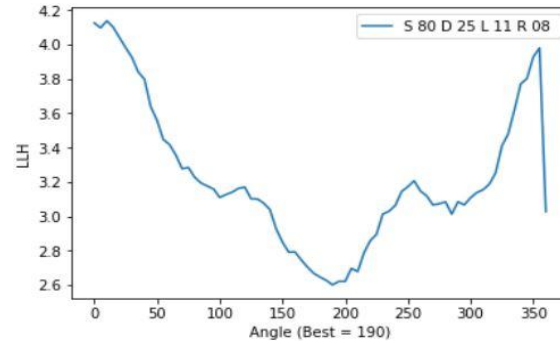
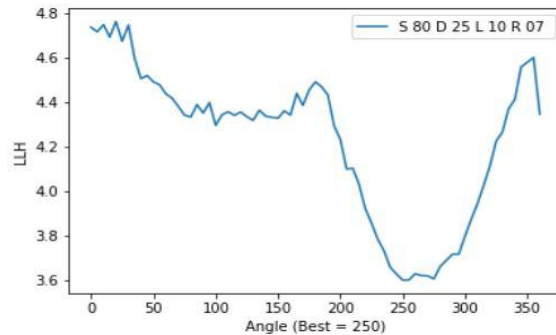
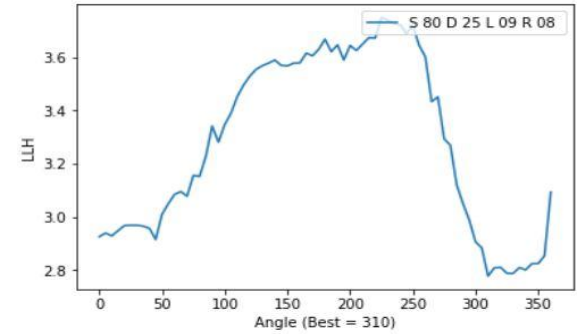
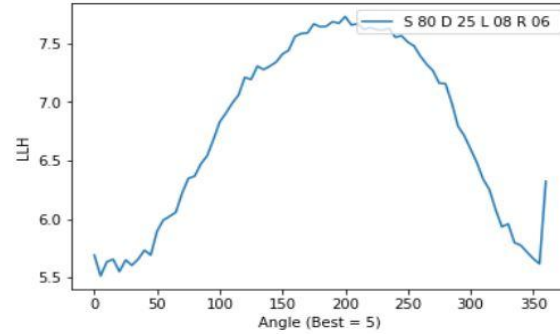
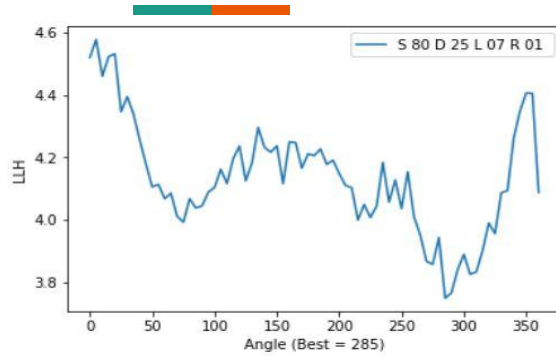


Results for 80-15

LED 7 only was fit in previous analysis

String	DOM	LED	Previous result	This Analysis	Standard deviation
80	15	07	146	146	5
80	15	08	86	94	9
80	15	09	26	9	14
80	15	10	326	333	7
80	15	11	266	289	6
80	15	12	206	211	5

String 80 - DOM 25 - LED 07 to 12: llh vs angle



Results for 80-25

LED 7 only was fit in previous analysis

String	DOM	LED	Previous Result	This Analysis	Standard deviation
80	25	07	70	296	7
80	25	08	10	13	8
80	25	09	310	330	9
80	25	10	250	266	8
80	25	11	190	184	10
80	25	12	130	89	84

Results



- 4 DOMs do not agree with previous angles: 25, 43, 57, 58
- 2 DOMs do not have data to calculate the orientation: 33, 39
- 7 DOMs do not have angles to compare: 26, 27, 33, 50, 51, 52, 54
- 47 DOMs are consistent with previous angles

Next steps:



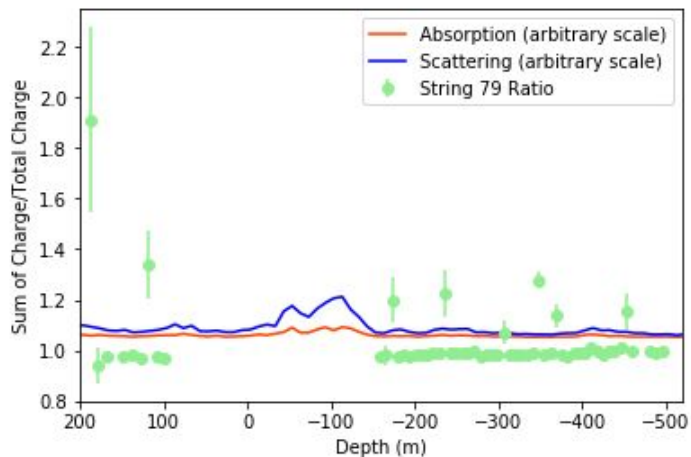
- Calculate the orientation for DOMs on other strings (36, 83, 84, 85)



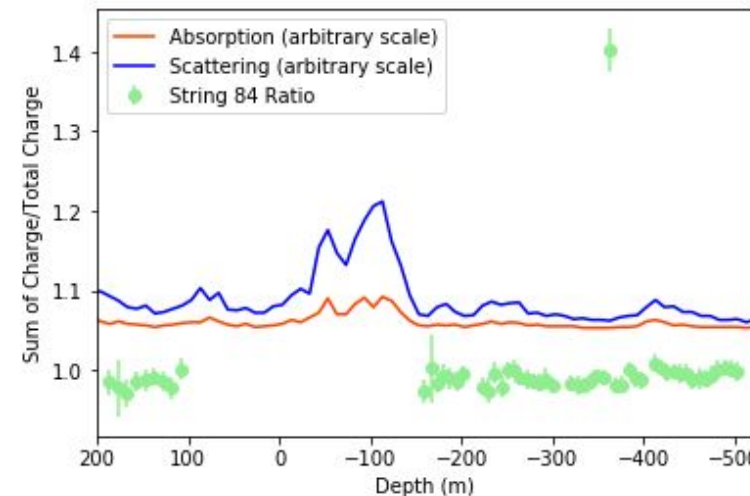
Charge Linearity

- Simple linearity test with new flasher data
- Is the total charges seen when 6 horizontal LEDs are flashed == sum of charges when six LEDs are flashed individually?
- Bash script to generate h5 files
- Edit charge calculation code to calculate ratio

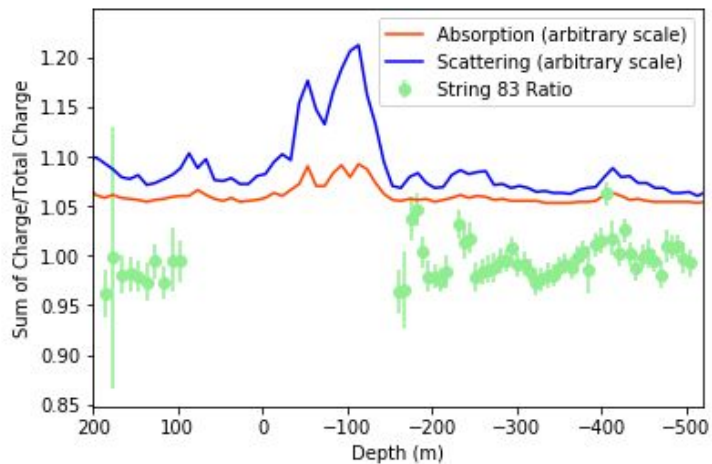
Ratio Of Sum of 6 Individual LEDs vs All 6 Flashing on String 79 with Ice Properties



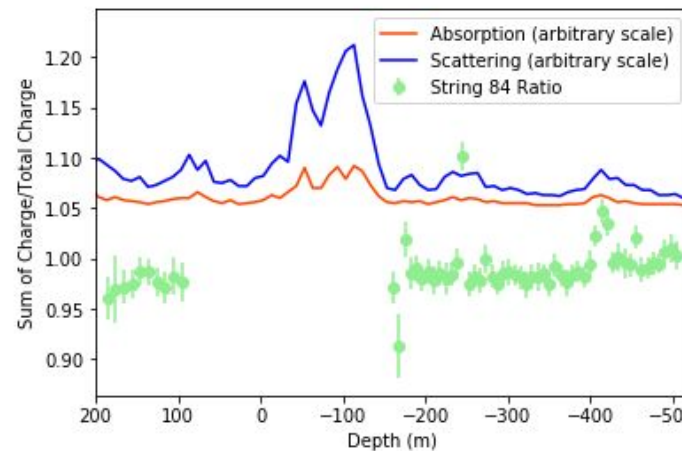
Ratio Of Sum of 6 Individual LEDs vs All 6 Flashing on String 80 with Ice Properties



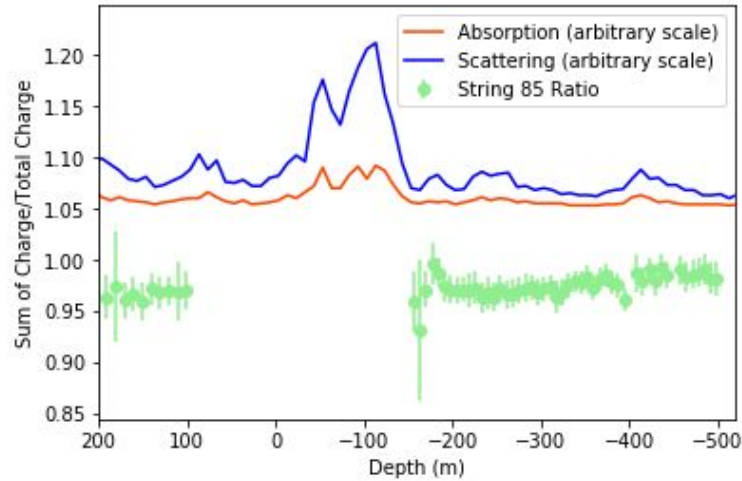
Ratio Of Sum of 6 Individual LEDs vs All 6 Flashing on String 83 with Ice Properties



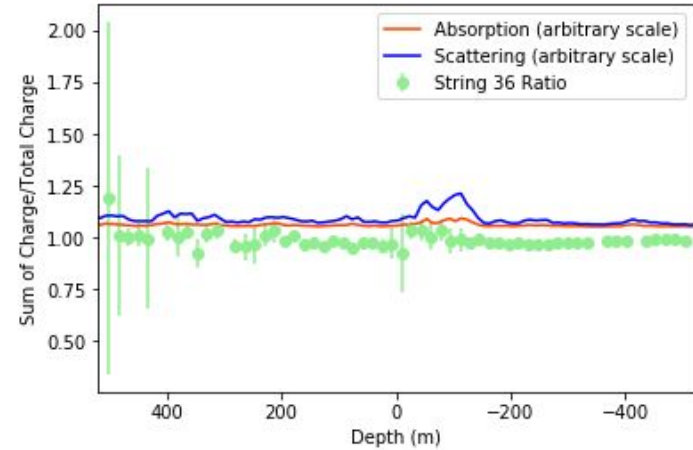
Ratio Of Sum of 6 Individual LEDs vs All 6 Flashing on String 84 with Ice Properties



Ratio Of Sum of 6 Individual LEDs vs All 6 Flashing on String 85 with Ice Properties



Ratio Of Sum of 6 Individual LEDs vs All 6 Flashing on String 36 with Ice Properties



- Ratio of sum of charge/total charge is generally less than 1
- Opposite to what is expected from saturation
- Maybe some correlation (by eye) with ice properties?
- Unsure what it all means



Flasher Wiki Page

- Wiki page contains info and data files for flashers.
- Needs to be completed for ALL Purpose Set 2017

80-1

run	subrun	string number	dom number	brightness	window	delay	mask	rate
129064	43	80	1	40	20	0	0040	10
129064	45	80	1	40	20	0	0080	10
129064	47	80	1	40	20	0	0100	10
129064	49	80	1	40	20	0	0200	10
129064	51	80	1	40	20	0	0400	10
129064	53	80	1	40	20	0	0800	10
129064	55	80	1	40	20	0	0fc0	10

36-12

run	subrun	string number	dom number	brightness	window	delay	mask	rate
129058	87	36	12	40	20	0	0040	10
129058	89	36	12	40	20	0	0080	10
129058	36	12	40	20	0	0100	10	
129058	36	12	40	20	0	0200	10	
129058	36	12	40	20	0	0400	10	
129058	36	12	40	20	0	0800	10	
129058	36	12	40	20	0	0fc0	10	



Python Code

- The Wiki Page has its own syntax, which can arrange the information into tables.
- Manually inserting all the information is time consuming.
- Generating a code that outputs the information as wiki Syntax is more efficient.
- SQL Data File, Flasher_Data_2017, Python Code, Wiki Syntax

Next Steps

- The code is now complete!!!
- Multiple students will work on verifying the information in the tables is correct.

36-1

run_id	run	subrun	string	om	brightness	width	delay	mask	delay
79227	129057	1	36	1	40	20	0	0040	10
79230	129057	3	36	1	40	20	0	0080	10
79233	129057	5	36	1	40	20	0	0100	10
79236	129057	7	36	1	40	20	0	0200	10
79239	129057	9	36	1	40	20	0	0400	10
79242	129057	11	36	1	40	20	0	0800	10
79245	129057	13	36	1	40	20	0	0fc0	10