Making EAARL Data Open

David Nagle

St. Petersburg Coastal and Marine Geology Center

JALBTCX Workshop, July 2016

U.S. Department of the Interior
U.S. Geological Survey

Overview

- EAARL and ALPS
- Problem and Solutions
- Waveform analysis algorithms

EAARL and ALPS

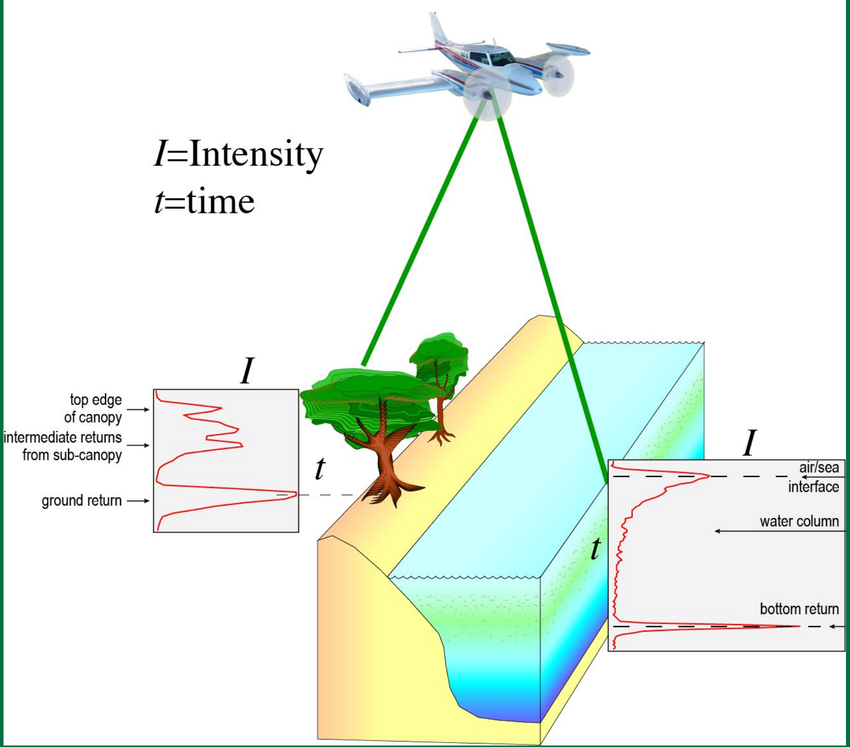
Experimental Advanced Airborne Research Lidar

EAARL–A (2001–2011)

- 532-nm green laser
- Relatively short laser pulse (1.3ns)
- Narrow receiver field-of-view (1.5-2 mrad)
- Digitized signal backscatter full waveforms

EAARL–B (2012–2014)

- Shorter laser pulse (0.7ns)
- Increased point density using 3 channels
- Added 4th channel with 18 mrad FOV for penetrating deeper water



USGS

Airborne Lidar Processing System (ALPS)

Core functionality:

- Post-processing to derive point clouds, using one of several algorithms depending on target type
- Automated filtering using Random Consensus Filter (RCF)
- Manual editing tools
- Can generate GeoTIFF DEMs, LAS, XYZ

Implementation:

- Most algorithms written in Yorick, with some critical parts in C
- Gridding in IDL
- GUI in Tcl/Tk
- Some scripts in Perl, Bash, Awk, C

Problem and Solutions

Problem & Solutions

Collaborators want to:

- Access the waveform data
- Understand how we generate data products
- Work with our algorithms

Barrier: ALPS / Yorick

Solutions:

- C and Python libraries for EAARL waveform I/O
- Open-File Report documenting algorithms
- Python library for algorithms

USGS

EAARL I/O Libraries

- C library and Python library
- EAARL TLD (raster/waveform data)
  - TLD = Type, length, data (variable-length records)
  - Raster has header + series of pulse records (which contain waveform data)
- EAARL EDB (index)
  - EDB = EAARL Database
  - Raster numbers = index into EDB
  - Raster timestamp facilitates lookup of individual rasters, enabling selective processing of user-defined areas of interest
- Expected ready in 6 to 12 months

USGS

Algorithms

- Open-File Report (OFR)
  - Nagle, David B., and Wright, C. Wayne, 2016, Algorithms used in the Airborne Lidar Processing System (ALPS): U.S. Geological Survey Open-File Report, 2016–1046, 45 p., http://dx.doi.org/10.3133/ofr20161046.
  - Short URL: http://go.usa.gov/xcHrV
  - Complete algorithm-level description of processing workflows
  - Includes pseudo code and Yorick sample code
- Python library
  - Complete though minimal workflow for first return data
  - Expected ready in 6 to 12 months

USGS

Algorithms

OFR + Library

- Waveform analysis:
  - Centroid analysis
- Slant range measurement
- Point projection
- RCF filter

OFR only

- Waveform analysis:
  - Leading edge detection
  - Bottom detection using water-column backscatter modeling

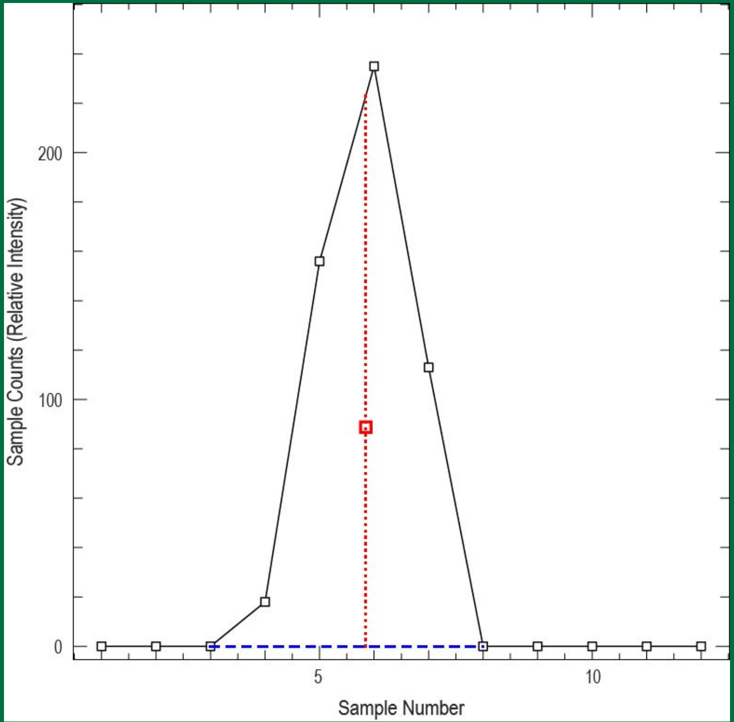
USGS

Waveform Analysis Algorithms

Centroid Analysis (OFR, Python)

- Center of energy in a waveform
- Used for:
  - Transmit
  - First surface (limited to first 12 ns)

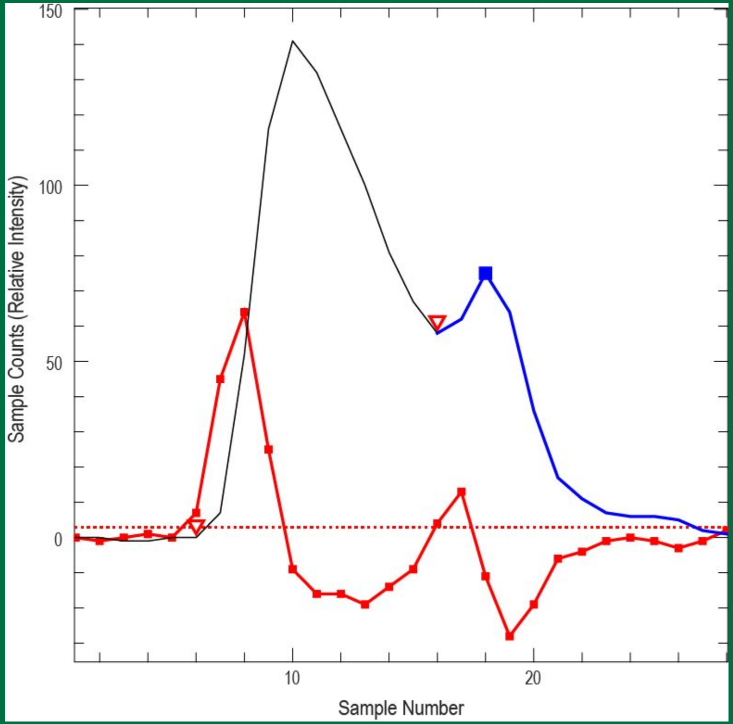
$$C = \frac{\sum_{i=1}^{n} i \cdot wf_i}{\sum_{i=1}^{n} wf_i}$$



USGS

Leading Edge Analysis (OFR only)

- Uses first-difference approach to find last return
- Often permits differentiation of convolved returns
- Used for:
  - Topography under vegetation
  - Shallow submerged topography



USGS

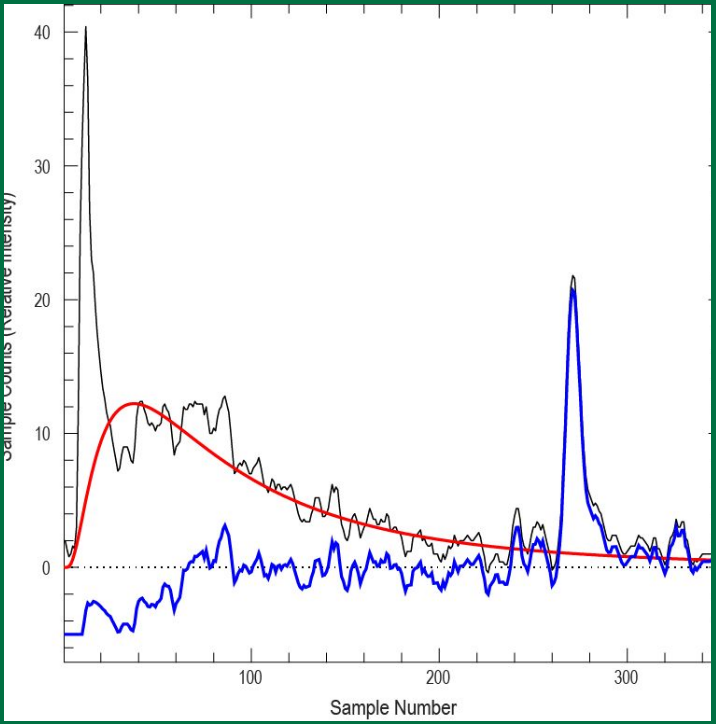
Bottom Detection Using Water-Column Backscatter Modeling (OFR only)

- Used for submerged topography, especially in deeper waters
- Compensates for water column backscatter and absorption by modeling them then subtracting from waveform
- First derivative used to find final peak
- Heuristics used to validate peak

USGS

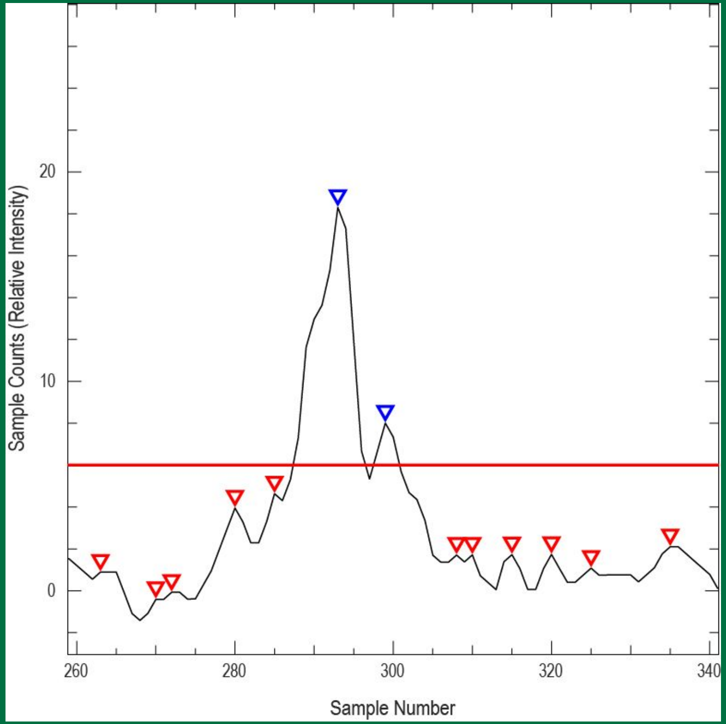
Bottom Detection Using Water-Column Backscatter Modeling (OFR only)

- Used for submerged topography, especially in deeper waters
- Compensates for water column backscatter and absorption by modeling them then subtracting from waveform
- First derivative used to find final peak
- Heuristics used to validate peak



Bottom Detection Using Water-Column Backscatter Modeling (OFR only)

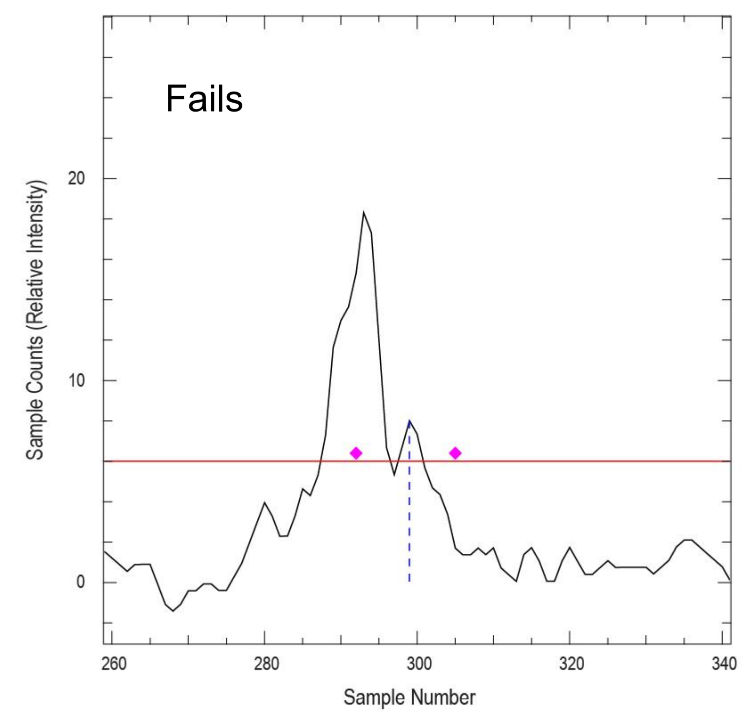
- Used for submerged topography, especially in deeper waters
- Compensates for water column backscatter and absorption by modeling them then subtracting from waveform
- First derivative used to find final peak
- Heuristics used to validate peak



USGS

Bottom Detection Using Water-Column Backscatter Modeling (OFR only)

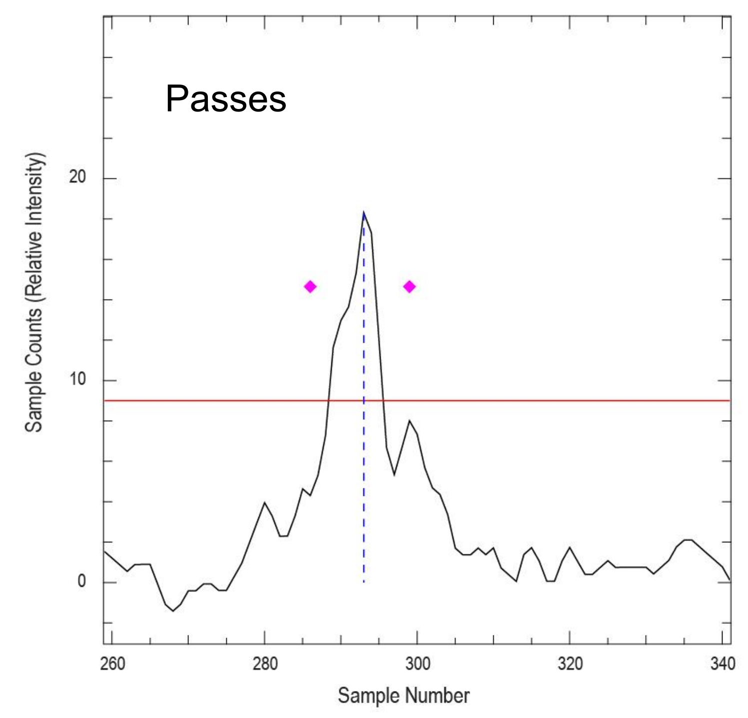
- Used for submerged topography, especially in deeper waters
- Compensates for water column backscatter and absorption by modeling them then subtracting from waveform
- First derivative used to find final peak
- Heuristics used to validate peak



USGS

Bottom Detection Using Water-Column Backscatter Modeling (OFR only)

- Used for submerged topography, especially in deeper waters
- Compensates for water column backscatter and absorption by modeling them then subtracting from waveform
- First derivative used to find final peak
- Heuristics used to validate peak



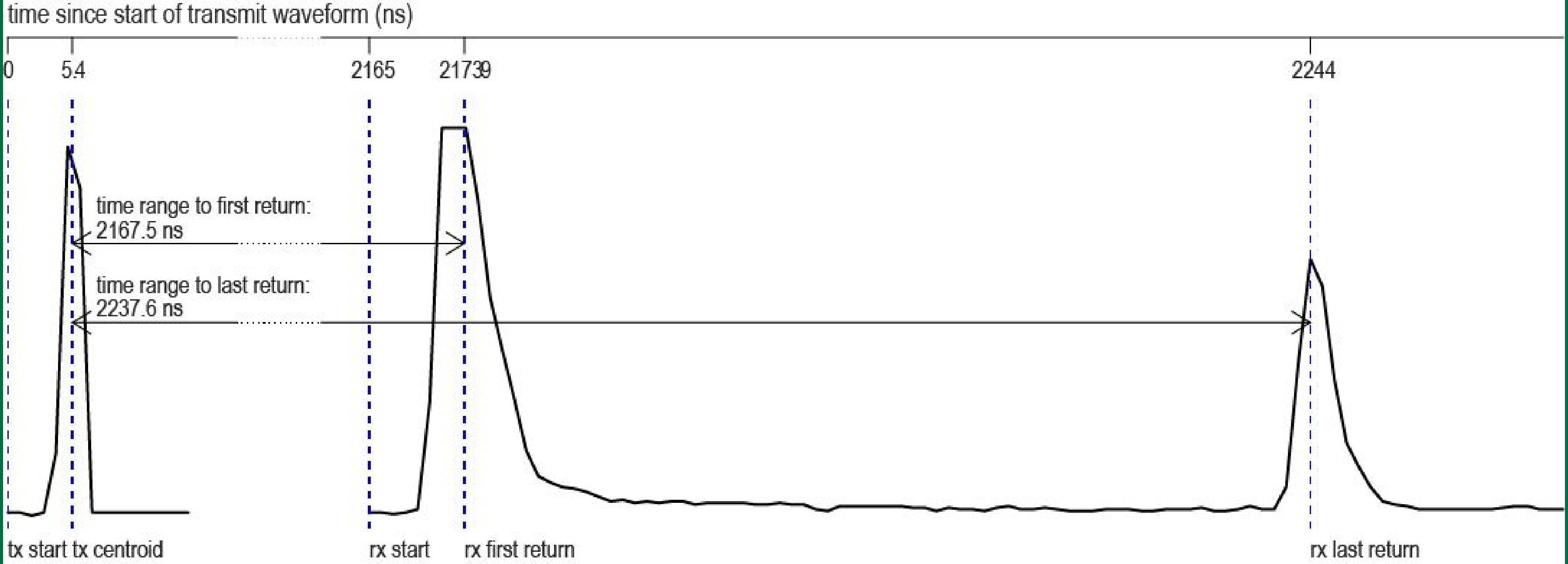
Summary

- OFR 2016–1046 — http://go.usa.gov/xcHrV
- C and Python libraries — 6 to 12 months
- David Nagle — dnagle@usgs.gov

Backup Slides
(in case of questions)

USGS

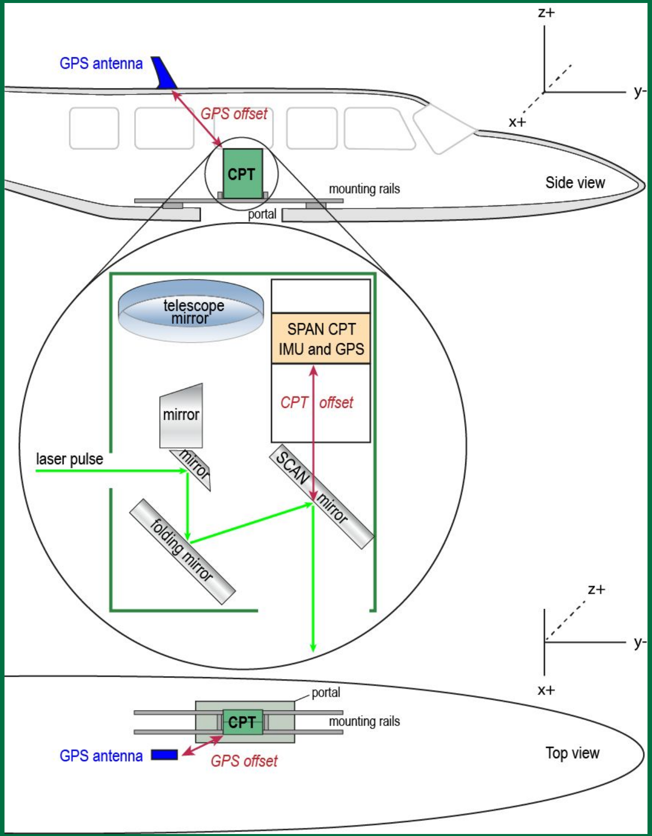
Slant Range Measurement (OFR, Python)



USGS

Point Projection (OFR, Python)

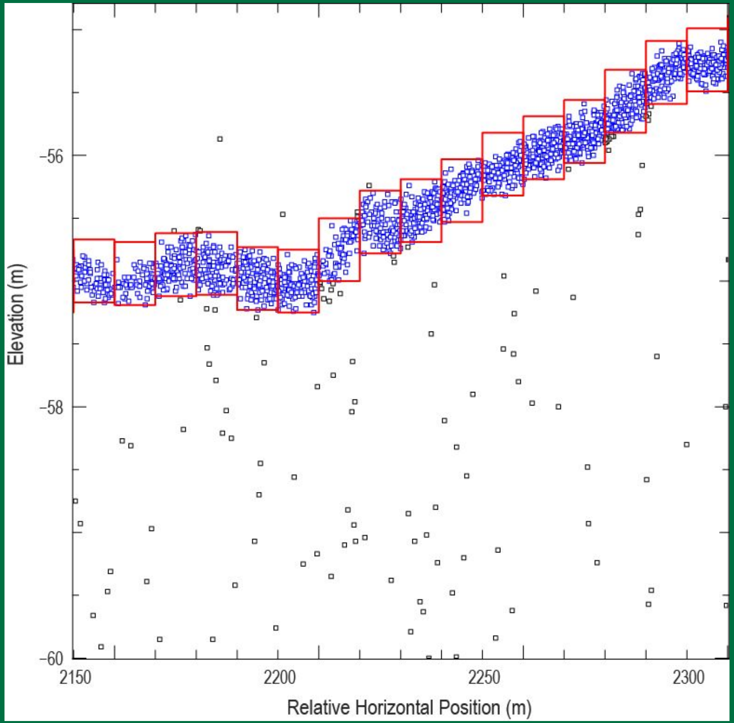
- Projects slant range measurement into real-world coordinates
- Involves:
  - Translations and rotations for changes in frame of reference
  - Calculation of angles of incidence and reflection of laser on oscillating scan mirror
  - Water correction for submerged targets to account for difference in refractive indices for air and water (OFR only)



USGS

Random Consensus Filter (OFR, Python)

- Modeled after concepts of RANSAC
- Intended to reduce noise
- One-dimensional RCF
    - Finds window of highest concentration in set of values
- Gridded RCF
    - Divides points into spatial grid cells
    - Applies one-dimensional RCF to each cell
- Multi-gridded RCF
    - Runs gridded RCF multiple times at offsets and accepts points that pass any
    - Gives each point a chance to be closer to center of cell



USGS

Gridding for Digital Elevation Models (OFR only)

- Implemented in IDL
- Uses Delaunay triangulation to generate irregular mesh
- Removes triangles from mesh based on thresholds for area and longest side
- Uses IDL built-in functions to grid the mesh
- Output saved to GeoTIFF

USGS

Summary

- OFR 2016–1046 — http://go.usa.gov/xcHrV
- C and Python libraries — 6 to 12 months
- David Nagle — dnagle@usgs.gov
- Project chief: Dave Zawada — dzawada@usgs.gov

USGS