ATM

Airborne Topographic Mapper

Airborne Topographic Mapper HDF5 Waveform Data Products

User Guide

Current Version:

https://atm.wff.nasa.gov/files/browser/ATM_waveform_product_user_manual.pdf

Version 1.2

November 20, 2019

Introduction **2**

Theory of Measurements **2**

Data Acquisition Methods **2**

Derivation Techniques and Algorithms **2**

Trajectory and Attitude Data **3**

Overview of the TX and RX Range Gate Structure **3**

Range Determination **4**

File Structure and Organization **5**

Level 1B ATM Geolocated Spot Elevation Measurements 5

Waveform Data 6

9) Identifying corresponding 532 and 1064 nm waveforms **9**

10) Test file for code development **9**

MATLAB® functions **11**

9.1 ATM_WVFM_READER 11

9.2 ATM_INFO 12

9.3 PLOT_WVFMS 12

IDL functions **14**

10.1 idl_read_ATMhdf.pro 14

1) Introduction

The Airborne Topographic Mapper (ATM) waveform data products are Level-1B data products that are compliant with NASA's Earth Science Data and Information System (ESDIS) approved standards for use in NASA Earth science data systems. In order to promote the use of ATM airborne laser altimetry data with science data products from satellite missions, such as ICESat-2, the ATM waveform data product is similar in format and structure to ICESat-2's ATLAS Science Algorithm Software Standard Data Product [*Lee*, 2018]. It is implemented using the HDF5 standard as defined by the HDF® Group. Currently, there are 3 different ATM waveform data products available from the National Snow and Ice Data Center (NSIDC):

Instrument	**Scan Angle [°]**	**Wavelength [nm]**	**Data Product**
ATM-T6	15°	532 nm	ILATMW1B
ATM-T7	2.5°	532 nm	ILNSAW1B
ATM-T7	2.5°	1064 nm	ILNIRW1B

Data from the dual-color, narrow-scan (2.5°) instrument are acquired by a single laser and transceiver, but by separate data systems. The near-infrared (NIR, 1064 nm) and green (532 nm) wavelength measurements are co-located spatially and synchronized temporally. The NIR and green laser shots are stored in separate HDF5 data files. For direct comparisons the data records from the two data systems need to be matched, as occasionally one or the other system will miss recording a laser shot. This co-registration can be performed with example Python™ or MATLAB® code finding matching shots in both data products. The process is described in Section 9.

2) Theory of Measurements

A laser altimeter measures range from the instrument to a target by measuring the elapsed time between emission of a laser pulse and detection of laser energy reflected by the target surface. Range to the target is calculated as half the elapsed emission/return time multiplied by the speed of light. Target range is converted to geographic position by integration with platform GPS and attitude or Inertial Measurement Unit (IMU) information.

3) Data Acquisition Methods

The ATM instrument package includes suites of LiDAR, Global Positioning System (GPS) and attitude measurement subsystems. The instrument package is installed onboard the aircraft platform and calibrated during ground testing procedures. Installation mounting offsets, the distances between GPS and attitude sensors and the ATM LiDARs, are measured using surveying equipment. One or more ground survey targets, usually aircraft parking ramps, are selected and surveyed on the ground using differential GPS techniques. Prior to missions, one or more GPS ground stations are established by acquiring low rate GPS data over long time spans. Approximately one hour prior to missions both the GPS ground station and aircraft systems begin data acquisition. During the aircraft flight, the ATM instrument suite acquires LiDAR, GPS and attitude sensor data over selected targets, including several passes at differing altitudes over the selected ground survey calibration sites. The aircraft and ground systems continue to acquire data one hour post-mission. Instrument parameters estimated from the surveys of calibration sites are used for post-flight calculation of laser footprint locations. These parameters are later refined using inter-comparison and analysis of ATM data where flight lines cross or overlap. A more detailed description can be found in Martin *et al.*, [2012].

4) Derivation Techniques and Algorithms

Each ATM surface elevation measurement corresponds to one laser pulse. The measurements have not been re-sampled. The transmitted laser pulse and the received backscatter pulse from the ground surface are photodetected and captured by a waveform digitizer. Post-flight processing of the waveforms yields the time of flight between transmitted and received signals. This time of flight value is converted to a distance compensated for speed of light through atmosphere. The scan azimuth of the LiDAR scanner mirror together with the aircraft attitude determine the pointing angle of the LiDAR. GPS aircraft position, pointing angle of the LiDAR, and range measured by the LiDAR are used to compute position of laser footprint on the ground.

5) Trajectory and Attitude Data

Aircraft position is determined by Global Navigation Satellite System (GNSS) systems that incorporate NAVSTAR Global Positioning System (GPS) and, for later campaigns, the Globalnaya Navigatsionnaya Sputnikovaya Sistema (GLONASS). Carrier phase measurements are logged by an antenna and receiver on the aircraft. In post-flight processing, these measurements are combined with similar measurements from static ground stations to produce a kinematic differential solution of the aircraft trajectory at 0.5 second intervals, and more recently at 0.1 second intervals.

Aircraft attitude is logged from a commercial Inertial Navigation System (INS), also known as an Inertial Measurement Unit (IMU).

6) Overview of the TX and RX Range Gate Structure

The analog output from the optical detector is captured by an 8-bit waveform digitizer sampling at a constant rate (2, 4 or 10 Giga samples per second or Gsps). A sequence of samples, or “range bins”, is recorded whenever the signal amplitude exceeds a programmable trigger threshold. Each sequence, or “range gate”, can contain a variable number of range bins depending on how long the signal exceeds the threshold. Each laser shot generates a laser waveform record, which can contain multiple range gates, each of which contains multiple waveform samples. The laser record contains the starting position of each range gate, from which the time of each range bin can be determined. For example, Figure 1 shows a laser waveform reconstructed from a laser record containing 6 range gates: one for the transmitted pulse (TX), and four return gates (RX) from a complex target (trees). Range gate #6 contains two distinct return pulses.

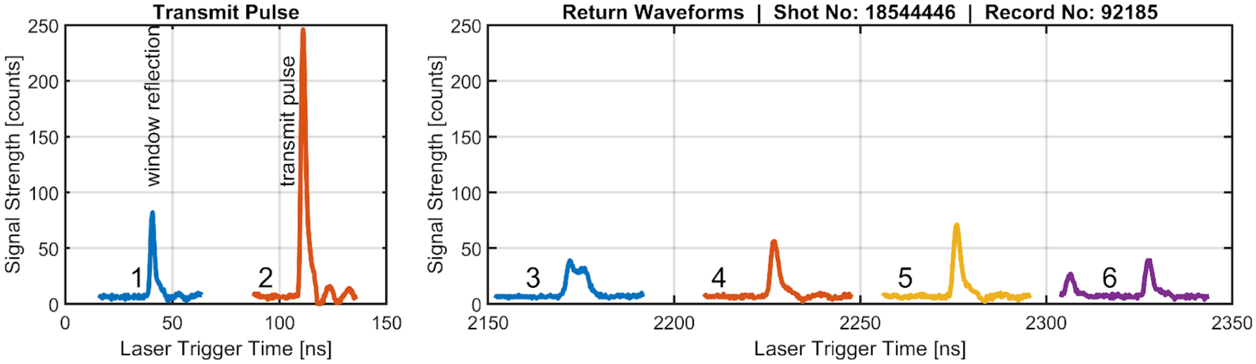

Figure 1: *Example of how ATM waveform data are organized into range gates. The example shows a laser shot over a complex target (tree) that has triggered 4 return range gates (#3 - #6) some of which contain multiple return pulses. The transmit pulse is recorded through a delay fiber (range gate #2) and is sometimes preceded by a window reflection (range gate #1).*

The transmitted laser pulse travels through an optical window in the nadir view port on the aircraft to the target. The reflection of the transmitted laser pulse on the optical window can exceed the amplitude trigger threshold and is then recorded in a range gate. In order to separate the recorded transmit pulse from the window reflection, the transmit pulse is routed through an optical delay fiber several meters in length so it appears several tens of nanoseconds after the window reflection. The length of the fiber can change with various system configurations and its delay is incorporated in the calibration for range determination. Since the window reflection does not occur on every laser shot, sometimes the recorded transmit pulse is in range gate #1 or range gate #2. Figure 1 shows an example that includes a window reflection. Therefore the recorded transmit pulse that is used for the ATM range determination is in range gate #2. The range gates that contain the transmit and receive pulses that were used for range determination are identified in the information included for each laser shot (Section 8).

The capability of recording multiple range gates of varying lengths for each laser shot requires a pointer and indexing scheme to access the waveform data within an HDF5 file that is described below.

7) Range Determination

Transmit and receive waveforms are captured in separate range gates. The time delay between transmitted and received waveforms includes the delay between the TX and RX range gates and the “tracked” location of the pulse within each gate. When the laser fires, an electronic trigger starts the digitizer counting the number of elapsed sampling intervals. The gate start position is the value of this counter at the first bin of the gate. The time relative to the trigger, or “trigger time”, of any range bin can be computed as the gate start position plus the bin position within the gate, multiplied by the digitizer sampling interval. The time between any range bins in the two gates can be accurately computed as the difference between the two trigger times.

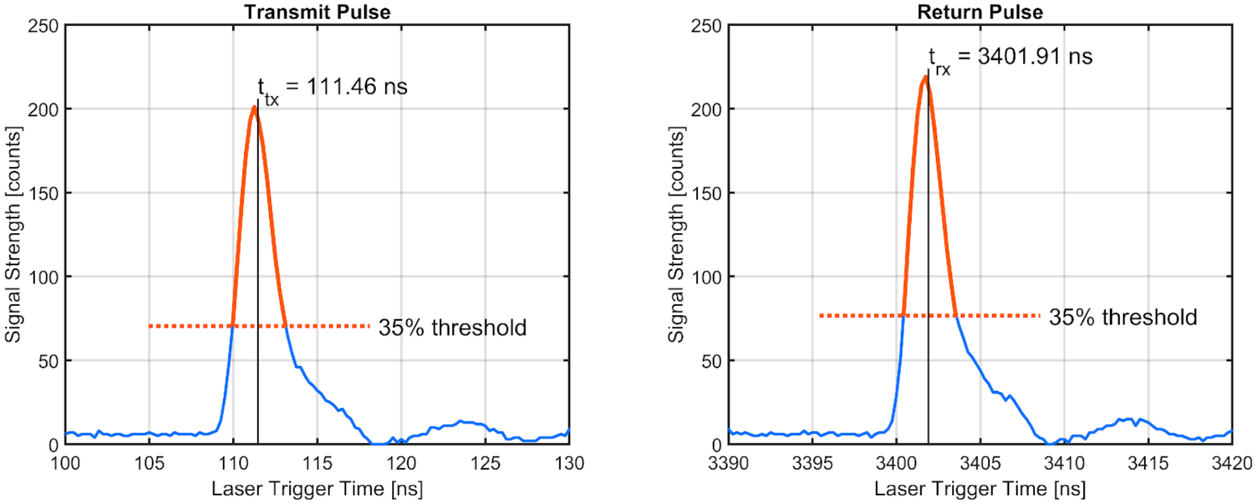

Figure 2: *Example of range determination using a centroid estimate that cuts off data points below 35% of the maximum amplitude of transmit and return pulse.*

A more complex task is the assignment of a sample time to a waveform. Figure 2 shows the two windows and the current ATM method for calculating the range measurement between the two windows. The ATM range determination is using a centroid estimate that cuts off all values below 35% of the maximum amplitude of the transmit or return pulse. The uncalibrated range between two positions in the TX and RX windows, where *c* is the speed of the propagation of light through the atmosphere is:

$$range_{uncalibrated}\,[m] = \tfrac{1}{2} \cdot c \cdot (t_{rx} - t_{tx})$$

The example shown in Figure 2 yields an uncalibrated range of 493.22 meters. As previously described the optical delay fiber and other system components introduce a range bias that is determined in ground tests by shooting the laser at a target at a known distance (Figure 3).

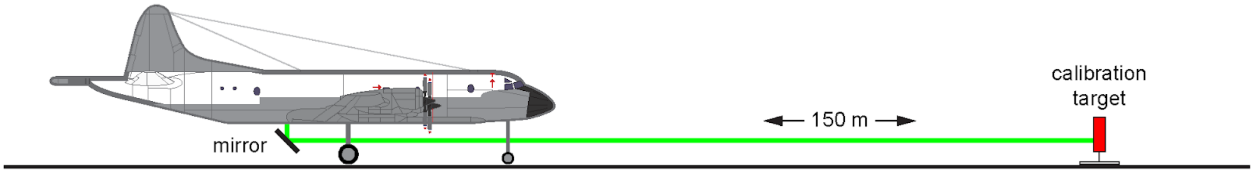

Figure 3: *Range bias determination (a.k.a. ground test) using a calibration target with a known distance. The distance to the calibration target is measured with an electronic distance meter (a.k.a. "total station") with an accuracy of a few mm.*

The range bias is a function of the return signal strength. For the ATM centroid tracker, the bias is almost constant for typical signal strengths encountered in flight, but deviates for weak signals that barely exceed the amplitude trigger threshold and for strong signals that saturate the digitizer. The range bias calibration determined from ground tests depends on the waveform tracking method. Users that wish to determine ranges using a different tracking method will need to develop a suitable calibration from the ground test data.

8) File Structure and Organization

The ATM waveform product follows the HDF convention of organizing data within groups and subgroups. Some groups contain information for each laser pulse:

/aircraft contains the aircraft's location and attitude interpolated to the times of the laser shots;
/ancillary_data contains the spatial and temporal limits for the file etc.;
/footprint contains the location of the ATM laser spots on the ground;
/laser contains pointing and range information for the laser; and
/time contains the UTC time for each laser pulse.

Other groups contain documentation and parameters necessary to use the data in a particular subgroup, or information related to the file as a whole. For example:

/mounting_parameters contains information used for computing the footprint location from the laser and aircraft information.
/ancillary_data/time/epoch_seconds_of_day would be used to convert /time/seconds_of_day into absolute time tags.

Level 1B ATM Geolocated Spot Elevation Measurements

The primary science data product are the geolocated and filtered spot elevation measurements in the group /footprint. These elevations correspond to the separate Level 1B ILATM1B data product that is available at NSIDC (version 2 in HDF format, or the older version 1 of ILATM1B and BLATM1B which is also known generically as "QFIT"). Data are organized in chronological order. For example: denoting the number of laser shots contained in the file as *N*, the timestamps of the laser shots would be an array of length *N* contained in /time/seconds_of_day, and the corresponding measured elevations would be an array of length *N* contained in /footprint/elevation.

Waveform Data

Waveform data for surveys beginning with Antarctica 2014 are stored in the subgroup /waveforms/twv. As mentioned previously (Section 6) each laser shot can be associated with a varying number of range gates that can also vary in length (number of digitizer samples) (Figure 5).

Elements in subgroup ./ancillary_data are single values. Subgroup ./shot contains arrays of *N* values (*e.g.,* ./number) corresponding to each laser shot. Arrays in subgroup ./gate correspond to each range gate. The largest subgroup (./wvfm) contains the 8-bit digitizer samples in the field ./amplitude as a concatenation of all the waveform gates recorded in the file.

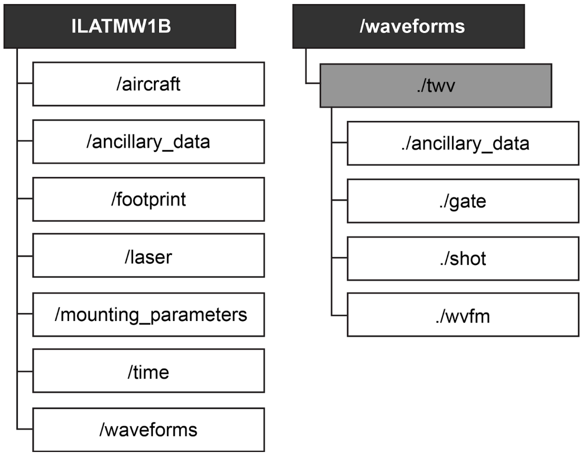

***Figure 4:** ATM HDF5 top-level group structure.*

The link between laser shots, range gates, and digitizer samples/range bins is implemented by a 1-based pointer/index scheme. The waveform data for a particular laser shot is found by locating the gates recorded for the laser shot, then locating the waveform samples associated with those gates. Consider a laser shot *j, (1≤j≤N)*. Using the conventions that ***a****(j)* is the *j*-th element of array ***a*** and ***a****(i:j)* is the sub-array taken from elements *i* through *j*, the time of the shot is given by `./shot/seconds_of_day(j).`

The number of gates associated with this laser shot is `./shot/gate_count(j).` Data for the first gate associated with this shot is located in the gate arrays at index `./shot/gate_start`. The first waveform sample for this first gate of this laser shot is located at `./shot(gate_start(j)).`

The value of this first waveform sample is `./wvfm/amplitude(./gate/wvfm_start (./shot/gate_start(j))).` The number of samples in each gate is stored in the array ./`gate/wvfm_length`. Therefore, the waveform recorded in the first gate of shot j can be read as:

```
./wvfm/amplitude(k2:k3)
where k=./shot/gate_start(j)
k2=./gate/wvfm_start(k)
k3=./gate/wvfm_start(k)+./gate/wvfm_length(k)-1
```

More generally, the digitized waveform for range gate *i* of the laser shot *j* would be:

```
./wvfm/amplitude(k2:k3)
Where k=./shot/gate_start(j)
k2=./gate/wvfm_start(k+i-1)
k3=./gate/wvfm_start(k+i-1)+./gate/wvfm_length(k+i-1)-1
```

In order to reassemble all range gates into a time tagged series the offset for the first range bin/sample needs to be known. This information is stored in the field `./gate/position` as number of digitizer samples since the laser was triggered. Together with the length of each range bin/sample in nanoseconds (`./ancillary_data/sample_interval = 0.25 ns` for 4 Giga samples per second digitization rate) the time in nanoseconds can be calculated using `./gate/position*sample_interval`. In this way, the range gates for a laser shot can be reassembled in order to determine the time of flight between transmit and receive pulse as shown in Figure 1 and Figure 2. Figure 5 illustrates the indexing scheme using values from an example data file. If the first range gate within a file starts at index 1 (`./gate/wvfm_start(1) = 1`) and is 192 range bins or samples long (`./gate/wvfm_length(1) = 192`) the second range gate will begin at index 193 (`./gate/wvfm_start(2) = 193`).

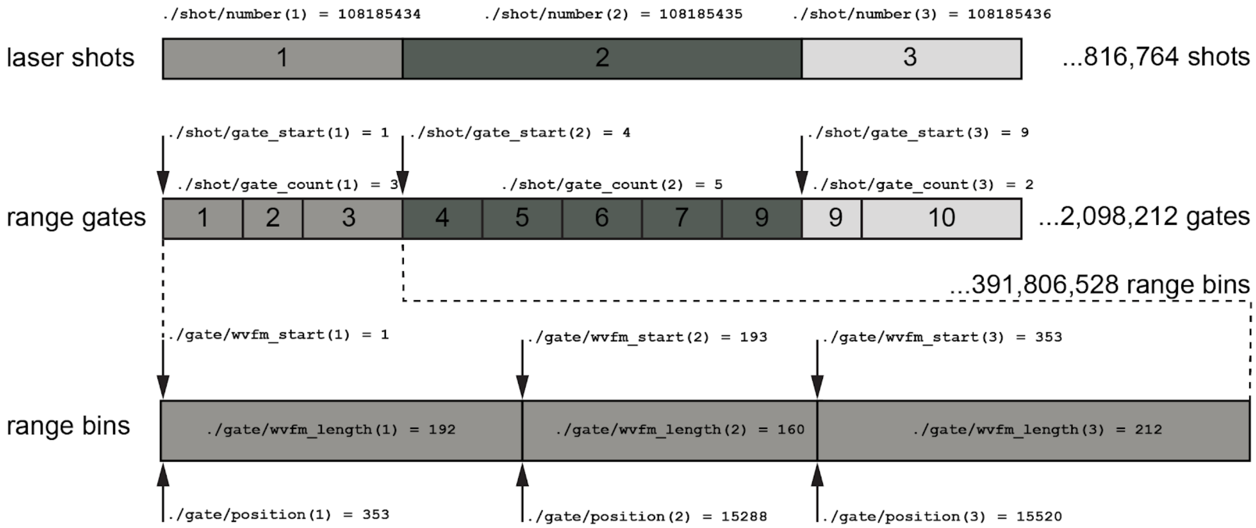
./shot/number(1) = 108185434
./shot/number(2) = 108185435
./shot/number(3) = 108185436
laser shots
1
2
3
...816,764 shots
./shot/gate_start(1) = 1
./shot/gate_start(2) = 4
./shot/gate_start(3) = 9
./shot/gate_count(1) = 3
./shot/gate_count(2) = 5
./shot/gate_count(3) = 2
range gates
1
2
3
4
5
6
7
9
9
10
...2,098,212 gates
...391,806,528 range bins
./gate/wvfm_start(1) = 1
./gate/wvfm_start(2) = 193
./gate/wvfm_start(3) = 353
range bins
./gate/wvfm_length(1) = 192
./gate/wvfm_length(2) = 160
./gate/wvfm_length(3) = 212
./gate/position(1) = 353
./gate/position(2) = 15288
./gate/position(3) = 15520

Figure 5: *Pointer and indexing schema for access of the waveform data (range gates) for a particular laser shot. Range gate numbers are also referred to as record numbers.*

The example file contains 816,764 individual laser shots whose unique shot identifiers are stored in the field `./shot/number`. The start index for the first range gate for each shot (`./shot/gate_start`) and the number of range gates for each shot (`./shot/gate_count`) have the same size as `./shot/numbers`. Together the 816,764 individual laser shots contain a total of 2,098,212 range gates, that are comprised of 391,806,528 digitizer samples.

Useful Parameters for Waveform Analysis

To aid the user working with waveforms the range gates that contain the transmit and receive pulses that were used for range determination are included for each laser shot. The field `/laser/gate_xmt` contains the 1-based gate number of the transmit pulse and `/laser/gate_rcv` contains the gate number of the primary receive pulse that was used for range determination. For waveform plotting and subselection the field `/waveforms/twv/ancillary_data/tx_start` and `./tx_end` contain the starting and ending bounds for the transmit pulse window in units of digitizer samples. In order to select only receive waveforms, range gates with times greater than `/waveforms/twv/ancillary_data/rx_start` can be selected.

Waveform Signal Quality and Complex Return Pulses

Several parameters, provided in the subgroup `/waveforms/twv/gate/pulse`, allow users to assess quality and complexity of waveforms:

Field	**Description**
`./area`	area of waveform pulse above noise floor
`./count`	number of pulses in gate (number of threshold crossings divided by 2)
`./sat count`	number of waveform amplitudes at saturation value
`./width`	width of pulse (number of samples) based on a threshold of 35% of peak

Complex returns: some return signals contain complex waveforms. Complex returns showing multiple peaks or a broadened pulse indicate interaction of the laser pulse with complex surfaces such as sea-ice pressure ridges, crevasses or melt ponds (Figure 6).

To identify these scientifically interesting waveforms several parameters can be used that indicate a deviation from a single return on a relatively smooth ice surface. The `./width` field indicates pulse broadening from interacting with a complex target or a steep surface within the laser footprint (Figure 7b). Only data points above the threshold (35% of maximum amplitude) are used.

The `./area` parameter also indicates pulse broadening but takes into account all data points above the noise floor. This allows identification of changes in the waveform in the tail of a laser pulse.The `./count` parameter indicates multiple targets for a single transmit pulse such as returns from the water surface and bottom of shallow melt ponds or multiple targets over extremely rough surfaces such as pressure ridges (Figure 7a).

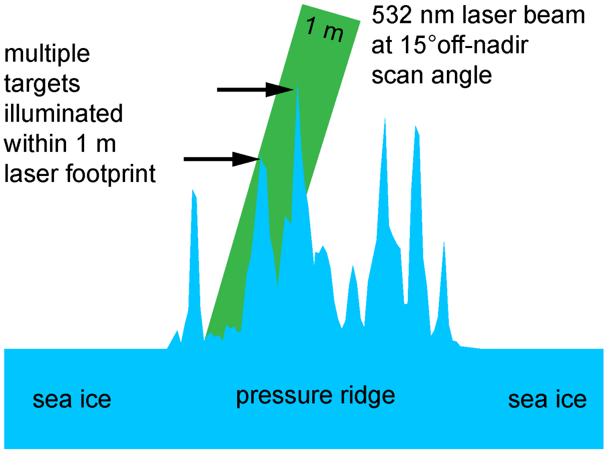
1 m
532 nm laser beam at 15°off-nadir scan angle
multiple targets illuminated within 1 m laser footprint
sea ice
pressure ridge
sea ice

***Figure 6:** Example of surface characteristics that can cause complex return waveforms, when the laser pulse hits multiple targets within the size of the footprint.*

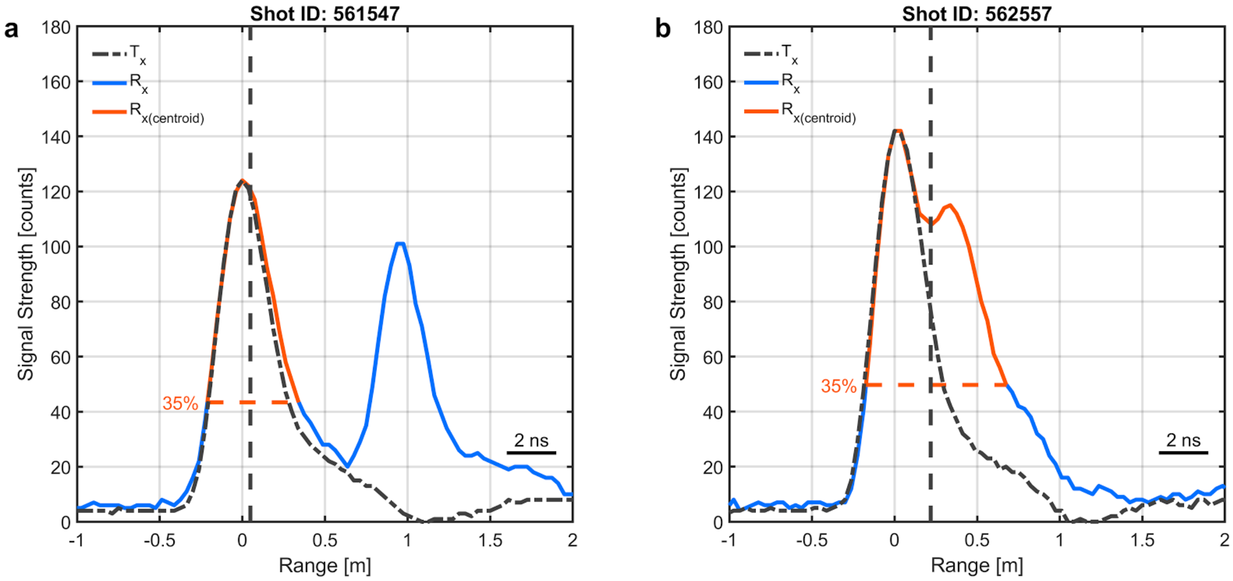

Figure 7: *Example of complex return waveforms (blue) over a sea-ice pressure ridge. The transmit waveform is shown in dashed gray and scaled to the maximum amplitude of the return pulse to show the deviation of the return pulse. Multiple separate return pulses can be identified by the `./count` parameter (a), while return pulses much broader than the transmit pulse (b) will be revealed by larger values in the `./width` field. The vertical dashed line in both panels marks the location of the centroid indicating the centroid estimates from complex return pulses need to be interpreted properly.*

9) Identifying corresponding 532 and 1064 nm waveforms

The green and NIR files are organized by time, each one containing data acquired during one minute. Files start and end on the minute boundaries. File names contain the starting time of each file. The green and IR data are in separate files, with a one-to-one correspondence between files, and between the laser shots contained in the files. The NIR files have don't have the `/footprint` directory containing the latitude, longitude, and elevation fields.

10) Test file for code development

For users who want to develop their own ingest code a test file with diagnostic amplitude values is available to help with code verification and testing. The test file consists of 20 records. The amplitude values in `/waveforms/twv/wvfm_amplitude` are set to specific values to help users verify that their code is reading the intended record and gate. E.g., all samples in record 1, gate 1 are set to 11. Record 1, gate 2 contains all 12's; record 2, gate 1 contains all 21's; etc. Because the maximum allowed number in `/waveforms/twv/wvfm_amplitude` is 255 for an 8 bit digitizer the file is limited to 20 records. The gates in the test file have varying lengths to allow for testing of the 1-based pointer/index scheme.

References

Lee, J., Ice, Cloud, and Land Elevation Satellite (ICESat-2) Project, Advanced Topographic Laser Altimeter System (ATLAS), Science Investigator-led Processing System (SIPS), ATLAS Science Algorithm Software, Standard Data Product (SDP), Volume 3: ATL03. 2018.

Martin, C. F., W. B. Krabill, S. S. Manizade, R. L. Russell, J. G. Sonntag, R. N. Swift, and J. K. Yungel. 2012. Airborne Topographic Mapper Calibration Procedures and Accuracy Assessment. Greenbelt, Md.: National Aeronautics and Space Administration, Goddard Space Flight Center. NASA Technical Report 2012-215891.

Software Tools for Reading ATM HDF5 Data

HDF5 is a stable data format with broad support in the scientific community. There are many free software tools available such as HDFView for browsing, editing and viewing HDF5 files. Common programming languages such as MATLAB® and IDL include native HDF5 support. Open source software libraries for reading and writing HDF5 files are freely available for C, C++, Fortran and Python™.

This section contains documentation and user guides for software tools developed by members of the ATM team for reading and viewing ATM HDF5 waveform data. The source code and documentation are available from the ATM website at https://atm.wff.nasa.gov/.

9) MATLAB® functions

Several MATLAB® functions are provided that can be used to import, select and plot waveform data and range gates based on search criteria.

9.1 ATM_WVFM_READER

The main MATLAB® function that provides basic capabilities for using ATM HDF5 waveform files within MATLAB® is `atm_wvfm_reader`. The `atm_wvfm_reader` function reads ATM waveform and QFIT data from an ATM HDF5 file and imports the data into a structured array (struct) in MATLAB®. The function allows to extract only waveforms and qfit elevations that satisfy spatial and temporal search criteria, i.e. it can be used for subsetting. Typing `help atm_wvfm_reader` in the MATLAB® command window shows the 5 different syntax options and includes an example how to call the function from within MATLAB®. In its simplest form the function imports an entire HDF5 file (skipping many groups and subgroups):

```
>> atm_wvfm = atm_wvfm_reader('ILATM1B_20170510_132857.atm6AT6.h5');
```

Possible function calls are:

```
>> atm_wvfm = atm_wvfm_reader(f_name_inp);
>> atm_wvfm = atm_wvfm_reader(f_name_inp,verbose);
>> atm_wvfm = atm_wvfm_reader(f_name_inp,verbose,poly_lon,poly_lat);
>> atm_wvfm = atm_wvfm_reader(f_name_inp,verbose,t_start,t_end);
>> atm_wvfm = atm_wvfm_reader(f_name_inp,verbose,poly_lon,poly_lat, t_start,
t_end);
```

The m-file of this function must be either in your search path or data directory or the folder needs to be added temporarily to MATLAB's search path using the MATLAB addpath() function. If the commands are not run from the folder that contains the HDF5 data file the filename must include the full path.

The output of `atm_wvfm_reader` is a MATLAB® structure array or "struct" consisting of data containers referred to as "fields". Structs are commonly used for storing heterogeneous data. The code can easily be modified to output a cell array instead of a struct. The downside of both, structs and cell arrays is that the contents of just a subset of an ATM HDF5 file will occupy more than an order of magnitude more memory than the size of the compressed HDF5 file. Saving the output struct into a MATLAB® file using MATLAB®'s data compression will result in significantly larger file sizes than the HDF5 input files and is not recommended. Figure 8 shows the basic structure of the output array.

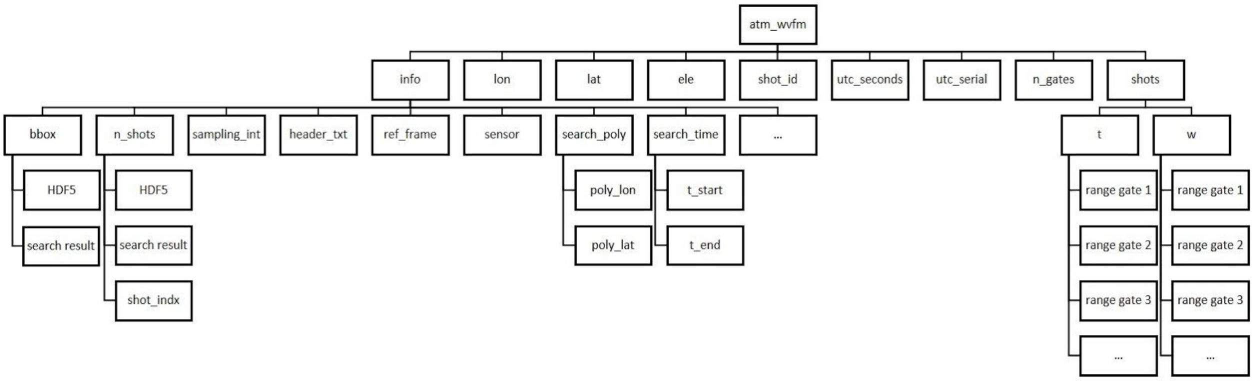

Figure 8: *Contents of output MATLAB® struct. The info field contains metadata about the input file, the search parameters used and the output results. The waveforms are stored in the field "shots" and contain all individual range gates for each shot that has satisfied the spatial and temporal search criteria.*

This basic example shows how to plot the data from all range gates in shot/record #5 in a struct named "atm_wvfm" using the laser trigger time in nanoseconds as the abscissa:

```
>> shot_nr = 5;
>> figure;
>> for i = 1:atm_wvfm.n_gates(shot_nr)
>>
plot(atm_wvfm.shots(shot_nr).wf(i).t,atm_wfm.shots(shot_nr).wf(i).w,'-');
>>    hold on;
>> end
```

9.2 ATM_INFO

The MATLAB® function `atm_wvfm_info` can be used to quickly gain information about the contents of an ATM HDF5 waveform file. It takes advantage of HDF5's capability to quickly access only parts of a file. The function returns a polygon with the minimum and maximum longitude and latitude of all laser shots in the file as well as the UTC time for the first and last laser shot. Its primary purpose is to quickly identify HDF5 files for close examination. The function can be called with the following syntax:

```
>> [poly_lon,poly_lat,t_start,t_end,data_fmt] = …
>> ATM_WVFM_INFO(f_name_inp,verbose);
```

Typing `help atm_wvfm_info` in the MATLAB® Command Window provides an overview of its functionality.

9.3 PLOT_WVFMS

Also included is a MATLAB® function called `plot_wvfms` that can be used to plot a small number of select waveforms and range gates. The function allows to select a list of shot numbers/indices or a time window, or a combination of both. The time window feature is useful for plotting transmit pulses, window reflections or return pulses. Possible function calls are:

```
>> plt_wvfms(wvfm_struct,shot_list);
>> plt_wvfms(wvfm_struct,shot_list,t_min);
>> plt_wvfms(wvfm_struct,shot_list,t_min,t_max);
>> plt_wvfms(wvfm_struct,shot_list,t_min,t_max,back_ground_color);
```

`wvfm_struct` is a MATLAB® structure array or "struct" created by `atm_wvfm_reader`. `shot_list` must be a row or column vector of shot indices to be plotted. `t_min` and `t_max` are laser trigger times in nanoseconds. `plt_wvfms` plots only range gates with `t >= t_min` and `t <= t_max`. `t_max` can be left unset or set to `Inf` to include all range gates after `t_min`. Typing `help plt_wvfms` at the MATLAB® prompt provides general guidance on how to use the function and some tips on how to select transmit pulse, window reflections and return pulses for plotting. The time windows to use for the particular data file are campaign dependent and are listed in the HDF5 fields `/waveforms/twv/ancillary_data/rx_start`, `./tx_start` and `./tx_end` in number of digitizer samples.

Figure 9 shows an example plot of select waveforms over a sea ice pressure ridge north of Ellesmere Island. Here are the commands from the MATLAB® script `make_example_plot.m` that will create the figure:

```
f_name_inp = 'ILATM1B_20170309_133117.atm6AT6.h5';
verbose = 1;
[atm] = atm_wvfm_reader(f_name_inp, verbose);

shot_list = [548141 549177 549180 549190 549195 549695 561547 562556 562557
562561 563089 563093]; % complex returns over pressure ridge
plt_wvfms(atm,shot_list,200,Inf,'w');
```

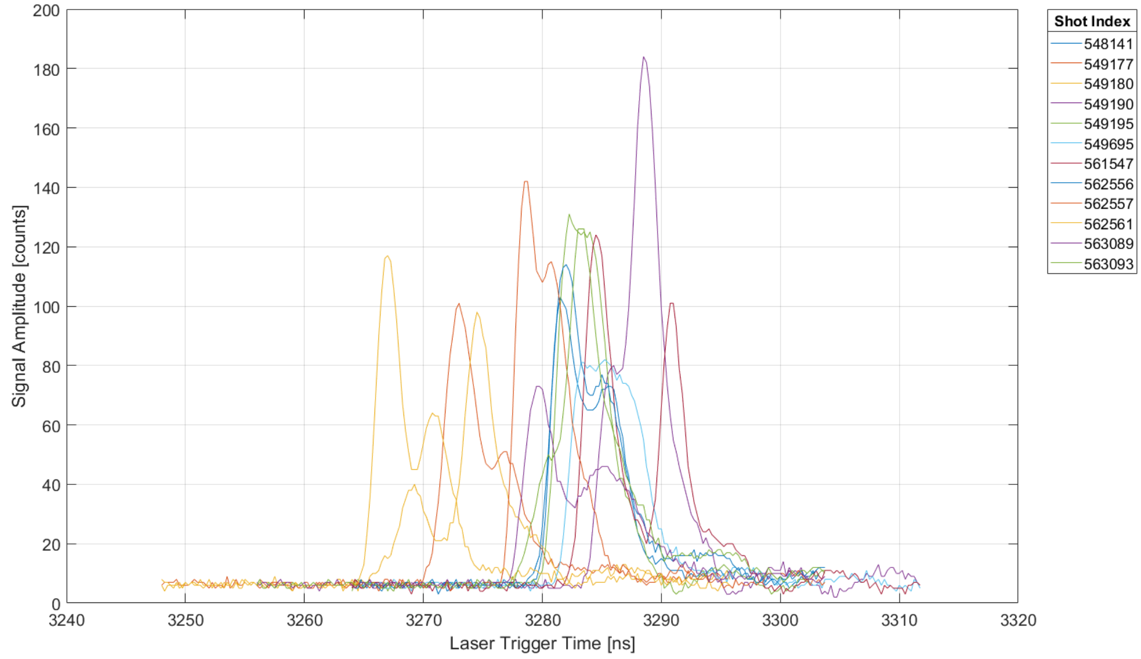

Figure 9: *Example plot showing complex waveforms over a pressure ridge north of Ellesmere Island produced with the above MATLAB commands.*

10) IDL functions

A simple example is given to demonstrate the IDL calls for opening and reading components of the HDF structure.

10.1 idl_read_ATMhdf.pro

```
; example IDL code for reading an ATM data file in HDF5 format
;
; reads the footprint data using two different methods.
; Three data arrays are extracted:
;  /footprint/latitude
;  /footprint/longitude
;  /footprint/elevation
;
; author: Serdar Manizade
; Date: 2017-Sep-07

; name of an input file (this one is a test file containing sea ice data)
file='ILNSAW1B_20171031_191346.atm6CT7.h5'
t0=systime(/seconds) ; measure how long each method requires

;===============================================
; METHOD 1: read the entire HDF structure at once
;===============================================
print,'Reading entire file: '+file
r=h5_parse(file,/READ_DATA)
print,'---- done reading'
```

```
help,/structure,r.footprint
elev1=r.footprint.elevation._data
lat_1=r.footprint.latitude._data
lon_1=r.footprint.longitude._data

t1=systime(/seconds) ; measure how long each method requires

;===============================================
; METHOD 2: open file, and read only the desired data elements
;===============================================
; open the HDF5 file
print,'Reading select fields from file: '+file
file_id = H5F_OPEN(file)
dataset_id1 = H5D_OPEN(file_id,'/footprint/elevation')
help,dataset_id1
elev2=h5d_read(dataset_id1)
lat_2=h5d_read(h5d_open(file_id,'/footprint/latitude'))
lon_2=h5d_read(h5d_open(file_id,'/footprint/longitude'))

t2=systime(/seconds) ; measure how long each method requires

;===============================================
; show information about the data variables read
help,elev1,elev2,lat_1,lat_2,lon_1,lon_2

print,'elapsed time (secs) for method 1:',t1-t0
print,'elapsed time (secs) for method 2:',t2-t1

; create two plots of the data
plot,lon_1,lat_1,psym=3,xtitle='Longitude (deg)',ytitle='Latitude (deg)'
window,/free
plot,lon_1,elev1,psym=3,xtitle='Longitude (deg)',ytitle='Elevation (m)'
;===============================================
```

The sample program prints the elapsed time for the two methods of reading the file. This example, which included full waveforms, produced the output:

```
elapsed time (secs) for method 1:    13.553
elapsed time (secs) for method 2:     0.069
```