

1 **Ice, Cloud, and Land Elevation Satellite 2 (ICESat-2)**

2
3 **Algorithm Theoretical Basis Document (ATBD)**

4
5 **for**

6
7 **Land - Vegetation Along-Track Products (ATL08)**

8
9
10
11 **Contributions by Land/Vegetation SDT Team Members**
12 **and ICESat-2 Project Science Office**

13 **(Amy Neuenschwander, Sorin Popescu, Ross Nelson, David Harding,**
14 **Katherine Pitts, John Robbins, Dylan Pederson, and Ryan Sheridan)**

15
16
17 **ATBD prepared by**

18 **Amy Neuenschwander and**

19 **Katherine Pitts**

20
21 **15 September 2019**

22
23
24 **Content reviewed: technical approach, assumptions, scientific soundness,**
25 **maturity, scientific utility of the data product**

28 **ATL08 algorithm and product change history**

29

ATBD Version	Change
2016 Nov	Product segment size changed from 250 signal photons to 100 m using five 20m segments from ATL03 (Sec 2)
2016 Nov	Filtered signal classification flag removed from classed_pc_flag (Sec 2.3.2)
2016 Nov	DRAGANN signal flag added (Sec 2.3.4)
2016 Nov	Do not report segment statistics if too few ground photons within segment (Sec 4.15 (3))
2016 Nov	Product parameters added: h_canopy_uncertainty, landsat_flag, d_flag, delta_time_beg, delta_time_end, night_flag, msw_flag (Sec 2)
2017 May	Revised region boundaries to be separated by continent (Sec 2)
2017 May	Alternative DRAGANN parameter calculation added (Sec 1.1.1)
2017 May	Set canopy flag = 0 when <i>L-km</i> segment is over Antarctica or Greenland regions (Sec 4.4 (1))
2017 May	Change initial canopy filter search radius from 3 m to 15 m (Sec 4.9 (6))
2017 May	Product parameters removed: h_rel_ph, terrain_thresh
2017 May	Product parameters added: segment_id, segment_id_beg, segment_id_end, dem_flag, surf_type (Sec 2)
2017 July	Urban flag added (Sec 2.4.17)
2017 July	Dynamic point spread function added (Sec 4.11 (6))
2017 July	Methodology for processing <i>L-km</i> segments with buffer added (Sec 4.1 (2), Sec 4.17)
2017 July	Revised alternative DRAGANN methodology (see bolded text in Sec 1.1.1)
2017 July	Added post-DRAGANN filtering methodology (Sec 4.7)
2017 July	Updated SNR to be estimated from superset of ATL03 and DRAGANN found signal used for processing ATL08 (Sec 2.5.17)
2017 September	More details added to DRAGANN description (Sec 4.3), and corrections to DRAGANN implementation (Sec 3.1.1, Sec 4.3 (9))
2017 September	Added Appendix A – very detailed DRAGANN description
2017 September	Revised alternative DRAGANN methodology (see bolded text in Sec 1.1.1)
2017 September	Clarified SNR calculation (Sec 2.5.17, Sec 4.3 (18))
2017 September	Added cloud flag filtering option (Sec Error! Reference source not found.)

**ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002**

2017 September	Added top of canopy median surface filter (Sec 3.5 (a), Sec 4.10 (3), Sec 4.12 (1-3))
2017 September	Modified 500 canopy photon segment filter (Sec 3.5 (c), Sec 4.12 (6))
2017 November	Added solar_azimuth, solar_elevation, and n_seg_ph to Reference Data group; parameters were already in product (Sec 2.4)
2017 November	Specified number of ground photons threshold for relative canopy product calculations (Sec 4.16 (2)); no number of ground photons threshold for absolute canopy heights (Sec 4.16.1 (1))
2017 November	Changed the ATL03 signal used in superset from all ATL03 signal (signal_conf_ph flags 1-4) to the medium-high confidence flags (signal_conf_ph flags 3-4) (Sec 3.1, Sec 4.3 (17))
2017 November	Removed Date parameter from Table 2.4 since UTC date is in file metadata
2018 March	Clarified that cloud flag filtering option should be turned off by default (Sec Error! Reference source not found.)
2018 March	Changed h_diff_ref QA threshold from 10 m to 25 m (Table 5.2)
2018 March	Added absolute canopy height quartiles, canopy_h_quartile_abs (<i>Later removed</i>)
2018 March	Removed psf_flag from main product; psf_flag will only be a QAQC alert (Sec 5.2)
2018 March	Added an Asmooth filter based on the reference DEM value (Sec 4.6 (4-5))
2018 March	Changed relief calculation to 95 th – 5 th signal photon heights. (Sec 4.6 (6))
2018 March	Adjusted the Asmooth smoothing methodology (Sec 4.6 (8))
2018 March	Recalculate the Asmooth surface after filtering outlying noise from signal, then detrend signal height data (Sec 4.7 (3-4))
2018 March	Added option to run alternative DRAGANN process again in high noise cases (Sec 4.3.2)
2018 March	Changed global land cover reference to MODIS Global Mosaics product (Sec 2.4.14)
2018 March	Adjusted the top of canopy median filter thresholds based on SNR (Sec 4.12 (1-2))
2018 March	Added a final photon classification QA check (Sec 4.14, Table 5.2)
2018 March	Added slope adjusted terrain parameters (<i>Later removed</i>)
2018 June	Replaced slope adjusted terrain parameters with terrain best fit parameter (Sec 2.1.14, 4.15 (2.e))

**ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002**

2018 June	Clarified source for water mask (Sec 2.4.15)
2018 June	Clarified source for urban mask (Sec 2.4.17)
2018 June	Added expansion to the terrain_slope calculation (Sec 4.15)
2018 June	Removed canopy_d_quartile
2018 June	Removed canopy_quartile_heights and canopy_quartile_heights_abs, replaced with canopy_h_metrics (Secs 2.2.3, 4.16 (6), 4.16.1 (5))
2018 *** draft 1	Delta_time specified as mid-segment time, rather than mean segment time (Sec 2.4.5)
2018 *** draft 1	QA/QC products to be reported on a per orbit basis, rather than per region (Sec 5.2)
2018 *** draft 1	Added more detail to landsat_flag description (Sec 2.2.23)
2018 *** draft 1	Added psf_flag back into ATL08 product, as it is also needed for the QA product (Sec 2.5.12)
2018 *** draft 1	Specified that the sigma_h value reported here is the mean of the ATL03 reported sigma_h values (Sec 2.5.7)
2018 *** draft 1	Removed n_photons from all subgroups
2018 *** draft 1	<p>Better defined the interpolation and smoothing methods used throughout:</p> <ul style="list-style-type: none"> • Error! Reference source not found. (3): Interpolation – nearest • 4.6 (5): Interpolation – PCHIP • 4.6 (8): Smoothing – moving average • 4.7 (3): Interpolation – PCHIP • 4.7 (3): Smoothing – moving average • 4.8 (3): Smoothing – moving average • 4.8 (4): Interpolation – linear • 4.8 (5): Smoothing – moving average • 4.8 (6): Interpolation – linear • 4.8 (7): Smoothing – moving average • 4.8 (8): Smoothing – Savitzky-Golay • 4.8 (9): Interpolation – linear • 4.8 (14): Interpolation – PCHIP • 4.10 (10): Interpolation – linear • 4.11 (all): Smoothing – moving average • 4.10 (6.b): Interpolation – linear • 4.12 (1.a): Interpolation – linear • 4.12 (1.c): Smoothing – lowess • 4.12 (4): Interpolation – PCHIP • 4.12 (7): Interpolation – PCHIP • 4.12 (9): Smoothing – moving average • 4.15 (2.e.i.1): Interpolation – linear

**ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002**

2018 *** draft 1	Added ref_elev and ref_azimuth back in (it was mistakenly removed in a previous version; Secs 2.5.3, 2.5.4)
2018 *** draft 1	Clarified wording of h_canopy_quad definition (Sec 2.2.17)
2018 *** draft 1	Updated segment_snowcover description to match the ATL09 snow_ice parameter it references (Sec 2.4.16) and added product reference to Table 4.2
2018 *** draft 1	Added ph_ndx_beg (Sec 2.5.21); parameter was already on product
2018 *** draft 1	Added dem_removal_flag for QA purposes (Sec 2.4.11; Table 5.2)
2018 *** draft 2	Reformatted QA/QC trending and trigger alert list into a table for better clarification (Table 5.3)
2018 *** draft 2	Replaced n_photons in Table 5.2 with n_te_photons, n_ca_photons, and n_toc_photons
2018 *** draft 2	Removed beam_number from Table 2.5. Beam number and weak/strong designation within gtx group attributes.
2018 *** draft 2	Clarified calculation of h_te_best_fit (Sec 4.15 (2.e))
2018 *** draft 2	Changed h_canopy and h_canopy_abs to be 98 th percentile height (Table 2.2, Sec 2.2.5, Sec 2.2.6, Sec 4.16 (4), Sec 4.16.1 (3))
2018 *** draft 2	Separated h_canopy_metrics_abs from h_canopy_metrics (Table 2.2, Sec 2.2.3, Sec 4.16.1 (5))
2018 October	Removed 99 th percentile from h_canopy_metrics and h_canopy_metrics_abs (Table 2.2, Sec 2.2.3, Sec 2.2.4, Sec 4.16 (4), Sec 4.16.1 (5))
2018 December	Renamed and reworded Section 1.1.1 to better indicate that the DRAGANN preprocessing step is not optional
2018 December	Specified that DRAGANN should use along-track time, and added time rescaling step (Sec 4.3 (1 - 4))
2018 December	Added DRAGANN changes made to better capture sparse canopy in cases of low noise rates (Sec 4.3, Appendix A)
2018 December	Made corrections to DRAGANN description regarding the determination of the noise Gaussian (Sec 3.1.1, Sec 4.3)
2018 December	Removed h_median_canopy and h_median_canopy_abs, as they are equivalent to canopy_h_metrics(50) and canopy_h_metrics_abs(50) (Table 2.2, Sec 4.16 (5), Sec 4.16.1 (4))
2018 December	Removed the requirement that > 5% ground photons required to calculate relative canopy height parameters (Table 2.2, Sec 4.16 (2))
2018 December	Added canopy relative height confidence flag (canopy_rh_conf) based on the percentage of ground and canopy photons in a segment (Table 2.2, Sec 4.16 (2))

**ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002**

2018 December	Added ATL09 layer_flag to ATL08 output (Table 2.5, Table 4.2)
2019 February	Adjusted cloud filtering to be based on ATL09 backscatter analysis rather than cloud flags (Sec 4.1)
2019 March 5	Updated ATL09-based product descriptions reported on ATL08 product (Secs 2.5.13, 2.5.14, 2.5.15, 1.1.1)
2019 March 5	Updated cloud-based low signal filter methodology, and moved to first step of ATL08 processing (Sec 4.1)
2019 March 13	Replace canopy_closure with new landsat_perc parameter (Table 2.2, Sec 2.2.24)
2019 March 13	Change ATL08 product output regions to match ATL03 regions (Sec 2), but keep ATL08 regions internally and report in new parameter atl08_regions (Table 2.4, Sec 2.4.19)
2019 March 13	Add methodology for handling short ATL08 processing segments at the end of an ATL03 granule (Sec 4.2), and output distance the processing segment length is extended into new parameter last_seg_extend (Table 2.4, Sec 2.4.20)
2019 March 13	Add preprocessing step for removing atmospheric and ocean tide corrections from ATL03 heights (<i>Later removed</i>)
2019 March 27	Remove preprocessing step for removing atmospheric and ocean tide corrections from ATL03 heights, since those values are now removed from the ATL03 photon heights.
2019 March 27	Replaced ATL03 region figure with corrected version (Figure 2.2)
2019 March 27	Specified that at least 50 classed photons are required to create the 100 m land and canopy products (Secs 2, 4.15(1), 4.16(1))
2019 March 27	Clarified that any non-extended segments would report a land_seg_extend value of 0 (Sec 4.2, Sec 2.4.20)
2019 April 30	Fixed the error in Eqn 1.4 for the sigma_topo value
2019 May 13	Specified for cloud flag carry-over from ATL09 that ATL08 will report the highest cloud flag if an 08 segment straddles two 09 segments. (Section 2.5)
2019 May 13	Changed parameter cloud_flag_asr to cloud_flag_atm since the cloud_flag_asr is likely not to work over land due to varying surface reflectance (Sec, 2.5)
2019 May 13	Add ATL09 parameter cloud_fold_flag to the ATL08 data product for future qa/qc checks for low clouds. (Secs, 2.5)
2019 May 13	Clarification on the calculation of gradient for slope that feeds into the calculation of the point spread function (Sec 4.11)

*ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002*

2019 July 1	Changed Landsat canopy cover percentage to 3 % (from original value of 5%)
--------------------	--

30
31

32 **Contents**

33	List of Tables.....	14
34	List of Figures.....	15
35	1 INTRODUCTION.....	17
36	1.1. Background.....	18
37	1.2. Photon Counting Lidar	20
38	1.3. The ICESat-2 concept.....	22
39	1.4. Height Retrieval from ATLAS.....	25
40	1.5. Accuracy Expected from ATLAS.....	26
41	1.6. Additional Potential Height Errors from ATLAS.....	28
42	1.7. Dense Canopy Cases	29
43	1.8. Sparse Canopy Cases	29
44	2. ATL08: DATA PRODUCT.....	31
45	2.1. Subgroup: Land Parameters	34
46	2.1.1. Georeferenced_segment_number_beg.....	35
47	2.1.2. Georeferenced_segment_number_end.....	35
48	2.1.3. Segment_terrain_height_mean	36
49	2.1.4. Segment_terrain_height_med.....	36
50	2.1.5. Segment_terrain_height_min.....	36
51	2.1.6. Segment_terrain_height_max.....	37
52	2.1.7. Segment_terrain_height_mode.....	37
53	2.1.8. Segment_terrain_height_skew.....	37
54	2.1.9. Segment_number_terrain_photons	37
55	2.1.10. Segment height_interp	38
56	2.1.11. Segment h_te_std.....	38

**ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002**

57	2.1.12	Segment_terrain_height_uncertainty	38
58	2.1.13	Segment_terrain_slope	38
59	2.1.14	Segment_terrain_height_best_fit	39
60	2.2	Subgroup: Vegetation Parameters	39
61	2.2.1	Georeferenced_segment_number_beg	42
62	2.2.2	Georeferenced_segment_number_end	42
63	2.2.3	Canopy_height_metrics_abs	42
64	2.2.4	Canopy_height_metrics	43
65	2.2.5	Absolute_segment_canopy_height	43
66	2.2.6	Segment_canopy_height	43
67	2.2.7	Absolute_segment_mean_canopy	44
68	2.2.8	Segment_mean_canopy	44
69	2.2.9	Segment_dif_canopy	44
70	2.2.10	Absolute_segment_min_canopy	44
71	2.2.11	Segment_min_canopy	44
72	2.2.12	Absolute_segment_max_canopy	44
73	2.2.13	Segment_max_canopy	45
74	2.2.14	Segment_canopy_height_uncertainty	45
75	2.2.15	Segment_canopy_openness	46
76	2.2.16	Segment_top_of_canopy_roughness	46
77	2.2.17	Segment_canopy_quadratic_height	46
78	2.2.18	Segment_number_canopy_photons	46
79	2.2.19	Segment_number_top_canopy_photons	47
80	2.2.20	Centroid_height	47
81	2.2.21	Segment_rel_canopy_conf	47
82	2.2.22	Canopy_flag	47

**ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002**

83	2.2.23	Landsat_flag.....	47
84	2.2.24	Landsat_perc.....	48
85	2.3	Subgroup: Photons.....	48
86	2.3.1	Indices_of_classed_photons.....	49
87	2.3.2	Photon_class.....	49
88	2.3.3	Georeferenced_segment_number.....	49
89	2.3.4	DRAGANN_flag.....	49
90	2.4	Subgroup: Reference data.....	49
91	2.4.1	Georeferenced_segment_number_beg.....	51
92	2.4.2	Georeferenced_segment_number_end.....	51
93	2.4.3	Segment_latitude.....	52
94	2.4.4	Segment_longitude.....	52
95	2.4.5	Delta_time.....	52
96	2.4.6	Delta_time_beg.....	52
97	2.4.7	Delta_time_end.....	52
98	2.4.8	Night_Flag.....	52
99	2.4.9	Segment_reference_DTM.....	53
100	2.4.10	Segment_reference_DEM_source.....	53
101	2.4.11	Segment_reference_DEM_removal_flag.....	53
102	2.4.12	Segment_terrain_difference.....	53
103	2.4.13	Segment_terrain_flag.....	53
104	2.4.14	Segment_landcover.....	53
105	2.4.15	Segment_watermask.....	54
106	2.4.16	Segment_snowcover.....	54
107	2.4.17	Urban_flag.....	54
108	2.4.18	Surface_type.....	55

**ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002**

109	2.4.19	ATL08_region.....	55
110	2.4.20	Last_segment_extend	55
111	2.5	Subgroup: Beam data.....	56
112	2.5.1	Georeferenced_segment_number_beg.....	58
113	2.5.2	Georeferenced_segment_number_end.....	58
114	2.5.3	Beam_coelevation.....	59
115	2.5.4	Beam_azimuth.....	59
116	2.5.5	ATLAS_Pointing_Angle	59
117	2.5.6	Reference_ground_track.....	59
118	2.5.7	Sigma_h.....	59
119	2.5.8	Sigma_along.....	60
120	2.5.9	Sigma_across.....	60
121	2.5.10	Sigma_topo.....	60
122	2.5.11	Sigma_ATLAS_LAND.....	60
123	2.5.12	PSF_flag.....	60
124	2.5.13	Layer_flag.....	61
125	2.5.14	Cloud_flag_atm	61
126	2.5.15	MSW	61
127	2.5.16	Cloud Fold Flag.....	62
128	2.5.17	Computed_Apparent_Surface_Reflectance	62
129	2.5.18	Signal_to_Noise_Ratio	62
130	2.5.19	Solar_Azimuth.....	62
131	2.5.20	Solar_Elevation.....	62
132	2.5.21	Number_of_segment_photons.....	62
133	2.5.22	Photon_Index_Begin	62
134	3	ALGORITHM METHODOLOGY.....	64

**ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002**

135	3.1	Noise Filtering	64
136	3.1.1	DRAGANN.....	65
137	3.2	Surface Finding	69
138	3.2.1	De-trending the Signal Photons.....	71
139	3.2.2	Canopy Determination	71
140	3.2.3	Variable Window Determination.....	73
141	3.2.4	Compute descriptive statistics	74
142	3.2.5	Ground Finding Filter (Iterative median filtering).....	76
143	3.3	Top of Canopy Finding Filter.....	78
144	3.4	Classifying the Photons	78
145	3.5	Refining the Photon Labels	79
146	3.6	Canopy Height Determination.....	83
147	3.7	Link Scale for Data products.....	83
148	4.	ALGORITHM IMPLEMENTATION.....	85
149	4.1	Cloud based filtering.....	88
150	4.2	Preparing ATL03 data for input to ATL08 algorithm.....	90
151	4.3	Noise filtering via DRAGANN	91
152	4.3.1	DRAGANN Quality Assurance	94
153	4.3.2	Preprocessing to dynamically determine a DRAGANN parameter.....	94
154	4.3.3	Iterative DRAGANN processing	97
155	4.4	Is Canopy Present	98
156	4.5	Compute Filtering Window.....	98
157	4.6	De-trend Data.....	99
158	4.7	Filter outlier noise from signal.....	100
159	4.8	Finding the initial ground estimate.....	100
160	4.9	Find the top of the canopy (if canopy_flag = 1)	102

**ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002**

161	4.10	Compute statistics on de-trended data	103
162	4.11	Refine Ground Estimates.....	104
163	4.12	Canopy Photon Filtering.....	106
164	4.13	Compute individual Canopy Heights	108
165	4.14	Final photon classification QA check.....	109
166	4.15	Compute segment parameters for the Land Products.....	109
167	4.16	Compute segment parameters for the Canopy Products.....	112
168	4.16.1	Canopy Products calculated with absolute heights.....	114
169	4.17	Record final product without buffer.....	114
170	5	DATA PRODUCT VALIDATION STRATEGY	115
171	5.1	Validation Data.....	115
172	5.2	Internal QC Monitoring	118
173	6	REFERENCES.....	125
174			

**ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002**

175 **List of Tables**

176	Table 2.1. Summary table of land parameters on ATL08.....	34
177	Table 2.2. Summary table of canopy parameters on ATL08.	40
178	Table 2.3. Summary table for photon parameters for the ATL08 product.....	48
179	Table 2.4. Summary table for reference parameters for the ATL08 product.....	50
180	Table 2.5. Summary table for beam parameters for the ATL08 product.....	56
181	Table 3.1. Standard deviation ranges utilized to qualify the spread of photons within	
182	moving window.....	75
183	Table 4.1. Input parameters to ATL08 classification algorithm.....	85
184	Table 4.2. Additional external parameters referenced in ATL08 product.	86
185	Table 5.1. Airborne lidar data vertical height (Z accuracy) requirements for	
186	validation data.	115
187	Table 5.2. ATL08 parameter monitoring.	118
188	Table 5.3. QA/QC trending and triggers.....	123
189		

190 **List of Figures**

191 Figure 1.1. Various modalities of lidar detection. Adapted from Harding, 2009..... 21

192 Figure 1.2. Schematic of 6-beam configuration for ICESat-2 mission. The laser
193 energy will be split into 3 laser beam pairs – each pair having a weak spot (1X) and a
194 strong spot (4X). 23

195 Figure 1.3. Illustration of off-nadir pointing scenarios. Over land (green regions) in
196 the mid-latitudes, ICESat-2 will be pointed away from the repeat ground tracks to
197 increase the density of measurements over terrestrial surfaces..... 24

198 Figure 1.4. Illustration of the point spread function, also referred to as Znoise, for a
199 series of photons about a surface..... 27

200 Figure 2.1. HDF5 data structure for ATL08 products..... 32

201 Figure 2.2. ATL03 granule regions; graphic from ATL03 ATBD (Neumann et al.)..... 33

202 Figure 2.3. ATL08 product regions..... 34

203 Figure 2.4. Illustration of canopy photons (red dots) interaction in a vegetated area.
204 Relative canopy heights, H_i , are computed by differencing the canopy photon height
205 from an interpolated terrain surface. 40

206 Figure 3.1. Combination of noise filtering algorithms to create a superset of input
207 data for surface finding algorithms. 65

208 Figure 3.2. Histogram of the number of photons within a search radius. This
209 histogram is used to determine the threshold for the DRAGANN approach..... 67

210 Figure 3.3. Output from DRAGANN filtering. Signal photons are shown as blue..... 69

211 Figure 3.4. Flowchart of overall surface finding method..... 70

212 Figure 3.5. Plot of Signal Photons (black) from 2014 MABEL flight over Alaska and
213 de-trended photons (red). 71

214 Figure 3.6. Shape Parameter for variable window size..... 74

215 Figure 3.7. Illustration of the standard deviations calculated for each moving
216 window to identify the amount of spread of signal photons within a given window.
217 76

218 Figure 3.8. Three iterations of the ground finding concept for L -km segments with
219 canopy..... 77

**ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002**

220	Figure 3.9. Example of the intermediate ground and top of canopy surfaces	
221	calculated from MABEL flight data over Alaska during July 2014.	80
222	Figure 3.10. Example of classified photons from MABEL data collected in Alaska	
223	2014. Red photons are photons classified as terrain. Green photons are classified as	
224	top of canopy. Canopy photons (shown as blue) are considered as photons lying	
225	between the terrain surface and top of canopy.....	81
226	Figure 3.11. Example of classified photons from MABEL data collected in Alaska	
227	2014. Red photons are photons classified as terrain. Green photons are classified as	
228	top of canopy. Canopy photons (shown as blue) are considered as photons lying	
229	between the terrain surface and top of canopy.....	82
230	Figure 3.12. Example of classified photons from MABEL data collected in Alaska	
231	2014. Red photons are photons classified as terrain. Green photons are classified as	
232	top of canopy. Canopy photons (shown as blue) are considered as photons lying	
233	between the terrain surface and top of canopy.....	83
234	Figure 5.1. Example of <i>L-km</i> segment classifications and interpolated ground	
235	surface.	122
236		

237 **1 INTRODUCTION**

238 This document describes the theoretical basis and implementation of the
239 processing algorithms and data parameters for Level 3 land and vegetation heights
240 for the non-polar regions of the Earth. The ATL08 product contains heights for both
241 terrain and canopy in the along-track direction as well as other descriptive
242 parameters derived from the measurements. At the most basic level, a derived surface
243 height from the ATLAS instrument at a given time is provided relative to the WGS-84
244 ellipsoid. Height estimates from ATL08 can be compared with other geodetic data and
245 used as input to higher-level ICESat-2 products, namely ATL13 and ATL18. ATL13
246 will provide estimates of inland water-related heights and associated descriptive
247 parameters. ATL18 will consist of gridded maps for terrain and canopy features.

248 The ATL08 product will provide estimates of terrain heights, canopy heights,
249 and canopy cover at fine spatial scales in the along-track direction. Along-track is
250 defined as the direction of travel of the ICESat-2 satellite in the velocity vector.
251 Parameters for the terrain and canopy will be provided at a fixed step-size of 100 m
252 along the ground track referred to as a segment. A fixed segment size of 100 m was
253 chosen to provide continuity of data parameters on the ATL08 data product. From an
254 analysis perspective, it is difficult and cumbersome to attempt to relate canopy cover
255 over variable lengths. Furthermore, a segment size of 100 m will facilitate a simpler
256 combination of along-track data to create the gridded products.

257 We anticipate that the signal returned from the weak beam will be sufficiently
258 weak and may prohibit the determination of both a terrain and canopy segment
259 height, particularly over areas of dense vegetation. However, in more arid regions we
260 anticipate producing a terrain height for both the weak and strong beams.

261 In this document, section 1 provides a background of lidar in the ecosystem
262 community as well as describing photon counting systems and how they differ from
263 discrete return lidar systems. Section 2 provides an overview of the Land and

264 Vegetation parameters and how they are defined on the data product. Section 3
265 describes the basic methodology that will be used to derive the parameters for ATL08.
266 Section 4 describes the processing steps, input data, and procedure to derive the data
267 parameters. Section 5 will describe the test data and specific tests that NASA's
268 implementation of the algorithm should pass in order to determine a successful
269 implementation of the algorithm.

270

271 **1.1. Background**

272 The Earth's land surface is a complex mosaic of geomorphic units and land
273 cover types resulting in large variations in terrain height, slope, roughness, vegetation
274 height and reflectance, often with the variations occurring over very small spatial
275 scales. Documentation of these landscape properties is a first step in understanding
276 the interplay between the formative processes and response to changing conditions.
277 Characterization of the landscape is also necessary to establish boundary conditions
278 for models which are sensitive to these properties, such as predictive models of
279 atmospheric change that depend on land-atmosphere interactions. Topography, or
280 land surface height, is an important component for many height applications, both to
281 the scientific and commercial sectors. The most accurate global terrain product was
282 produced by the Shuttle Radar Topography Mission (SRTM) launched in 2000;
283 however, elevation data are limited to non-polar regions. The accuracy of SRTM
284 derived elevations range from 5 – 10 m, depending upon the amount of topography
285 and vegetation cover over a particular area. ICESat-2 will provide a global distribution
286 of geodetic measurements (of both the terrain surface and relative canopy heights)
287 which will provide a significant benefit to society through a variety of applications
288 including sea level change monitoring, forest structural mapping and biomass
289 estimation, and improved global digital terrain models.

290 In addition to producing a global terrain product, monitoring the amount and

291 distribution of above ground vegetation and carbon pools enables improved
292 characterization of the global carbon budget. Forests play a significant role in the
293 terrestrial carbon cycle as carbon pools. Events, such as management activities
294 (Krankina et al. 2012) and disturbances can release carbon stored in forest above
295 ground biomass (AGB) into the atmosphere as carbon dioxide, a greenhouse gas that
296 contributes to climate change (Ahmed et al. 2013). While carbon stocks in nations
297 with continuous national forest inventories (NFIs) are known, complications with NFI
298 carbon stock estimates exist, including: (1) ground-based inventory measurements
299 are time consuming, expensive, and difficult to collect at large-scales (Houghton
300 2005; Ahmed et al. 2013); (2) asynchronously collected data; (3) extended time
301 between repeat measurements (Houghton 2005); and (4) the lack of information on
302 the spatial distribution of forest AGB, required for monitoring sources and sinks of
303 carbon (Houghton 2005). Airborne lidar has been used for small studies to capture
304 canopy height and in those studies canopy height variation for multiple forest types
305 is measured to approximately 7 m standard deviation (Hall et al., 2011).

306 Although the spatial extent and changes to forests can be mapped with existing
307 satellite remote sensing data, the lack of information on forest vertical structure and
308 biomass limits the knowledge of biomass/biomass change within the global carbon
309 budget. Based on the global carbon budget for 2015 (Quere et al., 2015), the largest
310 remaining uncertainties about the Earth's carbon budget are in its terrestrial
311 components, the global residual terrestrial carbon sink, estimated at 3.0 ± 0.8
312 GtC/year for the last decade (2005-2014). Similarly, carbon emissions from land-use
313 changes, including deforestation, afforestation, logging, forest degradation and
314 shifting cultivation are estimated at 0.9 ± 0.5 GtC /year. By providing information on
315 vegetation canopy height globally with a higher spatial resolution than previously
316 afforded by other spaceborne sensors, the ICESat-2 mission can contribute
317 significantly to reducing uncertainties associated with forest vegetation carbon.

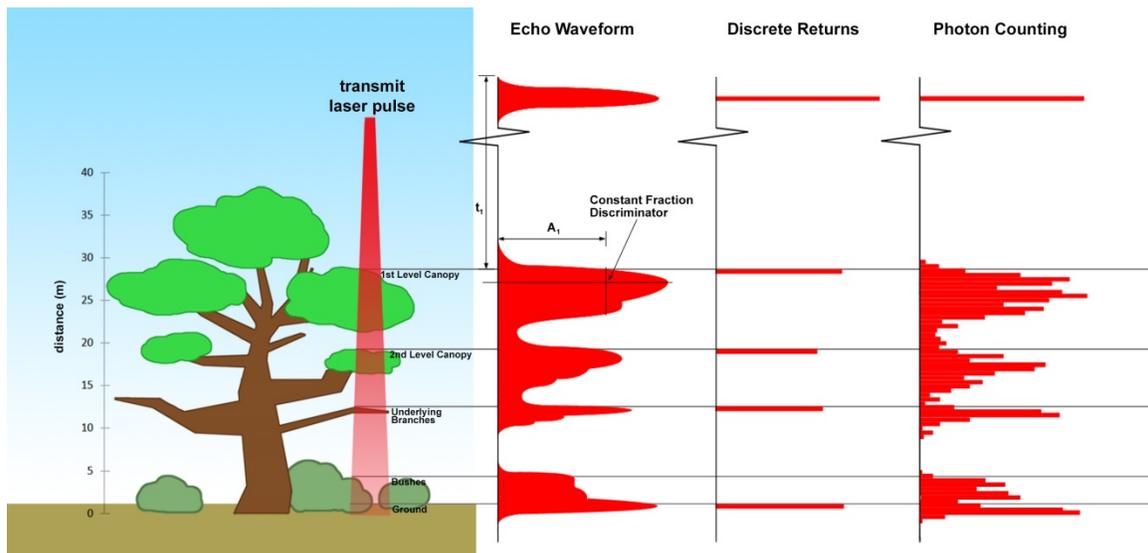
318 Although ICESat-2 is not positioned to provide global biomass estimates due
319 to its profiling configuration and somewhat limited detection capabilities, it is
320 anticipated that the data products for vegetation will be complementary to ongoing
321 biomass and vegetation mapping efforts. Synergistic use of ICESat-2 data with other
322 space-based mapping systems is one solution for extended use of ICESat-2 data.
323 Possibilities include NASA's Global Ecosystems Dynamics Investigation (GEDI) lidar
324 planned to fly onboard the International Space Station (ISS) or imaging sensors, such
325 as Landsat 8, or NASA/ISRO -NISAR radar mission.

326

327 **1.2 Photon Counting Lidar**

328 Rather than using an analog, full waveform system similar to what was utilized
329 on the ICESat/GLAS mission, ICESat-2 will employ a photon counting lidar. Photon
330 counting lidar has been used successfully for ranging for several decades in both the
331 science and defense communities. Photon counting lidar systems operate on the
332 concept that a low power laser pulse is transmitted and the detectors used are
333 sensitive at the single photon level. Due to this type of detector, any returned photon
334 whether from the reflected signal or solar background can trigger an event within the
335 detector. A discussion regarding discriminating between signal and background noise
336 photons is discussed later in this document. A question of interest to the ecosystem
337 community is to understand where within the canopy is the photon likely to be
338 reflected. Figure 1.1 is an example of three different laser detector modalities: full
339 waveform, discrete return, and photon counting. Full waveform sensors record the
340 entire temporal profile of the reflected laser energy through the canopy. In contrast,
341 discrete return systems have timing hardware that record the time when the
342 amplitude of the reflected signal energy exceeds a certain threshold amount. A photon
343 counting system, however, will record the arrival time associated with a single
344 photon detection that can occur anywhere within the vertical distribution of the
345 reflected signal. If a photon counting lidar system were to dwell over a surface for a

346 significant number of shots (i.e. hundreds or more), the vertical distribution of the
347 reflected photons will resemble a full waveform. Thus, while an individual photon
348 could be reflected from anywhere within the vertical canopy, the probability
349 distribution function (PDF) of that reflected photon would be the full waveform.
350 Furthermore, the probability of detecting the top of the tree is not as great as
351 detecting reflective surfaces positioned deeper into the canopy where the bulk of
352 leaves and branches are located. As one might imagine, the PDF will differ according
353 to canopy structure and vegetation physiology. For example, the PDF of a conifer tree
354 will look different than broadleaf trees.



355

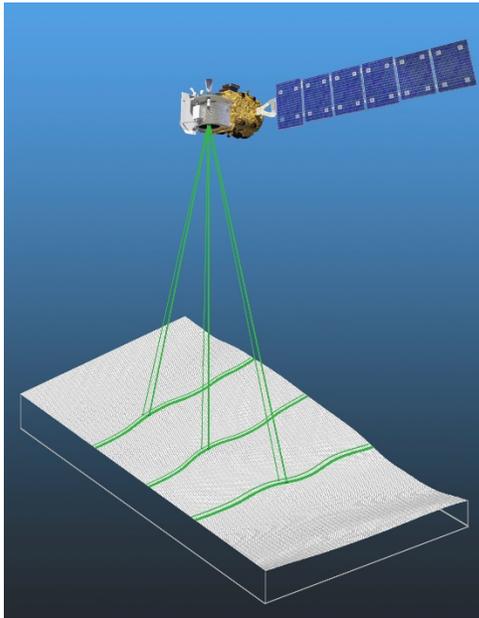
356 Figure 1.1. Various modalities of lidar detection. Adapted from Harding, 2009.

357 A cautionary note, the photon counting PDF that is illustrated in Figure 1.1 is
358 merely an illustration if enough photons (i.e. hundreds of photons or more) were to
359 be reflected from a target. In reality, due to the spacecraft speed, ATLAS will record 0
360 – 4 photons per transmit laser pulse over vegetation.

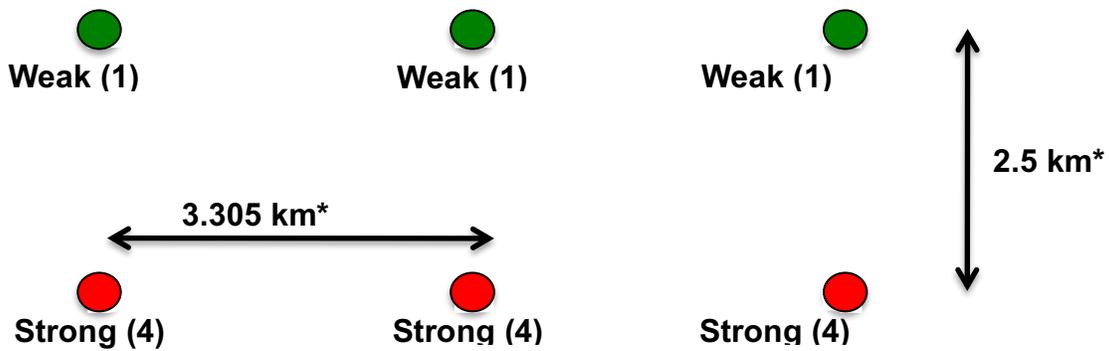
361

362 **1.3 The ICESat-2 concept**

363 The Advanced Topographic Laser Altimeter System (ATLAS) instrument
364 designed for ICESat-2 will utilize a different technology than the GLAS instrument
365 used for ICESat. Instead of using a high-energy, single-beam laser and digitizing the
366 entire temporal profile of returned laser energy, ATLAS will use a multi-beam,
367 micropulse laser (sometimes referred to as photon-counting). The travel time of each
368 detected photon is used to determine a range to the surface which, when combined
369 with satellite attitude and pointing information, can be geolocated into a unique XYZ
370 location on or near the Earth's surface. For more information on how the photons
371 from ICESat-2 are geolocated, refer to ATL03 ATBD. The XYZ positions from ATLAS
372 are subsequently used to derive surface and vegetation properties. The ATLAS
373 instrument will operate at 532 nm in the green range of the electromagnetic (EM)
374 spectrum and will have a laser repetition rate of 10 kHz. The combination of the laser
375 repetition rate and satellite velocity will result in one outgoing laser pulse
376 approximately every 70 cm on the Earth's surface and each spot on the surface is ~13
377 m in diameter. Each transmitted laser pulse is split by a diffractive optical element in
378 ATLAS to generate six individual beams, arranged in three pairs (Figure 1.2). The
379 beams within each pair have different transmit energies ('weak' and 'strong', with an
380 energy ratio of approximately 1:4) to compensate for varying surface reflectance. The
381 beam pairs are separated by ~3.3 km in the across-track direction and the strong and
382 weak beams are separated by ~2.5 km in the along-track direction. As ICESat-2 moves
383 along its orbit, the ATLAS beams describe six tracks on the Earth's surface; the array
384 is rotated slightly with respect to the satellite's flight direction so that tracks for the
385 fore and aft beams in each column produce pairs of tracks – each separated by
386 approximately 90 m.



387

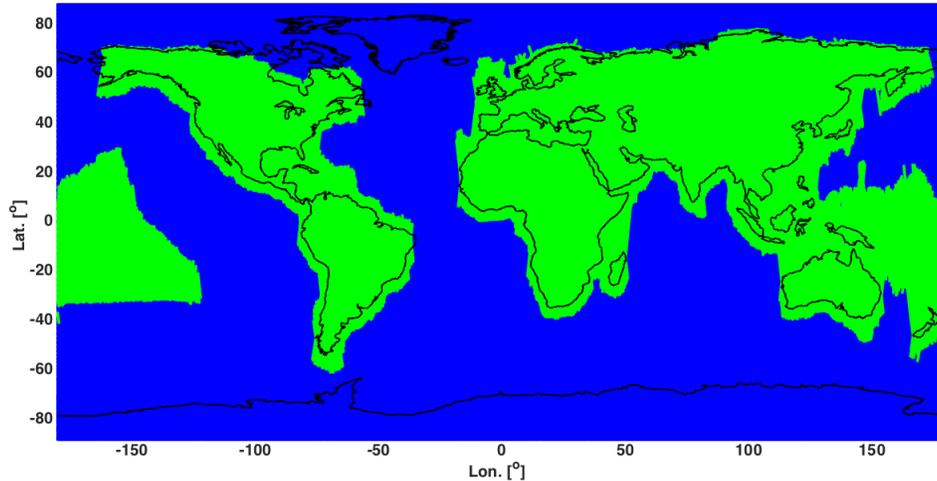


388

389 Figure 1.2. Schematic of 6-beam configuration for ICESat-2 mission. The laser energy will
390 be split into 3 laser beam pairs – each pair having a weak spot (1X) and a strong spot (4X).

391 The motivation behind this multi-beam design is its capability to compute
392 cross-track slopes on a per-orbit basis, which contributes to an improved
393 understanding of ice dynamics. Previously, slope measurements of the terrain were
394 determined via repeat-track and crossover analysis. The laser beam configuration as
395 proposed for ICESat-2 is also beneficial for terrestrial ecosystems compared to GLAS
396 as it enables a denser spatial sampling in the non-polar regions. To achieve a spatial
397 sampling goal of no more than 2 km between equatorial ground tracks, ICESat-2 will

398 be off-nadir pointed a maximum of 1.8 degrees from the reference ground track
399 during the entire mission.



400

401 Figure 1.3. Illustration of off-nadir pointing scenarios. Over land (green regions) in the
402 mid-latitudes, ICESat-2 will be pointed away from the repeat ground tracks to increase the
403 density of measurements over terrestrial surfaces.

404 ICESat-2 is designed to densely sample the Earth's surface, permitting
405 scientists to measure and quantitatively characterize vegetation across vast
406 expanses, e.g., nations, continents, globally. ICESat-2 will acquire synoptic
407 measurements of vegetation canopy height, density, the vertical distribution of
408 photosynthetically active material, leading to improved estimates of forest biomass,
409 carbon, and volume. In addition, the orbital density, i.e., the number of orbits per unit
410 area, at the end of the three year mission will facilitate the production of gridded
411 global products. ICESat-2 will provide the means by which an accurate "snapshot" of
412 global biomass and carbon may be constructed for the mission period.

413

414 **1.4 Height Retrieval from ATLAS**

415 Light from the ATLAS lasers reaches the earth's surface as flat disks of down-
416 traveling photons approximately 50 cm in vertical extent and spread over
417 approximately 14 m horizontally. Upon hitting the earth's surface, the photons are
418 reflected and scattered in every direction and a handful of photons return to the
419 ATLAS telescope's focal plane. The number of photon events per laser pulse is a
420 function of outgoing laser energy, surface reflectance, solar conditions, and scattering
421 and attenuation in the atmosphere. For highly reflective surfaces (such as land ice)
422 and clear skies, approximately 10 signal photons from a single strong beam are
423 expected to be recorded by the ATLAS instrument for a given transmit laser pulse.
424 Over vegetated land where the surface reflectance is considerably less than snow or
425 ice surfaces, we expect to see fewer returned photons from the surface. Whereas
426 snow and ice surfaces have high reflectance at 532 nm (typical Lambertian
427 reflectance between 0.8 and 0.98 (Martino, GSFC internal report, 2010)), canopy and
428 terrain surfaces have much lower reflectance (typically around 0.3 for soil and 0.1 for
429 vegetation) at 532 nm. As a consequence we expect to see 1/3 to 1/9 as many photons
430 returned from terrestrial surfaces as from ice and snow surfaces. For vegetated
431 surfaces, the number of reflected signal photon events per transmitted laser pulse is
432 estimated to range between 0 to 4 photons.

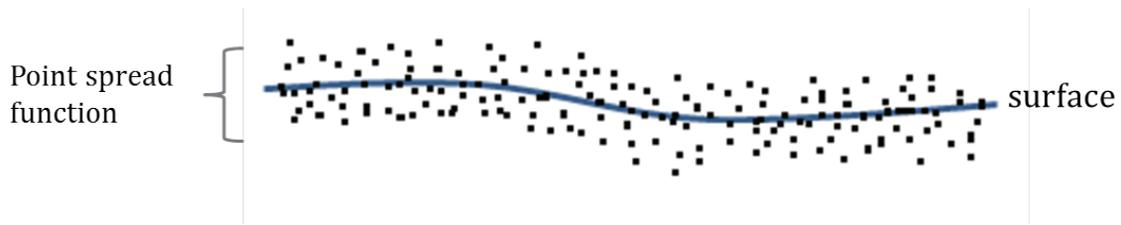
433 The time measured from the detected photon events are used to compute a
434 range, or distance, from the satellite. Combined with the precise pointing and attitude
435 information about the satellite, the range can be geolocated into a XYZ point (known
436 as a geolocated photon) above the WGS-84 reference ellipsoid. In addition to
437 recording photons from the reflected signal, the ATLAS instrument will detect
438 background photons from sunlight which are continually entering the telescope. A
439 primary objective of the ICESat-2 data processing software is to correctly
440 discriminate between signal photons and background photons. Some of this
441 processing occurs at the ATL03 level and some of it also occurs within the software

442 for ATL08. At ATL03, this discrimination is done through a series of three steps of
443 progressively finer resolution with some processing occurring onboard the satellite
444 prior to downlink of the raw data. The ATL03 data product produces a classification
445 between signal and background (i.e. noise) photons, and further discussion on that
446 classification process can be read in the ATL03 ATBD. In addition, all geophysical
447 corrections (e.g. ocean tide, solid earth tide models, etc.) are not applied to the
448 position of the individual geolocated photons at the ATL03 level, but they are
449 provided on the data product if there exists a need to apply them. Thus, all of the
450 heights processed in the ATL08 algorithm consists of the ATL03 heights with respect
451 to the WGS-84 ellipsoid.

452

453 ***1.5 Accuracy Expected from ATLAS***

454 There are a variety of elements that contribute to the elevation accuracy that
455 are expected from ATLAS and the derived data products. Elevation accuracy is a
456 composite of ranging precision of the instrument, radial orbital uncertainty,
457 geolocation knowledge, forward scattering in the atmosphere, and tropospheric path
458 delay uncertainty. The ranging precision seen by ATLAS will be a function of the laser
459 pulse width, the surface area potentially illuminated by the laser, and uncertainty in
460 the timing electronics. The requirement on radial orbital uncertainty is specified to
461 be less than 4 cm and tropospheric path delay uncertainty is estimated to be 3 cm. In
462 the case of ATLAS, the ranging precision for flat surfaces, is expected to have a
463 standard deviation of approximately 25 cm. The composite of each of the errors can
464 also be thought of as the spread of photons about a surface (see Figure 1.4) and is
465 referred to as the point spread function or Znoise.



466

467 Figure 1.4. Illustration of the point spread function, also referred to as Znoise, for a series
468 of photons about a surface.

469 The estimates of σ_{Orbit} , $\sigma_{troposphere}$, $\sigma_{forwardscattering}$, $\sigma_{pointing}$, and σ_{timing}
470 for a photon will be represented on the ATL03 data product as the final geolocated
471 accuracy in the X, Y, and Z (or height) direction. In reality, these parameters have
472 different temporal and spatial scales, however until ICESat-2 is on orbit, it is uncertain
473 how these parameters will vary over time. As such, Equation 1.1 may change once the
474 temporal aspects of these parameters are better understood. For a preliminary
475 quantification of the uncertainties, Equation 1.1 is valid to incorporate the instrument
476 related factors.

477
$$\sigma_Z = \sqrt{\sigma_{Orbit}^2 + \sigma_{trop}^2 + \sigma_{forwardscattering}^2 + \sigma_{pointing}^2 + \sigma_{timing}^2}$$
 Eqn. 1.1

478

479 Although σ_Z on the ATL03 product represents the best understanding of the
480 uncertainty for each geolocated photon, it does not incorporate the uncertainty
481 associated with local slope of the topography. The slope component to the geolocation
482 uncertainty is a function of both the geolocation knowledge of the pointing (which is
483 required to be less than 6.5 m) multiplied by the tangent of the surface slope. In a case
484 of flat topography (≤ 1 degree slope), $\sigma_Z \leq 25$ cm, whereas in the case of a 10 degree
485 surface slope, $\sigma_Z = 119$ cm. The uncertainty associated with the local slope will be
486 combined with σ_Z to produce the term $\sigma_{AtlasLand}$.

487
$$\sigma_{AtlasLand} = \sqrt{\sigma_Z^2 + \sigma_{topo}^2}$$
 Eqn. 1.2

488 $\sigma_{topo} =$ Eqn. 1.3

489 Ultimately, the uncertainty that will be reported on the data product ATL08
490 will include the $\sigma_{Atlas_{Land}}$ term and the local rms values of heights computed within
491 each data parameter segment. For example, calculations of terrain height will be
492 made on photons classified as terrain photons (this process is described in the
493 following sections). The uncertainty of the terrain height for a segment is described
494 in Equation 1.4, where the root mean square term of $\sigma_{Atlas_{Land}}$ and rms of terrain
495 heights are normalized by the number of terrain photons for that given segment.

496
$$\sigma_{ATL08_{segment}} = \sqrt{\sigma_{Atlas_{Land}}^2 + \sigma_{Zrms_{segment_class}}^2}$$
 Eqn. 1.4

497

498 **1.6 Additional Potential Height Errors from ATLAS**

499 Some additional potential height errors in the ATL08 terrain and vegetation
500 product can come from a variety of sources including:

- 501 a. Vertical sampling error. ATLAS height estimates are based on a
502 random sampling of the surface height distribution. Photons may
503 be reflected from anywhere within the PDF of the reflecting surface;
504 more specifically, anywhere from within the canopy. A detailed
505 look at the potential effect of vertical sampling error is provided in
506 Neuenschwander and Magruder (2016).
- 507 b. Background noise. Random noise photons are mixed with the
508 signal photons so classified photons will include random outliers.
- 509 c. Complex topography. The along-track product may not always
510 represent complex surfaces, particularly if the density of ground
511 photons does not support an accurate representation.

512 d. Vegetation. Dense vegetation may preclude reflected photon
513 events from reaching the underlying ground surface. An incorrect
514 estimation of the underlying ground surface will subsequently lead
515 to an incorrect canopy height determination.

516 e. Misidentified photons. The product from ATL03 combined with
517 additional noise filtering may not identify the correct photons as
518 signal photons.

519

520 **1.7 Dense Canopy Cases**

521 Although the height accuracy produced from ICESat-2 is anticipated to be
522 superior to other global height products (e.g. SRTM), for certain biomes photon
523 counting lidar data as it will be collected by the ATLAS instrument present a challenge
524 for extracting both the terrain and canopy heights, particularly for areas of dense
525 vegetation. Due to the relatively low laser power, we anticipate that the along-track
526 signal from ATLAS may lose ground signal under dense forest (e.g. >96% canopy
527 closure) and in situations where cloud cover obscures the terrestrial signal. In areas
528 having dense vegetation, it is likely that only a handful of photons will be returned
529 from the ground surface with the majority of reflections occurring from the canopy.
530 A possible source of error can occur with both the canopy height estimates and the
531 terrain heights if the vegetation is particularly dense and the ground photons were
532 not correctly identified.

533

534 **1.8 Sparse Canopy Cases**

535 Conversely, sparse canopy cases also pose a challenge to vegetation height
536 retrievals. In these cases, expected reflected photon events from sparse trees or
537 shrubs may be difficult to discriminate between solar background noise photons. The

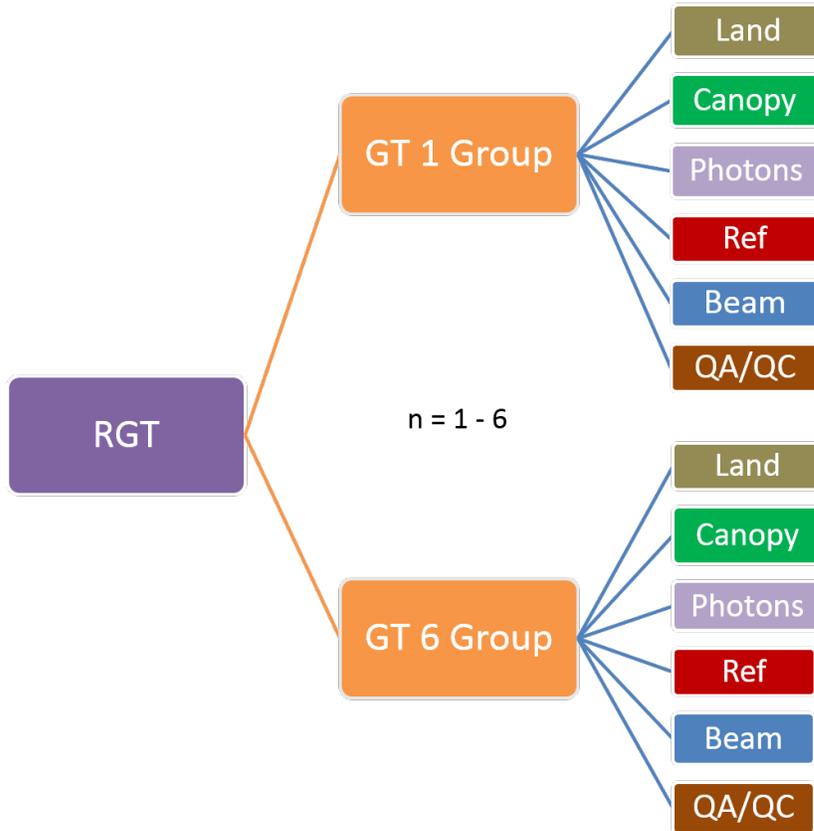
***ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002***

538 algorithms being developed for ATL08 operate under the assumption that signal
539 photons are close together and noise photons will be more isolated in nature. Thus,
540 signal (in this case canopy) photons may be incorrectly identified as solar background
541 noise on the data product. Due to the nature of the photon counting processing,
542 canopy photons identified in areas that have extremely low canopy cover <15% will
543 be filtered out and reassigned as noise photons.

544

545 **2. ATL08: DATA PRODUCT**

546 The ATL08 product will provide estimates of terrain height, canopy height,
547 and canopy cover at fine spatial scales in the along-track direction. In accordance with
548 the HDF-driven structure of the ICESat-2 products, the ATL08 product will
549 characterize each of the six Ground Tracks (GT) associated with each Reference
550 Ground Track (RGT) for each cycle and orbit number. Each ground track group has a
551 distinct beam number, distance from the reference track, and transmit energy
552 strength, and all beams will be processed independently using the same sequence of
553 steps described within ATL08. Each ground track group (GT) on the ATL08 product
554 contains subgroups for land and canopy heights segments as well as beam and
555 reference parameters useful in the ATL08 processing. In addition, the labeled photons
556 that are used to determine the data parameters will be indexed back to the ATL03
557 products such that they are available for further, independent analysis. A layout of
558 the ATL08 HDF product is shown in Figure 2.1. The six GTs are numbered from left to
559 right, regardless of satellite orientation.



560

561 Figure 2.1. HDF5 data structure for ATL08 products

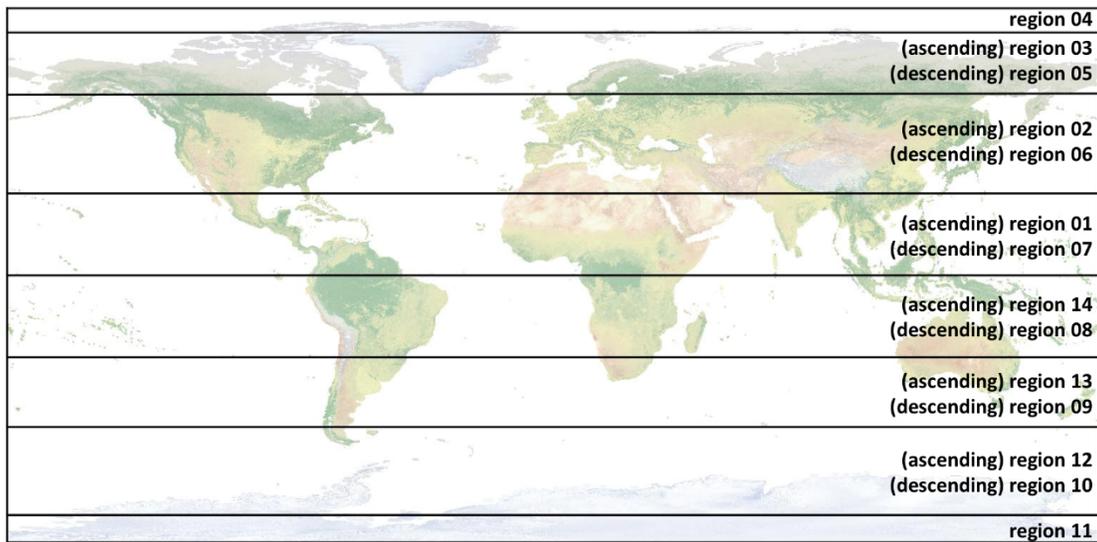
562

563 For each data parameter, terrain surface elevation and canopy heights will be
 564 provided at a fixed segment size of 100 meters along the ground track. Based on the
 565 satellite velocity and the expected number of reflected photons for land surfaces, each
 566 segment should have more than 100 signal photons, but in some instances there may
 567 be less than 100 signal photons per segment. If a segment has less than 50 classed
 568 (i.e., labeled by ATL08 as ground, canopy, or top of canopy) photons we feel this
 569 would not accurately represent the surface. Thus, an invalid value will be reported in
 570 all height fields. In the event that there are more than 50 classed photons, but a terrain
 571 height cannot be determined due to an insufficient number of ground photons, (e.g.

**ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002**

572 lack of photons penetrating through dense canopy), the only reported terrain height
573 will be the interpolated surface height.

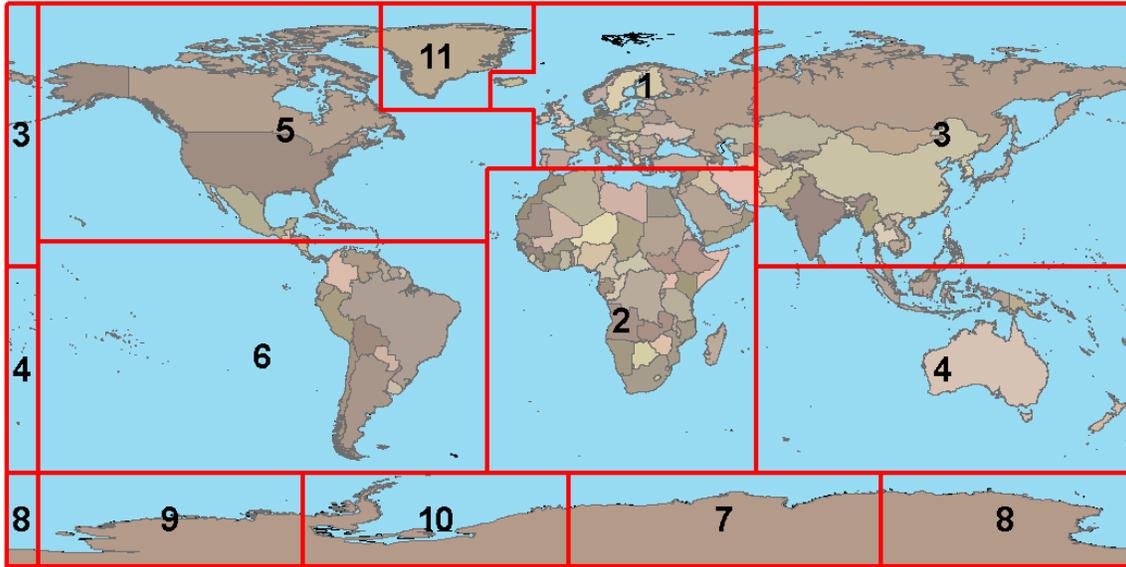
574 The ATL08 product will be produced per granule based on the ATL03 defined
575 regions (see Figure 2.2). Thus, the ATL08 file/name convention scheme will match
576 the file/naming convention for ATL03 –in attempt for reducing complexity to allow
577 users to examine both data products.



578

579 Figure 2.2. ATL03 granule regions; graphic from ATL03 ATBD (Neumann et al.).

580 The ATL08 product additionally has its own internal regions, which are
581 roughly assigned by continent, as shown by Figure 2.3. For the regions covering
582 Antarctica (regions 7, 8, 9, 10) and Greenland (region 11), the ATL08 algorithm will
583 assume that no canopy is present. These internal ATL08 regions will be noted in the
584 ATL08 product (see parameter atl08_region in Section 2.4.19). Note that the regions
585 for each ICESat-2 product are not the same.



586

587 Figure 2.3. ATL08 product regions.

588

589 **2.1 Subgroup: Land Parameters**

590 ATL08 terrain height parameters are defined in terms of the absolute height
591 above the reference ellipsoid.

592 Table 2.1. Summary table of land parameters on ATL08.

Group	Data type	Description	Source
segment_id_beg	Integer	First along-track segment_id number in 100-m segment	ATL03
segment_id_end	Integer	Last along-track segment_id number in 100-m segment	ATL03
h_te_mean	Float	Mean terrain height for segment	computed
h_te_median	Float	Median terrain height for segment	computed
h_te_min	Float	Minimum terrain height for segment	computed
h_te_max	Float	Maximum terrain height for segment	computed
h_te_mode	Float	Mode of terrain height for segment	computed

**ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002**

h_te_skew	Float	Skew of terrain height for segment	computed
n_te_photons	Integer	Number of ground photons in segment	computed
h_te_interp	Float	Interpolated terrain surface height at mid-point of segment	computed
h_te_std	Float	Standard deviation of ground heights about the interpolated ground surface	computed
h_te_uncertainty	Float	Uncertainty of ground height estimates. Includes all known uncertainties such as geolocation, pointing angle, timing, radial orbit errors, etc.	computed from Equation 1.4
terrain_slope	Float	Slope of terrain within segment	computed
h_te_best_fit	Float	Best fit terrain elevation at the 100 m segment mid-point location	computed

593

594 **2.1.1** Georeferenced_segment_number_beg

595 (parameter = segment_id_beg). The first along-track segment_id in each 100-m
596 segment. Each 100-m segment consists of five sequential 20-m segments provided
597 from the ATL03 product, which are labeled as segment_id. The segment_id is a seven
598 digit number that uniquely identifies each along track segment, and is written at the
599 along-track geolocation segment rate (i.e. ~20m along track). The four digit RGT
600 number can be combined with the seven digit segment_id number to uniquely define
601 any along-track segment number. Values are sequential, with 0000001 referring to
602 the first segment after the equatorial crossing of the ascending node.

603 **2.1.2** Georeferenced_segment_number_end

604 (parameter = segment_id_end). The last along-track segment_id in each 100-m
605 segment. Each 100-m segment consists of five sequential 20-m segments provided
606 from the ATL03 product, which are labeled as segment_id. The segment_id is a seven
607 digit number that uniquely identifies each along track segment, and is written at the

608 along-track geolocation segment rate (i.e. ~20m along track). The four digit RGT
609 number can be combined with the seven digit segment_id number to uniquely define
610 any along-track segment number. Values are sequential, with 0000001 referring to
611 the first segment after the equatorial crossing of the ascending node.

612 **2.1.3 Segment_terrain_height_mean**

613 (parameter = h_te_mean). Estimated mean of the terrain height above the
614 reference ellipsoid derived from classified ground photons within the 100 m segment.
615 If a terrain height cannot be directly determined within the segment (i.e. there are not
616 a sufficient number of ground photons), only the interpolated terrain height will be
617 reported. Required input data is classified point cloud (i.e. photons labeled as either
618 canopy or ground in the ATL08 processing). This parameter will be derived from only
619 classified ground photons.

620 **2.1.4 Segment_terrain_height_med**

621 (parameter = h_te_median). Median terrain height above the reference
622 ellipsoid derived from the classified ground photons within the 100 m segment. If
623 there are not a sufficient number of ground photons, an invalid value will be reported
624 –no interpolation will be done. Required input data is classified point cloud (i.e.
625 photons labeled as either canopy or ground in the ATL08 processing). This parameter
626 will be derived from only classified ground photons.

627 **2.1.5 Segment_terrain_height_min**

628 (parameter = h_te_min). Minimum terrain height above the reference ellipsoid
629 derived from the classified ground photons within the 100 m segment. If there are
630 not a sufficient number of ground photons, an invalid value will be reported –no
631 interpolation will be done. Required input data is classified point cloud (i.e. photons
632 labeled as either canopy or ground in the ATL08 processing). This parameter will be
633 derived from only classified ground photons.

634 **2.1.6** Segment_terrain_height_max

635 (parameter = h_te_max). Maximum terrain height above the reference
636 ellipsoid derived from the classified ground photons within the 100 m segment. If
637 there are not a sufficient number of ground photons, an invalid value will be reported
638 –no interpolation will be done. Required input data is classified point cloud (i.e.
639 photons labeled as either canopy or ground in the ATL08 processing). This parameter
640 will be derived from only classified ground photons.

641 **2.1.7** Segment_terrain_height_mode

642 (parameter = h_te_mode). Mode of the classified ground photon heights above
643 the reference ellipsoid within the 100 m segment. If there are not a sufficient number
644 of ground photons, an invalid value will be reported –no interpolation will be done.
645 Required input data is classified point cloud (i.e. photons labeled as either canopy or
646 ground in the ATL08 processing). This parameter will be derived from only classified
647 ground photons.

648 **2.1.8** Segment_terrain_height_skew

649 (parameter = h_te_skew). The skew of the classified ground photons within the
650 100 m segment. If there are not a sufficient number of ground photons, an invalid
651 value will be reported –no interpolation will be done. Required input data is classified
652 point cloud (i.e. photons labeled as either canopy or ground in the ATL08 processing).
653 This parameter will be derived from only classified ground photons.

654 **2.1.9** Segment_number_terrain_photons

655 (parameter = n_te_photons). Number of terrain photons identified in segment.

656 **2.1.10** Segment height_interp

657 (parameter = h_te_interp). Interpolated terrain surface height above the
658 reference ellipsoid from ATL08 processing at the mid-point of each segment. This
659 interpolated surface is the FINALGROUND estimate (described in section 4.9).

660 **2.1.11** Segment h_te_std

661 (parameter = h_te_std). Standard deviations of terrain points about the
662 interpolated ground surface within the segment. Provides an indication of surface
663 roughness.

664 **2.1.12** Segment_terrain_height_uncertainty

665 (parameter = h_te_uncertainty). Uncertainty of the mean terrain height for the
666 segment. This uncertainty incorporates all systematic uncertainties (e.g. timing,
667 orbits, geolocation, etc.) as well as uncertainty from errors of identified photons. This
668 parameter is described in Section 1, Equation 1.4. If there are not a sufficient number
669 of ground photons, an invalid value will be reported –no interpolation will be done.
670 Required input data is classified point cloud (i.e. photons labeled as either canopy or
671 ground in the ATL08 processing). This parameter will be derived from only classified
672 ground photons. The $\sigma_{segmentclass}$ term in Equation 1.4 represents the standard
673 deviation of the terrain height residuals about the FINALGROUND estimate.

674 **2.1.13** Segment_terrain_slope

675 (parameter = terrain_slope). Slope of terrain within each segment. Slope is
676 computed from a linear fit of the terrain photons. It estimates the rise [m] in relief
677 over each segment [100 m]; e.g., if the slope value is 0.04, there is a 4 m rise over the
678 100 m segment. Required input data are the classified terrain photons.

679 **2.1.14 Segment_terrain_height_best_fit**

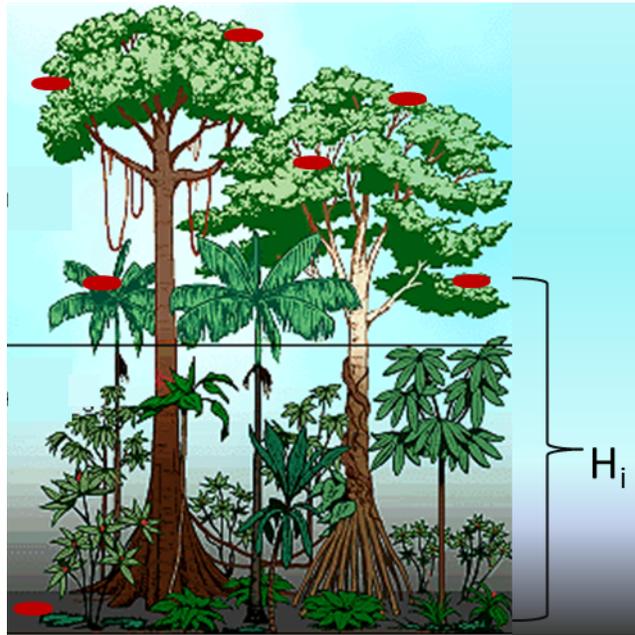
680 (parameter = h_te_best_fit). The best fit terrain elevation at the mid-point
681 location of each 100 m segment. The mid-segment terrain elevation is determined by
682 selecting the best of three fits – linear, 3rd order and 4th order polynomials – to the
683 terrain photons and interpolating the elevation at the mid-point location of the 100
684 m segment. For the linear fit, a slope correction and weighting is applied to each
685 ground photon based on the distance to the slope height at the center of the segment.

686

687 **2.2 Subgroup: Vegetation Parameters**

688 Canopy parameters will be reported on the ATL08 data product in terms of both
689 the absolute height above the reference ellipsoid as well as the relative height above
690 an estimated ground. The relative canopy height, H_i , is computed as the height from
691 an identified canopy photon minus the interpolated ground surface for the same
692 horizontal geolocation (see Figure 2.3). Thus, each identified signal photon above an
693 interpolated surface (including a buffer distance based on the instrument point
694 spread function) is by default considered a canopy photon. Canopy parameters will
695 only be computed for segments where more than 5% of the classed photons are
696 classified as canopy photons.

697



698

699 Figure 2.4. Illustration of canopy photons (red dots) interaction in a vegetated area.
 700 Relative canopy heights, H_i , are computed by differencing the canopy photon height from
 701 an interpolated terrain surface.

702 Table 2.2. Summary table of canopy parameters on ATL08.

Group	Data type	Description	Source
segment_id_beg	Integer	First along-track segment_id number in 100-m segment	ATL03
segment_id_end	Integer	Last along-track segment_id number in 100-m segment	ATL03
canopy_h_metrics_abs	Float	Absolute (H##) canopy height metrics calculated at the following percentiles: 25, 50, 60, 70, 75, 80, 85, 90, 95.	computed
canopy_h_metrics	Float	Relative (RH##) canopy height metrics calculated at the following percentiles: 25, 50, 60, 70, 75, 80, 85, 90, 95.	computed
h_canopy_abs	Float	98% height of all the individual absolute canopy heights for segment.	computed
h_canopy	Float	98% height of all the individual relative canopy heights for segment.	computed
h_mean_canopy_abs	Float	Mean of individual absolute canopy heights within segment	computed

**ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002**

h_mean_canopy	Float	Mean of individual relative canopy heights within segment	computed
h_dif_canopy	Float	Difference between h_canopy and canopy_h_metrics(50)	computed
h_min_canopy_abs	Float	Minimum of individual absolute canopy heights within segment	computed
h_min_canopy	Float	Minimum of individual relative canopy heights within segment	computed
h_max_canopy_abs	Float	Maximum of individual absolute canopy heights within segment. Should be equivalent to H100	computed
h_max_canopy	Float	Maximum of individual relative canopy heights within segment. Should be equivalent to RH100	computed
h_canopy_uncertainty	Float	Uncertainty of the relative canopy height (h_canopy)	computed
canopy_openness	Float	STD of relative heights for all photons classified as canopy photons within the segment to provide inference of canopy openness	computed
toc_roughness	Float	STD of relative heights of all photons classified as top of canopy within the segment	computed
h_canopy_quad	Float	Quadratic mean canopy height	computed
n_ca_photons	Integer4	Number of canopy photons within 100 m segment	computed
n_toc_photons	Integer4	Number of top of canopy photons within 100 m segment	computed
centroid_height	Float	Absolute height above reference ellipsoid associated with the centroid of all signal photons	computed
canopy_rh_conf	Integer	Canopy relative height confidence flag based on percentage of ground and canopy photons within a segment: 0 (<5% canopy), 1 (>5% canopy, <5% ground), 2 (>5% canopy, >5% ground)	computed
canopy_flag	Integer	Flag indicating that canopy was detected using the Landsat Tree Cover Continuous Fields data product	computed
landsat_flag	Integer	Flag indicating that Landsat Tree Cover Continuous Fields data product had more than 50% values >100 for L-km segment	computed
landsat_perc	Float	Average percentage value of the valid (value <= 100) Landsat Tree Cover	

Continuous Fields product for each 100
m segment

703

704 **2.2.1** Georeferenced_segment_number_beg

705 (parameter = segment_id_beg). The first along-track segment_id in each 100-m
706 segment. Each 100-m segment consists of five sequential 20-m segments provided
707 from the ATL03 product, which are labeled as segment_id. The segment_id is a seven
708 digit number that uniquely identifies each along track segment, and is written at the
709 along-track geolocation segment rate (i.e. ~20m along track). The four digit RGT
710 number can be combined with the seven digit segment_id number to uniquely define
711 any along-track segment number. Values are sequential, with 0000001 referring to
712 the first segment after the equatorial crossing of the ascending node.

713 **2.2.2** Georeferenced_segment_number_end

714 (parameter = segment_id_end). The last along-track segment_id in each 100-m
715 segment. Each 100-m segment consists of five sequential 20-m segments provided
716 from the ATL03 product, which are labeled as segment_id. The segment_id is a seven
717 digit number that uniquely identifies each along track segment, and is written at the
718 along-track geolocation segment rate (i.e. ~20m along track). The four digit RGT
719 number can be combined with the seven digit segment_id number to uniquely define
720 any along-track segment number. Values are sequential, with 0000001 referring to
721 the first segment after the equatorial crossing of the ascending node.

722 **2.2.3** Canopy_height_metrics_abs

723 (parameter = canopy_h_metrics_abs). The absolute height metrics (H##) of
724 classified canopy photons above the ellipsoid. The height metrics are sorted based on
725 a cumulative distribution and calculated at the following percentiles: 25, 50, 60, 70,
726 75, 80, 85, 90, 95. These height metrics are often used in the literature to characterize
727 vertical structure of vegetation. One important distinction of these canopy height

728 metrics compared to those derived from other lidar systems (e.g., LVIS or GEDI) is
729 that the ICESat-2 canopy height metrics are heights above the ground surface. These
730 metrics do not include the ground photons. Required input data are the absolute
731 canopy heights of all canopy photons.

732 **2.2.4** Canopy_height_metrics

733 (parameter = canopy_h_metrics). Relative height metrics above the estimated
734 terrain surface (RH##) of classified canopy photons. The height metrics are sorted
735 based on a cumulative distribution and calculated at the following percentiles: 25,
736 50, 60, 70, 75, 80, 85, 90, 95. These height metrics are often used in the literature to
737 characterize vertical structure of vegetation. One important distinction of these
738 canopy height metrics compared to those derived from other lidar systems (e.g., LVIS
739 or GEDI) is that the ICESat-2 canopy height metrics are heights above the ground
740 surface. These metrics do not include the ground photons. Required input data are
741 relative canopy heights above the estimated terrain surface for all canopy photons.

742 **2.2.5** Absolute_segment_canopy_height

743 (parameter = h_canopy_abs). The absolute 98% height of classified canopy
744 photon heights above the ellipsoid. The absolute height from classified canopy
745 photons are sorted into a cumulative distribution, and the height associated with the
746 98% height is reported.

747 **2.2.6** Segment_canopy_height

748 (parameter = h_canopy). The relative 98% height of classified canopy photon
749 heights above the estimated terrain surface. Relative canopy heights have been
750 computed by differencing the canopy photon height from the estimated terrain
751 surface in the ATL08 processing. The relative canopy heights are sorted into a
752 cumulative distribution, and the height associated with the 98% height is reported.

753 **2.2.7** Absolute_segment_mean_canopy

754 (parameter = h_mean_canopy_abs). The absolute mean canopy height for the
755 segment. Absolute canopy heights are the photons heights above the reference
756 ellipsoid. These heights are averaged.

757 **2.2.8** Segment_mean_canopy

758 (parameter = h_mean_canopy). The mean canopy height for the segment.
759 Relative canopy heights have been computed by differencing the canopy photon
760 height from the estimated terrain surface in the ATL08 processing. These heights are
761 averaged.

762 **2.2.9** Segment_dif_canopy

763 (parameter = h_dif_canopy). Difference between h_canopy and
764 canopy_h_metrics(50). This parameter is one metric used to describe the vertical
765 distribution of the canopy within the segment.

766 **2.2.10** Absolute_segment_min_canopy

767 (parameter = h_min_canopy_abs). The minimum absolute canopy height for
768 the segment. Required input data is classified point cloud (i.e. photons labeled as
769 either canopy or ground in the ATL08 processing).

770 **2.2.11** Segment_min_canopy

771 (parameter = h_min_canopy). The minimum relative canopy height for the
772 segment. Required input data is classified point cloud (i.e. photons labeled as either
773 canopy or ground in the ATL08 processing).

774 **2.2.12** Absolute_segment_max_canopy

775 (parameter = h_max_canopy_abs). The maximum absolute canopy height for
776 the segment. This product is equivalent to H100 metric reported in the literature. This
777 parameter, however, has the potential for error as random solar background noise

778 may not have been fully rejected. It is recommended that `h_canopy` or `h_canopy_abs`
779 (i.e., the 98% canopy height) be considered as the top of canopy measurement.
780 Required input data is classified point cloud (i.e. photons labeled as either canopy or
781 ground in the ATL08 processing).

782 **2.2.13** `Segment_max_canopy`

783 (parameter = `h_max_canopy`). The maximum relative canopy height for the
784 segment. This product is equivalent to RH100 metric reported in the literature. This
785 parameter, however, has the potential for error as random solar background noise
786 may not have been fully rejected. It is recommended that `h_canopy` or `h_canopy_abs`
787 (i.e., the 98% canopy height) be considered as the top of canopy measurement.
788 Required input data is classified point cloud (i.e. photons labeled as either canopy or
789 ground in the ATL08 processing).

790 **2.2.14** `Segment_canopy_height_uncertainty`

791 (parameter = `h_canopy_uncertainty`). Uncertainty of the relative canopy
792 height for the segment. This uncertainty incorporates all systematic uncertainties
793 (e.g. timing, orbits, geolocation, etc.) as well as uncertainty from errors of identified
794 photons. This parameter is described in Section 1, Equation 1.4. If there are not a
795 sufficient number of ground photons, an invalid value will be reported –no
796 interpolation will be done. In the case for canopy height uncertainty, the parameter
797 $\sigma_{segmentclass}$ is comprised of both the terrain uncertainty within the segment but also
798 the top of canopy residuals. Required input data is classified point cloud (i.e. photons
799 labeled as either top of canopy or ground in the ATL08 processing). This parameter
800 will be derived from only classified top of canopy photons. The canopy height
801 uncertainty is derived from Equation 1.4, shown below as Equation 1.5, represents
802 the standard deviation of the terrain points and the standard deviation of the top of
803 canopy height photons.

804 $\sigma_{ATL08_{segment_ch}} =$ Eqn 1.5

805

806 **2.2.15 Segment_canopy_openness**

807 (parameter = canopy_openness). Standard deviation of relative canopy
808 heights within each segment. This parameter will potentially provide an indicator of
809 canopy openness as a greater standard deviation of heights indicates greater
810 penetration of the laser energy into the canopy. Required input data is classified point
811 cloud (i.e. photons labeled as either canopy or ground in the ATL08 processing).

812 **2.2.16 Segment_top_of_canopy_roughness**

813 (parameter = toc_roughness). Standard deviation of relative top of canopy
814 heights within each segment. This parameter will potentially provide an indicator of
815 canopy variability. Required input data is classified point cloud (i.e. photons labeled
816 as the top of the canopy in the ATL08 processing).

817 **2.2.17 Segment_canopy_quadratic_height**

818 (parameter = h_canopy_quad). The quadratic mean relative height of classified
819 canopy photons. The quadratic mean height is computed as:

820
$$qmh = \sqrt{\sum_{i=1}^{n_{ca_photons}} \frac{h_i^2}{n_{ca_photons}}}$$

821 **2.2.18 Segment_number_canopy_photons**

822 (parameter = n_ca_photons). Number of canopy photons within each segment.
823 Required input data is classified point cloud (i.e. photons labeled as either canopy or
824 ground in the ATL08 processing).

825 **2.2.19** Segment_number_top_canopy_photons

826 (parameter = n_toc_photons). Number of top of canopy photons within each
827 segment. Required input data is classified point cloud (i.e. photons labeled as top of
828 canopy in the ATL08 processing).

829 **2.2.20** Centroid_height

830 (parameter = centroid_height). Optical centroid of all photons classified as
831 either canopy or ground points within a segment. The heights used in this calculation
832 are absolute heights above the reference ellipsoid. This parameter is equivalent to the
833 centroid height produced on ICESat GLA14.

834 **2.2.21** Segment_rel_canopy_conf

835 (parameter = canopy_rh_conf). Canopy relative height confidence flag based
836 on percentage of ground photons and percentage of canopy photons, relative to the
837 total classified (ground and canopy) photons within a segment: 0 (<5% canopy), 1
838 (>5% canopy and <5% ground), 2 (>5% canopy and >5% ground). This is a measure
839 based on the quantity, not the quality, of the classified photons in each segment.

840 **2.2.22** Canopy_flag

841 (parameter = canopy_flag). Flag indicating that canopy was detected using the
842 Landsat Continuous Cover product for the *L-km* segment. Currently, if more than 5%
843 of the Landsat CC pixels along the profile have canopy in them, we make the
844 assumption canopy is present along the entire *L-km* segment.

845 **2.2.23** Landsat_flag

846 (parameter = landsat_flag). Flag indicating that more than 50% of the Landsat
847 Tree Cover Continuous Fields product have values >100 (indicating water, cloud,
848 shadow, or filled values) for the *L-km* segment. Canopy is assumed present along the
849 *L-km* segment if landsat_flag is 1.

850 **2.2.24** Landsat_perc

851 (parameter = landsat_perc). Average percentage value of the valid (value <=

852 100) Landsat Tree Cover Continuous Fields product pixels that overlap within each

853 100 m segment.

854

855

856 **2.3 Subgroup: Photons**

857 The subgroup for photons contains the classified photons that were used to

858 generate the parameters within the land or canopy subgroups. Each photon that is

859 identified as being likely signal will be classified as: 0 = noise, 1 = ground, 2 = canopy,

860 or 3 = top of canopy. The index values for each classified photon will be provided such

861 that they can be extracted from the ATL03 data product for independent evaluation.

862 Table 2.3. Summary table for photon parameters for the ATL08 product.

Group	Data Type	Description	Source
classed_PC_indx	Float	Indices of photons tracking back to ATL03 that surface finding software identified and used within the creation of the data products.	ATL03
classed_PC_flag	Integer	Classification flag for each photon as either noise, ground, canopy, or top of canopy.	computed
ph_segment_id	Integer	Georeferenced bin number (20-m) associated with each photon	ATL03
d_flag	Integer	Flag indicating whether DRAGANN labeled the photon as noise or signal	computed

863

864 **2.3.1** Indices_of_classed_photons

865 (parameter = classed_PC_indx). Indices of photons tracking back to ATL03 that
866 surface finding software identified and used within the creation of the data products
867 for a given segment.

868 **2.3.2** Photon_class

869 (parameter = classed_PC_flag). Classification flags for a given segment. 0 =
870 noise, 1 = ground, 2 = canopy, 3 = top of canopy. The final ground and canopy
871 classification are flags 1-3. The full canopy is the combination of flags 2 and 3.

872 **2.3.3** Georeferenced_segment_number

873 (parameter = ph_segment_id). The segment_id associated with every photon in
874 each 100-m segment. Each 100-m segment consists of five sequential 20-m segments
875 provided from the ATL03 product, which are labeled as segment_id. The segment_id
876 is a seven digit number that uniquely identifies each along track segment, and is
877 written at the along-track geolocation segment rate (i.e. ~20m along track). The four
878 digit RGT number can be combined with the seven digit segment_id number to
879 uniquely define any along-track segment number. Values are sequential, with
880 0000001 referring to the first segment after the equatorial crossing of the ascending
881 node.

882 **2.3.4** DRAGANN_flag

883 (parameter = d_flag). Flag indicating the labeling of DRAGANN noise filtering for
884 a given photon. 0 = noise, 1=signal.

885

886 **2.4 Subgroup: Reference data**

887 The reference data subgroup contains parameters and information that are
888 useful for determining the terrain and canopy heights that are reported on the

**ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002**

889 product. In addition to position and timing information, these parameters include the
890 reference DEM height, reference landcover type, and flags indicating water or snow.

891 Table 2.4. Summary table for reference parameters for the ATL08 product.

Group	Data Type	Description	Source
segment_id_beg	Integer	First along-track segment_id number in 100-m segment	ATL03
segment_id_end	Integer	Last along-track segment_id number in 100-m segment	ATL03
latitude	Float	Center latitude of signal photons within each segment	ATL03
longitude	Float	Center longitude of signal photons within each segment	ATL03
delta_time	Float	Mid-segment GPS time in seconds past an epoch. The epoch is provided in the metadata at the file level	ATL03
delta_time_beg	Float	Delta time of the first photon in the segment	ATL03
delta_time_end	Float	Delta time of the last photon in the segment	ATL03
night_flag	Integer	Flag indicating whether the measurements were acquired during night time conditions	computed
dem_h	Float4	Reference DEM elevation	external
dem_flag		Source of reference DEM	external
dem_removal_flag	Integer	Quality check flag to indicate > 20% photons removed due to large distance from dem_h	computed
h_dif_ref	Float4	Difference between h_te_median and dem_h	computed
terrain_flg	Integer	Terrain flag quality check to indicate a deviation from the reference DTM	computed
segment_landcover	Integer4	Reference landcover for segment derived from best global landcover product available	external
segment_watermask	Integer4	Water mask indicating inland water produced from best sources available	external

**ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002**

segment_snowcover	Integer4	Daily snow cover mask derived from best sources	external
urban_flag	Integer	Flag indicating segment is located in an urban area	external
surf_type	Integer1	Flags describing surface types: 0=not type, 1=is type. Order of array is land, ocean, sea ice, land ice, inland water.	ATL03
atl08_region	Integer	ATL08 region(s) encompassed by ATL03 granule being processed	computed
last_seg_extend	Float	The distance (km) that the last ATL08 processing segment in a file is either extended or overlapped with the previous ATL08 processing segment	computed

892

893 **2.4.1 Georeferenced_segment_number_beg**

894 (parameter = segment_id_beg). The first along-track segment_id in each 100-m
 895 segment. Each 100-m segment consists of five sequential 20-m segments provided
 896 from the ATL03 product, which are labeled as segment_id. The segment_id is a seven
 897 digit number that uniquely identifies each along track segment, and is written at the
 898 along-track geolocation segment rate (i.e. ~20m along track). The four digit RGT
 899 number can be combined with the seven digit segment_id number to uniquely define
 900 any along-track segment number. Values are sequential, with 0000001 referring to
 901 the first segment after the equatorial crossing of the ascending node.

902 **2.4.2 Georeferenced_segment_number_end**

903 (parameter = segment_id_end). The last along-track segment_id in each 100-m
 904 segment. Each 100-m segment consists of five sequential 20-m segments provided
 905 from the ATL03 product, which are labeled as segment_id. The segment_id is a seven
 906 digit number that uniquely identifies each along track segment, and is written at the
 907 along-track geolocation segment rate (i.e. ~20m along track). The four digit RGT

908 number can be combined with the seven digit segment_id number to uniquely define
909 any along-track segment number. Values are sequential, with 0000001 referring to
910 the first segment after the equatorial crossing of the ascending node.

911 **2.4.3** Segment_latitude

912 (parameter = latitude). Center latitude of signal photons within each segment

913 **2.4.4** Segment_longitude

914 (parameter = longitude). Center longitude of signal photons within each
915 segment

916 **2.4.5** Delta_time

917 (parameter = delta_time). Mid-segment GPS time for the segment in seconds
918 past an epoch. The epoch is listed in the metadata at the file level.

919 **2.4.6** Delta_time_beg

920 (parameter = delta_time_beg). Delta time for the first photon in the segment
921 in seconds past an epoch. The epoch is listed in the metadata at the file level.

922 **2.4.7** Delta_time_end

923 (parameter = delta_time_end). Delta time for the last photon in the segment
924 in seconds past an epoch. The epoch is listed in the metadata at the file level.

925 **2.4.8** Night_Flag

926 (parameter = night_flag). Flag indicating the data were acquired in night
927 conditions: 0 = day, 1 = night. Night flag is set when solar elevation is below 0.0
928 degrees.

929 **2.4.9** Segment_reference_DTM

930 (parameter = dem_h). Reference terrain height value for segment determined
931 by the “best” DEM available based on data location. All heights in ICESat-2 are
932 referenced to the WGS 84 ellipsoid unless clearly noted otherwise. DEM is taken from
933 a variety of ancillary data sources: GIMP, GMTED, MSS. The DEM source flag indicates
934 which source was used.

935 **2.4.10** Segment_reference_DEM_source

936 (parameter = dem_flag). Indicates source of the reference DEM height. Values:
937 0=None, 1=GIMP, 2=GMTED, 3=MSS.

938 **2.4.11** Segment_reference_DEM_removal_flag

939 (parameter = dem_removal_flag). Quality check flag to indicate > 20%
940 classified photons removed from land segment due to large distance from dem_h.

941 **2.4.12** Segment_terrain_difference

942 (parameter = h_dif_ref). Difference between h_te_median and dem_h. Since the
943 mean terrain height is more sensitive to outliers, the median terrain height will be
944 evaluated against the reference DEM. This parameter will be used as an internal data
945 quality check with the notion being that if the difference exceeds a threshold (TBD) a
946 terrain quality flag (terrain_flg) will be triggered.

947 **2.4.13** Segment_terrain flag

948 (parameter = terrain_flg). Terrain flag to indicate confidence in the derived
949 terrain height estimate. If h_dif_ref exceeds a threshold (TBD) the terrain_flg
950 parameter will be set to 1. Otherwise, it is 0.

951 **2.4.14** Segment_landcover

952 (parameter = segment_landcover). Segment landcover will be based on best
953 available global landcover product used for reference. One potential source is the 0.5

954 km global mosaics of the standard MODIS land cover product (Channan et al, 2015;
955 Friedl et al, 2010; available online at <http://glcf.umd.edu/data/lc/index.shtml>). Here,
956 17 classes are defined ranging from evergreen (needle and broadleaf forest),
957 deciduous (needle and broadleaf forest), shrublands, woodlands, savanna and
958 grasslands, agriculture, to urban. The most current year processed for this product is
959 based on MODIS measurements from 2012.

960 **2.4.15** Segment_watermask

961 (parameter = segment_watermask). Water mask (i.e., flag) indicating inland
962 water as referenced from the Global Raster Water Mask at 250 m spatial resolution
963 (Carroll et al, 2009; available online at <http://glcf.umd.edu/data/watermask/>). 0 =
964 no water; 1 = water.

965 **2.4.16** Segment_snowcover

966 (parameter = segment_snowcover). Daily snowcover mask (i.e., flag)
967 indicating a likely presence of snow or ice within each segment produced from best
968 available source used for reference. The snow mask will be the same snow mask as
969 used for ATL09 Atmospheric Products: NOAA snow-ice flag. 0=ice free water;
970 1=snow free land; 2=snow; 3=ice.

971 **2.4.17** Urban_flag

972 (parameter = urban_flag). The urban flag indicates that a segment is likely
973 located over an urban area. In these areas, buildings may be misclassified as canopy,
974 and thus the canopy products may be incorrect. The urban flag is sourced from the
975 “urban and built up” classification on the MODIS land cover product (Channan et al,
976 2015; Friedl et al, 2010; available online at
977 <http://glcf.umd.edu/data/lc/index.shtml>). 0 = not urban; 1 = urban.

978 **2.4.18** Surface_type

979 (parameter = surf_type). The surface type for a given segment is determined at
980 the major frame rate (every 200 shots, or ~140 meters along-track) and is a two-
981 dimensional array surf_type(n, nsurf), where n is the major frame number, and nsurf
982 is the number of possible surface types such that surf_type(n, isurf) is set to 0 or 1
983 indicating if surface type isurf is present (1) or not (0), where isurf = 1 to 5 (land,
984 ocean, sea ice, land ice, and inland water) respectively.

985 **2.4.19** ATL08_region

986 (parameter = atl08_region). The ATL08 regions that encompass the ATL03
987 granule being processed through the ATL08 algorithm. The ATL08 regions are shown
988 by Figure 2.3. In ATL08 regions 11 (Greenland) and 7 - 10 (Antarctica), the
989 canopy_flag is automatically set to false for ATL08 processing.

990 **2.4.20** Last_segment_extend

991 (parameter = last_seg_extend). The distance (km) that the last ATL08 10 km
992 processing segment is either extended beyond 10 km or uses data from the previous
993 10 km processing segment to allow for enough data for processing the ATL03 photons
994 through the ATL08 algorithm. If the last portion of an ATL03 granule being processed
995 would result in a segment with less than 3.4 km (170 geosegments) worth of data,
996 that last portion is added to the previous 10 km processing window to be processed
997 together as one extended ATL08 processing segment. The resulting last_seg_extend
998 value would be a positive value of distance beyond 10 km that the ATL08 processing
999 segment was extended by. If the last ATL08 processing segment would be less than
1000 10 km but greater than 3.4 km, a portion extending from the start of current ATL08
1001 processing segment backwards into the previous ATL08 processing segment would
1002 be added to the current ATL08 processing segment to make it 10 km in length. The
1003 distance of this backward data gathering would be reported in last_seg_extend as a
1004 negative distance value. Only new 100 m ATL08 segment products generated from

1005 this backward extension would be reported. All other segments that are not extended
1006 will report a last_seg_extend value of 0.

1007

1008 **2.5 Subgroup: Beam data**

1009 The subgroup for beam data contains basic information on the geometry and
1010 pointing accuracy for each beam.

1011 Table 2.5. Summary table for beam parameters for the ATL08 product.

Group	Data Type	Units	Description	Source
segment_id_beg	Integer		First along-track segment_id number in 100-m segment	ATL03
segment_id_end	Integer		Last along-track segment_id number in 100-m segment	ATL03
ref_elev	Float		Elevation of the unit pointing vector for the reference photon in the local ENU frame in radians. The angle is measured from East-North plane and positive towards up	ATL03
ref_azimuth	Float		Azimuth of the unit pointing vector for the reference photon in the ENU frame in radians. The angle is measured from North and positive toward East.	ATL03
atlas_pa	Float		Off nadir pointing angle of the spacecraft	ATL03
rgt	Integer		The reference ground track (RGT) is the track on the earth at which the vector bisecting laser beams 3 and 4 is	ATL03

**ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002**

sigma_h	Float	pointed during repeat operations Total vertical uncertainty due to PPD and POD	ATL03
sigma_along	Float	Total along-track uncertainty due to PPD and POD knowledge	ATL03
sigma_across	Float	Total cross-track uncertainty due to PPD and POD knowledge	ATL03
sigma_topo	Float	Uncertainty of the geolocation knowledge due to local topography (Equation 1.3)	computed
sigma_atlas_land	Float	Total uncertainty that includes sigma_h plus the geolocation uncertainty due to local slope Equation 1.2	computed
psf_flag	integer	Flag indicating sigma_atlas_land (aka PSF) as computed in Equation 1.2 exceeds a value of 1m.	computed
layer_flag	Integer	Cloud flag indicating presence of clouds or blowing snow	ATL09
cloud_flag_asr	Integer	Cloud confidence flag from ATL09 indicating clear skies	ATL09
msw_flag	Integer	Multiple scattering warning product produced on ATL09	ATL09
asr	Float	Apparent surface reflectance	ATL09
snr	Float	Background signal to noise level	Computed
solar_azimuth	Float	The azimuth (in degrees) of the sun position vector from the reference photon bounce point position in the local ENU frame. The angle is measured from	ATL03g

solar_elevation	Float	North and is positive towards East. The elevation of the sun position vector from the reference photon bounce point position in the local ENU frame. The angle is measured from the East-North plane and is positive Up.	ATL03g
n_seg_ph	Integer	Number of photons within each land segment	computed
ph_ndx_beg	Integer	Photon index begin	computed

1012

1013 **2.5.1** Georeferenced_segment_number_beg

1014 (parameter = segment_id_beg). The first along-track segment_id in each 100-m
 1015 segment. Each 100-m segment consists of five sequential 20-m segments provided
 1016 from the ATL03 product, which are labeled as segment_id. The segment_id is a seven
 1017 digit number that uniquely identifies each along track segment, and is written at the
 1018 along-track geolocation segment rate (i.e. ~20m along track). The four digit RGT
 1019 number can be combined with the seven digit segment_id number to uniquely define
 1020 any along-track segment number. Values are sequential, with 0000001 referring to
 1021 the first segment after the equatorial crossing of the ascending node.

1022 **2.5.2** Georeferenced_segment_number_end

1023 (parameter = segment_id_end). The last along-track segment_id in each 100-m
 1024 segment. Each 100-m segment consists of five sequential 20-m segments provided
 1025 from the ATL03 product, which are labeled as segment_id. The segment_id is a seven
 1026 digit number that uniquely identifies each along track segment, and is written at the
 1027 along-track geolocation segment rate (i.e. ~20m along track). The four digit RGT
 1028 number can be combined with the seven digit segment_id number to uniquely define

1029 any along-track segment number. Values are sequential, with 0000001 referring to
1030 the first segment after the equatorial crossing of the ascending node.

1031 **2.5.3 Beam_coelevation**

1032 (parameter = ref_elev). Elevation of the unit pointing vector for the reference
1033 photon in the local ENU frame in radians. The angle is measured from East-North
1034 plane and positive towards up.

1035 **2.5.4 Beam_azimuth**

1036 (parameter = ref_azimuth). Azimuth of the unit pointing vector for the
1037 reference photon in the ENU frame in radians. The angle is measured from North and
1038 positive toward East.

1039 **2.5.5 ATLAS_Pointing_Angle**

1040 (parameter = atlas_pa). Off nadir pointing angle (in radians) of the satellite to
1041 increase spatial sampling in the non-polar regions.

1042 **2.5.6 Reference_ground_track**

1043 (parameter = rgt). The reference ground track (RGT) is the track on the earth
1044 at which the vector bisecting laser beams 3 and 4 (or GT2L and GT2R) is pointed
1045 during repeat operations. Each RGT spans the part of an orbit between two ascending
1046 equator crossings and are numbered sequentially. The ICESat-2 mission has 1387
1047 RGTs, numbered from 0001xx to 1387xx. The last two digits refer to the cycle number.

1048 **2.5.7 Sigma_h**

1049 (parameter = sigma_h). Total vertical uncertainty due to PPD (Precise Pointing
1050 Determination), POD (Precise Orbit Determination), and geolocation errors.
1051 Specifically, this parameter includes radial orbit error, σ_{orbit} , tropospheric errors,
1052 σ_{Trop} , forward scattering errors, $\sigma_{forwardscattering}$, instrument timing errors, σ_{timing} ,
1053 and off-nadir pointing geolocation errors. The component parameters are pulled

1054 from ATL03 and ATL09. Sigma_h is the root sum of squares of these terms as detailed
1055 in Equation 1.1. The sigma_h reported here is the mean of the sigma_h values reported
1056 within the five ATL03 geosegments that are used to create the 100 m ATL08 segment.

1057 **2.5.8 Sigma_along**

1058 (parameter = sigma_along). Total along-track uncertainty due to PPD and POD
1059 knowledge. This parameter is pulled from ATL03.

1060 **2.5.9 Sigma_across**

1061 (parameter = sigma_across). Total cross-track uncertainty due to PPD and
1062 POD knowledge. This parameter is pulled from ATL03.

1063 **2.5.10 Sigma_topo**

1064 (parameter = sigma_topo). Uncertainty in the geolocation due to local surface
1065 slope as described in Equation 1.3. The local slope is multiplied by the 6.5 m
1066 geolocation uncertainty factor that will be used to determine the geolocation
1067 uncertainty. The geolocation error will be computed from a 100 m sample due to the
1068 local slope calculation at that scale.

1069 **2.5.11 Sigma_ATLAS_LAND**

1070 (parameter = sigma_atlas_land). Total vertical geolocation error due to
1071 ranging, and local surface slope. The parameter is computed for ATL08 as described
1072 in Equation 1.2. The geolocation error will be computed from a 100 m sample due to
1073 the local slope calculation at that scale.

1074 **2.5.12 PSF_flag**

1075 (parameter = psf_flag). Flag indicating that the point spread function
1076 (computed as sigma_atlas_land) has exceeded 1m.

1077 **2.5.13 Layer_flag**

1078 (parameter = layer_flag). Flag is a combination of multiple ATL09 flags and
1079 takes daytime/nighttime into consideration. A value of 1 means clouds or blowing
1080 snow is likely present. A value of 0 indicates the likely absence of clouds or blowing
1081 snow. If no ATL09 product is available for an ATL08 segment, an invalid value will be
1082 reported.

1083 **2.5.14 Cloud_flag**

1084 (parameter = cloud_flag_asr). Cloud confidence flag from ATL09. Flag indicates
1085 potential clear skies from ATL09. If no ATL09 product is available for an ATL08
1086 segment, an invalid value will be reported. Cloud flags:

- 1087 0 = High confidence clear skies
- 1088 1 = Medium confidence clear skies
- 1089 2 = Low confidence clear skies
- 1090 3 = Low confidence cloudy skies
- 1091 4 = Medium confidence cloudy skies
- 1092 5 = High confidence cloudy skies

1093 **2.5.15 MSW**

1094 (parameter = msw_flag). Multiple scattering warning flag with values from -1 to
1095 5 as computed in the ATL09 atmospheric processing and delivered on the ATL09 data
1096 product. If no ATL09 product is available for an ATL08 segment, an invalid value will
1097 be reported. MSW flags:

- 1098 -1 = signal to noise ratio too low to determine presence of
- 1099 cloud or blowing snow
- 1100 0 = no_scattering
- 1101 1 = clouds at > 3 km
- 1102 2 = clouds at 1-3 km
- 1103 3 = clouds at < 1 km

1104 4 = blowing snow at < 0.5 optical depth

1105 5 = blowing snow at >= 0.5 optical depth

1106 **2.5.16** Computed_Apparent_Surface_Reflectance

1107 (parameter = asr). Apparent surface reflectance computed in the ATL09
1108 atmospheric processing and delivered on the ATL09 data product. If no ATL09
1109 product is available for an ATL08 segment, an invalid value will be reported.

1110 **2.5.17** Signal_to_Noise_Ratio

1111 (parameter = snr). The Signal to Noise Ratio of geolocated photons as
1112 determined by the ratio of the superset of ATL03 signal and DRAGANN found signal
1113 photons used for processing the ATL08 segments to the background photons (i.e.,
1114 noise) within the same ATL08 segments.

1115 **2.5.18** Solar_Azimuth

1116 (parameter = solar_azimuth). The azimuth (in degrees) of the sun position
1117 vector from the reference photon bounce point position in the local ENU frame. The
1118 angle is measured from North and is positive towards East.

1119 **2.5.19** Solar_Elevation

1120 (parameter = solar_elevation). The elevation of the sun position vector from
1121 the reference photon bounce point position in the local ENU frame. The angle is
1122 measured from the East-North plane and is positive up.

1123 **2.5.20** Number_of_segment_photons

1124 (parameter = n_seg_ph). Number of photons in each land segment.

1125 **2.5.21** Photon_Index_Begin

1126 (parameter = ph_ndx_beg). Index (1-based) within the photon-rate data of
1127 the first photon within this each land segment.

***ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002***

1128

1129

1130 **3 ALGORITHM METHODOLOGY**

1131 For the ecosystem community, identification of the ground and canopy surface
1132 is by far the most critical task, as meeting the science objective of determining global
1133 canopy heights hinges upon the ability to detect both the canopy surface and the
1134 underlying topography. Since a space-based photon counting laser mapping system
1135 is a relatively new instrument technology for mapping the Earth's surface, the
1136 software to accurately identify and extract both the canopy surface and ground
1137 surface is described here. The methodology adopted for ATL08 establishes a
1138 framework to potentially accept multiple approaches for capturing both the upper
1139 and lower surface of signal photons. One method used is an iterative filtering of
1140 photons in the along-track direction. This method has been found to preserve the
1141 topography and capture canopy photons, while rejecting noise photons. An advantage
1142 of this methodology is that it is self-parameterizing, robust, and works in all
1143 ecosystems if sufficient photons from both the canopy and ground are available. For
1144 processing purposes, along-track data signal photons are parsed into L -km segment
1145 of the orbit which is recommended to be 10 km in length.

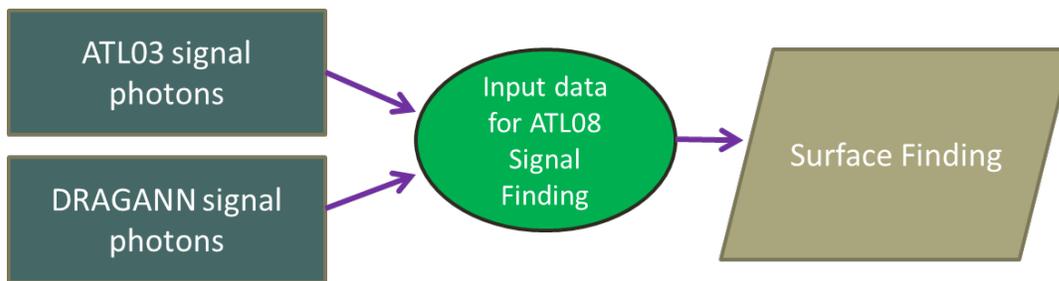
1146

1147 **3.1 Noise Filtering**

1148 Solar background noise is a significant challenge in the analysis of photon
1149 counting laser data. Range measurement data created from photon counting lidar
1150 detectors typically contain far higher noise levels than the more common photon
1151 integrating detectors available commercially in the presence of passive, solar
1152 background photons. Given the higher detection sensitivity for photon counting
1153 devices, a background photon has a greater probability of triggering a detection event
1154 over traditional integral measurements and may sometimes dominate the dataset.
1155 Solar background noise is a function of the surface reflectance, topography, solar
1156 elevation, and atmospheric conditions. Prior to running the surface finding

1157 algorithms used for ATL08 data products, the superset of output from the GSFC
1158 medium-high confidence classed photons (ATL03 signal_conf_ph: flags 3-4) and the
1159 output from DRAGANN will be considered as the input data set. ATL03 input data
1160 requirements include the latitude, longitude, height, segment delta time, segment ID,
1161 and a preliminary signal classification for each photon. The motivation behind
1162 combining the results from two different noise filtering methods is to ensure that all
1163 of the potential signal photons for land surfaces will be provided as input to the
1164 surface finding software. The description of the methodology for the ATL03
1165 classification is described separately in the ATL03 ATBD. The methodology behind
1166 DRAGANN is described in the following section.

1167



1168

1169 Figure 3.1. Combination of noise filtering algorithms to create a superset of input data for
1170 surface finding algorithms.

1171

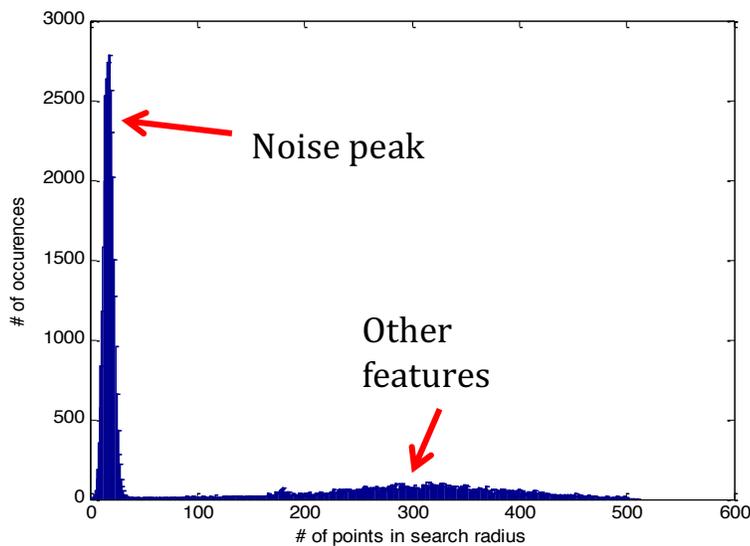
1172 **3.1.1 DRAGANN**

1173 The Differential, Regressive, and Gaussian Adaptive Nearest Neighbor
1174 (DRAGANN) filtering technique was developed to identify and remove noise photons
1175 from the photon counting data point cloud. DRAGANN utilizes the basic premise that
1176 signal photons will be closer in space than random noise photons. The first step of the
1177 filtering is to implement an adaptive nearest neighbor search. By using an adaptive
1178 method, different thresholds can be applied to account for variable amounts of

1206

1207 where r is the radius. A good practice is to first normalize the data set along each
1208 dimension before running the DRAGANN filter. Normalization prevents the algorithm
1209 from favoring one dimension over the others in the radius search (e.g., when the
1210 latitude and longitude are in degrees and height is in meters).

1211



1212

1213 Figure 3.2. Histogram of the number of photons within a search radius. This histogram is
1214 used to determine the threshold for the DRAGANN approach.

1215

1216 Once the radius has been computed, DRAGANN counts the number of points
1217 within the radius for each point and histograms that set of values. The distribution of
1218 the number of points, Figure 3.2, reveals two distinct peaks; a noise peak and a signal
1219 peak. The motivation of DRAGANN is to isolate the signal photons by determining a
1220 threshold based on the number of photons within the search radius. The noise peak
1221 is characterized as having a large number of occurrences of photons with just a few
1222 neighboring photons within the search radius. The signal photons comprise the broad
1223 second peak. The first step in determining the threshold between the noise and signal

1224 is to implement Gaussian fitting to the number of photons distribution (i.e., the
1225 distribution shown in Figure 3.2). The Gaussian function has the form

1226

$$1227 \quad g(x) = ae^{-\frac{(x-b)^2}{2c^2}} \quad \text{Eqn. 3.3}$$

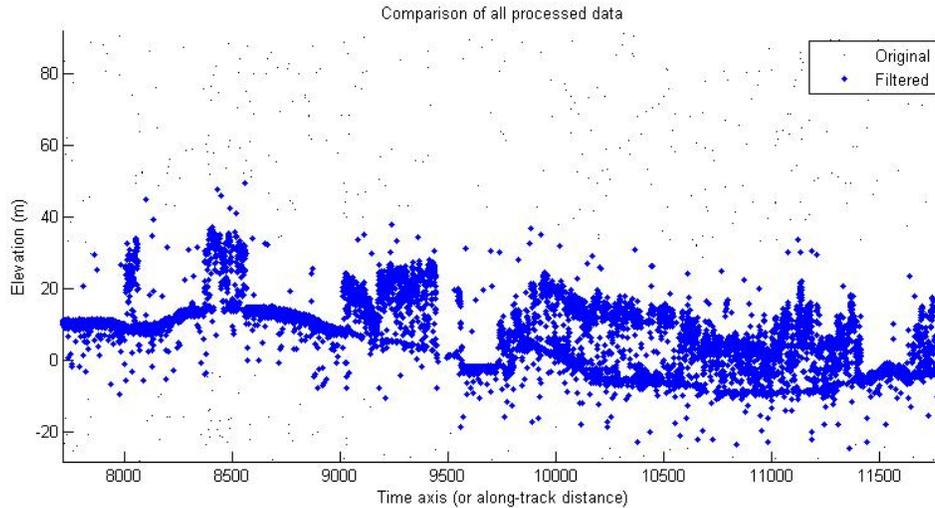
1228

1229 where a is the amplitude of the peak, b is the center of the peak, and c is the standard
1230 deviation of the curve. A first derivative sign crossing method is one option to identify
1231 peaks within the distribution.

1232 To determine the noise and signal Gaussians, up to ten Gaussian curves are fit
1233 to the histogram using an iterative process of fitting and subtracting the max-
1234 amplitude peak component from the histogram until all peaks have been extracted.
1235 Then, the potential Gaussians pass through a rejection process to eliminate those with
1236 poor statistical fits or other apparent errors (Goshtasby and O'Neill, 1994; Chauve et
1237 al. 2008). A Gaussian with an amplitude less than 1/5 of the previous Gaussian and
1238 within two standard deviations of the previous Gaussian should be rejected. Once the
1239 errant Gaussians are rejected, the final two remaining are assumed to represent the
1240 noise and signal. These are separated based on the remaining two Gaussian
1241 components within the histogram using the logic that the leftmost Gaussian is noise
1242 (low neighbor counts) and the other is signal (high neighbor counts).

1243 The intersection of these two Gaussians (noise and signal) determines a data
1244 threshold value. The threshold value is the parameter used to distinguish between
1245 noise points and signal points when the point cloud is re-evaluated for surface finding.
1246 In the event that only one curve passes the rejection process, the threshold is set at
1247 1σ above the center of the noise peak.

1248 An example of the noise filtered product from DRAGANN is shown in Figure
1249 3.3. The signal photons identified in this process will be combined with the coarse
1250 signal finding output available on the ATL03 data product.



1251

1252 Figure 3.3. Output from DRAGANN filtering. Signal photons are shown as blue.

1253 Figure 3.3 provides an example of along-track (profiling) height data collected
1254 in September 2012 from the MABEL (ICESat-2 simulator) over vegetation in North
1255 Carolina. The photons have been filtered such that the signal photons returned from
1256 vegetation and the ground surface are remaining. Noise photons that are adjacent to
1257 the signal photons are also retained in the input dataset; however, these should be
1258 classified as noise photons during the surface finding process. It is possible that some
1259 additional outlying noise may be retained during the DRAGANN process when noise
1260 photons are densely grouped, and these photons should be filtered out before the
1261 surface finding process. Estimates of the ground surface and canopy height can then
1262 be derived from the signal photons.

1263

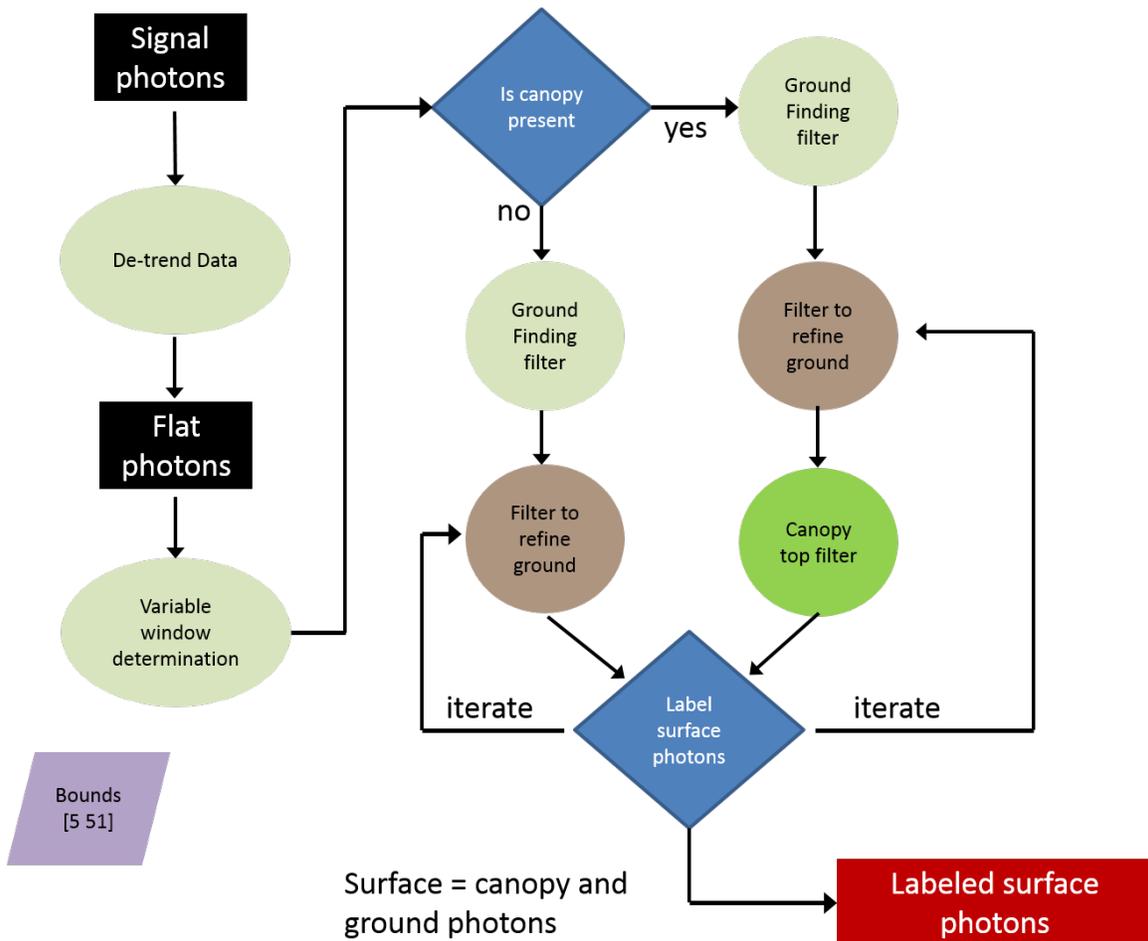
1264 **3.2 Surface Finding**

1265 Once the signal photons have been determined, the objective is to find the
1266 ground and canopy photons from within the point cloud. With the expectation that
1267 one algorithm may not work everywhere for all biomes, we are employing a
1268 framework that will allow us to combine the solutions of multiple algorithms into one

1269 final composite solution for the ground surface. The composite ground surface
 1270 solution will then be utilized to classify the individual photons as ground, canopy, top
 1271 of canopy, or noise. Currently, the framework described here utilizes one algorithm
 1272 for finding the ground surface and canopy surface. Additional methods, however,
 1273 could be integrated into the framework at a later time. Figure 3.4 below describes the
 1274 framework.

1275

1276



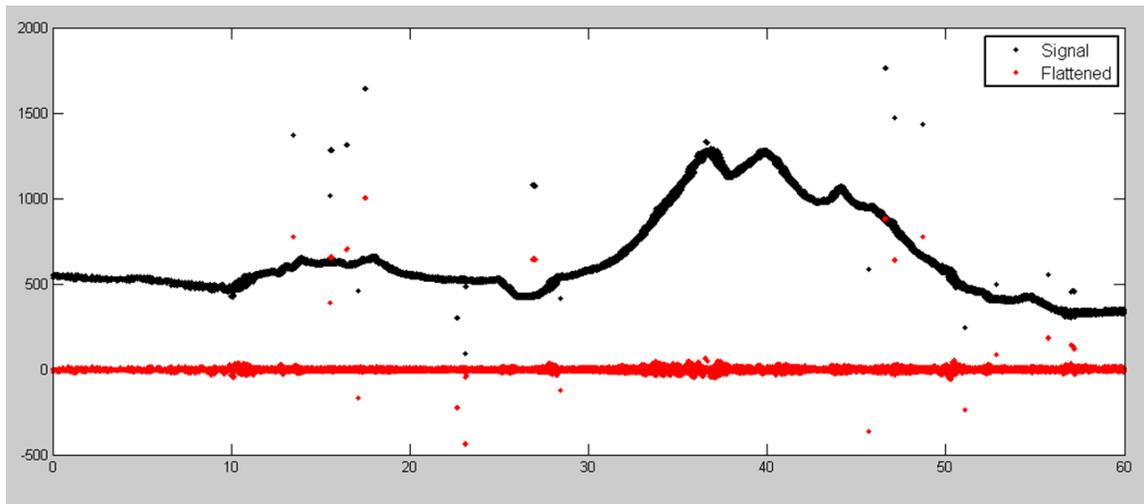
1277

1278 Figure 3.4. Flowchart of overall surface finding method.

1279

1280 3.2.1 De-trending the Signal Photons

1281 An important step in the success of the surface finding algorithm is to remove
1282 the effect of topography on the input data, thus improving the performance of the
1283 algorithm. This is done by de-trending the input signal photons by subtracting a
1284 heavily smoothed “surface” that is derived from the input data. Essentially, this is a
1285 low pass filter of the original data and most of the analysis to detect the canopy and
1286 ground will subsequently be implemented on the high pass data. The amount of
1287 smoothing that is implemented in order to derive this first surface is dependent upon
1288 the relief. For segments where the relief is high, the smoothing window size is
1289 decreased so topography isn’t over-filtered.



1290

1291 Figure 3.5. Plot of Signal Photons (black) from 2014 MABEL flight over Alaska and de-
1292 trended photons (red).

1293

1294 3.2.2 Canopy Determination

1295 A key factor in the success of the surface finding algorithm is for the software
1296 to automatically **account for the presence of canopy** along a given L -km segment.

**ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002**

1297 Due to the large volume of data, this process has to occur in an automated fashion,
1298 allowing the correct methodology for extracting the surface to be applied to the data.
1299 In the absence of canopy, the iterative filtering approach to finding ground works
1300 extremely well, but if canopy does exist, we need to accommodate for that fact when
1301 we are trying to recover the ground surface.

1302 Currently, the Landsat Tree Cover Continuous Fields dataset from the 2000
1303 epoch is used to set a canopy flag within the ATL08 algorithm. Each of these Landsat
1304 Tree Cover tiles contain 30 m pixels indicating the percentage canopy cover for
1305 vegetation over 5 m high in that pixel area. The 2000 epoch is used over the newer
1306 2005 epoch due to “striping” in the 2005 tiles, caused by the failure of the scan line
1307 corrector (SLC) in 2003. The striping artifacts result in inconsistent pixel values
1308 across a landscape which in turn can result in a tenfold difference in the average
1309 canopy cover percentage calculated between the epochs for a flight segment. There is
1310 currently available a 2015 Tree Cover Beta Release that utilizes Landsat 8 data. This
1311 new release of the 2015 Tree Cover product will replace the 2000 epoch for setting
1312 the canopy flag in the ATL08 algorithm. The Tree Cover data are available via ftp at
1313 <http://glcf.umd.edu/data/landsatTreecover/>.

1314 For each *L-km* segment of ATLAS data, a comparison is made between the
1315 midpoint location of the segment and the midpoint locations of the WRS Landsat tiles
1316 to find the closest tile that encompasses the *L-km* segment. Using the closest found
1317 tile, each signal photon’s X-Y location is used to identify the corresponding Landsat
1318 pixel. Multiple instances of the same pixels found for the *L-km* segment are discarded,
1319 and the percentage canopy values of the unique pixels determined to be under the *L-*
1320 *km* segment are averaged to produce an average canopy cover percentage for that
1321 segment. If the average canopy cover percentage for a segment is over 5% (threshold
1322 subject to change under further testing), then the ATL08 algorithm will assume the
1323 presence of canopy and identify both ground and vegetation photons in that

1324 segment's output. Else, the ATL08 algorithm uses a simplified calculation to identify
1325 only ground photons in that segment.

1326 The canopy flag determines if the algorithm will calculate only ground photons
1327 (canopy flag = 0) or both ground and vegetation photons (canopy flag = 1) for each *L-*
1328 *km* segment.

1329 For ATL08 product regions over Antarctica (regions 7, 8, 9, 10) and Greenland
1330 (region 11), the algorithm will assume only ground photons (canopy flag = 0) (see
1331 Figure 2.2).

1332

1333 **3.2.3 Variable Window Determination**

1334 The method for generating a best estimated terrain surface will vary depending
1335 upon whether canopy is present. *L-km segments* without canopy are much easier to
1336 analyze because the ground photons are usually continuous. *L-km segments* with
1337 canopy, however, require more scrutiny as the number of signal photons from ground
1338 are fewer due to occlusion by the vegetation.

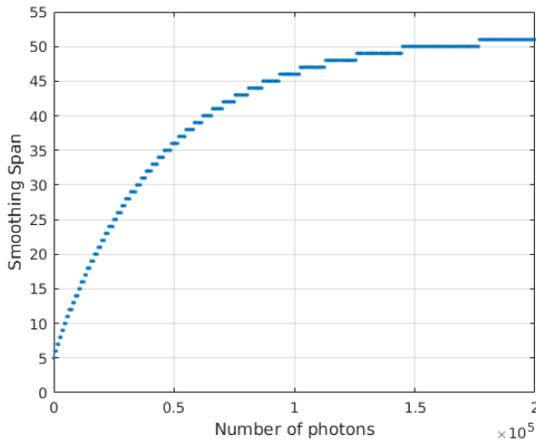
1339 There are some common elements for finding the terrain surface for both cases
1340 (canopy/no canopy) and with both methods. In both cases, we will use a variable
1341 windowing span to compute statistics as well as filter and smooth the data. For
1342 clarification, the window size is variable for each *L-km segment*, but it is constant
1343 within the *L-km segment*. For the surface finding algorithm, we will employ a
1344 Savitzky-Golay smoothing/median filtering method. Using this filter, we compute a
1345 variable smoothing parameter (or window size). It is important to bound the filter
1346 appropriately as the output from the median filter can lose fidelity if the scan is over-
1347 filtered.

1348 We have developed an empirically-determined shape function, bound between
1349 [5 51], that sets the window size (Sspan) based on the number of photons within each
1350 *L-km* segment.

1351
$$Sspan = \text{ceil}[5 + 46 * (1 - e^{-a*length})]$$
 Eqn. 3.4

1352
$$a = \frac{\log\left(1 - \frac{21}{51-5}\right)}{-28114} \approx 21 \times 10^{-6}$$
 Eqn. 3.5

1353 where *a* is the shape parameter and length is the total number of photons in the *L-km*
1354 segment. The shape parameter, *a*, was determined using data collected by MABEL and
1355 is shown in Figure 3.6. It is possible that the model of the shape function, or the
1356 filtering bounds, will need to be adjusted once ICESat-2/ATLAS is on orbit and
1357 collecting data.



1358
1359 Figure 3.6. Shape Parameter for variable window size.

1360

1361 **3.2.4 Compute descriptive statistics**

1362 To help characterize the input data and initialize some of the parameters used
1363 in the algorithm, we employ a moving window to compute descriptive statistics on
1364 the de-trended data. The moving window's width is the smoothing span function

1365 computed in Equation 5 and the window slides $\frac{1}{4}$ of its size to allow of overlap
1366 between windows. By moving the window with a large overlap helps to ensure that
1367 the approximate ground location is returned. The statistics computed for each
1368 window step include:

- 1369 • Mean height
- 1370 • Min height
- 1371 • Max height
- 1372 • Standard deviation of heights

1373

1374 Dependent upon the amount of vegetation within each window, the estimated
1375 ground height is estimated using different statistics. A standard deviation of the
1376 photon elevations computed within each moving window are used to classify the
1377 vertical spread of photons as belonging to one of four classes with increasing amounts
1378 of variation: open, canopy level 1, canopy level 2, canopy level 3. The canopy indices
1379 are defined in Table 3.1.

1380

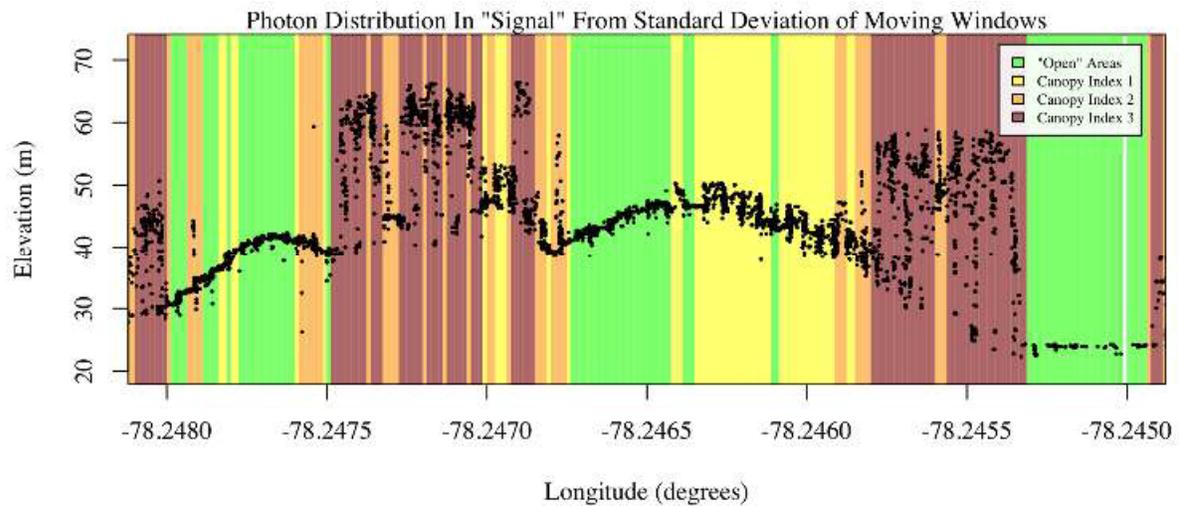
1381 Table 3.1. Standard deviation ranges utilized to qualify the spread of photons within
1382 moving window.

Name	Definition	Lower Limit	Upper Limit
Open	Areas with little or no spread in signal photons determined due to low standard deviation	N/A	Photons falling within 1 st quartile of Standard deviation
Canopy Level 1	Areas with small spread in signal photons	1 st quartile	Median

Canopy Level 2	Areas with a medium amount of spread	Median	3 rd quartile
Canopy Level 3	Areas with high amount of spread in signal photons	3 rd quartile	N/A

1383

1384



1385

1386 Figure 3.7. Illustration of the standard deviations calculated for each moving window to
1387 identify the amount of spread of signal photons within a given window.

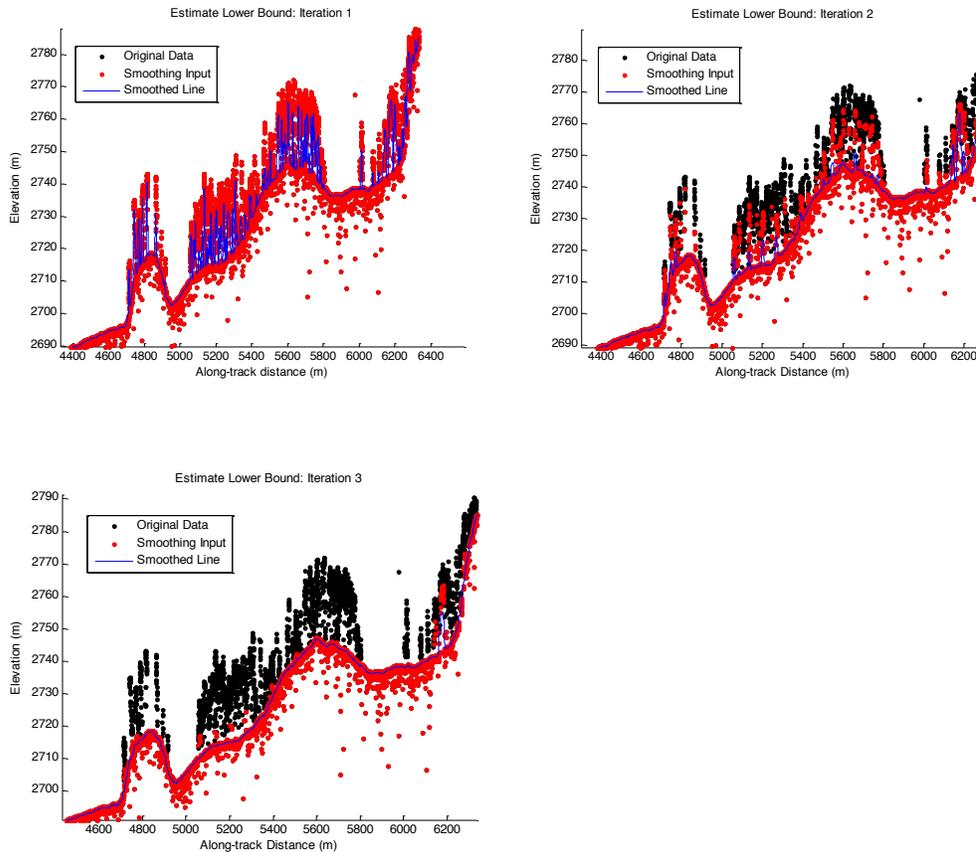
1388

1389 **3.2.5 Ground Finding Filter (Iterative median filtering)**

1390 A combination of an iterative median filtering and smoothing filter approach
1391 will be employed to derive the output solution of both the ground and canopy
1392 surfaces. The input to this process is the set of de-trended photons. Finding the
1393 ground in the presence of canopy often poses a challenge because often there are
1394 fewer ground photons underneath the canopy. The algorithm adopted here uses an
1395 iterative median filtering approach to retain/eliminate photons for ground finding in

1396 the presence of canopy. When canopy exists, a smoothed line will lay somewhere
1397 between the canopy top and the ground. This fact is used to iteratively label points
1398 above the smoothed line as canopy. The process is repeated five times to eliminate
1399 canopy points that fall above the estimated surface as well as noise points that fall
1400 below the ground surface. An example of iterative median filtering is shown in Figure
1401 3.8. The final median filtered line is the preliminary surface estimate. A limitation of
1402 this approach, however, is in cases of dense vegetation and few photons reaching the
1403 ground surface. In these instances, the output of the median filter may lie within the
1404 canopy.

1405



1406

1407

1408

1409 Figure 3.8. Three iterations of the ground finding concept for L -km segments with canopy.

1410

1411 **3.3 Top of Canopy Finding Filter**

1412 Finding the top of the canopy surface uses the same methodology as finding
1413 the ground surface, except now the de-trended data are “flipped” over. The “flip”
1414 occurs by multiplying the photons heights by -1 and adding the mean of all the heights
1415 back to the data. The same procedure used to find the ground surface can be used to
1416 find the indices of the top of canopy points.

1417

1418 **3.4 Classifying the Photons**

1419 Once a composite ground surface is determined, photons falling within the
1420 point spread function of the surface are labeled as ground photons. Based on the
1421 expected performance of ATLAS, the point spread function should be approximately
1422 35 cm rms. Signal photons that are not labeled as ground and are below the ground
1423 surface (buffered with the point spread function) are considered noise, but keep the
1424 signal label.

1425 The top of canopy photons that are identified can be used to generate an upper
1426 canopy surface through a shape-preserving surface fitting method. All signal photons
1427 that are not labeled ground and lie above the ground surface (buffered with the point
1428 spread function) and below the upper canopy surface are considered to be canopy
1429 photons (and thus labeled accordingly). Signal photons that lie above the top of
1430 canopy surface are considered noise, but keep the signal label.

1431

1432	FLAGS,	0 = noise
1433		1 = ground
1434		2 = canopy
1435		3 = TOC (top of canopy)

1436

1437 The final ground and canopy classifications are flags 1 – 3. The full canopy is
1438 the combination of flags 2 and 3.

1439

1440 **3.5 Refining the Photon Labels**

1441 During the first iteration of the algorithm, it is possible that some photons are
1442 mislabeled; most likely this would be noise photons mislabeled as canopy. To reject
1443 these mislabeled photons, we apply three criteria:

- 1444 a) If top of canopy photons are 2 standard deviations above a
1445 smoothed median top of canopy surface
- 1446 b) If there are less than 3 canopy indices within a 15m radius
- 1447 c) If, for 500 signal photon segments, the number of canopy photons
1448 is < 5% of the total (when SNR > 1), or < 10% of the total (when SNR
1449 <= 1). This minimum number of canopy indices criterion implies a
1450 minimum amount of canopy cover within a region.

1451 There are also instances where the ground points will be redefined. This
1452 reassigning of ground points is based on how the final ground surface is determined.
1453 Following the “iterate” steps in the flowchart shown in Figure 3.4, if there are no
1454 canopy indices identified for the *L-km* segment, the final ground surface is
1455 interpolated from the identified ground photons and then will undergo a final round
1456 of median filtering and smoothing.

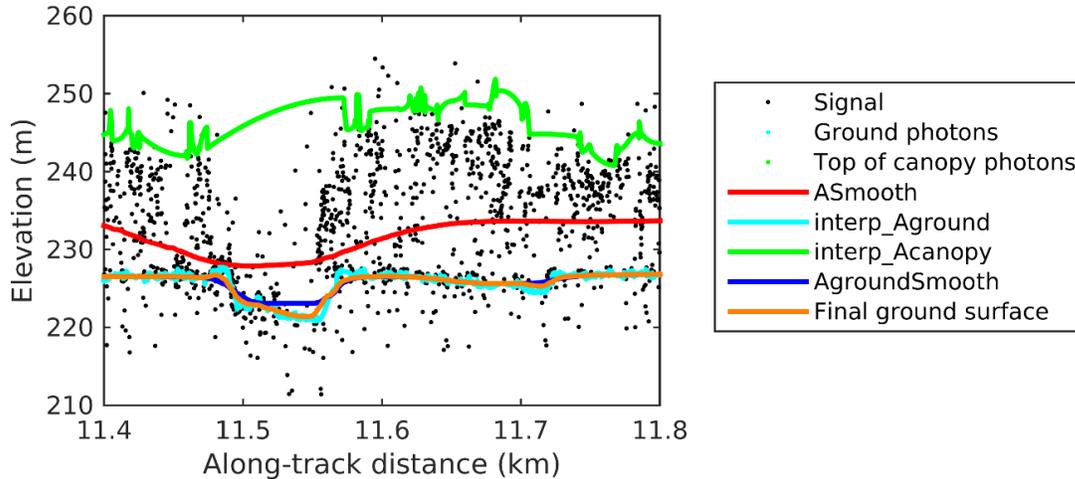
1457 If canopy photons are identified, the final ground surface is interpolated based
1458 upon the level/amount of canopy at that location along the segment. The final ground
1459 surface is a composite of various intermediate ground surfaces, defined thusly:

ASmooth heavily smoothed surface used to de-trend the signal data

Interp_Aground interpolated ground surface based upon the identified ground
photons

AgroundSmooth median filtered and smoothed version of Interp_Aground

1460



1461

1462 Figure 3.9. Example of the intermediate ground and top of canopy surfaces calculated from
1463 MABEL flight data over Alaska during July 2014.

1464

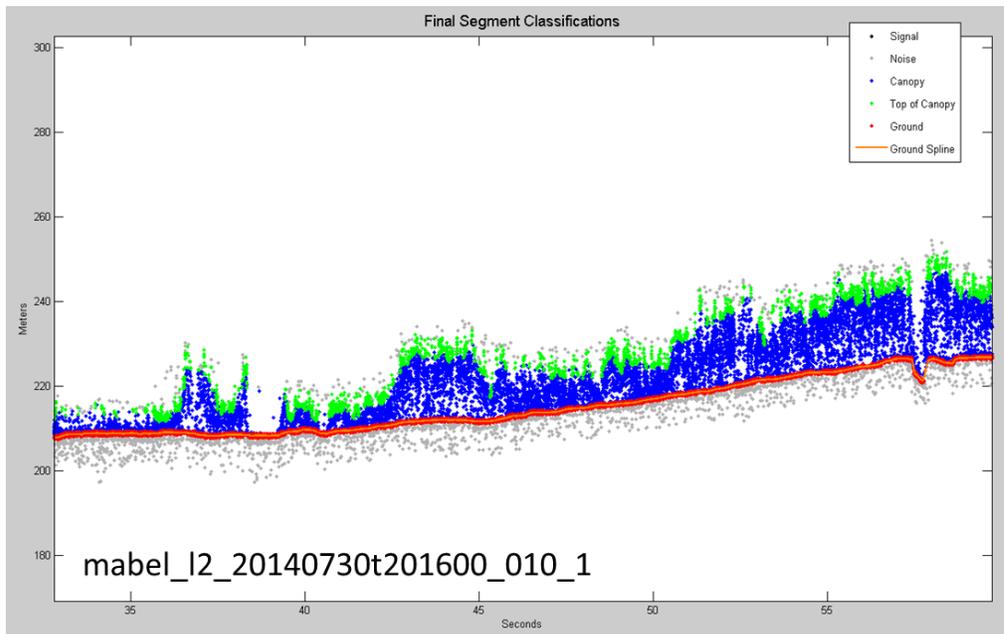
1465 During the first round of ground surface refinement, where there are canopy
1466 photons identified in the segment, the ground surface at that location is defined by
1467 the smoothed ground surface (AgroundSmooth) value. Else, if there is a location
1468 along-track where the standard deviation of the ground-only photons is greater than
1469 the 75% quartile for all signal photon standard deviations (i.e., canopy level 3), then
1470 the ground surface at that location is a weighted average between the interpolated
1471 ground surface (Interp_Aground*1/3) and the smoothed interpolated ground surface
1472 (AgroundSmooth*2/3). For all remaining locations long the segment, the ground
1473 surface is the average of the interpolated ground surface (Interp_Aground) and the
1474 heavily smoothed surface (ASmooth).

1475 The second round of ground surface refinement is simpler than the first.
1476 Where there are canopy photons identified in the segment, the ground surface at that
1477 location is defined by the smoothed ground surface (AgroundSmooth) value again.

1478 For all other locations, the ground surface is defined by the interpolated ground
1479 surface (Interp_Aground). This composite ground surface is run through the median
1480 and smoothing filters again.

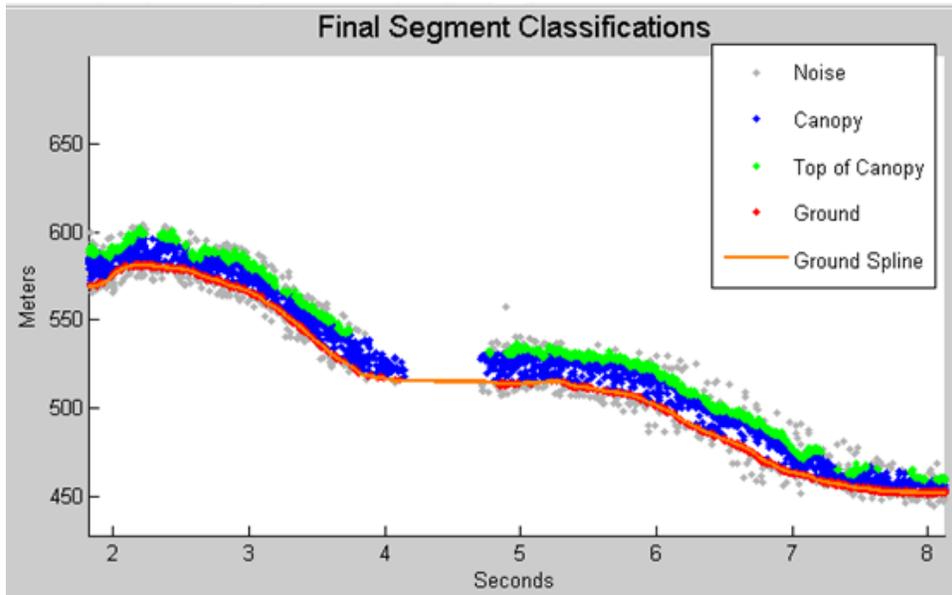
1481 The pseudocode for this surface refining process can be found in section 4.11.

1482 Examples of the ground and canopy photons for several MABEL lines are
1483 shown in Figures 3.10 – 3.12.



1484

1485 Figure 3.10. Example of classified photons from MABEL data collected in Alaska 2014.
1486 Red photons are photons classified as terrain. Green photons are classified as top of canopy.
1487 Canopy photons (shown as blue) are considered as photons lying between the terrain
1488 surface and top of canopy.



1489

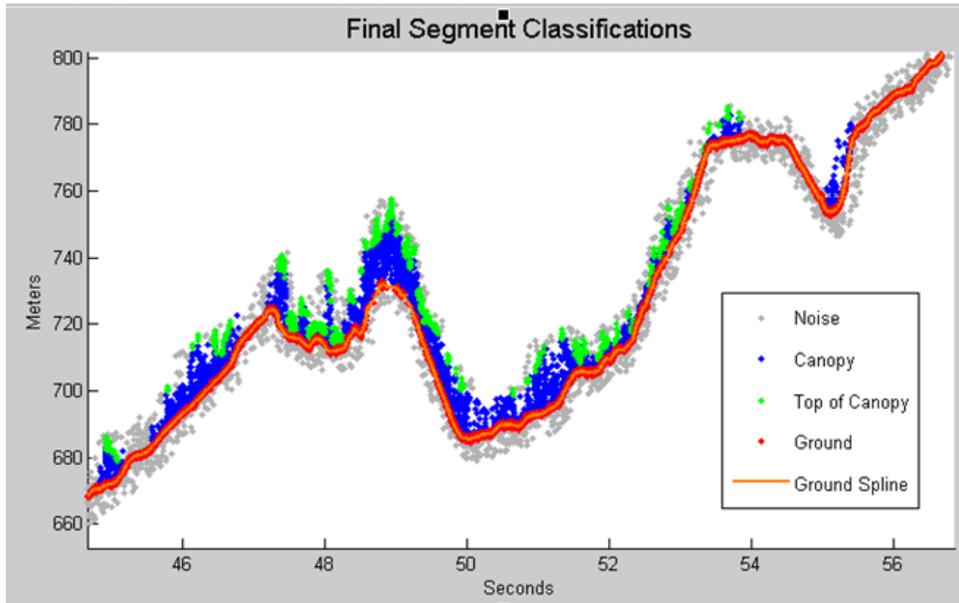
1490 Figure 3.11. Example of classified photons from MABEL data collected in Alaska 2014.

1491 Red photons are photons classified as terrain. Green photons are classified as top of canopy.

1492 Canopy photons (shown as blue) are considered as photons lying between the terrain

1493 surface and top of canopy.

1494



1495

1496 Figure 3.12. Example of classified photons from MABEL data collected in Alaska 2014.
1497 Red photons are photons classified as terrain. Green photons are classified as top of canopy.
1498 Canopy photons (shown as blue) are considered as photons lying between the terrain
1499 surface and top of canopy.

1500

1501 **3.6 Canopy Height Determination**

1502 Once a final ground surface is determined, canopy heights for individual
1503 photons are computed by removing the ground surface height for that photon's
1504 latitude/longitude. These relative canopy height values will be used to compute the
1505 canopy statistics on the ATL08 data product.

1506

1507 **3.7 Link Scale for Data products**

1508 The link scale for each segment within which values for vegetation parameters
1509 will be derived will be defined over a fixed distance of 100 m. A fixed segment length
1510 ensures that canopy and terrain metrics are consistent between segments, in addition

***ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002***

1511 to increased ease of use of the final products. A size of 100 m was selected as it should
1512 provide approximately 140 photons (a statistically sufficient number) from which to
1513 make the calculations for terrain and canopy height.

1514

1515 **4. ALGORITHM IMPLEMENTATION**

1516 Prior to running the surface finding algorithms used for ATL08 data products, the
 1517 superset of output from the GSFC medium-high confidence classed photons (ATL03
 1518 signal_conf_ph: flags 3-4) and the output from DRAGANN will be considered as the input
 1519 data set. ATL03 input data requirements include the along-track time, latitude, longitude,
 1520 height, and classification for each photon. The motivation behind combining the results
 1521 from two different noise filtering methods is to ensure that all of the potential signal
 1522 photons for land surfaces will be provided as input to the surface finding software.

1523 Table 4.1. Input parameters to ATL08 classification algorithm.

Name	Data Type	Long Name	Units	Description	Source
delta_time	DOUBLE	GPS elapsed time	seconds	Elapsed GPS seconds since start of the granule for a given photon. Use the metadata attribute granule_start_seconds to compute full gps time.	ATL03
lat_ph	FLOAT	latitude of photon	degrees	Latitude of each received photon. Computed from the ECEF Cartesian coordinates of the bounce point.	ATL03
lon_ph	FLOAT	longitude of photon	degrees	Longitude of each received photon. Computed from the ECEF Cartesian coordinates of the bounce point.	ATL03
h_ph	FLOAT	height of photon	meters	Height of each received photon, relative to the WGS-84 ellipsoid.	ATL03
sigma_h	FLOAT	height uncertainty	m	Estimated height uncertainty (1-sigma) for the reference photon.	ATL03
signal_conf_ph	UINT_1_LE	photon signal confidence	counts	Confidence level associated with each photon event selected as signal (0-noise, 1- added to allow for buffer but algorithm classifies as background, 2-low, 3-med, 4-high).	ATL03

**ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002**

segment_id	UNIT_32	along-track segment ID number	unitless	A seven-digit number uniquely identifying each along-track segment. These are sequential, starting with one for the first segment after an ascending equatorial crossing node.	ATL03
cab_prof	FLOAT	Calibrated Attenuated Backscatter	unitless	Calibrated Attenuated Backscatter from 20 to -1 km with vertical resolution of 30m	ATL09
dem_h	FLOAT	DEM Height	meters	Best available DEM (in priority of GIMP/ANTARCTIC/GMTED/MS S) value at the geolocation point. Height is in meters above the WGS84 Ellipsoid.	ATL09
Landsat tree cover	UINT_8	Landsat Tree Cover Continuous Fields	percentage	Percentage of woody vegetation greater than 5 meters in height across a 30 meter pixel	Global Land Cover Facility (Sexton, 2013)

1524

1525 Table 4.2. Additional external parameters referenced in ATL08 product.

Name	Data Type	Long Name	Units	Description	Source
atlas_pa				Off nadir pointing angle of the spacecraft	
ground_track				Ground track, as numbered from left to right: 1 = 1L, 2 = 1R, 3 = 2L, 4 = 2R, 5 = 3L, 6 = 3R	
dem_h				Reference DEM height	ANC06
ref_azimuth	FLOAT	azimuth	radians	Azimuth of the unit pointing vector for the reference photon in the local ENU frame in radians. The angle is measured from north and positive towards east.	ATL03
ref_elev	FLOAT	elevation	radians	Elevation of the unit pointing vector for the reference photon in the local ENU frame in radians. The angle is measured from east-	ATL03

**ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002**

				north plane and positive towards up.	
rgt	INTEGER_2	reference ground track	unitless	The reference ground track (RGT) is the track on the Earth at which a specified unit vector within the observatory is pointed. Under nominal operating conditions, there will be no data collected along the RGT, as the RGT is spanned by GT2L and GT2R. During slews or off-pointing, it is possible that ground tracks may intersect the RGT. The ICESat-2 mission has 1,387 RGTs.	ATL03
sigma_along	DOUBLE	along-track geolocation uncertainty	meters	Estimated Cartesian along-track uncertainty (1-sigma) for the reference photon.	ATL03
sigma_across	DOUBLE	across-track geolocation uncertainty	meters	Estimated Cartesian across-track uncertainty (1-sigma) for the reference photon.	ATL03
surf_type	INTEGER_1	surface type	unitless	Flags describing which surface types this interval is associated with. 0=not type, 1=is type. Order of array is land, ocean, sea ice, land ice, inland water.	ATL03 , Section 4
layer_flag	Integer	Consolidated cloud flag	unitless	Flag indicating the presence of clouds or blowing snow with good confidence	ATL09
cloud_flag_asr	Integer(3)	Cloud probability from ASR	unitless	Cloud confidence flag, from 0 to 5, indicating low, med, or high confidence of clear or cloudy sky	ATL09
msw_flag	Byte(3)	Multiple scattering warning flag	unitless	Flag with values from 0 to 5 indicating presence of multiple scattering, which may be due to blowing snow or cloud/aerosol layers.	ATL09
asr	Float(3)	Apparent surface reflectance	unitless	Surface reflectance as modified by atmospheric transmission	ATL09

snow_ice	INTEGER_ 1	Snow Ice Flag	unitless	NOAA snow-ice flag. 0=ice free water; 1=snow free land; 2=snow; 3=ice	ATL09
-----------------	---------------	------------------	----------	---	-------

1526

1527 **4.1 Cloud based filtering**

1528 It is possible for the presence of clouds to affect the number of surface photon
 1529 returns through signal attenuation, or to cause false positive classifications of
 1530 ground or canopy photons on low cloud returns. Either of these cases would reduce
 1531 the accuracy of the ATL08 product. To improve the performance of the ATL08
 1532 algorithm, ideally all clouds would be identified prior to processing through the
 1533 ATL08 algorithm. There will be instances, however, where low lying clouds (e.g.
 1534 <800 m above the ground surface) may be difficult to identify. Currently, ATL08
 1535 provides an ATL09 derived cloud flag (layer_flag) on its 100 m product and
 1536 encourages the user to make note of the presence of clouds when using ATL08
 1537 output. Unfortunately at present, a review of on-orbit data from ATL03 and ATL09
 1538 indicate that the cloud layer flag is not being set correctly in the ATL09 algorithm.
 1539 Ultimately, the final cloud based filtering process used in the ATL08 algorithm will
 1540 most likely be derived from parameters/flag on the ATL09 data product. Until the
 1541 ATL09 cloud flags are proven reliable, however, a preliminary cloud screening
 1542 method is presented below. This methodology utilizes the calibrated attenuated
 1543 backscatter on the ATL09 data product to identify (and subsequently remove for
 1544 processing) clouds or other problematic issues (i.e. incorrectly telemetered
 1545 windows). Using this new method, telemetered windows identified as having either
 1546 low or no surface signal due to the presence of clouds (likely above the telemetered
 1547 band), **as well as photon returns suspected to be clouds instead of surface returns,**
 1548 will be omitted from the ATL08 processing. This process, however, will not identify
 1549 the extremely low clouds (i.e. <800 m). The steps are as follows:

- 1550 1. Match up the ATL09 calibrated attenuated backscatter (cab_prof) columns to
 1551 the ATL03 granule being processed using segment ID.

- 1552 2. Flip the matching cab_prof vertical columns so that the elevation bins go
1553 from low to high.
- 1554 3. For each of the matching ATL09 cab_prof vertical columns, perform a cubic
1555 Savitsky-Golay smoothing filter with a span size of 15 vertical bins. Call this
1556 cab_smooth.
- 1557 4. Perform the same smoothing filter on each horizontal row of the cab_smooth
1558 output, this time using a span size of 7 horizontal bins. Call this
1559 cab_smoother.
- 1560 5. Create a low_signal logical array the length of the number of matching ATL09
1561 columns and set to false.
- 1562 6. For each column of cab_smoother:
- 1563 a. Set any values below 0 to 0.
- 1564 b. Set a logical array of cab_smoother bins that are below 15 km in
1565 elevation to true. Call this cab15.
- 1566 c. Using the ATL09 dem_h value for that column, find the ATL09
1567 cab_smoother bins that are 240 m above and 240 m below (~8 ATL09
1568 vertical bins each direction) the dem_h value. The bins found here that
1569 are also within cab15 are designated as sfc_bins.
- 1570 d. Find the maximum peak value of cab_smoother within the sfc_bins, if
1571 any. This will represent the surface peak.
- 1572 e. Find the maximum value of cab_smoother that is higher in elevation
1573 than the sfc_bins and within cab15, if any. This will represent the
1574 cloud peak.
- 1575 f. If there is no surface peak, set the low_signal flag to true.
- 1576 g. If there are both surface and cloud peak values returned, determine a
1577 surface peak / cloud peak ratio. If that ratio is less than or equal to 0.4,
1578 set low_signal flag for that column to true.
- 1579 7. After each matching ATL09 column of cab_smoother has been analyzed for
1580 low signal, assign the low_signal flag to an ATL03 photon resolution logical

1581 array by matching up the ATL03 photon segment_id values to the ATL09
1582 range of segment IDs for each ATL09 cab_prof column.

1583 8. For each ATL09 cab_prof column where the low_signal flag was not set, check
1584 for any ATL03 photons greater than 800 meters (TBD) in elevation away
1585 (higher or lower) from the ATL09 dem_h value. Assign an ATL03 photon
1586 resolution too_far_signal flag to true when this conditional is met.

1587 9. A logical array mask is created for any ATL03 photons that have either the
1588 low_signal flag or the too_far_signal flag set to true such that those photons
1589 will not be further processed by the ATL08 function.

1590

1591 **4.2 Preparing ATL03 data for input to ATL08 algorithm**

1592 1. Break up data into *L-km* segments. Segments equivalent of 10 km in along-
1593 track distance of an orbit would be appropriate.

1594 a. If the last portion of an ATL03 granule being processed would result
1595 in an *L-km* segment with less than 3.4 km (170 geosegments) worth of
1596 data, that last portion is added to the previous *L-km* processing
1597 window to be processed together as one extended *L-km* processing
1598 segment.

1599 i. The resulting **last_seg_extend** value would be reported as a
1600 positive value of distance beyond 10 km that the ATL08
1601 processing segment was extended by.

1602 b. If the last *L-km* segment would be less than 10 km but greater than 3.4
1603 km, a portion extending from the start of current *L-km* processing
1604 segment backwards into the previous *L-km* processing segment would
1605 be added to the current ATL08 processing segment to make it 10 km
1606 in length. Only new 100 m ATL08 segment products generated from
1607 this backward extension would be reported.

- 1608 i. The distance of this backward data gathering would be
1609 reported in **last_seg_extend** as a negative distance value.
- 1610 c. All other segments that are not extended will report a last_seg_extend
1611 value of 0.
- 1612 2. Add a buffer of 200 m (or 10 segment_id's) to both ends of each *L-km*
1613 segment. The total processing segment length is $(L-km + 2*buffer)$, but will
1614 be referred to as *L-km* segments for simplicity.
- 1615 a. The first *L-km* segment from an ATL03 granule would only have a
1616 buffer at the end, and the last *L-km* segment from an ATL03 granule
1617 would only have a buffer at the beginning.
- 1618 3. The input data for ATL08 algorithm is X, Y, Z, T (where T is time).

1619

1620 **4.3 Noise filtering via DRAGANN**

1621 DRAGANN will use ATL03 photons with all signal classification flags (0-4). These
1622 will include both signal and noise photons. This section give a broad overview of the
1623 DRAGANN function. See Appendix A for more details.

- 1624 1. Determine the relative along-track time, ATT, of each geolocated photon
1625 from the beginning of each *L-km* segment.
- 1626 2. Rescale the ATT with equal-time spacing between each data photon, keeping
1627 the relative beginning and end time values the same.
- 1628 3. Normalize the height and rescaled ATT data from 0 – 1 for each *L-km*
1629 segment based on the min/max of each field. So, $normtime = (time -$
1630 $mintime)/(maxtime - mintime)$.
- 1631 4. Build a kd-tree based on normalized Z and normalized and rescaled ATT.
- 1632 5. Determine the search radius starting with Equation 3.1. $P=[$ determined by
1633 preprocessor; see Sec 1.1.1], and $V_{total}=1$. N_{total} is the number of photons
1634 within the data *L-km* segment. Solve for V .
- 1635 6. Now that you know V , determine the radius using Equation 3.2.

- 1636 7. Compute the number of neighbors for each photon using this search radius.
- 1637 8. Generate a histogram of the neighbor count distribution. As illustrated in
- 1638 Figure 3.2, the noise peak is the first peak (usually with the highest
- 1639 amplitude).
- 1640 9. Determine the 10 highest peaks of the histogram.
- 1641 10. Fit Gaussians to the 10 highest peaks. For each peak,
- 1642 a. Compute the amplitude, a , which is located at peak position b .
- 1643 b. Determine the width, c , by stepping one bin at a time away from b and
- 1644 finding the last histogram value that is $> \frac{1}{2}$ the amplitude, a .
- 1645 c. Use the amplitude and width to fit a Gaussian to the peak of the
- 1646 histogram, as described in Equation 3.3.
- 1647 d. Subtract the Gaussian from the histogram, and move on to calculate
- 1648 the next highest peak's Gaussian.
- 1649 e. Reject Gaussians that are too near (< 2 standard deviations) and
- 1650 amplitude too low ($< 1/5$ previous amplitude) from the previous
- 1651 signal Gaussian.
- 1652 11. Reject any of the returned Gaussians with imaginary components.
- 1653 12. Determine if there is a narrow noise Gaussian at the beginning of the
- 1654 histogram. These typically occur when there is little noise, such as during
- 1655 nighttime passes.
- 1656 a. Search for the Gaussian with the highest amplitude, a , in the first 5%
- 1657 of the histogram
- 1658 b. Check if the highest amplitude is $\geq 1/10$ of the maximum of all
- 1659 Gaussian amplitudes
- 1660 c. Check if the width, c , of the Gaussian with the highest amplitude is \leq
- 1661 4 bins
- 1662 d. If these three conditions are met, save the $[a,b,c]$ values as $[a_0,b_0,c_0]$.
- 1663 e. If the three conditions are not met, search again within the first 10%.
- 1664 Repeat the process, incrementing the percentage of histogram

- 1665 searched by 5% up to 30%. As soon as the conditions are met, save
1666 the $[a_0, b_0, c_0]$ values and break out of the percentage histogram search
1667 loop.
- 1668 13. If a narrow noise peak was found, sort the remaining Gaussians from largest
1669 to smallest area, estimated by $a \cdot c$, then append $[a_0, b_0, c_0]$ to the beginning of
1670 the sorted $[a, b, c]$ arrays. If a narrow noise peak was not found, sort all
1671 Gaussians by largest to smallest area.
- 1672 a. If a narrow noise peak was not found, check in sorted order if one of
1673 the Gaussians are in the first 10% of the histogram. If so, it becomes
1674 the first Gaussian.
 - 1675 b. Reject any Gaussians that are fully contained within another.
 - 1676 c. Reject Gaussians whose centers are within 3 standard deviations of
1677 another, unless only two Gaussians remain
- 1678 14. If there are two or more Gaussians remaining, they are referred to as
1679 Gaussian 1 and Gaussian 2, assumed to be the noise and signal Gaussians.
- 1680 15. Determine the threshold value that will define the cutoff between noise and
1681 signal.
- 1682 a. If the absolute difference of the two Gaussians becomes near zero,
1683 defined as $< 1e-8$, set the first bin index where that occurs, past the
1684 first Gaussian peak location, as the threshold. This would typically be
1685 set if the two Gaussians are far away from each other.
 - 1686 b. Else, the threshold value is the intersection of the two Gaussians,
1687 which can be estimated as the first bin index past the first Gaussian
1688 peak location where there is a minimum absolute difference between
1689 the two Gaussians.
 - 1690 c. If there is only one Gaussian, it is assumed to be the noise Gaussian,
1691 and the threshold is set to $b + c$.
- 1692 16. Label all photons having a neighbor count above the threshold as signal.
- 1693 17. Label all photons having a neighbor count below the threshold as noise.

- 1694 18. Reject noise photons.
1695 19. Retain signal photons for feeding into next step of processing.
1696 20. Use Logical OR to combine DRAGANN signal photons with ATL03 medium-
1697 high confidence signal photons (flags 3-4) as ATL08 signal photons.
1698 21. Calculate a signal to noise ratio (SNR) for the *L-km* segment by dividing the
1699 number of ATL08 signal photons by the number of noise (i.e., all – signal)
1700 photons.

1701 **4.3.1 Preprocessing to dynamically determine a DRAGANN parameter**

1702 While a default value of $P=20$ was found to work well when testing with MABEL
1703 flight data, further testing with simulated data showed that $P=20$ is not sufficient in
1704 cases of very low or very high noise. Additional testing with real ATL03 data have
1705 shown the ground signal to be much stronger, and the canopy signal to be much
1706 weaker, than originally anticipated. Therefore, a preprocessing step for dynamically
1707 calculating P and running the core DRAGANN function is described in this
1708 subsection. This assumes *L-km* to be 10 km (with additional *L-km* buffering).

- 1709 1. Define a DRAGANN processing window of 170 segments (~ 3.4 km),
1710 and a buffer of 10 segments (~ 200 m).
1711 2. The buffer is applied to both sides of each DRAGANN processing
1712 window to create buffered DRAGANN processing windows
1713 (referenced as “buffered window” for the rest of this section) that will
1714 overlap the DRAGANN processing windows next to them.
1715 3. For each buffered window within the *L-km* segment, calculate a
1716 histogram of points with 1 m elevation bins.
1717 4. For each buffered window histogram, calculate the median counts.
1718 5. Bins with counts below the buffered window median count value are
1719 estimated to be noise. Calculate the mean count of noise bins.
1720 6. Bins with counts above the buffered window median count value are
1721 estimated to be signal. Calculate the mean count of signal bins.

- 1722 7. Determine the time elapsed over the buffered window.
- 1723 8. Calculate estimated noise and signal rates for each buffered window
- 1724 by multiplying each window's mean counts of noise bins and signal
- 1725 bins, determined from steps 5 and 6 above, by 1/(elapsed time) to
- 1726 return the rates in terms of points/meter[elevation]/second[across].
- 1727 9. Calculate a noise ratio for each window by dividing the noise rate by
- 1728 the signal rate.
- 1729 10. If, for all the buffered windows in the *L-km* segment, the noise rate is
- 1730 less than 20 and the noise ratio is less than 0.15; OR any noise rate is
- 1731 0; OR any signal rate is greater than 1000: re-calculate steps 3-9
- 1732 using the entire *L-km* segment. Continue with the following steps
- 1733 using results from the one *L-km* window (instead of multiple buffered
- 1734 windows).
- 1735 11. Now, determine the DRAGANN parameter, P, for each buffered
- 1736 window based on the following conditionals:
- 1737 a. If the signal rate is NaN (i.e., an invalid value), set the signal
- 1738 index array to empty and move on to the next buffered
- 1739 window.
- 1740 b. If noise rate < 20 || noise ratio < 0.15:
- 1741 P = signal rate
- 1742 If signal rate is < 5, P = 5; if signal rate > 20, P = 20
- 1743 c. Else P = 20.
- 1744 12. Run DRAGANN on the buffered window points using the calculated P.
- 1745 13. If DRAGANN fails to find a signal (i.e., only one Gaussian found), run
- 1746 DRAGANN again with P = 10.
- 1747 14. If DRAGANN still fails to find a signal, try to determine P a second time
- 1748 using the following conditionals:
- 1749 a. If (noise rate >= 20) ...
- 1750 && (signal rate > 100) ...

**ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002**

```
1751         && (signal rate < 250),
1752         P = (signal rate)/2
1753     b. Else if signal rate >= 250,
1754         if noise rate >= 250,
1755             P = (noise rate)*1.1
1756         else,
1757             P = 250
1758     c. Else, P = mean(noise rate, signal rate)
1759 15. Run DRAGANN on the buffered window points using the newly
1760     calculated P.
1761     a. If still no signal points are found, set a dragannError flag.
1762 16. If signal points were found by DRAGANN, for each buffered window
1763     calculate a signal check by dividing the number of signal points found
1764     via DRAGANN by the number of total points in the buffered window.
1765 17. If dragannError has been set, or there are suspect signal statistics, the
1766     following snippet of pseudocode will check those conditionals and try
1767     to iteratively find a better P value to run DRAGANN with:
1768
1769     try_count = 0
1770
1771     While dragannError ...
1772     || ( (noise rate >= 30) ...
1773         && (signal check > noise ratio) ...
1774         && (noise ratio >= 0.15) ) ...
1775     || (signal check < 0.001):
1776
1777         if P < 3,
1778             break
1779         else,
1780             P = P*0.75
1781         end
1782
1783     if try_count < 2
1784         Clear out signal index results from previous DRAGANN run
1785         Re-run DRAGANN with new P value
1786         Recalculate the signal check
```

```
1787         end
1788
1789         if no signal index results are returned
1790             P = P*0.75
1791         end
1792
1793         try_count = try_count + 1
1794
1795     end
1796
1797     18. If no signal photons are found by DRAGANN because only one
1798         Gaussian was found, set the threshold as  $b+c$  (i.e., one standard
1799         deviation away from the Gaussian peak location) for a final DRAGANN
1800         run. Otherwise, set the signal index array to empty and move on to the
1801         next buffered window.
1802
1803     19. Assign the signal values found from DRAGANN for each buffered
1804         window to the original DRAGANN processing window range of points.
1805
1806     20. Combine signal points from each DRAGANN processing window back
1807         into one  $L$ -km array of signal points for further processing.
```

1806

1807 **4.3.2 Iterative DRAGANN processing**

1808 It is possible in processing segments with high noise rates that DRAGANN will
1809 incorrectly identify clusters of noise as signal. One way to reduce these false positive
1810 noise clusters is to run the alternative DRAGANN process (Sec 1.1.1) again with the
1811 input being the signal output photons from the first run through alternative
1812 DRAGANN. Note that this methodology is still being tested, so by default this option
1813 should not be set.

- 1814 1. If $SNR < 1$ (TBD) from alternative DRAGANN run, run alternative DRAGANN
1815 process again using the output signal photons from first DRAGANN run as the
1816 input to the second DRAGANN run.
- 1817 2. Recalculate SNR based on output of second DRAGANN run.

1818

1819 **4.4 Is Canopy Present**

- 1820 1. If L -km segment is within an ATL08 region encompassing Antarctica (regions
1821 7, 8, 9, 10) or Greenland (region 11), assume no canopy is present: canopy
1822 flag = 0. Else:
- 1823 2. Determine the center Latitude/Longitude position for the L -km segment.
- 1824 3. Determine the corresponding tile from the Landsat continuous cover
1825 product.
- 1826 4. For each unique XY position in the ATLAS segment, extract the canopy cover
1827 value from the Landsat CC product
- 1828 5. Compute the average canopy cover value for the L -km segment (based on the
1829 Landsat values).
- 1830 6. If canopy cover > 5%, set canopy flag = 1 (assumes canopy is present)
- 1831 7. If canopy cover <= 5%, set canopy flag = 0 (assumes no canopy is present)

1832

1833 **4.5 Compute Filtering Window**

- 1834 1. Next step is to run a surface filter with a variable window size (variable in
1835 that it will change from L -km segment to L -km segment). The window-size is
1836 denoted as Window.
- 1837 2. $Window = \text{ceil}[5 + 46 * (1 - e^{-a*length})]$, where $length$ is the number of
1838 photons in the segment.
- 1839 3. $a = \frac{\log\left(1 - \frac{21}{51-5}\right)}{-28114} \approx 21 \times 10^{-6}$, where a is the shape parameter for the window
1840 span.

1841

1842 **4.6 De-trend Data**

- 1843 1. The input data are the signal photons identified by DRAGANN and the ATL03
1844 classification (signal_conf_ph) values of 3-4.
- 1845 2. Generate a rough surface by connecting all unique (time) photons to each
1846 other. Let's call this surface interp_A.
- 1847 3. Run a median filter through interp_A using the window size set by the
1848 software. Output = Asmooth.
- 1849 4. Define a reference DEM limit (ref_dem_limit) as 120 m (TBD).
- 1850 5. Remove any Asmooth values further than the ref_dem_limit threshold from
1851 the reference DEM, and interpolate the Asmooth surface based on the
1852 remaining Asmooth values. The interpolation method to use is the shape
1853 preserving piecewise cubic Hermite interpolating polynomial – hereafter
1854 labeled as “pchip” (Fritsch & Carlson, 1980).
- 1855 6. Compute the approximate relief of the *L-km* segment using the 95th - 5th
1856 percentile heights of the signal photons. We are going to filter Asmooth again
1857 and the smoothing is a function of the relief.
- 1858 7. Define the SmoothSize using the conditional statements below. The
1859 SmoothSize will be used to detrend the data as well as to create an
1860 interpolated ground surface later.

1861 SmoothSize = 2 * Window

- 1862 • If relief >= 900, SmoothSize = round(SmoothSize/4)
- 1863 • If relief >= 400 && <= 900, SmoothSize = round(SmoothSize/3)
- 1864 • If relief >= 200 && <= 400, SmoothSize = round(SmoothSize/2)
- 1865 8. Greatly smooth Asmooth by first running Asmooth 10 times through a
1866 median filter then a smoothing filter with a moving average method on the
1867 result. Both the median filter and the smoothing filter use a window size of
1868 SmoothSize.

1869

1870 **4.7 Filter outlier noise from signal**

- 1871 1. If there are any signal data that are 150 meters above Asmooth, remove them
1872 from the signal data set.
- 1873 2. If the standard deviation of the detrended signal is greater than 10 meters,
1874 remove any signal value from the signal data set that is 2 times the standard
1875 deviation of the detrended signal below Asmooth.
- 1876 3. Calculate a new Asmooth surface by interpolating (pchip method) a surface
1877 from the remaining signal photons and median filtering using the Window
1878 size, then median filter and smooth (moving average method) 10 times again
1879 using the SmoothSize.
- 1880 4. Detrend the signal photons by subtracting the signal height values from the
1881 Asmooth surface height values. Use the detrended heights for surface finding.

1882

1883 **4.8 Finding the initial ground estimate**

- 1884 1. At this point, the initial signal photons have been noise filtered and de-
1885 trended and should have the following format: X, Y, detrended Z, T (T=time).
1886 From this, the input data into the ground finding will be the ATD (along track
1887 distance) metric (such as time) and the detrended Z height values.
- 1888 2. Define a medianSpan as Window*2/3.
- 1889 3. Identifying the ground surface is an iterative process. Start by assuming that
1890 all the input signal height photons are the ground. The first goal is the cut
1891 out the lower height excess photons in order to find a lower bound for
1892 potential ground photons. This process is done 5 times and an offset of 4
1893 meters is subtracted from the resulting lower bound. The smoothing filter
1894 uses a moving average again:

1895 for j=1:5

```
1896         cutOff = median filter (ground, medianSpan)
1897         cutOff = smooth filter (cutOff, Window)
1898         ground = ground( (cutOff - ground) > -1 )
1899     end
1900     lowerbound = median filter (ground, medianSpan*3)
1901     middlebound = smooth filter (lowerbound, Window)
1902     lowerbound = smooth filter (lowerbound, Window) - 4
1903     end;
1904 4. Create a linearly interpolated surface along the lower bound points and only
1905    keep input photons above that line as potential ground points:
1906
1907         top = input( input > interp(lowerbound) )
1907 5. The next goal is to cut out excess higher elevation photons in order to find an
1908    upper bound to the ground photons. This process is done 3 times and an
1909    offset of 1 meter is added to the resulting upper bound. The smoothing filter
1910    uses a moving average:
1911
1912         for j = 1:3
1912             cutOff = median filter (top, medianSpan)
1913             cutOff = smooth filter (cutOff, Window)
1914             top = top( (cutOff - top) > -1 )
1915         end
1916         upperbound = median filter (top, medianSpan)
1917         upperbound = smooth filter (upperbound, Window) + 1
1918 6. Create a linearly interpolated surface along the upper bound points and
1919    extract the points between the upper and lower bounds as potential ground
1920    points:
1921
1922         ground = input( ( input > interp(lowerbound) ) & ...
1923             ( input < interp(upperbound) ) )
```

- 1923 7. Refine the extracted ground points to cut out more canopy, again using the
1924 moving average smoothing:
- 1925 For j = 1:2
1926 cutOff = median filter (ground, medianSpan)
1927 cutOff = smooth filter (cutOff, Window)
1928 ground = ground((cutOff - ground) > -1)
1929 end
- 1930 8. Run the ground output once more through a median filter using window side
1931 medianSpan and a smoothing filter using window size Window, but this time
1932 with the Savitzky-Golay method.
- 1933 9. Finally, linearly interpolate a surface from the ground points.
- 1934 10. The first estimate of canopy points are those indices of points that are
1935 between 2 and 150 meters above the estimated ground surface. Save these
1936 indices for the next section on finding the top of canopy.
- 1937 11. The output from the final iteration of ground points is temp_interpA – an
1938 interpolated ground estimate.
- 1939 12. Find ground indices that lie within +/- 0.5 m of temp_interpA.
- 1940 13. Apply the ground indices to the original heights (i.e., not the de-trended data)
1941 to label ground photons.
- 1942 14. Interpolate a ground surface using the pchip method based on the ground
1943 photons. Output is interp_Aground.

1944

1945 **4.9 Find the top of the canopy (if canopy_flag = 1)**

- 1946 1. The input are the ATD metric (i.e., time), and the de-trended Z values indexed
1947 by the canopy indices extracted from step 4.8(10).
- 1948 2. Flip this data over so that we can find a canopy “surface” by multiplying the
1949 de-trended canopy heights by -1.0 and adding the mean(heights).

- 1950 3. Finding the top of canopy is also an iterative process. Follow the same steps
1951 described in 4.8(2) – 4.8(9), but use the canopy indexed and flipped Z values
1952 in place of the ground input.
- 1953 4. Final retained photons are considered top of canopy photons. Use the indices
1954 of these photons to define top of canopy photons in the original (not de-
1955 trended) Z values.
- 1956 5. Build a kd-tree on canopy indices.
- 1957 6. If there are less than three canopy indices within a 15m radius, reassign
1958 these photons to noise photons.

1959

1960 **4.10 Compute statistics on de-trended data**

- 1961 1. The input data have been noise filtered and de-trended and should have the
1962 following input format: X, Y, detrended Z, T.
- 1963 2. The input data will contain signal photons as well as a few noise photons
1964 near the surface.
- 1965 3. Compute statistics of heights in the along-track direction using a sliding
1966 window. Using the window size (window), compute height statistics for all
1967 photons that fall within each window. These include max height, median
1968 height, mean height, min height, and standard deviation of all photon heights.
1969 Additionally, in each window compute the median height and standard
1970 deviation of just the initially classified top of canopy photons, and the
1971 standard deviation of just the initially classified ground photon heights.
1972 Currently only the median top of canopy, and all STD variables are being
1973 utilized, but it's possible that other statistics may be incorporated as
1974 changes/improvements are made to the code.
- 1975 4. Slide the window $\frac{1}{4}$ of the window span and recompute statistics along the
1976 entire *L-km* segment. This results in one value for each statistic for each
1977 window.

- 1978 5. Determine canopy index categories for each window based upon the total
1979 distribution of STD values for all signal photons along the *L-km* segment
1980 based on STD quartiles.
- 1981 6. Open canopy have STD values falling within the 1st quartile.
- 1982 7. Canopy Level 1 has STD values falling from 1st quartile to median STD value.
- 1983 8. Canopy Level 2 has STD values falling from median STD value to 3rd quartile.
- 1984 9. Canopy Level 3 has STD values falling from 3rd quartile to max STD.
- 1985 10. Linearly interpolate the window STD values (both for all photons and
1986 ground-only photons) back to the native along-track resolution and calculate
1987 the interpolated all-photon STD quartiles to create an interpolated canopy
1988 level index. This will be used later for interpolating a ground surface.
- 1989

1990 **4.11 Refine Ground Estimates**

- 1991 1. Smooth the interpolated ground surface 10 times. All further ground surface
1992 smoothing use the moving average method:

1993 For j= 1:10

1994 AgroundSmooth = median filter (interp_Aground, SmoothSize*5)

1995 AgroundSmooth = smooth filter (AgroundSmooth, SmoothSize)

1996 End

1997

- 1998 2. This output (AgroundSmooth) from the filtering/smoothing function is an
1999 intermediate ground solution and it will be used to estimate the final
2000 solution.

- 2001 3. If there are **no canopy indices** identified along the entire segment (OR
2002 canopy_flag = 0) AND relief >400 m

2003 FINALGROUND = median filter (Asmooth, SmoothSize)

2004 FINALGROUND = smooth filter (FINALGROUND, SmoothSize)

```
2005         Else
2006             FINALGROUND = AgroundSmooth
2007         end
2008     4. If there are canopy indices identified along the segment:
2009         If there is a canopy photon identified at a location along-track above the
2010         ground surface, then at that location along-track
2011             FINALGROUND = AgroundSmooth
2012         else if there is a location along-track where the interpolated ground STD has
2013         an interpolated canopy level >= 3
2014             FINALGROUND = Interp_Aground*1/3 + AgroundSmooth*2/3
2015         else
2016             FINALGROUND = Interp_Aground*1/2 + Asmooth*1/2
2017         end
2018     5. Smooth the resulting interpolated ground surface (FINALGROUND) once
2019         using a median filter with window size of SmoothSize, then a smoothing filter
2020         twice with window size of SmoothSize. Select ground photons that lie within
2021         the point spread function (PSF) of FINALGROUND.
2022     6. PSF is determined by sigma_atlas_land (Eq. 1.2) calculated at the photon
2023         resolution and thresholded between 0.5 to 1 m.
2024         a. Estimate the terrain slope by taking the gradient of
2025             FINALGROUND.Gradient is reported at the center of
2026             ((finalground(n+1)-finalground(n-1))/(dist_x(n+1)-dist_x(n-1)))/2
2027         b. Linearly interpolate the sigma_h values to the photon resolution.
2028         c. Calculate sigma_topo (Eq. 1.3) at the photon resolution.
2029         d. Calculate sigma_atlas_land at the photon resolution using the sigma_h
2030             and sigma_topo values at the photon resolution.
2031         e. Set PSF equal to sigma_atlas_land.
```

- 2032 i. Any PSF < 0.5 m is set to 0.5 m as the minimum PSF.
2033 ii. Any PSF > 1 m is set to 1 m as the maximum PSF. Set psf_flag to
2034 true.

2035

2036 **4.12 Canopy Photon Filtering**

- 2037 1. The first canopy filter will remove photons classified as top of canopy that
2038 are significantly above a smoothed median top of canopy surface. To
2039 calculate the smoothed median top of canopy surface:
- 2040 a. Linearly interpolate the median and standard deviation canopy
2041 window statistics, calculated from 4.10 (3), to the top of canopy
2042 photon resolution. Output variables: interpMedianC, interpStdC.
2043 b. Calculate a canopy window size using Eq. 3.4, where *length* = number
2044 of top of canopy photons. Output variable: winC.
2045 c. Create the median filtered and smoothed top of canopy surface,
2046 smoothedC, using a locally weighted linear regression smoothing
2047 method, “lowess” (Cleveland, 1979):

2048 smoothedC = median filter (interpMedianC, winC)

2049

2050 if SNR > 1, canopySmoothSpan = winC*2;

2051 else, canopySmoothSpan = smoothSpan;

2052

2053 smoothedC = smooth filter (smoothedC, canopySmoothSpan)

- 2054 d. Add the detrended heights back into the smoothedC surface:

2055 smoothedC = smoothedC + Asmooth

- 2056 2. Set canopy height thresholds based on the interpolated top of canopy STD:

**ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002**

2057 If SNR > 1, canopySTDthresh = 3; else, canopySTDthresh = 2;
2058 canopy_height_thresh = canopySTDthresh*interpStdC

2059 high_cStd = canopy_height_thresh > 10

2060 low_cStd = canopy_height_thresh < 3

2061 canopy_height_thresh(high_cStd) =
2062 canopy_height_thresh(high_cStd)/2

2063 canopy_height_thresh(low_cStd) = 3

2064 3. Relabel as noise any top of canopy photons that are higher than smoothedC +
2065 canopy_height_thresh.
2066 4. Next, interpolate a top of canopy surface using the remaining top of canopy
2067 photons (here we are trying to create an upper bound on canopy points). The
2068 interpolation method used is pchip. This output is named interp_Acanopy.
2069 5. Photons falling below interp_Acanopy and above FINALGROUND+PSF are
2070 labeled as canopy points.
2071 6. For 500 signal photon segments, if number of all canopy photons (i.e., canopy
2072 and top of canopy) is:
2073 < 5% of the total (when SNR > 1), OR
2074 < 10% of the total (when SNR <= 1),
2075 relabel the canopy photons as noise.
2076 7. Interpolate, using the pchip method, a new top of canopy surface from the
2077 filtered top of canopy photons. This output is again named interp_Acanopy.
2078 8. Again, label photons that lie between interp_Acanopy and
2079 FINALGROUND+PSF as canopy photons.
2080 9. Since the canopy points have been relabeled, we need to do a final
2081 refinement of the ground surface:
2082 If canopy is present at any location along-track

2083 FINALGROUND = AgroundSmooth (at that location)

2084 Else if canopy is not present at a location along-track

2085 FINALGROUND = interp_Aground

2086 Smooth the resulting interpolated ground surface (FINALGROUND) once
2087 using a median filter with window size of SmoothSize, then a moving average
2088 smoothing filter twice with window size of SmoothSize.

2089 10. Relabel ground photons based on this new (and last) FINALGROUND solution
2090 +/- a recalculated PSF (via steps in 4.11 (6)). Points falling below the buffer
2091 are labeled as noise.

2092 11. Using Interp_Acanopy and this last FINALGROUND solution + PSF buffer,
2093 label all photons that lie between the two as canopy photons.

2094 12. Repeat the canopy cover filtering: For 500 signal photon segments, if
2095 number of all canopy photons (i.e., canopy and top of canopy) is:
2096 < 5% of the total (when SNR > 1), OR
2097 < 10% of the total (when SNR <= 1),
2098 relabel the canopy photons as noise. This is the last canopy labeling step.

2099

2100 **4.13 Compute individual Canopy Heights**

- 2101 1. At this point, each photon will have its final label assigned in
2102 **classed_pc_flag**: 0 = noise, 1 = ground, 2 = canopy, 3 = top of canopy.
- 2103 2. For each individual photon labeled as canopy or top of canopy, subtract the Z
2104 height value from the interpolated terrain surface, FINALGROUND, at that
2105 particular position in the along-track direction.
- 2106 3. The relative height for each individual canopy or top of canopy photon will
2107 be used to calculate canopy products described in Section 4.16. Additional
2108 canopy products will be calculated using the absolute heights, as described in

2109 Section 4.16.1.

2110

2111 **4.14 Final photon classification QA check**

2112 1. Find any ground, canopy, or top of canopy photons that have elevations
2113 further than the ref_dem_limit from the reference DEM elevation value.

2114 Convert these to the noise classification.

2115 2. Find any relative heights of canopy or top of canopy photons that are greater
2116 than 150 m above the interpolated ground surface, FINALGROUND. Convert
2117 these to the noise classification.

2118 3. Find any FINALGROUND elevations that are further than the ref_dem_limit
2119 from the reference DEM elevation value. Convert those FINALGROUND
2120 elevations to an invalid value, and convert any classified photons at the same
2121 indices to noise.

2122 4. If more than 50% of photons are removed in a segment, set ph_removal_flag
2123 to true.

2124

2125 **4.15 Compute segment parameters for the Land Products**

2126 1. For each 100 m segment, determine the classed photons (photons classified
2127 as ground, canopy, or top of canopy).

2128 a. If there are fewer than 50 classed photons in a 100 m segment, do not
2129 calculate land or canopy products.

2130 b. If there are 50 or more classed photons in a 100 m segment, extract
2131 the ground photons to create the land products.

2132 2. If the number of ground photons > 5% of the total number of classed photons
2133 within the segment (this control value of 5% can be modified once on orbit):

2134 a. Compute statistics on the ground photons: mean, median, min, max,
2135 standard deviation, mode, and skew. These heights will be reported

- 2136 on the product as **h_te_mean**, **h_te_median**, **h_te_min**, **h_te_max**,
2137 **h_te_mode**, and **h_te_skew** respectively described in Table 2.1.
- 2138 b. Compute the standard deviation of the ground photons about the
2139 interpolated terrain surface, FINALGROUND. This value is reported as
2140 **h_te_std** in Table 2.1.
- 2141 c. Compute the residuals of the ground photon Z heights about the
2142 interpolated terrain surface, FINALGROUND. The product is the root
2143 sum of squares of the ground photon residuals combined with the
2144 **sigma_atlas_land** term in Table 2.5 as described in Equation 1.4. This
2145 parameter reported as **h_te_uncertainty** in Table 2.1.
- 2146 d. Compute a linear fit on the ground photons and report the slope. This
2147 parameter is **terrain_slope** in Table 2.1.
- 2148 e. Calculate a best fit terrain elevation at the mid-point location of the
2149 100 m segment:
- 2150 i. Calculate each terrain photon's distance along-track into the
2151 100 m segment using the corresponding ATL03 20 m products
2152 **segment_length** and **dist_ph_along**, and determine the mid-
2153 segment distance (expected to be 50 m \pm 0.5 m).
- 2154 1. Use the mid-segment distance to linearly interpolate a
2155 mid-segment time (**delta_time** in Table 2.4). Use the
2156 mid-segment time to linearly interpolate other mid-
2157 segment parameters: interpolated terrain surface,
2158 FINALGROUND, as **h_te_interp** (Table 2.1); **latitude**
2159 and **longitude** (Table 2.4).
- 2160 ii. Calculate a linear fit, as well as 3rd and 4th order polynomial fits
2161 to the terrain photons in the segment.
- 2162 iii. Create a slope-adjusted and weighted mid-segment variable,
2163 **weightedZ**, from the linear fit: Use **terrain_slope** to apply a
2164 slope correction to each terrain photon by subtracting the

2165 terrain photon heights from the linear fit. Determine the mid-
2166 segment location of the linear fit, and add that height to the
2167 slope corrected terrain photons. Apply a linear weighting to
2168 each photon based on its distance to the mid-segment location:
2169 $1 / \sqrt{(\text{photon distance along} - \text{mid-segment distance})^2}$.
2170 Calculate the weighted mid-segment terrain height, weightedZ:
2171 $\text{sum}(\text{each adjusted terrain height} * \text{its weight}) / \text{sum}(\text{all}$
2172 $\text{weights})$.

2173 iv. Determine which of the three fits is best by calculating the
2174 mean and standard deviation of the fit errors. If one of the fits
2175 has both the smallest mean and standard deviations, use that
2176 fit. Else, use the fit with the smallest standard deviation. If
2177 more than one fit has the same smallest mean and/or standard
2178 deviation, use the fit with the higher polynomial.

2179 v. Use the best fit to define the mid-segment elevation. This
2180 parameter is **h_te_best_fit** in Table 2.1.

2181 1. If **h_te_best_fit** is farther than 3 m from **h_te_interp** (best
2182 fit diff threshold), check if: there are terrain photons on
2183 both sides of the mid-segment location; or the elevation
2184 difference between weightedZ and **h_te_interp** is
2185 greater than the best fit diff threshold; or the number of
2186 ground photons in the segment is $\leq 5\%$ of total
2187 number of classified photons per segment. If any of
2188 those cases are present, use **h_te_interp** as the corrected
2189 **h_te_best_fit**. Otherwise use weightedZ as the corrected
2190 **h_te_best_fit**.

2191 f. Compute the difference of the median ground height from the
2192 reference DTM height. This parameter is **h_dif_ref** in Table 2.4.
2193

- 2194 3. If the number of ground photons in the segment \leq 5% of total number of
2195 classified photons per segment,
2196 a. Report an invalid value for terrain products: **h_te_mean**,
2197 **h_te_median**, **h_te_min**, **h_te_max**, **h_te_mode**, **h_te_skew**, **h_te_std**,
2198 **and h_te_uncertainty** respectively as described in Table 2.1.
2199 b. If the number of ground photons in the segment is \leq 5% of total
2200 number of classified photons in the segment, compute **terrain_slope**
2201 via a linear fit of the interpolated ground surface, FINALGROUND,
2202 instead of the ground photons.
2203 c. Report the mid-segment interpolated terrain surface, FinalGround, as
2204 **h_te_interp** as described in Table 2.1, and report **h_te_best_fit** as the
2205 **h_te_interp** value.
2206

2207 **4.16 Compute segment parameters for the Canopy Products**

- 2208 1. For each 100 m segment, determine the classed photons (photons classified as
2209 ground, canopy, or top of canopy).
2210 a) If there are fewer than 50 classed photons in a 100 m segment, do not
2211 calculate land or canopy products.
2212 b) If there are 50 or more classed photons in a 100 m segment, extract all
2213 canopy photons (i.e., canopy and top of canopy; henceforth referred to
2214 as “canopy” unless otherwise noted) to create the canopy products.
2215 2. Only compute canopy height products if the number of canopy photons is $>$
2216 5% of the total number of classed photons within the segment (this control
2217 value of 5% can be modified once on orbit).
2218 a) If the number of ground photons is also $>$ 5% of the total number of
2219 classed photons within the segment, set **canopy_rh_conf** to 2.
2220 b) If the number of ground photons is $<$ 5% of the total number of classed
2221 photons within the segment, continue with the relative canopy height
2222 calculations, but set **canopy_rh_conf** to 1.

- 2223 c) If the number of canopy photons is < 5% of the total number of classed
2224 photons within the segment, regardless of ground percentage, set
2225 canopy_rh_conf to 0 and report an invalid value for each canopy height
2226 variable.
- 2227 3. Again, the relative heights (height above the interpolated ground surface,
2228 FINALGROUND) have been computed already. All parameters derived in the
2229 section are based on relative heights.
- 2230 4. Sort the heights and compute a cumulative distribution of the heights. Select
2231 the height associated with the 98% maximum height. This value is **h_canopy**
2232 listed in Table 2.2.
- 2233 5. Compute statistics on the relative canopy heights. Min, Mean, Median, Max and
2234 standard deviation. These values are reported on the product as
2235 **h_min_canopy**, **h_mean_canopy**, **h_max_canopy**, and **canopy_openness**
2236 respectively in Table 2.2.
- 2237 6. Using the cumulative distribution of relative canopy heights, select the heights
2238 associated with the **canopy_h_metrics** percentile distributions (25, 50, 60, 70,
2239 75, 80, 85, 90, 95), and report as listed in Table 2.2.
- 2240 7. Compute the difference between h_canopy and canopy_h_metrics(50). This
2241 parameter is **h_dif_canopy** reported in Table 2.2 and represents an amount of
2242 canopy depth.
- 2243 8. Compute the standard deviation of all photons that were labeled as Top of
2244 Canopy (flag 3) in the photon labeling portion. This value is reported on the
2245 data product as **toc_roughness** listed in Table 2.2.
- 2246 9. The quadratic mean height, **h_canopy_quad** is computed by

2247
$$qmh = \sqrt{\frac{\sum_{i=1}^{N_{ca}} h_i^2}{N_{ca}}}$$

2248 where N_{ca} is the number of canopy photons in the segment and h_i are the
2249 individual canopy heights.

2250

2251 **4.16.1 Canopy Products calculated with absolute heights**

- 2252 1. The absolute canopy height products are calculated if the number of canopy
2253 photons is > 5% of the total number of classed photons within the segment.
2254 No number of ground photons threshold is applied for these.
- 2255 2. The **centroid_height** parameter in Table 2.2 is represented by all the classed
2256 photons for the segment (canopy & ground). To determine the centroid
2257 height, compute a cumulative distribution of all absolute classified heights
2258 and select the median height.
- 2259 3. Calculate **h_canopy_abs**, the 98th percentile of the absolute canopy heights.
- 2260 4. Compute statistics on the absolute canopy heights: Min, Mean, Median, and
2261 Max. These values are reported on the product as **h_min_canopy_abs**,
2262 **h_mean_canopy_abs**, and **h_max_canopy_abs**, respectively, as described in
2263 Table 2.2.
- 2264 5. Again, using the cumulative distribution of absolute canopy heights, select
2265 the heights associated with the **canopy_h_metrics_abs** percentile
2266 distributions (25, 50, 60, 70, 75, 80, 85, 90, 95), and report as listed in Table
2267 2.2.

2268 **4.17 Record final product without buffer**

- 2269 1. Now that all products have be determined via processing of the *L-km*
2270 segment with the buffer included, remove the products that lie within the
2271 buffer zone on each end of the *L-km* segment.
- 2272 2. Record the final *L-km* products and move on to process the next *L-km*
2273 segment.

2274

2275

2276 **5 DATA PRODUCT VALIDATION STRATEGY**

2277 Although there are no Level-1 requirements related to the accuracy and precision
2278 of the ATL08 data products, we are presenting a methodology for validating terrain
2279 height, canopy height, and canopy cover once ATL08 data products are created.
2280 Parameters for the terrain and canopy will be provided at a fixed size of 100 m along
2281 the ground track referred to as a segment. Validation of the data parameters should
2282 occur at the 100 m segment scale and residuals of uncertainties are quantified (i.e.
2283 averaged) at the 5-km scale. This 5-km length scale will allow for quantification of
2284 errors and uncertainties at a local scale which should reflect uncertainties as a
2285 function of surface type and topography.

2286

2287 **5.1 Validation Data**

2288 Swath mapping airborne lidar is the preferred source of validation data for the
2289 ICESat-2 mission due to the fact that it is widely available and the errors associated
2290 with most small-footprint, discrete return data sets are well understood and
2291 quantified. Profiling airborne lidar systems (such as MABEL) are more challenging to
2292 use for validation due to the low probability of exact overlap of flightlines between
2293 two profiling systems (e.g. ICESat-2 and MABEL). In order for the ICESat-2 validation
2294 exercise to be statistically relevant, the airborne data should meet the requirements
2295 listed in Table 5.1. Validation data sets should preferably have a minimum average
2296 point density of 5 pts/m². In some instances, however, validation data sets with a
2297 lower point density that still meet the requirements in Table 5.1 may be utilized for
2298 validation to provide sufficient spatial coverage.

2299 Table 5.1. Airborne lidar data vertical height (Z accuracy) requirements for validation data.

ICESat-2 ATL08 Parameter	Airborne lidar (rms)
Terrain height	<0.3 m over open ground (vertical) <0.5 m (horizontal)

**ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002**

Canopy height	<2 m temperate forest, < 3 m tropical forest
Canopy cover	n/a

2300

2301 Terrain and canopy heights will be validated by computing the residuals between the
2302 ATL08 terrain and canopy height value, respectively, for a given 100 m segment and
2303 the terrain height (or canopy height) of the validation data for that same
2304 representative distance. Canopy cover on the ATL08 data product shall be validated
2305 by computing the relative canopy cover ($cc = \text{canopy returns} / \text{total returns}$) for the
2306 same representative distance in the airborne lidar data.

2307 It is recommended that the validation process include the use of ancillary data sets
2308 (i.e. Landsat-derived annual forest change maps) to ensure that the validation results
2309 are not errantly biased due to non-equivalent content between the data sets.

2310 Using a synergistic approach, we present two options for acquiring the required
2311 validation airborne lidar data sets.

2312

2313 **Option 1:**

2314 We will identify and utilize freely available, open source airborne lidar data as the
2315 validation data. Potential repositories of this data include OpenTopo (a NSF
2316 repository or airborne lidar data), NEON (a NSF repository of ecological monitoring
2317 in the United States), and NASA GSFC (repository of G-LiHT data). In addition to
2318 small-footprint lidar data sets, NASA Mission data (i.e. ICESat and GEDI) can also be
2319 used in a validation effort for large scale calculations.

2320

2321 **Option 2:**

**ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002**

2322 Option 2 will include Option 1 as well as the acquisition of additional airborne lidar
2323 data that will benefit multiple NASA efforts.

2324 GEDI: With the launch of the Global Ecosystems Dynamic Investigation
2325 (GEDI) mission in 2018, there are tremendous synergistic activities for
2326 data validation between both the ICESat-2 and GEDI missions. Since the
2327 GEDI mission, housed on the International Space Station, has a
2328 maximum latitude of 51.6 degrees, much of the Boreal zone will not be
2329 mapped by GEDI. The density of GEDI data will increase as latitude
2330 increases north to 51.6 degrees. Since the data density for GEDI would
2331 be at its highest near 51.6 degrees, we would propose to acquire
2332 airborne lidar data in a “GEDI overlap zone” that would ample
2333 opportunity to have sufficient coverage of benefit to both ICESat-2 and
2334 GEDI for calibration and validation.

2335 We recommend the acquisition of new airborne lidar collections that will meet our
2336 requirements to best validate ICESat-2 as well as be beneficial for the GEDI mission.
2337 In particular, we would like to obtain data over the following two areas:

- 2338 1) Boreal forest (as this forest type will NOT be mapped with GEDI)
- 2339 2) GEDI high density zone (between 50 to 51.6 degrees N). Airborne lidar data
2340 in the GEDI/ICESat-2 overlap zone will ensure cross-calibration between
2341 these two critical datasets which will allow for the creation of a global,
2342 seamless terrain, canopy height, and canopy cover product for the
2343 ecosystem community.

2344 In both cases, we would fly data with the following scenario:

2345 Small-footprint, full-waveform, dual wavelength (green and NIR), high point density
2346 (>20 pts/m²) and, over low and high relief locations. In addition, the newly acquired
2347 lidar data must meet the error accuracies listed in Table 5.1.

2348 Potential candidate acquisition areas include: Southern Canadian Rocky Mountains
2349 (near Banff), Pacific Northwest mountains (Olympic National Park, Mt. Baker-
2350 Snoqualmie National Forest), and Sweden/Norway. It is recommended that the
2351 airborne lidar acquisitions occur during the summer months to avoid snow cover in
2352 either 2016 or 2017 prior to launch of ICESat-2.

2353

2354 **5.2 Internal QC Monitoring**

2355 In addition to the data product validation, internal monitoring of data
2356 parameters and variables is required to ensure that the final ATL08 data quality
2357 output is trustworthy. Table 5.2 lists a few of the computed parameters that should
2358 provide insight into the performance of the surface finding algorithm within the
2359 ATL08 processing chain.

2360 Table 5.2. ATL08 parameter monitoring.

Group	Description	Source	Monitor	Validate in Field
h_te_median	Median terrain height for segment	computed		Yes against airborne lidar data. The airborne lidar data should have an absolute accuracy of <30 cm rms.
n_te_photons n_ca_photons n_toc_photons	Number of classed (sum of terrain, canopy, and top of canopy) photons in a 100 m segment	computed	Yes. Build an internal counter for the number of segments in a row	

**ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002**

h_te_interp	Interpolated terrain surface height, FINALGROUND	computed	<p>where there aren't enough photons (currently a minimum of 50 photons per 100 m segment is used)</p> <p>Difference h_te_interp and h_te_median and determine if the value is > a specified threshold. 2 m is suggested as the threshold value. This is an internal check to evaluate whether the median elevation for a segment is roughly the same as the interpolated surface height.</p>
h_dif_ref	Difference between h_te_median and ref_dem	computed	<p>This value will be computed and flagged if the difference is > 25 m. The</p>

**ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002**

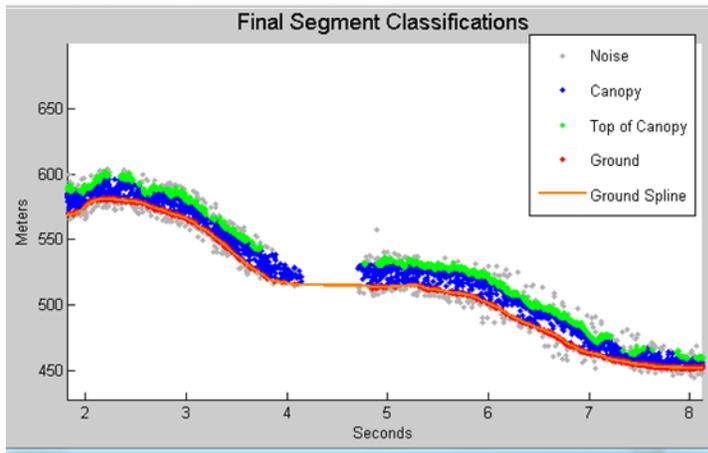
h_canopy	95% height of individual canopy heights for segment	computed	reference DEM is the onboard DEM. Yes, > a specified threshold (e.g. 60 m)	Yes against airborne lidar data. The canopy heights derived from airborne lidar data should have a relative accuracy <2 m in temperate forest, <3 m in tropical forest
h_dif_canopy	Difference between h_canopy and canopy_h_metrics(50)	computed	Yes, this is an internal check to make sure the calculations on canopy height are not suspect	
psf_flag	Flag is set if computed PSF exceeds 1m	computed	Yes, this is an internal check to make sure the calculations are not suspect	
ph_removal_flag	Flag is set if more than 50% of classified photons in a segment is removed during final QA check	computed		

**ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002**

dem_removal_flag	Flag is set if more than 20% of classified photons in a segment is removed due to a large distance from the reference DEM	computed	Yes, this will check if bad results are due to bad DEM values or because too much noise was labeled as signal
-------------------------	---	----------	---

2361

2362 In addition to the monitoring parameters listed in Table 5.2, a plot such as what is
2363 shown in Figure 5.1 would be helpful for internal monitoring and quality
2364 assessment of the ATL08 data product. Figure 5.1 illustrates in graphical form what
2365 the input point cloud look like in the along-track direction, the classifications of each
2366 photon, and the estimated ground surface (FINALGROUND).



2367

2368 Figure 5.1. Example of *L-km* segment classifications and interpolated ground surface.

2369

**ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002**

2370 The following parameters are to be calculated and placed in the QA/QC group on the
2371 HDF5 data file, based on Table 5.2 of the ATL08 ATBD. Statistics shall be computed
2372 on a per-granule basis and reported on the data product. If any parameter meets the
2373 QA trigger conditional, an alert will be sent to the ATL08 ATBD team for product
2374 review.

2375 Table 5.3. QA/QC trending and triggers.

QA/QC trending description	QA trigger conditional
Percentage of segments with > 50 classed photons	None
Max, median, and mean of the number of contiguous segments with < 50 classed photons	None
Number and percentage of segments with difference in $h_{te_interp} - h_{te_median}$ is greater than a specified threshold (2 m TBD)	> 50 segments in a row
Max, median, and mean of h_{diff_ref} over all segments	None
Percentage of segments where $h_{diff_ref} > 25$ m	Percentage > 75%
Percentage of segments where the h_{canopy} is > 60m	None
Max, median, and mean of h_{diff}	None
Number and percentage of Landsat continuous tree cover pixels per processing (L-km) segment with values > 100	None
Percentage of segments where psf_flag is set	Percentage > 75%

**ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002**

Percentage of classified photons removed in a segment during final photon QA check	Percentage > 50% (i.e., ph_removal_flag is set to true)
Percentage of classified photons removed in a segment during the reference DEM threshold removal process	Percentage > 20% (i.e., dem_removal_flag is set to true)

2376

2377

2378 **6 REFERENCES**

2379

2380 Carroll, M. L., Townshend, J. R., DiMiceli, C. M., Noojipady, P., & Sohlberg, R. A.
2381 (2009). A new global raster water mask at 250 m resolution. *International Journal of*
2382 *Digital Earth*, 2(4), 291–308. <http://doi.org/10.1080/17538940902951401>

2383 Channan, S., K. Collins, and W. R. Emanuel (2014). Global mosaics of the standard
2384 MODIS land cover type data. University of Maryland and the Pacific Northwest
2385 National Laboratory, College Park, Maryland, USA.

2386 Chauve, Adrien, et al. (2008). Processing full-waveform lidar data: modelling raw
2387 signals. *International archives of photogrammetry, remote sensing and spatial*
2388 *information sciences 2007*, 102-107.

2389 Cleveland, W. S. (1979). Robust Locally Weighted Regression and Smoothing
2390 Scatterplots. *Journal of the American Statistical Association*, 74(368), 829–836.
2391 <http://doi.org/10.2307/2286407>

2392 Friedl, M.A., D. Sulla-Menashe, B. Tan, A. Schneider, N. Ramankutty, A. Sibley and X.
2393 Huang (2010). MODIS Collection 5 global land cover: Algorithm refinements and
2394 characterization of new datasets, 2001-2012, Collection 5.1 IGBP Land Cover,
2395 Boston University, Boston, MA, USA.

2396 Fritsch, F.N., and Carlson, R.E. (1980). Monotone Piecewise Cubic Interpolation.
2397 *SIAM Journal on Numerical Analysis*, 17(2), 238–246.
2398 <http://doi.org/10.1137/0717021>

2399 Goshtasby, A., and O’Neill, W.D. (1994). Curve fitting by a Sum of Gaussians.
2400 *Graphical Models and Image Processing*, 56(4), 281-288.

**ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002**

- 2401 Goetz and Dubayah (2011). Advances in remote sensing technology and
2402 implications for measuring and monitoring forest carbon stocks and change. *Carbon*
2403 *Management*, 2(3), 231-244. doi:10.4155/cmt.11.18
- 2404 Hall, F.G., Bergen, K., Blair, J.B., Dubayah, R., Houghton, R., Hurtt, G., Kellndorfer, J.,
2405 Lefsky, M., Ranson, J., Saatchi, S., Shugart, H., Wickland, D. (2011). Characterizing 3D
2406 vegetation structure from space: Mission requirements. *Remote sensing of*
2407 *environment*, 115(11), 2753-2775
- 2408 Harding, D.J., (2009). Pulsed laser altimeter ranging techniques and implications for
2409 terrain mapping, in *Topographic Laser Ranging and Scanning: Principles and*
2410 *Processing*, Jie Shan and Charles Toth, eds., CRC Press, Taylor & Francis Group, 173-
2411 194.
- 2412 Neuenschwander, A.L. and Magruder, L.A. (2016). The potential impact of vertical
2413 sampling uncertainty on ICESat-2/ATLAS terrain and canopy height retrievals for
2414 multiple ecosystems. *Remote Sensing*, 8, 1039; doi:10.3390/rs8121039
- 2415 Neuenschwander, A.L. and Pitts, K. (2019). The ATL08 Land and Vegetation Product
2416 for the ICESat-2 Mission. *Remote Sensing of Environment*, 221, 247-259.
2417 <https://doi.org/10.1016/j.rse.2018.11.005>
- 2418 Neumann, T., Brenner, A., Hancock, D., Robbins, J., Saba, J., Harbeck, K. (2018). ICE,
2419 CLOUD, and Land Elevation Satellite – 2 (ICESat-2) Project Algorithm Theoretical
2420 Basis Document (ATBD) for Global Geolocated Photons (ATL03).
- 2421 Olson, D. M., Dinerstein, E., Wikramanayake, E. D., Burgess, N. D., Powell, G. V. N.,
2422 Underwood, E. C., D'Amico, J. A., Itoua, I., Strand, H. E., Morrison, J. C., Loucks, C. J.,
2423 Allnutt, T. F., Ricketts, T. H., Kura, Y., Lamoreux, J. F., Wettengel, W. W., Hedao, P.,
2424 Kassem, K. R. (2001). Terrestrial ecoregions of the world: a new map of life on Earth.
2425 *Bioscience*, 51(11), 933-938.

***ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002***

2426 Sexton, J.O., Song, X.-P., Feng, M. Noojipady, P., Anand, A., Huang, C., Kim, D.-H.,
2427 Collins, K.M., Channan, S., DiMiceli, C., Townshend, J.R.G. (2013). Global, 30-m
2428 resolution continuous fields of tree cover: Landsat-based rescaling of MODIS
2429 Vegetation Continuous Fields with lidar-based estimations of error. *International*
2430 *Journal of Digital Earth*, 130321031236007. doi:10.1080/17538947.2013.786146.

2431

2432 **Appendix A**

2433 **DRAGANN Gaussian Deconstruction**

2434 John Robbins

2435 20151021

2436

2437 Updates made by Katherine Pitts:

2438 20170808

2439 20181218

2440

2441 **Introduction**

2442 This document provides a verbal description of how the DRAGANN (Differential,
2443 Regressive, and Gaussian Adaptive Nearest Neighbor) filtering system deconstructs
2444 a histogram into Gaussian components, which can also be called *iteratively fitting a*
2445 *sum of Gaussian Curves*. The purpose is to provide enough detail for ASAS to create
2446 operational ICESat-2 code required for the production of the ATL08, Land and
2447 Vegetation product. This document covers the following Matlab functions within
2448 DRAGANN:

2449 • mainGaussian_dragann

2450 • findpeaks_dragann

2451 • peakWidth_dragann

2452 • checkFit_dragann

2453

2454 Components of the k-d tree nearest-neighbor search processing and histogram
2455 creation were covered in the document, *DRAGANN k-d Tree Investigations*, and have
2456 been determined to function consistently with UTexas DRAGANN Matlab software.

2457

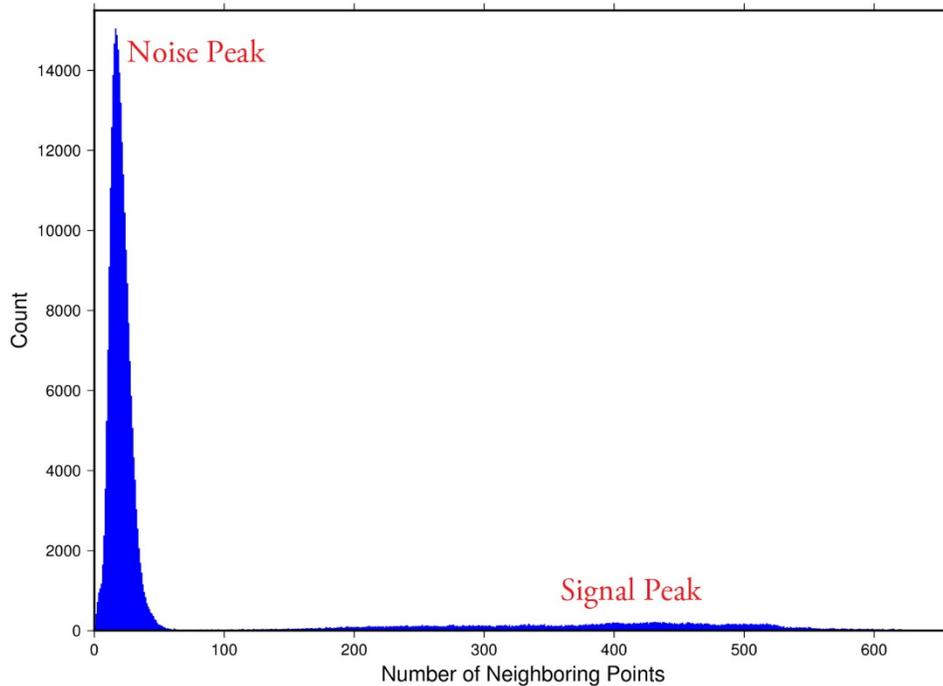
2458 **Histogram Creation**

2459 Steps to produce a histogram of nearest-neighbor counts from a normalized photon
2460 cloud segment have been completed and confirmed. Figure A.1 provides an example
2461 of such a histogram. The development, below, is specific to the two-dimensional
2462 case and is provided as a review.

2463 The histogram represents the frequency (count) of the number of nearby photons
2464 within a specified radius, as ascertained for each point within the photon cloud. The
2465 radius, R , is established by first normalizing the photon cloud in time (x-axis) and in
2466 height (y-axis), i.e., both sets of coordinates (time & height) run from 0 to 1; then an

2467 average radius for finding 20 points is determined based on forming the ratio of 20
2468 to the total number of the photons in the cloud (N_{total}): $20/N_{total}$.

2469



2470

2471 **Figure A.1.** Histogram for Mabel data, channel 43 from SE-AK flight on July 30, 2014
2472 at 20:16.

2473 Given that the total area of the normalized photon cloud is, by definition, 1, then this
2474 ratio gives the average area, A , in which to find 20 points. A corresponding radius is
2475 found by the square root of A/π . A single equation describing the radius, as a
2476 function of the total number of photons in the cloud (remembering that this is done
2477 in the cloud normalized, two-dimensional space), is given by

2478
$$R = \sqrt{\frac{20/N_{total}}{\pi}} \quad (A.1)$$

2479 For the example in Figure A.1, R was found to be 0.00447122. The number of
2480 photons falling into this radius, at each point in the photon cloud, is given along the
2481 x-axis; a count of their number (or frequency) is given along the y-axis.

2482

2483 **Gaussian Peak Removal**

2484

2485 At this point, the function, mainGaussian_dragann, is called, which passes the
2486 histogram and the number of peaks to detect (typically set to 10).

2487 This function essentially estimates (i.e., fits) a sequence of Gaussian curves, from
2488 larger to smaller. It determines a Gaussian fit for the highest histogram peak, then
2489 removes it before determining the fit for the next highest peak, etc. In concept, the
2490 process is an iterative sequential-removal of the ten largest Gaussian components
2491 within the histogram.

2492 In the process of *sequential least-squares*, parameters are re-estimated when input
2493 data is incrementally increased and/or improved. The present problem operates in
2494 a slightly reverse way: the data set is fixed (i.e., the histogram), but components
2495 within the histogram (independent Gaussian curve fits) are removed sequentially
2496 from the histogram. The paper by *Goshtasby & O'Neill (1994)* outlines the concepts.

2497 Recall that a Gaussian curve is typically written as

$$2498 \qquad y = a \cdot \exp(-(x - b)^2 / 2c^2) \qquad (A.2)$$

2499 where a = the height of the peak; b = position of the peak; and c = width of the bell
2500 curve.

2501 The function, mainGaussian_dragann, computes the $[a, b, c]$ values for the ten
2502 highest peaks found in the histogram. At initialization, these $[a, b, c]$ values are set to
2503 zero. The process begins by locating histogram peaks via the function,
2504 findpeaks_dragann.

2505

2506 **Peak Finding**

2507 As input arguments, the findpeaks_dragann function receives the histogram and a
2508 minimum peak size for consideration (typically set to zero, which means all peaks
2509 will be found). An array of index numbers (i.e., the “number of neighboring points”,
2510 values along x-axis of Figure A.1) for all peaks is returned and placed into the
2511 variable peaks.

2512 The methodology for locating each peak goes like this: The function first computes
2513 the derivatives of the histogram. In Matlab there is an intrinsic function, called diff,
2514 which creates an array of the derivatives. Diff essentially computes the differences
2515 along sequential, neighboring values. “ $Y = \text{diff}(X)$ calculates differences between
2516 adjacent elements of X .” [from Matlab Reference Guide] Once the derivatives are
2517 computed, then findpeaks_dragann enters a loop that looks for changes in the sign
2518 of the derivative (positive to negative). It skips any derivatives that equal zero.

2519 For the k th derivative, the “next” derivative is set to $k+1$. A test is made whereby if
2520 the $k+1$ derivative equals zero and $k+1$ is less than the total number of histogram

2521 values, then increment “*next*” to $k+2$ (i.e., find the next negative derivative). The test
2522 is iterated until the start of the “down side” of the peak is found (i.e., these iterations
2523 handle cases when the peak has a flat top to it).

2524 When a sign change (positive to negative) is found, the function then computes an
2525 approximate index location (variable *maximum*) of the peak via

$$2526 \quad \quad \quad \textit{maximum} = \textit{round} \left(\frac{\textit{next} - k}{2} \right) + k \quad \quad \quad (\text{A.3})$$

2527 These values of *maximum* are retained in the peaks array (which can be *grown* in
2528 Matlab) and returned to the function mainGaussian_dragann.

2529 Next, back within mainGaussian_dragann, there are two tests to determine whether
2530 the first or last elements of the histogram are peaks. This is done since the
2531 findpeaks_dragann function will not detect peaks at the first or last elements, based
2532 solely on derivatives. The tests are:

2533 If (histogram(1) > histogram(2) && max(histogram)/histogram(1) < 20) then
2534 insert a value of 1 to the very first element of the peaks array (again, Matlab can
2535 easily “grow” arrays). Here, max(histogram) is the highest peak value across the
2536 whole histogram.

2537 For the case of the last histogram value (say there are N-bins), we have

2538 If (histogram(N) > histogram(N-1) && max(histogram)/histogram(N) < 4) then
2539 insert a value of N to the very last element of the peaks array.

2540 One more test is made to determine whether there any peaks were actually found
2541 for the whole histogram. If none were found, then the function,
2542 mainGaussian_dragann, merely exits.

2543

2544 **Identifying and Processing upon the Ten Highest Peaks**

2545 The function, mainGaussian_dragann, now begins a loop to analyze the ten highest
2546 peaks. It begins the n^{th} loop (where n goes from 1 to 10) by searching for the largest
2547 peak among all remaining peaks. The index number, as well as the magnitude of the
2548 peak, are retained in a variable, called *maximum*, with dimension 2.

2549 In each pass in the loop, the $[a,b,c]$ values (see eq. 2) are retained as output of the
2550 function. The values of a and b are set equal to the index number and peak
2551 magnitude saved in *maximum*(1) and *maximum*(2), respectively. The c -value is
2552 determined by calling the function, peakWidth_dragann.

2553 *Determination of Gaussian Curve Width*

2554 The function, peakWidth_dragann, receives the whole histogram and the index
2555 number (maximum(1)) of the peak for which the value c is needed, as arguments.
2556 For a specific peak, the function essentially searches for the point on the histogram
2557 that is about $\frac{1}{2}$ the size of the peak and that is furthest away from the peak being
2558 investigated (left and right of the peak). If the two sides (left and right) are
2559 equidistant from the peak, then the side with the smallest value is chosen ($> \frac{1}{2}$
2560 peak).

2561 Upon entry, it first initializes c to zero. Then it initializes the index values left, xL and
2562 right, xR as index-1 and index+1, respectively (these will be used in a loop,
2563 described below). It next checks whether the n^{th} peak is the first or last value in the
2564 histogram and treats it as a special case.

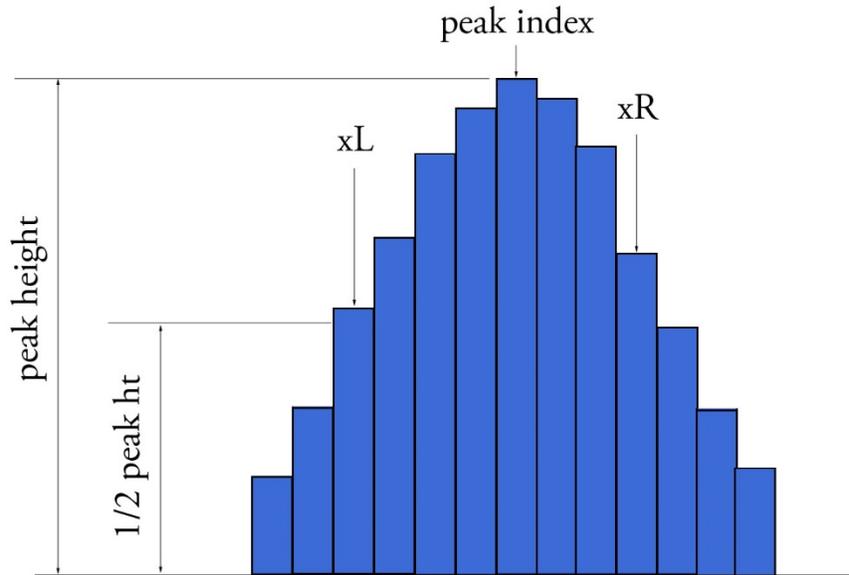
2565 At initialization, first and last histogram values are treated as follows:

2566 If first bin of histogram (peak = 1), set left = 1 and xL = 1.

2567 If last bin of histogram, set right = m and xR = m , where m is the final index of the
2568 histogram.

2569 Next, a search is made to the left of the peak for a nearby value that is smaller than
2570 the peak value, but larger than half of the peak value. A while-loop does this, with
2571 the following conditions: (a) left > 0, (b) histogram value at left is \geq half of histo
2572 value at peak and (c) histo value at left is \leq histo value at peak. When these
2573 conditions are all true, then xL is set to left and left is decremented by 1, so that the
2574 test can be made again. When the conditions are no longer met (i.e., we've moved to
2575 a bin in the histogram where the value drops below half of the peak value), then the
2576 program breaks out of the while loop.

2577 This is followed by a similar search made upon values to the right of the peak. When
2578 these two while-loops are complete, we then have the index numbers from the
2579 histogram representing bins that are above half the peak value. This is shown in
2580 Figure A.2.



2581

2582 **Figure A.2.** Schematic representation of a histogram showing xL and xR parameters
2583 determined by the function `peakWidth_dragann`.

2584 A test is made to determine which of these is furthest from the middle of the peak. In
2585 Figure A.2, xL is furthest away and the variable x is set to equal xL . The histogram
2586 “height” at x , which we call V_x , is used (as well as x) in an inversion of Equation A.2
2587 to solve for c :

2588
$$c = \sqrt{\frac{-(x-b)^2}{2 \ln\left(\frac{V_x}{a}\right)}} \quad (\text{A.4})$$

2589 The function, `peakWidth_dragann`, now returns the value of c and control returns to
2590 the function, `mainGaussian_dragann`.

2591 The `mainGaussian_dragann` function then picks-up with a test on whether the
2592 returned value of c is zero. If so, then use a value of 4, which is based on an *a priori*
2593 understanding that c usually falls between 4 and 6. If the value of c is not zero, then
2594 add 0.5 to c .

2595 At this point, we have the $[a,b,c]$ values of the Gaussian for the n^{th} peak. Based on
2596 these values, the Gaussian curve is computed (via Equation A.2) and it is removed
2597 (subtracted) from the current histogram (and put into a new variable called
2598 `newWave`).

2599 After a Gaussian curve is removed from the current histogram, the following peak
2600 width calculations could potentially have a V_x value less than 1 from a . This would

2601 cause the width, c , to be calculated as unrealistically large. Therefore, a check is put
2602 in place to determine if $a - V_x < 1$. If so, V_x is set to a value of $a - 1$.

2603 *Numeric Optimization Steps*

2604 The first of the optimization steps utilizes a Full Width Half Max (*FWHM*) approach,
2605 computed via

2606
$$FWHM = 2c\sqrt{2\ln 2} \tag{A.5}$$

2607 A left range, L_r , is computed by $L_r = \text{round}(b - FWHM/2)$. This tested to make sure it
2608 doesn't go off the left edge of the histogram. If so, then it is set to 1.

2609 Similarly, a right range, R_r , is computed by $R_r = \text{round}(b + FWHM/2)$. This is also tested
2610 to be sure that it doesn't go off the right edge of the histogram. If so, then it is set to
2611 the index value for the right-most edge of the histogram.

2612 Using these new range values, create a temporary segment (between L_r and R_r) of
2613 the newWave histogram, this is called errorWave. Also, set three delta parameters
2614 for further optimization:

2615 DeltaC = 0.05; DeltaB = 0.02; DeltaA = 1

2616 The temporary segment, errorWave is passed to the function checkFit_dragann,
2617 along with a set of zero values having the same number of elements as errorWave,
2618 the result, at this point, is saved into a variable called oldError. The function,
2619 checkFit_dragann, computes the sum of the squares of the difference between two
2620 histogram segments (in this case, errorWave and zeros with the same number of
2621 elements as errorWave). Hence, the result, oldError, is the sum of the squares of the
2622 values of errorWave. This function is applied in optimization loops, to refine the
2623 values of b and c , described below.

2624 *Optimization of the b-parameter.* The do-loop operates at a maximum of 1000 times.
2625 It's purpose is to refine the value of b , in 0.02 increments. It increments the value of
2626 b by DeltaB, to the right, and computes a new Gaussian curve based on $b + \Delta b$, which
2627 is then removed from the histogram with the result going into the variable
2628 newWave. As before, checkFit_dragann is called by passing the range-limited part of
2629 newWave (errorWave) and returning a new estimate of the error (newError) which
2630 is then checked against oldError to determine which is smaller. If newError is \geq
2631 oldError, then the value of b that produced oldError is retained, and the testing loop
2632 is exited.

2633 *Optimization of the c-parameter.* Now the value of c is optimized, first to the left,
2634 then to the right. It is performed independently of, but similarly, to the b -parameter,
2635 using do-loops with a maximum of 1000 passes. These loops increment (to right) or
2636 decrement (to left) by a value of 0.05 (DeltaC) and use checkFit_dragann to, again,

2637 check the quality of the fit. The loops (right and left) kick-out when the fit is found to
2638 be smallest.

2639 The final, optimized Gaussian curve is now removed (subtracted) from the
2640 histogram. After removal, a statement “corrects” any histogram values that may
2641 drop below zero, by setting them to zero. This could happen due to any mis-fit of the
2642 Gaussian.

2643 The n^{th} loop is concluded by examining the peaks remaining in the histogram
2644 without the peak just processed by sending the n^{th} -residual histogram back into the
2645 function findpeaks_dragann. If the return of peak index numbers from
2646 findpeaks_dragann reveals more than 1 peak remaining, then the index numbers for
2647 peaks that meet these three criteria are retained in an array variable called these:

- 2648 1. The peak must be located above $b(n)-2*c(n)$, and
 - 2649 2. The peak must be located below $b(n)+2*c(n)$, and
 - 2650 3. The height of the peak must be $< a(n)/5$.
- 2651

2652 The peaks meeting all three of these criteria are to be eliminated from further
2653 consideration. What this accomplishes is eliminate the nearby peaks that have a size
2654 lower than the peak just previously analyzed; thus, after their elimination, only
2655 leaving peaks that are further away from the peak just processed and are
2656 presumably “real” peaks. The n^{th} iteration ends here, and processing begins with the
2657 revised histogram (after having removed the peak just analyzed).

2658

2659 **Gaussian Rejection**

2660 The function mainGaussian_dragann returns the $[a,b,c]$ parameters for the ten
2661 highest peaks from the original histogram. The remaining code in dragann examines
2662 each of the ten Gaussian peaks and eliminates the ones that fail to meet a variety of
2663 conditions. This section details how this is accomplished.

2664 First, an approximate area, $\text{area1}=a*c$, is computed for each found peak and b , for all
2665 ten peaks, being the index of the peaks, are converted to an actual value via
2666 $b+\text{min}(\text{numptsinrad})-1$ (call this allb).

2667 Next, a rejection is made for all peaks that have any component of $[a,b,c]$ that are
2668 imaginary (Matlab isreal function is used to confirm that all three components are
2669 real, in which case it passes).

2670 To check for a narrow noise peak at the beginning of the histogram in cases of low
2671 noise rates, such as during nighttime passes, a check is made to first determine if the
2672 highest Gaussian amplitude, a , within the first 5% of the histogram is $\geq 1/10 * \text{the}$

2673 maximum amplitude of all Gaussians. If so, that peak's Gaussian width, c , is checked
2674 to determine if it is ≤ 4 bins. If neither of those conditions are met in the first 5%,
2675 the conditions are rechecked for the first 10% of the histogram. This process is
2676 repeated up to 30% of the histogram, in 5% intervals. Once a narrow noise peak is
2677 found, the process breaks out of the incremental 5% histogram checks, and the
2678 noise peak values are returned as $[a0, b0, c0]$.

2679 If a narrow noise peak was found, the remaining peak area values, $area1$ ($a*c$), then
2680 pass through a descending sort; if no narrow noise peak was found, all peak areas go
2681 through the descending sort. So now, the $[a,allb,c]$ -values are sorted from largest
2682 "area" to smallest, these are placed in arrays $[a1, b1, c1]$. If a narrow noise peak was
2683 found, it is then appended to the beginning of the $[a1, b1, c1]$ arrays, such that $a1 =$
2684 $[a0 a1]$, $b1 = [b0 b1]$, $c1 = [c0 c1]$.

2685 In the case that a narrow noise peak was not found, a test is made to check that at
2686 least one of the peaks is within the first 10% of the whole histogram. It is done
2687 inside a loop that works from peak 1 to the number of peaks left at this point. This
2688 loop first tests whether the first (sorted) peak is within the first 10% of the
2689 histogram; if so, then it simply kicks out of the loop. If not, then it places the loop's
2690 current peak into a holder ($ihold$) variable, increments the loop to the next peak and
2691 runs the same test on the second peak, etc. Here's a Matlab code snippet:

```
2692     inds = 1:length(a1);  
2693     for i = 1:length(b1)  
2694         if b1(i) <= min(numptsinrad) + 1/10*max(numptsinrad)  
2695             if i==1  
2696                 break;  
2697             end  
2698             ihold = inds(i);  
2699             for j = i:-1:2  
2700                 inds(j) = inds(j-1);  
2701             end  
2702             inds(1) = ihold;  
2703             break  
2704         end  
2705     end
```

2706

2707 The j -loop expression gives the $init_val:step_val:final_val$. The semi-colon at the end
2708 of statements causes Matlab to execute the expression without printout to the user's
2709 screen. When this loop is complete, then the indexes ($inds$) are re-ordered and
2710 placed back into the $[a1,b1,c1]$ and $area1$ arrays.

2711 Next, are tests to reject any Gaussian peak that is entirely encompassed by another
2712 peak. A Matlab code snippet helps to describe the processing.

```
2713     % reject any gaussian if it is fully contained within another  
2714     isR = true(1,length(a1));  
2715     for i = 1:length(a1)  
2716         ai = a1(i);
```

```
2717     bi = b1(i);
2718     ci = c1(i);
2719     aset = (1-(c1/ci).^2);
2720     bset = ((c1/ci).^2*2*bi - 2*b1);
2721     cset = -(2*c1.^2.*log(a1/ai)-b1.^2+(c1/ci).^2*bi^2);
2722     realset = (bset.^2 - 4*aset.*cset >= 0) | (a1 > ai);
2723     isR = isR & realset;
2724 end
2725 a2 = a1(isR);
2726 b2 = b1(isR);
2727 c2 = c1(isR);
```

2728

2729 The logical array isR is initialized to all be true. The i-do-loop will run through all
2730 peaks. The computations are done in array form with the variables aset,bset,cset all
2731 being arrays of length(a1). At the bottom of the loop, isR remains “true” when
2732 either of the conditions in the expression for realset is met (the single “|” is a logical
2733 “or”). Also, the nomenclature, “.*” and “.^”, denote element-by-element array
2734 operations (not matrix operations). Upon exiting the i-loop, the array variables
2735 [a2,b2,c2] are set to the [a1,b1,c1] that remain as “true.” [At this point, in our test
2736 case from channel 43 of East-AK Mable flight on 20140730 @ 20:16, six peaks are
2737 still retained: 18, 433, 252, 33, 44.4 and 54.]

2738 Next, reject Gaussian peaks whose centers lay within 3σ of another peak, unless only
2739 two peaks remain. The code snippet looks like this:

```
2740     isR = true(1, length(a2));
2741     for i = 1:length(a2)
2742         ai = a2(i);
2743         bi = b2(i);
2744         ci = c2(i);
2745         realset = (b2 > bi+3*ci | b2 < bi-3*ci | b2 == bi);
2746         realset = realset | a2 > ai;
2747         isR = isR & realset;
2748     end
2749     if length(a2) == 2
2750         isR = true(1, 2);
2751     end
2752     a3 = a2(isR);
2753     b3 = b2(isR);
2754     c3 = c2(isR);
```

2755

2756 Once again, the isR array is initially set to “true.” Now, the array, realset, is tested
2757 twice. In the first line, one of three conditions must be true. In the second line, if
2758 realset is true or $a2 > ai$, then it remains true. At this point, we’ve pared down, from
2759 ten Gaussian peaks, to two Gaussian peaks; one represents the noise part of the
2760 histogram; the other represents the signal part.

2761 If there are less than two peaks left, a thresholding/histogram error message is
2762 printed out. If the lastTryFlag is not set, DRAGANN ends its processing and an empty
2763 IDX value is returned. The lastTryFlag is set in the preprocessing function which

2764 calls DRAGANN, as multiple DRAGANN runs may be tried until sufficient signal is
2765 found.

2766 If there are two peaks left, then set the array [a,b,c] to those two peaks. [At this
2767 point, in our test case from channel 43 of East-AK Mable flight on 20140730 @
2768 20:16, the two peaks are: 18 and 433.]

2769

2770 **Gaussian Thresholding**

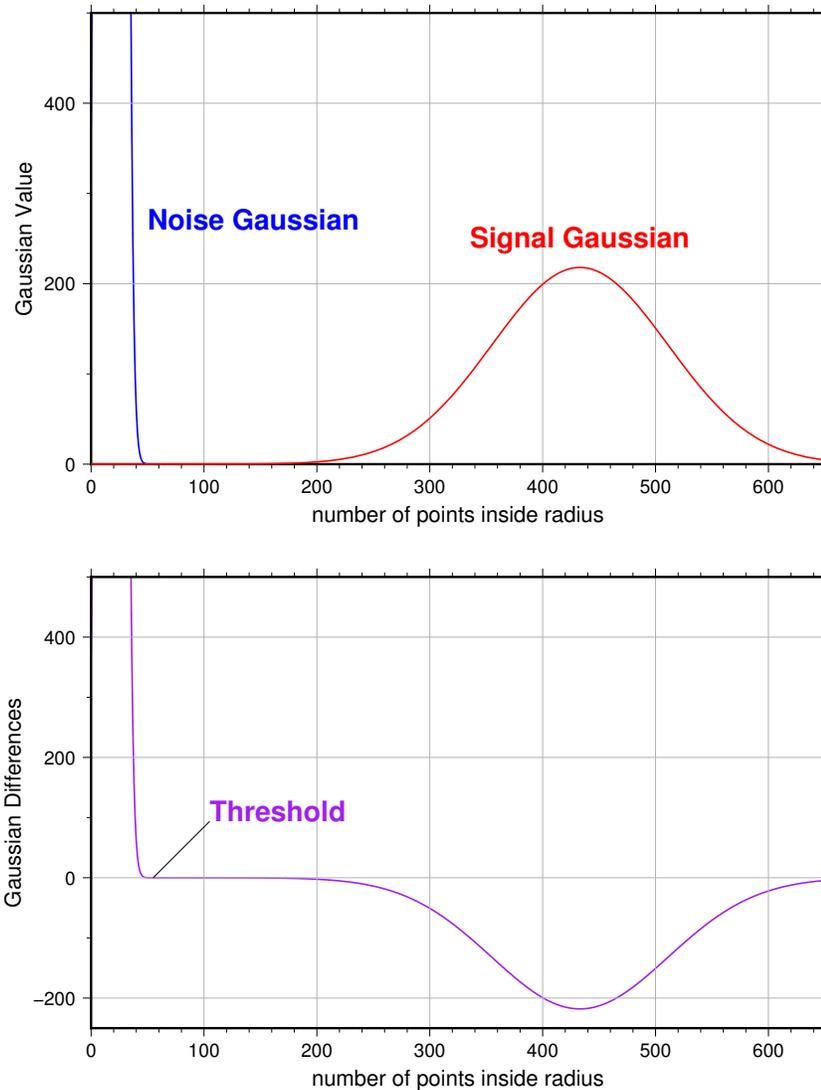
2771 With the two Gaussian peaks identified as noise and signal, all that is left is to
2772 compute the threshold value between the Gaussians.

2773 An array of xvals is established running from min(numptsinrad) to
2774 max(numptsinrad). In our example, xvals has indices between 0 and 653. For each
2775 of these xvals, Gaussian curves (allGauss) are computed for the two Gaussian peaks
2776 [a,b,c] determined at the end of the previous section. This computation is performed
2777 via a function called gaussmaker which receives, as input, the xvals array and the
2778 [a,b,c] parameters for the two Gaussian curves. An array of heights of the Gaussian
2779 curves is returned by the function, computed with Equation A.2. In Matlab, the
2780 allGauss array has dimension 2x654. An array, noiseGauss is set to be equal to the
2781 1st column of allGauss.

2782 An if-statement checks whether the b array has more than 1 element (i.e., consisting
2783 of two peaks), if so, then nextGauss is set to the 2nd column of allGauss, and a
2784 difference, noiseGauss-nextGauss, is computed.

2785 The following steps are restricted to be between the two main peaks. First, the first
2786 index of the absolute value of the difference that is near-zero (defined as 1e-8) is
2787 found, if it exists, and put into the variable diffNearZero. This is expected to be found
2788 if the two Gaussians are far away from each other in the histogram.

2789 Second, the point (i.e., index) is found of the minimum of the absolute value of the
2790 difference; this index is put into variable, signchanges. This point is where the sign
2791 changes from positive to negative as one moves left-to-right, up the Gaussian curve
2792 differences (noise minus next will be positive under the peak of the noise curve, and
2793 negative under the next (signal) curve). Figure A.3 (top) shows the two Gaussian
2794 curves. The bottom plot shows their differences.



2795

2796 **Figure A.3.** Top: two remaining Gaussian curves representing the noise (blue) and
2797 signal (red) portions of the histogram in F1gure A.1. Bottom: difference noise –
2798 signal of the two Gaussian curves. The threshold is defined as the point where the
2799 sign of the differences change.

2800 If there is any value stored in diffNearZero, that value is now saved into the variable
2801 threshNN. Else, the value of the threshold in signchanges is saved into threshNN,
2802 concluding the if-statement for b having more than 1 element.

**ICESat-2 Algorithm Theoretical Basis Document for Land-Vegetation
Along-Track Products (ATL08)
Release 002**

2803 An else clause ($b \neq 1$), merely sets threshNN to $b+c$, i.e., 1-standard deviation away
2804 from mean of the (presumably) noise peak.

2805 The final step is mask the signal part of the histogram where all indices above the
2806 threshNN index are set to logical 1 (true). This is applied to the numptsinrad array,
2807 which represents the photon cloud. After application, dragann returns the cloud
2808 with points in the cloud identified as “signal” points.

2809 The Matlab code has a few debug statements that follow, along with about 40 lines
2810 for plotting.

2811

2812 **References**

2813 Goshtasby, A & W. D. O’Neill, Curve Fitting by a Sum of Gaussians, *CVGIP: Graphical*
2814 *Models and Image Processing*, V. 56, No. 4, 281-288, 1994.