ICESat-2
Algorithm Theoretical Basis Document
for the Atmosphere

Part II:

Detection of Atmospheric Layers and Surface Using a Density-Dimension Algorithm

Geomath Code Version v114.0
ASAS Code Release v5.3
ATLAS Data Product ATL09

Ute Herzfeld
Stephen Palm
David Hancock

ute.herzfeld@colorado.edu

v11.0
6 February 2020

Author's addresses

Ute C. Herzfeld ute.herzfeld@colorado.edu
Geomathematics, Remote Sensing and Cryospheric Sciences
Department of Electrical, Computer and Energy Engineering
University of Colorado Boulder
Boulder, Colorado 80309-0574
U.S.A.

Stephen P. Palm stephen.p.palm@nasa.gov
Science Systems and Applications Inc. (SSAI)
NASA Goddard Space Flight Center
Mesoscale Atmosphere Processes Lab, Code 612
U.S.A.

David W. Hancock david.w.hancock@nasa.gov
Cryospheric Sciences Laboratory, Code 615.W
National Aeronautics and Space Administration
Goddard Space Flight Center
Wallops Flight Facility
Bldg. N-159, Room E-218
Wallops Island, VA 23337
U.S.A.

Contents

0 Versions, Source References and Change Logs **9**

0.1 Citation 9

0.2 Version v11.0, Source Reference and Change Log 9

0.3 Version v10.0, Source Reference and Change Log 12

0.4 Versions v1.0, v2.0, Source References 15

0.5 Version v3.0., Source Reference 15

0.6 Version v5.0, Source Reference 15

0.7 Version v6.0, Source Reference and Change Notes 15

0.8 Version v7.0, Source Reference and Change Log 16

0.9 Version v7.1, Source Reference and Change Log 16

0.10 Versions v7.x, Source References and Change Log Compared to v6.0 19

0.11 Version v8.0, Source Reference and Change Log 21

0.12 Version v8.1, Source Reference and Change Log 23

0.13 Version v9.0, Source Reference and Change Log 24

1 Data **29**

1.1 Data Format 29

2 Mathematical Concepts of the Density-Dimension Algorithm **31**

2.1 Background and Motivation of the Density-Dimension Algorithm 31

2.2 Main Mathematical Concepts 34

M.1 Radial Basis Function 34

M.2 Anisotropy Norm 35

M.3 Density Centers 36

M.4 Density-Dimension and Application as a Noise Filter . . . 39
M.5 Discrimination of Optically Thick and Optically Thin Clouds and other Atmospheric Layers . . . 39
M.6 Removal of Small Clusters and Determination of Final Cloud Mask . . . 40
M.7 Determination of Atmospheric Layers and Their Top and Bottom Boundaries 40
M.8 Density of a Column and Density of an Atmospheric Layer . . . 40

3 Algorithm Steps and Pseudocode: Density-Dimension Algorithm for ATLAS Atmosphere Data 42

3.0 Mask Handlers . . . 42
3.1 Step 1: Set Parameters and Load Data . . . 43
3.1.1 Set Parameters . . . 43
3.1.2 Load Data . . . 46
3.2 Step 2: Calculate Density . . . 50
3.2.1 Computation of Density — Overview of Steps . . . 50
3.2.1.1 Step 2.1: Read in Kernel Control Parameters . . . 50
3.2.1.2 Step 2.2: Calculate Dimensions of the Kernel . . . 52
3.2.1.3 Step 2.3: The Norm: Determination of the Distance that is Needed in the *RBF* . . . 53
3.2.1.4 Step 2.4: Kernel Calculation . . . 53
3.2.1.5 Step 2.5: Normalization of the Kernel Values . . . 53
3.2.2 Description . . . 54
3.3 Step 3: Using Density as a Dimension: A Density-Based Automatically-Adapting Noise Filter . . . 62
3.3.1 Creating Downsampled Density Arrays (Method A) . . . 63
3.3.2 Threshold Determination (Method A) . . . 66

3.3.3 Application of a Binary Matrix to Outline Cloud Areas (=First Approximation): Cloud Boundary/ Cloud Area Determination (Method A) 70
3.3.4 Method B for Auto-Adaptive Determination of Thresholds Using the Density-Dimension Algorithm 73
3.3.5 Application of Thresholds to Derive First Cloud Mask (Binary Matrix), Method B 75
3.3.6 Synthesis of Methods A and B 77
3.3.6.1 Where do the Differences Occur? 77
3.3.6.2 Density Field Calculation 77
3.3.6.3 Threshold Determination 79
3.3.7 Quantile Calculation 87
3.4 Step 4: Removal of Small Clusters: Determination of Cloud Areas - Final 90
3.5 Step 5: Output Data in Cloud Area (Cloud Mask) 98
3.6 Step 6: Layer Boundaries (Top/ Bottom) 99
3.7 Step 7: Layer Density, Density Sum per Vertical Profile and Other Derived Parameters 117
3.8 Flow of Algorithm Steps 118
3.9 Adjustable Parameters 123
3.10 Running Density Twice 127
3.10.1 Concept 127
3.10.2 Code for Running Density Twice 133
3.10.3 Description of Mask Handler 134

4 Application to 2012 MABEL Data **136**

5 Validation **139**

6 Analysis of 2013 M-ATLAS Data **142**

6.1 Correction . . . 142

6.2 Application: Analysis of a Data Set with Three Different Types of Conditions . . . 148

7 Sensitivity Studies (for 2013 M-ATLAS Data) 154

7.1 Sensitivity Studies for Single-Density Runs . . . 154

7.2 Sensitivity Studies for Double-Density Runs . . . 156

8 Analysis of GLAS-Data-Based Simulated ICESat-2 Data (2016 Version) 157

8.1 Simulation . . . 157

8.2 Analysis . . . 160

8.3 Validation . . . 174

9 Sensitivity Studies for GLAS-Data-Based Simulated ICESat-2 Data (2016 Version) 178

10 Sensitivity Studies for Analysis of 2017-Oct Version of GLAS-based Simulated ATL04 Data 179

10.1 Summary, Motivation and Data Sets . . . 179

10.2 Single-Density Runs Versus Double-Density Runs . . . 180

10.3 Results and Consequences for Algorithm Applications: Running Density Twice, $t56$, $t64$ . . . 181

10.4 Sensitivity Studies for Single-Density Runs . . . 183

10.5 Sensitivity Studies for Double-Density Runs . . . 186

11 Quality Assessment 188

11.1 Summary . . . 188

11.2 For Data Dictionary . . . 188

11.3 Q/A Algorithm . . . 189

11.4 Q/A Plots . . . 191

11.5 Applications . . . 192

12 Testing **194**

12.1 Testing Steps . . . 195

12.2 Quantile Calculation – Changes . . . 213

13 First Layer Classifications: Surface and Blowing Snow **214**

13.1 Motivation for First Classification Algorithms . . . 214

13.2 Algorithm for Determination of Ground Surface from Atmospheric Data (CALIOP Test Version) . . . 216

13.3 Identification/Classification of Blowing Snow . . . 218

13.4 Implementation: Threading of Ground Surface Classification Into the Existing DDA-atmos . . . 219

14 Analysis of First ICESat-2 ATLAS Data After Launch and Sensitivity Study for ATLAS Atmosphere Data **222**

14.1 Experiment Setup, Data and Results for Parameter Implementation. t56. . . . 222

14.2 Sensitivity Study and Post-Launch Q/A . . . 224

15 Post-launch Q/A Considerations **226**

15.1 On the Range of Half-Gap Layer Confidence Values . . . 226

15.1.1 Background: NRB values and the DDA . . . 226

15.1.2 Data sets for used for testing (here) and submitted to playground (at Goddard) 227

15.1.3 Examination of existence of confidence outside of [0,1] . . . 227

16 Sensitivity Study for Pre-Release Data Version v950, Necessitated by Change in Background and NRB Calculation in ATL04 **230**

17 Day-Night-Twilight: Implementation of Three Sets of Parameters for DDA-atmos Dependent on Day-Time (Sun-Elevation Angle) 232

17.1 Summary . . . 232

17.2 Algorithm Change . . . 232

17.3 Determination of Default Parameter Sets for Day - Night - Twilight . . . 234

18 On Ground and Cloud — Algorithm for Surface-Height Determination and for a Cloud-Based Ground-Detection Flag 240

18.1 Motivation . . . 240

18.2 Cloud-Based Ground-Detection Flag (DDA-atmos): Algorithm . . . 241

18.3 Analysis Illustrating Ground Flag Detection Algorithm . . . 242

18.4 Algorithm for Ground Height Determination and a Cloud-Based Ground-Detection Flag (DDA-atmos) . . . 244

18.5 Whither to? . . . 248

18.6 Results . . . 249

18.7 Consequences/Applications of Ground Detection (2019-06-21) . . . 250

18.7.1 Definition of a Ground Mask (ground_mask), to be applied to NRB Values to Ignore Ground-Center-Bins in Kernel Calculation . . . 250

18.7.2 Classification of the Ground Layer, Using a Kernel-Height Sized Ground Mask (fat_ground_mask) for a Simplified Interim Solution . . . 252

18.7.3 Utilizing Layer Counting Rather Than Ground-Layer Classification as an Interim Solution . . . 254

19 Sensitivity Study to Optimize Parameters for the First Public Release of ICESat-2 Data Products (ASAS code v5.1; 951 data) 255

20 Thoughts about Layers and Detection Capability of ICESat-2 ATLAS: Chal-

lenges and Opportunities for Atmospheric Research **257**

20.1 Solution for the "bubbly regions" problem in aerosols and other tenuous layers . . . 257

20.2 Thoughts about layers and detection capability of ICESat-2 ATLAS 258

21 Coder's Corner and Known Issues in ATL09 Atmospheric Data Products (Related to ATBD Part II) **259**

21.1 Summary 259

21.2 Code changes implemented in ASAS v5.3 259

21.3 Code changes that are still outstanding or in progress 259

21.4 Open Problems 260

References **263**

Appendix **266**

A Variables calculated in this ATBD Atmosphere, Part II **266**

B Index **268**

C Abbreviations **271**

APPENDIX S: Sensitivity Studies with Figures **272**

7 Sensitivity Studies (for 2013 M-ATLAS Data) **272**

7.1 Sensitivity Studies for Single-Density Runs 272

7.2 Sensitivity Studies for Double-Density Runs 286

9 Sensitivity Studies for GLAS-Data-Based Simulated ICESat-2 Data (2016 Version) **294**

10 Sensitivity Studies for Analysis of 2017-Oct Version of GLAS-based Simulated

ATL04 Data **305**

10.1 Summary, Motivation and Data Sets 305

10.2 Single-Density Runs Versus Double-Density Runs 306

10.3 Results and Consequences for Algorithm Applications: Running Density Twice, $t56$, $t64$ 307

10.4 Sensitivity Studies for Single-Density Runs 309

10.5 Sensitivity Studies for Double-Density Runs 324

14 Analysis of First ICESat-2 ATLAS Data After Launch and Sensitivity Study for ATLAS Atmosphere Data **337**

14.1 Experiment Setup, Data and Results for Parameter Implementation. t56. 337

14.2 Sensitivity Study and Post-Launch Q/A 339

16 Sensitivity Study for Pre-Release Data Version v950, Necessitated by Change in Background and NRB Calculation in ATL04 **351**

19 Sensitivity Study to Optimize Parameters for the First Public Release of ICESat-2 Data Products (ASAS code v5.1; 951 data) **364**

Detection of Atmospheric Layers and Surface Using a Density-Dimension Algorithm (Code Version v114.0)

0 Versions, Source References and Change Logs

0.1 Citation

This document may be cited as Herzfeld et al. [2020].

Part I refers to *ICESat-2 Algorithm Theoretical Basis Document for the Atmosphere, Part I: Level 2 and Level 3 Data Products* [Palm et al., 2019].

0.2 Version v11.0, Source Reference and Change Log

The **ATBD Part II version v11.0 of 2020-02-04** [Herzfeld et al., 2020] uses

atbd.atmos.icesat2.20200206.v11.sensimove.nodraft.v6.pdf

and is based on Geomath developer code version v114.0 of of January 2020 for data analysis.

This ATBD is referenced as Herzfeld et al. [2020], for the full citation, see the reference section. This is the ATBD version that accompanies the third public release of ICESat-2 data products, which includes atmospheric data products ATL04 and ATL09.

The corresponding Part I is Palm et al. [2019], for the full citation, see the reference section. [Reference needs to be updated for public release March 2020 version.]

Changed Sections and New Sections in v11.0 (compared to v10.0)

Here, the changes in the document are listed, and for each new section, the implementation status is annotated in addition to the code change requirement list above.

(1) All sensitivity studies with figures have been moved to a new Appendix S, which is placed after the references and the previous Appendices A, B, C. The original sensitivity study sections are retained in their original places in the main document, so that section numbering is unchanged from the previous version.

(2) Changed section 13 on blowing snow and ground.

(3) Changed section 18 on a cloud-based ground flag and on ground detection (surface detection) using a new algorithm based on the DDA.

(4) Updated all python code included in the ATBD to the current geomath code version, v114.

(5) Added changes to section M.3, regarding neighborhood of the kernel and normalization.

(6) Threaded mask handler into the algorithm steps in section 3. To this end, added an introductory section "Mask Handler" in section 3, as section 3.0.

(7) Added an explicit description of the flow of mask handling to section (3.10).

(8) Added a note regarding normalization to section (3.2.2)

(9) Changed section (3.2.2): added note on explicit calculation of dot product.

(10) Added a note to section 3.10.1 regarding folding over of kernel. Folding of kernel was implemented in earlier code versions, but is not used any more.

(11) Updated references.

(12) Added David Hancock as coauthor.

Change Log for ATBD Atmosphere, Part II, v11.0 (compared to v10.0)

This version of the ATBD atmosphere is based on code version v114 (Jan 2020) of the Geomathematics, Remote Sensing and Cryospheric Sciences Laboratory at the University of Colorado Boulder.

ASAS Code Change Requirements and Completions

(1) Code changes implemented in ASAS v5.3

(1) Section (11) Quality assessment: Half-gap confidence has been implemented and verified.

(2) Flow of mask handling, described in section (3.10), has been implemented/ improved and verified.

(3) Neighborhood of the kernel and normalization in the density calculation have been updated.

(4) Folding over of kernel has been eliminated.

(2) Code changes that are still outstanding or in progress

Details are described in the section 21.

(1) Section 13 on blowing snow and ground has not been implemented yet.

(2) Changed section 18 on a cloud-based ground flag and on ground detection (surface detection) using a new algorithm based on the DDA. This entire block is in progress for v5.4.

(3) Components of mask handling, described in section (3.10), may have to tested and verified for density run 2.

0.3 Version v10.0, Source Reference and Change Log

Source Reference

The **ATBD Part II version v10.0 of 2019-05-06** [Herzfeld and Palm, 2019] uses

atbd.atmos.icesat2.20190506.v10.windex.nodraftv2.tex

and is based on Geomath developer code version v112.0 of of April 2019 for data analysis.

This ATBD is referenced as Herzfeld and Palm [2019], for the full citation, see the reference section. This is the ATBD version that accompanies the first public release of ICESat-2 data products, which includes atmospheric data products ATL04 and ATL09.

The corresponding Part I is [Palm et al., 2018a], for the full citation, see the reference section. [Reference needs to be updated for public release May 2019 version.]

Change Log for ATBD Atmosphere, Part II, v10.0 (compared to v9.0

Code Change Requirements for SIPS/ASAS

This version of the ATBD atmosphere is based on code version v112 (April 2019) of the Geomathematics, Remote Sensing and Cryospheric Sciences Laboratory at the University of Colorado Boulder.

(1) The density-dimension algorithm is designed to be run with a set of algorithm-specific parameters. The main code change in this version is the implementation of three parameter sets, one each for day-time, night-time and twilight conditions, to match the calculation of the input data, normalized radiometric backscatter (NRB) data dependent on sun-elevation angle, a change described in Part I and implemented recently. The replacement of the parameter set for the DDA-atmos by three parameter sets, specific for day/night/twilight, is already implemented in the operational code for atmospheric data products, ASAS code version v5.1.

(2) Section (11): The Q/A algorithm component introduced in an earlier version of this document is not yet implemented in ASAS code version 5.1 at time of this document (May 6, 2019). In addition to its primary functionality, the Q/A algorithm can be applied to avoid counting subtle variations in tenuous aerosol layers as layer boundaries. Note that the Q/A algorithm

has not been changed, but section (20) added.

(3) Section (13): New algorithm components first described in v9.0 include the first classifications of layers: ground surface and blowing snow. These have not yet been by SIPS/ASAS at time of this document (May 6, 2019). A new ground flag is developed and introduced in this document (v10.0), based on the ground detection algorithm. The ground-detection flag is described in Section (18).

(4) Section (20.1), titled "Solution for the "bubbly regions" problem in aerosols and other tenuous layers" needs to be finalized and implemented. This requires implementation of the Q/A measure first.

(5) Section (20.2) leads to the requirement to put **Combined Mask** onto the product.

(6) For more details on code change requirements, a new section (21) "Coder's Corner and Known Issues in ATL09 Atmospheric Data Products (Related to ATBD Part II)" is included.

Changed Sections and New Sections in v10.0 compared to v9.0

Here, the changes in the document are listed, and for each new section, the implementation status is annotated in addition to the code change requirement list above.

(1) Table 7, the parameter table for algorithm-specific parameters of the DDA-atmos is updated from table 7 on page 261 of ATBD part 2, v9.0 from 2018-12-20.

(2) New section (16): Sensitivity Study for Pre-Release Data Version v950, Necessitated by Change in Background and NRB Calculation in ATL04
— *Parameter change implemented in ASAS*

(3) New section (17): Day-Night-Twilight: Implementation of Three Sets of Parameters for DDA-atmos Dependent on Day-Time (Sun-Elevation Angle)
— *Already implemented in ASAS v5.1*

(4) New section (18): On Ground and Cloud - Development of a Ground-Detection Flag Based on the DDA-atmos
— *Needs to be implemented in ASAS*

(5) New section (19): Sensitivity Study to Optimize Parameters for the First Public Release of ICESat-2 Data Products (ASAS code v5.1; 951 data)

— *Parameter change implemented in ASAS for public release production in May 2019*

(6) New section (20): Thoughts about Layers and Detection Capability of ICESat-2 ATLAS: Challenges and Opportunities for Atmospheric Research

(7) New Subsection (20.1): Solution for the "bubbly regions" problem in aerosols and other tenuous layers

— *Needs to be finalized in v10.1 and implemented in ASAS*

(8) New Subsection (20.2): Thoughts about layers and detection capability of ICESat-2 ATLAS

— *Combined_Mask needs to be added to product; is described in section (3)*

(7) New section (21): Coder's Corner and Known Issues in ATL09 Atmospheric Data Products (Related to ATBD Part II)

— *This section includes three types of items: (1) Differences between the Geomath group's developer code and the ASAS v5.1 code for atmospheric data products*

(2) Problematic data situations in the current release, v951

(3) Open problems

(8) Source Code and Change Logs (Section 0): This section has been reorganized and now lists the information regarding the latest version, v10.0, first, and information about previous versions in following subsections.

(9) References: References have been updated.

(10) Appendix: The table numbers in the Appendix have been changed to reflect new tables in the added new sections.

0.4 Versions v1.0, v2.0, Source References

Versions v1.0, v2.0. The early versions v1.0, v2.0 of this document (Jan 2014, April 2014) describe the algorithm version v4, which is the version presented in the Atmosphere Algorithm Telecon 2013-08-09 (August 9, 2013), see presentation given by S. Palm (Reference: Palm, Yang, Herzfeld, Algo-telecon-pdf), and small updates in the code.

0.5 Version v3.0., Source Reference

The **ATBD version v3.0 of 2014-08-08** is based on code version v6 of May 2014. The changes are included in the pseudocode section already. This code is used in the 2012 MABEL data analysis (included in the ATBD version of 2014-10-08). This code is referred to as Method A in the pseudocode sections, wherever there are differences.

Version v4.0. The **ATBD version v4.0 of 2014-11-01** is based code version v103.0 of October 2014; this code is used for the new 2013 M-ATLAS data analysis. This is included in the pseudocode as well. This code is referred to as Method B in the pseudocode section, wherever there are differences to Method A. MABEL and M-ATLAS data have different characteristics, which motivate some changes in adjustable parameters and code.

0.6 Version v5.0, Source Reference

The **ATBD version v5.0 2015-06-04** uses
atbd.atmos.icesat2.20150604c.wpseudocode.tex and is based on v103.0 of October 2014 for data analysis, using improved parameter combinations for MABEL/ M-ATLAS data analysis. ATBD v5.0 of 2015-06-04 is the first version where part I and part II were compiled separately.

0.7 Version v6.0, Source Reference and Change Notes

The **ATBD version v 6.0 2015-10-31** Herzfeld et al. [2015] uses
atbd.atmos.icesat2.201501031.wpseudocode.modular.tex and is based on v105.0 of October 2015 for data analysis. This code version includes the method A/B synthesis, which integrates the two methods A and B that were developed for determination of the auto-adaptive threshold function for 2012 MABEL and 2013 M-ATLAS data analysis respectively. The integrated approach is described

here and should be the only approach that needs to be implemented by SIPS, because it is upward compatible with all previous algorithms. Previous descriptions are kept in this document version for redundancy and to allow recreation of analyses based on earlier experiments and data sets. The integrated method is applied to analysis of GLAS-data-based simulated ICESat-2 data sets which were created in 2015.

0.8 Version v7.0, Source Reference and Change Log

The **ATBD Part II version v7.0 of 2016-08-24** [Herzfeld and Palm, 2016b] uses atbd.atmos.icesat2.20160824.tex and is based on v106.0 of August 2016 for data analysis. The code version v106.0 is essentially the same as v105.0. Small differences include numerical implementation of kernel calculation, which yields identical results; see section (4.3.2) on "Calculate Density", which is now broken into steps. All python code listings have been sourced to code version v106.0. Additional code listings have been included for "Determination of Layer Boundaries" in section (4.3.7) and for " Running Density Twice" in section (4.3.10). All pseudo-code versions that need updating will be included shortly (and are not included here) - in ATBD Part II, v7.1. A new sensitivity study will be included in ATBD Part II, v7.1 (when instrument parameters will be more fully determined, especially an update of power is needed.) Current sensitivity studies are sufficiently up-to-date for SIPS code implementation, as determined with SIPS (David Hancock, 2016-August-18). The recent sensitivity studies and analyses, included in this document in section (9), use GLAS-data-based simulated ICESat-2 data sets which were created in 2015. Current work uses SIPS-produced ATL04 products which differ somewhat from the NRB data used here. As in all previous versions of this ATBD part II, previous descriptions are kept in this document version for redundancy and to allow recreation of analyses based on earlier experiments and data sets.

0.9 Version v7.1, Source Reference and Change Log

The **ATBD Part II version v7.1 of 2016-09-23** [Herzfeld and Palm, 2016a] uses atbd.atmos.icesat2.20160923.tex and is based on code version v106.0 of August 2016 for data analysis. The difference between ATBD Part II, v7.1 and ATBD Part II, v7.0 is this section, in which the changes between ATBD v6.0 and v7.x are documented (for v7.0, v7.1). Listing 27 (from v7.0) was deleted.

Algorithm. There are no algorithm changes in this ATBD Part II, compared to the last version v6.0 of the ATBD Part II.

Changes in the text concern some of the explanations of the implementation of the algorithm, as have resulted from discussion with the SIPS during implementation of the Density-Dimension Algorithm (DDA) described here (by Jesse Wimert, coordinated by David Hancock).

The code version v106.0 is essentially the same as v105.0. Small differences include numerical implementation of kernel calculation, which yields identical results; see section (4.3.2) on "Calculate Density", which is now broken into steps. (Specifically: There is a spot in the code where we used to divide by 30, then multiply by it later - by 30/280 (it was in there for symmetry reasons between x and y directions), since this appeared confusing, that multiplication was moved to a different spot in the code. This is NOT an algorithm change however, rather an expression of coding taste - see Listing 7 for the current version of the Compute-density function in v106.0).

Rounding: In the kernel calculation (Compute-density function, Listing 7) the type of rounding is "round" (round to the nearest integer), changed from rounding by using the "ceiling" function (round to the next highest integer) in code v105.0. This affects the calculation of kernel sizes in some cases. Resultant values for examples match those in the Table 2d to facilitate trouble-shooting during code implementation.

Algorithm description. The description of the algorithm step for density calculation (Step 2) has been improved and is now broken into sub-steps (2.1)-(2.5), see section (3.2) on "Calculate Density".

Code listings. All python code listings have be sourced to code version v106.0. Additional code listings have been included for " Running Density Twice" in section (4.3.10). All pseudo-code versions that need updating will be included shortly (and are not included here) - in ATBD Part II, v7.2.

Sensitivity Studies. The recent sensitivity studies and analyses, included in this document in section (4.9), use GLAS-data-based simulated ICESat-2 data sets which were created in 2015. Current work uses SIPS-produced ATL04 products which differ somewhat from the NRB data used here. A new sensitivity study is not needed, because current sensitivity studies are sufficiently up-to-date for SIPS code implementation, as determined by the authors in discussion with SIPS (David Hancock, 2016-August-18) - but see section on *Outlook*.

Compatibility with previous versions of the ATBD for atmospheric data products, Part II. As in all previous

versions of this ATBD part II, previous descriptions are kept in this document version for redundancy and to allow recreation of analyses based on earlier experiments and data sets. Sections not changed were scrutinized and found up-to-date.

0.10 Versions v7.x, Source References and Change Log Compared to v6.0

The table of contents changed according to other changes.

Here the sections on "Version and Source Reference" for v7.0 and on "Version, Source Reference and Change Log" for v7.1 have been included.

(M.2) *Parameters:* Anisotropy factors a_m for data units in meters in meters, a_{bin} for data units in bins (pixels). The anisotropy factor itself does not have a unit. [changed in v7.1].

1.1 Each profile consists of the sum of 400 laser shots with a vertical (bin) resolution of 30 m (more exactly, 29.9m as determined in August 2016).

Pseudo-code sections were deleted. They will be replaced.

All python code listings were replaced by listings pulled from code v106.0 (2016-08-17), this is referenced in the caption of each listing. Note that there are still listings from previous code versions in the document, e.g. in section 4.3.3. Necessity is apparent when reading those sections.

4.3.1 Step 1: Set Parameters and Load Data. Subsection (4.3.1.1) on "Set Parameters" was included and Listing 1 added, listings 2-5 updated to reflect all the data load options that were needed, as new data became available during the ICESat-2 project, data collection and algorithm and product development.

4.3.1 (see Change 2016-08-23: height bin 29.9m) - added as necessary wherever the old height of 30 m is mentioned, to avoid confusion.

[Change 2016-08-23:] The size of the height bin was determined to be 29.9m rather than 30m.

4.3.2 Step 2: Calculate Density. This entire section was re-written to create a more streamlined presentation of the algorithm and its implementation. New code listings were included. In subsection "Output", a figure for the more typical kernel from example (t8) is included in Figure 2a, in addition to the kernel from the example that is used throughout the section (now Fig. 2b). Example (t8) is the example that was used during the code implementation by SIPS (J. Wimert) and our group (Geomathematics CU Boulder) to ascertain that results from our two code versions match.

4.3.3 Step 3: Using Density as a Dimension: A Density-Based Automatically-Adapting Noise Filter. This section was carefully scrutinized during code implementation by SIPS and found up-to-date. The only part needed for implementation is the Method A/B synthesis (4.3.3.6) and following. However, it is nice to have the original methods A and B to help a reader understand the concepts. New listings were pulled from code v106.0.

4.3.4 Removal of Small Clusters: Determination of Cloud Areas - Final. Listings were updated using code v106.0. Pseudo-code was checked and left in the document.

4.3.5 Step 5: Output Data in Cloud Area no changes.

4.3.6 Step 6: Layer Boundaries (Top/ Bottom). The simple algorithm for determination of layer boundaries was not changed. Pseudo code included in Listing 27 in v7.0 and removed in v7.1.

4.3.10 Running Density Twice. Code for Running Density Twice was included from v106.0.

Throughout the document, a few typos and other small items were corrected throughout the document, and figure numbers and section references were adjusted as needed.

Acknowledgements. Discussions with David Hancock, Jeff Lee and Jesse Wimert during code implementation at the SIPS are equally appreciated and have resulted in improvements of the description.

References. Updated bibliography file (bib-file).

Outlook. The future version referred to as ATBD Part II, v7.1 in the "Version and Source Reference" section for ATBD Part II, v7.0 is now referred to as ATBD Part II, v7.2. A new sensitivity study will be included in ATBD Part II, v7.2 (when instrument parameters will be more fully determined by the engineering group, especially an update of power is needed.)

Pseudo-code sections will be replaced.

An algorithm part for Q/A will be implemented and tested.

0.11 Version v8.0, Source Reference and Change Log

Source Reference

The **ATBD Part II version v8.0 of 2017-11-17** [Herzfeld and Palm, 2017] uses atbd.atmos.icesat2.20171117.v8.tex and is based on code version v110.0 of November 2017 for data analysis.

It is referenced as Herzfeld and Palm [2017], for the full citation, see the reference section.

The corresponding Part I is [Palm et al., 2017a], for the full citation, see the reference section.

Change Log: Differences between Version 8.0 and Version 7.1

Note in version numbers. There is no version (v7.2) for Part II, instead, there is this current new version, (v8.0), which includes a sensitivity study.

Section (4.4) of (v7.1) deleted. Section (4.4) was titled "Useful algorithm components from previous versions" and included only "(4.4.1) Ratio Cluster Algorithm". With the design of the new algorithm for a confidence flag as a measure of layer detection quality (see section (4.11) in v8.0), the previous section (4.4) of (v7.2) becomes obsolete. Note that the deletion of section (4.4) has moved all following section numbers.

Data reference. Data referred to as "GLAS-data-based simulated ICESat-2 data" in ATBD v7.x (2016) are now referred to as "GLAS-data-based simulated ICESat-2 data (2016 version)", to distinguish from "GLAS-data-based simulated ICESat-2 data (ATL04) of Oct-2017". Differences in the characteristics of these two data sets are described in section (4.10).

Section (4.3.6) is entirely rewritten. Algorithm update. Determination of layer boundaries. The algorithm for determination of layer boundaries, based on the final mask, has been improved. The algorithm follows the same basic rules for layer boundary determination.

New Section (4.3.3.7) Quantile Calculation. During testing of the code implementation by the SIPS in October 2017, we discovered that the algorithm used in quantile calculation as part of the threshold function can contribute significantly to the error in results between two different code implementations. Care needs to be taken when using a library function (python, fortran or any other language). Library functions typically only differ in the interpolation step between actually occurring values, but since near the threshold values used in the DDA

the density values are relatively scarce, this difference matters. The effect is illustrated in section (4.12) "Testing". The old and new algorithms are described Section 4.3.3.7.

New Section (4.10) Sensitivity Studies for Analysis of 2017-Oct Version of GLAS-based Simulated ATL04 Data. In this section, differences in the characteristics of "GLAS-data-based simulated ICESat-2 data (ATL04) of Oct-2017" data compared to "GLAS-data-based simulated ICESat-2 data (2016 version)" are described. A new sensitivity study is carried out to determine a set of algorithm-specific parameters for auto-adaptive analysis of ATL04 data (with Oct 2017 characteristics.). An important result is that the DDA-algorithm option "running density twice" is required to ascertain correct detection of different types of atmospheric layers during day-time and night-time conditions. As the application of the newly-developed Q/A measure "half-gap confidence flag" (see section (4.11) Quality Assessment) shows, the layer detection using the double-density runs with the parameter sets (t56) [and (t64)] yields throughout high confidences (mostly 0.8) and somewhat lower confidences where appropriate. – Why two parameter sets at this point? See Section (4.12) on Testing.

New Section (4.11) Quality Assessment. New Algorithm. An algorithm that quantifies confidence as a numerical value (with absolute value between 0 and 1) is introduced for quality assessment. Mathematical Q/A algorithm description, Q/A plots and applications are included as subsection. The algorithm to be used is termed "Half-gap confidence". In addition, the half-gap confidence is compared to an alternative "3-bin confidence". "Half-gap confidence" is a better measure than "3-bin confidence".

New Section (4.12) Testing. This section includes information on the process of testing the code implementation by the SIPS, comparison with the CU code and criteria for accepting code matches. This is work in progress.

New Section (4.13) Coder's Corner. Because "Testing" at time of writing of ATBD v8.0 (November 2017) is still in progress, an informal section "Coder's Corner" is added to facilitate picking up the testing process whenever possible at the SIPS. This section should be removed before the ATBD is passed on to NASA Headquarters.

0.12 Version v8.1, Source Reference and Change Log

Source Reference

The **ATBD Part II version v8.1 of 2018-03-21** [Herzfeld and Palm, 2018a] uses atbd.atmos.icesat2.20180321.v8.windex.v2.tex and is based on code version v110.0 of November 2017 for data analysis.

It is referenced as Herzfeld and Palm [2018a], for the full citation, see the reference section.

The corresponding Part I is [Palm et al., 2017a], for the full citation, see the reference section.

Change Log: Differences between Version 8.1 and Version 8.0

No algorithm changes. There are no algorithm changes in version v8.1 compared to v8.0 and the code is v110.0, the same code as used in v8.0.

Index with hyperlinks (Appendix A). Changes in this version concern improvements that make the ATBD more user-friendly: An index has been included, with hyperlinks from terms in the index to the locations in the text where the referenced terms are explained.

List of abbreviations (Appendix B). A list of commonly used abbreviations is included as Appendix B.

Section, figure and table numbering. This is the first version in which the sections are numbered within Part II, starting with section 1. In previous versions the entire part II was counted as section 4 of the ATBD Atmosphere {Part I, Part II} combined. This change is motivated by the fact that Part I, v7.3 has 7 sections, not 3 as previously. For example, section *4.1 Data* of v8.0 is now section *1 Data* of v8.1. The order of the sections remained the same. Numbering of figures and tables is changed accordingly (the digit 4 has been dropped).

Typos. Some typographical errors were corrected.

Table of content has changed according to other changes made.

References. The bibliography has been updated.

0.13 Version v9.0, Source Reference and Change Log

Source Reference

The **ATBD Part II version v9.0 of 2018-12-20** [Herzfeld and Palm, 2018b] uses atbd.atmos.icesat2.201812201.v9.windex.v2.nodraft.tex and is based on code version v111.0 of December 2018 for data analysis.

It is referenced as Herzfeld and Palm [2018b], for the full citation, see the reference section.

The corresponding Part I is [Palm et al., 2018a], for the full citation, see the reference section.

Change Log for ATBD Atmosphere, Part II, v9.0 (first post-launch version of the ATBD) compared to v8.1

Code Change Requirements for SIPS/ASAS

This version of the ATBD atmosphere is based on code version v111 (December 2018) of the Geomathematics, Remote Sensing and Cryospheric Sciences Laboratory at the University of Colorado Boulder. There are no code changes required in this version for the existing DDA-atmos as implemented by SIPS/ASAS.

New algorithm components described in v9.0 include the first classifications of layers: ground surface and blowing snow. The Q/A algorithm component introduced in an earlier version of this document (see, section 11) is not yet implemented by SIPS/ASAS at time of this document (Dec. 20, 2018).

Changed Sections

1. **Section (1) Data** was updated to include post-launch ICESat-2 ATLAS data.

2. In **section (3.10)** on "Running density twice", the following note was added:
   Note (2018-Dec-19). The option of "running density twice", developed in 2013, was not deemed necessary for optimized data analysis until 2017/2018 and is now adopted as a component of the operational code for (post-launch) ICESat-2 data. This is based on a sensitivity study of 2017-Oct Version of GLAS-based simulated ATL04 data (section 10) and a sensitivity study of the first post-launch data collected with the ICESat-2 ATLAS instrument (section 15). As a

side note, the re-institution of "running density twice" in the operational code for post-launch data shows that it is a good idea to keep old algorithm components in the ATBD.

3. In **section 10**, we highlighted the results of the sensitivity study by introducing section titles, to facilitate identification of essential consequences for code applications after launch.

   (10.1) Summary, Motivation and Data Sets

   (10.2) Results and Consequences for Algorithm Applications: Running Density Twice, $t56$, $t64$

   Previous subsections 10.1 ff are now 10.3 ff. A few sentences have been streamlined and it is indicated that the 2017=oct simulated data are the last pre-launch data used in algorithm development.

4. **New section (13) First Layer Classifications: Surface and Blowing Snow.** This includes motivation (13.1), algorithms for identification of ground surface (13.2) and blowing snow (13.3) and implementation notes (13.4).

5. **New section (14) Analysis of first ICESat-2 ATLAS Data after launch and sensitivity study for ATLAS atmosphere data.**

6. **New section (15) Post-Launch Q/A Considerations**

7. Section **Coder's Corner** remains the last section of this ATBD, part II, and is now **section 16.**

8. **Updated reference list.** Includes blowing snow references and ATBD updates.

9. **Appendix A** was added, listing all input and output variables described in this ATBD Atmosphere, Part II. The Appendices A and B, version 8.1, are now appendices B and C, version v9.0.

10. The **Table of Contents** changed according to other changes.

Preambula. The description of the density-dimension algorithm in part II is organized in the following sections:

1 Data

2 Mathematical Concepts of the Algorithm

3 Algorithm Steps

4 Application to 2012 MABEL data

5 Validation

6 Analysis of 2013 M-ATLAS Data

7 Sensitivity Studies (for 2013 M-ATLAS Data)

8 Analysis of GLAS-Data-Based Simulated ICESat-2 Data (2016 Version)

9 Sensitivity Studies (for GLAS-Data-Based Simulated ICESat-2 Data (2016 Version)

10 Sensitivity Studies for Analysis of 2017-Oct Version of GLAS-based Simulated ATL04 Data

11 Quality Assessment

12 Testing

13 First Layer Classifications: Surface and Blowing Snow

14 Analysis of First ICESat-2 ATLAS Data After Launch and Sensitivity Study for ATLAS Atmosphere Data

15 Post-Launch Q/A Considerations

16 Coder's Corner

Appendix

Section *1 Data* describes the data set that is used for the current version of the algorithm.

Section *2 Mathematical Concepts of the Algorithm* describes any non-standard mathematical concept that is used in the algorithm, as a reference.

Section *3 Algorithm Steps* describes the algorithm in a step-by-step form, so that a programmer can work along these steps and implement the algorithm. Our algorithm (the prototype version) is implemented in modular form. Pseudo-code and essential sections of the original python code are

included in section 3.

versions of the atmosphere algorithm (but not in the current version) and which may be useful in future forms of the algorithm, depending on the results of future MABEL data collections.

Section *4 Application to 2012 MABEL Data* is included to demonstrate the algorithm steps and includes figures. This section uses MABEL Data from 2012 observations.

Section *5 Validation* uses CPL (Cloud Physics Lidar) data for validation of the algorithm. The location of the clouds seen in the CPL data were not known to the developers of the algorithm and software.

Section *6 Analysis of 2013 M-ATLAS Data* includes a correction algorithm and applications of an improved version of the auto-adaptive density-dimension algorithm to a data set including different atmospheric and noise conditions and different types of atmospheric layers.

Section *7 Sensitivity Studies (for 2013 M-ATLAS Data)* demonstrates sensitivity of analysis results to changes in the algorithm"fixed" parameters to better prepare for possibly different characteristics in ICESat-2 ATLAS data as may be collected post-launch.

In Section *8 Analysis of GLAS-Data-Based Simulated ICESat-2 Data (2016 Version)*, simulated ICESat-2 data based on GLAS data from ICESat, rather than on MABEL data, are analyzed, which leads to use of different algorithm parameters.

Section *9 Sensitivity Studies (for GLAS-Data-Based Simulated ICESat-2 Data (2016 Version)* is the analog to section (7) and presents sensitivity of analysis results to changes in the algorithm parameters for GLAS-based ICESat-2 data.

Section *10 Sensitivity Studies for Analysis of 2017-Oct Version of GLAS-based Simulated ATL04 Data (Oct-2017 Version)* presents sensitivity of analysis results to changes in the algorithm parameters for GLAS-based Simulated ATL04 Data (October 2017 version)

Section *11 Quality Assessment* introduces confidence measures for layer detection.

In Section *12 Testing*, steps of algorithm implementation by the SIPS are summarized, results compared and criteria for acceptance of code matching derived.

In Section *13 First Layer Classifications: Surface and Blowing Snow*, algorithms for identification of the surface and blowing snow in atmospheric data and are introduced. These represent the first classification algorithms, based on the layer detection using the DDA-atmosphere.

Section *14 Analysis of First ICESat-2 ATLAS Data After Launch and Sensitivity Study for ATLAS Atmosphere Data* represents the beginning of post-launch ICESat-2 ATLAS data analysis and Quality Assessment (Q/A algorithm runs) The Satellite was launched on September 15, 2018. The Advanced Topographic Laser Altimeter System (ATLAS) is the single primary instrument aboard ICESat-2.

In Section *15 Post-Launch Q/A Considerations*, the Q/A algorithm is applied to ICESat-2 ATLAS data and studied in more detail.

Section *16 Coder's Corner* is introduced as an an informal section to facilitate picking up the testing process whenever possible at the SIPS. This section should be removed before the ATBD is passed on to NASA Headquarters.

1 Data

The ICESat2-atmosphere algorithm described here is written for analysis of ICESat-2 Advanced Topographic Laser Altimeter System (ATLAS) data and uses simulated ICESat-2 data, based on Multiple Altimeter Beam Experimental Lidar (MABEL) data collected in 2012 and 2013, and simulated ICESat-2 data, based on GLAS data, for algorithm development and demonstration in sections 1-13. These data include both day-time data and night-time data. Instrument effects may be different for ICESat-2 ATLAS data. Some parameters in the algorithms are changeable to allow for adjustments that may be needed after launch; these parameters are termed algorithm parameters.

The algorithm versions have been tested for the growing collection of theory-based simulated data, GLAS-data based simulations and MABEL-data based simulations, for day-time and night-time data and hence for various forms of simulated or observed noise levels.

First ICESat-2 ATLAS data are analyzed starting in section 14 after launch of the satellite on September 15, 2018.

The density-part of the atmosphere algorithm is written and described such that neighborhood definitions match the special format of the ICESat-2 atmosphere data, as summarized below, i.e. no further adaptation of the mathematical concepts to the atmosphere data is necessary when implementing these algorithms. The most notable difference between ICESat-2 atmosphere data and ICESat-2 Earth surface data is that for atmosphere, 400-shot sum data are recorded, binned horizontally and vertically, whereas for all Earth surfaces, individual photons are recorded.

1.1 Data Format

The format of the ATL04 data, described in Part 1, used for determination of clouds and other atmospheric layers and their boundaries is as follows:

As described in the introduction of this document, the level 0 (raw) data consist of 3 profiles generated from the 3 strong laser beams. The profiles range in height from 13.75 km above the local value of the DEM to 0.25 km below. Each profile consists of the sum of 400 laser shots with

a vertical (bin) resolution of 30 m (more exactly, 29.9m as determined in August 2016). There are a total of 467 bins in each of the 3 profiles, which are downlinked from the spacecraft at 25 Hz. Each bin contains the number of photon counts received for that height range over the 400 shot summing interval (equivalent to about 280 m along track). Nominal space-craft velocity is 7000m/s, laser repetition rate is 10KHz, which corresponds to 0.7 m along-track-distance per laser shot. Each profile is the sum of 400 shots, hence atmosphere data are recorded at 25Hz and each bin represents 280 meters along-track.

The input data set is output as a figure (data.png, Figure 1). Additional figures are given in sections 4 "Application to 2012 MABEL Data" and 6 "Analysis of 2013 M-ATLAS Data".

The input to the cloud layer detection algorithm is normalized relative backscatter (NRB) created and stored on the ATL04 product (Section 2). NRB is created by subtracting the background from the raw photon count data, multiplying the result by the square of the range from the satellite to the bin in question and dividing by the laser energy. As part of the ATL04 process, the NRB data are stored in a constant altitude frame that spans -1 to 20 km with respect to the ellipsoid (700 bins). Bins in this frame that do not contain data are given the value of -9999.

The data are stored as a 2-dimensional array. The size of this file is 700 bins in elevation and variable length in along-track distance. Bins are counted from top to bottom. There is one line (one record) per vertical profile, so that records can be appended as data get collected during the mission. ICESat-2 data will be provided in hdf05 format. In the hdf05 format, each profile is stored in fields 0-700 (0 on the left, 700 on the right side of the line). The number of valid bins is 467 and the indeces (bin numbers) of the top and bottom bin with a valid entry are provided in the ATL04 data records.

Note. In the older examples given in this ATBD version (examples created until 20140725), height ranges from -1000m to +14000 m relative to the onboard DEM and data are given as a 2-dimensional array with 500 bins in elevation and variable length in along-track direction.

Subnote. Notes like these are included throughout the document to facilitate reconstruction of analysis results during code implementation by the SIPS or anyone checking the pseudo/code or his/her own software implementation (troubleshooting).

2 Mathematical Concepts of the Density-Dimension Algorithm

2.1 Background and Motivation of the Density-Dimension Algorithm

The algorithm is aimed at detection of clouds and other atmospheric layers, such as blowing snow and aerosols, and their boundaries in ICESat-2 ATLAS data.

Satellite radar altimeters and laser altimeters, for example the Geoscience Laser Altimeter System (GLAS) aboard ICESat, used to apply the concept of pulse-limited altimetry, where a strong signal is transmitted [Zwally et al., 2005, Schutz et al., 2005]. To determine range, a Gaussian waveform is fitted to the received signal and the maximum of that waveform used to identify the two-way travel-time of the signal between the satellite and the surface of reflectance (the Earth's surface or a cloud layer). ICESat-2 will utilize a next-generation multi-beam micro-pulse photon-counting laser altimeter. This altimeter will transmit many pulses of low energy, which facilitates a higher repetition rate. On the receive side, reflections from every single photon will be recorded as discrete received events; this will include noise photons and signal photons. The determination of signal versus noise for a micro-pulse photon-counting laser altimeter hence requires a new mathematical concept that will take the role of the waveform analysis in classic pulse-limited altimetry. In pulse-limited altimetry, a data aggregation is given by the bundeling of the energy in the strong signal and the (generally working) separation of two different signals in time, whereas for micro-pulse photon-counting laser altimeter data a data aggregation needs to be performed mathematically. The data aggregation algorithm needs to be able to perform a separation of noise and signals for diffuse reflectors, such as clouds, regions of high aerosols, the Earth's surface and layers of blowing snow. The problem is mathematically ill-posed, especially for the case of day-time altimeter data, where high noise results from ambient light. The principal idea of the algorithm is that any reflector will result in a higher spatial density of received photons than the spatial density of photons recorded in other areas (background or noise areas).

The basic concept of the algorithm is the calculation of density for each recorded data point: That is, (a) for each photon for ice surface, vegetation and other land cover, or (b) for each point in the 2-dimensional matrix of recorded data (for atmosphere), i.e. for each (along-track profile, altitude)

interval. For atmosphere data, the intervals contain photon counts (or NRB values, as described in the previous sections of this ATBD). The data aggregation is performed by an application of the radial basis function (described in 2.2, M.1). Numerically, the radial basis function is applied as a multiplication between a weight matrix and the observed photon count values, as a moving-window operation. As the algorithm moves throughout the data set, the point or interval for which density is calculated is termed "density center" (see 2.2, M.3). The actual form of the RBF is controlled by three parameters: window size, anisotropy and standard deviation (which are interdependent). The anisotropy factor extends the search window farther in horizontal direction than in vertical direction. which matches the observation that layers have a larger horizontal then vertical extension. The calculation of density performs the data aggregation, on which all other algorithm steps are based. This RBF-density can be thought of as the counterpart of the waveform or histogram in pulse-limited altimetry, and we shall see that it is a powerful and versatile concept for micro-pulse photon-counting laser altimeter data analysis in general.

A main objective of the algorithm is to identify physically meaningful reflectors and distinguish them from background noise, artifacts and detector dark counts. Atmospheric layers include clouds, aerosol layers and blowing snow, and in addition, ground is included in the atmosphere data. To separate reflectors of interest (here: atmospheric layers) from noise regions, a threshold between density of reflectors and density of noise needs to be determined automatically. Because background conditions change, for instance due to surface reflectance and time of day, the threshold determination algorithm needs to adapt automatically to conditions. This is achieved by the concept of density calculated as an additional dimension, i.e. the solution for the best threshold will be searched in a space of a larger dimension; the algorithm is termed "Density-dimension algorithm (DDA)".

In the determination of a ground surface (ice sheets, sea ice, vegetation, land), a surface detection and classification algorithm can utilize the fact that the ground surface is a continuous reflector, which can either be searched for as a layer in the received signal or expected near a DEM. In contrast, such an assumption does not hold for atmospheric layers, which can be present "anywhere" within the lowest 15 km of the atmosphere and their position changes on a short time scale. This necessitates an automated determination of signal density thresholds without a-priori knowledge of location of a noise-box or noise layer, i.e. the algorithm needs to be independent of a-priori control parameters. These requirements are met by the auto-adaptive density-dimension algorithm.

There are several fine points to the implementation of the DDA, which will be described in the sequel: Mathematical concepts in section 2.2 and implementation in the pseudo-code section (3).

Resolution of results: A characteristic of the expected results from the ICESat-2 ATLAS instrument is that the photon-counts may be relatively low and often not exceed background values much and hence the gradient between density of optically thin clouds (such as high Cirrus clouds) or aerosols layers (from pollution or distant volcanic eruptions) to the surrounding atmosphere can be very small. For optically thin layers, this fact requires aggregation of data over a large neighborhood, to yield density values that allow to separate noise from atmospheric layers at all. As we shall see, not loosing layers is a challenge in analysis of ATLAS atmosphere data. For optically thick and possibly spatially well-confined narrow layers data aggregation over a large neighborhood is not needed (as enough points can be found in smaller neighborhoods), and also not desirable, because a larger window may introduce a larger smearing effect (depending on the coefficients in the weight matrix). In conclusion, there are two objectives which suggest different controls of algorithm parameters:

(1) Detection of atmospheric layers with small gradients to surrounding regions (small ratios of backscatter). Not loosing optically thin layers.

(2) Precise determination of layer boundaries, wherever possible, especially for optically thick and spatially narrow layers.

Both seemingly contrary goals can be met by running the DDA algorithm twice with different parameters, first with a smaller window and second with a larger window (and different sigma) and combing the resultant cloud masks (layer masks). The vertical resolution of results is the same as the vertical diameter of the window; however, since the weights taper to the outside of the search window, a much higher resolution than window size is generally achieved. The effect of applying the data aggregation using density is that smaller and weaker features become visible with distinction than in the raw data (see section 6 on validation).

2.2 Main Mathematical Concepts

In this subsection mathematical concepts are described that are utilized or specifically developed for the density-dimension algorithm for atmosphere and hence may go beyond commonly known mathematical concepts. Implementation of the algorithm is described in section (3).

(M.1) Radial Basis Function

The radial basis function is the basic mathematical concept used in the data aggregation for calculation of density. The data aggregation by density calculation forms the basis for all other algorithm steps.

A *radial basis function (rbf)* is a real-valued function whose value depends on distance from a *center* $c \epsilon \mathcal{D}$ for all x in a definition area $\mathcal{D}$

$$\Phi(x, c) = \Phi(\|x - c\|) \tag{1}$$

with respect to any norm $\|\cdot\|$. In the algorithm, we utilize a Gaussian radial basis function (letting $r = x - c$ and $s \epsilon \mathcal{R}$)

$$\Phi(r) = e^{-(\frac{r}{\sqrt{2}s})^2} \tag{2}$$

Visualized as a surface in $\mathcal{R}^3$, this rbf has the shape of (half) a Gaussian bell curve rotated around the location of a center $c \epsilon \mathcal{R}^2$. In the photon-data analysis, we have $c \epsilon \mathcal{R}^3$ and the surface is in $\mathcal{R}^4$. More formally, the Gaussian probablility density function is

$$f_{normpdf} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\frac{x-\mu}{\sqrt{2}\sigma})^2} \tag{3}$$

with standard deviation σ and mean μ of the population; replacing $\sigma = s$ and $\mu = 0$ yields eqn (4):

$$\Phi(r) = \sigma\sqrt{2\pi} f_{normpdf} \tag{4}$$

(see Herzfeld et al. [2014, 2017], Buhmann [2003]).

The radial basis function is especially useful for data aggregation for the photon-counting-laser-altimetry problem, because points close to the center point are given a high weight, and weights taper off towards the outside of the search window, following the Gaussian function. This property yields a weight matrix that enhances features (layers) of different sizes and thicknesses.

(M.2) Anisotropy Norm

Using an anisotropy norm is motivated by the notion that cloud layers (as seen in lidar data) have a tendency to extend more in the horizontal direction than in the vertical direction. When the anisotropy norm is combined with the radial basis function, points found in a horizontal direction from the center point are weighted higher than points found in a vertical direction. The following algorithm implements a matrix multiplication that is an affine transformation of the density function (the radial basis function) into a function of ellipsoidal shape. This is implemented by the following algorithm: The anisotropy norm is defined as

$$\|v\|_a = \|Av\|_2 \tag{5}$$

for any vector $v\epsilon\mathcal{R}^3$, with a transformation matrix

$$A = \begin{pmatrix} \frac{1}{a} & 0 & 0 \\ 0 & \frac{1}{a} & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{6}$$

where $a\epsilon\mathcal{R}$.

This is applied to the density centers c and all their neighboring points in eqn. (6) as

$$\|x - c\|_a = \|A(x - c)\|_2 \tag{7}$$

Points of the same *rbf* value $\Phi(\|x - c\|_a)$ are now located on an ellipsoid with axes (a,a,1) around the center point c and (half) Gaussian bell curves along each radial line. The density value $f_d(c)$ then reflects the above-mentioned tendency of cloud layers (as seen in lidar data) to have a larger horizontal than vertical extension (for $a > 1$ with a given in meters). In the 2-dimensional realization of the simulated data set, a transformation matrix

$$A = \begin{pmatrix} \frac{1}{a} & 0 \\ 0 & 1 \end{pmatrix} \tag{8}$$

is used.

The value of a=3 in the anisotropy matrix is hard-coded in version v4 of the atmosphere algorithm and in all previous versions, but an algorithm parameter in v101-v105 and flowing versions. It has been found that the values of 3 works well in all applications and data sets analyzed to date,

including 2012 and 2013 MABEL-based simulated ATLAS data (M-ATLAS data). Different values of a have been used in 2013 data analyses and parameter sensitivity studies (see Tables 5 and 6). To use an isotropic search, the value must be set to 1, i.e. A becomes the identity matrix.

On units. For ICESat-2 atmosphere data, we need to distinguish between dimensions and factors in meters and in data bins ("shot sums", see section 1.1 "Data Format"). Since each atmosphere data bin represents 280 m along-track and 30m in height (or range), there is already an anisotropy factor of approximately 9 inherent in the data format. Hence

$$a_m = \frac{280}{30} a_{bin} \approx 9 a_{bin} \tag{9}$$

and

$$A_m = \begin{pmatrix} \frac{1}{3} & 0 \\ 0 & 1 \end{pmatrix} \tag{10}$$

corresponds to approximately

$$A_{bin} = \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix} \tag{11}$$

or (squish in the other direction in bin units, which was used in some analyses)

$$A_{bin} = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{3} \end{pmatrix} \tag{12}$$

The relationship between a_m and a_{bin} is explicitly included in the pseudo-code in section (3).

Parameters: Anisotropy factors a_m for data units in meters in meters, a_{bin} for data units in bins (pixels). The anisotropy factor itself does not have a unit.

(M.3) Density Centers

Identification of points within clouds (or other atmospheric features, such as aerosol or blowing snow layers) is motivated by the observation that a cloud is a diffuse reflector, but points within the clouds have a high probability of being located within clusters of other parts of the clouds, a property that does not hold for reflections of ambient light or noise outside of the clouds. To

identify points located inside clusters or clouds of points with higher density, the rbf concept is applied as follows:

For the photon-data analysis problem, the definition set $\mathcal{D}$ is the set of all photons (in a track or window). For each point $c\epsilon\mathcal{D}$, a density value $f_d(c)$ is calculated by summing up rbf values for all neighbors within a given radius r, as follows. The density value is an aggregation of values recorded in a neighborhood of the center point, with close-by points given higher weights and weights decreasing by distance. First, a weight matrix is calculated as

$$W(c,x) = W_c(x) = \Phi(\|x-c\|_a) \tag{13}$$

with $x\epsilon\mathcal{D}_c = \{\tilde{x}\epsilon\mathcal{D} : \|\tilde{x}-c\|_\infty \leq r\}$ the set of all points within a given rectangular box around the center point c (note that in this initial distance determination simply the infinity-norm (absolute distance in each direction) $\|\cdot\|_\infty$ is used). In the radial basis function, we use a norm $\|\cdot\|_a$ that takes anisotropy into account, as described in section (M.2). The specific dependencies of the search area and the norm are given in the implementation section (3). Then the density value, $f_d(c)$, is calculated as

$$f_d(c) = \sum_{x\epsilon\mathcal{D}_c} W_c(x)z(x) \tag{14}$$

where $z(x)$ are the bin data (NRB values, transformed "shot sums"). The normalized density value $f_d^{norm}(c)$ is

$$f_d^{norm}(c) = \frac{\sum_{x\epsilon\mathcal{D}_c} W_c(x)z(x)}{\sum_{x\epsilon\mathcal{D}_c} W_c(x)} \tag{15}$$

where the denominator works as a normalization factor. The advantage of using the normalized density function over the density function is that the mathematical description of the auto-adaptive threshold determination for discrimination of atmospheric layers from background is more transparent.

— The concept of density centers is illustrated in Figures 1-3.

Search neighborhood $\mathcal{D}_c$. Matching the format of the ICESat-2 atmosphere data, which are recorded in rectangular bins (NRB values calculated from 400-shot sums, see section (1.1)), the

search neighborhood used in the density calculation is a rectangular box, consisting of $2r_x + 1$ bins in along-track direction and $2r_y + 1$ bins in height (the "+1" originates from the center of the neighborhood.)

Eqn. (15) is a dot product between the kernel (the weight matrix) and a moving window of NRB values in the data field; both matrices are of size 7by7 (for run1) and 7 by 13 (for run 2), using anisotropy 10,20 and sigma 3,3 (algorithm specific parameter values are for code version v114.0, matching ASAS code release v5.3, March 2020, ATBD, part II, v11).

Derivation of matrix sizes is explained in the remainder of subsection (M.3). Alternatively, the reader is referred to Table 2d and section (3.2.1.2) for reading up on calculation of the kernel dimensions.

Normalization Note that formula 15 suggests that we normalize using the components of the dot product. For edge cases, when the full kernel width or height would hit masked locations (Geomath code) of "invalids" (ASAS code), one has to standardize by summing up the weights that are used, as opposed to all weights. This means, we do not multiply with the full kernel for edge cases. *Note this code change is new for ASAS release v5.3 and geomath code v114.0, Jan 2020.*

The actual value of the radius is a parameter that can be changed in the algorithm. For example, for 2012 MABEL/ M-ATLAS data analysis, the radius parameter is 3 for small neighborhoods and 5 for large neighborhoods. A radius value of 3 ($r_x = r_y = r = 3$) results in a box of size 7 in x (along track) and 7 in z (altitude, range); a radius value of 5 results in a box of size 11 by 11. For more examples, see the parameter table (Table 2d).

The size of the search neighborhood can be selected using the radius values directly, or by prescribing the values of the standard deviation σ in the density function (in eqn. (3,4) in (M.1)) and anisotropy (in equation (7), section (M.2)). If only σ and a_m are given, 2σ is selected to determine the neighborhood, resulting in a kernel matrix with $(4\sigma + 1)$ rows and $(4\sigma a_{bin} + 1)$ columns. If the radius parameter(s) are given in addition, the kernel of the density function is defined by the weights derived using σ and a_m for an area determined by r_x and r_y, and weights are normalized for bins within this area. Typically, integer values are selected for σ. To ascertain that integer neighborhood dimensions result, rounding to the nearest integers is applied (note that code version v105 used $\lceil 2\sigma \rceil$, the ceiling function, which results in the next-largest integer). Relationships between neighborhood size, standard deviation and anisotropy and resulting effects on the success

of the algorithm in layer detection are analyzed and visualized in detail in sensitivity studies in sections (7), (9) and (10).

Parameters: radius used for density determination, radius1 - for density run1, radius2 - for density run2 (see Table 2d). Values for σ and a_m are also given in Table 2d.

(M.4) Density-Dimension and Application as a Noise Filter

As stated in section 2.1, an automated determination of a threshold between clouds and background atmosphere needs to be possible. This task is complicated by the fact that surface reflectivity varies along-track and noise levels change by orders of magnitude between day-time and night-time observations. Hence we need to account for this along-track variability and program an auto-adaptive algorithm to separate noise versus signal and associate signal returns to cloud layers.

In this algorithm, this task is accomplished by the concept of density dimension. The terminology "using density as a dimension" means that density is calculated along-track with an automated determination of a threshold that adapts to noise levels and the level of total returned photons. The threshold is calculated in density space. Implementation of this concept is described in section 3.

(M.5) Discrimination of Optically Thick and Optically Thin Clouds and other Atmospheric Layers

The density-dimension algorithm can be employed to distinguish between optically thick and optically thin clouds, or other atmospheric layers such as aerosols and blowing snow. Spatially narrow layers of clouds with high optical depth have the property that a high density value will be reached in a small neighborhood search. Spatially broad layers of clouds of lower density require a neighborhood search with a larger neighborhood. A simple application of the density-dimension algorithm is to use a fixed neighborhood throughout the entire analysis. This single-density approach allows to detect clouds and aerosols and the ground surface. It works well for 2012 MABEL day-time and night-time data for a large range of flights (see Figs. 1-8, 9 and 11-13).

However, since smaller neighborhoods are sufficient to detect optically dense layers of spatially thin cloud, the following alternative approach can be applied to derive more detailed images of clouds (this involves running density twice).

There are two alternatives to apply the algorithm: Alternative (A): In a simple version, density is

performed once, resulting in the ATL09 product parameter density1. All other steps are performed as described in section 3. Alternative (B): First, the density operator is applied once, using a small neighborhood (e.g., radius 3); then all other processing steps are applied. A binary cloud mask is determined. In preparation of the second step, the area of the dense clouds (area within the cloud mask) is replaced by random points with the same spatial distribution as noise (in the along-track region). Second, the density operator is applied again, using a larger neighborhood (such as radius 5); along with other processing steps. This identifies the thinner clouds (density2) and a binary mask, mask2. The binary masks from density1 and density2 are combined to create the cloud mask (union of the two mask areas).

The number of density runs becomes a control parameter in the algorithm (see section (3)).

(M.6) Removal of Small Clusters and Determination of Final Cloud Mask

The goal of the algorithm is to derive atmospheric layers and their top and bottom boundaries. In the analysis of ICESat-2 ATLAS atmosphere data, there will be a search for a maximum of six layers (up to 15km above the Earth Surface, here approximated by the on-board DEM). The cloud mask determined in step (M.5) may leave small areas that appear like speckles and likely only the large, simply connected areas are clouds. An algorithm that removes any small clusters is applied. The numerical algorithm used for small-cluster removal is described in the pseudo-code section (3). The resultant cloud mask defines the location of atmospheric layers.

Parameter. In the current implementation, any clusters that are simply connected areas of less than 300 bins are removed (see Table 2d).

(M.7) Determination of Atmospheric Layers and Their Top and Bottom Boundaries

A mask is given as a binary data set with a 1 for "cloud bin" and a 0 for "not-cloud bin". Layers must be at least 90 m thick. The height of the top-most 1 in a series of at least three consecutive 1-s is identified as the top of a cloud/atmospheric layer, the height of the bottom-most 1 in a series of at least three consecutive 1-s is identified as the bottom of a cloud/atmospheric layer. Exact implementation of the code is given in the pseudo-code section (3), as gaps in cloud layers must also have a minimal thickness.

(M.8) Density of a Column and Density of an Atmospheric Layer

Density of a column is calculated by summing up all density values in a vertical profile (=column). Density of a layer is calculated by summing up all density values within the given layer in a vertical profile.

3 Algorithm Steps and Pseudocode: Density-Dimension Algorithm for ATLAS Atmosphere Data

This section includes pseudocode with explanations of algorithm steps, along with essential sections of the original code. The original code is written in python (the python code is given in blue boxes). The pseudocode is a translation from python into a generic algorithmic language with comments (the pseudocode with comments is given in yellow boxes, pseudocode in black font and comments in green font).

In text sections that accompany the code listings, the annotated pseudocode is linked to the equations in the mathematics sections.

Figures are included to illustrate the computational steps as well as to indicate at which point in the code the figures can be created (i.e. at which point in the code the result or product has been created that is needed to create a given figure). The actual code for figure creation is not included.

3.0 Mask Handlers

The geomath code uses mask handlers to manage the location of valid data for the 15km atmospheric data range (467 bins within the larger frame of 700 profiles), to identify missing value location, to pass cloud regions identified in density pass 1 to the next density pass 2, and to combine results from the two density runs into a final mask, from which cloud boundaries are determined. Mask handling is especially critical when using the algorithm option of "Running Density Twice", described in section (3.10). Therefore an explicit description of the flow of mask handling is included in section (3.10).

3.1 Step 1: Set Parameters and Load Data

This part of the algorithm sets the parameters for the DDA and loads the data. At the end of this step, we have valid_data_mask. Density calculation in run 1 uses this mask.

3.1.1 Set Parameters

Set the algorithm parameters given in the command line call of the python code in Listing 1.

```
cmdline_parser.add_option(
    '-n', '--name',
    dest='run_name',
    metavar='DIR',
    default=time.strftime(start_time).replace(' ','_'),
    help='Run/experiment identifier and name of output directory')

cmdline_parser.add_option(
    '-v', '--verbose',
    dest='loglevel',
    default=logging.INFO,
    action='store_const',
    const=logging.DEBUG,
    help='Changes logging level from WARNING to DEBUG')

cmdline_parser.add_option(
    '-e', '--end',
    dest='end',
    type='int',
    default=None,
    metavar='END',
    help='Stops algorithm after END profiles')

cmdline_parser.add_option(
    '-a', '--anisotropy',
    dest='anisotropy',
    type='str',
    default='10',
    help='Anisotropy factor for kernel as horizontal/vertical. Multiple values will do
```

```
    hierarchy of densities.')

cmdline_parser.add_option(
    '-d', '--downsample',
    dest='downsample',
    type='str',
    default='5',
    help='Downsample factor. Multiple values will do heirarchy of densities.')

cmdline_parser.add_option(
    '-s', '--sigma',
    dest='sigma',
    type='str',
    default='100',
    help='Standard deviation of gaussian kernel (# of pixels vertically). Multiple values
    will do heirarchy of densities.')

cmdline_parser.add_option(
    '-c', '--cutoff',
    dest='cutoff',
    type='str',
    default='2',
    help='Cutoff gaussian kernel after number of stddevs. Multiple values will do heirarchy
    of densities.')

cmdline_parser.add_option(
    '-t', '--threshold_factor',
    dest='threshold_factor',
    type='str',
    default='5',
    help='Adaptive factor for threshold local quantile. Multiple values will do heirarchy
    of densities.')

cmdline_parser.add_option(
    '-T', '--threshold-bias',
    dest='threshold_bias',
    type='str',
    default='50',
    help='Base threshold for threshold. Multiple values will do heirarchy of densities.')
```

```
cmdline_parser.add_option(
    '-m', '--min-cluster',
    dest='min_cluster',
    type='str',
    default=300,
    help='Minimum cloud cluster size in number of pixels')

cmdline_parser.add_option(
    '-L', '--threshold-window',
    dest='threshold_window',
    type='str',
    default='0',
    help='Enter n for threshold segment length = 2n+1. Multiple values will do heirarchy of
   densities.')

cmdline_parser.add_option(
    '-q', '--quantile',
    dest='quantile',
    type='str',
    default='0.5',
    help='Quantile 0 < q < 1. Multiple values will do heirarchy of densities. Used for Day
  and Twilight data.')

cmdline_parser.add_option(
    '-Q', '--quantile_night',
    dest='quantile_night',
    type='str',
    default='0.5',
    help='Quantile values used for Night time data.')

cmdline_parser.add_option(
    '-D', '--dem',
    dest='dem_filepath',
    type='str',
    default='0',
    help='Path to DEM text file in case of NRB text file input (never use with h5 files)')
```

Listing 1: Python Code v114.0 (2020-02-06): Set Parameters.

3.1.2 Load Data

The following "Load Data" module is specific to loading data from the ATL04 data products. It also includes options for loading ANC39 data and text files.

```
def load_nrb_data(nrb_filepath,dem_filepath):
  """
  The load_nrb_data function is the data loading function used to load the raw
    data before initialization of an nrb_image object.
  """
  dem_flag = 0 # only get dem data (for surface finding) if available
  print 'The filepath is: ', nrb_filepath
  logger = logging.getLogger('atmos_DDA')

  if nrb_filepath.find('ANC39') != -1:
    fid = h5py.File(nrb_filepath, 'r')
    shot_sums = np.array(fid['/profile_2/atm_bins'])
    elev_top = np.array(fid['/profile_2/atm_tw_top'])
    nn,mm = shot_sums.shape
    nrb_data = np.ones((nn,700))*3.4028235e38
    # Convert to vertically aligned nrb-type matrix with 700 bins instead of 467
    for i in xrange(nn):
      top_bin_temp = int((20000.0 - elev_top[i])/30.0) + 1
      for j in xrange(mm):
        if (top_bin_temp+j) < 700:
          nrb_data[i,top_bin_temp+j] = shot_sums[i,j]
    nrb_data = np.flipud(np.transpose(nrb_data))
  elif nrb_filepath.find('.h5') != -1:
    fid = h5py.File(nrb_filepath, 'r')
    nrb_data = np.array(fid['/profile_2/nrb_profile'])
    nrb_data = np.flipud(np.transpose(nrb_data))
    dem_height = np.array(fid['/profile_2/dem_h'])
    dem_flag = 1
    solar_elev = np.array(fid['/profile_2/solar_elevation']) # added in solar elevation
    data to tell time of day. Adam 9-23-19
    solar_elev = np.flipud(np.transpose(solar_elev))
    windspeed = np.array(fid['/profile_2/windspeed']) # added in windspeed. Adam
    10-22-19
    ds_va_bin_h = np.array(fid['/profile_2/ds_va_bin_h'])
```

```
    ds_va_bin_h = np.flipud(ds_va_bin_h) # added in bin heights. Adam 12-3-19
  elif nrb_filepath.find('.txt') != -1:
    #nrb_data = np.flipud(np.loadtxt(nrb_filepath))
    nrb_data = (np.loadtxt(nrb_filepath, delimiter=' ')) #I commented this out to use my
    own txt files. Signed Alec 10-11-17
    if dem_filepath != '0':
      dem_height = (np.loadtxt(dem_filepath, delimiter=' '))
    else:
      dem_height = np.array([])

  elif nrb_filepath.find('.hdf') != -1:
    fid = h5py.File(nrb_filepath, 'r')
    nrb_data = np.array(fid['/Total_Attenuated_Backscatter_532'])
    nrb_data = np.flipud(np.transpose(nrb_data))
  else:
    logger.error('Unable to load {} in either text or HDF5 format. Is loading of this
    filetype supported in load_nrb_data()?'.format(nrb_filepath))
    raise(ValueError)

  return(nrb_data, dem_height, solar_elev, windspeed, ds_va_bin_h)
```

Listing 2: Python Code v114.0 (2020-02-06): Load NRB Data.

This part of the algorithm simply loads the hdf5 files, which are described in section *(1) Data*. The data are stored as a 2-dimensional array containing the NRB-corrected values per bin. The size of this file is 700 bins in height with 467 valid data bins and near-fixed length in along-track distance. Each bin represents 280 meters along-track (derived using satellite velocity in orbit) and 30 meters in elevation (or range), the total altitude range is -1000m to 14000m embedded in 20000m. The data set is output as a figure (data.png, Figure 1). Additional figures are given in section *4 Application to 2012 MABEL Data*.

[Change 2014-07-25:] The ATLAS data are be in a frame of 700 profiles in elevation, to account for the fact that the atmosphere data is recorded relative to the onboard DEM, for an interval of 250m below the DEM to 14750m above the DEM. The data is then included in a data frame of a fixed references of [-1000m, 20000m] to ascertain that neighboring bins represent the same height. Note there are now 700 bins of 30m height for a total height of 21000m, with measured values for an interval of 15000m and flag values for padding at top/ bottom of each profile. The flag value is

-9999. (see Change 2016-08-23: height bin 29.9m).

Note that examples compiled before 20140725 use the previous data format of 500 elevation bins.

In case of actual ICESat-2 data processing, the input data are be NRB_Prof (as provided in ATL04, see Table 2.7, Part I), instead of photon sums, given as a Float(700,3), for 3 strong beams from 20 to -1km, based on the local DEM value, with vertical resolution of 30 m (exactly: 29.9m). The integer(3) NRB_Top_Bin, also in ATL04, gives the starting (top) bin number within the -1 to 20km frame where data begins, for each of the three strong beams. There are always a total of 467 valid bins (unless there are missing data). For both inputs, see Table 3 "ATL04 Product Parameters" in section 2.3.5, Part I.

The NRB profiles are created from the profiles of raw photon counts - supplied from ATL02 by subtracting the background, multiplying by the square of the range from the satellite to the return height and normalizing by the laser energy (see ATBD Part I).

Note on Corrections. The data analysis can also be based directly on photon counts; these are recorded in ATL02 (see Part I) under the file name /atlas/pcex/atmosphere_sw/atm_bins. The "/atlas/pcex/atmosphere_sw/" part of that name is the HDF group. Alternatively, photon counts ("400 shot sums") can be reconstructed from the inverse operations of noise removal and range-correction. This step is relevant at the current stage of code development (2014-10-07), as it affects the processing of simulated ATLAS data based on airborne MABEL data and their range and noise characteristics. This is further discussed in a section on corrections and simulations, section (6.1).

[Change 2016-08-23:] The size of the height bin was determined to be 29.9m rather than 30m.

Note on MABEL and simulated ATLAS data:

2012 Examples are based on MABEL data and a simple simulation algorithm (one can simply use the MABEL data).

2013 examples use NRB and range-square correction and a more complex simulation. The correction analysis and effects on noise levels is given in a separate section (6.1).

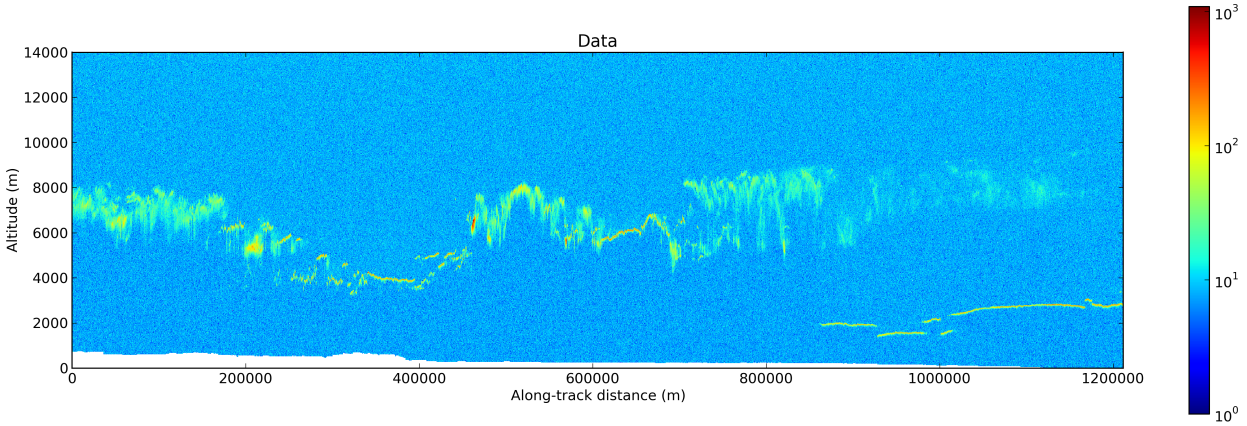


Figure 1. Data. MABEL data set 02Apr12.02. [Based on ATLAS data simulation without range-square correction and without NRB correction.]

3.2 Step 2: Calculate Density

Summary: Density (Density1) is calculated following equation (5) using a neighborhood of a radius of 5, this results in a box of size 11 by 11 as a moving window. In the radial basis function, a mean of zero and a standard deviation of 20 is used. The radial basis function depends on distance as independent variable. The distance is modified using a linear map to emphasize bins along the along track axis. Density is calculated for every point in the data set. *Note that other values are used for neighborhood in later studies, see Table 2d.*

3.2.1 Computation of Density — Overview of Steps

Note: This overview is written to facilitate faster code implementation, however, the detailed instructions in the main description section (3.2.2) are still valid. The equation numbers repeat.

Note: now $y_{res} = 29.9$, it used to be $y_{res} = 30$, so we need to explicitly use y_{res} in the code and the algorithm description.

(3.2.1.1) Step 2.1: Read in Kernel Control Parameters

The central part of the density calculation is the calculation of the Gaussian kernel for the radial basis function, which performs the data aggregation. The calculation of the kernel is controlled by the parameters a_m, σ and *cutoff*, which are read in, as described in section (3.1) "Step 2: Read in algorithm-specific parameters and load data". From these three parameters, the dimensions of the kernel are derived (see Step 2.2).

a_m — anisotropy factor for data in meters (e.g.: $a_m = 10$)

σ_{bin} — $\sigma = \sigma_{bin}$ the standard-deviation of the rbf for data in pixels (= for data in bins), for the vertical direction (for y), (e.g. $\sigma = \sigma_{bin} = 3$)

cutoff — the number of standard deviations after which the kernel is " cut off".

Alternatively, the kernel can be controlled by the dimensions n, m and a subset of the three parameters, either $\{a_m, \sigma\}$ or $\{a_m, \mathit{cutoff}\}$. Defining n, m first makes it easier to design a kernel from the viewpoint of a moving window averaging or convolution, while defining the statistical parameters σ and *cutoff* first allows to design the kernel from a Gaussian point of view. The anisotropy factor a_m always needs to be defined. The algorithm allows for these different options to control the kernel.

(3.2.1.2) Step 2.2: Calculate Dimensions of the Kernel

[see page 44 of ATBD, v 6.0]

$$n = \text{int}(2 \text{ round}(\sigma \text{ cutoff})) + 1 \tag{16}$$

$$\begin{aligned} m &= \text{int}(2 \text{ round}(\sigma \text{ cutoff } \frac{y_{res}}{x_{res}} \ a_m)) + 1 \\ &= \text{int}(2 \text{ round}(\sigma \text{ cutoff } a_{bin})) + 1 \end{aligned} \tag{17}$$

using

$$a_{bin} = \frac{y_{res}}{x_{res}} \ a_m \approx \frac{30}{280} \ a_m \approx \frac{1}{9} \ a_m \tag{18}$$

Note that the anisotropy is unit-less (dimensionless), but it takes a different value, when calculated for data in meters or calculated for data in pixels.

[compare p. 14-15 in ATBD, v 6.0]

Note: If one uses $\sigma' = 30\sigma$ (as in code version v105, the version used in ATBD v6.0) then the equations for n and m in the algorithm or code can assume a more symmetrical form:

$$n = \text{int}(2 \text{ round}(\frac{\sigma'}{y_{res}} \text{ cutoff})) + 1 \tag{19}$$

$$m = \text{int}(2 \text{ round}(\frac{\sigma'}{x_{res}} \text{ cutoff } a_m)) + 1 \tag{20}$$

Note that $\sigma' = 30\sigma$ converts from {standard-dev for units in pixels in the y-direction} to {standard-dev for units in meters in the y-direction}.

However, v106 now uses the most intuitive form of these equations, which are eqn 1 and the first line of eqn (2)!!!!! This avoids having to worry about the standard-deviation and its relationships to units all together.

(3.2.1.3) Step 2.3: The Norm: Determination of the Distance that is Needed in the *RBF*

The next step is to calculate the distance between a center point and an neighboring point, as needed in the *rbf*. This is calculated for each pixel (i, j) of the kernel, as follows:

$$dist(i,j) = \text{norm}\Big(\frac{1}{a_m}\ (j - \frac{m-1}{2})\ x_{res},\ (i - \frac{n-1}{2})\ y_{res}\Big) \quad \text{for} \quad i = 1, \ldots, n; j = 1, \ldots, m \tag{21}$$

where norm is the library function for the Euclid norm (2-norm). The distance is now in meters.

(3.2.1.4) Step 2.4: Kernel Calculation

$$\text{kernel(i,j)} \ = \ \text{Gaussian}\big(\text{loc} = 0, \text{scale} = \sigma\ y_{res}).pdf(dist(i,j))\big) \tag{22}$$

i.e. here one has to use the standard deviation σ in meters in the y-direction (multiplied by y_{res}). The formalism is that of applying a 1-dimensional Gaussian function with center in 0 and standard deviation $\sigma' = \sigma\ y_{res}$ to the anisotropy-norm-based distance (between a center and a point).

Note: If calling a library function for the Gaussian function, one needs to check whether that function uses the standard deviation or the variance!

(3.2.1.5) Step 2.5: Normalization of the Kernel Values

- Sum up the kernel values and divide each (i,j) entry by the sum, according to eqn. 15 (in ATBD v6.0).

Update for ASASS release v5.3, geomath code v114, 2020-Jan

According to eqn 22, we standardize the kernel values before the dot-product multiplication, or convolution. In the edge case of hitting invalids, we have to adjust by summing up only those kernel weights that are actually multiplied by valids. *Code changed to:* Standardize, after the masks/invalids have been passed through the density calculation (ASAS and geomath).

- in python or pseudocode:

$$kernel(i,j) \ = \ \frac{kernel(i,j)}{kernel.sum} \tag{23}$$

3.2.2 Description

Search neighborhood $\mathcal{D}_c$. Dependencies of search neighborhood determination are described in section (M.3).

Treatment of edges:

- Version v4 and version v6 code: To avoid edge effects, density is not calculated for points which are located within the neighborhood search radius of the edge of the data set. Bins on the boundary are assigned a density value of zero.

- Version v103.0 code: To avoid edge effects, density for points within the neighborhood search radius is calculated by "folding the kernel over", i.e. reflecting the data from the inside of the data window to the outside, to cover the area needed by the kernel; i.e. points from the inside are mirrored to the outside. This is a built-in capability of the convolution algorithm used.

- Version v105 and v106: The edge-foldover for the kernel is not implemented (because larger sections of data were used). If large data gaps occur during the operational phase of the mission, an edge-foldover of the kernel may be nice to have.

- Version v114 and ASAS code release v5.3: Changed to not using the *convolve* function any more in geomath python code.

  Background: Geomath code used to implement the density calculation by using the convolve function in python (see, eqn.21 and 22), whereas ASAS implements the calculation of the dot product directly in F90. This resulted in differences in mask handling (or so-called "invalids" and has been changed. Geomath code v114 calculates the dot product directly. See new python code snippet.

Detail. In order to aggregate points for the density calculation, a distance function is needed. This involves an anisotropy matrix (see equation (7)), called squish-matrix in the pseudocode. Each point in the data set will be a density center, and the distance function (with anisotropy norm) will be applied to all points as given eqn. (8).

The radial basis function (RBF) is then applied, it is a real-valued function whose value depends on the distance from a center, as described in sections (M.1)-(M.3). The function *normpdf* is the Gaussian probability density function given in equation (3). A generic *normpdf* function may be available in a software library, this may typically be applied using mean=0 (because in our application the distance to the center point is already calculated). Alternatively, the result can be calculated for the distance value, r, as in eqn. (4).

Application to all points in the data set is performed.

Parameters: Notice that a neighborhood of radius 5 is used here in 2012 data analysis, this can be changed, and different values are used for 2013 data analysis and in the sensitivity study. However, as the triple-data-set example demonstrates, the radius value does not need to be changed for each data set (i.e. this does not affect the auto-adaptive capabilities of the algorithm; instead, the algorithm developers found in sensitivity experiments that the previously selected fixed parameters could be improved upon.) – See sections on sensitivity studies (7), (9) and (10).

Note that in the following calculations, the term *weight matrix* is used for a matrix of radial basis function values; and *weight function* is used in place of *RBF function*.

Parameters: Anisotropy factor. Recalling that the anisotropy factor in meters, a_m is related to the anisotropy factor a_{bin} in bin units by

$$a_m = \frac{280}{30} a_{bin} \approx 9 a_{bin} \tag{16}$$

the default values are $a_m = 3$ and $a_{bin} = 1/3$ in 2012 MABEL data analysis (see figures (1-9 and 11-12); in 2013 M-ATLAS analysis we use $a_{m1} = 10$ and $a_{m2} = 20$ for two density runs (see Table 2d). The effect of changing the anisotropy parameter is illustrated in the sensitivity studies in section (7).

To examine and illustrate the effects of changing the fixed parameters on the density calculation and resultant cloud layer determination, a set of sensitivity studies is carried out in section (7). This is included to allow later adaptation of the algorithm, in case the actual noise characteristics of the ATLAS data after launch change. The algorithm is now written such that the parameters can be easily changed, but, once determined, are automatically passed through the code modules.

Normalization option. To get a more automated handle on the parameters, the analysis of the 2013 M-ATLAS data uses the density function with a normalization factor (eqn. (15)), such that the weights in the kernel will sum up to 1. To recall, the kernel is the Gaussian kernel intersected with

the search window. The 2012 MABEL data analysis is carried out without normalization by the sum of weights (eqn. (14)), but with normalization by maximum. The code uses the default setting "normalization = false", referring to normalization by sum of weights. The normalization option is actualized by passing a flag to the respective module (see listing). The 2013 M-ATLAS data analysis uses "normalization = true" (eqn. (15)), as does the analysis of GLAS-based simulated data.

Update ATBD. v11, Jan 2020; ASAS release v5.3 and geomath code v114.0, Jan 2020. As noted in (M.3), normalization for edge cases was changed. For edge cases, when the full kernel width or height would hit masked locations (Geomath code) of "invalids" (ASAS code), one has to standardize by summing up the weights that are used, as opposed to all weights. This means, we do not multiply with the full kernel for edge cases.

Next, the density values are calculated by summing up as in eqn. (5), this is given in the listing "Call Compute-Density".

Code Listings

There are two listings here. The first is the function call from the main code that includes updates to the logfile, plotting commands, and general housekeeping (listing "Call Compute-Density"). The second is the definition of the Compute-density function (Listing "Compute-density function").

```
115 def compute_density(image):
    """
    The compute_density function computes the density array data.density from raw
      nrb data in data.raw_nrb_data by calculating a kernel and then convolving the
      raw data with the kernel, data.kernel.

    Note: scipy.ndimage.filters.convolve is NOT mask aware. Use appropriate fill
      value.
120
    Note: using astropy.convolution would allow for nan values which get
      interpolated.
    """
    x_res, y_res = image.pixel_dimensions

125 # Extract parameters from image.parameters
    # We multiply sigma by y_res because sigma is originally given in units of
      vertical pixels and we need it in meters now
    sigma = image.parameters['sigma']*y_res
    cutoff = image.parameters['cutoff']
    anisotropy = image.parameters['anisotropy']
130
    nk = int(2*round(sigma/y_res*cutoff))+1
    mk = int(2*round(sigma/x_res*cutoff*anisotropy))+1

    kernel = np.zeros((nk, mk))
135 for i in xrange(nk): # rows is y-axis
      for j in xrange(mk): # cols is x-axis
        x = norm((1/anisotropy*(j-(mk-1)/2.)*x_res, (i-(nk-1)/2.)*y_res))
        kernel[i,j] = gaussian(loc=0, scale=sigma).pdf(x)

140 # Normalize the kernel
    #kernel = kernel/kernel.sum()
    image.kernel = kernel / kernel.sum()
```

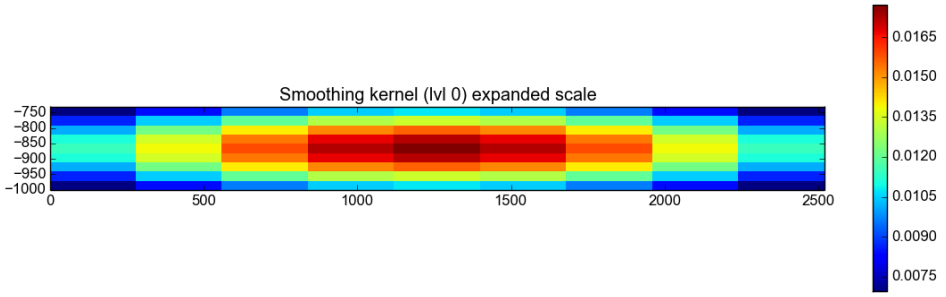
```
# i = bins
# j = profiles

# updated density calculation to account for edge cases, changed from using
  convolve function to processing step by step. Adam 10-19

(bins,profs) = image.adjusted_nrb_data.shape
rx = nk//2
ry = mk//2
nrb_data_masked = np.ma.array(data=image.adjusted_nrb_data, mask=image.valid_mask).filled(
  np.nan)
density = np.zeros((bins,profs))

for i in range(bins):
  for j in range(profs):
    dot_density = 0
    kernel_sum = 0
    if(np.isnan(nrb_data_masked[i,j])): continue
    ii = 0
    for s in range(i-rx,i+rx+1,1):
      jj = 0
      for t in range(j-ry,j+ry+1,1):
        if(s < 0 or t < 0 or s >= bins or t >= profs): continue

        if(np.isfinite(nrb_data_masked[s,t])):
          kernel_sum += kernel[ii,jj]
          dot_density += (kernel[ii,jj]*image.adjusted_nrb_data[s,t])

        jj+=1

      ii+=1

    if(dot_density == 0 or kernel_sum == 0):
      density[i,j] = 0
    else:
      density[i,j] = (dot_density / kernel_sum)

#image_density = np.where(np.isnan(density),0.0,density)
```

```
180   image.density = density
```

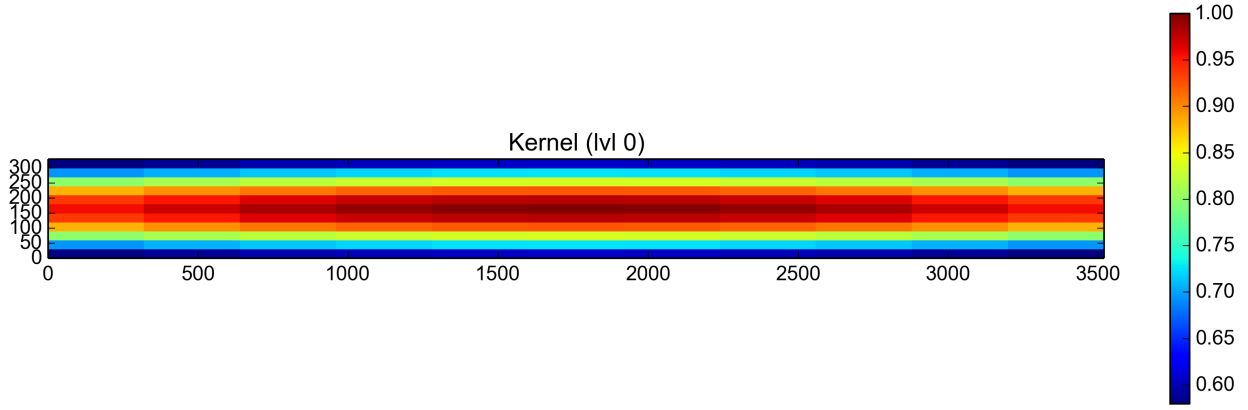
Listing 3: Python Code v114.0 (2020-02-06): Call “compute-density” function.

Output

A figure of the weight matrix $W_c(x)$ is created (see eqn. 13), given in Figure 2.



(a)



(b)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
1																																			
2															1,1	1,2	1,3	1,4	1,5																
3															2,1	2,2	2,3	2,4	2,5																
4															3,1	3,2	3,3	3,4	3,5																
5															4,1	4,2	4,3	4,4	4,5																
6															5,1	5,2	5,3	5,4	5,5																
7																																			
8																																			
9																																			

(c)

Figure 2. Illustration of density calculation as convolution with kernel (weight matrix). (a) Typical weight matrix (kernel matrix) used in the Density-Dimension Algorithm. This example is taken from sensitivity study (t8) for GLAS-based simulated ICESat-2 data [see Table 4 and section 8, Figure 31-1]. (b) Weight matrix (kernel matrix) used for analysis of MABEL data set 02Apr12.02. This kernel uses r=5 and results in an (11,11) weight matrix. $\sigma = 140$ makes the kernel appear unusual. (c) Convolution of data matrix with kernel is implemented as a moving-window operation: point-wise multiplication of the weight matrix (kernel matrix) with a window of the data set. In the illustration, a kernel of (5,5) is applied to a data set of 35 profiles and 9 height boxes per profile. Center point of the example window is (17,4). Figure in (b) thanks to Mark Vaughan.

At this point, the density matrix can be written out and a plot of density created, as shown in the figure below (Fig. 4.3).

[Output: density plot, {date}dens5.out]

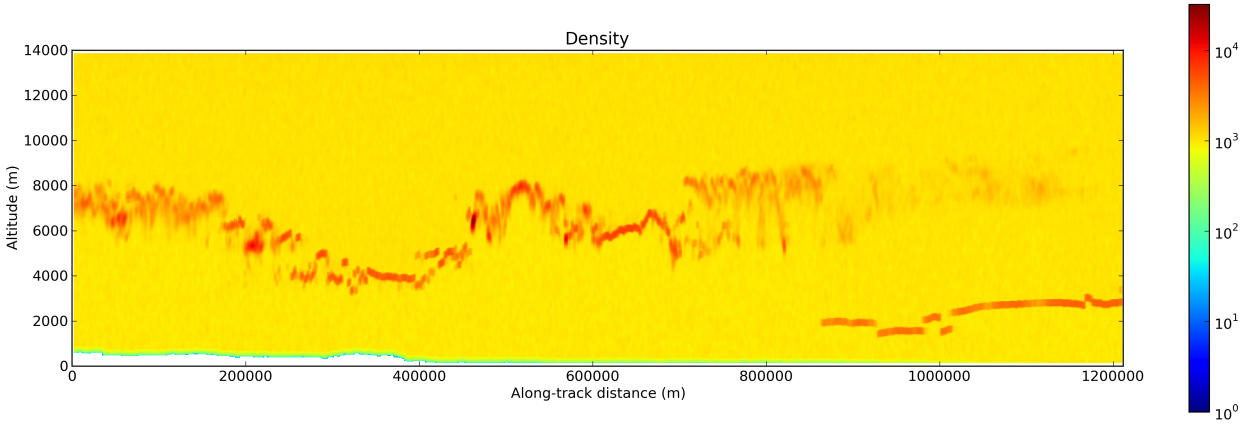


Figure 3. Density. MABEL data set 02Apr12.02

3.3 Step 3: Using Density as a Dimension: A Density-Based Automatically-Adapting Noise Filter

The objective of this step is to separate reflectors (clouds, aerosol layers, blowing snow) from background regions. This is achieved by application of the density-dimension algorithm, which yields a density-based noise filter that moves along-track and automatically adapts to the variable conditions of reflectivity and noise levels. Motivation and background of the ideas programmed here are described in section (M.4).

Throughout the time of algorithm development, simulation of different data sets and collection of data from airborne simulator instruments of the ICESat-2 ATLAS instrument, several different versions of the density-dimension algorithm have been implemented, especially of the auto-adapative threshold determination function. In the following sections, algorithm components are identified as "Method A" and "Method B". Method A was originally developed for analysis of 2012 MABEL data (sections 4 and 5), and Method B was originally developed for analysis of 2013 M-ATLAS (sections 6 and 7), however, both methods work for any atmospheric lidar data set (with similar properties and formats). New for code versions v104.0 (September 2015) and v105.0 (October 2015), an algorithm that synthesizes methods A and B was developed and implemented. The integrated approach is described in section 3.3.6 (ATBD part II, v6.0 and v7.0) and should be the only approach that needs to be implemented by SIPS, because it is upward compatible with all previous algorithms. Previous descriptions are kept in this document version for redundancy and to allow recreation of analyses based on earlier experiments and data sets. The integrated method is applied to analysis of GLAS-data-based simulated ICESat-2 data sets which were created in 2015 by S. Palm and K. Barbieri (sections 8 and 9).

The algorithm components are implemented as described in the following sections and in Tables 2a-d.

3.3.1 Creating Downsampled Density Arrays (Method A)

For an area of interest (this can be a long flight segment), an array of density values is created (in the previous step). The density array is downsampled by a factor of 5 in each direction, the downsampled array contains the maximal value in bins of 5 by 5 (submatrices of size 5 by 5) of the original density array. The 5-by-5 sized submatrices contain small regions, the density characteristics of each region are stored in the downsampled array. For profiles of 500 bins per profile (per vertical column), the downsampled array has 100 points per vertical column, generally the down sampled array has (number-of-bins)/5 points per vertical column. In width each vertical column of the original data array (and hence of the density array) corresponds to 280 meters along-track, hence each column of the downsampled array corresponds to 1,400 meters alongtrack. The down sampled array is simply a reduction in resolution by a factor of five in each direction.

Pseudo-code for this step is given in the listing "Creating downsampled density".

```
define downsample(density_matrix, neighborhood, bin):
    # performs maximum-downsampling
    # arguments:
    # bin - amount of downsampling (default 5)
    # neighborhood - neighborhood used in density calculation

    # shape returns the size of the matrix data
    [number_of_rows, number_of_columns] = shape(density_matrix)

    # take away zero padding
    density_matrix_no_pads = density_matrix[neighborhood:-neighborhood, neighborhood:-
    neighborhood]

    # create matrix of zeros size number_of_rows/bin,number_of_columns/bin
    downsampled_density = zeros(number_of_rows/bin, number_of_columns/bin)

    # i iterates 0 to number_of_rows-1 by steps of bin
    for i in 0 to number_of_rows-1 by bin:
        #j iterates 0 to number_of_columns-1 by steps of bin
        for j in 0 to number_of_columns-1 by bin:

            #submatrix is a matrix size bin by bin
```

```
            submatrix = density_matrix[i:i+bin, j:j+bin]

            # assign the max value of the submatrix to the downsampled_density
    matrix
            downsampled_density[i/bin, j/bin] = max(submatrix)



    #find the max value within the downsampled_density matrix
    max_filler = max(downsampled_density)

    #replace all 0 in downsampled_density with the max_filler
    replace(0, max_filler, downsampled_density)

    return downsampled_density
```

Listing 4: Pseudo-Code v103 (2014-10-30): Creating Downsampled Density (Method A)

```
define downsample(dens, neighborhood, bin):
    bi = 0
    while sum(dens[bi]) == 0: bi+=1
    bj = -1
    while sum(dens[bj]) == 0: bj-=1
    k = 0
    while sum(dens[:,k]) == 0: k+=1
    l = -1
    while sum(dens[:,l]) == 0: l-=1

    dens_n0 = dens[bi:bj-1,k:l-1]

    x,y = dens_n0.shape

    ny = y/bin
    nx = x/bin

    chy = y % bin
    y-=chy

    chx = x % bin
    x-=chx
```

```
    downsampled = dens_n0[:x,:y].reshape([nx, x/nx, ny, y/ny]).max(3).max(1)

    refi = downsampled.max()

    downsampled = array([[refi if v == 0 else v for v in downsampled[i]] for i in xrange(len(
    downsampled))])

    return downsampled
```

Listing 5: Python Code v103 (2014-10-30): Creating Downsampled Density (Method A)

3.3.2 Threshold Determination (Method A)

In the next step, the downwsampled array that contains the maximal density values of the small regions is examined to create noise regions and to determine a signal-noise threshold. The algorithm moves along track, using segment lengths of one vertical column in the down sampled matrix, corresponding to 5 columns in the original density matrix. For each segment, a number (50 in this code version) of 5-by-5 bins with lowest maximal values is identified. The maximal value of the density values in this set of 50 is used as the threshold between noise and cloud-signal for this along-track segment. This threshold value is applied to identify clouds in this along-track segment.

The pseudo-code for this step is given in listing "Finding thresholds".

Note that this step is different for method B.

Parameters:

- number of bins that form the noise area set (50 bins of size 5 by 5), 50 lowest values used for threshold determination for noise regions. Note total is 100 bins in the downsampled array.

- along-track size of the segment for which a noise-threshold is determined (1 column of the down sampled matrix, equal to 5 columns of the original matrix, equal to 1400 meters)

```
define determine_thresholds(downsampled_density, bin_count, neighborhood):
    # computes threshold vector
    # arguments:
    # downsampled_density – downsampled density matrix
    # bin_count – number of bins to ignore (default 50)
    # neighborhood – neighborhood used in density calculation

    # shape returns the size of the matrix data
    [number_of_rows, number_of_columns] = shape(downsampled_density)

    # for each column index in downsampled_density
    for j in 0 to number_of_columns-1:
        #for each integer from 0 to one less then bin_count
        for i in 0 to bin_count-1:
            #find the first occurance of the minimum
            #and replace it with the max_filler
            first_replace(min(downsampled_density[:,j]), max_filler)
```

```
    # find the minimum of each column
    # downsampled_thresholds will be a vector size 1 by number_of_columns/bin
    downsampled_thresholds = min(downsampled_density, 1)

    # start is a vector of size 1 by neighborhood filled will the first
    downsampled_threshold value
    start = repeat(downsampled_thresholds[0], neighborhood)

    # middle is a vector that has every element of downsampled_thresholds repeated
     bin times
    middle = repeat(downsampled_thresholds, bin)

    # start is a vector of size 1 by neighborhood filled will the last
    downsampled_threshold value
    end = repeat(downsampled_thresholds[number_of_columns-1], neighborhood)

    # combine the start, middle, end to have a vector size 1 by number_of_columns
    thresholds = cat(start, middle, end)

    return thresholds
```

Listing 6: Pseudo-Code v103 (2014-10-30):: Determination of Thresholds (Method A)

```
def determine_thresholds(downsampled, bin, bin_count, neighborhood):
    refi = downsampled.max()
    min_indexes = []
    min_values = []
    for i in range(bin_count):
        min_indexes.append(argmin(downsampled,0))
        min_values.append(downsampled.min(0))
        for i in range(len(min_indexes[-1])):
            downsampled[min_indexes[-1][i],i] = refi

    min_values = array(min_values)
    max_values = min_values.max(0)
    max_values = array([max_values[0]]*bi+[item for sublist in [[x]*bin for x in max_values]
    for item in sublist]+[max_values[-1]]*(chx+1)+[max_values[-1]]*(abs(bj)+1))
```

```
    thresholds = max_values


    return thresholds
```

Listing 7: Python Code v103 (2014-10-30):: Determination of Thresholds (Method A)

The auto-adaptive threshold is illustrated in Figure 4.

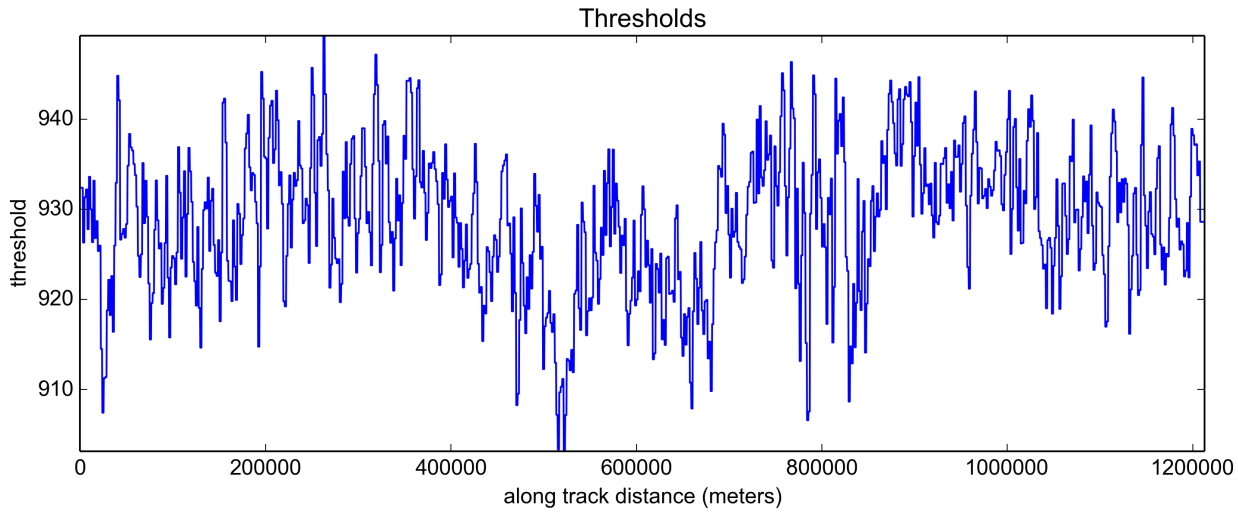


Figure 4. Auto-adaptive threshold levels, MABEL data set 02Apr12.02.

3.3.3 Application of a Binary Matrix to Outline Cloud Areas (=First Approximation): Cloud Boundary/ Cloud Area Determination (Method A)

A binary matrix is a matrix filled with zeros and ones. Ones identify cloud areas, zeros identify background areas/ noise areas. Using the threshold determined for each along-track location (i.e. for each segment of 5 columns), a value of 1 or zero is assigned for each location in the original matrix according to the following rule (where x(i,j) is the density value in the original matrix):

if x(i,j) in the original matrix is larger than threshold in this segment + threshold-bias, then binary(i,j)=1

else bin(i,j)=0

Parameter: a threshold-bias of 70 is used.

((
for j in 0 to number_of_columns:
for i in 0 to number_of_rows:
if density_matrix[i,j]>threshold_bias + threshold[j]:
then : Binary_matrix[i,j] = 1
else : Binary_matrix[i,j] = 0
end - i-loop
end - j-loop
))

```
define compute_binary_mask(density_matrix, threshold_bias, thresholds):
    # computes binary mask of cloud regions
    # arguments:
    # density_matrix - matrix of computed densities
    # threshold_bias - overall threshold bias
    # thresholds - vector of thresholds

    # shape returns the size of the matrix data
    [number_of_rows, number_of_columns] = shape(density_matrix)

    # binary_matrix is a matrix size number_of_rows,number_of_columns filled with
    zeros
    binary_matrix = zeros([number_of_rows, number_of_columns])
```

```
    # j iterates over integers from 0 to number_of_columns-1
    for j in 0 to number_of_columns-1:
        # temp_vector is a vector size number_of_rows by 1
        # elements in temp_vector are 1 where the density_matrix value is greater
    than the threshold for the jth column
        for i in 0 to number_of_rows-1:
            temp_vector[i] = density_matrix[i, j] > threshold_bias + thresholds[j]

        # assign the temp_vector to the jth column of the binary_matrix
        binary_matrix[:, j] = temp_vector

    return binary_matrix
```

Listing 8: Pseudo-Code v103 (2014-10-30):: Creating Binary Matrix

```
define compute_binary_mask(dens, threshold_bias, thresholds):
    binary_matrix = array([[dens[j,i] > threshold_bias+max_values[i] for i in xrange(len(
    dens[j]))] for j in xrange(len(dens))])

    return binary_matrix
```

Listing 9: Python Code v103 (2014-10-30):: Creating Binary Matrix

At this point in the algorithm, the first approximation of the cloud areas can be output and plotted, as seen in the following figure (Fig. 5):

[Output: cloud areas (approximate), {date}bin5.out]

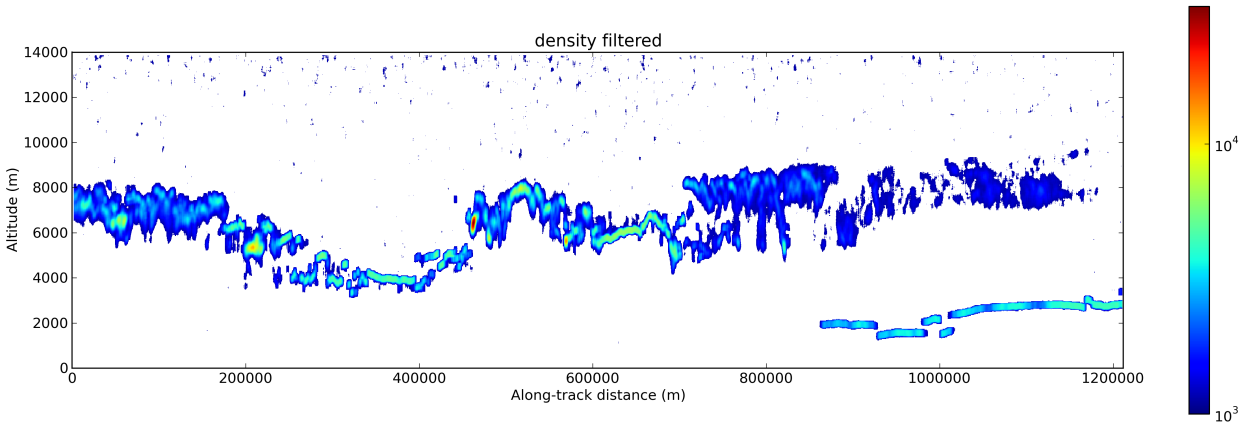


Figure 5. Binary matrix applied to density. MABEL data set 02Apr12.02

3.3.4 Method B for Auto-Adaptive Determination of Thresholds Using the Density-Dimension Algorithm

The analysis of M-ATLAS data from 2013 MABEL data necessitated analyzing data sets of different noise, background and data characteristics.

An alternative thresholding method was created to allow finer adaptation of threshold levels to variable conditions. In the density calculation, the weight matrix is normalized so that density represents a weighted average of values (eqn. (15)).

In method A, the neighborhood size is input as a controlling parameter for the kernel, according to

$$n = m = 2r + 1 \tag{17}$$

to create a kernel (weight matrix) of dimensions (m, n) [m in x-direction (along-track direction), n in y-direction (height box in a profile)].

In method B, the kernel dimensions are usually (by default) calculated from the standard deviation σ (in bins) and anisotropy factor a_m (in meters), according to

$$n = \text{int}(2\ \text{ceil}(\frac{\sigma}{y_{res}}\ \text{cutoff})) + 1 \tag{18}$$

$$m = \text{int}(2\ \text{ceil}(\frac{\sigma}{x_{res}}\ \text{cutoff}\ a_m)) + 1 \tag{19}$$

where σ=σ_{bin} is the standard deviation given in bins or pixels (noting that $\sigma_m = 30\sigma_{bin}$ matching the format of the atmospheric data), cutoff the number of standard deviations after which the Gaussian function is cut off (the default is cutoff = 2), a_m the anisotropy factor (given in meters), $y_{res} = 30\ m$ the size of height boxes in a profile and $x_{res} = 280\ m$ the along-track size of boxes, ceil denotes the ceiling function (the smallest integer larger than the value; i.e. ceil(4.9) = 5 and ceil(4) = 4, and int denotes the entire or integer function; i.e. int(4.9) = 4 and int(4) = 4. If a neighborhood value r is prescribed in addition to σ and a_m, then the kernel is of size $(2r+1, 2r+1)$ with matrix values determined using σ and a_m symmetrical to the kernel center (as for method A). Method B can use normalized kernel values or not normalized values (see flow diagrams). Method A does not use normalized kernel values.

To give an example, for the analysis of the M-ATLAS triple data set (see Fig. 19) the following values are used (see also Table 1b): For density run 1, $\sigma = 3$ and $a_m = 10$, hence $n = 13$ and $r_{1,y} = 6$, the half-size in y-direction; for the x-direction, $m = 13$ and $r_{1,x} = 6$ as well. For density run 2, $\sigma = 6$ and $a_m = 20$, hence $n = 25$ and $r_{2,y} = 12$, the half-size in y-direction; for the x-direction, $m = 49$ and $r_{2,x} = 24$.

Next, the 90% quantile is calculated for an along track moving window of 20 columns (about 6 km). Then, the threshold is calculated by multiplying by an overall sensitivity factor and then adding an overall bias. The sensitivity factor scales the influence of the local variation, while the bias modifies the overall density threshold (it is 1 so far).

threshold[i] = threshold_bias + threshold_sensitivity * (90% quantile in window at i)

Parameters. A threshold bias of 70 is used as a default. In the example using the MABEL-based triple data set (Fig. 19), a threshold bias of 60 is used (see also Table 2b, c). Other examples are given in the applications section (6) . A threshold-sensitivity of 1 is used.

This method version is used in application example M-ATLAS 2013 - see Application section (6.3).

```
define compute_thresholds(density_matrix, threshold_bias, threshold_sensitivity, window):
    # computes vector of thresholds, one for each column in the data
    # arguments:
    # density_matrix – matrix of computed densities
    # threshold_bias – overall bias to thresholds
    # threshold_sensitivity – sensitivity of thresholds to shot variation
    # window – window (in pixels) to use for shot variation

    [number_of_rows, number_of_columns] = shape(density_matrix)

    # initialize to vector with same length.
    thresholds = zeros(number_of_columns)
    for i in 0 to number_of_columns-1:
        # indices which bracket columns to use for threshold calculation, clipped
    at left and right ends
        lower_index = max(0, i-round(window/2))
        upper_index = min(number_of_columns-1, i+round(window/2))
```

```
        # each column threshold is a linear combination of threshold bias and
    window column quantile
        thresholds[i] = threshold_bias + threshold_sensitivity*quantile(density_matrix[:,
    lower_index:upper_index], .9)

    return thresholds
```

Listing 10: Pseudo-Code v103 (2014-10-30):: Determination of Thresholds (Method B)

```
define compute_thresholds(density_matrix, threshold_bias, threshold_sensitivity, window):
    n,m = data.shape
    thresholds = zeros((1,m))
    for i in xrange(m):
        sl = slice(max(0, i-floor(window/2)), min(m, i+floor(window/2)))
        thresholds[0,i] = threshold_bias + mquantiles(data[:,sl], .9) * threshold_sensitivity

    return thresholds
```

Listing 11: Python Code v103 (2014-10-30):: Determination of Thresholds (Method B)

3.3.5 Application of Thresholds to Derive First Cloud Mask (Binary Matrix), Method B

The pseudo-code and pycode for application of thresholds to derive a first cloud mask is then as followts:

```
define compute_binary_mask(density_matrix, thresholds):
    # computes binary mask of cloud regions
    # arguments:
    # density_matrix - matrix of computed densities
    # threshold_bias - overall threshold bias (default 70)
    # thresholds - vector of thresholds

    # shape returns the size of the matrix data
    [number_of_rows, number_of_columns] = shape(density_matrix)

    # binary_matrix is a matrix size number_of_rows,number_of_columns filled with
    zeros
```

```
    binary_matrix = zeros([number_of_rows, number_of_columns])

    # j iterates over integers from 0 to number_of_columns
    for j in 0 to number_of_columns-1:
        # temp_vector is a vector size number_of_rows by 1
        # elements in temp_vector are 1 where the density_matrix value is greater
    than the threshold for the jth column
        for i in 0 to number_of_rows-1:
            temp_vector[i] = density_matrix[i, j] > thresholds[j]

        # assign the temp_vector to the jth column of the Binary_matrix
        binary_matrix[:, j] = temp_vector

    return binary_matrix
```

Listing 12: Pseudo-Code v103 (2014-10-30):: Application of Thresholds to Derive First Cloud Mask (Binary Matrix) - Method B

```
define compute_binary_mask(density_matrix, thresholds):
    binary_matrix = density_matrix - thresholds > 0

    return binary_matrix
```

Listing 13: Python Code v103 (2014-10-30):: Application of Thresholds to Derive First Cloud Mask (Binary Matrix) - Method B

3.3.6 Synthesis of Methods A and B

3.3.6.1 Where do the Differences Occur?

There are two parts of the code where there is a difference between methods A and B:

(1) In the calculation of the density field

(2) In the threshold determination

3.3.6.2 Density Field Calculation

Methods A and B control the weight matrix/ the kernel differently, by means of different parameters. These have already been synthesized in v5.0 atbd. The remaining differences are the following:

Method B uses scaling (normalization) by the sum of weights in equation (15), whereas Method A uses scaling by the maximum density value found in the neighborhood used for kernel calculation. However, in future we will only use method B, as in code v103, to calculate density, with the normalization on (logical "true" in Table 2b,d). Normalization "true" is also used in the Method A/B synthesis.

Results of earlier analyses can still be reproduced with code version v105.0 (and later versions) by using different parameters to control the kernel. Table 2d also tells how to do this.

The kernel is controlled by the following parameters:

(1) The standard deviation, σ, of the radial basis function, as used in equations (1-4) in section (M.1). We distinguish $\sigma = \sigma_{bin}$, the standard deviation given in bins or pixels, matching the format of the atmospheric data, and $\sigma_m = 30\sigma_{bin}$.

(2) The anisotropy factor, a, as described in section (M.2) and equations (5-12).

(3) The number of standard-deviations used, termed *cutoff*

as described for method B in section 3.3.4, from which kernel size (n, m) is derived. Alternatively, the kernel can be controlled by prescribing

(1) kernel size (n, m),

(2) the standard deviation $\sigma = \sigma_{bin}$

(3) anisotropy a_m

in which case the parameter *cutoff* will be calculated internally.

Knowing this, density field calculation can be implemented as for method B.

3.3.6.3 Threshold Determination

Threshold determination in Method A depends on the following parameters (see 3.3.2-3.3.3):

$$f_A = f_A(bin = downsampling, bin_count, threshold_segment_length_downsampled, threshold_bias) \tag{20}$$

Threshold determination in Method B depends on the following parameters (see 3.3.4-3.3.5):

$$f_B = f_B(downsampling, threshold_segment_length, threshold_bias, threshold_sensitivity) \tag{21}$$

where *threshold_segment_length* is the same as *downsampling* length in the original matrix,

threshold_bias is additive and *threshold_sensitivity* is multiplicative.

Note that *threshold_bias*, or any other additive component in the threshold determination, will have to be scaled by the range-squared factor that comes in with any correction!

Method A uses

threshold[i] = threshold_bias + 1 * (bin_count=max_density in 5x5 neighborhood)

threshold[i] = threshold_bias + 1 * (50% quantile in downsampled matrix at profile i)
= threshold_bias + 1 * (50% quantile in downsampled matrix of bin_count=max_density in 5x5 (bin x bin) neighborhood of orig mx)

where downsampled matrix is (for $bin = 5$)

$$a^*(i,j) = max(a(i_k, j_r)_{\{i_k=-2,2,j_r=-2,2\}}) = max(a(i_k, j_r)_{\{i_k=-[bin/2],[bin/2],j_r=-[bin/2],[bin/2]\}})$$

In words, Method A proceeds as follows: Downsample the original density matrix by 5 ($bin = downsampling$), by replacing the density value in a central point (i, j) by the maximal density in its (5x5), or (bin x bin), neighborhood in the original density matrix.

This yields a matrix with 100 rows. The threshold is the 50th-largest value in the downsampled matrix. This corresponds to a 50% quantile (the median). For the "new" 700-row original matrix,

one needs to take care to handle the "NaN" values properly, that stem from the embedding of the data profile into the DEM/Geoid reference frame. Note that the 50% quantile is taken in the downsampled matrix. *threshold_segment_length_downsampled* is 1 in the examples from 2012 data (i.e. the along-track windowing effect is only 5 profiles, equal to 1400 m). By introducing the variable *threshold_segment_length_downsampled*, we are allowing the possibility of using larger along-track windows.

Here is the motivation for method A: If identifying ground, it is possible to find a window that is definitely noise by going 150m above the ground. For clouds, this is not possible, since clouds or aerosols can be encountered at any height above the surface. The idea of Method A is to find a collection of small non-cloud regions (noise regions) that together form the noise area, and the threshold functions to separate noise characteristics from cloud/aerosol/layer characteristics. The downsampling process makes sure that we are not looking at individual pixels but at small non-cloud areas. This is an important concept.

Now to method B:

Method B uses

threshold[i] = threshold_bias + threshold_sensitivity * (90% quantile in window at i)

i.e., the threshold is determined in the original matrix, leaving the downsampling concept out, but using several profiles (*downsampling* profiles, e.g. 5 or 20 profiles) to create the window for taking the quantile. This explains why the quantile is lower. However, the concept of the small non-clouds is missing, and this will be reintroduced in the synthesized method A/B.

The method A/B synthesis uses the following steps for threshold determination:

$$f_{A,B} = f_{A,B}(downsampling, bin_count, threshold_segment_length_downsampled, threshold_bias, threshold_sensitivity) \tag{22}$$

and proceeds by the following steps (labeled T.x for threshold-determination steps):

Step (T.1) Downsample original density matrix by a factor $bin = downsampling$ (e.g. bin=5)

Step (T.2) Take maximum value in a (5x5) (binxbin) neighborhood of the point (i,j), using the eqn in the box below, where $A = (a(i,j))_{i,j}$ is the original density matrix and $A^* = (a^*(i,j))_{i,j}$ is the downsampled matrix:

$$a^*(i,j) = max(a(i_k, j_r)_{\{i_k=-2,2,j_r=-2,2\}}) = max(a(i_k, j_r)_{\{i_k=-[bin/2],[bin/2],j_r=-[bin/2],[bin/2]\}})$$

Step (T.3) Use *threshold_segment_length_downsampled* for the window size in the downsampled matrix, if additional averaging is desired (e.g. 20)

Step (T.4) Take a quantile (default 50% quantile) in downsampled matrix for *threshold_segment_length_downsampled* columns

Step (T.5) Use the second equation with S=threshold_segment_length_downsampled

threshold[i] = threshold_bias + threshold_sensitivity * (50% quantile in window in downsampled mx at i)

threshold[i] = threshold_bias + threshold_sensitivity * (50% quantile for S columns in downsampled matrix at column i)
= threshold_bias + threshold_sensitivity * (50% quantile in S columns in downsampled matrix of bin_count=max_density in 5x5 (bin x bin) neighborhood of orig mx at col i in downsampled mx)

Allowing the quantile to be a parameter, *quantile* Q, we get

threshold[i] = threshold_bias + threshold_sensitivity * (Q% quantile for S columns in downsampled matrix at column i)
= threshold_bias + threshold_sensitivity * (Q% quantile in S columns in downsampled matrix of bin_count=max_density in 5x5 (bin x bin) neighborhood of orig mx at col i in downsampled mx)

To get f_A from $f_{A,B}$:

1. In Step (T.1), set $downsampling = bin = 5$.
2. Keep Step (T.2) as is.
3. In Step (T.3), use $threshold_segment_length_downsampled = 1$.
4. In Step (T.4), use Q=50%.
5. In Step (T.5), use $threshold_bias = 70$ and $threshold_sensitivity = 1$.

To get f_B from $f_{A,B}$:

1. In Step (T.1), set $downsampling = 1$.
2. Keep Step (T.2) as is. Note that since in (T.1) $downsampling = 1$, the downsampling by maximum in a 1 x 1 neighborhood will default to the identity operation, i.e. with these setting we are effectively skipping Steps (T.1) and (T.2).
3. In Step (T.3), use $threshold_segment_length_downsampled = 5$ or $= 20$ (not equal to 1).
4. In Step (T.4), use Q=90%.
5. In Step (T.5), use $threshold_bias = 60$ and $threshold_sensitivity = 1$ (or whatever, i.e. different values were used in the sensitivity studies in the ATBD v.5 and thereafter in the so-called "v10" sensi study).

The threshold function for method A/B synthesis can be implemented based on the equations in the framed boxes and uses the following python code:

There are two listings: (1) Threshold function call and (2) Threshold function definition.

```
            ### Compute thresholds
300         if len(algo.steps) == options.end_step: break
            algo.start_step(Step(name='Compute thresholds',
                    vis_funcs=[plot_thresholds]))
            algo.steps[-1].set_visualize(len(algo.steps) in visualize_steps)
            thresholds = compute_thresholds(density, threshold_bias, threshold_factor,
    downsample, quantile, threshold_segment_length)
305         globals().update(locals())
            algo.steps[-1].done()
```

Listing 14: Python Code v106.0 (2016-08-17): Call compute-threshold function

```
def compute_thresholds(data, threshold_bias, threshold_factor, downsample, quantile,
    threshold_segment_length):
140     n,m = data.shape
    thresholds = zeros((1,m))
    num_boxes = ceil(n/downsample)
    maximums = zeros((num_boxes,m))
    bottom_index = arange(0,n,downsample)
145
    # Get maximums for each downsample-by-downsample bin for each column
    if downsample > 1:
        for i in xrange(m):
            counter = 0
150         s2 = slice(max(0, i-floor(downsample/2)), min(m, i+floor(downsample/2)))
            for bi in nditer(bottom_index):
                s1 = slice(bi, bi + downsample -1);
                max_temp = nanmax(data[s1,s2])
                if not max_temp:
155                 maximums[counter,i] = nan
                else:
                    if max_temp < 0:
                        maximums[counter,i] = 0
                    else:
```

```
                    maximums[counter,i] = max_temp
            counter = counter +1
    else:
        maximums = data



    # Get quantile for each num_boxes-by-2*threshold_segment_length-1 bin
    for j in range(m):
        sl = slice(max(0, j-threshold_segment_length), min(m, j + threshold_segment_length)
    +1)
        thresholds[0,j] = threshold_bias + mquantiles(maximums[:,sl],quantile) *
    threshold_factor



    return thresholds
```

Listing 15: Python Code v106.0 (2016-08-17): Compute-threshold function

These listing, taken from Python Code v106.0 (2016-08-17), are essentially identical to the listings from v105 that were given in ATBD v6.0 (October 2015).

Parameter: The size of the bins in y-direction is still a fixed parameter ($y_{res} = 30$) and should be changed to $y_{res} = 29.9$ as determined August 2016.

Examples of Data Analyses Using Method A/B for Threshold Determination

Method Method A/B for threshold determination has been applied in all recent data analyses since 2015, especially in the current (August 2016) state-of-the art analyses of GLAS-based simulated ICESat-2 data. Examples are given in section (8). The standard test runs for code implementation are

(t3) for a double-density run (see Figure 30)

(t8) for a single-density run (see Figure 31-1)

Furthermore, this version is used in the study of automated adaptability and validation in the twice-around-the-Earth-run in section (8) and in the sensitivity studies in section (9) ["The Movie"]. All algorithm-specific parameters are given with the respective figures in sections (8) and (9).

1. Check whether f_A creates a threshold vector for each column in the original mx (i.e. does the whole operation for each profile) or for each column in the downsampled mx only (Note that a new column vector is created for each column in the original mx after thresholding, but it may use the same threshold for 5 columns).

2. The first implementation (v104) of the synthesized method $f_{A,B}$ does the operation for each column in the orig mx = for each profile, but only for every 5th index in height=row, to save time.

3. Implemented in v104 using an offset k to calculate *threshold_seg_length* from $2k + 1$ in the downsampled mx

4. then *threshold_seg_length*[*pix*]=*bin* x *threshold_seg_length*

5. and *threshold_seg_length*[*m*]=*bin* x *threshold_seg_length* x 280

6. since NRB values can be negative, replace negative NRB values by 0.

To illustrate how to use the method A/B synthesis, the new code version v104 is applied to GLAS-data-based simulated ICESat-2 data, using all default values, and the result is shown in Figure AB.1. The log is given on the next page. An application with better parameters and a new sensitivity study are presented in sections (8) and (9), using v105.

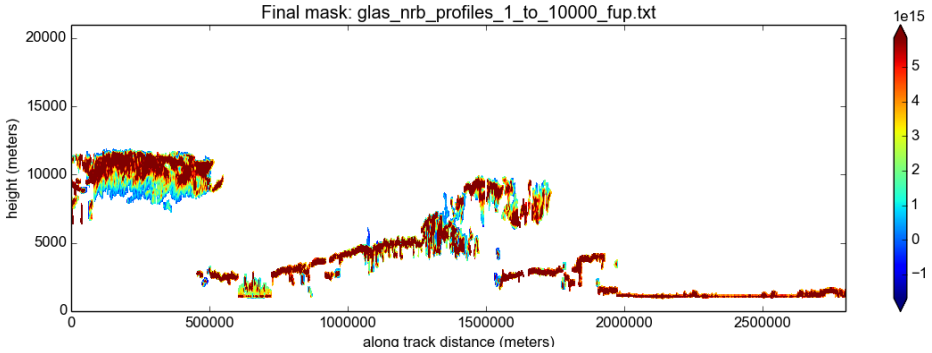


Figure AB.1. Application of the new synthesized method A/B to simulated ICESat-2 data, based on 2 orbits of GLAS data, NRB values, first 10,000 profiles. For log file, see next page.

```
arce78-16-dhcp:plotsAB uch$ more glas_nrb_1_10000_test4.log
2015-08-31 18:01:14,974 - atmos_algo - INFO - Pixel resolution: 280x30 meters
2015-08-31 18:01:14,975 - atmos_algo - INFO - Plot directory: plots/glas_nrb_1_10000_test4
2015-08-31 18:01:14,975 - atmos_algo - INFO - Aniso factor(s) (1=isotropic): [10.0, 20.0]
2015-08-31 18:01:14,975 - atmos_algo - INFO - Sigma(s): [3.0, 6.0] pixels (vertically)
2015-08-31 18:01:14,975 - atmos_algo - INFO - Cutoff(s): [1.0]*sigma(s)
2015-08-31 18:01:14,975 - atmos_algo - INFO - Neighborhoods: [None] pixels
2015-08-31 18:01:14,975 - atmos_algo - INFO - Downsample: [5] pixels
2015-08-31 18:01:14,975 - atmos_algo - INFO - Threshold factors(s): [3.0, 2.0]*100%
2015-08-31 18:01:14,975 - atmos_algo - INFO - Base threshold(s): [70.0, 0.0]*100%
2015-08-31 18:01:14,976 - atmos_algo - INFO - Minimum cluster size: 600 pixels
2015-08-31 18:01:14,976 - atmos_algo - INFO - Threshold segment length(s): [2, 2]
2015-08-31 18:01:14,976 - atmos_algo - INFO - Quantiles(s): [0.5]
2015-08-31 18:01:14,976 - atmos_algo - INFO - STEP 1
2015-08-31 18:01:14,976 - atmos_algo - INFO - Load data step starting
2015-08-31 18:01:18,352 - atmos_algo - INFO - Loaded ../../../data/GLAS_atlas_sim/glas_nrb_profiles_1_to_10000_fup.txt
2015-08-31 18:01:18,352 - atmos_algo - INFO - Load data step done
2015-08-31 18:01:21,738 - atmos_algo - INFO - STEP 2
2015-08-31 18:01:21,738 - atmos_algo - INFO - Compute density step starting
2015-08-31 18:01:21,768 - atmos_algo - INFO - Kernel shape: 7x9
2015-08-31 18:01:22,233 - atmos_algo - INFO - Compute density step done
2015-08-31 18:01:25,697 - atmos_algo - INFO - STEP 3
2015-08-31 18:01:25,697 - atmos_algo - INFO - Compute thresholds step starting
2015-08-31 18:05:20,907 - atmos_algo - INFO - Compute thresholds step done
2015-08-31 18:05:21,053 - atmos_algo - INFO - STEP 4
2015-08-31 18:05:21,053 - atmos_algo - INFO - Apply threshold filter step starting
2015-08-31 18:05:21,094 - atmos_algo - INFO - Apply threshold filter step done
2015-08-31 18:05:31,132 - atmos_algo - INFO - STEP 5
2015-08-31 18:05:31,132 - atmos_algo - INFO - Remove small clusters step starting
2015-08-31 18:05:31,240 - atmos_algo - INFO - Remove small clusters step done
2015-08-31 18:05:40,840 - atmos_algo - INFO - STEP 6
2015-08-31 18:05:40,840 - atmos_algo - INFO - Delete data step starting
2015-08-31 18:05:40,847 - atmos_algo - INFO - Delete data step done
2015-08-31 18:05:40,847 - atmos_algo - INFO - STEP 7
2015-08-31 18:05:40,848 - atmos_algo - INFO - Compute density step starting
2015-08-31 18:05:40,863 - atmos_algo - INFO - Kernel shape: 13x27
2015-08-31 18:05:43,098 - atmos_algo - INFO - Compute density step done
2015-08-31 18:05:46,410 - atmos_algo - INFO - STEP 8
2015-08-31 18:05:46,410 - atmos_algo - INFO - Compute thresholds step starting
2015-08-31 18:09:45,140 - atmos_algo - INFO - Compute thresholds step done
2015-08-31 18:09:45,244 - atmos_algo - INFO - STEP 9
2015-08-31 18:09:45,244 - atmos_algo - INFO - Apply threshold filter step starting
2015-08-31 18:09:45,280 - atmos_algo - INFO - Apply threshold filter step done
2015-08-31 18:09:51,635 - atmos_algo - INFO - STEP 10
2015-08-31 18:09:51,635 - atmos_algo - INFO - Remove small clusters step starting
2015-08-31 18:09:51,739 - atmos_algo - INFO - Remove small clusters step done
2015-08-31 18:09:57,307 - atmos_algo - INFO - STEP 11
2015-08-31 18:09:57,307 - atmos_algo - INFO - Delete data step starting
2015-08-31 18:09:57,314 - atmos_algo - INFO - Delete data step done
2015-08-31 18:09:57,314 - atmos_algo - INFO - STEP 12
2015-08-31 18:09:57,314 - atmos_algo - INFO - Combine masks pre-closing step starting
2015-08-31 18:09:57,342 - atmos_algo - INFO - Combine masks pre-closing step done
2015-08-31 18:10:08,349 - atmos_algo - INFO - STEP 13
2015-08-31 18:10:08,350 - atmos_algo - INFO - Compute confidence measure step starting
2015-08-31 18:10:11,668 - atmos_algo - INFO - Compute confidence measure step done
2015-08-31 18:10:14,692 - atmos_algo - INFO - Total running time: 9.00 minutes
```

3.3.7 Quantile Calculation

During testing of the code implementation by the SIPS in October 2017, we discovered that the algorithm used in quantile calculation can contribute significantly to the error in results between two different code implementations. Care needs to be taken when using a library function (python, fortran or any other language). Library functions typically only differ in the interpolation step between actually occurring values, but since near the threshold values used in the DDA the density values are relatively scarce, this difference matters. The effect is illustrated in section (12) "Testing". The old and new algorithms are described here.

mquantiles

The original quantile implementation (v103.0-109.0) used python's mquantiles function (scipy.stats.mstats.mquantiles). The mquantiles function finds the empirical quantile for a data array. Sample quantiles are defined by

$$Q(p) = (1 - \gamma) \cdot x_j + \gamma \cdot x_{j+1}$$

where x_j is the j-th order statistic of an array A, and γ is a function of $j = floor(n \cdot p + m)$, of $m = \alpha_p + p \cdot (1 - \alpha_p - \beta_p)$ and of $g = n \cdot p + m - j$. The value $p = p(k)$ is given by

$$p(k) = (k - \alpha_p)/(n + 1 - \alpha_p - \beta_p)$$

for some statistic order k of the data array. The *mquantiles* function takes values of α_p and β_p as optional inputs. If unspecified, the defaults values of $\alpha_p = 0.4$ and $\beta_p = 0.4$ are used which give an approximately unbiased quantile. We used these default values in our original implementation. For more information see (https://docs.scipy.org/doc/scipy-0.19.1/reference/generated/scipy.stats.mstats.mquantiles.html).

"Rounding"

A problem with the *mquantiles* function is that it only allows to specify continuous rather than discrete functions for association of quantiles. A "discrete" function will only use those density values that actually occur in the distribution, whereas a continuous function will interpolate between

actually occurring values. As the closest option to a discrete function, we used linear interpolation of values (see Figure 39-1 and 39-2) to test the sensitivity of quantile and hence threshold function values to the quantile implementation.

The current version (v110.0) uses the term "Rounding", referring to a discrete association function for quantiles, which follows the definition of a quantile in its simple form. New code is written for this, rather than using python's *mquantiles* function.

A short description of the "rounding" quantile algorithm is as follows:

Given an array A and a quantile fraction $p \leq 1$, we find the the p-quantile of the array by first ordering the array (lowest to highest magnitude) and removing any missing values. The size of the array A with no missing values is given by n. The index i_p of the p-quantile in A is then found by

$$i_p = round(p * n)$$

Thus, the p- quantile of the data array A is given by

$$Q(p) = A(i_p)$$

```
def compute_thresholds(image):
  """
  The compute_thresholds function defines the per profile threshold value along
    track and sets the image.thresholds attribute
  """
  (n,m) = image.density.shape
  image.thresholds = np.zeros((1,m))

  # Extract necessary parameter values from image.parameters dictionary
  threshold_window = image.parameters['threshold_window']
  threshold_bias = image.parameters['threshold_bias']
  threshold_factor = image.parameters['threshold_factor']
  quantile = image.parameters['quantile']

  # Mask density so we don't introduce invalid data into thresholding
  valid_density = np.ma.array(image.density, mask=image.valid_mask, fill_value=np.nan).filled
    ()
```

```
  # Loop through each profile and assign a threshold
  for j in range(m):
    sl = slice(max(0, j-threshold_window), min(m, j + threshold_window)+1)
    # New quantile a la jesse, i.e. nearest integer, 11/9/2017 (Tom)
    quantile_data = valid_density[:,sl] # Data for which quantile is computed (i.e. a
    column of coluns depending on window size)
    quantile_data_1d = np.ravel(quantile_data) # turn the data into a 1D array
    quantile_data_1d_nonans = quantile_data_1d[~np.isnan(quantile_data_1d)] # remove nans
    quantile_sort = np.sort(quantile_data_1d_nonans) # sort the data lowest to highest
    quantile_length = len(quantile_sort) #number of entries in quantile data
    quantile_index = int(round(quantile*quantile_length))
    quantile_val = quantile_sort[quantile_index-1] # minus 1 since python index starts
    at 0 (where FORTRAN starts at 1)


    image.thresholds[0,j] = threshold_bias + quantile_val * threshold_factor
```

Listing 16: Python Code v110.0 (2017-11-17): Quantile Calculation as part of Threshold Calculation.

3.4 Step 4: Removal of Small Clusters: Determination of Cloud Areas - Final

The idea of this step is to remove small clusters of points that are not actually clouds (artifact removal).

After application of the density-based noise filter (in step 3), small clusters of noise points remain. In this step, small clusters are removed. The search for neighbors in a cluster is carried out in 4 directions. In algorithm v4, the algorithm is applied with a cluster size of 300. Another common value is 600. At this step, an iteration through the density field (resultant from step 3) is performed. Whenever a cluster of size smaller than 300 is encountered, that cluster is removed. The result is a density array $dens_filtered_small - clust - removed$, this is the region of clouds as identified by the algorithm.

The set of points determined is $dens_filtered_small - clust - removed$

Parameter: the size of small clusters is 300 points or less (size_threshold = 300), in later version, a size of 600 points is used. Cluster size is an algorithm specific parameter and varies as listed in Tables 5 and 6.

In the current implementation (v106.0, 2016-08-17 and v105, October 2015), the "remove-small-clusters" routine uses a built-in python function from the "morphology" package called remove_small_objects. Documentation can be found at

`http://scikit-image.org/docs/dev/api/skimage.morphology.html#skimage.morphology.remove_small_objects`

```
320             ### Minimum cluster filter
                if len(algo.steps) == options.end_step: break
                algo.start_step(Step(name='Remove small clusters',
                        vis_funcs=[plot_boundary_masked_density]))
                algo.steps[-1].set_visualize(len(algo.steps) in visualize_steps)
325             level_mask = logical_not(morphology.remove_small_objects(
                        logical_not(density.mask), min_size=min_cluster))
                density.mask = level_mask
                level_masks.append(level_mask)
```

```
    globals().update(locals())
330 algo.steps[-1].done()
```

Listing 17: Python Code v106.0 (2016-08-17): Remove small clusters.

For earlier versions of the code, we developed a set of routines for small-cluster removal (also called: remove artifacts), which is described in the following:

The listing for the small-cluster removal has three parts, because it is coded using two subroutines, function 1 and function 2.

(1) Function1 for Small Cluster Removal (Neighbors Search)

(2) Function2 for Small Cluster Removal (Replace Values)

(3) Small Cluster Removal

```
define neighbors_search([i, j], binary_matrix):
    # find non-zero neighbors of a pixel
    # arguments:
    # [i,j] - size 2 vector of indices
    # binary_matrix - binary matrix to search

    # set neighbors to be a empty list
    neighbors = []

    # check to make sure the neighbor index is valid
    if i < number_of_columns-1 and j < number_of_rows-1:
        # check if neighbor index contains a 1:
        if binary_matrix[i+1,j+1] == 1: neighbors.append([i+1,j+1])
        # append adds the item to the end of the list neighbors

    # check to make sure the neighbor index is valid
    if i > 0 and j > 0:
        # check if neighbor index contains a 1:
        if binary_matrix[i-1,j-1] == 1: neighbors.append([i-1,j-1])
        # append adds the item to the end of the list neighbors

    # check to make sure the neighbor index is valid
    if i > 0 and j < number_of_rows-1:
        # check if neighbor index contains a 1:
        if binary_matrix[i-1,j+1] == 1: neighbors.append([i-1,j+1])
        # append adds the item to the end of the list neighbors
```

```
    # check to make sure the neighbor index is valid
    if i < number_of_columns-1 and j > 0:
        # check if neighbor index contains a 1:
        if binary_matrix[i+1,j-1] == 1: neighbors.append([i+1,j-1])
        # append adds the item to the end of the list neighbors

    # check to make sure the neighbor index is valid
    if i < number_of_columns-1:
        # check if neighbor index contains a 1:
        if binary_matrix[i+1,j] == 1: neighbors.append([i+1,j])
        # append adds the item to the end of the list neighbors

    # check to make sure the neighbor index is valid
    if i > 0:
        # check if neighbor index contains a 1:
        if binary_matrix[i-1,j] == 1: neighbors.append([i-1,j])
        # append adds the item to the end of the list neighbors

    # check to make sure the neighbor index is valid
    if j < number_of_rows-1:
        # check if neighbor index contains a 1:
        if binary_matrix[i,j+1] == 1: neighbors.append([i,j+1])

    # check to make sure the neighbor index is valid
    if j > 0:
        # check if neighbor index contains a 1:
        if binary_matrix[i,j-1] == 1: neighbors.append([i,j-1])

    # return all the neighbors found
    return neighbors
```

Listing 18: Pseudo-Code v103 (2014-10-30): Function1 for Small Cluster Removal (Neighbors Search)

```
def neighbors_search(i,j,bin):
    neighbors = []
    i_len = len(bin)
    j_len = len(bin[0])
    if i < i_len-1 and j < j_len-1 and bin[i+1,j+1] == 1: neighbors.append([i+1,j+1])
    if i > 0 and j > 0 and bin[i-1,j-1] == 1: neighbors.append([i-1,j-1])
```

```
    if i > 0 and j < j_len-1 and bin[i-1,j+1] == 1: neighbors.append([i-1,j+1])
    if i < i_len-1 and j > 0 and bin[i+1,j-1] == 1: neighbors.append([i+1,j-1])
    if i < i_len-1 and bin[i+1,j] == 1: neighbors.append([i+1,j])
    if i > 0 and bin[i-1,j] == 1: neighbors.append([i-1,j])
    if j < j_len-1 and bin[i,j+1] == 1: neighbors.append([i,j+1])
    if j > 0 and bin[i,j-1] == 1: neighbors.append([i,j-1])
    return neighbors
```

Listing 19: Python Code v103 (2014-10-30): Function1 for Small Cluster Removal (Neighbors Search)

```
define points_to_value(list_of_points, value, binary_matrix):
    # fill locations in matrix with a value
    # arguments:
    # list_of_points - list of size 2 vectors containing indices
    # binary_matrix - matrix to have elements filled

    for points in list_of_points:
        binary_matrix[points] = value

    return binary_matrix
```

Listing 20: Pseudo-Code v103 (2014-10-30): Function2 for Small Cluster Removal (Replace Values)

```
def points_to_val(t,val,bin):
    for pt in t:
        bin[pt[0],pt[1]] = val
    return bin
```

Listing 21: Python Code v103 (2014-10-30): Function2 for Small Cluster Removal (Replace Values)

Using the two functions listed above, the small cluster removal is the performed as follows:

```
define remove_small_clusters(Binary_matrix, size_threshold):
    # Remove connected regions with less than minimum number of pixels.
    # Arguments:
    # Binary_matrix - binary matrix of cloud regions
    # Size_threshold - the minimum size of the shapes kept
```

```
    # declare Filtered_matrix as a copy of Binary matrix
    Filtered_matrix = copy(Binary_matrix)

    [number_of_rows, number_of_columns] = shape(Binary_matrix)
    # i iterates over integers from 0 to number_of_rows
    for i in 0 to number_of_rows:
        # j iterates over integers from 0 to number_of_columns
        for j in 0 to number_of_columns:

            # if the current point is has value of 1:
            if bin[i][j] == 1:
                #define search_list to contain the current point
                search_list = [[i,j]]

                #define total_list to contain all points in the current shape
                total_list = [[i,j]]

                #while there are still points to search:
                while length(search_list) != 0:
                    #remove the last element of the search list
                    temp_i, temp_j = search_list.pop()

                    #find the immediate neighbors with value 1 around temp_i,temp_j
                    temp_neighbors = neighbors_search([temp_i,temp_j],Filtered_matrix)

                    #Add all the new neighbors to the search_list
                    search_list=cat(search_list,temp_neighbors)

                    #Add all the new neighbors to the total_list
                    total_list = cat(total_list,temp_neighbors)

                    #Switch the current values of the new neighbors to 2 so they are
     not found again.
                    Filtered_matrix = points_to_value(neighbors,2,Filtered_matrix)

                #if the number of points in the total shape is less then the
    size_threshold:
                if length(total_list) < size_threshold:
                    #remove the points from the Filtered matrix be setting their
```

```
    values to 0
                Filtered_matrix = points_to_value(total_list,0,Filtered_matrix)

    #Replaces all the values of 2 to 1
    replace(2,1,Filtered_matrix)

    return Filtered_matrix
```

Listing 22: Pseudo-Code v103 (2014-10-30): Small Cluster Removal

```
def remove_small_clusters(binary_matrix, min_cluster):
    for scan_i in xrange(len(binary_matrix)):
        for scan_j in xrange(len(binary_matrix[0])):
            if binary_matrix[scan_i,scan_j] == 1:
                n = [[scan_i,scan_j]]
                all_n = [[scan_i,scan_j]]
                while len(n):
                    ni, nj = n.pop()
                    sys.stderr.write("%d,%d: neigh %d,%d\r" % (scan_i,scan_j,ni,nj))
                    t=neighbors_search(ni,nj,binary_matrix)
                    n+=t
                    all_n+=t
                    binary_matrix = points_to_val(t,2,binary_matrix)

                if len(all_n) < min_cluster:
                    binary_matrix = points_to_val(all_n,0,binary_matrix)

    return binary_matrix
```

Listing 23: Python Code v103 (2014-10-30): Small Cluster Removal

At this point in the algorithm, the following are output:

[Output: {date}density_pts300.out]

(see Figure 6).

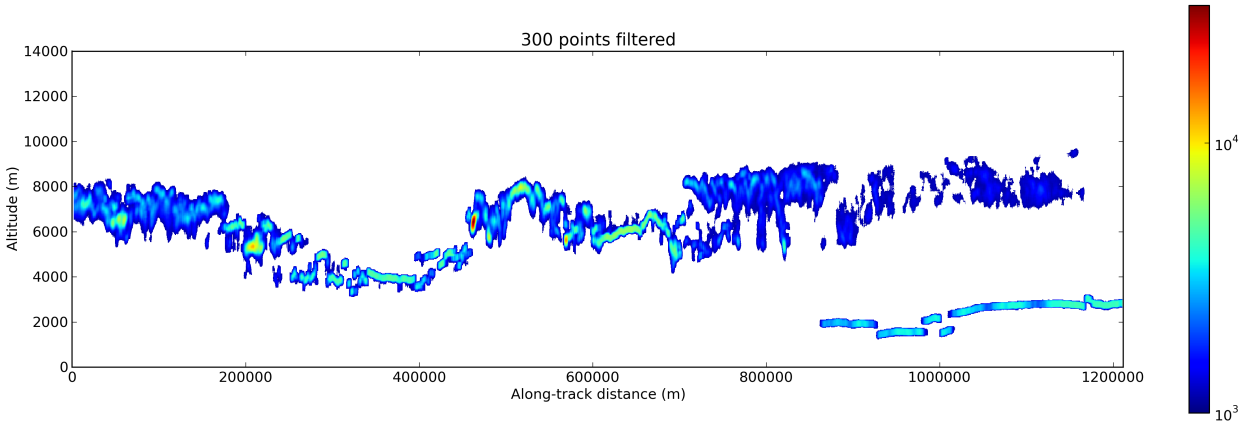


Figure 6. Cloud areas with density, final (after small-cluster removal). MABEL data set 02Apr12.02

3.5 Step 5: Output Data in Cloud Area (Cloud Mask)

The region of clouds $dens_filtered_small - clust - removed$ (determined in step 4) is used as a mask to identify the corresponding data values for cloud areas.

At this point in the algorithm, the following are output (see Figure 7):

[output: {date}data_pts300.png]

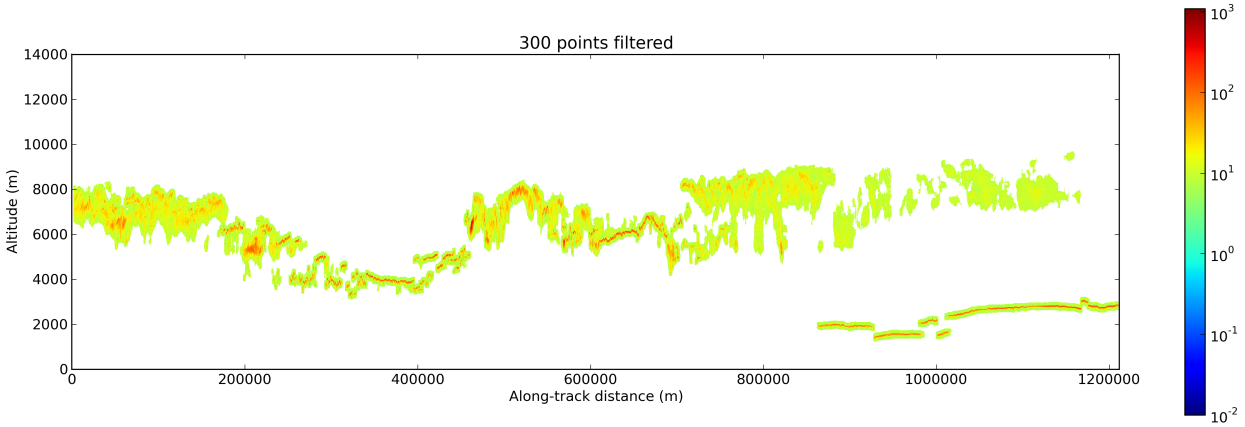


Figure 7. Cloud areas with data, final (after small-cluster removal). MABEL data set 02Apr12.02

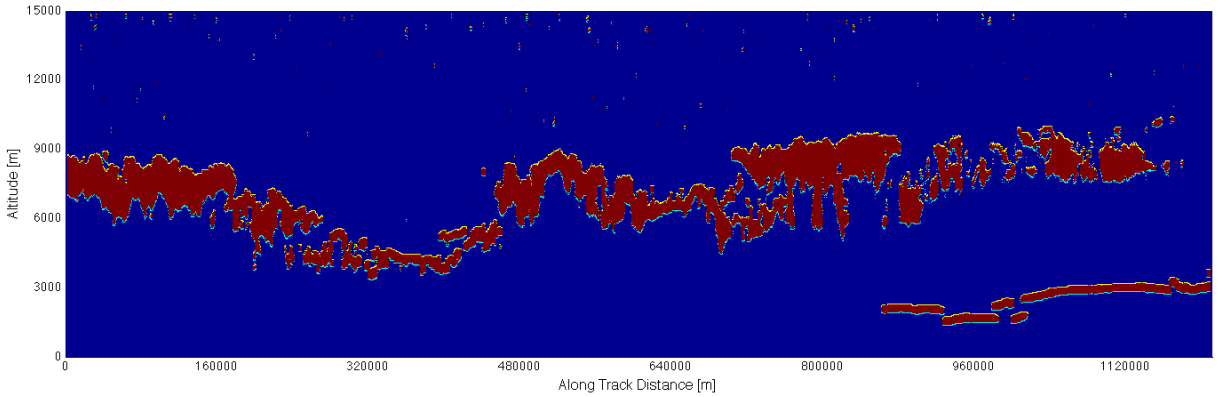


Figure 8-1. Layer boundaries. M-ATLAS Data Set 02Apr12.02, Output using the so-called old algorithm. Top: Yellow, Bottom: Blue, Cloud Region: Red, Up to 6 Atmospheric Layers. An example for MABEL-based data is given in Fig. 19.

3.6 Step 6: Layer Boundaries (Top/ Bottom)

Cloud boundaries can be determined directly from the result of step 5.

The rules for identification of layers are simple:

(i) A layer must be at least 3 bins thick.

(ii) A gap between two layers must be at least 3 bins thick.

(iii) Counting layer dominates over counting gaps.

(iv) Layers are identified per profile.

However, analysis of the problem indicates that an explicit algorithm is needed to implement these rules. Such an algorithm is presented here. Note that bins are counted from top to bottom and layers are counted from top to bottom.

Old Algorithm [(v7.1) and before]

(see section 3.6 “Step 6: Layer Boundaries (Top/Bottom)” in the ATBD, v7.1., p. 74ff.// (atbd.atmos.icesat2.20160

1. Outer loop: move along track

2. Inner loop: move along a vertical profile, from top to bottom, in the binary cloud mask. The mask has "1" for cloud, "0" for non-cloud.

3. Identify the top of a layer (Layer_Top in Table 1) as the location of a 1 in a sequence of 0-1.

4. Identify the bottom of a layer (LayerBot in Table 1) as the location of a 1 in a sequence of 1-0.

5. If two layers are separated by less than 3 bins (90m), they will count as one layer (i.e. the area between the two layers will be considered to be inside one layer).

6. A layer must be at least 3 bins thick (90m).

7. There are a maximum of 6 layers expected, based on knowledge in atmospheric sciences. Hence each variable that is associated with cloud layers is a 6-component vector for each of the three beams (Layer_bot, Layer_top and associated flags, attributed and optical depths, see Table 1 in section 2).

Motivation for an Explicit Algorithm

There are two problems with the old algorithm:

(a) Number of layers: There can be more than 6 layers, as found in several examples of ICESat-2 data simulated from GLAS data. In previous work on layer detection in atmospheric lidar data, 6 was the maximal number of layers that can be identified (Steven Palm), hence 6 was used as the maximal number for layers on the product ATL09. The reason that more than 6 layers can be detected is most likely a result of the ability of the DDA to detect tenuous cloud layers. The number of possible layers has already been increased to 10 (October 2017) by SIPS. Note that ground counts as a layer here.

(b) Identification of layer boundaries:

While the rules for identification of layers are simple, the old algorithm (listed above) can miss "loner bins" on top of an identified cloud, because it performs a pass in only one direction. The new algorithm consists of two passes, a downward pass and an upward pass, each resulting in a mask, and the joint mask of the 2 passes gives the layer boundaries.

An explicit bi-directional algorithm for identification of atmospheric layer boundaries (layerbot, layertop), based on the final cloud mask calculated by the DDA

This algorithm is first correctly implemented in code version v110.0 and developed for v109.0.

To avoid missing "loner bins" above a determined cloud layer, a new, bidirectional layer detection routine is implemented. The algorithm accomplishes this by running a pass of the current version of the layer detection routine in two directions - from the top down, and the bottom up - and saving a partial cloud layer mask from each pass. Then, the union (logical or) of both the upward pass cloud layer mask and downward pass cloud layer mask will produce cloud layers that satisfy rules (i)-(iv). The differences between a one-directional and a bi-directional layer-identification algorithm are exemplified in Table 1.

Bin Number	Binary Cloud Mask	Downward Pass Layer Mask	Upward Pass Layer Mask	Union
1	0	In Cloud	In Cloud	In Cloud
2	0	In Cloud	In Cloud	In Cloud
3	0	In Cloud	In Cloud	In Cloud
4	1	—	—	—
5	1	—	—	—
6	1	—	—	—
7	0	—	In Cloud	In Cloud
8	1	—	In Cloud	In Cloud
9	0	—	In Cloud	In Cloud
10	1	—	In Cloud	In Cloud
11	0	In Cloud	In Cloud	In Cloud
12	0	In Cloud	In Cloud	In Cloud
13	0	In Cloud	In Cloud	In Cloud
14	1	In Cloud	—	In Cloud
15	0	In Cloud	—	In Cloud
16	1	In Cloud	—	In Cloud
17	0	In Cloud	—	In Cloud
18	1	—	—	—
19	1	—	—	—
20	1	—	—	—

Table 1: An example of the improved layer detection routine including all potential loner bins in the appropriate cloud layer by taking the union of two cloud layer masks. The directional layer masks are produced by running a version of the current layer detection routine in both the upward, and downward direction. Binary Cloud Mask is the output of Step 5 of the DDA (the final mask). 0 - cloud, 1 - not cloud.

Algorithm and Pseudo Code

Cloud boundaries are determined directly from the final binary cloud mask resulting from Step 5 as follows:

Step 1: Produce Complete Cloud Layer Mask

1. Outer loop: move along track

   (a) Inner loop (Upward Pass): move along the given vertical profile, from bottom to top, in the binary cloud mask. If 3 consecutive non-masked bins (0s) are encountered, enter a cloud layer. Once in a cloud layer, if 3 consecutive masked bins (1s) are encountered, exit the layer. This produces the upward partial cloud layer mask.

   (b) Inner loop (Downward Pass): move along the given vertical profile, from top to bottom, in the binary cloud mask. Perform the same operations as in the upward pas loop to produce a downward partial cloud layer mask.

2. Take the union of both partial cloud layer masks to create a complete cloud layer mask.

Step 2: Determine Layer Tops and Bottoms, and Tally Total Layers Found

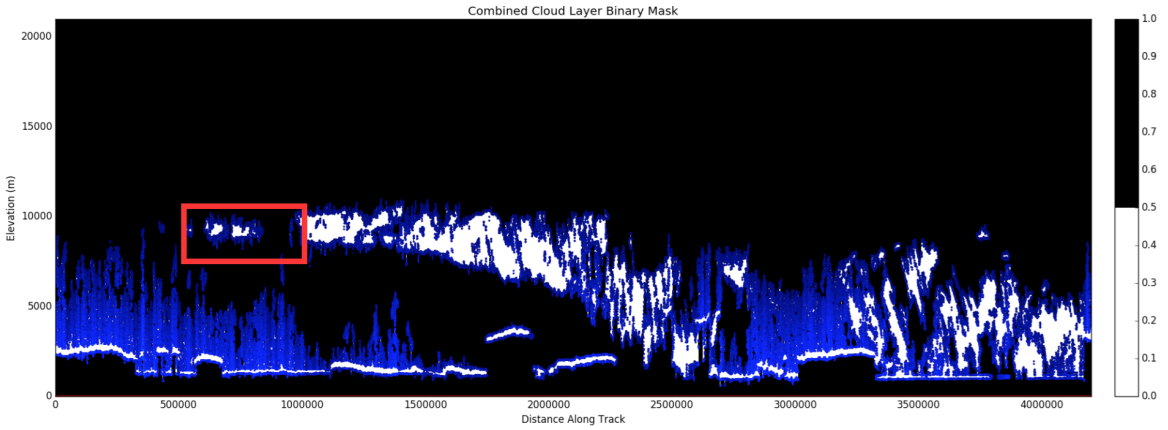
1. Outer loop: move along track

   (a) Inner loop: move along the given vertical profile, from top to bottom, in the complete cloud layer mask produced in Step 1 above. When the complete cloud layer mask indicates that a cloud layer has been entered, add one to the tally of total layers found, and mark this as the top of the appropriate layer. When the complete cloud layer mask indicates that cloud layer has been exited, mark this as the bottom of the appropriate layer.

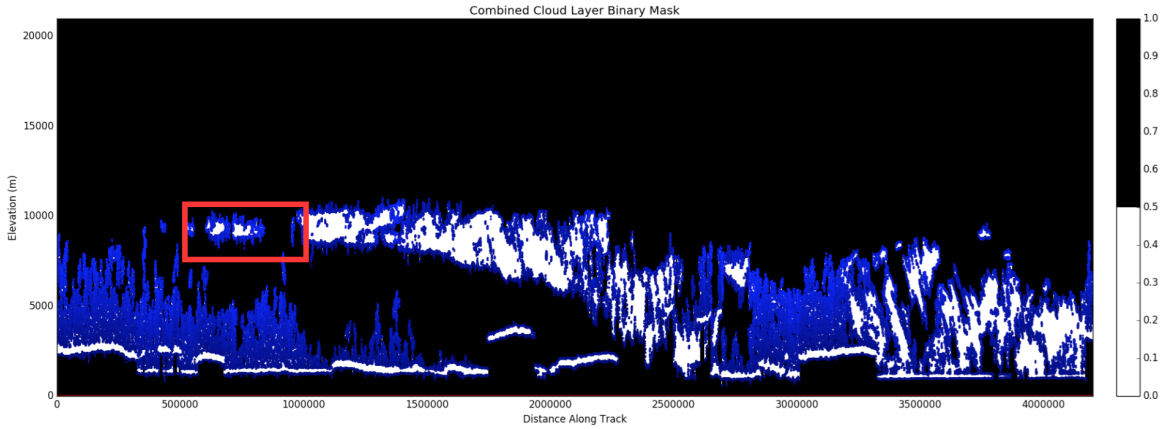
Examples

The following results were produced using parameter set (t8), and are compared with corresponding results from the current method for reference. Note that in all figures, the colored dots indicate the layer tops and bottoms.

Note that in Figures 8-2 and 8-4, when reviewing an entire dataset, differences between the two algorithms are not immediately obvious. However, upon closer inspection, we see that small numbers of bins have been added to the tops of cloud layers in the Figures 8-2 and 8-4, indicating that the old layer detection routine does indeed mis-represent cloud layers by excluding loner bins. This is made apparent in the zoomed in Figures 8-3 and 8-5, where clear additions are visible to the tops of select cloud layers. That is, with the addition of an upward pass of the layer detection routine, we do in fact gain valuable additional information about the location of cloud layers.

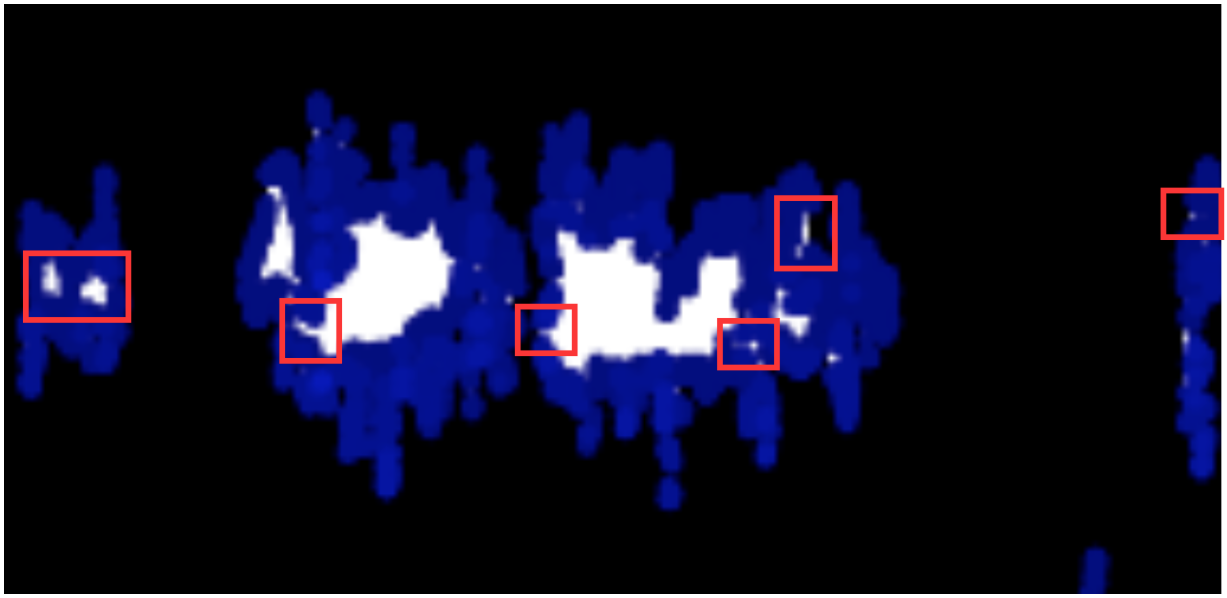


(a) The combined cloud layer binary mask from ATL04_short_night_run.h5 produced with the old (v108.1) layer detection method.

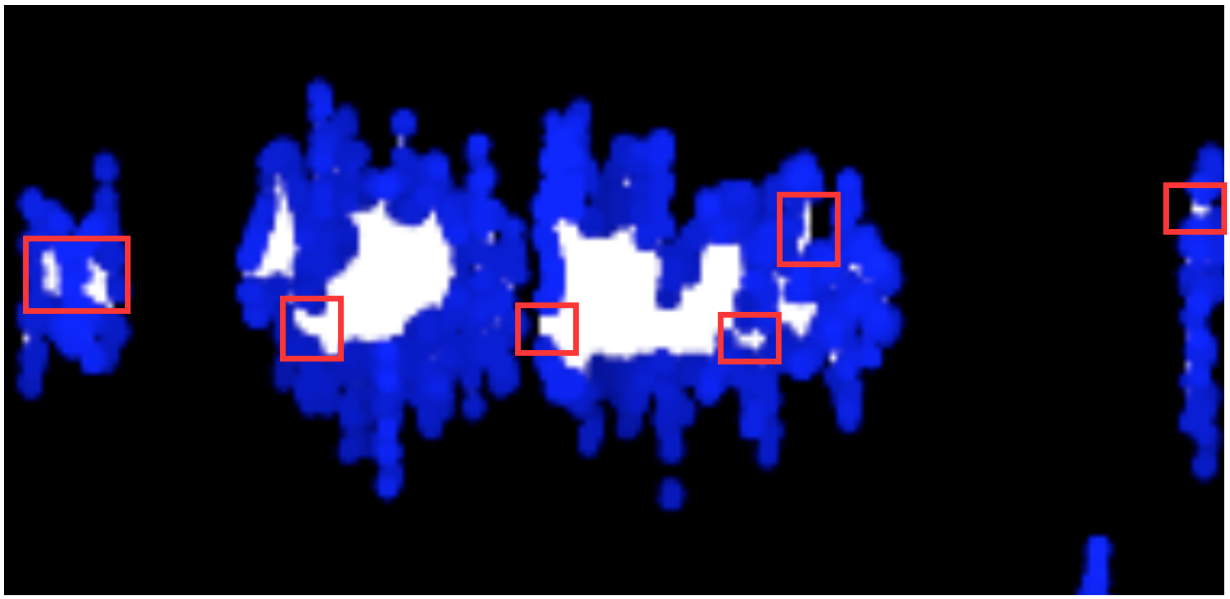


(b) The combined cloud layer binary mask from the same dataset, produced with the new (v109.0) method.

Figure 8-2. Macro Comparison - Red boxes indicate the zoomed regions in Figure 8-3.

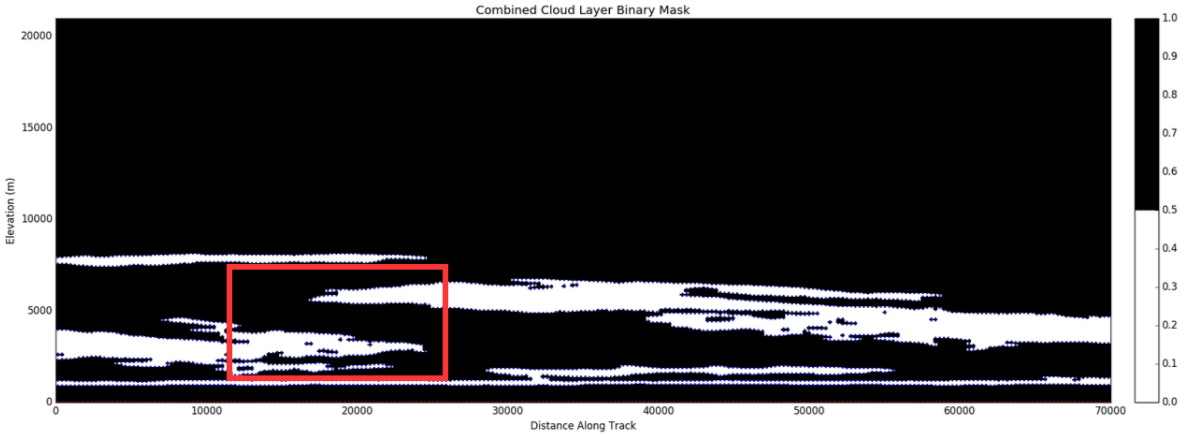


(a) Zoomed in on the combined cloud layer binary mask from ATL04_short_night_run.h5 produced with the old (v108.1) layer detection method.

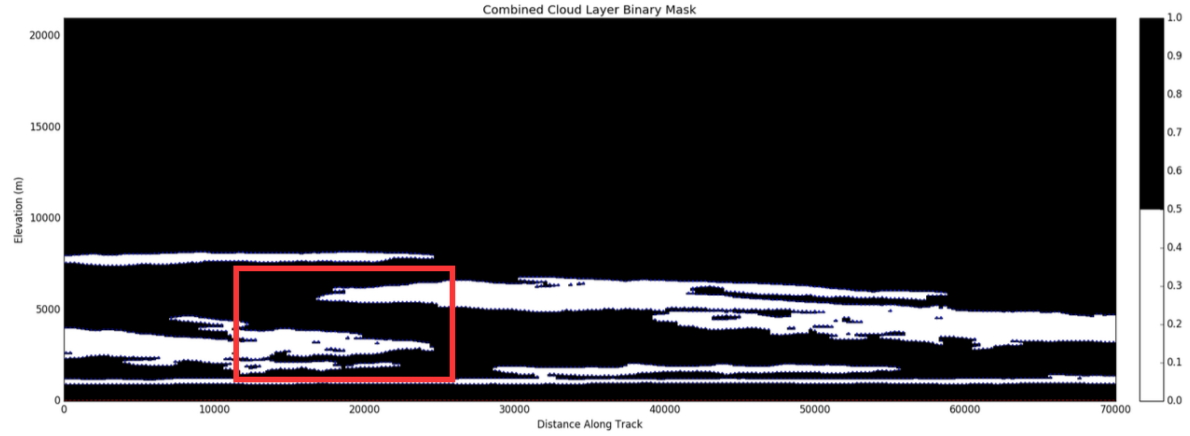


(b) Zoomed in on the combined cloud layer binary mask from the same dataset, produced with the new (v109.0) method.

Figure 8-3. Micro Comparison - Red boxes highlight differences.

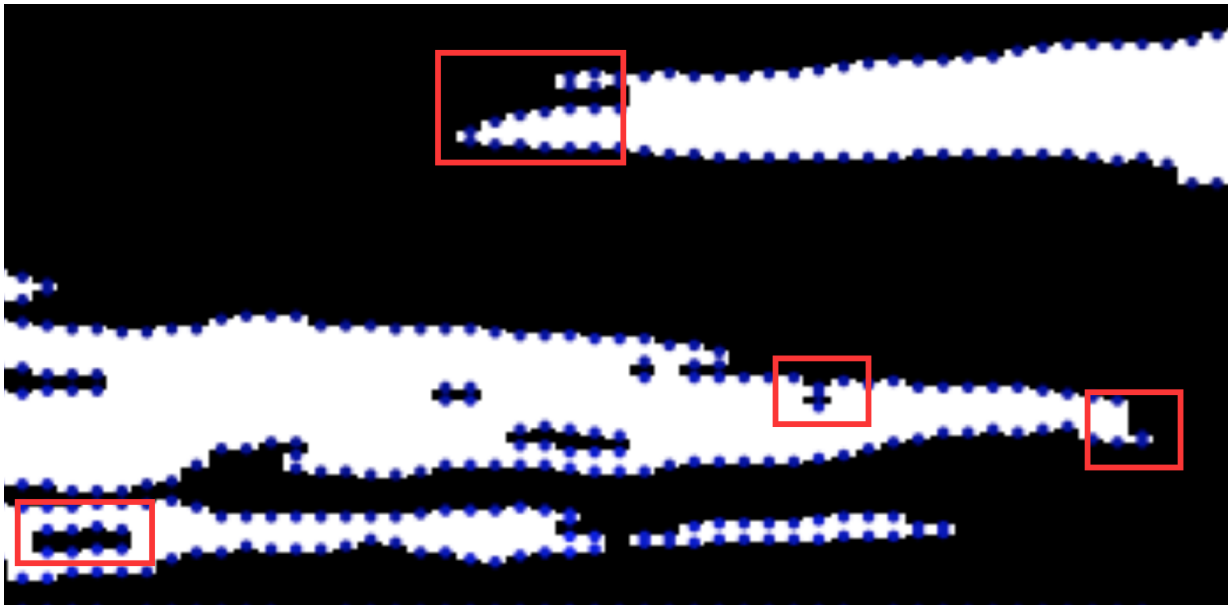


(a) The combined cloud layer binary mask from a small 700x250 sample dataset, produced with the old (v108.1) layer detection method.

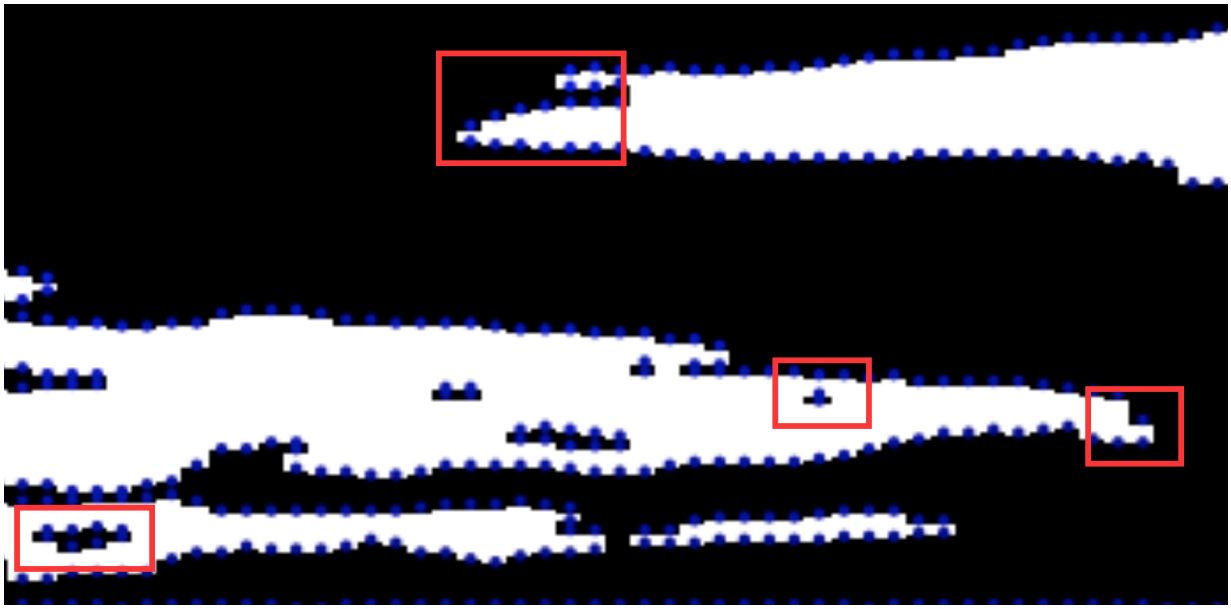


(b) The combined cloud layer binary mask from the same dataset, produced with the new (v109.0) method.

Figure 8-4. Macro Comparison - Red boxes indicate the zoomed regions in Figure 8-5.



(a) Zoomed in on the combined cloud layer binary mask from a small 700x250 sample dataset, produced with the old (v108.1) layer detection method.



(b) Zoomed in on the combined cloud layer binary mask from the same dataset, produced with the new (v109.0) method.

Figure 8-5. Micro Comparison - Red boxes highlight differences.

Side Note on Limiting the Total Number of Cloud Layers. Given that there are a total of 467 valid bins (in the vertical direction) in an NRB object, with each cloud layer occupying a minimum of 6 bins, the theoretical maximum number of cloud layers that the algorithm could find is actually 77. The maximal number of layers that can be output on the ATL09 product is 10.

Python Implementation

```
def compute_cloud_layers_v2(cloud_image):
    """
    Adam Hayes, 14 January 2020

    This function is extremely similar to the function above, but it extends to
      allow different values
    of layer separation and layer thickness to be implemented by using loops on the
       upward and downward passes.
    """



    max_layers = cloud_image.cloud_layer_tops.shape[0]
    (n,m) = cloud_image.combined_decluster_mask.shape

    cloud_layer_mask_down = np.zeros((n,m), dtype=bool)
    cloud_layer_mask_up = np.zeros((n,m), dtype=bool)

    layer_sep = 2
    layer_thick = 3

    for j in range(m):

      #Pass in the upward direction.
      incloud = False

      for i in np.arange(n-1,2,-1):
        if incloud == False: #requirements for given bin to be marked as part of a
      cloud layer

          # Layer Thickness
          cnt=0
          for x in range(layer_thick):
```

```
        if cloud_image.combined_decluster_mask[i-x,j]==False:
          cnt+=1
      if cnt == layer_thick:
        cloud_layer_mask_up[i,j] = True #Set the cloud layer mask to indicate a cloud
 layer in this location
        incloud = True

    elif incloud == True: #requirements for a given bin to NOT be marked as part of
a cloud layer

      # Layer Separation
      cnt=0
      for y in range(layer_sep):
        if cloud_image.combined_decluster_mask[i-y,j]==True:
          cnt+=1
        elif cloud_image.combined_decluster_mask[i-y,j] == False:
          cloud_layer_mask_up[i,j] = True #If the requirements to leave a cloud layer
 are NOT met, set the cloud layer mask to indicate a cloud layer in this
location.
          break
      if cnt == layer_sep:
        incloud = False

# Pass in the downward direction.
incloud = False

for i in np.arange(0,n-3,1):
  if incloud == False: #requirements for given bin to be marked as part of a
cloud layer

    # Layer Thickness
    cnt=0
    for x in range(layer_thick):
      if cloud_image.combined_decluster_mask[i+x,j]==False:
        cnt+=1
    if cnt == layer_thick:
      cloud_layer_mask_up[i,j] = True #Set the cloud layer mask to indicate a cloud
 layer in this location
      incloud = True
```

```
    elif incloud == True: #requirements for a given bin to NOT be marked as part of a cloud layer

      # Layer Separation
      cnt=0
      for y in range(layer_sep):
        if i+y >= 700: break

        if cloud_image.combined_decluster_mask[i+y,j] == True:
          cnt+=1
        elif cloud_image.combined_decluster_mask[i+y,j] == False:
          cloud_layer_mask_down[i,j] = True #If the requirements to leave a cloud layer are NOT met, set the cloud layer mask to indicate a cloud layer in this location.
          break
      if cnt == layer_sep:
        incloud = False

#Take the union (logical or) of the up and down cloud layer masks to create the final cloud mask.
cloud_layer_mask_combined = np.logical_or(cloud_layer_mask_up, cloud_layer_mask_down) # True indicates locations where a cloud layer exists.
cloud_image.cloud_mask = ~cloud_layer_mask_combined #since we want our final cloud mask to indicate False where a cloud layer exists.

#Finally, we use the final cloud mask to calculate cloud tops and bottoms, and limit total number of layers.

for j in range(m):
  k=0 #reset number of layers found in the current column
  incloud = False
  for i in np.arange(n-1,0,-1):
    if k==max_layers:
      break
    if incloud == False: #requirements to mark the top of a cloud layer.
      if cloud_layer_mask_combined[i,j]==True:
        cloud_image.cloud_layer_tops[k,j] = i  #set i as the index of the top of the kth cloud layer in the jth column.
```

```
            incloud = True
        elif incloud == True: #requirements to mark the bottom of a cloud layer.
410         if cloud_layer_mask_combined[i,j]==False:
                cloud_image.cloud_layer_bottoms[k,j]=i+1  #set i+1 as the index of the bottom
    of the kth cloud layer in the jth column.
                incloud = False
                k=k+1
```

Listing 24: Python Code v106.0 (2020-02-06): Call "compute cloud layers" function.

Applications

This new algorithm for derivation of layer boundaries is applied to the examples of GLAS-based simulated ATL04 data, for the test data set used in the sensitivity study (7000+ profiles (7143)) for the following parameter combinations:

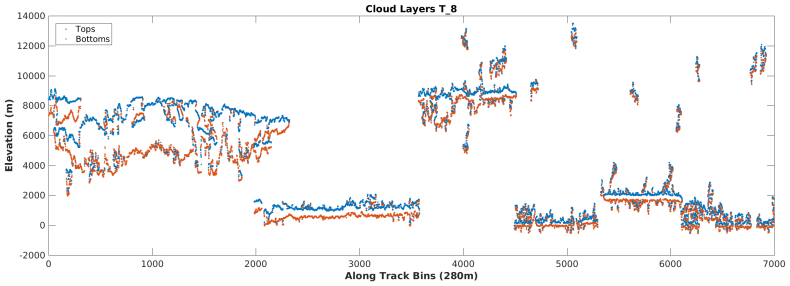
(t8) single-density run deemed best in the 2016 state-of-the-art simulated GLAS-based ICESat-2 type data,

(t54) single-density run, best parameter combination for analysis of 2017-Oct version of GLAS-based simulated ATL04 data

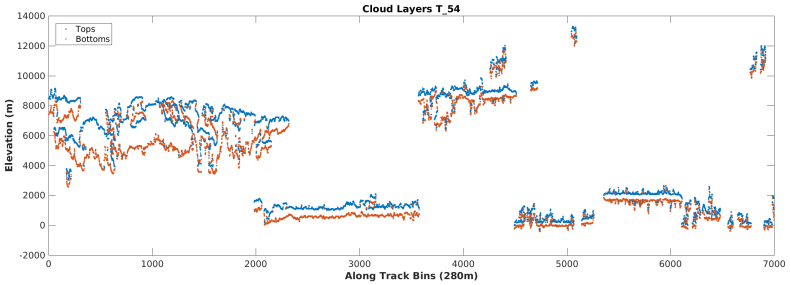
(t56) double-density run, best parameter combination for analysis of 2017-Oct version of GLAS-based simulated ATL04 data

(t64) double-density run, alternative best parameter combination for analysis of 2017-Oct version of GLAS-based simulated ATL04 data (run1 cluster size 200, otherwise same parameters as in $t56$)

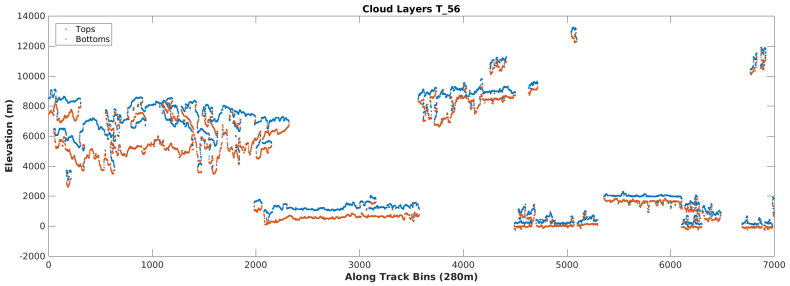
(a)



(b)



(c)



(d)

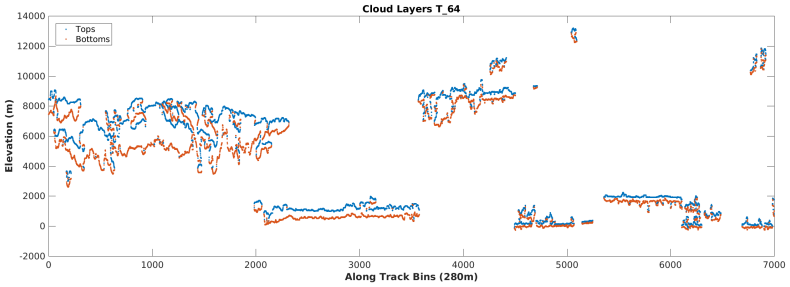


Figure 8-5. Atmospheric layer boundaries. Applied to 7000+ (7143) profile synthetic data set representing different cloud types and night-time/ day-time transition, 2017-Oct version of GLAS-based simulated ATL04 data. Blue - layer tops, red – layer bottoms.
(a) t8, (b) t54, (c) t56, (d) t64. See Tables 5 and 6 for parameters.

Figure 8-5 shows several things:

(1) The new algorithm that absorbs loner bins into the cloud layers yields layer-tops and layer-bottoms that have a natural appearance. This means, cloud layers are connected over large height ranges.

(2) The parameter combination, $t8$, which worked best for the 2016 state-of-the-art simulated GLAS-based ICESat-2 type data, renders ill-defined layer-tops and bottoms especially at day time (right part of the data set). Some false positives appear, especially around the layer boundaries. This indicates that the parameters that determine the threshold function do not match the characteristics of the data any more. The change in NRB value determination (ATL04) requires a new set of parameters.

(3) The parameter combination, $t54$, is the best result for a single-density run, as determined in the sensitivity study. The layer boundaries are much better defined than in in the $t8$ run. However, the layer boundaries are still somewhat ragged for night-time data and sub-optimally defined for day-time data. As the sensitivity study shows, it is not possible to retain tenuous clouds, while suppressing false positives, using a single-density run.

(4) This necessitates using a double-density run, which allows to identify optically thick layers in the first run (using a smaller kernel and a very strict threshold function) and, in the second density run, identify the tenuous clouds, atmospheric layers and most clouds during day-time conditions (using a larger kernel and a less strict threshold function). Notice that tenuous clouds (on the left) are now connected in the vertical direction, except for likely natural gaps, rain (?) falls out of the layer at a possible inversion (aerosols with clouds at the inversion height), but no false positives remain and the layer tops during day-time are smooth. Single clouds are retained during day-time conditions. $t56$ is the parameter combination used in most experiments in October/November 2017 (and deemed best for current state-of-the-art data characteristics in ATL04).

(5) Varying parameters around those of $t56$ and trouble-shooting remaining differences between CU code and SIPS code, we noticed that a smaller cluster size in run1 (200 rather than 300 pixels) retains all good characteristics of $t56$ and appears to slightly improve them. The cluster size of 200 also renders the algorithm more robust (in the sense that all small speckles are already filtered out and first-order cloud layers are more continuous). This is ($t64$). Parameters are otherwise the same as in ($t56$). Note this may be good to know in testing, going forward, as the CU declustering step

and the SIPS declustering step employ similar, but not the exact same function.

3.7 Step 7: Layer Density, Density Sum per Vertical Profile and Other Derived Parameters

Layer density (Layer_Dens in Table 1) is calculated as the sum of density values per column, in each of the maximally 6 layers. If two density runs are used, then density1 will be available for all layers (also those identified in density run 2) and density2 will be available only for the layers identified in density run 2.

Density per vertical profile is calculated as the sum of the density values in all identified layers for each along-track location (sum of density values in cloud regions). This value gives physical meaning to the density approach. (Column_Dens1(3), Column_Dens2(3)).

The sum of density values per vertical profile is related to the optical depth used in atmospheric sciences. However, it is not the same as optical depth, because the recorded data only extend to about 14 km above the Earth's surface, while derivation of optical depth requires recording the entire atmospheric column in a given location. A function relating column sum of density to column optical depth will be determined in work in progress of U. Herzfeld, S. Palm and Y. Yang.

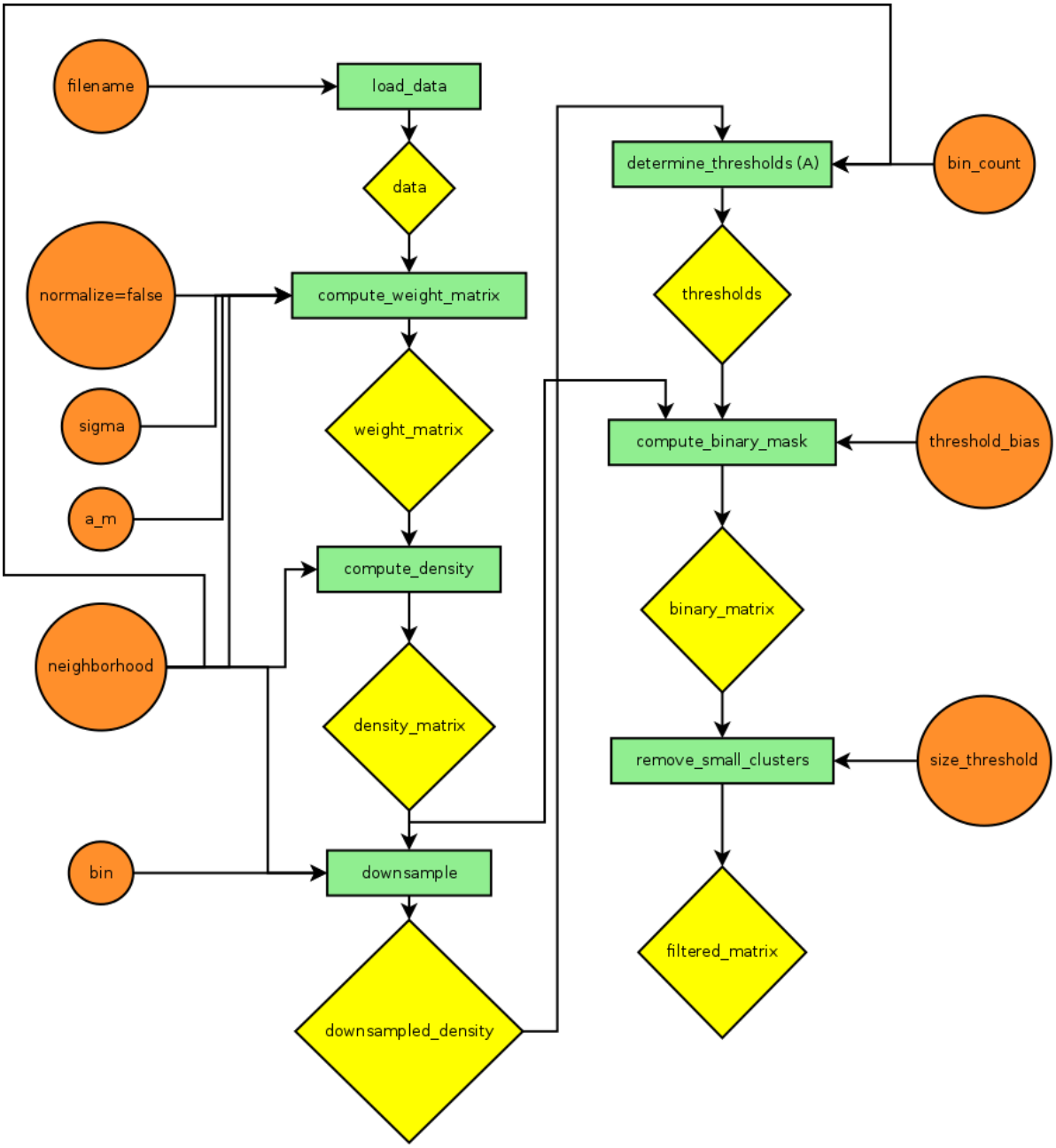
Other parameters. Discrimination of a blowing snow layer from surface returns in areas without blowing snow, based on density, appears feasible, but test data are needed to test and confirm this and specify thresholds.

Classification of types flayers into optically thin and optically thick clouds, aerosols and blowing snow appears equally possible and is a matter of ongoing research. Space for variables to output these results is reserved to simplify software development (see Table 1).

During the mission, the DDA will be applied to atmosphere data in near-real time, as the data come in. If the DDA is additionally applied to calibrated data, then comparisons of layer density across the life of the mission will become possible.

3.8 Flow of Algorithm Steps

For method version A (2012 MABEL Data Analysis), the flow of algorithm steps is illustrated in the algorithm diagram (A) and in the listing "Density-dimension algorithm (version A, code version v6)".



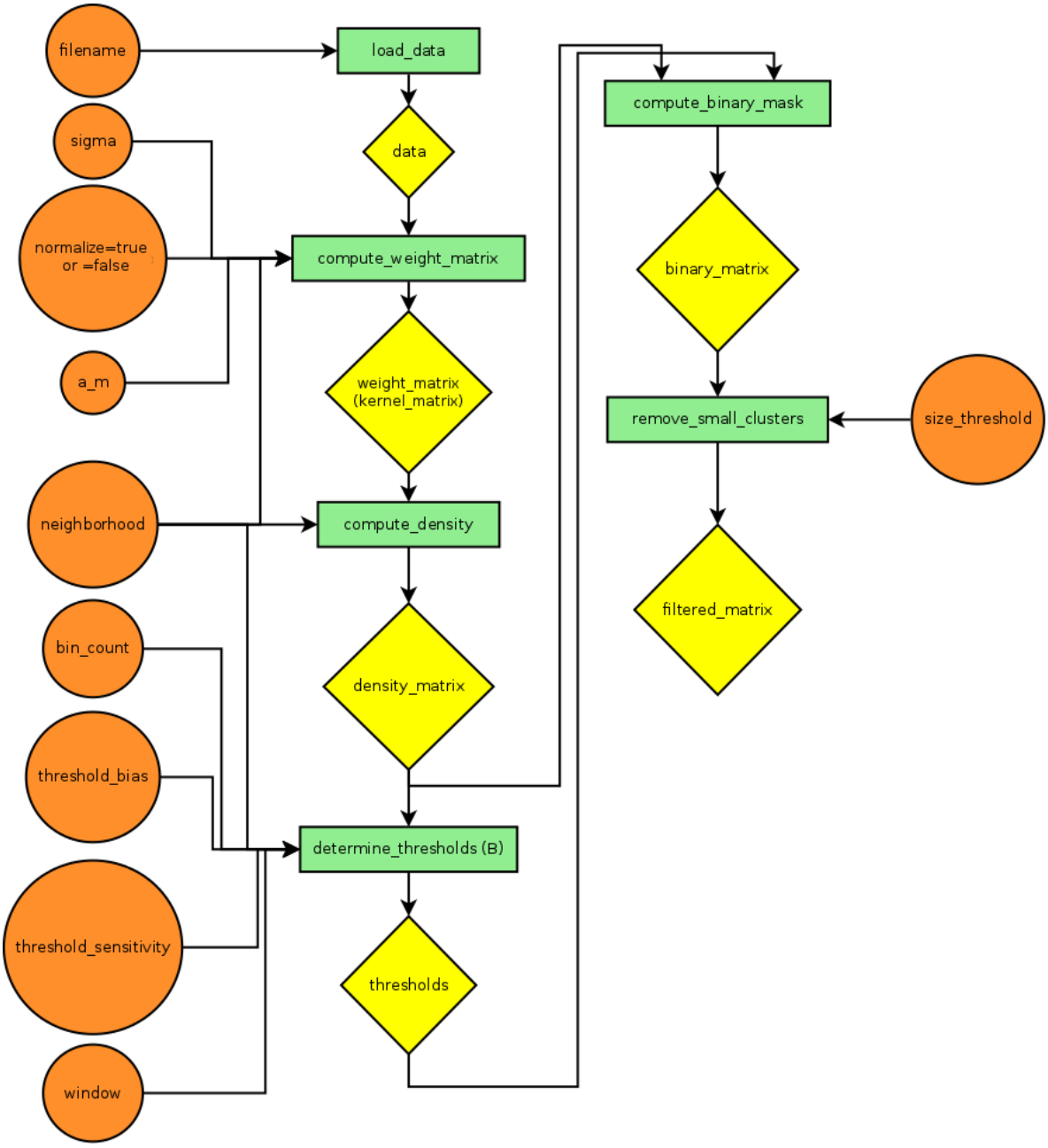
Algorithm Diagram, Method Version A

```
data = load_data(filename)
masks = []
for i in 1 to num_passes:
    neighborhood = neighborhood_list[i]
    a_m = a_m_list[i]
    sigma = sigma_list[i]
    weight_matrix = compute_weight_matrix(neighborhood, a_m, sigma, normalize=False)
    density_matrix = compute_density(data, weight_matrix)
    downsampled_density = downsample(density_matrix, neighborhood, bin)
    thresholds = determine_thresholds(downsampled_density, bin_count, neighborhood)
    initial mask = compute_binary_mask(density_matrix, threshold_bias, thresholds)
    masks.append(remove_small_clusters(initial_mask, size_threshold))
combined_mask = logical_or(masks)
```

Listing 25: Pseudo-Code: Density-dimension algorithm version A v6

This describes the option of using density once. For using density twice (see section (3.10)), every step after "load data" is applied again in a second run after replacing the data in the masked out regions with zeros. Note that running density twice is not needed for 2012 MABEL data analysis, the same global parameters were used for all examples and the auto-adaptive thresholding provided good results, as seen in Figures 1-9 and 11-12. The number of density runs is passed to the code as a parameter *num_passes*, the value of parameter *num_passes* is typically equal to 1 or 2 (see Listing 27).

For method version B (2013 M-ATLAS Data Analysis), the flow of algorithm steps is illustrated in the algorithm diagram (B) and in the listing "Density-dimension algorithm (version B, code version v103)".



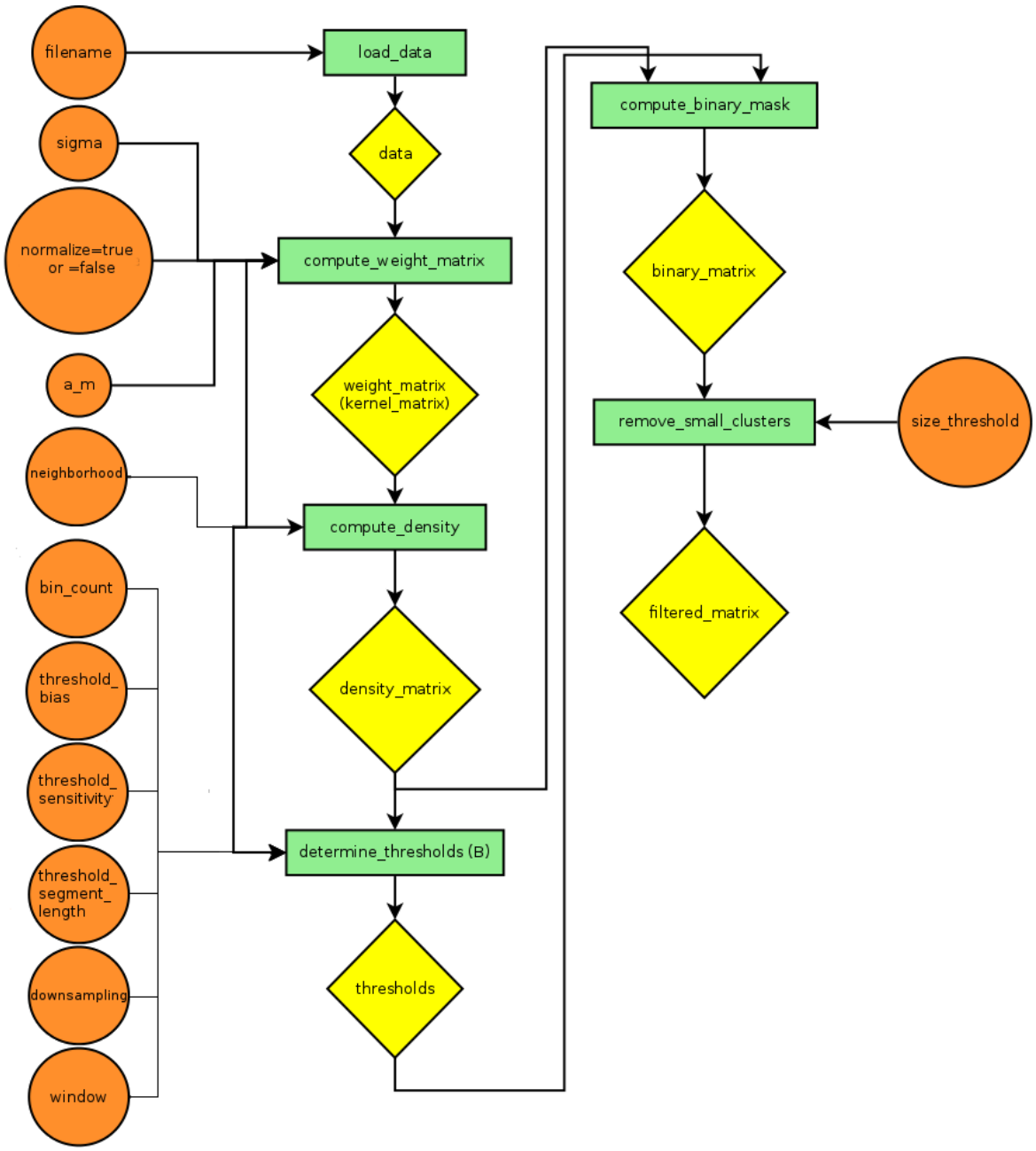
Algorithm Diagram, Method Version B

```
data = load_data(filename)
masks = []
for i in 1 to num_passes:
    neighborhood = neighborhood_list[i]
    a_m = a_m_list[i]
    sigma = sigma_list[i]
    weight_matrix = compute_weight_matrix(neighborhood=None, a_m, sigma, normalize=True)
    density_matrix = compute_density(data, weight_matrix, neighborhood)
    thresholds = determine_thresholds(density_matrix, threshold_bias, threshold_sensitivity,
    window)
    initial mask = compute_binary_mask(density_matrix, threshold_bias, thresholds)
    masks.append(remove_small_clusters(initial_mask, size_threshold))
combined_mask = logical_or(masks)
```

Listing 26: Pseudo-Code: Density-dimension algorithm version B v103

This describes the option of using density once, for using density twice (see section (3.10)), every step after "load data" is applied again in a second run after replacing the data in the masked out regions with zeros. Note that running density twice is applied in 2013 M-ATLAS data analysis, the same global parameters were used for all examples and the auto-adaptive thresholding provided good results, as seen in Figure 19. The number of density runs is passed to the code as a parameter *num_passes*, the value of parameter *num_passes* is typically equal to 1 or 2 (see Listing 28).

For the integrated method version A/B (v105), the flow of algorithm steps is illustrated in the algorithm diagram (A/B). See also Table 2d.



Algorithm Diagram, Method A/B Synthesis

3.9 Adjustable Parameters

The algorithm includes adjustable parameters, as given in the following table. The effect of some of these parameters on the classification results is examined in sensitivity studies in section (7), (9) and (10).

Adjustable Parameters in the Density-Dimension Algorithm (Method Version A)

Variable	Value	Explanation	Section	Notes
num_passes	1 or 2	number of density runs	3.10	
r=r1, neighborhood	3	radius 1, search radius for density1, results in neighborhood boxes of size 2r+1	2.2:M.2	larger for 2013 r2 data
r=r2 , neighborhood	5	radius 2, search radius for density2, results in neighborhood boxes of size 2r+1	2.2:M.2	only used if density is run twice, larger for 2013 r2 data
bin, downsampling	5	downsampling window size for noise area determination	3.3.2	don't change this
bin_count	50	number of bins that form the noise area set (50 bins of size 5 by 5), 50 lowest values used for threshold determination for noise regions	3.3.2	
threshold_segment_length	1	along-track size of the segment for which a noise-threshold is determined (1 column of the downsampled matrix, equal to 5 columns of the original matrix, equal to 1,600 meters)	3.3.2	
threshold_bias	70	threshold-bias	3.3.3	
size_threshold	300	the size of small clusters is 300 points or less	3.4	
σ_n, s, sigma	140	standard-deviation (a parameter that affects the density kernel)	2.2:M.2	see sensitivity study
squish_matrix	diag(0.3, 1)	anisotropy (a parameter that affects the density kernel, see sensitivity study)	2.2:M.2	see sensitivity study

Table 2a

Adjustable Parameters in the Density-Dimension Algorithm (Method Version B)

Variable	Value(s)	Explanation	Section	Notes
num_passes	1 or 2	number of density runs (first value for run 1, second value for run 2)	3.10	
a_m, a_m	10,20	anisotropy factor	2.2:M.2, 3.2	2 values for 2 density runs
σ_n, s, sigma	3,6	standard-deviation (a parameter that affects the density kernel)	2.2:M.1, 3.3.3	see sensitivity studies (section 7,9); 2 values for 2 density runs
r_1	6	radius 1, search radius for density1, results in neighborhood of size $2r_2 + 1 = 13$	3.3.4-3.3.5	value calculated from σ_n and a_m
r_2	large	radius 2, search radius for density2, results in neighborhood of size $2r_2 + 1$	3.3.4-3.3.5	value calculated from σ_n and a_m
window	20, 20	along-track size of the segment for which a noise-threshold is determined (equal to 5600 meters)	3.3.3	20 used in both runs
threshold_bias	60,0	threshold-bias determines overall density sensitivity level	3.3.3	2 values for 2 density runs
threshold_sensitivity	1,1	adjusts sensitivity to local variation	3.3.3	2 values for 2 density runs
size_threshold	600, 600	the size of small clusters is 600 points or less	3.4	600 used in both runs
normalization T/F	true	normalization option in weight matrix for density function is applied	2.2:M.3	

Table 2b

Note: Values given here are used for the example given in section 7.2 that demonstrates that the algorithm will auto-adapt to different conditions (so-called triple data set). Other values used in the sensitivity studies.

```
python atmos_algo.py --help

Usage: atmos_algo.py [options] datafile

Options:
  -h, --help            show this help message and exit
  -o DIR, --output-dir=DIR
                        Output directory
  -n DIR, --name=DIR    Series/experiment identifier
  -v, --verbose         Debug logging level mode
  -e END, --end=END     Stop after END points
  -S END, --end-step=END
                        Stop after END steps
  -g, --visualize       Show interactive visualizations
  -G STEPS, --visualize-steps=STEPS
                        STEPS to visualize only
  -p, --save-plots      Save plots visualizations
  -l FILE, --log=FILE   Write log to FILE
  -a ANISO_FACTOR, --aniso-factor=ANISO_FACTOR
                        Anisotropy factor. Multiple values will be applied to hierarchy
                        of densities.
  -w NEIGHBORHOOD, --neighborhood=NEIGHBORHOOD
                        Prescribe the neighborhood of the kernel in pixels.
                        Default is NONE, and calculation from SIGMA, CUTOFF, ANISO_FACTOR.
                        Multiple values will be applied to hierarchy of densities.
  -d DOWNSAMPLE, --downsample=DOWNSAMPLE
                        Downsample factor. Default is 5.
  -s SIGMA, --sigma=SIGMA
                        Standard deviation of Gaussian kernel (meters).
                        Multiple values will be applied to hierarchy of densities.
  -c CUTOFF, --cutoff=CUTOFF
                        Cut off Gaussian kernel after number of CUTOFF stddevs. Default is 2.
                        Multiple values will be applied to hierarchy of densities.
  -t THRESHOLD_FACTOR, --threshold_factor=THRESHOLD_FACTOR
                        Adaptive factor for threshold local quantile.  Default is 1. Multiple
                        values will be applied to hierarchy of densities.
  -T THRESHOLD_BIAS, --threshold-bias=THRESHOLD_BIAS
                        Base threshold for threshold. Multiple values will be applied to
                        hierarchy of densities.
  -m MIN_CLUSTER, --min-cluster=MIN_CLUSTER
                        Minimum cloud size in pixels. Default is 300.
  --no-r2               Remove r2 correction
  --no-bg               Remove bg correction
  --no-bgr2             Remove both r2 and bg correction
  --correct-power       Apply r^power correction (r^0.6)
```

Table 2c: Parameters for algorithm Method B (code version v103)

Adjustable Parameters in the Density-Dimension Algorithm (Method Version A/B, v106)

Variable	Value(s)	Explanation	Section	Notes
num_passes	1 or 2	number of density runs (first value for run 1, second value for run 2)	3.10	
a_m, a_m	10,20	anisotropy factor (in meters)	2.2:M.2, 3.2	2 values for 2 density runs
σ, s, sigma, $\sigma = \sigma_{bin}$, $\sigma_m = y_{res} \cdot \sigma_{bin}$	3,6	standard-deviation (a parameter that affects the density kernel). Given in pixels in y-direction	2.2:M.1, 3.3.3	see sensitivity studies (7,9); 2 values for 2 density runs
cutoff	1,1	number of std-deviations used after which kernel size is cut off	2.2:M1, 3.3.3	see sensitivity studies (7,9); 2 values for 2 density runs
$r_{1,x}$	3	kernel size in x-direction for run 1, results in neighborhood of size $m_1 = 2r_{1,x} + 1 = 7$	3.3.4-3.3.5	calculated from σ_n, a_m and cutoff
$r_{1,y}$	3	kernel size in y-direction for run 1, results in neighborhood of size $n_1 = 2r_{1,y} + 1 = 7$	3.3.4-3.3.5	calculated from σ_n and cutoff
$r_{2,x}$	12	kernel size in x-direction for run 2, results in neighborhood of size $m_2 = 2r_{2,x} + 1 = 25$	3.3.4-3.3.5	calculated from σ_n, a_m and cutoff
$r_{2,y}$	6	kernel size in y-direction for run 2, results in neighborhood of size $n_1 = 2r_{2,y} + 1 = 13$	3.3.4-3.3.5	calculated from σ_n, and cutoff
downsampling, d	1,1	downsampling window size for noise area determination	3.3.3-3.3.5	default 5
threshold_bias, T	60E+13,0	threshold-bias determines overall density sensitivity level	3.3.3	2 values for 2 density runs
threshold_sensitivity, t	1,1	adjusts sensitivity to local variation	3.3.3	2 values for 2 density runs
size_threshold	600, 600	the size of small clusters is 600 points or less	3.4	600 used in both runs
threshold_segment_length, L	2,2	along-track size of segment for which a noise-threshold is determined	3.3.3	default 2
quantile, q	0.75, 0.75	quantile of maximum densities for threshold determination	3.3.6	default 0.5
normalization T/F	true	normalization option in weight matrix for density function is applied	2.2:M.3	
correct-power	p	optional power correction for aircraft data	7.1	range multiplied with $r^{0.6}$ for $r > 1$, effectively $r^{1.4}$ instead of r^2

Table 2d

Note: Values given are examples, see sections 7.2 and 9. Other values used in the sensitivity studies.

3.10 Running Density Twice

The algorithm option "running density twice" is the used for analysis of ICESat-2 ATLAS (post-launch) data generating product ATL09 (starting with the first data release and through release 3, ASAS code release v5.3, March 2020).

3.10.1 Concept

Running density twice serves as a way to distinguish mathematically between optically thin and optically thick cloud layers. Different settings of the neighborhood are applied. For example, a smaller neighborhood run first will detect optically dense, spatially thin cloud layers. In the next step, the area of dense clouds is replaced by points with the spatial characteristics of the noise area, or noise bin. In some versions of the code (v6, v103), the area of dense clouds is simply masked out. In the next step, density is calculated, but for a larger neighborhood. This allows detection of the optically thin, spatially thick clouds. Next, both the areas of optically thick and optically thin clouds are combined (joint set simply). — This concept was applied to some of the simulated ICESat-2 data, based on GLAS data. Figure 9 shows the result of cloud detection running density once. Figure 10 shows the result of cloud detection running density twice, and since thin clouds and thick clouds are distinguished, the latter is actually a cloud classification.

The analysis of 2013 $range^2$ and background-corrected M-ATLAS data also applies "running density twice", see section (6).

Computational note. Folding over of kernel. The simplification in the code/pseudo-code of masking out the cloud area of the first density run and replacing it with zeros (rather than noise values) may lead to edge effects around the areas of the clouds determined in the first run. This may be solved by a method such as folding the kernel over. Implementation and testing of this smaller improvement is TBD.
Update for geomath code v114 and ASAS release v5.3 code, Jan 2020. The idea of "folding kernel over" has been abandoned. Folding of kernel was implemented in earlier code versions, but is not used any more.

Note (2018-Dec-19). The option of "running density twice", developed in 2013, was not deemed necessary for optimized data analysis until 2017/2018 and is now adopted as a component of the

operational code for (post-launch) ICESat-2 data. This is based on a sensitivity study of 2017-Oct Version of GLAS-based simulated ATL04 data (section 10) and a sensitivity study of the first post-launch data collected with the ICESat-2 ATLAS instrument (section 15). As a side note, the re-institution of "running density twice" in the operational code for post-launch data shows that it is a good idea to keep old algorithm components in the ATBD.

(1)

(2)

(3)

(4)

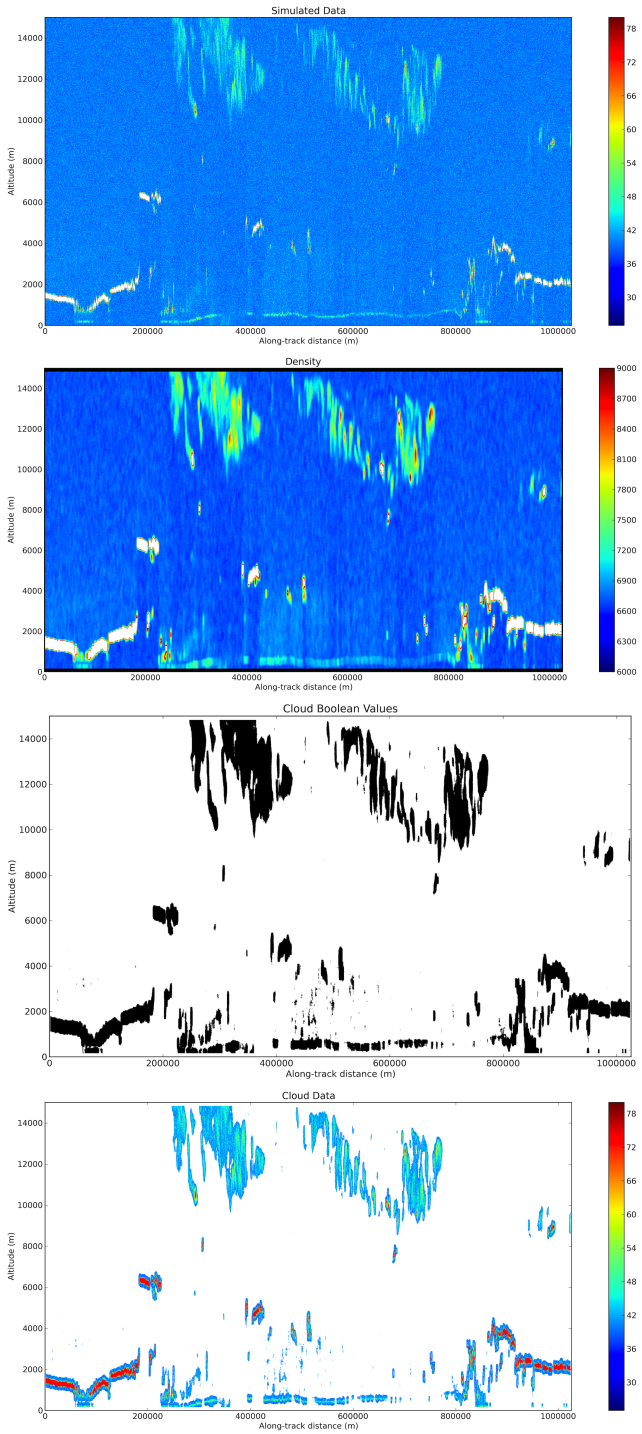


Figure 9. Cloud detection through application of the density-dimension algorithm with one density run to a high-noise data set (Data: Simulated ATLAS data based on GLAS 532 nm data, high noise (0.5 Mhz)).

(1) Simulated ATLAS data based on GLAS 532 nm data

(2) Density

(3) Binary map

(4) Cloud regions - data

(5)

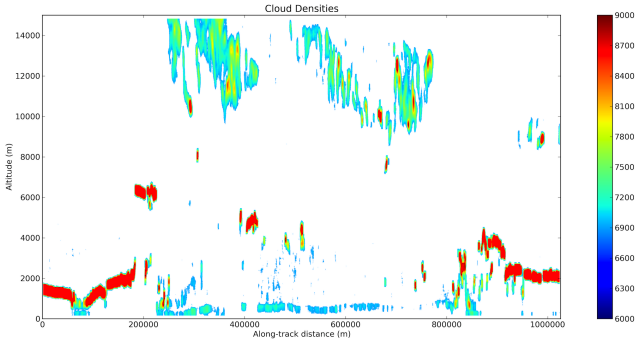


Figure 9 ctd. Cloud detection through application of the density-dimension algorithm with one density run to a high-noise data set (Data: Simulated ATLAS data based on GLAS 532 nm data, high noise (0.5 Mhz)).

(5) Cloud regions — Density (recalculated)

(1)

(2)

(3)

(4)

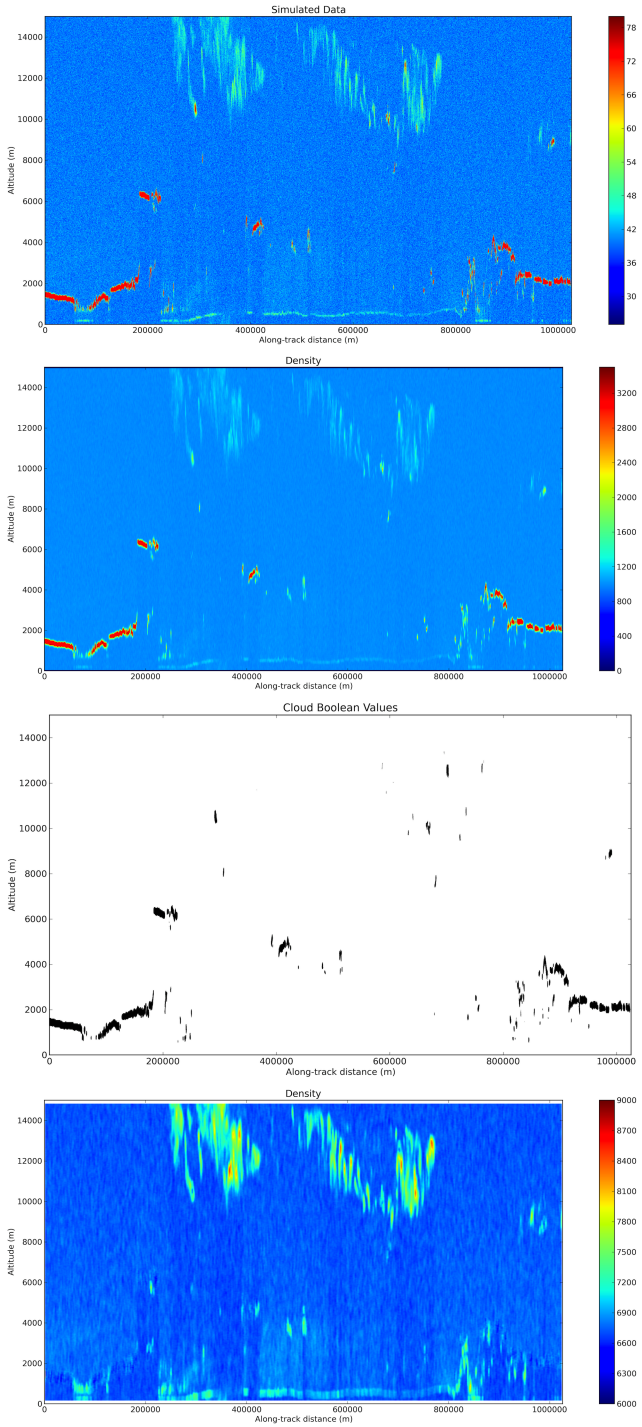


Figure 10. Cloud Classification through application of the density-dimension algorithm with two density runs to a high-noise data set (Data: Simulated ATLAS data based on GLAS 532 nm data, high noise (0.5 Mhz)).

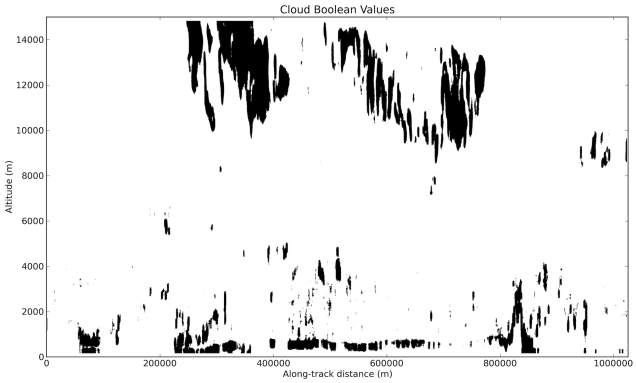
(1) Simulated ATLAS data based on GLAS 532 nm data

(2) Density (small search neighborhood)

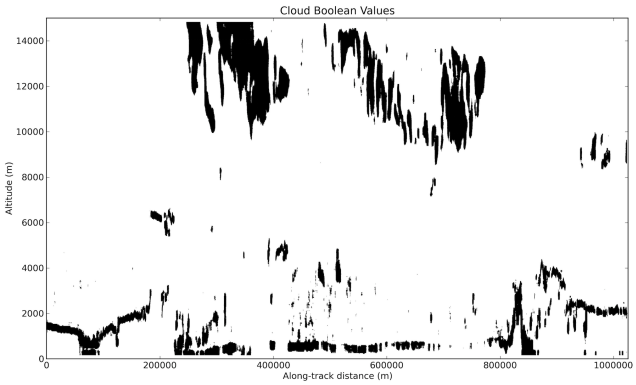
(3) Binary map (dense clouds)

(4) Density (large search neighborhood; no dense, stratified clouds)

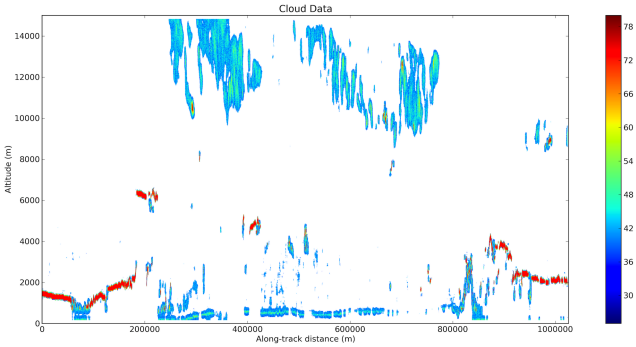
(5)



(6)



(7)



(8)

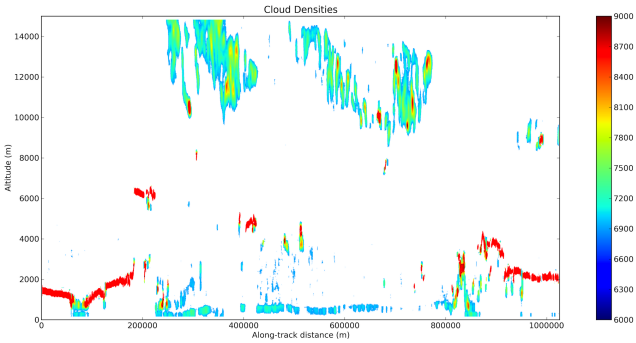


Figure 10, ctd. Cloud Classification through application of the density-dimension algorithm with two density runs to a high-noise data set (Data: Simulated ATLAS data based on GLAS 532 nm data, high noise (0.5 Mhz)).

(5) Binary map (less dense clouds)

(6) Binary map (all clouds)

(7) Cloud regions — Data (all clouds)

(8) Cloud regions — Density (recalculated); red: optically dense clouds; green-blue: optical thin clouds (ground at bottom)

3.10.2 Code for Running Density Twice

The first run is aimed at detection of optically dense, spatially narrow clouds, whereas the objective of the second run is to detect tenuous cloud layers and aerosols.

For a double-density calculation, one needs to remove the high density photons (corresponding to dense clouds) identified in the first cloud determination (density calculation, threshold function application and small-cluster removal) before the second cloud determination. The second run will typically use a larger kernel (a kernel with larger dimensions) and possibly a higher anisotropy value.

```
        ### Delete data
        if len(algo.steps) == options.end_step: break
335     algo.start_step(Step(name='Delete data',
                vis_funcs=[]))
        algo.steps[-1].set_visualize(len(algo.steps) in visualize_steps)
        combined_mask = reduce(logical_and, level_masks).astype(bool)
        histo[logical_not(combined_mask)] = 0
340     globals().update(locals())
        algo.steps[-1].done()
```

Listing 27: Python Code v106.0 (2016-08-17): Remove high density photons from first density calculation.

After the second density run finds photons identified as clouds, the results must be combined with those from the first run.

```
    ### Combine layers before closing
345 if len(algo.steps) == options.end_step: break
    algo.start_step(Step(name='Combine masks pre-closing',
            vis_funcs=[plot_boundary_masked_counts_final,
plot_boundary_masked_density_final]))
    algo.steps[-1].set_visualize(len(algo.steps) in visualize_steps)
    raw_histo.mask = combined_mask
350 globals().update(locals())
    algo.steps[-1].done()
```

Listing 28: Python Code v106.0 (2016-08-17): Combine results from the two density runs.

3.10.3 Description of Mask Handler

Following is a description of the mask handling for the option of "running density twice", as implemented in Geomath code v114 (Jan 2020). The mask handling is introduced in section (3.0) in general. Note that outputs of masks are plotted and thus results of mask handling are found in every figure of every sensitivity study. The log file given in section (3.3.6) provides an additional way to familiarize oneself with the mask handling, as it gives and example of running density twice and creating the masks in proper order (log file of an earlier code version).

(1) **valid_data_mask** — *Section 3.1.2: Load Data.*

At the end of this step, we have valid_data_mask. Density calculation in run 1 uses this mask.

(2) **density_mask_1**

- for input, use valid_data_mask.

- Calculate density, as described in section (3.2.1) and (3.2.2).

- Apply Thresholding, using method A/B for threshold determination in section (3.3). Output *density_mask_1.*

- Downsampling (section 3.3.1) is not affected by "invalids", simply use 5 profiles and all the (valid) values therein, and form a quantile. Because the quantile is defined for a set, it does not matter how large the set is (or, specifically, whether it has missing values). In other words, no special handling of "invalids" is necessary. The mask handler does not need to be mentioned here specifically, because no special coding is required.

- section 3.3.3 describes forming the first mask, **density_mask_1**.

(3) **density_filtered_declustered_mask_1 = final_mask_1**

The term "filtered" comes from the thresholding step. Declustering from the small-cluster-removal step, section (3.4).

The result is shown in (3.5). One outputs final_mask_1 .

(4) **Density run 2:** Use **valid_data_mask = final_mask_1**.

Then start the second density run and complete all its steps. Mask-specifically:

(5) **density_mask_2**

- Calculate density, as described in section (3.2.1) and (3.2.2).

- Apply Thresholding, using method A/B for threshold determination (p.78); in section (3.3). Output *density_mask_2*.

- Downsampling (section 3.3.1) is not affected by "invalids", simply use 5 profiles and all the (valid) values therein, and form a quantile. Because the quantile is defined for a set, it does not matter how large the set is (or, specifically, whether it has missing values). In other words, no special handling of "invalids" is necessary. The mask handler does not need to be mentioned here specifically, because no special coding is required.

(6) **density_filtered_declustered_mask_2 = final_mask_2**

- Perform declustering for density_mask_2, making sure to use the algorithm-specific parameter for minimum cluster size for the second density run.

- Output *density_filtered_declustered_mask_2 = final_mask_2*

(7) **combined_mask**

- form the joint set of *final_mask_1* and *final_mask_2*

(i.e. a point is in *combined_mask* if it is in *final_mask_1* or in *final_mask_2*. Make sure to do this properly, if you are using 0s instead of 1s for inside/outside masks. geomath code uses 1s where ASAS uses 0s. The ASAS way is more intuitive).

Note that the figure order in the sensitivity studies also tells you the order in which the masks are created.

4 Application to 2012 MABEL Data

In this section we give examples of the application of algo version v4 to simulated ICESat-2 data based on MABEL data.

Example in Figure 11 demonstrates the method. Comparison with CPL data shows that we got all the clouds right (see, section *(4.5) Validation*). Data are day-time data (day-time data constitute the more difficult case compared to night-time data, because of the presence of ambient light, resulting in high noise.

The example in Figure 12 includes night-time data and day-time data (and shows a much larger section of the MABEL-data track). The "hump" near the rhs of the plot is a crossing of Greenland. The power of the algorithm really shows where the reflectance-based noise over Greenland increases, but clouds can still be detected. In more technical words, the capability of the density-dimension algorithm allows automated adjustment of the noise threshold to changing environmental conditions, including increased reflectance over the Greenland inland ice and change from night-time to day-time observations. (This is not possible with an a-priori noise-bin algorithm, even using the density-part of the algorithm).

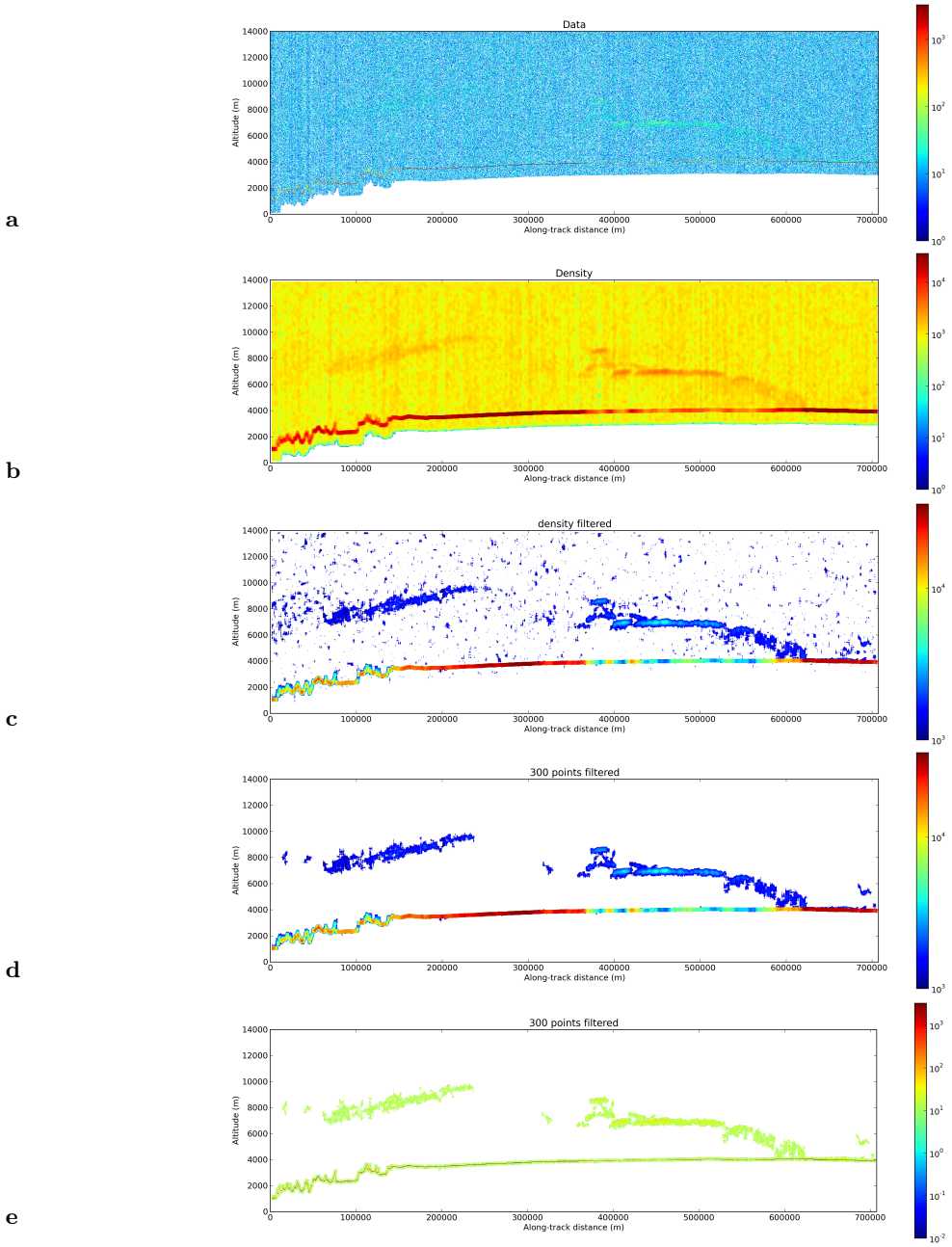


Figure 11. Analysis of day-time data (data set for 12April12.01, simulation set 01, based on MABEL atmosphere data collected 12April 2012). Results of density-dimension algorithm version v4.

(a) Data (data.png)

(b) Density (dens5.png), uses $r_1 = 5$

(c) "Density filtered" - Density-dimension algorithm applied (bin5.png)

(d) "300 points filtered" - Cloud areas with density; after application of small-cluster removal (density_pts300.png)

(e) "300 points filtered" - Data in the same region that is seen in (d) (data_pts300.png)

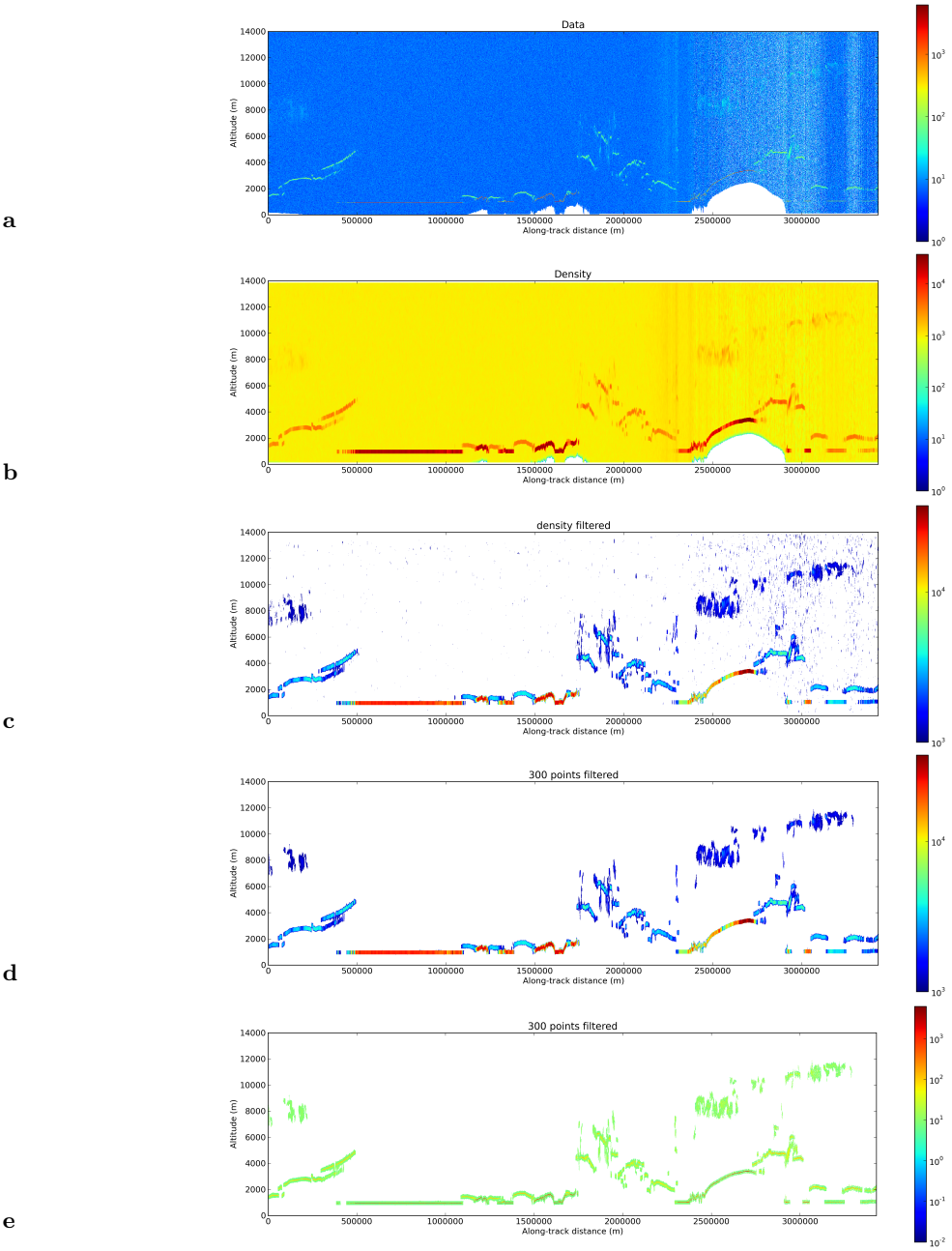


Figure 12. Analysis of night-time data and day-time data (data set for 02Apr12.03, simulation set 03, based on MABEL atmosphere data collected 02April 2012). Results of density-dimension algorithm version v4. The hump on the right is a crossing of Greenland.

(a) Data (data.png)

(b) Density (dens5.png)

(c) "Density filtered" - Density-dimension algorithm applied (bin5.png)

(d) "300 points filtered" - Cloud areas with density; after application of small-cluster removal (density_pts300.png)

(e) "300 points filtered" - Data in the same region that is seen in (d) (data_pts300.png)

5 Validation

In this section, we include some visual quality checks of the algorithm performance. Figure (4.13) highlights an area of optically thin clouds, that can be detected in ATLAS night-time data using the density-dimension algorithm.

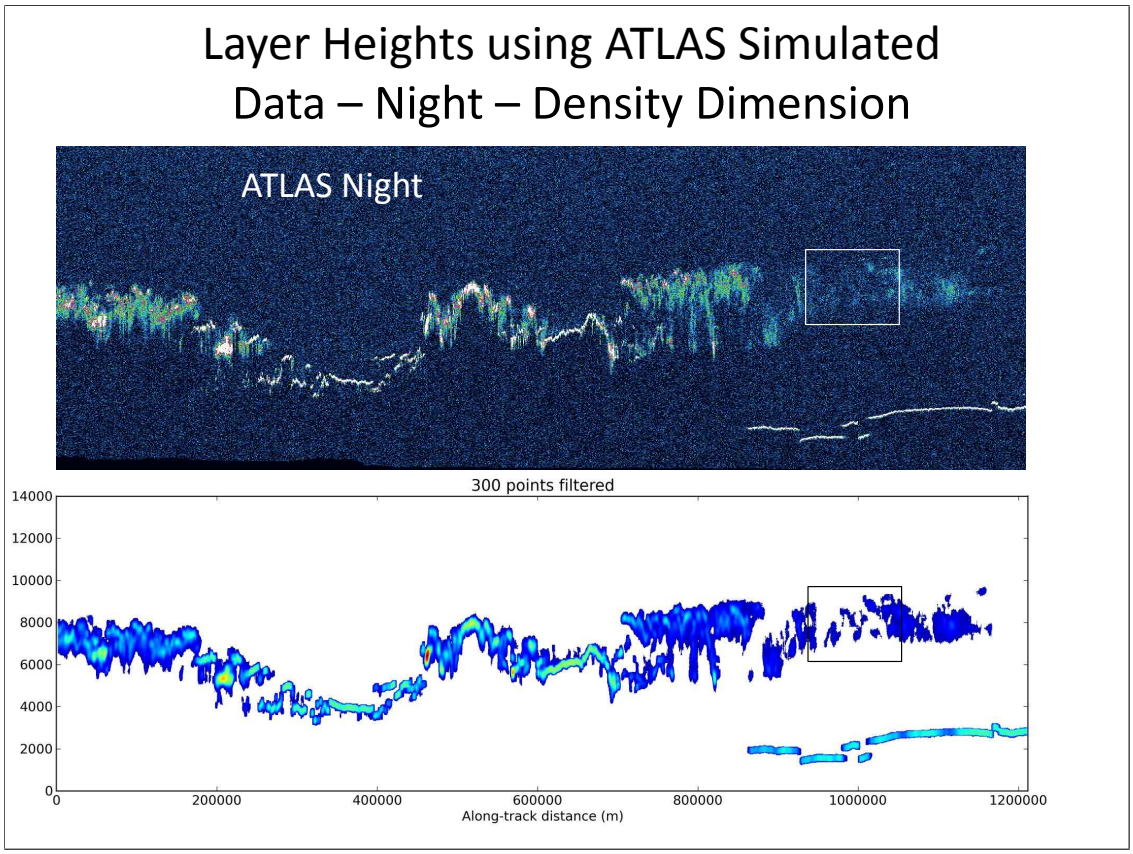


Figure 13. Detection of clouds in MABEL-based ATLAS day-time data using the Density-Dimension Method. MABEL data set 02Apr12.02 (same data set as used in the algorithm description, Figures 1-8).

During some of the 2012 MABEL flights, Cloud Physics Lidar (CPL) data were collected simultaneously. This allows a visual comparison of MABEL data analysis results with CPL data, which can be used to examine the performance of the cloud-layer-detection algorithm. In the following, "ATLAS" data are simulated ICESat-2 data based on MABEL data from 2012 ("M-ATLAS" data). (Note that these data have not been range-corrected nor background-noise-corrected.)

Figure 14 shows that even for day-time data, which have a much higher noise level than night-time data, most cloud layers that are visible in CPL data can be detected in M-ATLAS data using the

density-dimension algorithm, although they are barely visible to the eye in the ATLAS data.

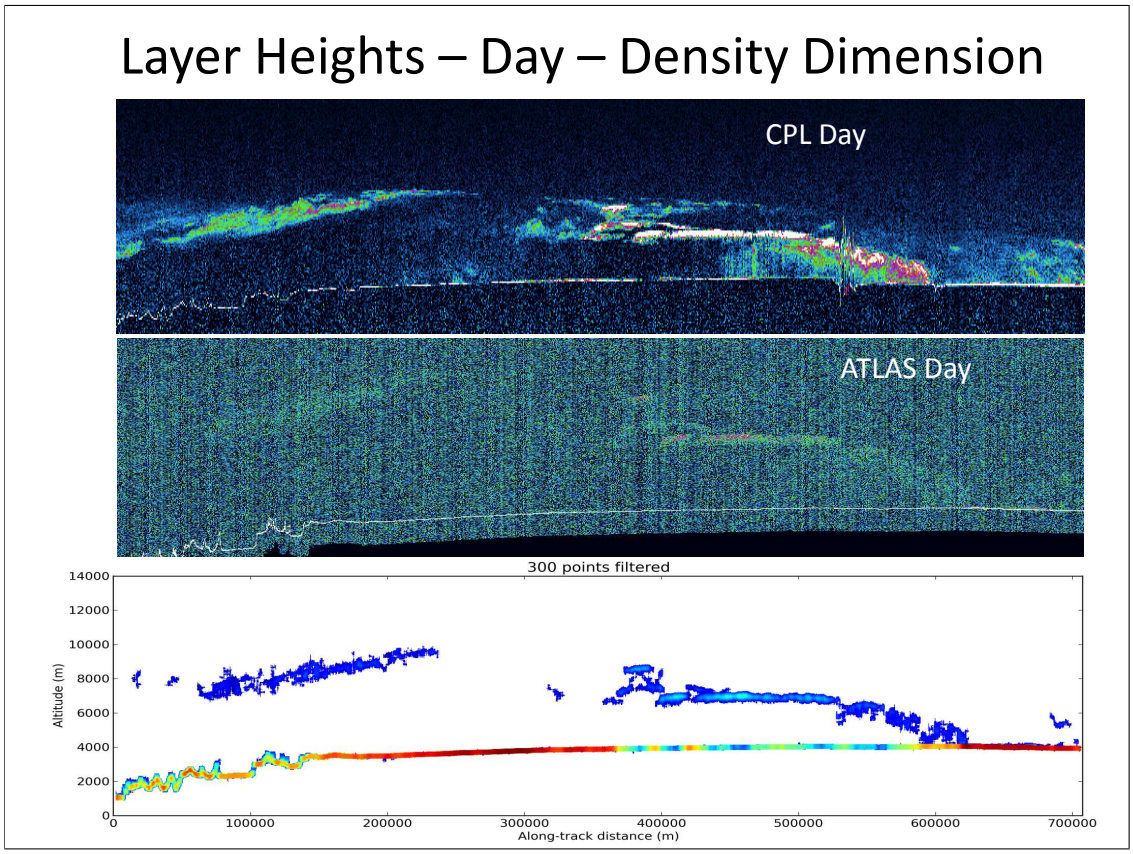


Figure 14. Detection of clouds over Greenland in ATLAS data using the Density-Dimension Method. Top: CPL data. Middle: MABEL-based ATLAS day-time data. Bottom: Result of analysis with density-dimension algorithm.

In Figure 15, the situation is worse, as higher noise obscures the clouds in the ATLAS data entirely (middle panel), however, some of the clouds that are recorded in CPL data can still be identified in the ATLAS data using the density-dimension algorithm. For these 2012 examples, density was run once, using parameters as in the Table 2a (method version A).

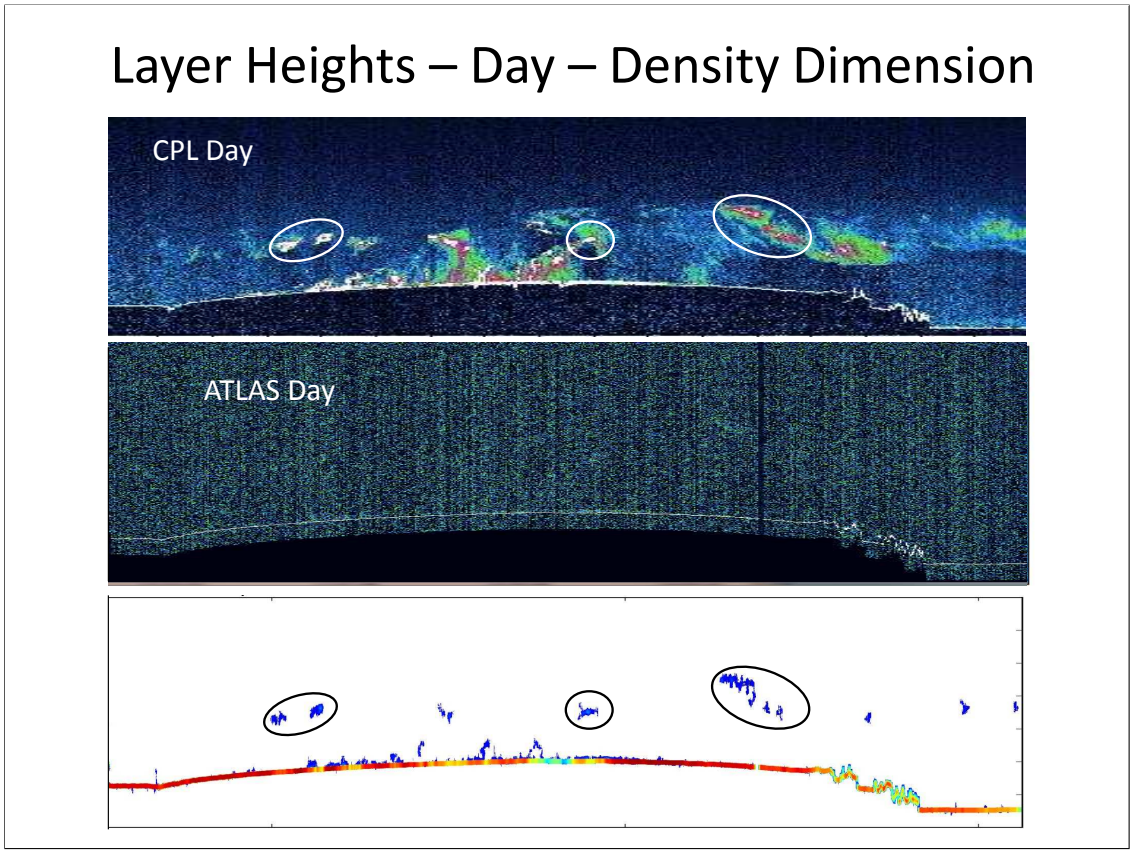


Figure 15. Detection of clouds over eastern Greenland in ATLAS data using the Density-Dimension Method. Top: CPL data. Middle: MABEL-based ATLAS day-time data. Bottom: Result of analysis with density-dimension algorithm.

6 Analysis of 2013 M-ATLAS Data

Algorithm flow: Method B

Introductory Notes. Because MABEL atmosphere data collected in 2012 and 2013 MABEL-based simulated ATLAS (M-ATLAS) data have different characteristics, some parts of the code have been changed for the M-ATLAS data analysis. Essential steps of the algorithm are the same. Where there is a branch in the methods, MethodA refers to code used for 2012 MABEL data analysis and MethodB refers to code used for 2013M-ATLAS data analysis. Both versions and the range of the parameters employed are useful for preparing data analysis of the ICESat-2 ATLAS data. A sensitivity study further demonstrates dependencies on parameters and indicates flexibility of the algorithm frame work for adjustments that may be needed post-launch. Lessons learned from ICESat suggest that flexibility to adapt to changes in data characteristics may be a good preparation for the ICESat-2 Mission.

6.1 Correction

Simulated data. The simulation uses MABEL photon counts, binned into bins matching the 400-shot sums of expected ICESat-2 ATLAS data. Early M-ATLAS data are in essence MABEL 2012 data. 2013 M-ATLAS data are constructed to mimic NRB Profile data (see section 2 and section 3.1):

$$NRB = (raw_photon_count - background_noise)\ r^2 \tag{23}$$

where r is the range from the aircraft to the return height.

NRB is then normalized.

Corrections. The range correction causes the simulated NRB_Prof data to exhibit a vertical gradient, as is apparent in the cloud free regions indicated in Figure 16. The range correction also has

a multiplicative effect on the noise level, increasing with range. This effect is particularly severe, because the height of the atmosphere data extends to the height of the observing aircraft above the ground (it will be different - smaller - for satellite data). This effect however, complicates the identification of atmospheric layers in the simulated M-ATLAS data.

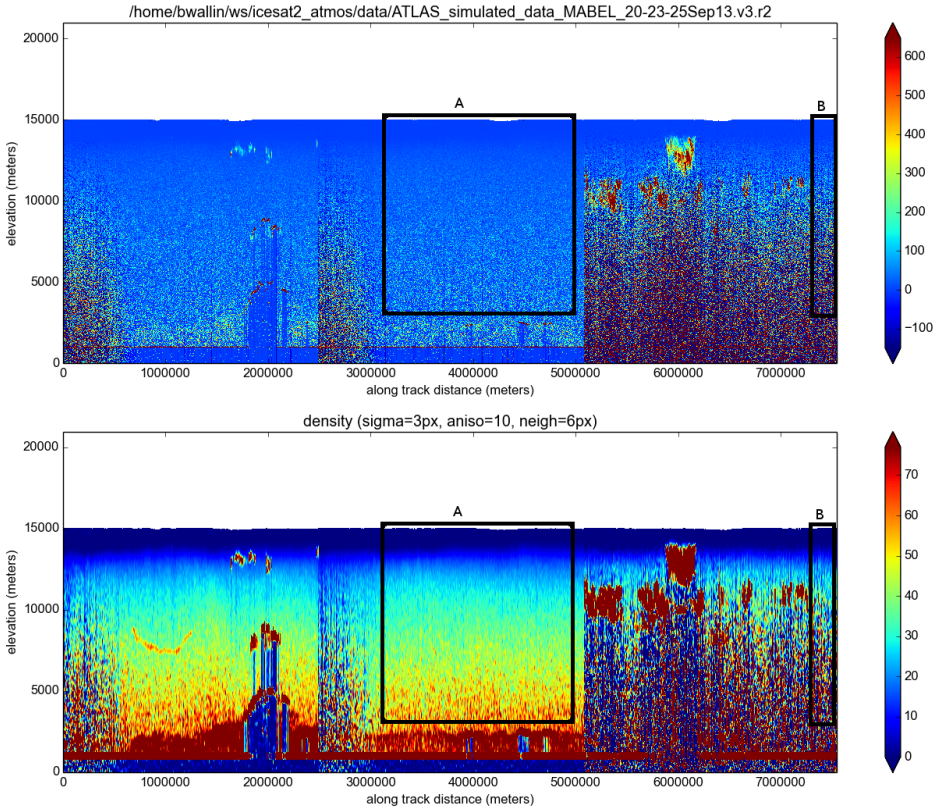


Figure 16. Example of noise gradients for cloud-free regions in background-subtracted range-square corrected M-ATLAS data atlas-simulated-data-mabel-20-23-25sep13.v3.r2. Gradient samples indicated in Boxes A and B.

This effect can be analyzed and corrected for as follows:

After averaging the values in areas A and B for each height bin (see Figure 17), the gradient, g, follows a power curve

$$g \propto r^{0.6} \tag{24}$$

and thus can be corrected for, while also reducing the noise amplification, by reversing part of the

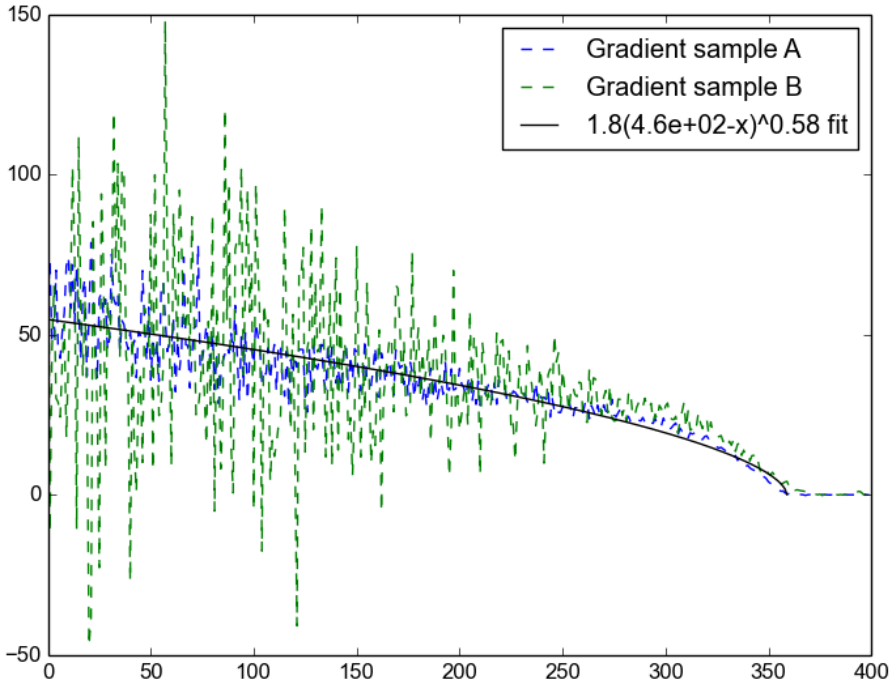


Figure 17. Example of noise gradients for cloud-free regions in background-subtracted range-square corrected M-ATLAS data atlas-simulated-data-mabel-20-23-25sep13.v3.r2. Gradient sample areas indicated in Fig. 16.

range square correction

$$z = \frac{NRB}{r^{0.6}} \tag{25}$$

This correction step is applied after loading data and before the analysis, using Method version B, as summarized in Listing 28, with application of density calculation twice. The effect of the correction can be seen in the following figure (18).

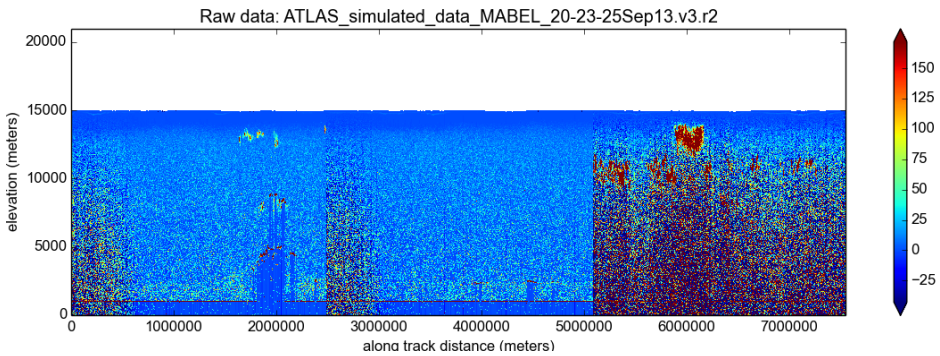


Figure 18. Simulated data set after range-dependent correction. M-ATLAS data atlas-simulated-data-mabel-20-23-25sep13.v3.r2

Pseudocode for these steps is given here:

```
define compute_range_matrix(data, start_range):
    # computes ranges from source for each element of data matrix
    # arguments:
    # data — input matrix of data
    # start_range — range of top valid data point (provided with dataset)

    cell_height = 30

    # shape returns the size of the matrix data
    [number_of_rows, number_of_columns] = shape(data)

    # zeros([n, m]) returns an n by m matrix filled with zeros
    range_matrix = zeros([number_of_rows, number_of_columns])
    for j in 0 to number_of_columns-1:
        # r is None for flagged data at top
        r = None
        for i in 0 to number_of_rows-1:
            # No data flag is -9999
            if histo[number_of_columns-i, j] != -9999:
                if r is None:
                    r = max(0, start_range[j])
                else:
                    r += cell_height/1000
                range_matrix[number_of_columns-i, j] = r

    return range_matrix
```

Listing 29: Pseudo-Code: Range Calculation

```
define correct_power(data, start_range):
    # computes ranges from source for each element of data matrix
    # arguments:
    # data — input matrix of data
    # start_range — range of top valid data point (provided with dataset)

    # shape returns the size of the matrix data
    [number_of_rows, number_of_columns] = shape(data)
```

```
    # compute range from source for each measurement
    range_matrix = compute_range_matrix(data, start_range)

    # set elements less than 1 to 1 to avoid near range numerical artifacts
    for i in 0 to number_of_rows-1:
        for j in 0 to number_of_columns-1:
            if range_matrix[i, j] < 1:
                range_matrix[i, j] == 1

    # apply power correction
    for i in 0 to number_of_rows-1:
        for j in 0 to number_of_columns-1:
            data = data[i,j]/range_matrix[i,j]^(.6)

    return data
```

Listing 30: Pseudo-Code: Range Calculation

6.2 Application: Analysis of a Data Set with Three Different Types of Conditions

To demonstrate the auto-adaptive capabilities of the density-dimension algorithm, a data set was composed from three different flight segments, which are exemplary of different types of conditions, noise characteristics and different types of atmospheric layers.

The first segment is MABEL-based simulated ATLAS data from September 20, 2013. It begins in the late afternoon and continues into nighttime. This data segment contains boundary layer aerosol in the first two thirds and multi-layer clouds in the last third or so of the data. The boundary layer aerosol is confined below about 2-3 km and the clouds are at about 5, 8 and 12 km altitude. The second segment, from September 23, is mainly clear, but does have a well-defined boundary layer below about 2 km composed of aerosol. There are also some sporadic cumulus clouds at the boundary layer top. The third segment is from a daytime MABEL flight on September 25, 2013 and contains many more clouds between 8 and 13 km altitude. In the analysis that follows, these three segments are joined together to better represent the varying background and atmospheric conditions that ATLAS will encounter.

The analysis uses algorithm flow for Method B and the following parameters (as given in Table 2b). (The information below is included in Table 2b, but given here as well for clarity.)

Anisotropy factor in meters: $a_m = 10$ in density run 1, $a_m = 20$ in density run 2

$sigma = 3$ in density run 1, $sigma = 6$ in density run 2

cutoff =1 in in density runs 1 and 2

Kernel sizes: density run 1: $n_1 = 7$ and $r_{1,y} = 3$; $m_1 = 7$ and $r_{1,x} = 3$; kernel (7,7)

density run 2: $n_2 = 13$ and $r_{2,y} = 6$; $m_2 = 25$ and $r_{2,x} = 12$; kernel (13,25)

Window size for threshold determination (threshold_segment_length): 20 (in both runs)

Threshold-bias: 60.0 in density run 1, 0.0 in density run 2

Threshold sensitivity: 1 (in both runs)

Size of clusters in small-cluster removal step: 600 (in both runs)

Optional power correction for aircraft data used: range multiplied with $r^{0.6}$ for $r > 1$, effectively $r^{1.4}$ instead of r^2

These parameters will be used as the default parameters in future analyses. Results are given in Figure 19.

There are no CPL data available for the MABEL flights in 2013, hence a validation as for 2012 data cannot be performed.

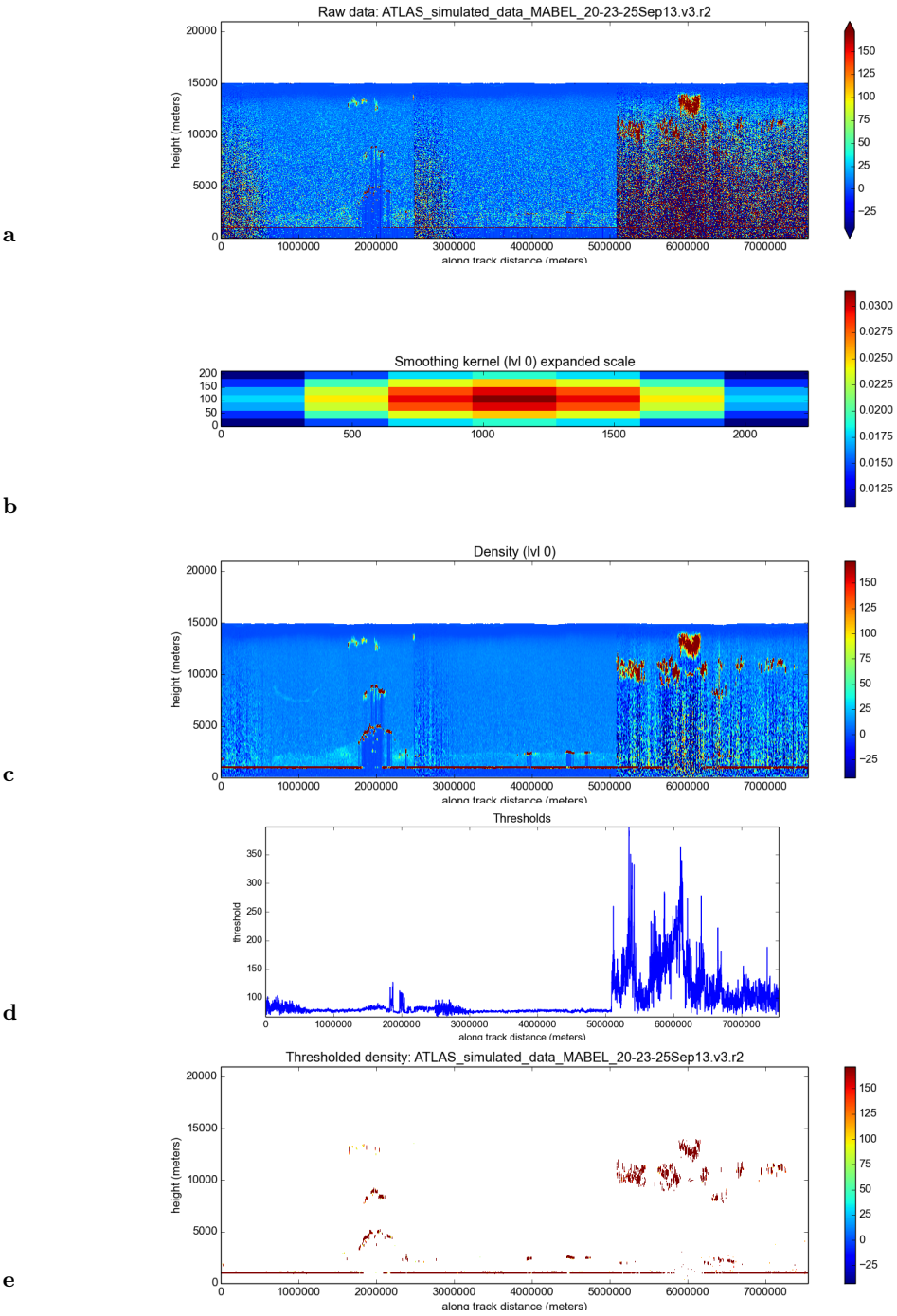


Figure 19. Analysis of triple M-ATLAS data set ATLAS_simulated_data_MABEL_20-23-25Sep13.v3.r2 of night-time data and day-time data (based on MABEL atmosphere data collected 20-23-25Sep2013). Data simulation with background and range-square correction. Results of analysis using density-dimension algorithm, version B.

(a) Data after application of range-dependent correction

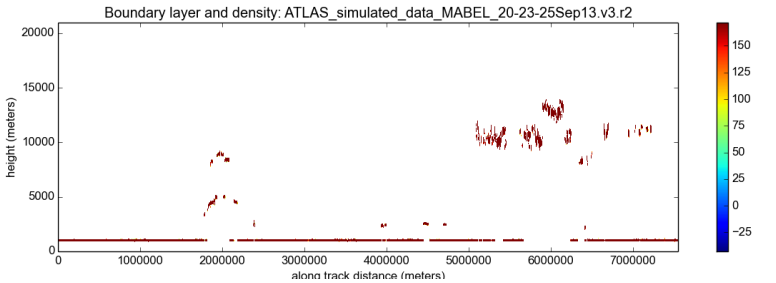
(b) Kernel used in density function in run1 (radius1=3, nbhd=7 ($r_{1,x} = r_{1,y} = 3$), $\sigma = 3$, $a_{m_2} = 10$, cutoff=1)

(c) Density1

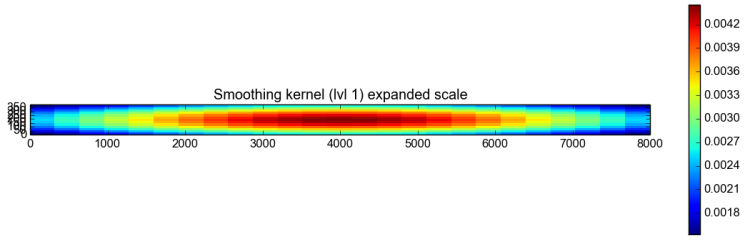
(d) Thresholds

(e) Cloud mask 1 (before small-cluster removal)

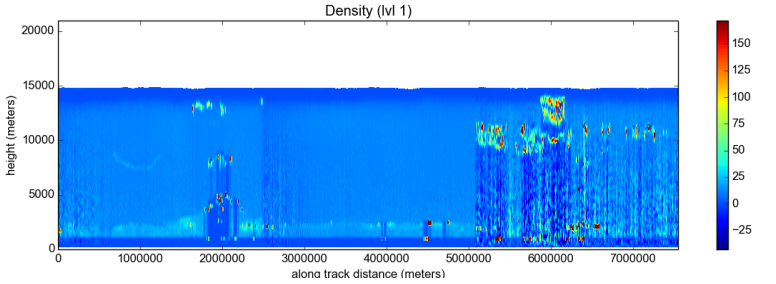
f



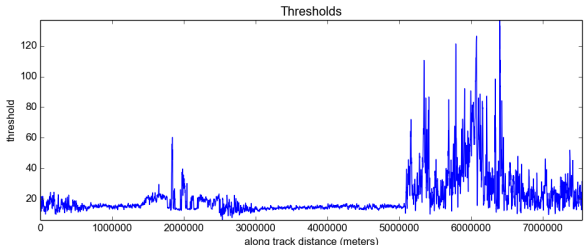
g



h



i



j

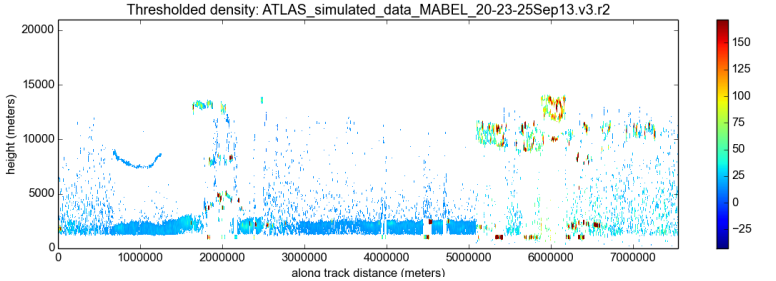


Figure 19, ctd. Analysis of triple M-ATLAS data set ATLAS_simulated_data_MABEL_20-23-25Sep13.v3.r2 of night-time data and day-time data (based on MABEL atmosphere data collected 20-23-25Sep2013). Data simulation with background and range-square correction. Results of analysis using density-dimension algorithm, version B.

(f) Cloud mask 1 (after small-cluster removal)

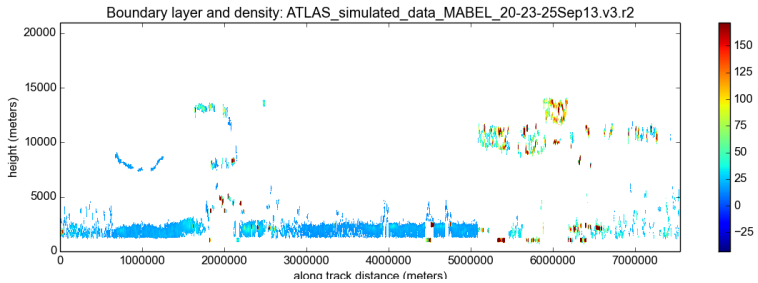
(g) Kernel used in density function in run2 ($r_{2,x} = 12$, m=25, $r_{2,y} = 6$, n=13, $\sigma = 6$, cutoff=1, $a_{m_2} = 20$)

(h) Density2

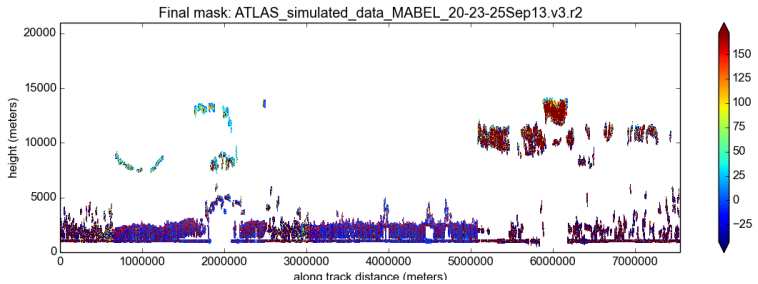
(i) Thresholds

(j) Cloud mask 2 (before small-cluster removal). Shows that small-cluster removal is needed.

k



l



m

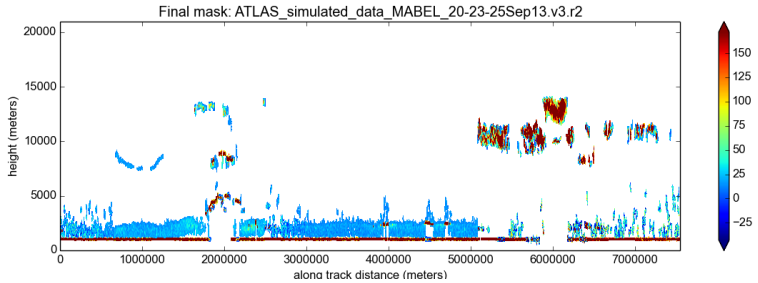


Figure 19, ctd. Analysis of triple M-ATLAS data set ATLAS_simulated_data_MABEL_20-23-25Sep13.v3.r2 of night-time data and day-time data (based on MABEL atmosphere data collected 20-23-25Sep2013). Data simulation with background and range-square correction. Results of analysis using density-dimension algorithm, version B.

(k) Cloud mask 2 (after small-cluster removal)

(l) Combined cloud mask with data

(m) Combined cloud mask with density 1

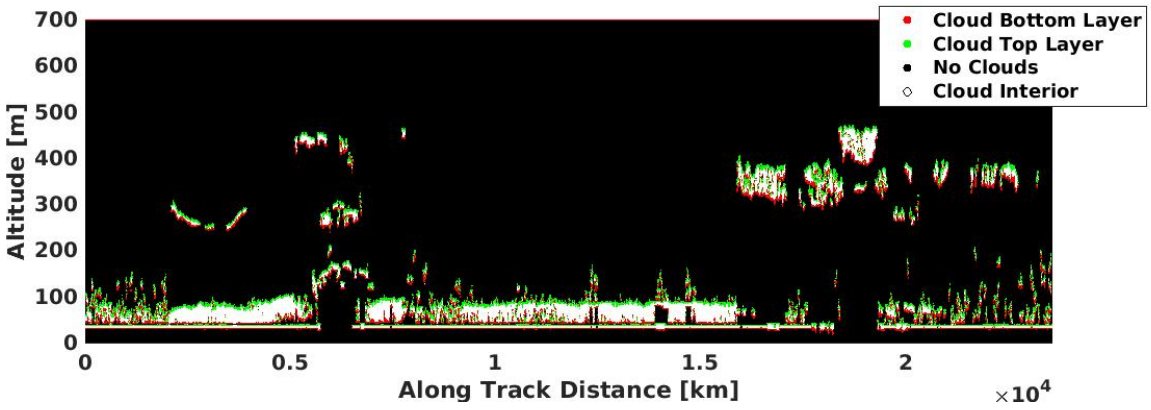


Figure 19, ctd. (o) Layer boundaries for example in Fig. 19. Data set ATLAS_simulated_data_MABEL_20-23-25Sep13.v3.r2.

7 Sensitivity Studies (for 2013 M-ATLAS Data)

The sensitivity studies are carried out to analyze and illustrate the effects of varying the parameters and prepare for analysis of future ICESat-2 data that may have different characteristics.

7.1 Sensitivity Studies for Single-Density Runs

To demonstrate the effects of the primary parameters, experiments with single-density are given in section (7.1) The following experiments are carried out: Experiment 1: Changing the neighborhood (Figure 20),
experiment 2: changing σ (Figure 21),
and experiment 3: changing anisotropy (for two different neighborhoods, Figure 22).

In detail, the following experiments are carried out:

Experiment 1: Changing the neighborhood r (Figure 20)

Fig. 20a: $\sigma = 5$, $a_m = 10$, r=2, kernel (5,5)

Fig. 20b: $\sigma = 5$, $a_m = 10$, r=3, kernel (7,7)

Fig. 20c: $\sigma = 5$, $a_m = 10$, r=4, kernel (9,9)

Fig. 20d: $\sigma = 5$, $a_m = 10$, r=5, kernel (11,11)

Fig. 20e: $\sigma = 5$, $a_m = 10$, r=6, kernel (13,13)

Fig. 20f: $\sigma = 5$, $a_m = 10$, r=7, kernel (15,15)

Fig. 20g: $\sigma = 5$, $a_m = 10$, r=8, kernel (17,17)

Fig. 20h: $\sigma = 5$, $a_m = 10$, r=10, kernel (21,21)

Experiment 2: Changing σ (Figure 21)

Fig. 21a: r=5, kernel (11,11), $a_m = 10$, $\sigma = 1$

Fig. 21b: r=5, kernel (11,11), $a_m = 10$, $\sigma = 2$

Fig. 21c: r=5, kernel (11,11), $a_m = 10$, $\sigma = 3$

Fig. 21d: r=5, kernel (11,11), $a_m = 10$, $\sigma = 4$

Fig. 21e: r=5, kernel (11,11), $a_m = 10$, $\sigma = 6$

Fig. 21f: r=5, kernel (11,11), $a_m = 10$, $\sigma = 8$

Fig. 21g: r=5, kernel (11,11), $a_m = 10$, $\sigma = 16$

Fig. 21h: r=5, kernel (11,11), $a_m = 10$, $\sigma = 140$

Experiment 3: Changing anisotropy a_m (with r=5 and r=10) (Figure 22)

Fig. 22a: $\sigma = 5$, $a_m = 3$, r=5, kernel (11,11)

Fig. 22b: $\sigma = 5$, $a_m = 3$, r=10, kernel (21,21)

Fig. 22c: $\sigma = 5$, $a_m = 10$, r=5, kernel (11,11)

Fig. 22d: $\sigma = 5$, $a_m = 10$, r=10, kernel (21,21)

Fig. 22e: $\sigma = 5$, $a_m = 20$, r=5, kernel (11,11)

Fig. 22f: $\sigma = 5$, $a_m = 20$, r=10, kernel (21,21)

Fig. 22g: $\sigma = 5$, $a_m = 30$, r=5, kernel (11,11)

Fig. 22h: $\sigma = 5$, $a_m = 30$, r=10, kernel (21,21)

All experiments use the following parameters: cutoff=2 (i.e. 2 standard-deviations), minimal cluster size not removed: 600,

For each experiment and sub experiment, the following resultant figure panels are given from top to bottom:
(1) weight matrix (kernel),
(2) density,
(3) preliminary cloud mask after application of thresholds, with density values within cloud areas, and (4) density in cloud areas, for final cloud mask after application of small-cluster removal.

NOTE: Figures 20-22 are located in Appendix S (Sensitivity Studies with Figures, Section 7).

7.2 Sensitivity Studies for Double-Density Runs

The complete figure series for the double-density run is given in Figure 23;

parameters:

threshold_segment_length(=window) = 20, σ = 3,6, a_m = 10, 20, base_threshold = 70, 0, threshold_sensitivity = 1, 1.

To illustrate the dependencies in the framework of "running density twice", the following experiments are undertaken; parameters listed are parameters changed for density run 2:

Experiment 1: Changing threshold window (Figure 24), (a) window=10, (b) window= 30

Experiment 2: Changing threshold sigma (Figure 25), (a) σ = 5, (b) σ = 7

Experiment 3: Changing anisotropy (Figure 26), (a) a_m = 10, (b) a_m = 30

Experiment 4: Changing base threshold (Figure 27),

˜ (a) base_threshold = -3, (b) base_threshold = 3

Experiment 5: Changing threshold sensitivity (Figure 28),

˜ (a) threshold_sensitivity= 0.9, (b) threshold_sensitivity= 1.1

Experiment 6: Changing minimum cluster size (Figure 29),

˜ (a) size_threshold= 100, (b) size_threshold= 500

For each experiment, the following figures are shown (in 6 panels, top to bottom:)

(1) Kernel of density run 2,

(2) density 2,

(3) threshold of density run 2,

(4) preliminary cloud mask, after threshold application to density 2,

(5) cloud mask after small-cluster removal for density run 2,

(6) combined, final cloud mask from density runs 1 and 2, with density-1 values shown.

NOTE: Figures 23-29 are located in Appendix S (Sensitivity Studies with Figures, Section 7).

8 Analysis of GLAS-Data-Based Simulated ICESat-2 Data (2016 Version)

Following analysis of simulated ICESat-2 data based on MABEL data collected in 2013 and formal review of algorithms and ATBD in February 2015, GLAS-data-based ICESat-2 data were simulated and analyzed to further broaden our understanding of expected performance of the ATLAS instrument and data return from the ICESat-2 Mission. To demonstrate the auto-adaptive capabilities of the DDA and to provide a large example, a data set containing two full orbits was generated. Day-night transition are obvious in the analyzed examples in the following figures. The first transition from night-time data to day-time data occurs near 6000 km in along-track distance.

8.1 Simulation

The GLAS calibrated, attenuated backscatter (data product GLA07) are used as basis for the ATLAS atmospheric simulation. With knowledge of the ATLAS instrument characteristics (see Table 3), the GLA07 data can be used directly to obtain the ATLAS photon counts ($P(z)$) through application of the following equation:

$$P(z) = \frac{P_e}{r^2}\beta(z)T^2(z)\Delta z A_t Q_e T_{opt} N_a$$

Where:

r - The range from the satellite to the height z (in m)

$\beta(z)$ - the calibrated attenuated backscatter cross section at height z ($m^{-1}sr^{-1}$)

Δz - the bin size in meters (30 m)

A_t - Area of telescope (m, effective)

$T(z)$ - Atmospheric transmission from top of atmosphere to height z

Q_e - Quantum efficiency of detector

T_{opt} - Transmission of the receiver system optics

N_a - Number of shots summed (nominally 400)

And P_e is the number of photons transmitted by ATLAS which is defined by the laser energy (E) as:

$$P_e = \frac{E\lambda}{hc}$$

Where λ is the laser wavelength (532 nm), h the Planck constant and c the speed of light. The product of $\beta(z)$ and $T(z)$ is essentially the GLAS calibrated attenuated backscatter from the GLA07 product. The GLAS background, which is stored on the GLA07 product in units of photon counts per bin, is scaled to an equivalent ATLAS background by again using knowledge of the ATLAS instrument parameters. This is simply a scaling factor that is computed from the ratio of instrument parameters that govern background magnitude (ATLAS/GLAS). This ratio has a value of 0.1 per shot. However, GLAS used a 40 Hz laser, while ATLAS will have a 10 KHz laser. Thus, for a given unit of time, ATLAS will collect 25 times more background photons than GLAS ($10{,}000/40 \cdot 0.1$). ATLAS detector dark counts are also added to the background (10 KHz).

Since the GLAS atmospheric profile spans the vertical range of -1 to 40 km, the folding effect that ATLAS will experience (due to the 10 KHz laser) can be simulated. ATLAS simulated data based on the GLAS data from the 15 to 30 km range and the 30 to 45 km range is added to the ATLAS simulated data from the lowest 15 km. This is done bin by bin as the scattering at any height z (where Z $<$ 15 km) is equal to the sum of the scattering at height Z, $Z + 15$ km, $Z + 30$ km, etc.

ATLAS Instrument Parameter	**Current (pre-lab measured) Value**
Laser Repetition Rate	10 KHz
Laser Energy	120 μ J
Telescope Effective Area	0.43 m^2
Telescope FOV	83 μr
Detector Quantum Efficiency	0.15
Detector Dead Time	3 ns
Detector Dark Count Rate	10 KHz
Bandpass Filter Width	30 pm
Nominal Receiver Optics Throughput	0.30
Nominal Orbit Height	495 km
Laser/Telescope FOV Spot Size (on ground)	14 m/ 41 m

Table 3 Pertinent ATLAS instrument parameters and their values at present.

8.2 Analysis

The analysis of the GLAS-based simulated ICESat-2 data is carried out by application of the DDA using code version v105, which includes the method A/B synthesis. As can be seen in Figure 30, application of the DDA with the parameters that worked best for analysis of M-ATLAS data (as presented in section 6.2) does not yield especially good results for the GLAS-based ICESat-2 data. This is as expected, because the two types of simulated data have different statistical properties. Notably, using the parameter combinations and exactly the same algorithm that worked for analysis of M-ATLAS data (method version B and a double-density run), the algorithm picks up many false positives in the lower density ranges. This effect may be attributed to the fact that MABEL-based simulated data have weaker signals and more background noise than GLAS-based simulated data. As an aside, this example illustrates nicely that we learn new things from each experiment and that the algorithm needs to be adjustable to different instrument characteristics and observation characteristics, as these may also change during the operational part of the mission.

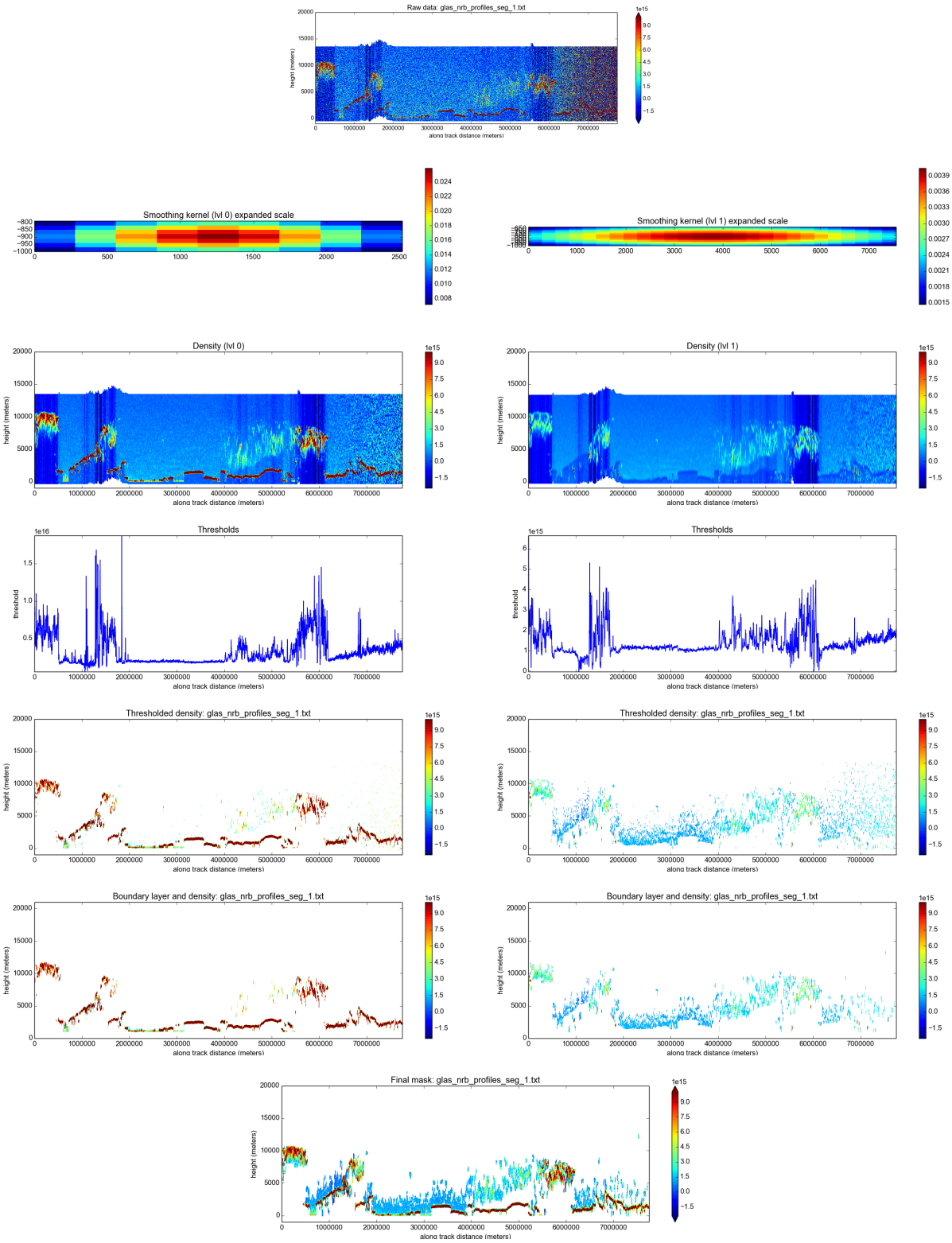


Figure 30. Analysis of GLAS–data based simulated ICESat-2 data using parameters that worked best for MABEL-based simulated ICESat-2 data (M-ATLAS data) and method version B in the method A/B code (v105). Double-density run (t3). Segment 1 of 2-orbit data set.

σ = 3,6	cutoff = 1,1
a_m = 10,20	min cluster size = 600,600
base_threshold = 60E+13,0	downsampling = 1,1
threshold_sensitivity = 1,1	threshold_segment_length = 2,2
quantile = 0.9,0.9	

For GLAS-based simulated ICESat-2 data, an analysis with the following parameters works best (see Figure 31 (t8)):

$\sigma = 3$	cutoff = 1
$a_m = 10$	min cluster size = 600
base_threshold = 60E+13	downsampling = 1
threshold_sensitivity = 0.9	threshold_segment_length = 2
quantile = 0.75	

Table 4. Parameters for single-density analysis of GLAS-data-based simulated ICESat-2 data (t8)

The analysis is carried out for two orbits of GLAS-based simulated ICESat-2 data. The results indicate, among other things, that the algorithm adapts well across boundaries between day-time and night-time data.

Only one density run was needed for this analysis. The parameter combinations for (t8) were determined in a sequence of 15 test runs with different parameter combinations. Then a sensitivity study was carried out, varying each parameter around the values used in (t8), described in section (4.9).

Corrections and NRB data. The range-dependent correction described in section (6.1) does not need to be applied, because these data sets are based on satellite data. note that the data have a factor of E+13.

Notes: Kernel size determined using the *ceiling* function for rounding (as in previous code versions and analyses), according to

$m = 2 \cdot ceil(\sigma_m / x_{res} \cdot cutoff \cdot a_m) + 1$

$n = 2 \cdot ceil(\sigma_m / y_{res} \cdot cutoff) + 1$

where m and n are the kernel dimensions and $\sigma_m = 30\sigma_{bin} = 30\sigma$ and a_m is the anisotropy factor in meters. Bins are 30 by 280 meters, i.e. $x_{res} = 280$ and $y_{res} = 30$.

Results are shown in Fig. 31. For visualization purposes, the 2 orbits of data, which are ≈ 276000

profiles in total, are split into 10 segments of equal size ($\approx$ 27600 each). Analysis uses code v105.

(a)

(b)

(c)

(d)

(e)

(f)

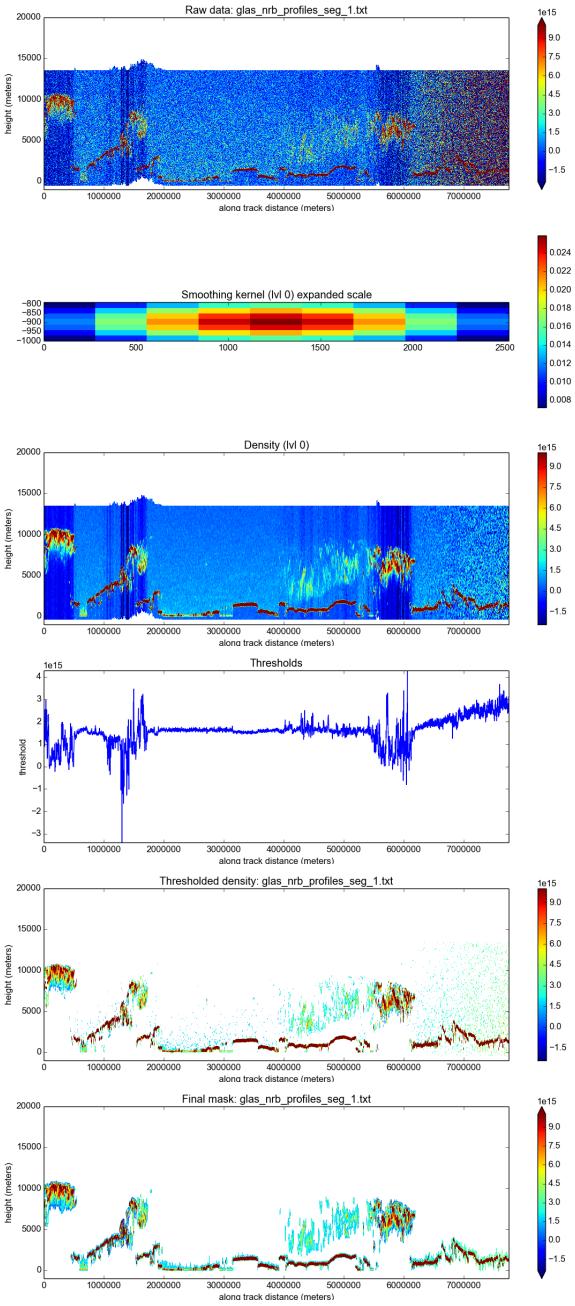


Figure 31-1. Analysis of GLAS–data based simulated ICESat-2 data using the DDA (method A/B, code v105, "best" parameter combination (t8), single-density run). Segment 1 of 2-orbit data set. (a) Raw data, (b) Kernel, (c) Density, (d) Adaptive threshold, (e) Density after threshold is applied and (f) Final density mask after (after small clusters are removed).

$\sigma = 3$	cutoff = 1
$a_m = 10$	min cluster size = 600
base_threshold = 60E+13	downsampling = 1
threshold_sensitivity = 0.9	threshold_segment_length = 2
quantile = 0.75	

(a)

(b)

(c)

(d)

(e)

(f)

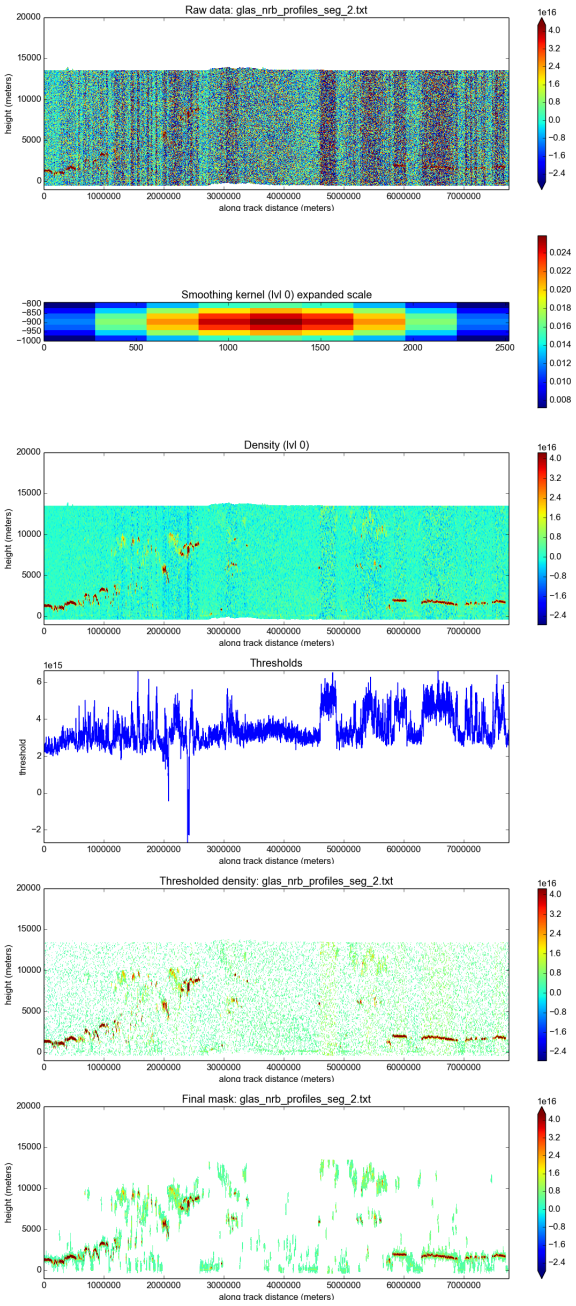


Figure 31-2. Analysis of GLAS–data based simulated ICESat-2 data using the DDA (method A/B, code v105, "best" parameter combination (t8), single-density run). Segment 2 of 2-orbit data set. (a) Raw data, (b) Kernel, (c) Density, (d) Adaptive threshold, (e) Density after threshold is applied and (f) Final density mask after (after small clusters are removed).

$\sigma = 3$	cutoff = 1
$a_m = 10$	min cluster size = 600
base_threshold = 60E+13	downsampling = 1
threshold_sensitivity = 0.9	threshold_segment_length = 2
quantile = 0.75	

(a)

(b)

(c)

(d)

(e)

(f)

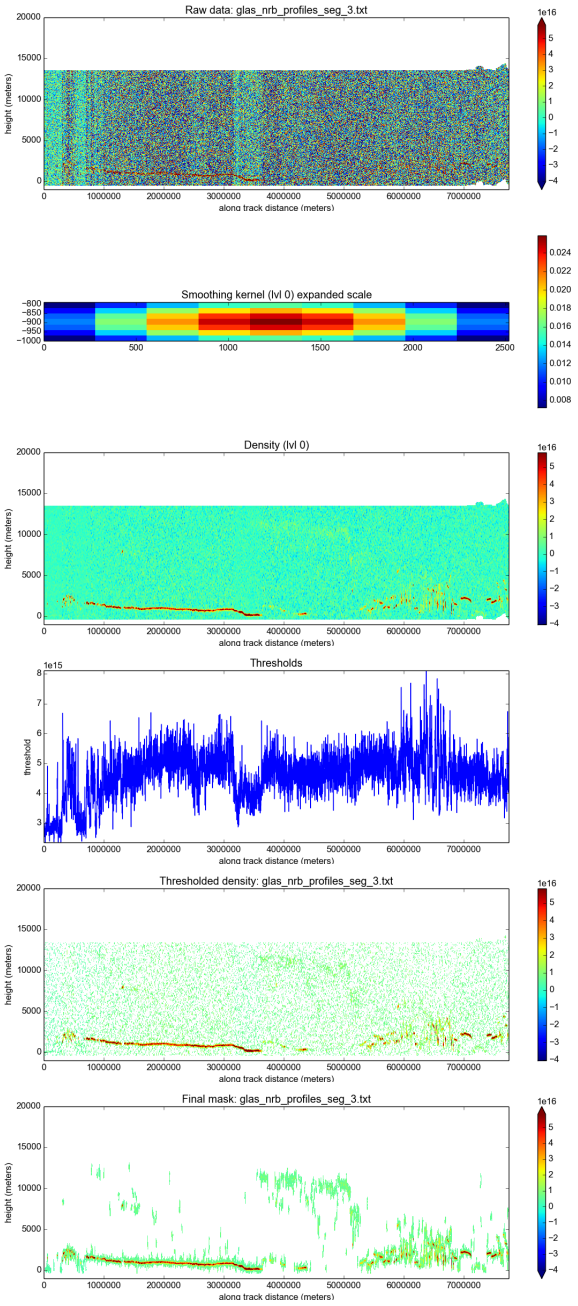


Figure 31-3. Analysis of GLAS–data based simulated ICESat-2 data using the DDA (method A/B, code v105, "best" parameter combination (t8), single-density run). Segment 3 of 2-orbit data set. (a) Raw data, (b) Kernel, (c) Density, (d) Adaptive threshold, (e) Density after threshold is applied and (f) Final density mask after (after small clusters are removed).

$\sigma = 3$	cutoff = 1
$a_m = 10$	min cluster size = 600
base_threshold = 60E+13	downsampling = 1
threshold_sensitivity = 0.9	threshold_segment_length = 2
quantile = 0.75	

(a)

(b)

(c)

(d)

(e)

(f)

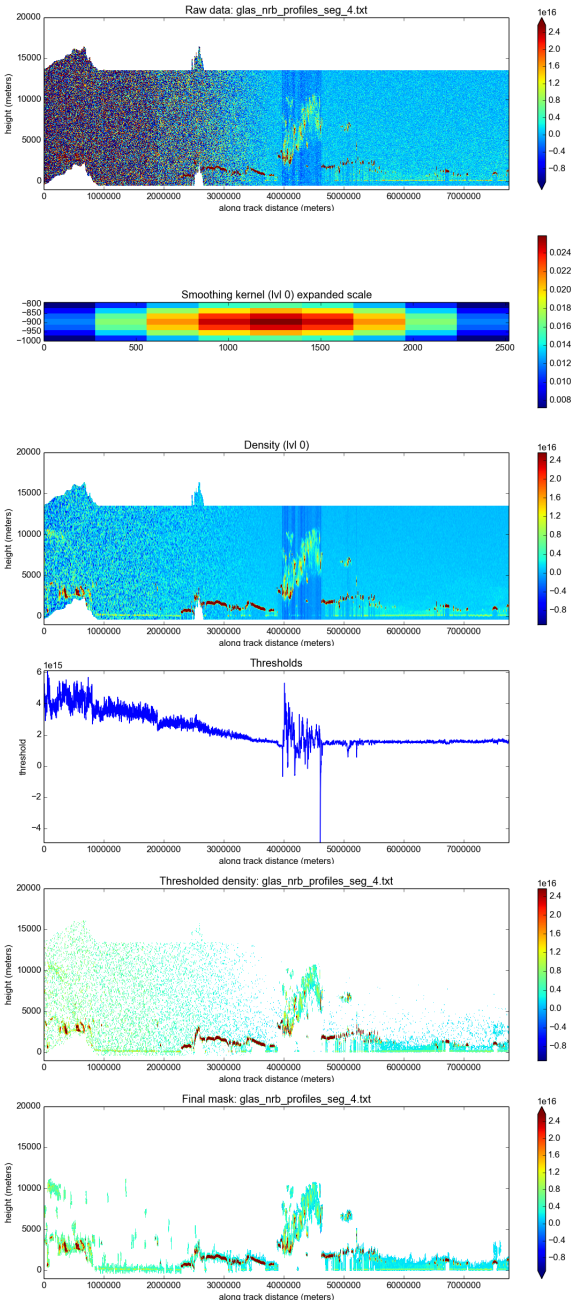


Figure 31-4. Analysis of GLAS–data based simulated ICESat-2 data using the DDA (method A/B, code v105, "best" parameter combination (t8), single-density run). Segment 4 of 2-orbit data set. (a) Raw data, (b) Kernel, (c) Density, (d) Adaptive threshold, (e) Density after threshold is applied and (f) Final density mask after (after small clusters are removed).

$\sigma = 3$	cutoff = 1
$a_m = 10$	min cluster size = 600
base_threshold = 60E+13	downsampling = 1
threshold_sensitivity = 0.9	threshold_segment_length = 2
quantile = 0.75	

(a)

(b)

(c)

(d)

(e)

(f)

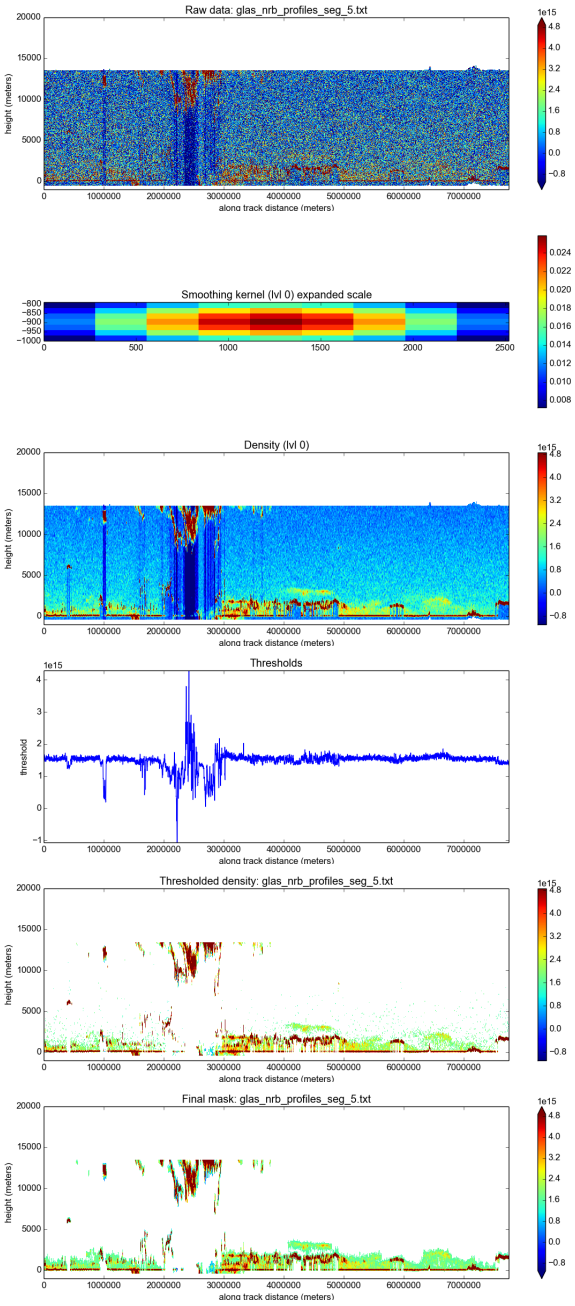


Figure 31-5. Analysis of GLAS–data based simulated ICESat-2 data using the DDA (method A/B, code v105, "best" parameter combination (t8), single-density run). Segment 5 of 2-orbit data set. (a) Raw data, (b) Kernel, (c) Density, (d) Adaptive threshold, (e) Density after threshold is applied and (f) Final density mask after (after small clusters are removed).

$\sigma = 3$	cutoff = 1
$a_m = 10$	min cluster size = 600
base_threshold = 60E+13	downsampling = 1
threshold_sensitivity = 0.9	threshold_segment_length = 2
quantile = 0.75	

(a)

(b)

(c)

(d)

(e)

(f)

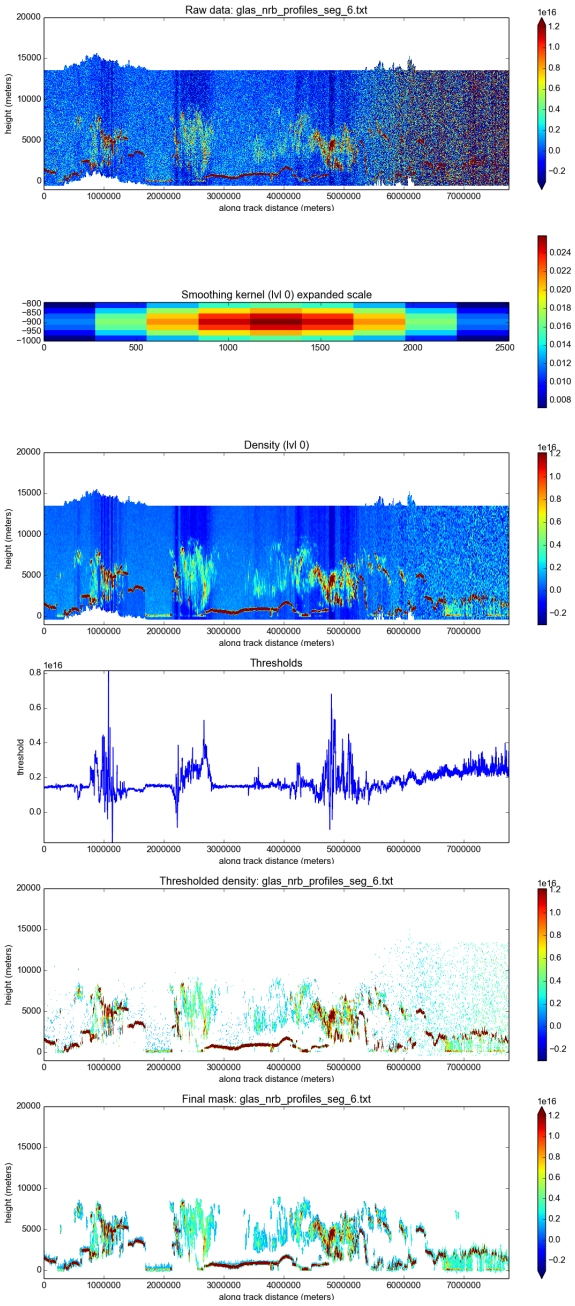


Figure 31-6. Analysis of GLAS–data based simulated ICESat-2 data using the DDA (method A/B, code v105, "best" parameter combination (t8), single-density run). Segment 6 of 2-orbit data set. (a) Raw data, (b) Kernel, (c) Density, (d) Adaptive threshold, (e) Density after threshold is applied and (f) Final density mask after (after small clusters are removed).

$\sigma = 3$	cutoff = 1
$a_m = 10$	min cluster size = 600
base_threshold = 60E+13	downsampling = 1
threshold_sensitivity = 0.9	threshold_segment_length = 2
quantile = 0.75	

(a)

(b)

(c)

(d)

(e)

(f)

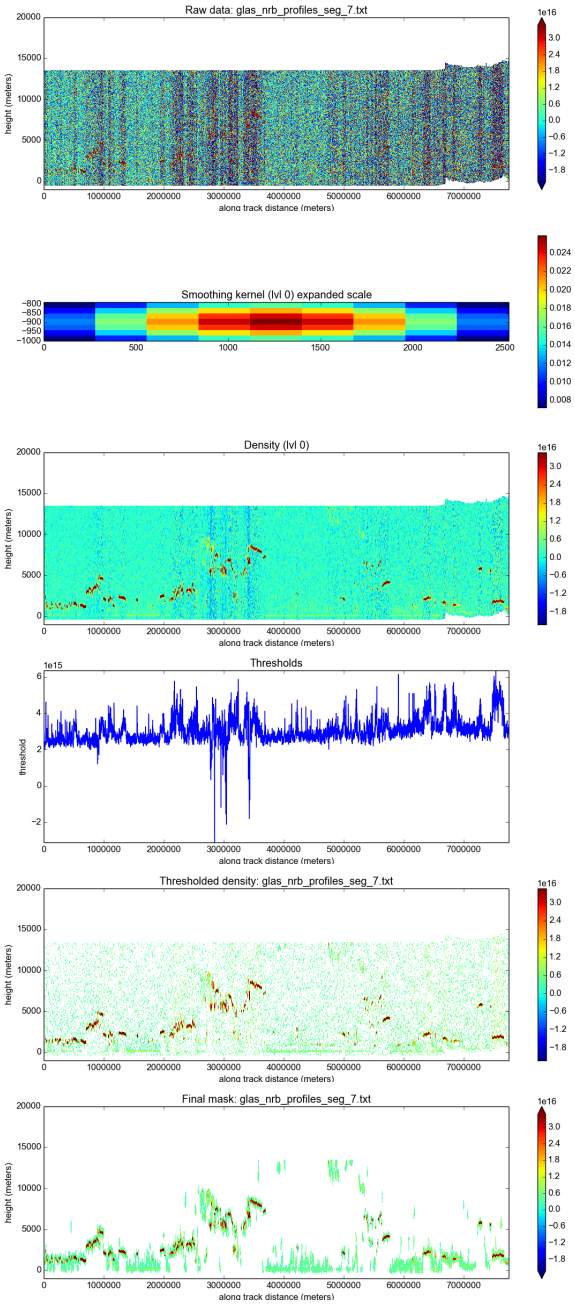


Figure 31-7. Analysis of GLAS–data based simulated ICESat-2 data using the DDA (method A/B, code v105, "best" parameter combination (t8), single-density run). Segment 7 of 2-orbit data set. (a) Raw data, (b) Kernel, (c) Density, (d) Adaptive threshold, (e) Density after threshold is applied and (f) Final density mask after (after small clusters are removed).

$\sigma = 3$	cutoff = 1
$a_m = 10$	min cluster size = 600
base_threshold = 60E+13	downsampling = 1
threshold_sensitivity = 0.9	threshold_segment_length = 2
quantile = 0.75	

(a)

(b)

(c)

(d)

(e)

(f)

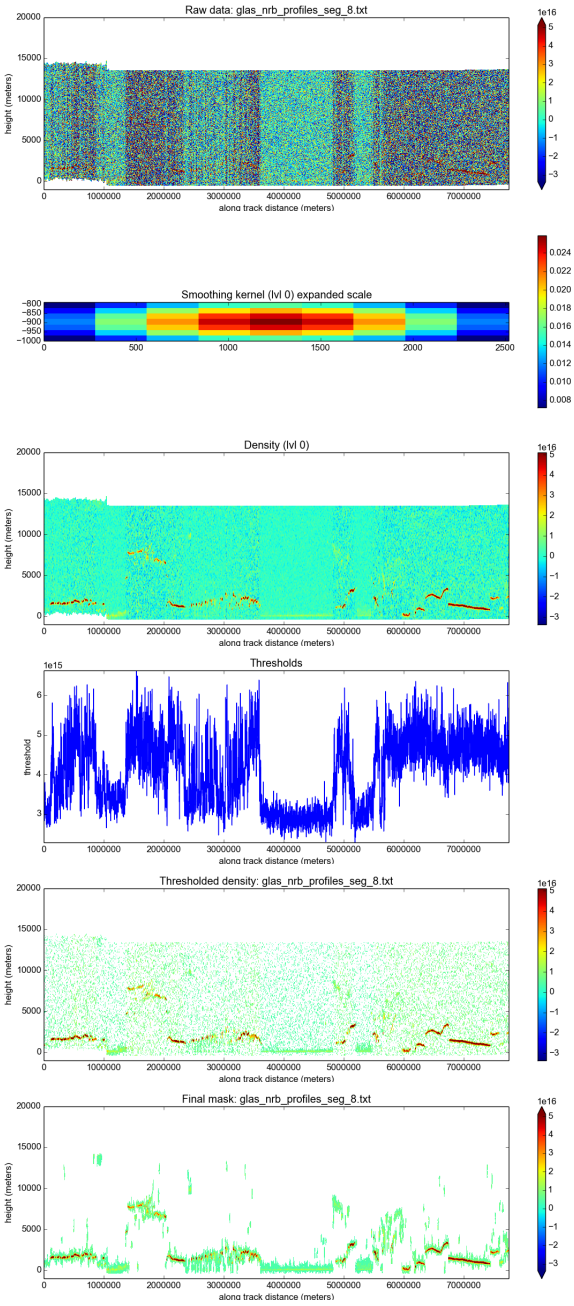


Figure 31-8. Analysis of GLAS–data based simulated ICESat-2 data using the DDA (method A/B, code v105, "best" parameter combination (t8), single-density run). Segment 8 of 2-orbit data set. (a) Raw data, (b) Kernel, (c) Density, (d) Adaptive threshold, (e) Density after threshold is applied and (f) Final density mask after (after small clusters are removed).

$\sigma = 3$	cutoff = 1
$a_m = 10$	min cluster size = 600
base_threshold = 60E+13	downsampling = 1
threshold_sensitivity = 0.9	threshold_segment_length = 2
quantile = 0.75	

(a)

(b)

(c)

(d)

(e)

(f)

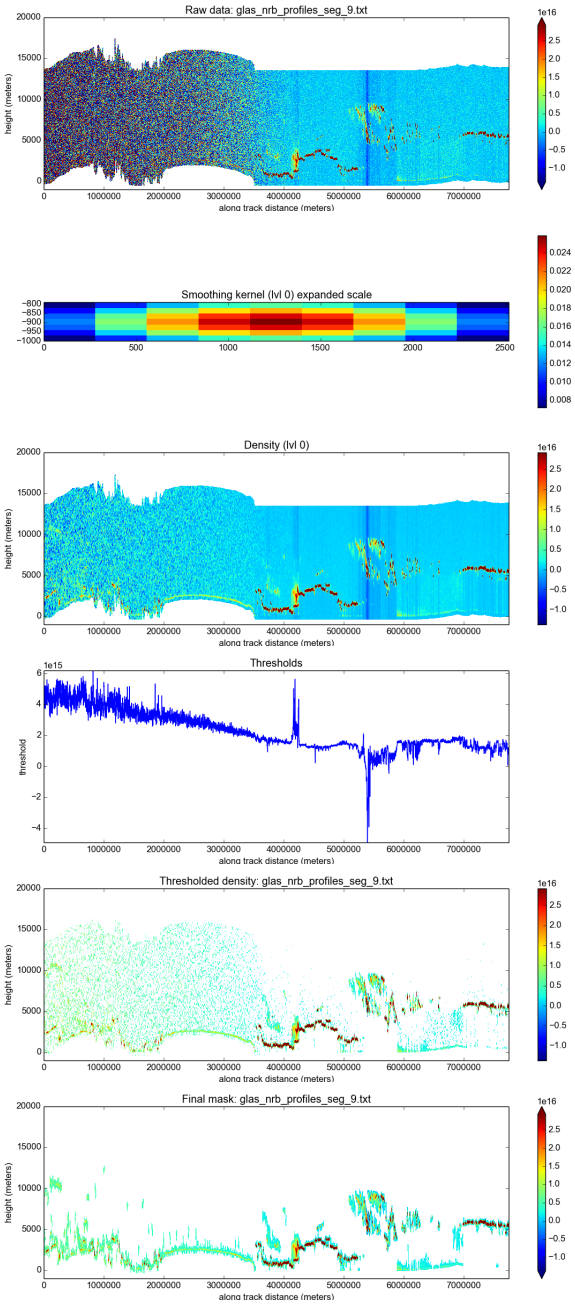


Figure 31-9. Analysis of GLAS–data based simulated ICESat-2 data using the DDA (method A/B, code v105, "best" parameter combination (t8), single-density run). Segment 9 of 2-orbit data set. (a) Raw data, (b) Kernel, (c) Density, (d) Adaptive threshold, (e) Density after threshold is applied and (f) Final density mask after (after small clusters are removed).

$\sigma = 3$	cutoff = 1
$a_m = 10$	min cluster size = 600
base_threshold = 60E+13	downsampling = 1
threshold_sensitivity = 0.9	threshold_segment_length = 2
quantile = 0.75	

(a)

(b)

(c)

(d)

(e)

(f)

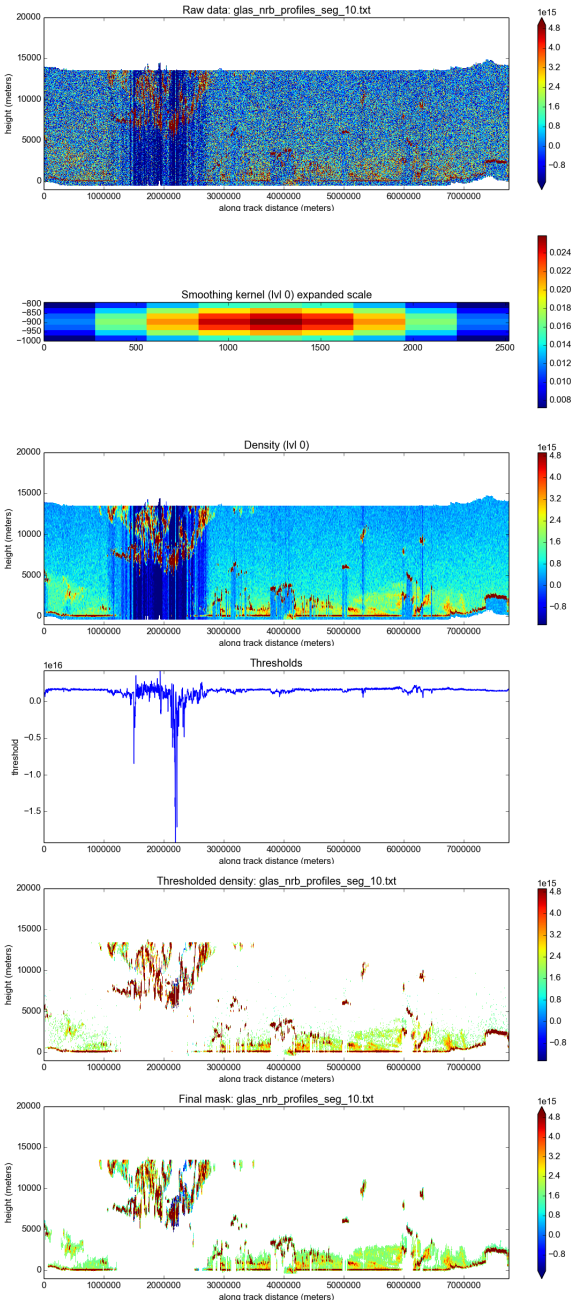


Figure 31-10. Analysis of GLAS–data based simulated ICESat-2 data using the DDA (method A/B, code v105, "best" parameter combination (t8), single-density run). Segment 10 of 2-orbit data set. (a) Raw data, (b) Kernel, (c) Density, (d) Adaptive threshold, (e) Density after threshold is applied and (f) Final density mask after (after small clusters are removed).

$\sigma = 3$	cutoff = 1
$a_m = 10$	min cluster size = 600
base_threshold = 60E+13	downsampling = 1
threshold_sensitivity = 0.9	threshold_segment_length = 2
quantile = 0.75	

8.3 Validation

To evaluate the performance of the DDA for analysis of GLAS-data-based simulated ICESat-2 data (2016 version), a movie was created that runs through the entire two-orbit data set. The top running panel shows the simulated ICESat-2 data, from which the atmospheric layer boundaries were determined. The bottom running panel shows the original GLAS data, with the layer boundaries superimposed (in yellow). The movie can be downloaded from the ICESat-2 SDT website. To keep this document self-contained, segments of the movie are shown in Figures 34 and 35. To aid in visual interpretation of the results, the height of 15km is indicated by a white line. Data will only be recorded to 13.75 km above ground (as represented by on-board DEM) and 0.25 km below ground. Hence clouds in the upper 1.25 km below the white line cannot be detected by the algorithm.

The white line at 15 km illustrates the the height range of the folding effect that is described in the section on data simulation: Since the GLAS atmospheric profile spans the vertical range of -1 to 40 km, the folding effect that ATLAS will experience (due to the 10 KHz laser) can be simulated. ATLAS simulated data based on the GLAS data from the 15 to 30 km range and the 30 to 45 km range is added to the ATLAS simulated data from the lowest 15 km. This is done bin by bin as the scattering at any height z (where Z $<$ 15 km) is equal to the sum of the scattering at height Z, $Z + 15$ km, $Z + 30$ km, etc. As a result, layers that are folded over from above 15km can be seen in some regions. They cannot be statistically distinguished from layers that exist below 15 km. An example of the folding effect is seen in panel (5) of Fig. 32.

Numbers in the figures refer to latitude and sun elevation angle.

The movie (Fig. 34-35) shows the results of application of the DDA with parameters from run (t8). The results indicate that the algorithm functions well across boundaries of day-time and night-time observations and adapts to changes in ASR. Tenuous layers are detected most of the time. Performance during night time is excellent. For day-time data, cloud layers are detected in many cases, while missed in some other cases. Layers were also missed in GLASA data in some instances. This version of the code includes ground, where detected, in the layers. The exemplary segments that are enlarged in Figure 35 illustrate the detection capability of the DDA especially during day-time.

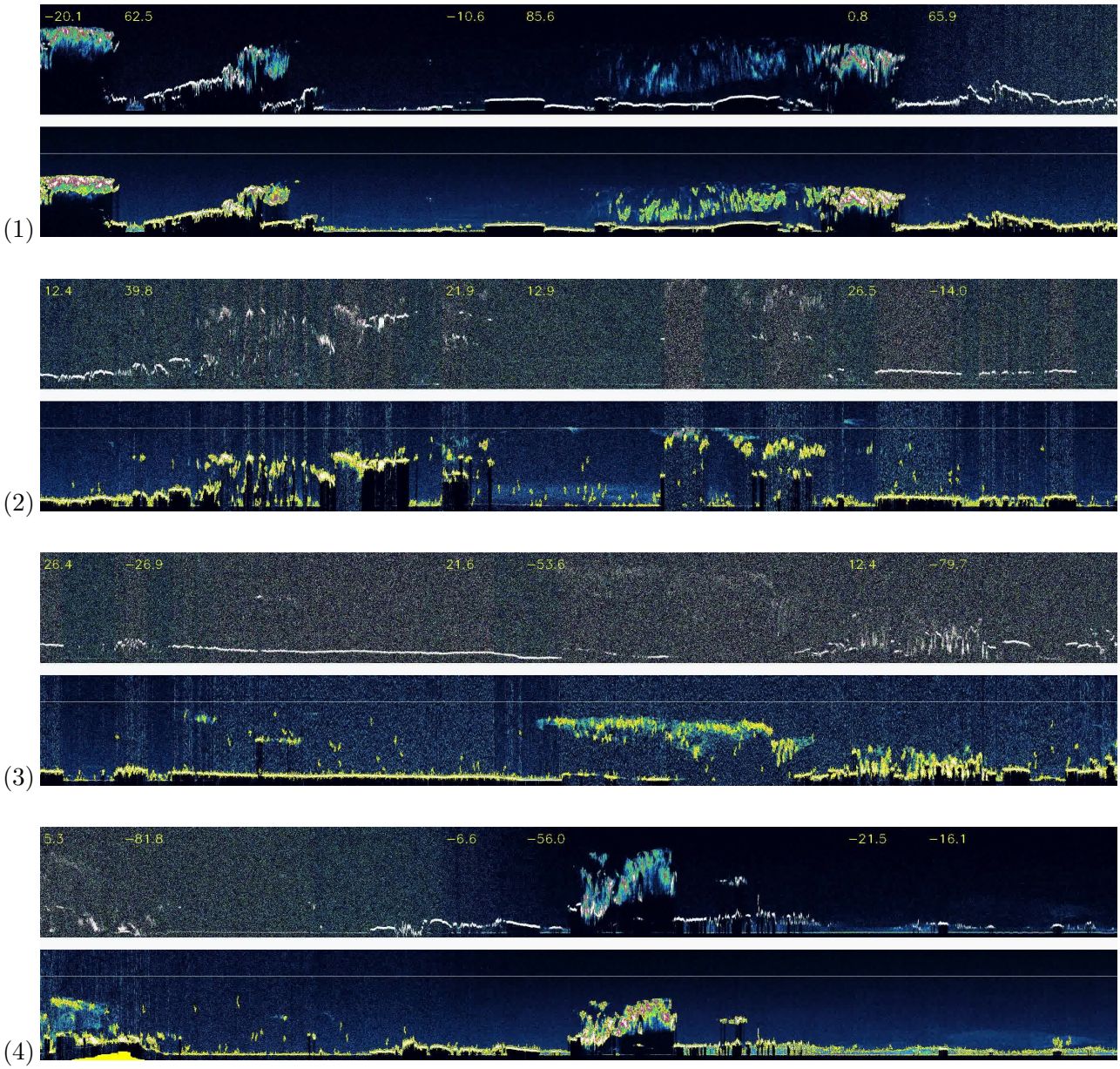


Figure 32. Evaluation of DDA analysis of GLAS-based simulated ICESat-2 data, 2 orbits. Results for run (t8). Top panels show simulated ICESat-2 data, as used for layer identification with the DDA, 13.25 km to -0.25 km relative to the DEM. Bottom panels show GLAS data to a height of 20 km above Earth surface, with a line at 15 km, and, superimposed as yellow lines, layer boundaries as identified in DDA analysis of simulated ICESat-2 data. GLAS data was not available during analysis.

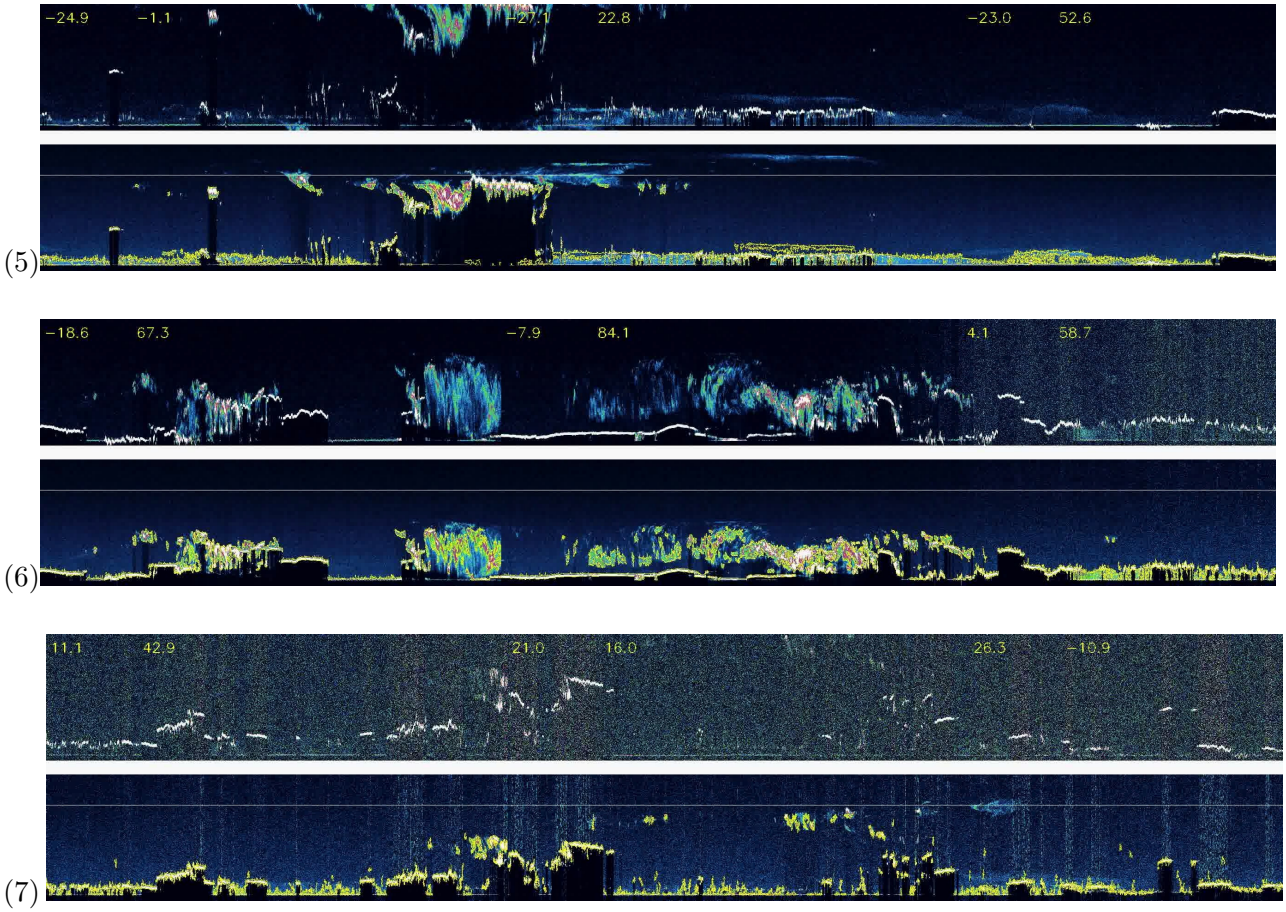
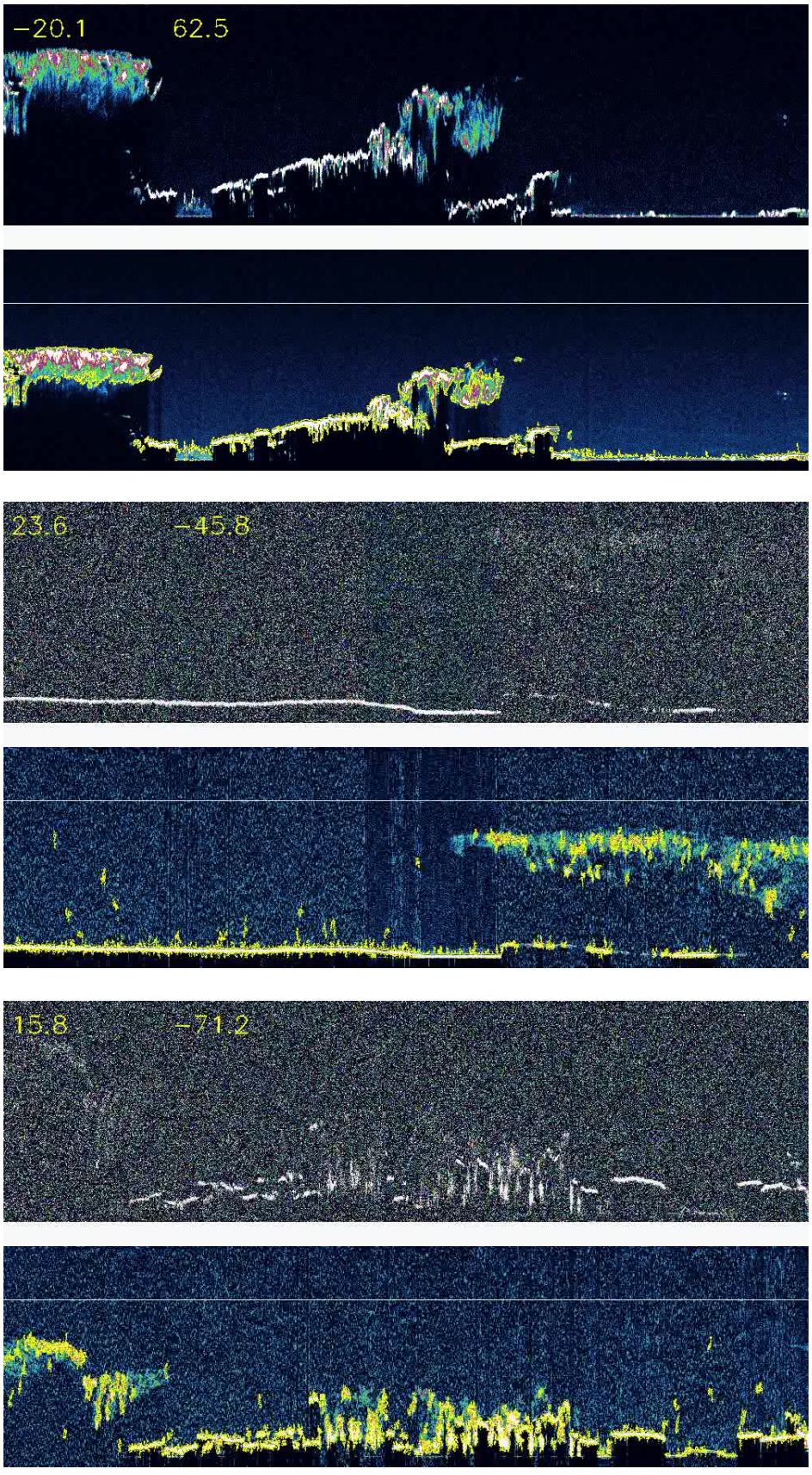


Figure 32, ctd. Evaluation of DDA analysis of GLAS-based simulated ICESat-2 data, 2 orbits. Results for run (t8). Top panels show simulated ICESat-2 data, as used for layer identification with the DDA, 13.25 km to -0.25 km relative to the DEM. Bottom panels show GLAS data to a height of 20 km above Earth surface, with a line at 15 km, and, superimposed as yellow lines, layer boundaries as identified in DDA analysis of simulated ICESat-2 data. GLAS data was not available during analysis. [Note this version of the figure does not include the trailing end of the movie].



a

b

c

Figure 33. Exemplary segments of the movie in Figure 34 enlarged to show details. (a) Night-time data, start of profile (part of segment 1). (b), (c) Day-time data.

9 Sensitivity Studies for GLAS-Data-Based Simulated ICESat-2 Data (2016 Version)

Starting from the parameter combinations used in t8 and deemed best, a sensitivity study was carried out systematically, varying each parameter to above and below the parameter combination of t8 (which still remained best). Quality assessment (what is best?) is carried out by creation of a movie that shows layer boundaries from the DDA superimposed on the original GLAS data. The movie can be accessed under [url]. Here, an illustration of the effect of changing parameters in the sensitivity study is presented in Figure 34, which shows the analysis results for a segment with night-day transitions near near 6000 km.

Note auto-adaptive capability of the algorithm at the night-time day-time transition

All examples were run with code version v105. The sensitivity study includes single-density and double-density runs, and parameter combinations that correspond to method A and method B.

NOTE: Figure 34 is located in Appendix S (Sensitivity Studies with Figures, Section 9).

10 Sensitivity Studies for Analysis of 2017-Oct Version of GLAS-based Simulated ATL04 Data

10.1 Summary, Motivation and Data Sets

Summary. In this section, differences in the characteristics of "GLAS-data-based simulated ICESat-2 data (ATL04) of Oct-2017" data compared to "GLAS-data-based simulated ICESat-2 data (2016 version)" are described. A new sensitivity study is carried out to determine a set of algorithm-specific parameters for auto-adaptive analysis of ATL04 data (with Oct 2017 characteristics.). An important result is that the DDA-algorithm option "running density twice" is required to ascertain correct detection of different types of atmospheric layers during day-time and night-time conditions. As the application of the newly-developed Q/A measure "half-gap confidence flag" (see section (11) Quality Assessment) shows, the layer detection using the double-density runs with the parameter sets (t56) [and (t64)] yields throughout high confidences (mostly 0.8) and somewhat lower confidences where appropriate. – Why two parameter sets at this point? See Section (12) on Testing.

Necessity. Algorithm refinement for the upstream data products, ATL02 and ATL04, and code development and implementation for those products resulted in different characteristics of the NRB values in ATL04, compared to those of the 2016 state-of-the-art simulated GLAS-based ICESat-2 type data. Especially, the energy value in the NRB calculation changed, resulting in NRB values that are almost an order of magnitude larger. The NRB calculation now also includes an ad-hoc identification of optically thick clouds, masking of those clouds and subtraction of everything outside of this mask as "background" (see Part1 of this ATBD). Since this pseudo-background de facto includes thin clouds and aerosols, NRB values can be negative. Because of the division by range-squared, the values can be negative on the order of -1E27.

If we disregard the change of data characteristics and apply the DDA in a single-density run using parameter combination (t8), which was determined best for the 2016 GLAS-data-based simulated ICESat-2 data, we find (see Figure 34-1) that

(a) tenuous cloud layers and aerosol layers are still detected [good]

(b) false positives occur (areas identified as clouds that are likely not clouds) [bad].

These facts require a fresh determination of the set of algorithm-specific parameters. As before, parameters are changed and a new sensitivity study is carried out. Notably, changes in energy levels may also happen after launch, hence it is important to understand the algorithm sensitivity to changes in data characteristics.

In addition, the task of developing confidence measure for atmospheric layer detection as part of quality assessment (Q/A) requires a well-functioning layer detection algorithm, which includes a set of algorithm specific parameters that is matched to the data characteristics.

Data set. In the sensitivity studies, we use a relatively short synthetic data set of 7143 profiles, which includes several sections selected from a 70000 profile data set to include different types of clouds (morphologically complex tenuous cloud, optically thick cloud, aerosol layer) and data from night times and day times. The advantage of using a short data set in sensitivity studies is that visual inspection is possible. Different data situations need to be included to ascertain layer detection with auto-adaptive capabilities. For this study, ICESat-2 type data were simulated based on GLAS data, using the process described in section (9). Data were passed through the SIPS coded processing chain, and the resultant NRB data from product ATL04 are used here. These data are referred to as "GLAS-based simulated ATL04 data (2017-Oct version)". These data are also used in testing of the SIPS code implementation, in comparison of CU code and SIPS code, and in Q/A development. These data constitute the last pre-launch data utilized in algorithm development.

10.2 Single-Density Runs Versus Double-Density Runs

The analysis of the 2016 state-of-the-art GLAS-based simulated data performed well using single-density runs with parameter combination (t8). Some of the previously analyzed data sets required double-density runs (running density twice, see section (3.10)), for instance, early GLAS-based simulated data and 2013 M-ATLAS Data (section (8)). In this section, we perform sensitivity

studies for single-density runs and double-density runs, to analyze the trade-off between the two options for the ATL04 GLAS-based data sets. To recall, the trade-off is as follows:

(a) Single-density runs require less computer time.

(b) The algorithm is computationally inexpensive (mostly linear algebra), hence fast. Therefore double-density runs can be afforded computationally.

(c) A double-density run allows to detect clouds layers of very different spatial characteristics - optically thick, but possibly spatially thin (but not necessarily spatially thin) cloud layers in runs 1. In the second run, the thick cloud layers are ignored and a larger kernel is used to facilitate aggregation of photon counts (or NRB values) over a larger region, which brings out tenuous cloud and atmospheric layers.

10.3 Results and Consequences for Algorithm Applications: Running Density Twice, $t56$, $t64$

In summary, the results of the sensitivity studies are as follows (see also section 3.6 on layer boundaries):

(1) The parameter combination, $t8$, which worked best for the 2016 state-of-the-art simulated GLAS-based ICESat-2 type data, renders ill-defined layer-tops and bottoms especially at day time (right part of the data set). Some false positives appear, especially around the layer boundaries. This indicates that the parameters that determine the threshold function do not match the characteristics of the data any more. The change in NRB value determination (ATL04) requires a new set of parameters.

(2) The parameter combination, $t54$, is the best result for a single-density run, as determined in the sensitivity study. The layer boundaries are much better defined than in the $t8$ run. However, the layer boundaries are still somewhat ragged for night-time data and sub-optimally defined for day-time data. As the sensitivity study shows, it is not possible to retain tenuous clouds, while suppressing false positives, using a single-density run.

(3) This necessitates using a double-density run, which allows to identify optically thick layers in the first run (using a smaller kernel and a very strict threshold function) and, in the second

density run, identify the tenuous clouds, atmospheric layers and most clouds during day-time conditions (using a larger kernel and a less strict threshold function). Notice that tenuous clouds (on the left) are now connected in the vertical direction, except for likely natural gaps, rain (?) falls out of the layer at a possible inversion (aerosols with clouds at the inversion height), but no false positives remain and the layer tops during day-time are smooth. Single clouds are retained during day-time conditions. $t56$ is the parameter combination used in most experiments in October/November 2017 (and deemed best for current state-of-the-art data characteristics in ATL04).

(4) Varying parameters around those of $t56$ and trouble-shooting remaining differences between CU code and SIPS code, we noticed that a smaller cluster size in run1 (200 rather than 300 pixels) retains all good characteristics of $t56$ and appears to slightly improve them. The cluster size of 200 also renders the algorithm more robust (in the sense that all small speckles are already filtered out and first-order cloud layers are more continuous). This is $t64$. Parameters are otherwise the same as in $t56$. Note this may be good to know in testing, going forward, as the CU declustering step and the SIPS declustering step employ similar, but not the exact same function.

In consequence, double-density runs will be used going forward for analysis of ICESat-2 simulated data and ATLAS atmosphere data after launch.

The ATL09 data product will include the density fields from run 1 and from run 2 (see Table 2d).

Details regarding interpretation of the individual sensitivity experiments are given after the figures.

10.4 Sensitivity Studies for Single-Density Runs

The following table shows the parameter combinations that were used for single-density runs:

Parameter Name	Sigma	Anisotropy	Cutoff	Down-sample	Minimum Cluster Size	Threshold Bias	Threshold Factor	Threshold Window	Quantile
t8*	3	10	1	1	600	6E+14	0.9	2	0.75
t32*	3	10	1	1	600	12E+14	0.9	2	0.75
t33*	3	10	1	1	600	10E+14	0.9	2	0.75
t34*	3	10	1	1	600	10E+14	0.9	3	0.75
t35	3	10	1	1	600	6E+14	0.9	3	0.75
t36*	3	10	1	1	600	10E+14	0.9	4	0.75
t37	3	10	1	1	600	6E+14	0.9	4	0.75
t38*	3	10	1	1	600	10E+14	1	2	0.75
t39	3	10	1	1	600	6E+14	1	2	0.75
t40	3	10	1	1	600	6E+14	1	2	0.9
t41	3	10	1	1	600	6E+14	1	2	0.85
t42	3	10	1	1	600	6E+14	0.9	2	0.9
t43	3	10	1	1	600	6E+14	0.9	2	0.85
t44*	3	10	1	1	600	10E+14	1	2	0.9
t45*	3	10	1	1	600	10E+14	1	2	0.85
t46*	3	10	1	1	600	10E+14	0.9	2	0.9
t47*	3	10	1	1	600	10E+14	0.9	2	0.85
t53*	3	10	1	1	600	10E+14	0.9	2	0.75
t54*	3	10	1	1	600	10E+14	0.9	2	0.8

Table 5: **Single Density Runs.** *Denotes an experiment for which results are shown in Figure 35.
(t8) Best parameter combination for analysis of 2016 state-of-the-art GLAS-based simulated ICESat-2 data.
(t54) Best parameter combination for a single density run for analysis of 2017-Oct GLAS-based ATL04 data.

For each run, the following plots are shown:

(a) Raw NRB Data - Valid Bins

(b) Kernel Matrix

(c) Density - Valid Bins

(d) Thresholds Along Track

(e) Density - Thresholded

(f) Density - Declustered

(g) Final Cloud Mask

The final mask in figure panel (g) is derived from the mask shown in panel (f) by application of the algorithm for cloud layers (3-bin rules and inclusion of loner bins).

Explanations of results in Figure 35

NOTE: Figure 35 is located in Appendix S (Sensitivity Studies with Figures, Section 10).

(1) **Starting from (t8).** As noted in the summary above, the parameter combination $t8$ renders ill-defined layer-tops and bottoms especially at day time (right part of the data set). Some false positives appear, especially around the layer boundaries. False positives are not limited to day-time conditions.

(2) **Fixing the threshold offset. (t32) with $T = 12E + 14$ and (t33) with $T = 10E + 14$.** The first thing to match is the base threshold, or threshold offset, T, to the new types of NRB data. The value used in (t32) works better and is kept for future runs.

(3) **Investigate the role of the segment length. (t34) with $L = 3$t, (t36) with $L = 4$ and (t53) with $L = 20$.** Here we look into the question: can the raggedy tops of layers, especially of optically thick layers, be smoothed by increasing the segment length? Note this is used in the determination of thresholds - same threshold function per segment, and the total length is $2L + 1$. (T8) uses $L = 2$, hence total length 5. Results for (t34) show that the raggedy edges remain but get a little wider, and even wider for (t36). The effect of the larger segment length on the threshold function is obvious for (t53), but the false positives do not disappear. This indicates that the raggedy edges and false positives cannot be smoothed out using threshold segment length. The visually ragged spots are much larger (wider) than the segment length. Keep $L = 2$.

(4) **Do we really need the threshold-sensitivity factor in the threshold function? (t38) with $t = 1$, also (t44) , (t45), (t46), (t47), (t53) and (t54).** It turns out that it is much harder to achieve a well-working threshold function without using a threshold-sensitivity factor. Experiment (t38) shows this. Other combinations of threshold function parameters were also tried, as documented in Table 5. use $t = 0.9$.

(5) **Setting quantile. (t44) , (t45), (t46), (t47), (t53) and (t54). Quantile 0.75, 0.8, 0.85, 0.9**.
Selected $Q = 0.8$. (t44) with $q = 0.9$ - quantile too high, $t = 1$ too high, clouds are missed.
(t45) with $q = 0.85$ - quantile too high, $t = 1$ too high, clouds are missed.
(t46) with $q = 0.9$ - quantile too high, $t = 0.9$, more clouds, but too many tenuous clouds are still missed. No rain falls out of the clouds.
(t47) with $q = 0.85$ - quantile too high, $t = 0.9$, more clouds, but too many tenuous clouds are still missed.
(t54) with $q = 0.8$ - best.

(6) **(t54) is the best parameter combination for a single-density run.** But it does not meet the two requirements (no false positives, no raggedy edges), but retain tenuous clouds, aerosol layers in day-time and night time, keep cloud structure (in left segment), keep rain falling out of the clouds.

This necessitates application of running density twice to meet the cloud detection criteria!

10.5 Sensitivity Studies for Double-Density Runs

The following table shows the parameter combinations that were used for double-density runs:

Parameter Set Name	Sigma	Anisotropy	Cutoff	Down-sample	Minimum Cluster Size	Threshold Bias	Threshold Factor	Threshold Window	Quantile
t48*	3	10	1	1	600	10E+14	1	2	0.9, 0.7
t49*	3	10	1	1	600	10E+14	1	2	0.95, 0.7
t50	3	10	1	1	600	10E+14	1	2	0.99, 0.8
t51	3	10, 20	1	1	600	10E+14	1	2	0.99, 0.8
t52	3	10, 20	1	1	300, 600	10E+14	1	2	0.99, 0.8
t55*	3	10, 20	1	1	300, 600	10E+14	1, 0.9	2	0.99, 0.8
t56*	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.99, 0.8
t57	3	10, 15	1	1	300, 600	10E+14	0.9, 1	2	0.99, 0.8
t58*	3	10, 20	1	1	400, 600	10E+14	0.9, 1	2	0.99, 0.8
t59*	3	10, 20	1	1	300, 600	12E+14	0.9, 1	2	0.99, 0.8
t60	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.95, 0.8
t61	3	10, 15	1	1	300, 600	10E+14	0.9, 1	2	0.95, 0.8
t62	3	10, 15	1	1	300, 600	12E+14	0.9, 1	2	0.95, 0.8
t63*	3	10, 30	1	1	300, 600	10E+14	0.9, 1	2	0.99, 0.8
t64*	3	10, 20	1	1	200, 600	10E+14	0.9, 1	2	0.99, 0.8
t65*	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.97, 0.8
t66*	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.98, 0.8
t67*	3	10, 20	1	1	150, 600	10E+14	0.9, 1	2	0.99, 0.8

Table 6: **Double Density Runs.** *Denotes an experiment for which results are shown in Figure 36. (t56) Best parameter combination for a double density run for analysis of 2017-Oct GLAS-based ATL04 data. (t64) Alternative best parameter combination for a double density run for analysis of 2017-Oct GLAS-based ATL04 data (declustering size 200 in run 1, otherwise same as (t56)).

For each experiment, the following plots are shown (left column, density run 1 (labeled 0), right column, density run 2 (labeled 1, py convention)):

(a) Raw NRB Data - Valid Bins

(b) Kernel Matrix

(c) Density - Valid Bins

(d) Thresholds Along Track

(e) Density - Thresholded

(f) Density - Declustered

(g) left: Final Cloud Mask (Binary) , right: Final Cloud Mask over Density 1

Explanations of results in Figure 36

NOTE: Figure 36 is located in Appendix S (Sensitivity Studies with Figures, Section 10).

(1) Balance the quantiles between density run1 and density run2 **(t48)-(t50).** (t48), (t49) Not good. Keep false positives, make holes in the tenuous cloud layers. Threshold function is bad, quantiles too low. (t50) reaches acceptable quantiles.

(2) Different anisotropies for density run1 and density run2. Also different min cluster sizes. (t56) is the best parameter combination. Notably, (t55), where the threshold sensitivity factors are switched, is a lot worse. (t56) identifies just enough clouds in pass 1, to bring out the more tenuous layers in the second density run. Layers are smooth on top, with rain falling out of the bottom. Note that the small clouds (day-time section) are actually clouds, visible in the density fields. (t63) with anisotropy (10,30) makes the tenuous clouds a little too wide. Use (10,20) for anisotropies.

(3) Different cluster sizes. (t64) with min cluster size 200, otherwise parameters same as (t56). Mean confidence for (t64) is higher (0.801) than for (t56), making this a slightly better parameter combination than (t56) (conf 0.79 on average). (t67) with min cluster size 150: results are not much different from those of (t64). Because more day time data need to be analyzed and because of robustness, keep (t64) as the best run. See the section on "Testing" - the SIPS code uses a somewhat different implementation of the declustering routine, this still needs to be tested. Hence opt for min cluster size of 200.

(4) **Checking intermediate quantiles, (t65) with** $q = 0.97, 0.8$**, (t66) with** $q = 0.98, 0.8$**.** The quantiles of $q = 0.99, 0.8$ indeed yield the best results, evaluated by the same criteria as before. Hence (t56) and (t64) are best. Testing of the declustering routines is still to be done.

In summary, very good results are obtained using double-density runs with parameters (t56)!

11 Quality Assessment

11.1 Summary

A confidence measure will be defined for each layer. The idea of the confidence measure is to provide a numerical value for the confidence in layer detection, that works for optically thick clouds, tenuous clouds and aerosols. The particular strength of the DDA lies in its ability to detect tenuous cloud layers and aerosols in night-time and day-time conditions. Specifically, the confidence measure needs to indicate when tenuous clouds are detected with high confidence.

An algorithm that quantifies confidence as a numerical value is introduced for quality assessment. Mathematical Q/A algorithm description, Q/A plots and applications are included as subsection.

The algorithms yields a vector of values

`layer_conf_dens(layerno)`

where *layerno* is the maximal number of cloud layers (currently 10). The layer confidence generally has a value between 0 and 1, but can assume values outside of this range.

The algorithm to be used is termed "Half-gap confidence". In addition, the half-gap confidence is compared to an alternative "3-bin confidence". "Half-gap confidence" is a better measure than "3-bin confidence".

11.2 For Data Dictionary

`layer_conf_dens(layerno)`

The measure *layer confidence*, or *layer confidence (from density-dimension algorithm)* or *layer confidence (from DDA)*, calculated for each detected cloud layer, quantifies the confidence of detection of a given layer from the NRB values, using the DDA. Layer confidence (DDA) is a vector with *layerno* values, one value per detected layer (NaN for vector entries for which no layer was actually detected). Currently, $layerno = 10$, the maximal number of detectable layers is 10. Layer confidence is

normalized such that values generally fall between zero and 1 and assume a good spread across the range [0,1] for most atmospheric layers. However, layer confidence can assume values outside of this range (see subsection (11.5) "On the range of half-gap layer confidence values" below).

11.3 Q/A Algorithm

The concept of this confidence measure is to utilize the ratio of average density inside a cloud to average density in the region surrounding the cloud. A confidence value is given for each layer in ATL09.

We use the convention: n_profile_valid = $n_{pv} = 467$ for the number of valid bins in each profile. A bin n takes an integer value between 1 and 467, where the profile top = 1 and profile bottom = 467. Consider cloud layer s in column (profile) j: $[n_{s,j}^{top}, n_{s,j}^{bot}]$.

Definition: The **jth cloud layer thickness** is defined as:

$$cloud_thick_j = ct_j = |n_{s,j}^{top} - n_{s,j}^{bot}| + 1$$

Definition: The **half-gap distance** to the next cloud layer rounded to the nearest integer is defined as follows:

$$dist_above = max\left\{3, round\left(\frac{|n_{s,j}^{top} - n_{s-1,j}^{bot}| - 1}{2}\right)\right\}$$

If s is the top cloud layer, then take the half-gap to the top of the profile (bin 1):

$$dist_above = max\left\{3, round\left(\frac{n_{s,j}^{top} - 1}{2}\right)\right\}$$

Similarly,

$$dist_below = max\left\{3, round\left(\frac{|n_{s,j}^{bot} - n_{s+1,j}^{top}| - 1}{2}\right)\right\}$$

If s is the lowest cloud layer, then take the half-gap to the bottom of the profile (bin n_{pv}):

$$dist_below = max\left\{3, round\left(\frac{n_{pv} - n_{s,j}^{bot}}{2}\right)\right\}$$

The number 3 in the maximum function arises from the definition of a cloud-layer, which requires a minimum of 3 consecutive bins of the same type to define a layer between cloud and non-cloud (see section (3.7) in ATBD Atmos Part II).

Definition: **Half-gap confidence ($k_{s,j}$)**: The confidence measure for cloud layer s in the jth column ($k_{s,j}$), termed *layer confidence(dda)*, or more specifically, *half-gap layer confidence*,

`layer_conf_dda(layerno)`

where *layerno* is the maximal number of cloud layers (currently 10), is determined by calculating the following quantities.

The average density in the half-gap above and the half-gap below the cloud is given by

$$A = A_{s,j} = \frac{\left[\sum_{i=1}^{dist_above} f_d^{norm}\left(n_{s,j}^{top} - i\right)\right] + \left[\sum_{i=1}^{dist_below} f_d^{norm}\left(n_{s,j}^{bot} + i\right)\right]}{dist_above + dist_below}$$

The average density in the cloud is given by

$$B = B_{s,j} = \frac{\left[\sum_{i=n_{s,j}^{top}}^{n_{s,j}^{bot}} f_d^{norm}(i)\right]}{|n_{s,j}^{top} - n_{s,j}^{bot}| + 1}$$

The confidence is given by

$$k_{s,j} = 1 - \frac{A_{s,j}}{B_{s,j}}$$

Since B is for a cloud and A is for a non-cloud, we always have $A < B$, and generally $0 < \frac{A}{B} < 1$ and $||k_{s,j}|| < 1$. Confidence close to 1 – good, close to zero – bad. However, layer confidence can assume values outside of this range (see section (15.1) "On the range of half-gap layer confidence values").

The confidence value remains meaningful, if it is outside of the range [0,1]. If confidence is larger than one, then the cloud determination is even better.

Definition: **3-bin confidence ($k_{s,j}^3$)**: This definition of confidence compares the average density inside the cloud to the average density of the three adjacent bins outside the cloud. Let *dist_below* =

$dist_above = 3$ and use the same formulas for A and B to attain $A^3_{s,j}$ and $B^3_{s,j}$. The 3-bin confidence is then given by

$$k^3_{s,j} = 1 - \frac{A^3_{s,j}}{B^3_{s,j}}$$

11.4 Q/A Plots

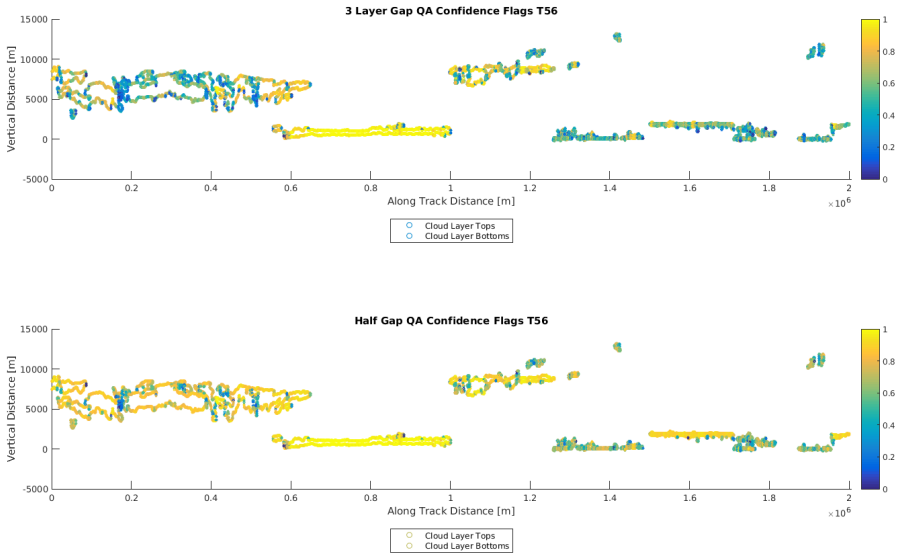


Figure 37. Confidence measures 3-bin confidence (top) and half-gap confidence (bottom). Applied to 7000+ (7143) profile synthetic data set representing different cloud types and night-time/ day-time transition, 2017-Nov version of GLAS-based simulated ATL04 data. Confidence value shown for layer top and layer bottom applies to entire layer (t56). See Table 6 for parameters.

As seen in Figure 37, both confidence measures work well for optically thick cloud layers, such as the layer between 0.6 and 1 ($\times 10^6$ m) along track distance. However, for complex and tenuous layers during night-time, such as the layer between 0 and 0.6 ($\times 10^6$ m) along track distance, and for any layers during day-time (1.2 to 2 $\times 10^6$ m along track distance), half-gap confidence is a better measure than 3-bin confidence. The measure 3-bin confidence works well only in situations where the density gradient is high at the cloud boundary. The detection of tenuous layers and layers during day-time relies on the difference of aggregated values over a larger region and this is reflected in the half-gap confidence. Therefore, half-gap confidence will be reported as a Q/A

measure on the product ATL09. Space for a second Q/A measure is reserved for future use.

11.5 Applications

The half-gap confidence measure is now applied to evaluate those results from the sensitivity studies that were deemed best in their category, (a) t8, (b) t54, (c) t56, (d) t64 (see Figure 38). Assume, for the sake of a thought experiment, that data characteristics changed unknown to the user while parameters of (t8) were being applied. Now the confidence measure tells us that some clouds have a low confidence, and these are the false positives (apparently detected layers that are not actually clouds). Similarly, the confidence measure allows to decide that the double-density runs (t56) and (t64) yield better results than the single-density run (t54), by visual inspection of the regions of lower confidence.

In the clouds of complex morphology, such as the layer between 0 and 0.6 ($\times 10^6$ m) along track distance, some areas of low confidence occur because the half-distance is very short internal to the clouds. This is actually correct, because in these locations several layers are identified, whereas in neighboring regions only one layer is found.

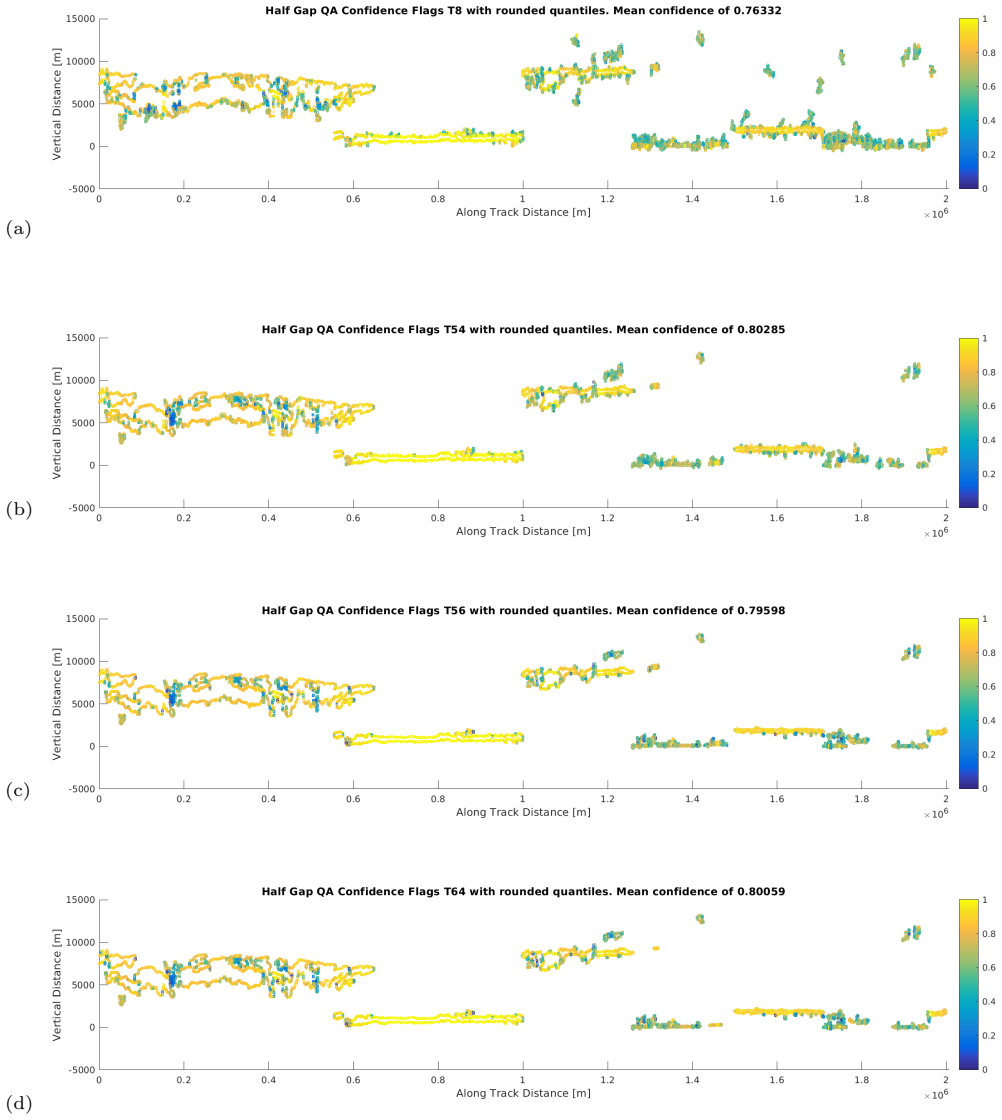


Figure 38. Half-gap confidence of determination of atmospheric layer boundaries. Applied to 7000+ (7143) profile synthetic data set representing different cloud types and night-time/ day-time transition, 2017-Nov version of GLAS-based simulated ATL04 data. Confidence value shown for layer top and layer bottom applies to entire layer. (a) t8, (b) t54, (c) t56, (d) t64. See Tables 5, 6 and 7 for parameters.

12 Testing

Code testing is part of the process of implementation of the algorithm, developed at CU Boulder, by the SIPS. In this section, results of comparisons of the CU code and the SIPS code are described, and criteria for acceptance of the SIPS code derived. For comparison output data sets and associated figures are derived for each algorithm step (presented in subsection on "Testing Steps"). At time of writing, testing is still in progress.

Interim results are as follows:

(1) Density matrix matches perfectly.

(2) Density field calculation for density run 1 matches well

(3) Threshold function: Matching threshold functions is a complex topic. First implementation of the threshold function had to be checked. However, differences in the threshold function values can arise when different library functions are used for quantile calculation. This is further examined in the section on "Quantile Calculation – Changes" (12.2). After this divergence was discovered, we wrote a piece a code for the CU algorithm that calculates quantiles the same way as the SIPS FORTRAN code. Now, thresholds match very well (see figure panel "threshold quantile with rounding, Threshold Relative Error pass 0 (t56)" in Figure 39-4). Comparison with figure panel "threshold quantile with linear interpolation, Threshold Relative Error pass 0 (t56)" in Figure 39-2 demonstrates that the quantile calculation was a major source of error. However, errors propagate from the declustering step onwards.

(4) Decluster algorithm. Here significant differences exist between CU code and SIPS code (see figure panel "threshold quantile with rounding, Decluster Mask Discrepancy pass 0 (t56)" in Figure 39-4). The ASAS decluster mask shows that large sections of optically thick clouds are lost in the current code version.

(5) The number of nrb valid bins per profile appears to fluctuate in the input ATL04 data. This would lead to an error that propagates through the end of run 1 and into run 2 where large differences are noted.

Code testing also motivated the following small CU algorithm changes:

(1) Implementation of masking of run 1 cloud regions before calculation of run 2 densities:

(1a) NaN handling: In evaluating the kernel, any bins neighboring the center bin that have values contribute to the sum. Alternative is to apply the kernel only if all neighboring bins have values. The alternative would lead to more data loss.

(1b) Inside the mask, values are replaced by zeroes in code version v110.0. This is suboptimal. We have to explore how to best fill the mask region.

(2) Change of quantile calculation (see section on "Quantile Calculation – Changes" (12.2)): (a) mquantiles default, (b) linear interpolation, (c) rounding [rounding is now the accepted method for quantile calculation].

12.1 Testing Steps

For comparison output data sets and associated figures are derived for each algorithm step. Here we present figures showing results of the CU DDA runs and the SIPS/ASAS DDA runs for each algorithm step, and an additional comparison plot per step. When sufficient matching is achieved, we move to the next step. The work presented here is carried out in collaboration between the CU group and the SIPS group; all plots are created by the CU group. Data sets used are the GLAS-based simulated ATL04 data sets, as before (7143 profiles synthetic data set).

In the following, we present 4 series of figures:

(1) (t54) single-density run, linear interpolation for quantile determination

(2) (t56) double-density run, linear interpolation for quantile determination

(3) (t54) single-density run, rounding for quantile determination

(4) (t56) double-density run, rounding for quantile determination

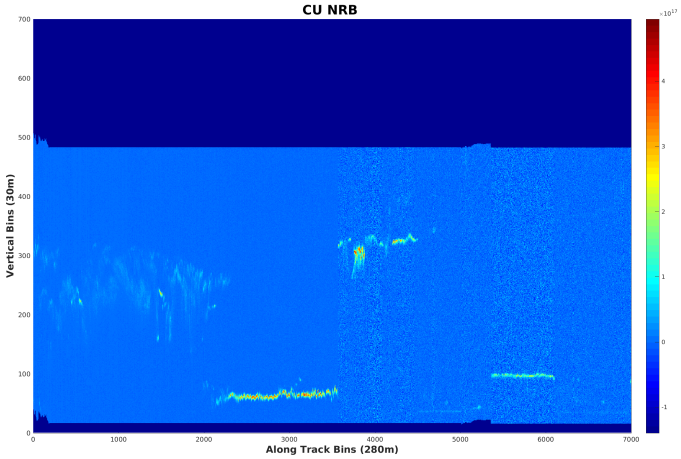
Percentage of bins with mask differences between CU Code and SIPS code is really low (1%) but number of profiles with incorrectly identified layers is high and needs more work.

Results: Number of profiles with incorrectly identified layers: (total profiles: 7143).

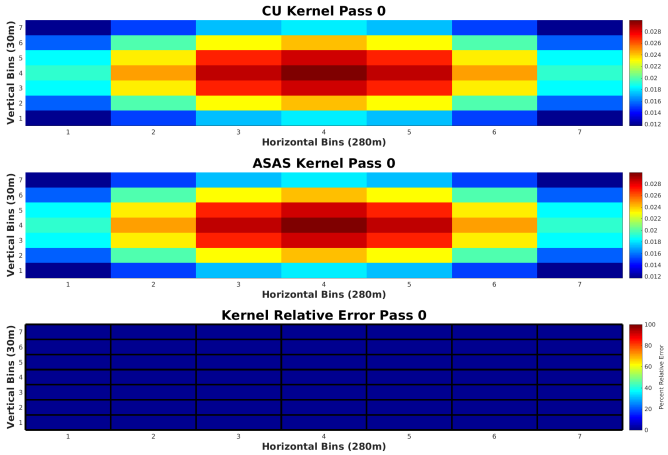
T_54_LinearInterp: 2003 incorrect profiles → 28.614% T_56_LinearInterp: 2354 incorrect profiles → 33.629%

T_54_Rounding: 2068 incorrect profiles → 29.543% T_56_Rounding: 2386 incorrect profiles → 34.086%

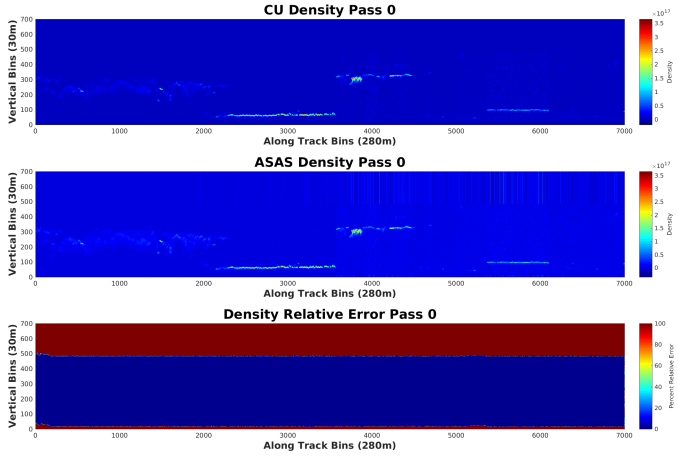
Threshold Quantile with Linear Interpolation. 11.15.2017 v9 T_54



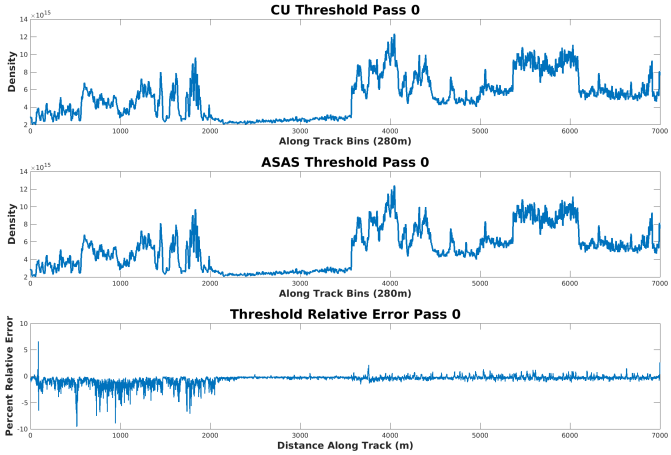
Threshold Quantile with Linear Interpolation. 11.15.2017 v9 T_54



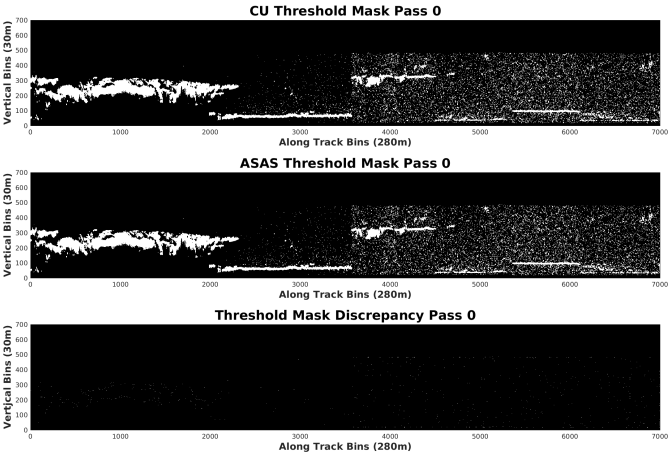
Threshold Quantile with Linear Interpolation. 11.15.2017 v9 T_54



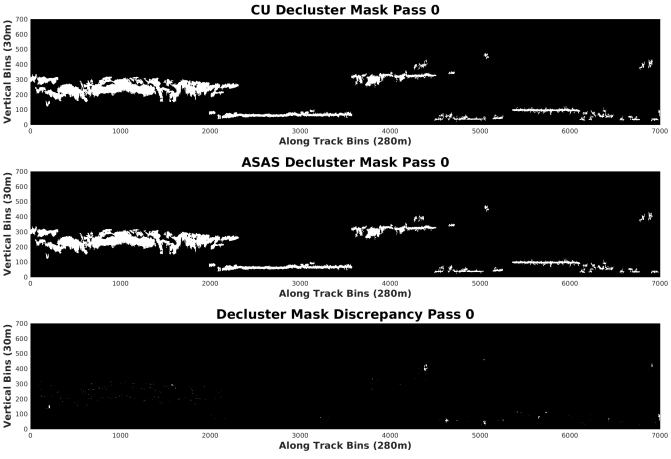
Threshold Quantile with Linear Interpolation. 11.15.2017 v9 T_54



Threshold Quantile with Linear Interpolation. 11.15.2017 v9 T_54



Threshold Quantile with Linear Interpolation. 11.15.2017 v9 T_54



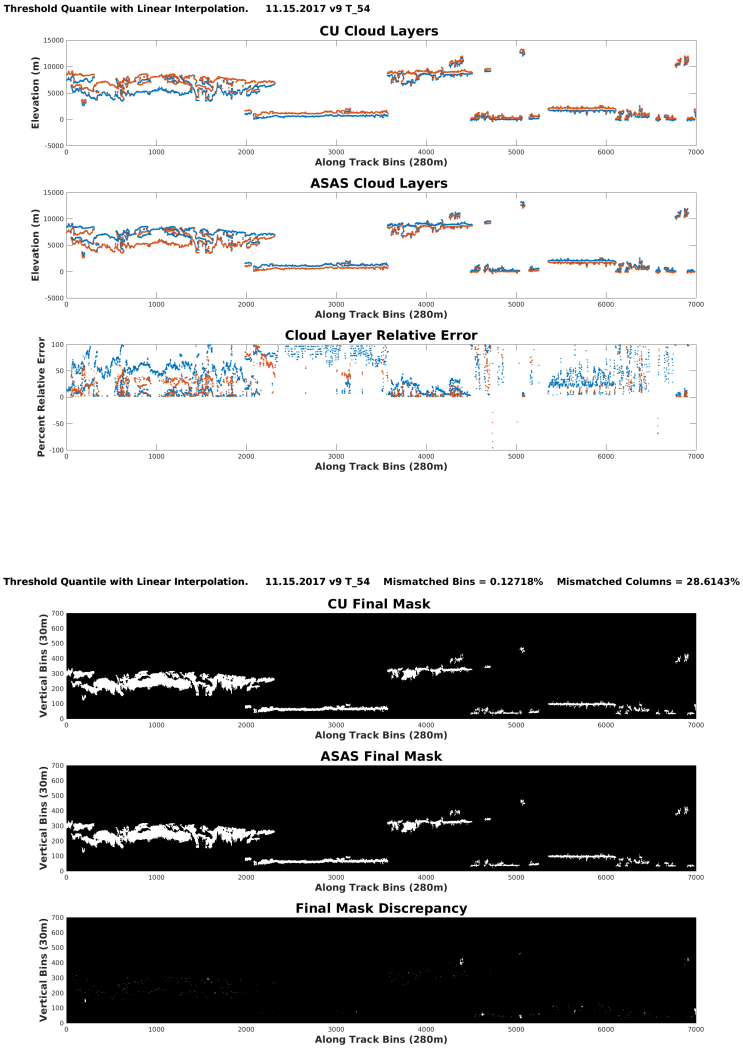
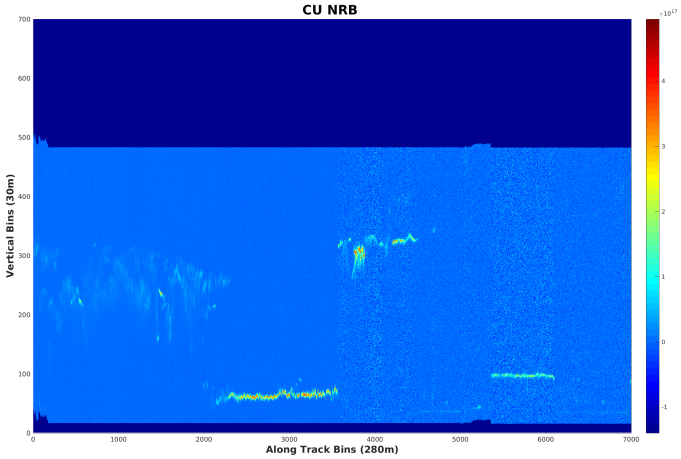
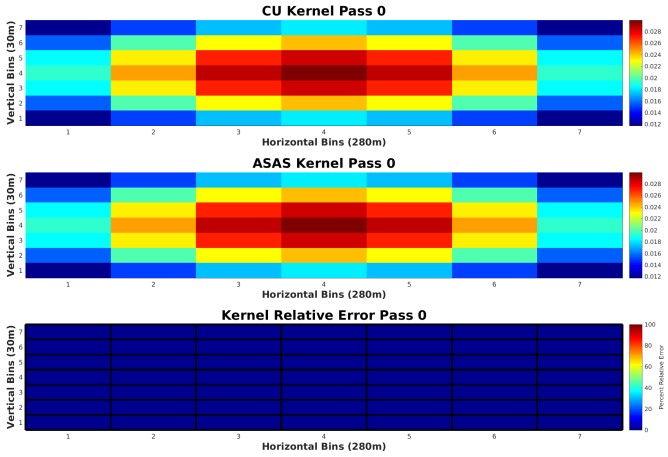


Figure 39-1. Testing: (t54) single density run, linear interpolation for quantile determination

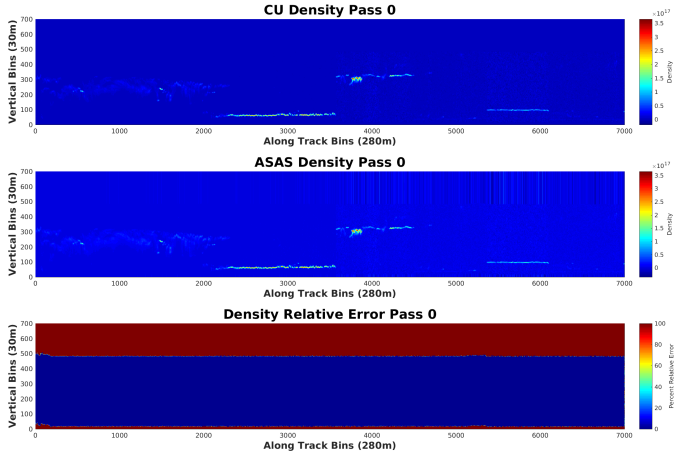
Threshold Quantile with Linear Interpolation. 11.15.2017 v9 T_56



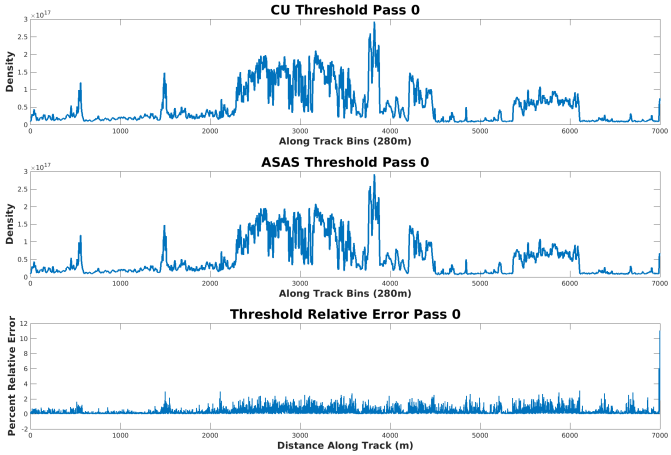
Threshold Quantile with Linear Interpolation. 11.15.2017 v9 T_56



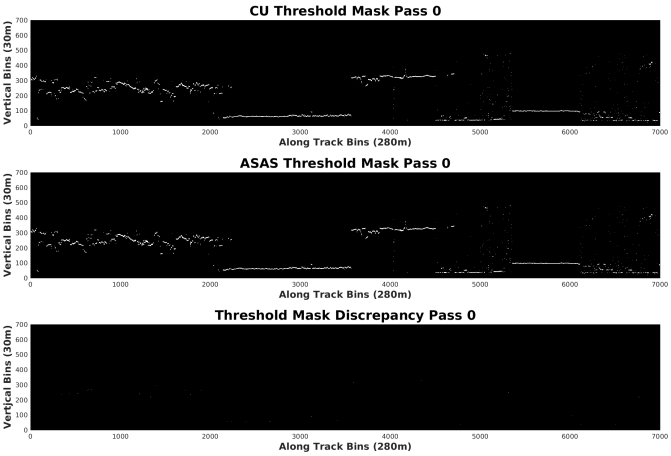
Threshold Quantile with Linear Interpolation. 11.15.2017 v9 T_56



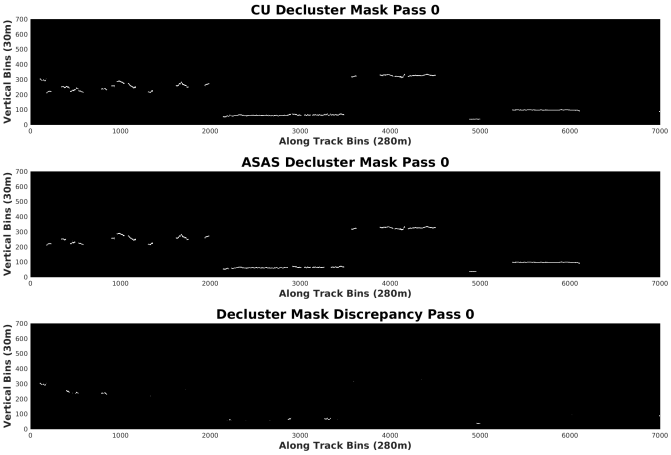
Threshold Quantile with Linear Interpolation. 11.15.2017 v9 T_56



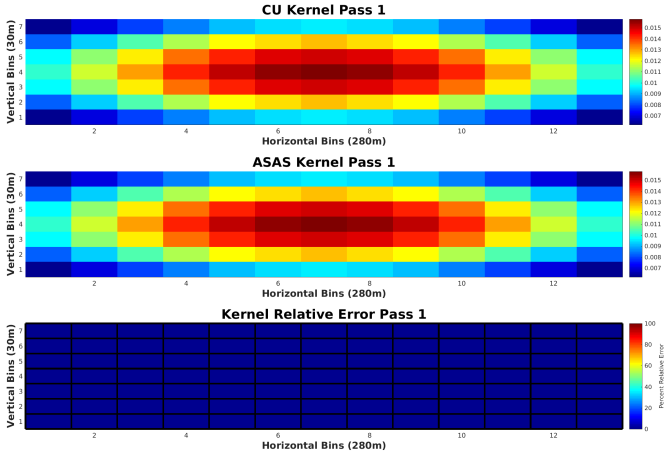
Threshold Quantile with Linear Interpolation. 11.15.2017 v9 T_56



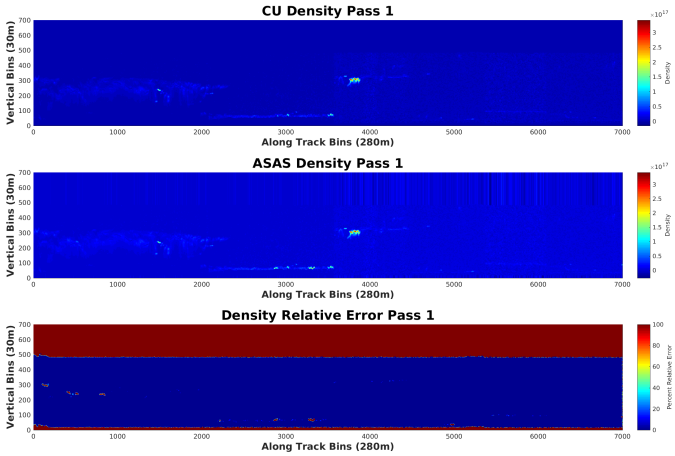
Threshold Quantile with Linear Interpolation. 11.15.2017 v9 T_56



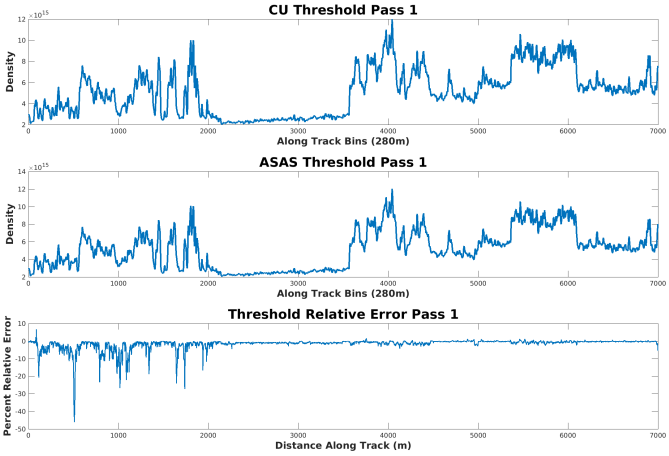
Threshold Quantile with Linear Interpolation. 11.15.2017 v9 T_56



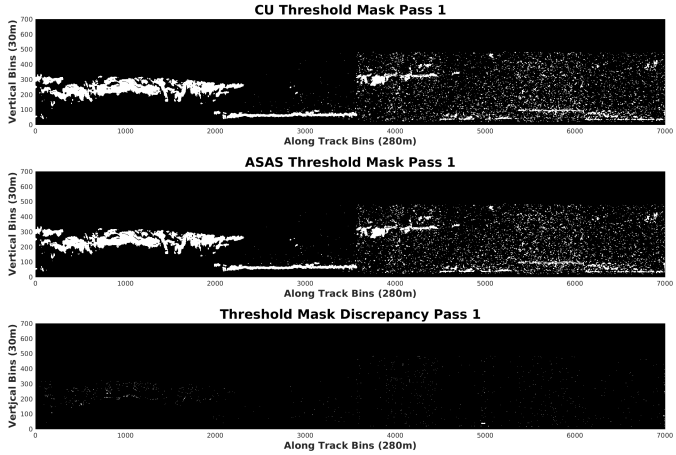
Threshold Quantile with Linear Interpolation. 11.15.2017 v9 T_56



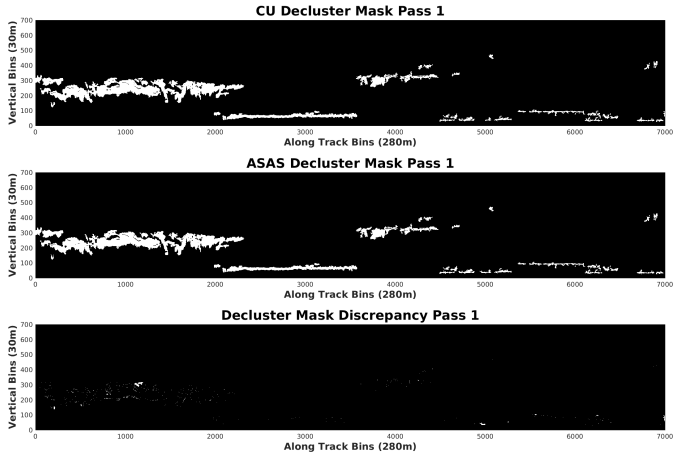
Threshold Quantile with Linear Interpolation. 11.15.2017 v9 T_56



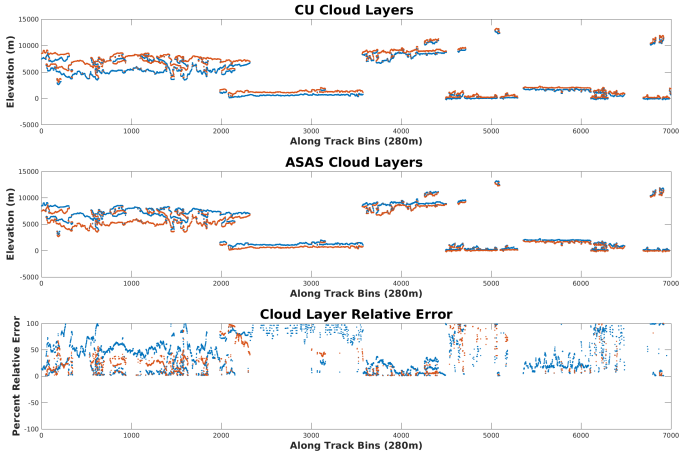
Threshold Quantile with Linear Interpolation. 11.15.2017 v9 T_56



Threshold Quantile with Linear Interpolation. 11.15.2017 v9 T_56



Threshold Quantile with Linear Interpolation. 11.15.2017 v9 T_56



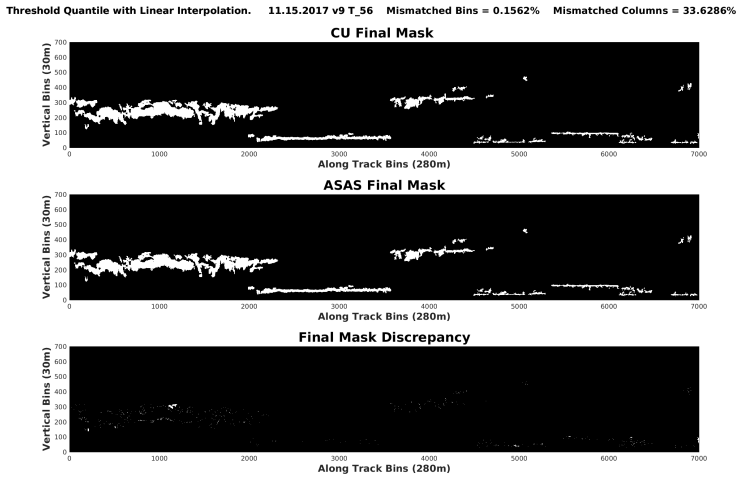
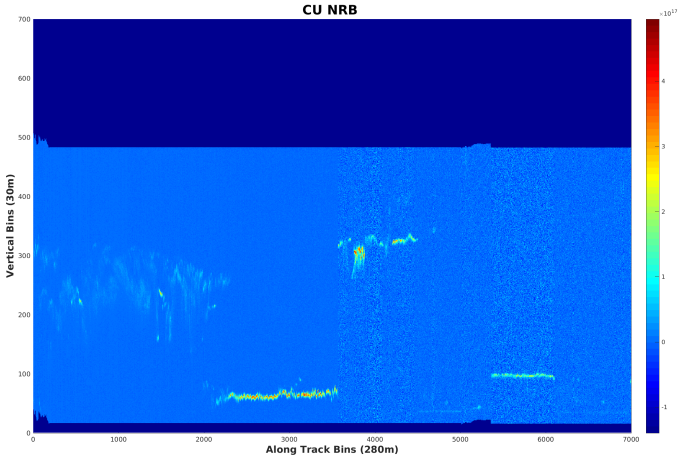
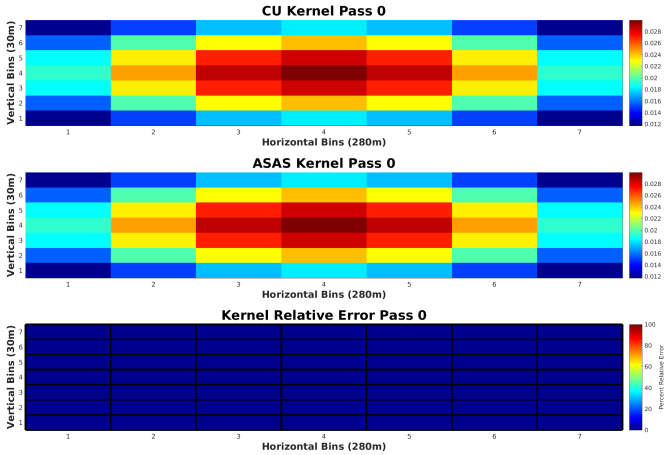


Figure 39-2. Testing: (t56) double density run, linear interpolation for quantile determination

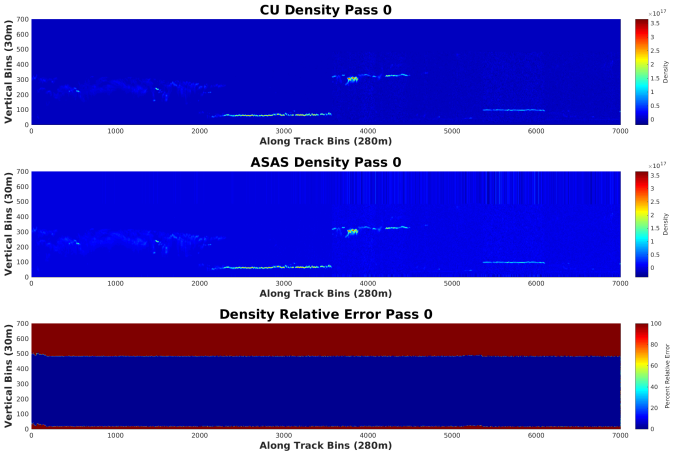
Threshold Quantile with Rounding. 11.15.2017 v10 T_54



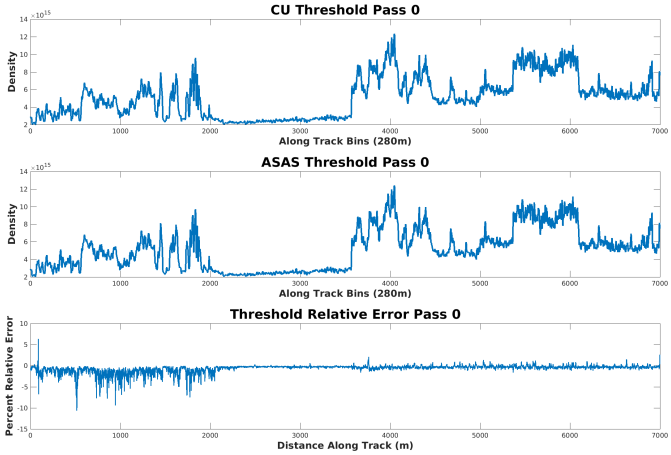
Threshold Quantile with Rounding. 11.15.2017 v10 T_54



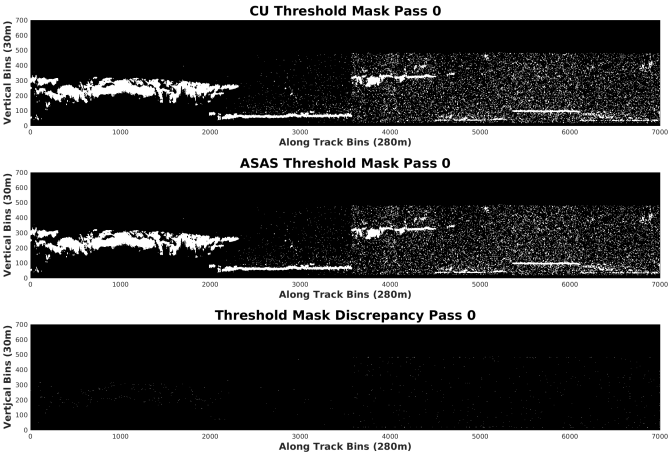
Threshold Quantile with Rounding. 11.15.2017 v10 T_54



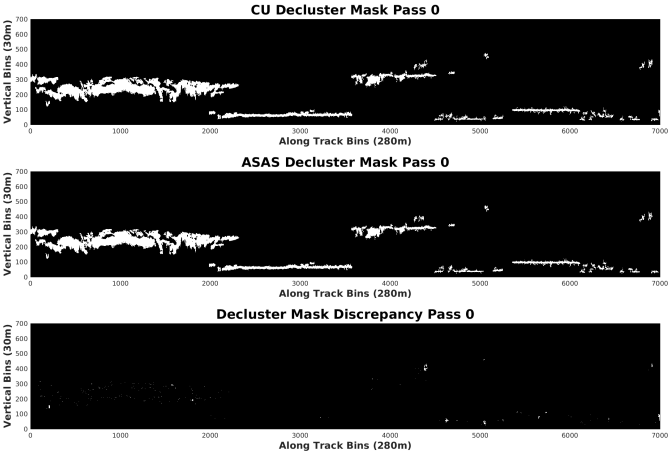
Threshold Quantile with Rounding. 11.15.2017 v10 T_54



Threshold Quantile with Rounding. 11.15.2017 v10 T_54



Threshold Quantile with Rounding. 11.15.2017 v10 T_54



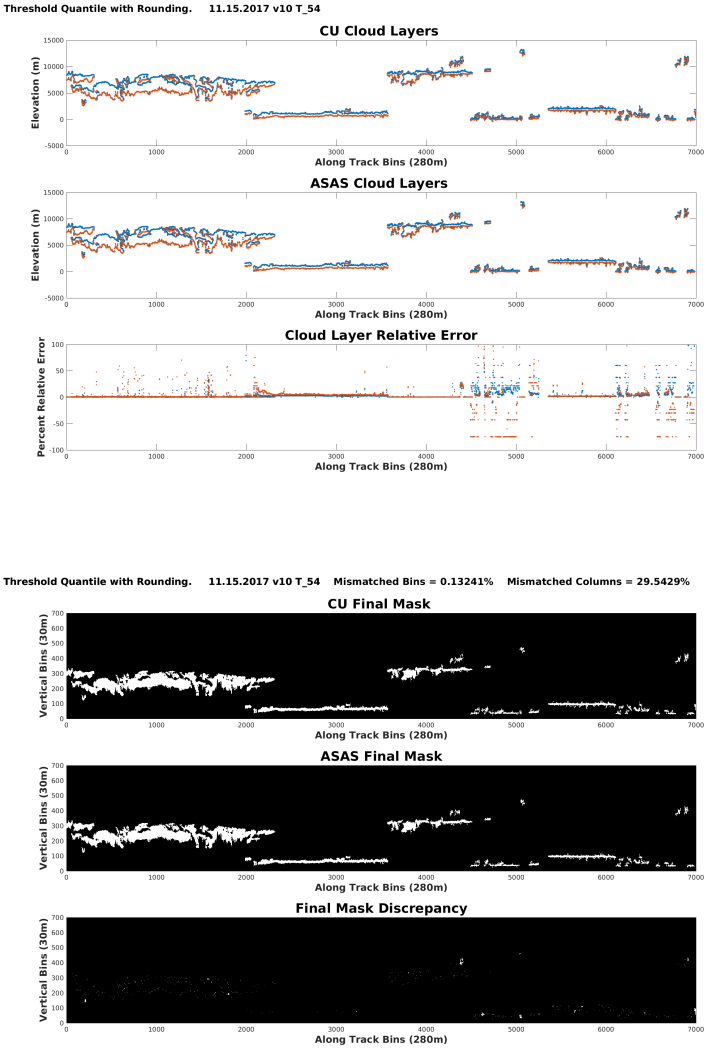
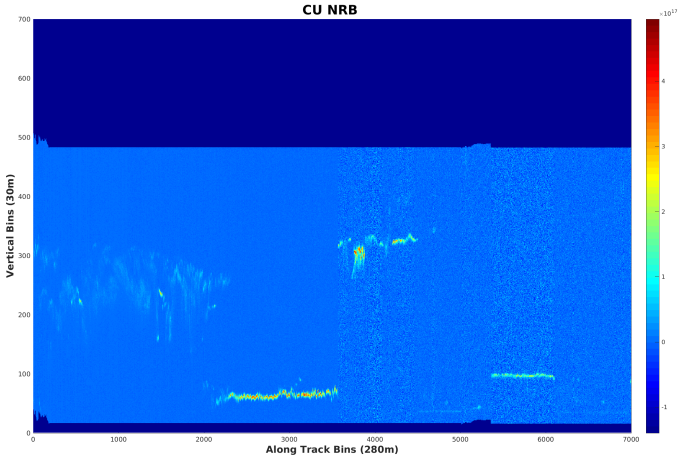
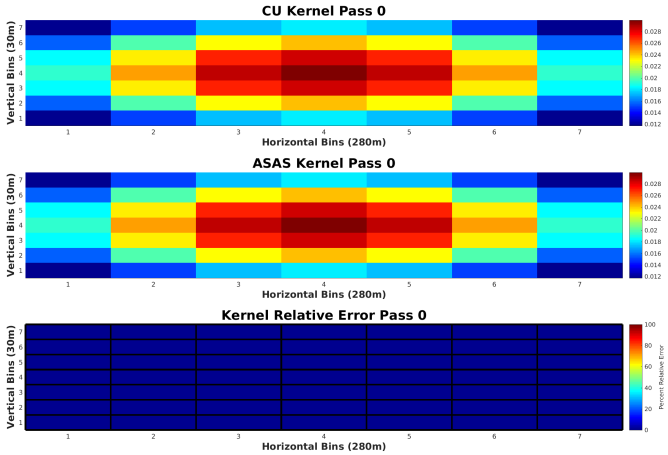


Figure 39-3. Testing: (t54) single density run, rounding for quantile determination

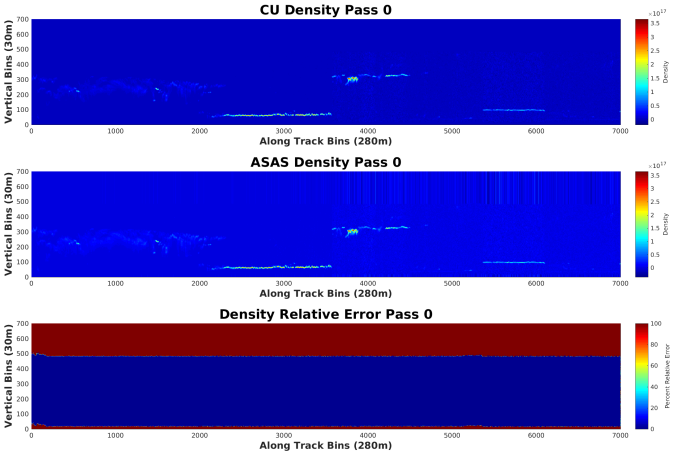
Threshold Quantile with Rounding. 11.15.2017 v10 T_56



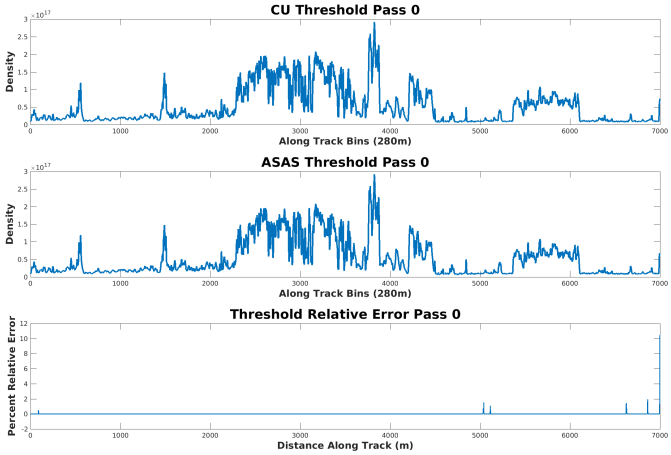
Threshold Quantile with Rounding. 11.15.2017 v10 T_56



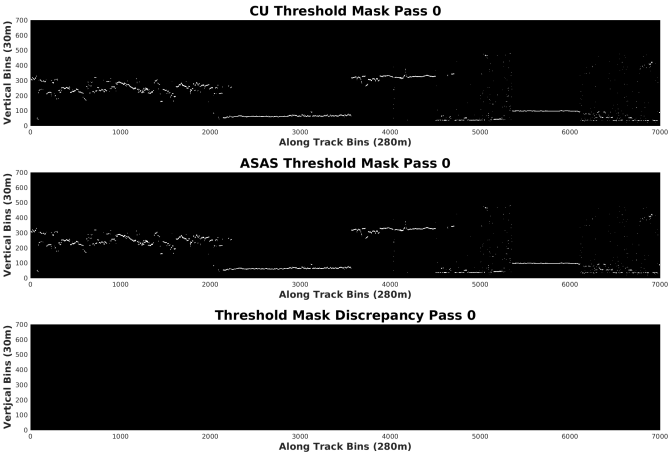
Threshold Quantile with Rounding. 11.15.2017 v10 T_56



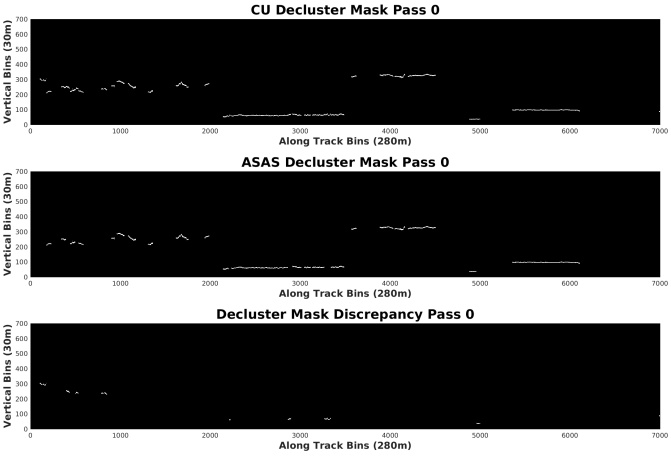
Threshold Quantile with Rounding. 11.15.2017 v10 T_56



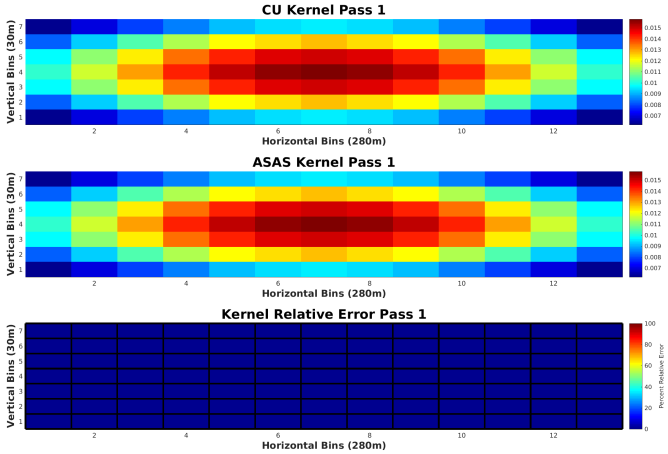
Threshold Quantile with Rounding. 11.15.2017 v10 T_56



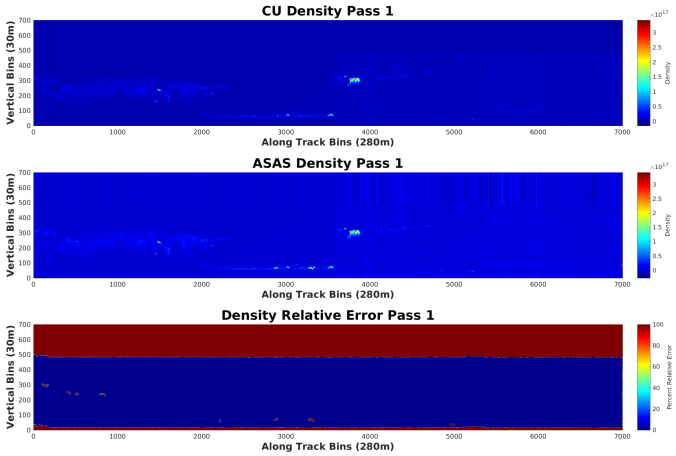
Threshold Quantile with Rounding. 11.15.2017 v10 T_56



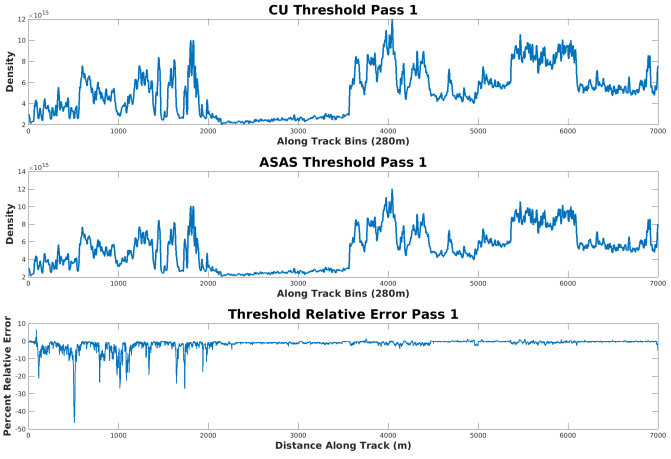
Threshold Quantile with Rounding. 11.15.2017 v10 T_56



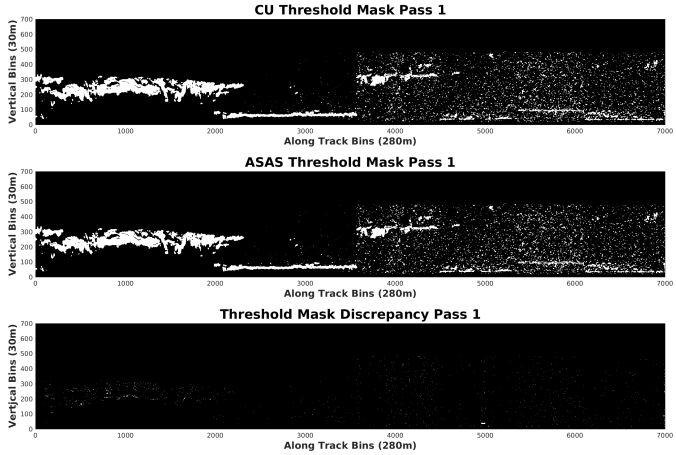
Threshold Quantile with Rounding. 11.15.2017 v10 T_56



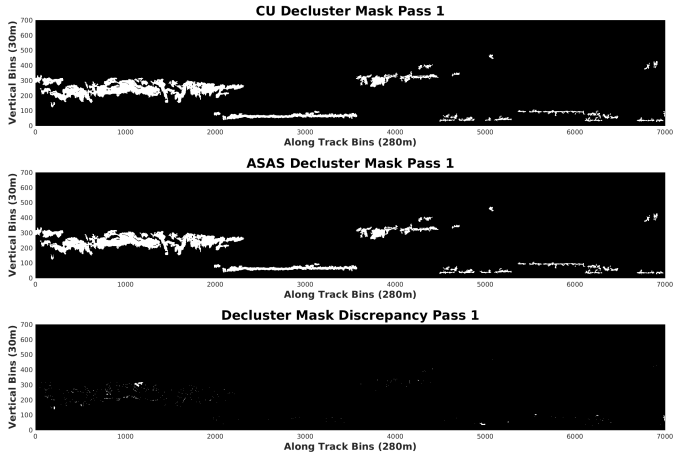
Threshold Quantile with Rounding. 11.15.2017 v10 T_56



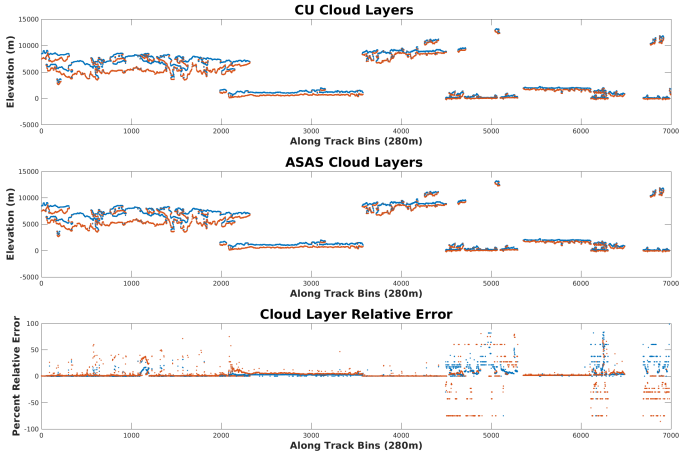
Threshold Quantile with Rounding. 11.15.2017 v10 T_56



Threshold Quantile with Rounding. 11.15.2017 v10 T_56



Threshold Quantile with Rounding. 11.15.2017 v10 T_56



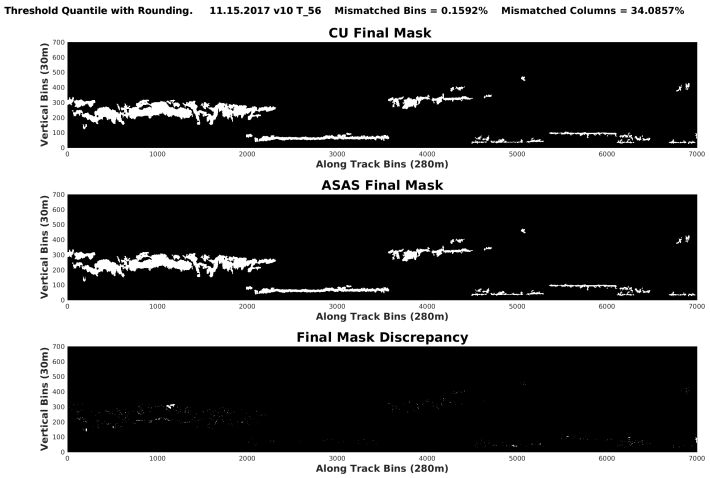


Figure 39-4. Testing: (t56) double density run, rounding for quantile determination

12.2 Quantile Calculation – Changes

Testing the implementation of the threshold function led to implementation of a different algorithm for calculation of quantiles in the CU algorithm. Rather than using the scipy library function *mquantiles* (as in v103.0 - v109.0), in v110.0 we use a directly coded so-called "rounding algorithm". The term "Rounding" refers to a discrete association function for quantiles, which follows the definition of a quantile in its simple form. The algorithm is described in section (3.3.7).

13 First Layer Classifications: Surface and Blowing Snow

13.1 Motivation for First Classification Algorithms

Hitherto, the goal of algorithm development for ATL09 data products has been identification of atmospheric layers, including optically thick and tenuous clouds, aerosols and blowing snow and of the surface in ICESat-2 atmospheric data (ATL-04 data). All these layer types can be identified in the data, as analyses and case studies in the previous sections have indicated.

Results are provided as outputs on the product ATL09. For each layer, bottom and top are calculated and included as outputs on ATL09. The Earth surface (ground) is usually the lowest of the identified layers (maximally 10), unless the surface is obscured by optically thick clouds or other atmospheric layers that attenuate the remaining signal. However, the type of the atmospheric layer or surface is not classified.

Studies of atmospheric processes may require classification of the layer type. As the first example of a classification, the surface, which is usually a strong layer at the bottom of the data matrix, is identified and moved to a separate data product field (surface_h_dens). Ongoing research on blowing snow motivates the classification of a blowing snow layer.

Note (2018-Dec-19): It should be noted that the algorithms for these classifications have been derived using simulated data based on CALIOP data and testing for ICESat-2 data needs to be carried out after algorithm implementation.

Note (2020-Jan-16): The current version (v11.0) of this ATBD Atmosphere Part II reflects algorithm updates for ICESat-2 data. The algorithms for ground height determination and ground flag determination are described in section 18. The original code version from v9.0 is marked as "CALIOP Test Version" in this document. The description of the CALIOP Test version is retained in v11.0 for code-trouble shooting reasons, while the new algorithm is being implemented.

Blowing snow and ground surface

Motivated by the work on blowing snow and its role in the cryospheric and climatic system, reported by Palm et al. [2017b, 2018c,b], an algorithm component that detects blowing snow and classifies it as such is developed and included in the ATBD. The identification of blowing snow requires that ground is classified as such.

Note that using the algorithm described so far ground and blowing snow will be identified as a single layer, following the rule that if there is a gap of less than three bins wide between two layers, then the layers will be joined into a single layer. Because blowing snow, if present, starts directly above the ground, 90m, or three bins, cannot always be expected.

Theory

Criteria for identification of blowing snow are the following:

1. Blowing snow starts immediately above ground.
2. Blowing snow height cannot exceed 500m.
3. Windspeed must be greater than 5m/s for blowing snow to occur.
4. Blowing snow has a higher density than surrounding atmosphere.

Ground Surface and Blowing Snow Algorithm Components (CALIOP Test Version)

The blowing-snow algorithm has two components: (1) identification of ground surface using density mask1, and (2) identification of the blowing snow layer using the field of density2.

The blowing snow layer itself is determined using density2 (the density field from the second density run), the ground surface is found using mask1 (the mask from density1).

The original version of the algorithm was developed using ICESat-2 atmospheric data, simulated from CALIOP data (from the CALIPSO Mission). In this version (V11.0, 2020-Jan-14), algorithm and algorithm-specific parameters are updated using current analysis (v2.0) of post-launch ICESat-2 data.

13.2 Algorithm for Determination of Ground Surface from Atmospheric Data (CALIOP Test Version)

For v11.0 algorithm, see section 18.

The idea of the algorithm is to identify the top of the lowest atmospheric layer (in the mask1 from density run1) and check whether this layer is the ground surface:

Top of lowest layer identified in density1mask (mask from density-run 1).

- Start by searching the region 10 to 24 bins above the NaNs.

Parameters: Note that the bin numbers 10, 24 may be specific to the CALIOP-based data simulation here and not correct for ICESat-2 data. The NaNs (Not-a-Number) relate to the DEM in the ATL09 code - see section *(13.4) Implementation Notes*.

For CALIOP-data based simulated data: Identify ground bin as the top bin of the lowest layer (within the index range [10,24] above the NaNs), i.e. the highest bin that is inside the layer. This algorithm piece does not require running the layer-identification first. The layer identification uses the combined mask (from density1 and density2), the ground determination uses only the mask from density1.

The ground-finding method finds the top and bottom of the lowest "atmospheric" layer within the 10 to 24 bin density1 mask, using a counting method, as follows:

1. Set a flag $incloud = FALSE$.
2. Count up from bottom. As soon as in mask, set $incloud = TRUE$. Register index i_{bot} for bottom of ground layer.
3. Count up, switch back to $incloud = FALSE$ for NaNs in the mask (to ascertain that the incloud flag is set back to FALSE for the next column).
4. Register $i_{top} - 1$ as the bin index for top of the layer.
5. The ground height is the height of bin $i_{top} - 1$.

Note. When implementing a counting algorithm, note that the geomathematics group development

algorithm counts from the bottom up, whereas the atmosphere arrays of simulated ICESat-2 data usually count indices starting from the top down.

Changes for ICESat-2 ATLAS data

(1) Identify ground bin as the central bin of the lowest layer (within an index range [b1, b2] above the DEM).

(2) Define surface height as the height of the center of the central bin of the lowest layer.

(3) Height of a data bin is defined to be height of the center of the surface bin in ATBD atmos, Part 1. This change is made to match the height convention in ATBD atmos, Part 1.

(4) This algorithm piece does not require running the layer-identification first. The layer identification uses the combined mask (from density1 and density2), the ground determination uses only the mask from density1.

13.3 Identification/Classification of Blowing Snow

NOTE (2020-January 16, v11.0): *This approach is part of the CALIOP Test Version. A new approach is in development as of writing of ATBD Part II, v11.0. Blowing snow from the DDA is not implemented yet. The concepts described here are still valid.*

The identification of blowing snow requires ancillary data: Wind speed. These are provided in the ATL04 data.

The identification uses the following information and rules 1-4 listed under "Theory" in section (13.1).

1. location of ground surface (see previous section 13.2)
2. density field
3. windspeed vector

Algorithm Steps

Loop through each column of data in the density field 2 (from density run 2), its corresponding wind speed vector and ground height vector. We will call the iteration of this loop i. The following process will be stepped through in each iteration of the loop.

- Step 1: Check to see if the windspeed exceeds 5m/s. If not, no blowing snow layer exists, so continue to the next column of data (next iteration of the loop i+1).
- Step 2: Subset the density field column to contain bins only above ground, if ground was not identified, continue to the next column of data i+1.
- Step 3: Check to see if the lowest bin (just above ground) exceeds some density threshold, termed *blowsnow_threshold* for blowing snow (which will be discussed later). If not, continue to i+1.
- Step 4: Count upward the number of bins that exceed this density tolerance for blowing snow. If this exceeds 500m, no blowing snow is identified, continue to i+1.

The value *blowsnow_threshold* used as threshold for blowing snow is the same as the threshold used in the density-2 run, but here it is applied only to the bottom layer. Make sure to use a different

variable than for *threshold(2)* (see below!). Note that *threshold(2)* is defined by an equation using several algorithm parameters (Section 3.3.6.3, T5).

Density Threshold for Blowing Snow Detection (CALIOP Test Version)

The blowing snow layer uses the rules (1-4) listed under "Theory" above and is built on the density-mask2, using the same thresholds as in the density-run2 for atmospheric layer detection. It is feasible to implement a specific threshold just for blowing snow. Based on the test data set from CALIOP data, the tenuous cloud threshold (second density run threshold) is a good threshold to isolate blowing snow.

Note on necessity of a separate blowing-snow algorithm loop: Notice that we need to identify the blowing-snow layer at this point in the algorithm, because this step is run before the layer-detection algorithm. I assume that the same simple counting algorithm is used to identify the top of the blow-snow layer (as for ground), i.e. we do not apply a criterion of "min thickness of cloud is three, min gap size above cloud is three", so we can find a 30m-thick blowing snow layer.

Parameter *blowsnow_threshold***:** It may be possible and/or necessary to determine a specific blowing-snow threshold for ICESat-2 atmospheric data after launch. To implement this, utilize *threshold(2)* as a default for *blowsnow_threshold* and set a better value after several case studies on blowing snow have been carried out. Note that *threshold(2)* is defined by an equation using several algorithm parameters (Section 3.3.6.3, T5).

An example of the classification of ground surface and blowing snow in CALIOP-based simulated ICESat-2 data is given in Figure 40. Note that in the figure ground level is plotted as height of the highest bin in the ground layer, whereas the algorithm to be implemented for ICESat-2 ATLAS data should use the top of the highest bin to search for blowing snow and the center of the ground layer as height associated with ground surface.

13.4 Implementation: Threading of Ground Surface Classification Into the Existing DDA-atmos

NOTE (2020-Jan-16, v11.0): *See section 18. Ground is classified from mask 1 and mask 2.*

This section describes where the ground surface classification should be implemented in the DDA-atmos.

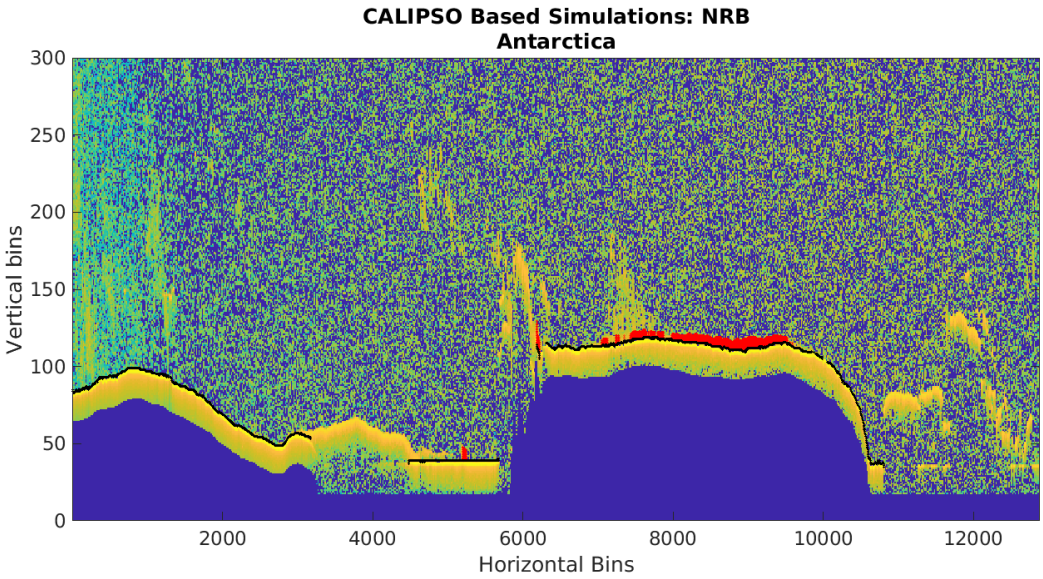


Figure 40. Classification of blowing snow and ground, using ground determination. Data source: ICESat-2 data simulated based on CALIOP data. Data set created by S. Palm June 2018.

(1) Use the DDA with running density twice, i.e. this routine requires that the number of density runs is two.

(2) After calculation of *final mask1*, identify ground, using the algorithm described in section (13.2) and appropriate parameter values for the number of bins above/below the DEM. Store identified ground bins in a new appropriate product variable (*ground_from_density*). This can be implemented using a mask or bin identifiers. Do not change *final mask1*, because the calculation of density2 (the density field from the second density run) is based on proper execution of the DDA-atmos.

(3) Run the main DDA-atmos to the end (through run2 and mask2, then combined mask, then layer boundary determination)

(4) Then check:

If a point (bin) is in both a layer and in the ground variable, THEN take it out of the layers.

(5) Identify ground bin as the central bin of the lowest layer (within an index range [b1, b2] above the DEM). Define surface height as the height of the center of the central bin of the lowest layer.

Parameters: Values for number of bins above/below the DEM and other counters counters have to be implemented as algorithm-specific parameters (=changeable).

Testing Need and Discussion of Concept. The last step (4) does not address the vertical width of the surface layer nor the problem with the 3 times 30m bins in layer smoothing, which is performed in the algorithm module of layer boundary determination (see section (3.6)). Theoretically, this problem should not come up for ground alone, but one needs to make sure that it cannot occur numerically. However, if blowing snow is present or another thick aerosol layer that connects to the ground without a 3-bin (90m) gap, then the layer smoothing algorithm in (3.6) will amalgamate ground and near-ground blowing snow into a single layer. This is the reason to identify the ground layer from density mask1 and not after layer identification.

The algorithms in this section should be tested for regions with blowing snow, after implementation of the new NRB calculation and the Q/A algorithm for the DDA-atmos.

The following steps should be part of testing the ground and blowing snow algorithm components:

Testing Steps

(1) Identify several examples of ground with and without blowing snow and test for those examples.

(2) Testing can be carried out using the existing NRB calculation, but the needs to be repeated for after implementation of the new NRB calculation.

(3) Test the amalgamation routine that is used in the identification of layer tops and layer bottoms, using a value of 2 instead of 3 for minimal layer thickness and minimal layer separation. Also identify layer heights without amalgamation, and compare.

(4) Compare ground heights identified from density with ground heights identified in ATBD, part1.

(5) Note that the surface layer may be several bins thick, depending on the kernel, and thus alternative counting algorithms may be useful. However, ultimately the blowing snow layer should be identified using a version of the DDA-ice applied to the raw point cloud data.

14 Analysis of First ICESat-2 ATLAS Data After Launch and Sensitivity Study for ATLAS Atmosphere Data

ICESat-2 was launched on September 15, 2018 from Vandenberg Airforce Base in California on the last Delta-2 Rocket of United Launch Alliance.

14.1 Experiment Setup, Data and Results for Parameter Implementation. t56.

Initial data sets created after launch are version 200. To optimize performance of the algorithm for post-launch data, a sensitivity study was conducted using several parameter combinations. The parameter sets used in this sensitivity study are **t(55) - (t72)**, which are listed in Table 7.

The sensitivity study was carried out on a subset of a composite of several cloud layer characteristics.

`ATL04_20181016022104_02670101_200_01.h5`

is the granule that was used to create

`sensi_subset_20181016022104`

The subset has 8219 profiles.

Result. Following a sensitivity study of the algorithm specific parameters conducted using several parameter combinations, **t(55) - (t72)**, we determined that the optimal parameter combination is $t56$. Notably, this is the same parameter combination that worked best for the last round of pre-launch-data, along with $t64$. Parameter values are given in the Table 7.

Comparison to v200 ATL09s and conclusion (2018-October-23). The sensitivity studies are run by using ATL04 data and the geomath version of the DDA-atmos, along with the Q/A algorithm (which at this time has not been implemented in the SIPS version of the code) to produce ATL09 data. Then those ATL09 results are compared to the matching ATL09 data files (v200), derived by SIPS/ASAS.

The only change required to match $t56$ was

300, 600 for size_threshold1, size_threshold2

The min cluster size is called size_threshold1, size_threshold2 (for density run1, run 2 respectively). In ATL09[rest].h5:

`/ancillary_data/atmosphere/size_threshold1`

Initially used values were 600, 600 . These values are reported on the .h5 file in ATL09. The values are used in code that creates ATL09s from ATL04s.

This parameter change has been made for v201.

Note. NRB value calculation will change in the near future (for v203 or v204) and may require a new sensitivity study.

14.2 Sensitivity Study and Post-Launch Q/A

Sensitivity studies are run for the parameter combinations **t(55) - (t72)** given in Table 7. The parameter sets are defined such that higher and lower values are tested individually for each of the final parameters. If a better combination is encountered in a sensitivity run, then lower and higher values around those last winning parameters are tested in a new sensitivity study. This process is repeated until no better values are found. In critical cases, intermediate parameters are tested as well. Note that Table 7 also includes parameter sets used for sensitivity studies that became necessary after changes to the input data in ATL04.

The following steps are illustrated for each parameter set shown in Figure 41 (by rows)

1. Raw NRB data (valid bins)[ATL04 input data to the DDA-atmos]; Cloud Layer Boundaries over Raw NRB Data
2. Pass 0: Kernel Matrix; Pass 1: Kernel Matrix
3. Pass 0: Density; Pass 1: Density
4. Pass 0: Thresholds Along Track; Pass 1: Thresholds Along Track
5. Pass 0: Density - Thresholded; Pass 1: Density - Thresholded
6. Pass 0: Density - Declustered (Mask 1); Pass 1: Density - Declustered (Mask 2)
7. Pass 0: Final Mask with Density (Combined Mask); QA: Half-Gap Confidence

Note that the code counts *pass 0* and *pass 1*; these are referred to as *density run 1* and *density run 2* in the text.

NOTE: Figure 41 is located in Appendix S (Sensitivity Studies with Figures, Section 14).

Parameter Set Name	Sigma	Anisotropy	Cutoff	Down-sample	Minimum Cluster Size	Threshold Bias	Threshold Factor	Threshold Window	Quantile
t55	3	10, 15	1	1	300, 600	10E+14	1, 0.9	2	0.99, 0.8
t56*	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.99, 0.8
t57*	3	10, 15	1	1	300, 600	10E+14	0.9, 1	2	0.99, 0.8
t58*	3	10, 20	1	1	400, 600	10E+14	0.9, 1	2	0.99, 0.8
t59*	3	10, 20	1	1	300, 600	12E+14	0.9, 1	2	0.99, 0.8
t60*	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.95, 0.8
t61	3	10, 15	1	1	300, 600	10E+14	0.9, 1	2	0.95, 0.8
t62	3	10, 15	1	1	300, 600	12E+14	0.9, 1	2	0.95, 0.8
t63*	3	10, 30	1	1	300, 600	10E+14	0.9, 1	2	0.99, 0.8
t64*	3	10, 20	1	1	200, 600	10E+14	0.9, 1	2	0.99, 0.8
t65	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.97, 0.8
t66	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.98, 0.8
t67	3	10, 20	1	1	150, 600	10E+14	0.9, 1	2	0.99, 0.8
t68*	3	10, 20	1	1	300, 600	9E+14	0.9, 1	2	0.99, 0.8
t69*	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.99, 0.7
t70*	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.99, 0.9
t71	3	10, 20	1	1	600, 600	10E+14	0.9, 1	2	0.99, 0.8
t72	3	10, 20	1	1	300, 600	11E+14	0.9, 1	2	0.99, 0.8
t73	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.99, 0.6
t74* †	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.99, 0.55
t75	3	10, 20	1	1	300, 1000	10E+14	0.9, 1	2	0.99, 0.55
t76	3	10, 20	1	1	300, 1500	10E+14	0.9, 1	2	0.99, 0.55
t77	3	10, 20	1	1	300, 2000	10E+14	0.9, 1	2	0.99, 0.55
t78	3	10, 20	1	1	300, 2000	10E+14	0.9, 1	2	0.99, 0.6
t80 †	3	10, 20	1	1	150, 600	10E+14	0.9, 1	2	0.99, 0.55
t81	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.99, 0.45
t82	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.99, 0.3
t83	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.99, 0.5
t84 †	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.99, 0.45
t85 †	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.95, 0.5
t86	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.95, 0.65
t87 †	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.97, 0.55
t88 †	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.97, 0.65
t89 †	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.97, 0.45

Table 7: **Parameter combinations used in sensitivity studies of the DDA-atmos applied to ATLAS data after launch (double density runs).** The table is for using double density runs ($num_passes = 2$). If two values are listed, then value 1 is for density-run1 and value 2 is for density-run1. If one value is listed, this value is used for both density-run1 and density-run2. (*) and (†) denote experiments for which results are shown in a figure.
(*)(1) Sensitivity study for first after-launch data analysis (except (t74)). Results in Figure 41. (t56) Parameter combination used in the official algorithm. Algorithm run on sub-sampled regions in the ATL04_20181016022104_02670101_200_01.h5 data file using returns from profile 2 of the ATLAS beam configuration.
(*)(2) Sensitivity study for after-launch data analysis, for ASAS code v5.0. Results in Figure 43. Algorithm run on sub-sampled regions in the ATL04_20181017T002107_02810101_950_01.h5 data file using returns from profile 2 of the ATLAS beam configuration.
(t69) Best compromise data set used for after-launch data analysis, ASAS atmos code version v5.0 and for the case that the same parameter set is used for day/night/dusk.
(3) Default values for 3 parameter sets are:
day: paramset1=t60, night: paramset2=t74; dusk: paramset3=t60 (see Figure 44). Algorithm run on sub-sampled regions in the ATL04_20181017T002107_02810101_950_01_dda3.h5 data file using returns from profile 3.
† **(4) Sensitivity study for first public release of atmospheric data (release date: May 2019), for ASAS atmos code v5.1. Results in Figure 46.** Algorithm run on sub-sampled regions in the ATL04_20181017T002107_02810101_951_01.h5 data file using returns from profile 3 of the ATLAS beam configuration.
Parameter sets are: **day: paramset1=t60, night: paramset2=t87; dusk: paramset3=t60.**

15 Post-launch Q/A Considerations

15.1 On the Range of Half-Gap Layer Confidence Values

This section is in progress during the post-launch data analysis and testing phase (i.e. data values / NRB values may change somewhat, as of 2018-Dec-20). Q/A code is currently being implemented by SIPS/ ASAS. The figures in this section are created using development code of the geomath group. As are all figures in the ATBD so far.

The definition section (11.3) for half-gap confidence originally stated that: Layer confidence is normalized such that values generally fall between zero and 1 and assume a good spread across the range [0,1] for most atmospheric layers. However, layer confidence can assume values outside of this range.

15.1.1 Background: NRB values and the DDA

Calculation of NRB values is set in a different mathematical frame work than the DDA for atmosphere. NRB calculation is described in summary in section *1. Data* of this document (ATBD Atmos Part II) and in detail in ATBD Atmos Part I. The NRB equations (3.1 and 3.2) are on page 25 (section 3.3) of the ATBD Atmos Part I.

NRB calculation includes a step of background subtraction, which is in essence subtraction of an average value of some region (just one profile from the shot-sum of counts for that profile).

This yields negative values, but so far is only a linear shift. After that, range correction for distance to satellite is performed (non-linear), which shifts the range of input values to negative and positive values with numbers ranging to / on the order of E+14. The density calculation creates a sum of values (without standardization). Because of the definition of the threshold function, the density values in a cloud will be positive. If the parameter combination is ill-chosen, there can be negative density values in the clouds, but with optimized param combinations (such as t56, t64), this will not happen. The possibility of negative A exists.

The confidence value remains meaningful. If confidence is larger than one, the cloud determination is even better.

15.1.2 Data sets for used for testing (here) and submitted to playground (at Goddard)

(1a) sensi_subset_20181016022104

(1) ATL04_20181016022104_02670101_200_01.h5

(2) ATL04_20181016035521_02680101_200_01.h5

Data set (1a) is a subset created from granule (1), to represent several (6) different situations of clouds and other atmospheric layers during day-time, night-time and dusk. The data set (1) covers about half an orbit on the northern hemisphere, ascending to near pole and descending back over the Arctic ocean. Data set (1) includes about 60k profiles and hence it is not possible to discern the results of changes in the algorithm-specific parameters in plots of the entire granule. Data set (1a) has 8219 profiles.

Data set (2) is the next data set collected after data set (1) on 2018-Oct-16. Data set (2) has data over the southern hemisphere. Data set (1) has data over the northern hemisphere. This is coincidence, as each data set = granule should be a full orbit.

Data sets (1) and (2) were submitted to "playground" for parameter testing and testing of the implementation of the Q/A method (half-gap layer confidence).

15.1.3 Examination of existence of confidence outside of [0,1]

Here, data set (1a) is re-analyzed to look at the possibility of layer confidence values below zero and above 1. Analysis carried out 2018-Oct-26.

Parameter combination: **(t56)**

The parameter combination (t56) was found to be optimal for post-launch data, as determined by the post-launch sensitivity study carried out October 18-21, 2018.

The results of the re-analysis are shown in the following figures.

In Figure 42a, the color scale set so that negative densities in density field 1 show up as black:

(Case 1) During night time, solid black regions below cloud layers can occur (negative NRB values), seen around 1700000 to end of data set in right third of the figure. Detection of the cloud layers

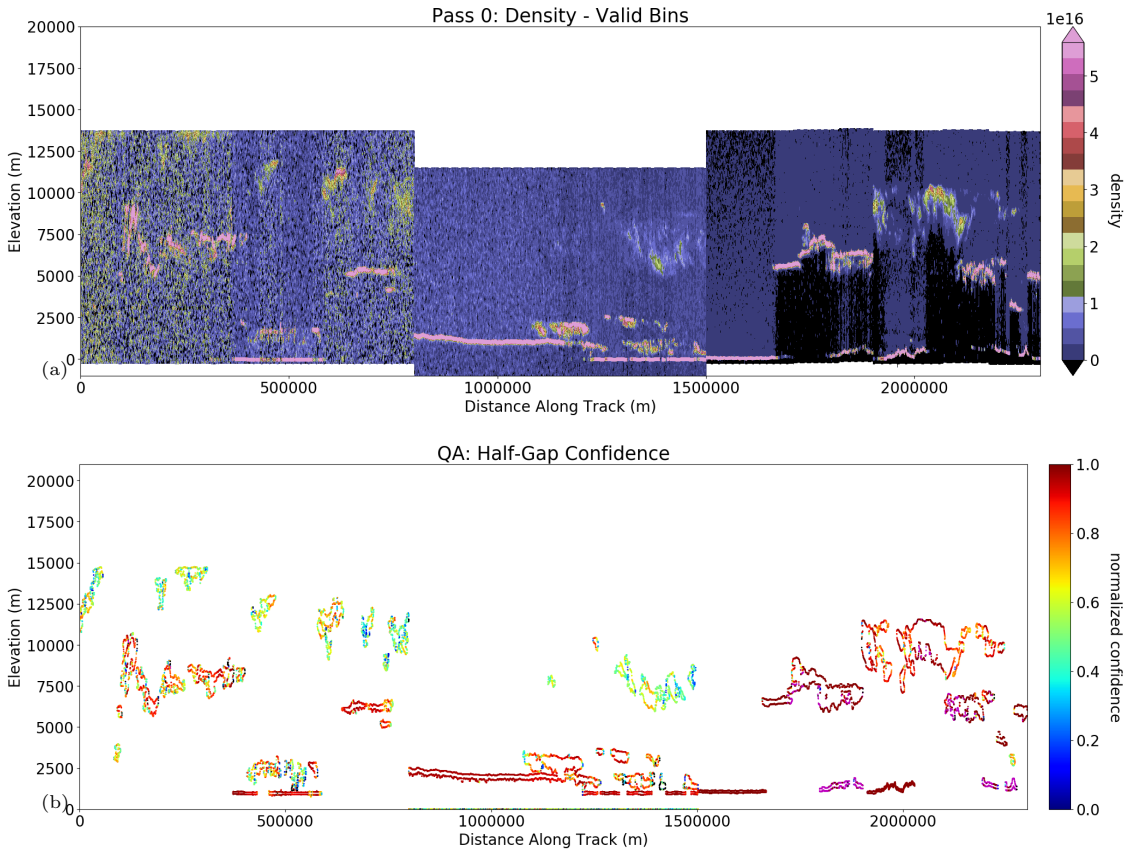


Figure 42. Half-gap confidence of determination of atmospheric layer boundaries. Check on values outside of [0,1]. Applied to 8000+ (8219) profile synthetic data set representing different cloud types and night-time/ day-time/ dusk transition, post-launch ATLAS data sensi_subset_20181016022104 based on ATL04_20181016022104_02670101_200_01.h5. Parameter combination t56 (See Tables 5 and 6 for parameters). (a) Density mask with a color scheme that shows negative NRB values in black. (b) Half-gap confidence. Confidence values larger than 1 in magenta, smaller than zero in black. Values in [0,1] as in color scale.

above and below these regions of negative NRB values can yield really high confidence values, seen in Figure 40b in magenta. The cloud layers stand out optically and high confidence in their detection is adequate.

Highest confidence values found in this analysis are up to 2000. "Magenta" confidence values occurred for about 2000 points, of approximately a total of 40,000 points (counting points along the cloud boundaries).

(Case 2) In addition, fuzzy regions with black spots here and there occur during day-time observations and during dusk (see, Fig. 40a). These probably yield values of A and B that are really close to each other. Here, the "scooping up" of regions identified as cloud into non-cloud and

vice versa (see definition of layer boundaries in section (3.7)) may yield confidence below zero in Figure 40b, marked by black dots. Notably, these black dots occur along vertical cloud boundaries or non-boundaries. If confidence is less than zero, the cloud layer determination is not certain; i.e. the confidence value remains meaningful. We attribute the existence of "black confidence" to the smoothing of the cloud boundaries. The cloud exists, but the location of the boundary is approximate.

"Black" confidence values occurred for about 2000 points, the lowest values found was -604 (determined numerically).

While values of [-604, 2000] appear to be far outside of the range [0,1], they are on the order of -10^2 to 10^3, and NRB values range from -10^{16} to 10^{16}. [threshold parameter is 10^{14}]

Testing To Do. We should really check what the SIPS code does in these black locations (i.e. difference their/our cloud boundaries).

Thought. It may be worth-while to differentiate between confidence in detection of a cloud and confidence in the location of the boundary? For black dots, the uncertainty is likely due to the cloud being small and the location of its boundary being uncertain (smoothed following the criteria for the layer top/ layer bottom algorithm).

16 Sensitivity Study for Pre-Release Data Version v950, Necessitated by Change in Background and NRB Calculation in ATL04

Creation of a good data product requires improvements of the algorithms and updates of the code used for analysis, which is a process that initially starts from the lower-level products and continues through the higher-level products. Changes of the calculations of background in the atmospheric data in ATL04 have led to changes in calculation of the normalized radiometric backscatter (NRB) values, also in ATL04, on which the detection of atmospheric density, layers and related results from the DDA-atmos is based.

Following such a change in background and NRB data in ATL04 in ASAS code version v5.0 (data labeled pre-release test version v950), a new sensitivity study was run to optimize DDA-parameter sets for the updated ATL04 data. Studies are run for the parameter sets **t(55) - (t74)** given in Table 7, rerunning **(t55) - (t72)** and adding two new parameter sets, **(t73), (t74)**.

The DDA-atmos was run on sub-sampled regions in the data file
ATL04_20181017T002107_02810101_950_01.h5
using returns from profile 2 of the ATLAS beam configuration. Note that the atmospheric data products utilize only the three strong beams and the relationship of beam numbers to profile numbers changes whenever the satellite's measuring array is flipped. Specific information is found in the data products. Beam 2 is always associated with profile 2, but numbers 1 and 3 flip.

Result: (t69) is the best compromise data set used for after-launch data analysis, ASAS atmos code version v5.0 and for the case that the same parameter set is used for day/night/twilight.

The results are reported here, because they are used in following sections to inform later optimizations of parameter sets. The following steps are illustrated for each parameter set shown in Figure 43 (by rows):

1. Raw NRB data (valid bins)[ATL04 input data to the DDA-atmos]; Cloud Layer Boundaries over Raw NRB Data

2. Pass 0: Kernel Matrix; Pass 1: Kernel Matrix

3. Pass 0: Density; Pass 1: Density

4. Pass 0: Thresholds Along Track; Pass 1: Thresholds Along Track

5. Pass 0: Density - Thresholded; Pass 1: Density - Thresholded

6. Pass 0: Density - Declustered (Mask 1); Pass 1: Density - Declustered (Mask 2)

7. Pass 0: Final Mask with Density (Combined Mask); QA: Half-Gap Confidence

NOTE: Figure 43 is located in Appendix S (Sensitivity Studies with Figures, Section 16).

17 Day-Night-Twilight: Implementation of Three Sets of Parameters for DDA-atmos Dependent on Day-Time (Sun-Elevation Angle)

17.1 Summary

Objective of code change: Use three day-time dependent parameter sets for the layer detection in the DDA-atmos.

Motivation: In ATL04, a change was made to use day-time-dependent calculation of background and thus, of NRB values. Dependent on ranges of sun-elevation angles, three times of day are now defined in the latest code update: (1) day-time, (2) night-time, (3) twilight, and a different algorithm is applied for NRB calculation for each time range. The DDA-atmosphere is always optimized to match the NRB calculation. The parameter set used hitherto to drive the DDA-atmos is a compromise for atmospheric layer detection among the day/night/twilight data situations, which have distinctly different background characteristics. At this point, definition of three DDA-atmos parameter sets specific to the three ranges of day-time is the natural step to match the three day-time ranges in the NRB data. The DDA-parameters that will be multiplied into three parameter sets are listed in Table 8.

17.2 Algorithm Change

A small algorithm change is required:

So far, there has been one parameter set for the DDA-atmos (with double-density runs). The number of density runs is a parameter as well, *num_passes*. New, there will be three parameter sets; i.e. we introduce a triplet of parameter values for each parameter, making it three parameter sets,

*param_set_*1 (for day), *param_set_*2 (for night), *param_set_*3 (for twilight).

The parameters are given in Table 8, with their ATL09 variable name (as pulled from an ATL09 file) and common name for explanation, matching the latest version of the ATBD for atmospheric data products from Dec. 2018.

ATL09 Variable Name	Common Name
a_m1	anisotropy factor pass 1
a_m2	anisotropy factor pass 2
cutoff1	cutoff pass 1
cutoff2	cutoff pass 2
downsample1	downsample factor pass 1
downsample2	downsample factor pass 2
num_passes	number of passes
quantile1	quantile pass 1
quantile2	quantile pass 2
sigma1	sigma pass 1
sigma2	sigma pass 2
size_threshold1	minimum cluster size pass 1
size_threshold2	minimum cluster size pass 2
thresh_bias1	threshold bias pass 1
thresh_bias2	threshold bias pass 2
thresh_sensitivity1	threshold sensitivity factor pass 1
thresh_sensitivity2	threshold sensitivity factor pass 2
threshold_segment_length1	threshold segment length pass 1
threshold_segment_length2	threshold segment length pass 2

Table 8: DDA-atmos algorithm-specific parameters

The identification of day-time is carried using sun-elevation angle, in ATL04. The same sun-elevation angle decision tree will be used for day-time/night-time/twilight for the DDA-atmos.

Algorithm:

If *sun_elev_angle* is in *day_range* then use *param_set_1*.

Else if *sun_elev_angle* is in *night_range* then use *param_set_2*.

Else use *param_set_3* [this will be for twilight].

Current settings (as of v5.1) of the day-time ranges:

```
Night-time is when the solar elevation angle is < =-7.0 degrees.
Day-time is when solar elevation is > -1.0 degrees
Twilight is the solar elevation in between these angles. (> -7.0 and <= -1.0)
```

The ASAS adjustable parameters that define this are:

```
backg_min_solar_elev = -7.0
backg_max_solar_elev = -1.0
```

They are on ATL04 in the ancillary group (in v 5.0 and later).

Note that for v5.0, their values are -5.0 and -3.0, respectively.

Combination

night_range= [-90.0, *backg_min_solar_elev*]

day_range=]*backg_max_solar_elev*, 90.0]

17.3 Determination of Default Parameter Sets for Day - Night - Twilight

This affects DDA-atmos parameters listed in Table 8. The selection of the preliminary parameter sets is based on a sensi-study run for v950 data, collected October 17, 2018 and calculated Feb 7, 2019.

Data set used:

`ATL04_20181017T002104_02810101_950_01.h5 (profile 3 in ATLAS configuration)`

The sensitivity study uses a subset that covers different situations of clouds and aerosols during each time range of day/night/twilight. The entire data granule was run for (t69), the previously selected parameter set.

Previously used best compromise parameter set (using same parameter set for all day-times): **(t69)** (see Table 7). (The set (t69) is like **(t56)**, but quantile2=0.7. (t56) has quantile2=0.8).

Results: Default Parameter Determination (see Table 7 and Figure 44):

num_passes = 2 for all three time frames, day, twilight, night; i.e. double-density runs are used.

num_passes may be different for future runs or data situations. So far, double-density runs have worked best for all post-launch data analyzed.

(1) For day time: Use paramset1=t60

(like t56 and t69, but quantile1=0.95)

Reasoning: Bring out more clouds in density-run1 to avoid artefacts (false positives) in run2.

(2) For night-time: Use paramset2=t74

(like t56, but quantile2=0.55)

Evaluation: At night time, lower quantile 2 brings out more tenuous clouds and still does occultation below clouds properly (if mask1 is used for a ground loss flag). Note that the larger kernel in density-run2 brings out ground under intermittent clouds, which is good.

(3) For twilight: Use paramset3=t60

(like t56 and t69, but quantile1=0.95)

Reasoning: Keep the values from day-time until further experiments. The subset in our sensitivity study does not have clouds during twilight. (t56) run for the entire data set performs ok for the twilight section, as much as one can tell in a non-subsetted analysis.

Note. Following implementation of further code changes in ASAS v5.1, a new sensitivity study was performed to further optimize results for the first public release, see section (19).

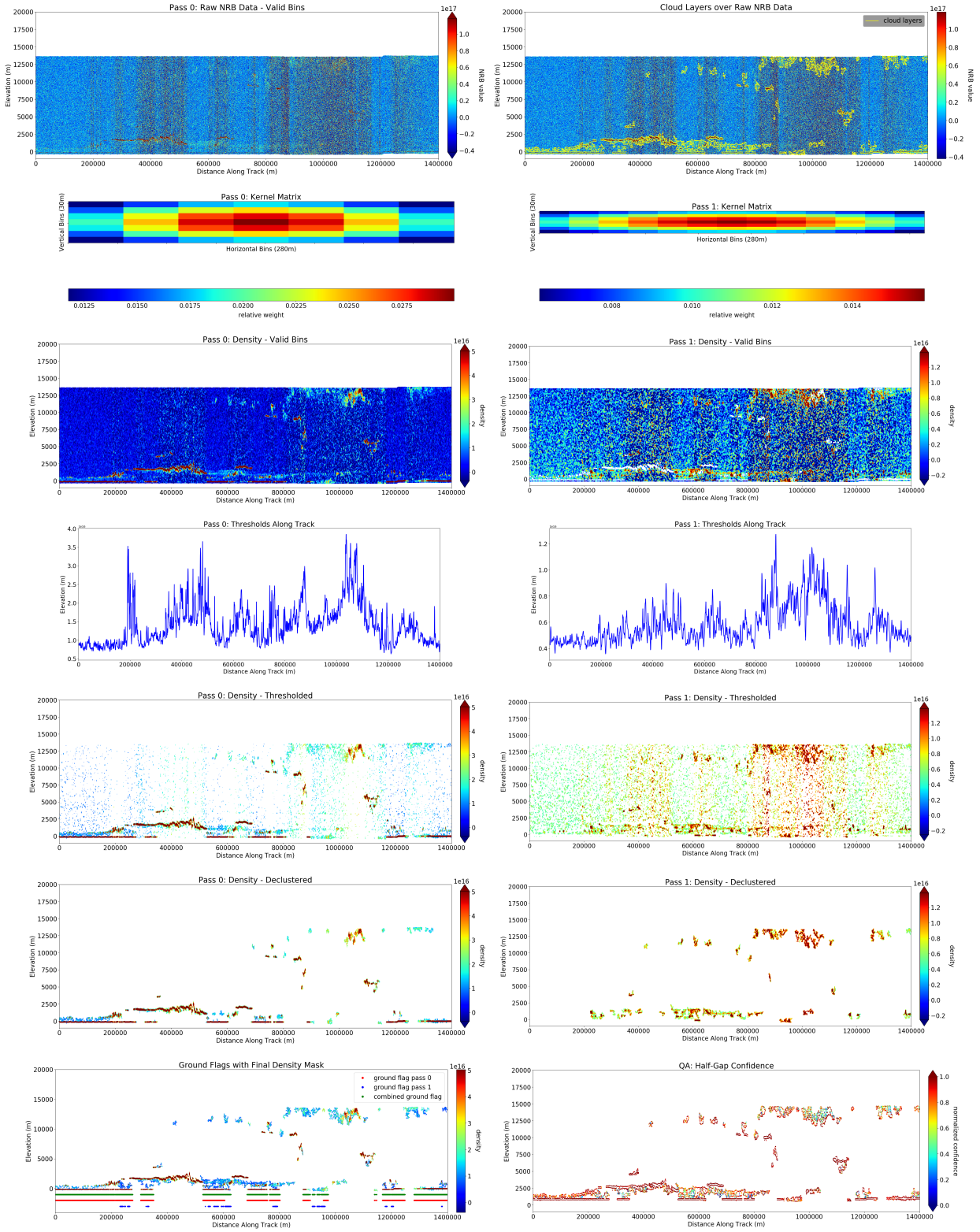


Figure 44 - 1. Sensitivity analysis of ICESat-2 ATLAS data post-launch: (t60) applied to day-time data. Algorithm run on sub-sampled regions in the ATL04_20181017T002107_02810101_950_01_dda3.h5 data file using returns from profile 3 of the ATLAS beam configuration.

σ = 3,3	cutoff = 1,1
a_m = 10,20	min cluster size = 300,600
threshold_bias = 10E+14,10E+14	downsampling = 1,1
threshold_sensitivity = 0.9,1	threshold_segment_length = 2,2
quantile = 0.95,0.8	

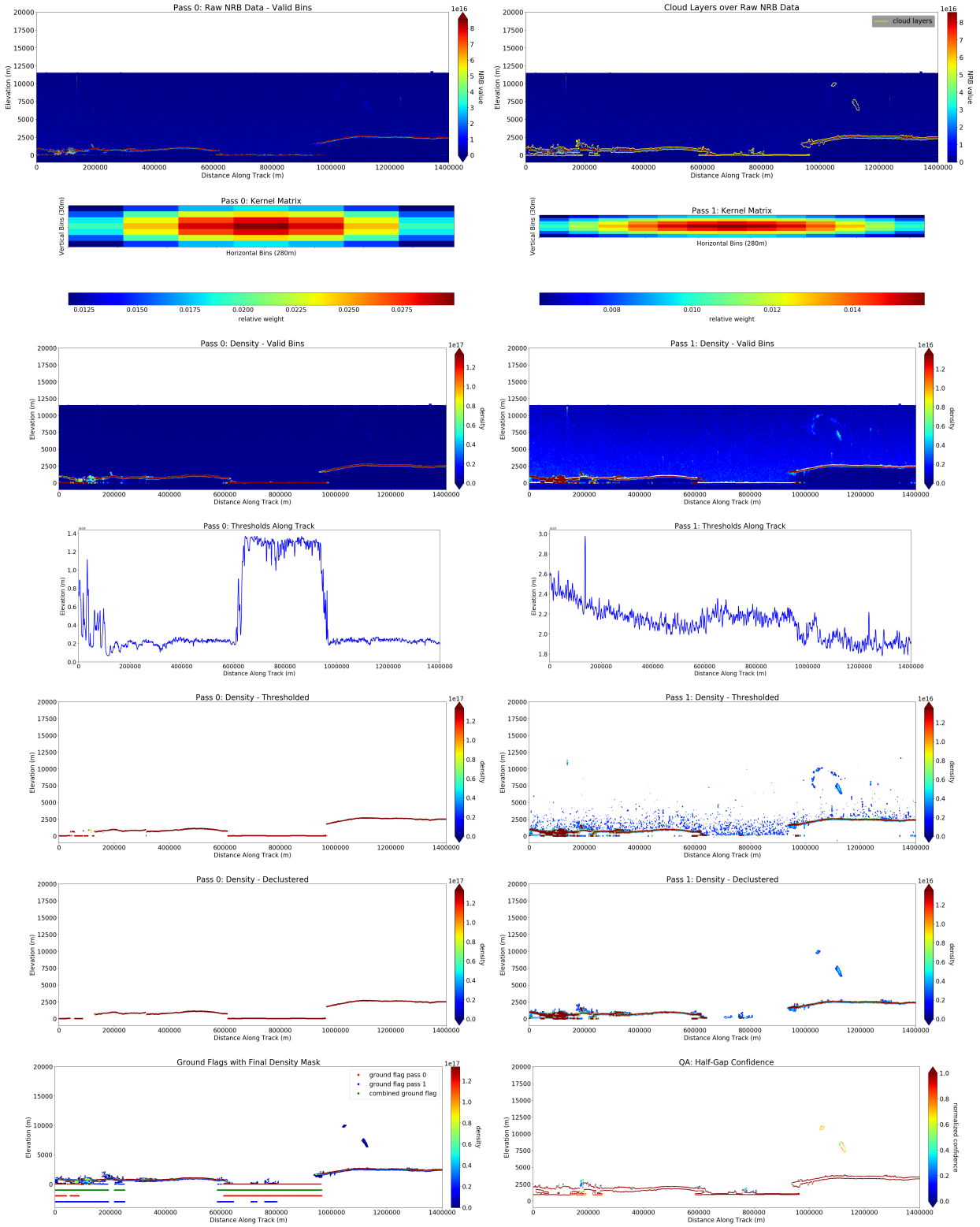


Figure 44 - 2. Sensitivity analysis of ICESat-2 ATLAS data post-launch: (t74) applied to night-time data. Algorithm run on sub-sampled regions in the ATL04_20181017T002107_02810101_950_01_dda3.h5 data file using returns from profile 3 of the ATLAS beam configuration.

σ = 3,3	cutoff = 1,1
a_m = 10,20	min cluster size = 300,600
threshold_bias = 10E+14,10E+14	downsampling = 1,1
threshold_sensitivity = 0.9,1	threshold_segment_length = 2,2
quantile = 0.99,0.55	

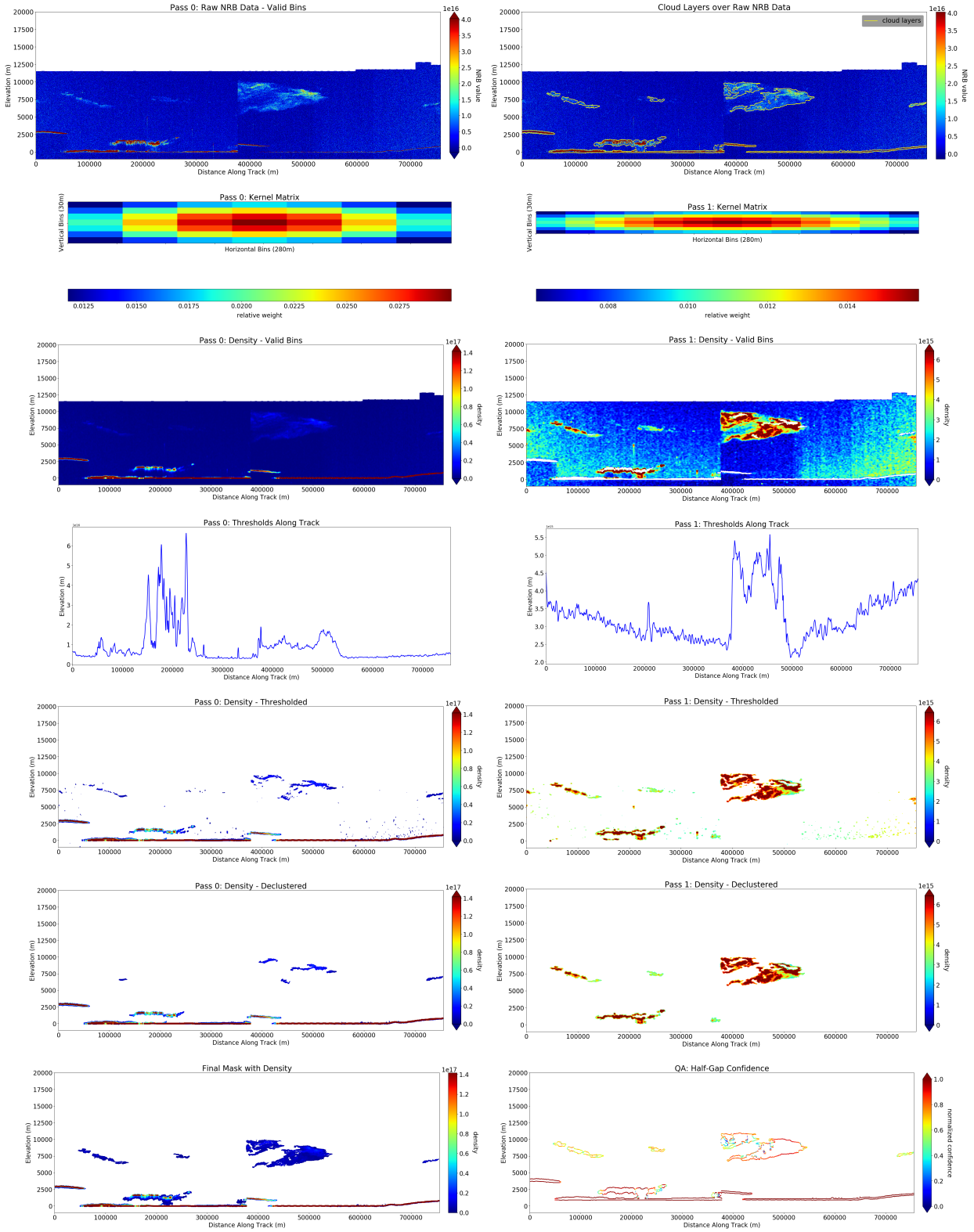


Figure 44 - 3. Sensitivity analysis of ICESat-2 ATLAS data post-launch: (t60) applied to twilight data. Algorithm run on sub-sampled regions in the ATL04_20181017T002107_02810101_950_01_dda3.h5 data file using returns from profile 3 of the ATLAS beam configuration.

σ = 3,3	cutoff = 1,1
a_m = 10,20	min cluster size = 300,600
threshold_bias = 10E+14,10E+14	downsampling = 1,1
threshold_sensitivity = 0.9,1	threshold_segment_length = 2,2
quantile = 0.95,0.8	

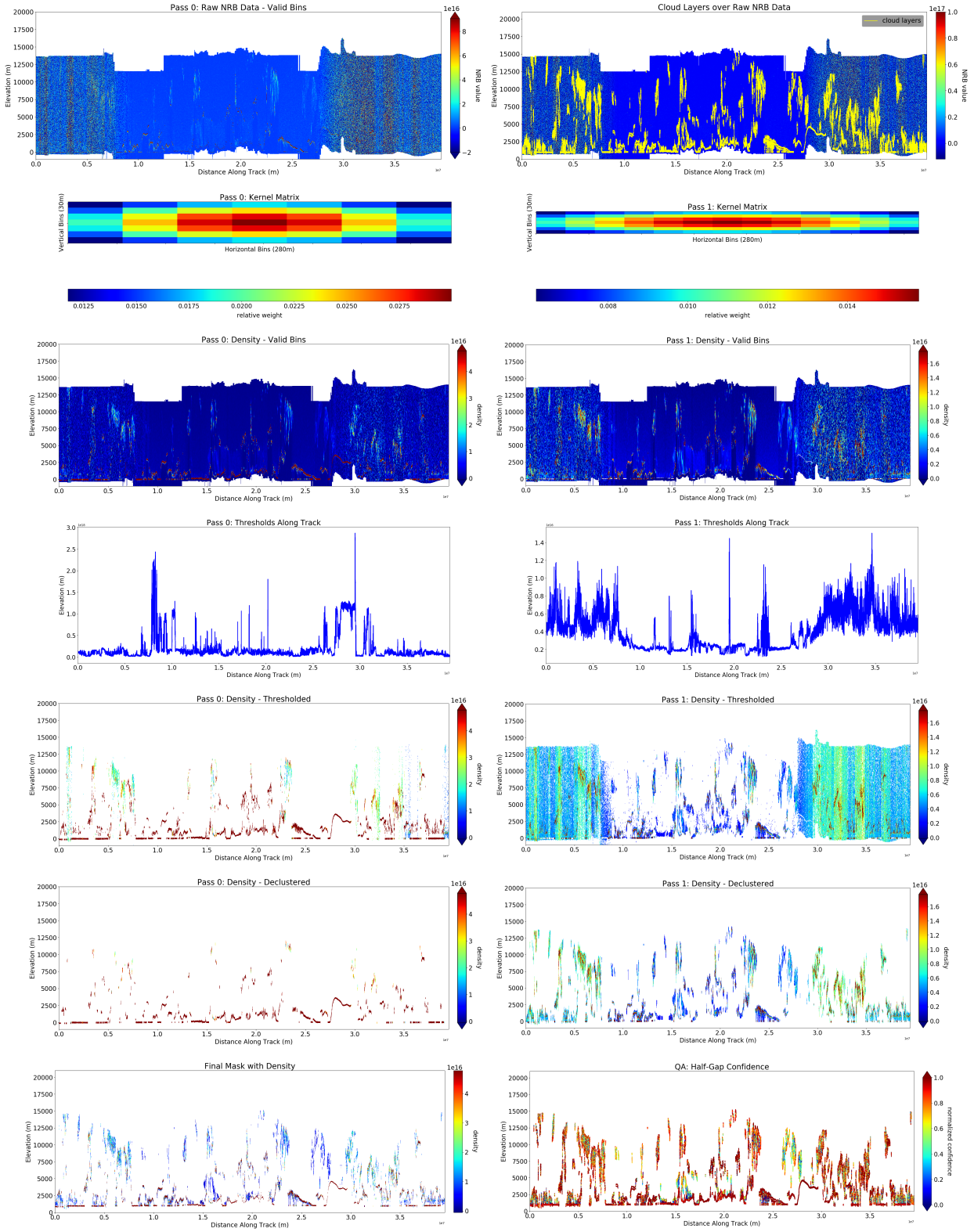


Figure 44 - 4. Sensitivity analysis of ICESat-2 ATLAS data post-launch: (t56) for Day/night/-dusk. Algorithm run entire granule ATL04_20181017T002107_02810101_950_01_dda3.h5 data file using returns from profile 3 of the ATLAS beam configuration.

σ = 3,3	cutoff = 1,1
a_m = 10,20	min cluster size = 300,600
threshold_bias = 10E+14,10E+14	downsampling = 1,1
threshold_sensitivity = 0.9,1	threshold_segment_length = 2,2
quantile = 0.99,0.8	

18 On Ground and Cloud — Algorithm for Surface-Height Determination and for a Cloud-Based Ground-Detection Flag

18.1 Motivation

One of the two main applications of the atmospheric data analysis is to indicate ground detection for all other data products, in the photon cloud returned from the Earth surface, including land and sea ice, canopy and ground under canopy, and inland regions including inland water bodies. To this end, flags indicative of ground detectability based on atmospheric data products are developed. One such flag is described in Part I of the ATBD atmosphere. A second cloud-based ground-detection flag is described here, based on the DDA-atmos. It utilizes the ground-detection algorithm introduced in section (13) of this document, Part II of the ATBD atmosphere. Based on post-launch experience with ICESat-2 atmospheric data and point-cloud data (ATL03), a new ground determination algorithm is documented here *updated 2020-Jan-16 for ATBD Part II v.11.0)* The new algorithm also recognizes that three different parameter sets have been implemented for day/night/twilight, to match the different background calculation and thus NRB calculation methods applied in part1 of the ATBD atmosphere.

Note: It is important to understand that the cloud-based ground-detection flag tells us where we *should* be able to find ground. Whether ground is actually found depends on the algorithm employed for ground detection, which is described elsewhere, for example, in the ATBDs for ATL03, ATL06, ATL08, and in Herzfeld et al. [2017].

18.2 Cloud-Based Ground-Detection Flag (DDA-atmos): Algorithm

NOTE: *For geomath and ASAS code implementation versions, see section (18.3) The current state-of-the art algorithm is described in section (18.4) ff.*

Cloud-Based Ground-Detection Flag (DDA-atmos): Algorithm I

(1) utilize double-density runs

(2) first density run identifies strong layers (gets mask1)

(3) second density run identifies weak layers (gets mask2)

(4) ground flag uses a combination of mask1 and mask2

(5) use masks, not atmospheric layer boundaries (avoids problems in situations of blowing snow, aerosols and other ground-near layers)

(6) flag rule: identify a bin as ground if bin is in mask i (i=1,2) and within 3 bins of the DEM (DEM listed on ATL04).

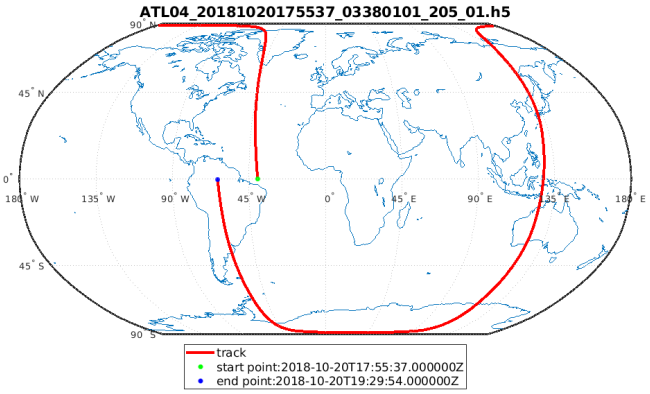
(7) ground height rule: if several bins meet this criterion, then use (counting from bottom up) the (n+1)-th bin (where kernel height is 2n+1) to determine ground height as height of bin-center.

This is illustrated in the following analysis.

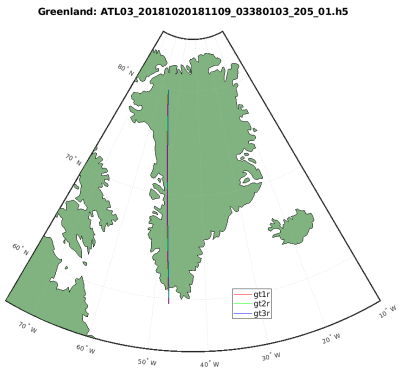
18.3 Analysis Illustrating Ground Flag Detection Algorithm

Data files and code versions:

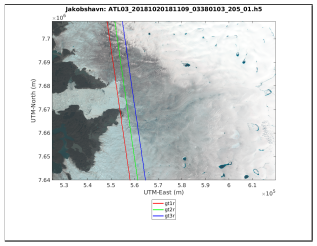
- v205 data set (see Figure 45)
- ASAS code v5.0 for atmospheric data products (ATL04)
- Geomath developer code for DDA-atmos v112 (to illustrate v5.1 and later results that will be implemented; CCB approved and in the queue as of 6 May 2019, v10.0 ATBD)



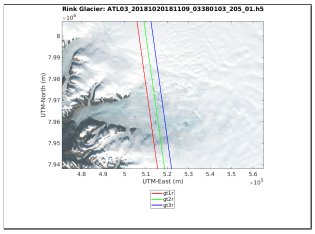
(a) ATL04_track_jak_20181020175537



(b) ATL03_track_Greenland_20181020181109



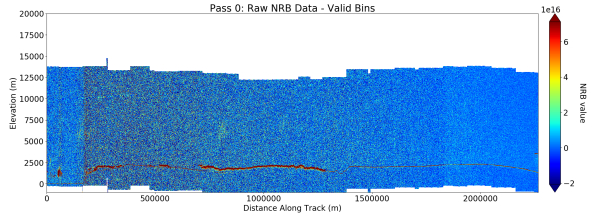
(c) ATL03_track_Jakobshavn _20181020181109



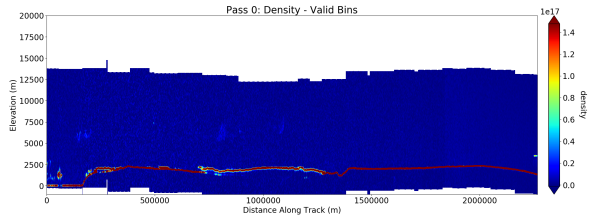
(d) ATL03_track_Rink _20181020181109

Figure 45. (a) Track plot of ATL04 data set used for analysis of cloud-based ground-detection flag, (b) location of same track over Greenland, (c) track location over Ilulissat Ice Stream (Jakobshavn Isbræ), (d) track location over Rink Glacier, Greenland. Background for (c), (d) Landsat.

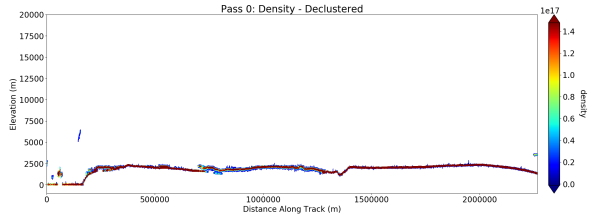
Figure 46 shows results of the essential DDA processing steps used in determination of the cloud-based ground-detection flag.



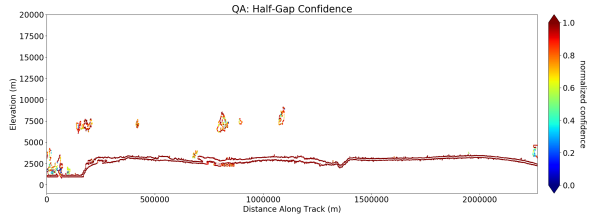
(a) Raw NRB Data (from ATL04)



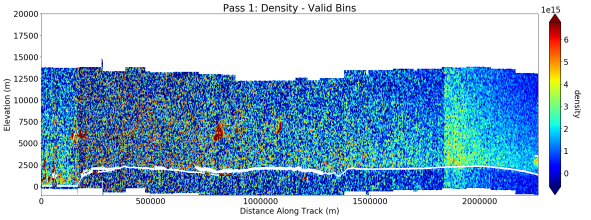
(b) Density field 1 (on ATL09)



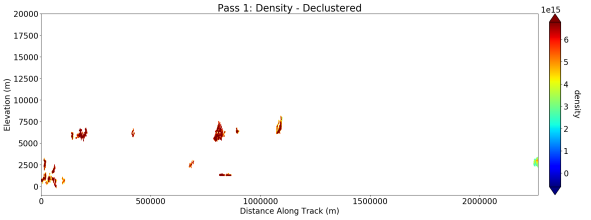
(c) Density field 1 (on ATL09)



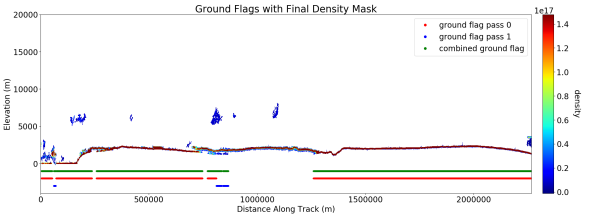
(g) Q/A results



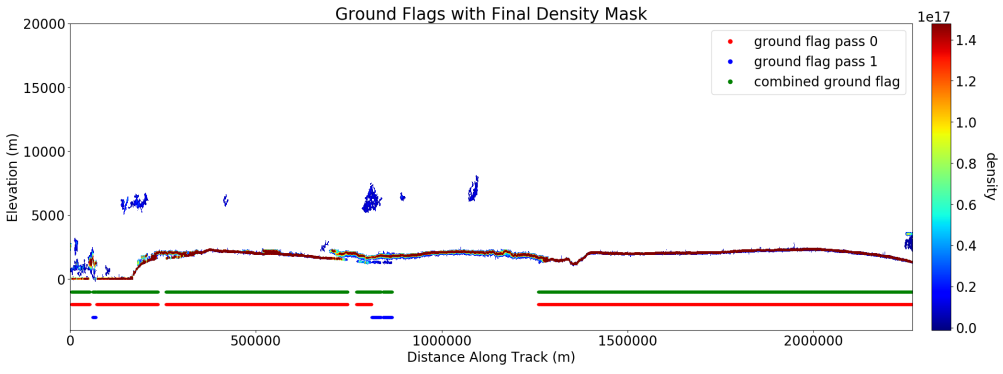
(d) Density field 2 (on ATL09)



(e) Density 2 thresholded and declustered (Mask2)



(f) Combined "cloud" mask and cloud-based ground-detection flag



(h) Combined "cloud" mask and cloud-based ground-detection flag

Figure 46. Ground flag algorithm. (a) - Raw NRB data (from ATL04), (b) Density field 1, (c) Density 1 thresholded and declustered (Mask1), (d) Density field 2, (e) Density 2 thresholded and declustered (Mask2), (f) Combined "cloud" mask and cloud-based ground-detection flag, (g) Q/A results, (h) same as (f), enlarged to facilitate interpretation of different cloud-ground situations.

18.4 Algorithm for Ground Height Determination and a Cloud-Based Ground-Detection Flag (DDA-atmos)

(A) Principles for implementation

These few items are written to help the coder identify the place where to run the new ground identification module.

(1) Number of runs: utilize double-density runs *num_passes = 2*, see Table in Appendix A.

(2) Role of mask1: first density run identifies strong layers (gets mask1)

(3) Role of mask2: second density run identifies weak layers (gets mask2)

(4) Ground flag uses a combination of mask1 and mask2 (simply: combined_mask is the joint set of mask1 and mask 2.

Rule: A bin x is in combined_mask if (x is in mask1) or (x is in mask2).

(5) Use masks, not atmospheric layer boundaries (avoids problems in situations of blowing snow, aerosols and other ground-near layers), to determine ground height. (We'll get to layers later).

(6) Make sure to utilize the three parameter sets for day/dusk/night, following the ASAS-implementation standard, i.e. the order is alphabetical:

*param_set_*1 (for day), *param_set_*2 (for night), *param_set_*3 (for twilight).

Note this is a chance compared to the description in section 17 in ATBD part 2, v10.0.

Since there are three param-sets for each of the two density runs, there need to be 6 parameter sets:

*param_set_*$(i, 1)$ (for day), *param_set_*$(i, 2)$ (for night), *param_set_*$(i, 3)$ (for twilight), for $i = 1, 2$ the number of the density run.

To simplify, write:

*param_set_*(i, j) for $i = 1, 2$ the number of the density run and $j = 1, 2, 3$ the identifier of the time of day.

All this said - while the code needs to properly utilize parameters from the correct param_set, we note that the only parameters that matter for the ground height determination are the ker-

nel controlling parameters standard deviation, $\sigma = \sigma_{bin}$, and cutoff k in the vertical direction (see section 3.2.2.1, p. 46), and these are constant for all the currently used cases:

$$\sigma_{(i,j)} = \sigma = 3 \tag{26}$$

$$k_{(i,j)} = k = 1 \tag{27}$$

for $i = 1, 2$ and $j = 1, 2, 3$ (see Table in Appendix A and Table 8 and section 17). In section 17 we see that only the quantiles changes for day/night/twilight.

[Calling cutoff=k here for short. This is the variable name k used in papers on the DDA.]

(7) Calculation of the dimension of the kernel:

As described in section (3.2.1.2), p. 48, the dimension of the kernel in the vertical direction is

$$n_{(i,j)} = n = 7 \tag{28}$$

for $i = 1, 2$ and $j = 1, 2, 3$, derived from

$$n_{(i,j)} = n = 2n' + 1 \tag{29}$$

where $n' = 3$ for the vertical half-size of the kernel.

The vertical size of the kernel is 7 bins. This is important to remember.

So far, nothing to code, the above are all things only to remember.

(B) New algorithm steps for ground flag, ground height

Now steps relevant for the calculation of the new variables: To identify ground height, the algorithm relies on the ancillary variable, DEM-height, dem_h, reported in ATL04.

(To identify blowing snow, rely also on the ancillary variable wind velocity (to exceed 5m/s) and identification of regions, including Antarctica, Greenland, mountainous regions, sea ice, small ice sheets in the Arctic. Anywhere where blowing snow can occur.)

Step 1: Proceed per profile, i.e. carry the following steps out for each profile.

We define the set of potential ground-bins, $\mathcal{G}'_1$ as the intersection of mask $\mathcal{M}_1$ and the set of 3 bins above and below the DEM-height:

$$\mathcal{D}_3 = \{dem_h - 3, dem_h - 2, dem_h - 1, dem_h, ..., dem_h + 3\} \tag{30}$$

$$\mathcal{G}'_1 = \mathcal{D}_3 \cap \mathcal{M}_1 \tag{31}$$

Note this yields up to 7 bins per profile and assumes that the DEM is determined correctly.

Step 2:

Rules:

- If $\mathcal{G}'_1$ is the empty set, then move to the next profile.

- If $\mathcal{G}'_1$ is not empty, then assign the *ground_center_bin*, x_{gc} for an index $gc\epsilon\{1, \ldots, 700\}$ as the location (bin) with the highest normalized density value (from density run 1, see equation 15) of all elements in $\mathcal{G}'_1$:

$$x_{gc}\epsilon\mathcal{G}'_1 \wedge \{f_d^{norm}(x_{gc}) \geq f_d^{norm}(x) \quad \forall x_\epsilon \mathcal{G}'_1\} \tag{32}$$

- Then set

$$\mathcal{G}_1 = \{x_{gc}\} \tag{33}$$

This means, the ground set is in this case determined as the point x_{gc} from mask 1.

Clarifications:

Rule

Mask1 is dominant compared to mask2. If ground is found in mask1, its always best.

Consequently:

- If ground is found in equation (8), i.e. if

$$\mathcal{G}_1 = \{x_{gc}\} \neq \emptyset \tag{34}$$

then do not run the ground determination process for the second density run for this profile. Go to Steps 6 and 7. (i.e. determine ground-height and set grd-flag to 1) Move to next profile.

- Else: do steps 4, 5, i.e. run the determination for $\mathcal{G}_2$, and then 6, 7 (if ground is found).

Step 3: *Important:* We can skip this step for the ground height determination.

Remove the NRB value matching point x_{gc} from the NRB set, i.e. set this NRB value to "invalid".

Step 4. Same as Step 2, but now for mask2, from the second density run.

Proceed per profile, i.e. carry the following steps out for each profile.

We define the set of potential ground-bins, $\mathcal{G}'_2$ as the intersection of mask $\mathcal{M}_2$ and the set of 3 bins above and below the DEM-height:

$$\mathcal{G}'_2 = \mathcal{D}_3 \cap \mathcal{M}_2 \tag{35}$$

Note this yields up to 7 bins per profile.

Step 5:

Rules:

- If $\mathcal{G}'_2$ is the empty set, then move to the next profile.

- If $\mathcal{G}'_2$ is not empty, then assign the *ground_center_bin*, x_{gc} for an index $gc\epsilon\{1, \ldots, 700\}$ as the location (bin) with the highest normalized density value (from density run 2, see equation 15) of all elements in $\mathcal{G}'_1$:

$$x_{gc}\epsilon\mathcal{G}'_2 \wedge \{f_d^{norm}(x_{gc}) \geq f_d^{norm}(x) \quad \forall x_\epsilon \mathcal{G}'_2\} \tag{36}$$

- Then set

$$\mathcal{G}_2 = \{x_{gc}\} \tag{37}$$

This means, the ground set is in this case determined as the point x_{gc} from mask 2.

Note that there cannot be a x_{gc} from both masks, because mask1 values are taken out before run2 (colloquially speaking).

Step 6:

Ground height is now defined as the height above WGS84 of the center of the bin x_{gc}. Using center of bin is to match to convention introduced for heights in part1 of the ATBD atmosphere.

Step 7:

Define a ground flag: Set f_{grddda}= ground_flag_dda =0 for all profiles at beginning of the loop over the profiles.

$$\text{If} \quad \mathcal{G}_1 \neq \emptyset \vee \mathcal{G}_2 \neq \emptyset \Rightarrow f_{grddda} = 1 \tag{38}$$

This means, if ground bin found using combined mask then set the flag to 1, i.e. the flag now indicated that ground is found using the DDA-atmos.

18.5 Whither to?

For removing ground layers while retaining the possibility to identify blowing snow or aerosols that start right above the ground, we need to discuss a revision of the layer-identification routine.

For now, we simply implement the code for determination of the ground height value and the ground determination flag.

Figures. The figures in subsection 18.2 are still valid to illustrate the concept of using *mask1* and *mask2* for ground detection.

Blowing snow detection

Section 13.2 stays in place, essentially.

18.6 Results

We now have

(a) A ground flag from DDA (both runs, both masks)

(b) Ground height.

Both are worth reporting on the product. These will be useful for the surface-product user community.

18.7 Consequences/Applications of Ground Detection (2019-06-21)

Implementation of the above sections is currently underway, as of writing of ATBD II, v11.0 (2020-Jan-16), for ASAS code version v4.0. For the next code development steps, there are several algorithm options for the classification of near-surface atmospheric layers, including blowing snow and aerosols, and surface, described here.

Background: It is not entirely trivial to clearly identify the part of the returned signal that is coming from ground, as opposed to near ground atmosphere.

Algorithm development: The ground detection yields several options for applications to downstream coding. The possibilities are described below, with implications and notes on potential advantages/disadvantages.

Note: For all three options, we still need to discuss a revision of the layer-identification routine, if we want to be able to identify blowing snow or aerosols that start right above the ground.

18.7.1 Definition of a Ground Mask (ground_mask), to be applied to NRB Values to Ignore Ground-Center-Bins in Kernel Calculation

Subsection (1.2.2) of this document titled *"Algorithm for Cloud-Based Ground-Detection Flag (DDA-atmos) and Ground Height Determination"*, Step 3 already says to mask the ground bin in the NRB data set. This can be coded as another mask, to avoid changing the NRB values.

Step 1. Define a ground mask (from both runs), as in the above steps.

Step 2. For a grd-bin (value 1 in grd mask), ignore this bin in the DDA kernel calculation.

Step 1. In detail, define a ground mask (from both runs), as follows:

grd_mask is a binary data field, of size 700.

Loop through profiles. For a given profile:

Set $grd_mask(i) = 0$ for $i = 1, \ldots, 700$

If $f_{grddda} = 1$ then $grd_mask(gc) = 1$, noting that $gc \epsilon \{1, \ldots, 700\}$ is the index of the ground-center bin (determined for the second or first mask, as described in Step 2 or Step 5 of (1.2.2), respectively).

Or more simply:

If bin-number is equal to ground-center-bin-number, then ignore this bin in the DDA kernel calculation, in both runs.

Note on algorithm flow:

- First, determine the ground-center-bin from both DDA density runs.
    - Run DDA as normal, run1.
    - Determine ground-center-bin from run 1.
    - Run DDA as normal, run2.
    - If $f_{grddda} = 0$, determine ground-center-bin from run 2.
- Define ground mask (from both runs).
- Apply ground mask to NRB values in density-run 1. Otherwise, run DDA as usual.
- Apply ground mask to NRB values in density-run 2. Otherwise, run DDA as usual.
- Determine atmos layers and their tops, bottoms, as usual.

Discussion:

- This will be the best way to ignore the ground signal, because there is no spreading of the ground signal into neighboring bins. It is expected to yield the best results, from all we know and have discussed, and allow for straightforward implementation of aerosols, low arctic fog, diamond dust and blowing snow, which would all have different thresholds in a classification algorithm (to be specified later).
- Note that there may be cases where the ground signal affects two bins in the same profile (or 3, according to Steve Palm, if topographic relief is very high); we found these cases to be rare enough to ignore them for now.
- Is it computationally prohibitively expensive to do density runs just to determine the ground-center-bins?

18.7.2 Classification of the Ground Layer, Using a Kernel-Height Sized Ground Mask (fat_ground_mask) for a Simplified Interim Solution

The follwoing is a computationally more lightweight option, while still better than "removing" (a StevePalm terminology) entire layers, as soon as they may intersect with ground.

Concept: Identify ground-center-bin, then include kernel-height bins in the ground layer. This has an error of n' , with $n' = 3$ for the current selection of parameters) or 90m. The ground layer height will be $n = 7$ (see eqns. 3, 4).

Algorithm: If ground center bin stems from density-run1, i.e. if

$$\mathcal{G}_1 = \{x_{gc}\} \neq \emptyset \tag{39}$$

then label it after run 1: $x_{gc} = x_{gc_1}$.

If ground center bin stems from density-run2, i.e. if

$$\mathcal{G}_2 = \{x_{gc}\} \neq \emptyset \tag{40}$$

then label it after run 2: $x_{gc} = x_{gc_2}$.

Define fat_grd_mask as a binary array, Set $fat_grd_mask(j) = 0$ for $j = 1, \ldots, 700$

If $f_{grddda} = 1$ then

[For i=1,2:

$fat_grd_mask(j) = 1$ for $j \epsilon \{gc_i - n'_i, ..., gc_i + n'_i\}$

]

This will set 7 bins to 1, centered at ground-center-bin, for current parameter selections.

Note on algorithm flow:

- Run DDA as normal, run1.
- Determine ground-center-bin from run 1.
- Run DDA as normal, run2.
- If $f_{grddda} = 0$, determine ground-center-bin from run 2.
- If $f_{grddda} = 1$, define fat_ground_mask (from the appropriate run, i).
- Apply fat_ground_mask: Intersect mask_i and fat_ground_mask to create grd_reduced_mask. Move resultant bins into a ground layer.
- Form combined_mask_reduced, using the possibly existing grd_reduced_mask.
- Apply existing layer determination algorithm with the 3-bin amalgamation rule to combined_mask_reduced.

Discussion:

- This option is faster to implement than the options described in (18.7.1).
- The advantage over removing entire layers is that aerosols, blowing snow, diamond dust and low, ground-touching clouds will still remain a cloud layer, whenever they are taller than 3 bins (90m) above the ground.

18.7.3 Utilizing Layer Counting Rather Than Ground-Layer Classification as an Interim Solution

This is the easiest solution. Set layer-number=number-of-layers-that-are-filled, and go "if $f_{grddda} = 1$ then layer-number=layer-number-1". This would solve Yuekui's problem of mapping cloud existence compared to ASR in the clear-sky figures presented at a recent meeting and telecon.

Algorithm Flow:

- Run DDA as normal.
- Determine the ground-center-bin from both DDA density runs.
  - Run DDA as normal, run1.
  - Determine ground-center-bin from run 1.
  - Run DDA as normal, run2.
  - If $f_{grddda} = 0$, determine ground-center-bin from run 2.
- Apply existing layer determination algorithm with the 3-bin amalgamation rule to combined_mask.
- Define *layerno* as the number of layers detected by the DDA.

  If $f_{grddda} = 1$ then set $layerno = layerno - 1$.

Discussion:

- This option does not require changing the atmospheric layer concept for now, while allowing to map clear sky versus cloudy conditions.

19 Sensitivity Study to Optimize Parameters for the First Public Release of ICESat-2 Data Products (ASAS code v5.1; 951 data)

Summary. ICESat-2 Atmospheric Data Products will be released together with other data products in the first public release, planned for late May 2019. In preparation for this release, results of ASAS code v5.1 (v951 data) are analyzed in a new sensitivity study. Few changes in the input data in ATL04 resulted in a small parameter change for night-time data, while parameter sets for day-time data and twilight data remained the same. The change of parameters for night-time data is prompted by the analysis of aerosol layers compared to other tenuous layers.

Description

The sensitivity study was run for v951 data, using parameter sets (t74, t80-t89). Parameter sets are listed in Table 7.

Algorithm run on sub-sampled regions in the file
ATL04_20181017T002107_02810101_951_01.h5
using returns from profile 3 of the ATLAS beam configuration. regions subsampled to represent different types of clouds and aerosols. All regions are for night-time data.

The following steps are illustrated for each parameter set shown in Figure 47 (by rows):

1. Raw NRB data (valid bins)[ATL04 input data to the DDA-atmos]; Cloud Layer Boundaries over Raw NRB Data
2. Pass 0: Kernel Matrix; Pass 1: Kernel Matrix
3. Pass 0: Density; Pass 1: Density
4. Pass 0: Thresholds Along Track; Pass 1: Thresholds Along Track
5. Pass 0: Density - Thresholded; Pass 1: Density - Thresholded
6. Pass 0: Density - Declustered (Mask 1); Pass 1: Density - Declustered (Mask 2)
7. Pass 0: Final Mask with Density (Combined Mask); Q/A: Half-Gap Confidence

NOTE: Figure 47 is located in Appendix S (Sensitivity Studies with Figures, Section 19).

Results:

(I) Continue to use double-density runs.

(II) Continue to use three parameter sets for day-time/night-time/twilight.

(1) **For day-time: Use paramset1=t60**
(as before, section 17).

(2) **For night-time: Use paramset2=t87**
Use **(t87)** with quantile $q_1 = 0.97$ and quantile $q_2 = 0.55$ for night-time conditions.
Previously used **(t74)** with quantile $q_1 = 0.99$ and quantile $q_2 = 0.55$ for night-time conditions.

(3) **(3) For twilight: Use paramset3=t60**
(as before, section 17).

Reasoning: Parameter set (t87) shifts detection of a more versatile set of clouds into the run-1 group of optically thick clouds and thus enhances the range of clouds and aerosols that can be discriminated in the second density run. While the parameter change appears to be small, the effect is important "down the road" for the detection of layer variability within tenuous layers and aerosols, as opposed to low-level spatial variability in intensity. Using the term "intensity" for values several variables: NRB values and density fields.

See the next section (under construction).

20 Thoughts about Layers and Detection Capability of ICESat-2 ATLAS: Challenges and Opportunities for Atmospheric Research

This section is under construction for version v10.0 of this ATBD and will be finalized for v10.1.

The first subsection follows a specific, technical line of thought to solve a specific problem. The second subsection is the result of some bigger-picture considerations concerning the possibilities of the sensor.

20.1 Solution for the "bubbly regions" problem in aerosols and other tenuous layers

Problematic Topic ("bubbly regions"): In case of low-level spatial variability in intensity within an aerosol layer, the current algorithm can identify too many layers within the aerosol layer. In the case shown here towards the end of the test data set (Figure 47, along-track distance above 4000000), the NRB values as well as the density fields reveal a "bubbly region" (short for "region of low-level spatial variability in intensity"). The current algorithm employed for assigning layer boundaries to the combined final mask identifies the layer boundaries, whenever a layer is at most three bins thick and separated by at least three bins from the neighboring layers (simplification; the full algorithm is described in section (3.6) of this ATBD). As results from pass1 (2nd pass) of the density field indicate, there are a few different regions within the aerosol layer, but not as many as the number of layer boundaries suggest.

Solution for Problematic Topic ("bubbly regions"): Analysis of the Q/A figures in Figure 47 and other sensitivity studies indicates that a threshold-criterion put on the half-gap Q/A measure will provide a solution to this problem: Q/A is generally very low (0.1) where bubbly regions exist, but higher for acceptable layer boundaries (0.3 or 0.4 in all data sets analyzed so far.)

Implementation of a threshold that is based on the Q/A measure requires implementation of the Q/A measure. This will require a different layer-determination algorithm.

20.2 Thoughts about layers and detection capability of ICESat-2 ATLAS

Some of the understanding of atmospheric data products derived from atmospheric lidar observations is driven by the potential and limitations of exisiting data from previous (and current) satellite and airborne sensors. The ICESat-2 ATLAS data appear to have a higher sensitivity to small-scale spatial variability in the atmosphere. The final product in the ATL09 data products is, so far, the set of layer boundaries (layer bottom and layer top for several cloud, aerosol, blowing snow and other layers), which is a customary product for atmospheric lidar data that has been used by the atmospheric science community.

Some of our first results indicate that we are able to observe smaller scale features within a given atmospheric layer, and these features resemble sub-layer features in reality.

Algorithm Result: Topic 1: The **Combined_Mask** needs to be put onto the product.

This is the mask that is the combinations of mask1 and mask2, from the two density runs, each calculated after thresholding and declustering. The combined mask is shown in most sensitivity analysis figures. The Combined_Mask captures the spatial variability of the ATLAS data after separation of signals and artefacts and thus is expected to provide a high-resolution product for atmospheric research.

The Combined_Mask is not identical with the **pre_layer_mask**. A pre_layer_mask is calculated from the Combined_Mask as part of the layer boundary determination algorithm (section (3.6)). The algorithm for this will change in connection with the full version of section (21.1) in v10.1. However, the layer detection algorithm includes considerable eliminiation of certain features of interest and amalgamates neighboring atmospheric regions of potentially different types.

21 Coder's Corner and Known Issues in ATL09 Atmospheric Data Products (Related to ATBD Part II)

21.1 Summary

This section covers imperfections in the atmospheric data products, which fall into three groups:

(1) Differences between the Geomath group's developer code and the ASAS v5.3 code for atmospheric data products

At time of the first public release of the data products, implementation of some components of the algorithm described here (DDA-atmos) into ASAS code had not been fully completed. The main reason is that code changes required after launch for earlier data products (ATL02, ATL04) take precedence over finalizing code for later data products (in the sense of ICESat-2 data product processing chain). Related algorithm steps for the DDA-atmos are being implemented continuously by SIPS into ASAS code for the next releases. Current release v5.3 includes updates as described below.

(2) Open problems

21.2 Code changes implemented in ASAS v5.3

(1) Section (11) Quality assessment: Half-gap confidence has been implemented and verified.

(2) Flow of mask handling, described in section (3.10), has been implemented/ improved and verified.

(3) Neighborhood of the kernel and normalization in the density calculation have been updated.

(4) Folding over of kernel has been eliminated.

21.3 Code changes that are still outstanding or in progress

(1) Changed section 18 on a cloud-based ground flag and on ground detection (surface detection) using a new algorithm based on the DDA. This entire block is in progress for v5.4.

(2) Section 13 on blowing snow and ground has not been implemented yet. This also needs to be updated, following completion of the implementation of section 18 on ground determination and ground flags, which is currently in progress.

(3) Components of mask handling, described in section (3.10), may have to tested and verified for density run 2.

21.4 Open Problems

(1) Section 20 needs to be finalized.

(2) Other things we have to output: Layer Density, Density Sum per Vertical Profile and Other Derived Parameters (see section (3.6)).

(3) Ground and Cloud. The last step (4) in 13.4 does not address the vertical width of the surface layer nor the problem with the 3 times 30m bins in layer smoothing, which is performed in the algorithm module of layer boundary determination (see section (3.6)). Theoretically, this problem should not come up for ground alone, but one needs to make sure that it cannot occur numerically. However, if blowing snow is present or another thick aerosol layer that connects to the ground without a 3-bin (90m) gap, then the layer smoothing algorithm in (3.6) will amalgamate ground and near-ground blowing snow into a single layer.

For this reason, we suggest that the ground surface should be classified as such after calculation of mask 1 and mask 2, but not joined into the same layer as potentially existing blowing snow.

(4) An ad-hoc algorithm for identification was provided in ATBD, Part I, for interim implementation in the ATL09 product. The ad-hoc algorithm will be replaced by the proper algorithm anyways. Mentioned here in case it creates inconsistencies with the DDA-atmos.

(5) The atmosphere algorithm family is designed with a minimal layer height of 90m (3bins) and minimal layer separation of 90m (3bins) in mind. Current data products based release v5.3 are produced with minimal layer height of 20 bins and minimal layer separation of 4 bins. This is an ad-hoc fix intended to suppress reporting of internal-layer variability, but as a result, layers are lost.

1. A criterion based on half-gap confidence will be implemented to reduce reporting of internal-layer variability.

Acknowledgements

The assistance of Brian McDonald, Bruce Wallin, Thomas Trantow, Gavin Medley, Levi Kurlander, Alec Stiller, Adam Hayes, Connor Myers, Tasha Markley and Mohamed Al-rasbi, all Geomathematics Group, Department of Electrical, Computer and Energy Engineering, University of Colorado Boulder, in implementation and testing of code for the density-dimension algorithm in part II is thankfully acknowledged. Discussions with Mark Vaughan, NASA Langley Research Center, Hampton, VA, on improvements of this ATBD are gratefully appreciated. Discussions with David Hancock, Jeff Lee, Jesse Wimert and Kristine Barbieri during code implementation at the SIPS are equally appreciated and have resulted in improvements of the description. Work supported by NASA Cryospheric Sciences and the ICESat-2 Project.

References

M. D. Buhmann. *Radial basis functions: theory and implementations*, volume 12. Cambridge University Press, Cambridge, England, 2003.

U. Herzfeld, B. McDonald, B. Wallin, T. Markus, T. Neumann, and A. Brenner. An algorithm for detection of ground and canopy cover in micropulse photon-counting lidar altimeter data in preparation of the ICESat-2 mission. *IEEE Transactions Geoscience and Remote Sensing*, 54(4): 2109–2125, 2014. doi: 10.1109/TGRS.2013.2258350.

U. Herzfeld, T. Trantow, D. Harding, and P. Dabney. Surface-height determination of crevassed glaciers — Mathematical principles of an Auto-Adaptive Density-Dimension Algorithm and validation using ICESat-2 Simulator (SIMPL) data. *IEEE Transactions in Geoscience and Remote Sensing*, 55(4):1874–1896, 2017. doi: 10.1109/TGRS.2016.2617323.

Ute Herzfeld and Stephen Palm. *ICESat-2 Algorithm Theoretical Basis Document for the Atmosphere, Part II: Detection of Atmospheric Layers and Surface Using a Density Dimension Algorithm, v7.1, September 23, 2016.* NASA ICESat-2 Project, 2016a. 163p.

Ute Herzfeld and Stephen Palm. *ICESat-2 Algorithm Theoretical Basis Document for the Atmosphere, Part II: Detection of Atmospheric Layers and Surface Using a Density Dimension Algorithm, v7.0, August 24, 2016.* NASA ICESat-2 Project, 2016b. 163p.

Ute Herzfeld and Stephen Palm. *ICESat-2 Algorithm Theoretical Basis Document for the Atmosphere, Part II: Detection of Atmospheric Layers and Surface Using a Density Dimension Algorithm, v8.0, November 17, 2017.* NASA ICESat-2 Project, 2017. 237p.

Ute Herzfeld and Stephen Palm. *ICESat-2 Algorithm Theoretical Basis Document for the Atmosphere, Part II: Detection of Atmospheric Layers and Surface Using a Density Dimension Algorithm, v8.1, March 21, 2018.* NASA ICESat-2 Project, 2018a. 255p.

Ute Herzfeld and Stephen Palm. *ICESat-2 Algorithm Theoretical Basis Document for the Atmosphere, Part II: Detection of Atmospheric Layers and Surface Using a Density Dimension Algorithm, v9.0, December 20, 2018.* NASA ICESat-2 Project, 2018b. 288p.

Ute Herzfeld and Stephen Palm. *ICESat-2 Algorithm Theoretical Basis Document for the Atmosphere, Part II: Detection of Atmospheric Layers and Surface Using a Density Dimension*

Algorithm, v10.0, May 1, 2019. NASA ICESat-2 Project, 2019. doi: 10.5067/TE10T7E5TNGK. 328p.

Ute Herzfeld, Stephen Palm, and Yueki Yang. *ICESat-2 Algorithm Theoretical Basis Document for the Atmosphere, Part II: Detection of Atmospheric Layers and Surface Using a Density Dimension Algorithm, v6.0, October 31, 2015*. NASA ICESat-2 Project, 2015. 153p.

Ute Herzfeld, Stephen Palm, and David Hancock. *ICESat-2 Algorithm Theoretical Basis Document for the Atmosphere, Part II: Detection of Atmospheric Layers and Surface Using a Density Dimension Algorithm, v11.0, February 6, 2020*. NASA ICESat-2 Project, 2020. doi: 10.5067/TE10T7E5TNGK. 374p.

Stephen Palm, Ute Herzfeld, and Yueki Yang. *ICESat-2 Algorithm Theoretical Basis Document for the Atmosphere, Part I: Level 2 and Level 3 Data Products, v7.3, November 15, 2017*. NASA ICESat-2 Project, 2017a. 79p.

Stephen Palm, Vinay Kayetha, Yuekui Yang, and Rebecca Pauly. Blowing snow sublimation and transport over antarctica from 11 years of calipso observations. *The Cryosphere*, pages 2555–2569, 2017b. doi: 10.5194/tc-11-2555-2017.

Stephen Palm, Ute Herzfeld, and Yueki Yang. *ICESat-2 Algorithm Theoretical Basis Document for the Atmosphere, Part I: Level 2 and Level 3 Data Products, v8.0, December 2018*. NASA ICESat-2 Project, 2018a. 79p.

Stephen Palm, Vinay Kayetha, and Yuekui Yang. Toward a satellite-derived climatology of blowing snow over antarctica. *Journal of Geophysical Research*, 2018b. doi: 10.1029/2018JD028632.

Stephen Palm, Yuekui Yang, Vinay Kayetha, and Julien Nicolas. Insight into the thermodynamic structure of blowing-snow layers in antarctica from dropsonde and calipso measurements. *Journal of Applied Meteorology and Climatology*, 2018c. doi: 10.1175/JAMC-D-18-0082.1.

Steven Palm, Yuekui Yang, and Ute Herzfeld. *ICESat-2 Algorithm Theoretical Basis Document for the Atmosphere, Part I: Level 2 and 3 Data Products, August 19, 2019*. NASA ICESat-2 Project, 2019. doi: 10.5067/X6N528CVA8S9. 104p.

B.E. Schutz, H.J. Zwally, C.A. Shuman, D. Hancock, and J.P. DiMarzio. Overview of the ICESat Mission. *Geophysical Research Letters*, 32(21), 2005. doi: 10.1029/2005GL024009.

H.J. Zwally, M. Giovinetto, J.Li, H. Cornejo, M.Beckley, A. Brenner, J. Saba, and D. Yi. Mass changes of the Greenland and Antarctic ice sheets and shelves and contributions to sea-level rise: 1992–2002. *Journal of Glaciology*, 51(175):509–527, 2005.

Appendix

A Variables calculated in this ATBD Atmosphere, Part II

Common Variable Name used in ATBD	Formal Variable Name used in ATL09 Product	Value	Variable Type
Sigma	sigma1, sigma2	3, 3	float
Anisotropy	a_m1, a_m2	10, 20	float
Cutoff	cutoff1, cutoff2	1, 1	float
Downsampling	downsample1, downsample2	1, 1	integer
Minimum Cluster Size	size_threshold1, size_threshold2	300,600	integer
Threshold Bias	thresh_bias1, thresh_bias2	10E+14, 10E+14	float
Threshold Factor	thresh_sensitivity1, thresh_sensitivity2	0.9, 1	float
Threshold Window	threshold_segment_length1, threshold_segment_length2	2, 2	integer
Quantile	quantile1, quantile2	0.99, 0.8	float
Minimum bin size for layer separation	layer_sep	3	integer
Minimum bin size for cloud thickness	layer_thick	3	integer
Maximum cloud layers in profile	max_layer	10	integer
Number of Algorithm Passes	num_passes	2	integer

Table 9: Algorithm parameter names in the DDA-atmos algorithm.

Common Variable Name used in ATBD	**Formal Variable Name used in ATL09 Product**	**Variable Type**
Density	density_pass1, density_pass2	$m \times n$ float array
Bottom Layer Height	layer_bot	$m \times$max_layer float array
Top Layer Height	layer_top	$m \times$max_layer float array
Layer Density	layer_dens	$m \times$max_layer float array
Layer Confidence	layer_con	$m \times$max_layer float array
Surface Height from Density	surface_h_dens	$m \times 1$ float array
Blowing Snow Density	bsnow_dens	$m \times 1$ float array
Height of Blowing Snow Top from Density	bsnow_h_dens	$m \times 1$ float array

Table 10: Output parameter names in the DDA-atmos algorithm. m = number of profiles, n = number of vertical bins in a single profile.

B Index

3-bin confidence – See definition

anisotropy factor – DDA parameter (a, a_m), see (M.2) and (3.2.1.1). For algorithm parameters in general see (3.9) and Table 2d

anisotropy norm – See (M.2)

atmospheric layers – See (3.6)

autoadaptive thresholding – See (3.3.6.3)

confidence – part of Q/A (11)

cloud boundaries – See (3.6)

cloud mask – See (3.5)

clusters – See (M.6) and (3.4)

cutoff – DDA parameter ($cutoff$), see (3.2.1.1). For algorithm parameters in general see (3.9) and Table 2d

density dimension – See (3.2) and (3.3)

Density-Dimension Algorithm (DDA) – See (2) and (3)

double-density run – See (10.1)

downsampling – DDA parameter (*downsampling*, d) originally from Method A, but also implemented in the final Method A/B. See (3.3.1). For algorithm parameters in general see (3.9) and Table 2d

final cloud mask – See (3.5)

gaussian kernel – See equation 13 and (3.2.1.1)

GLAS-based simulations – See (8.1)

half-gap confidence – See definition

kernel – See equation 13 and (3.2.1.1)

MABEL-based simulations – Simulated ICESat-2 data based on MABEL data. See (6)

M-ATLAS data – Simulated ICESat-2 data based on MABEL data. See (6)

minimum cluster size – DDA parameter (*min cluster size*, *size_threshold*), see (3.4). For algorithm parameters in general see (3.9) and Table 2d

neighborhood – DDA parameter ($\mathcal{D}_c$). See (M.3) and (3.2.2). For algorithm parameters in general see (3.9) and Table 2d

noise filter – See (M.4)

Normalized Relative Backscatter (NRB) – See (1.1)

optical thickness – The degree to which a cloud modifies the light passing through it. For discrimination of optically thick and optically thin clouds, see (M.5)

range correction – Algorithm option to correct M-ATLAS data. See (6.1)

quantile – A DDA parameter (*quantile*, q) used in threshold determination (3.3.6.3). For the quantile calculation see (3.3.7). For algorithm parameters in general see (3.9) and Table 2d

radial basis function (RBF) – See (M.1)

search neighborhood – DDA parameter ($\mathcal{D}_c$). See (M.3) and (3.2.2). For algorithm parameters in general see (3.9) and Table 2d

power correction – Algorithm option to correct M-ATLAS data. See (6.1)

sigma – DDA parameter (σ), see (M.1). For algorithm parameters in general see (3.9) and Table 2d

single-density run – See (10.1)

small clusters – See (M.6) and (3.4)

tenuous clouds – Optically thin clouds, see (M.5)

t3 – double-density run deemed best in the 2016 state-of-the-art simulated GLAS-based ICESat-2 type data, introduced here

t8 – single-density run deemed best in the 2016 state-of-the-art simulated GLAS-based ICESat-2 type data, see Table 5 for parameter values

t54 – single-density run, best parameter combination for analysis of 2017-Oct version of GLAS-based simulated ATL04 data, see Table 5 for parameter values

t56 – double-density run, best parameter combination for analysis of 2017-Oct version of GLAS-based simulated ATL04 data, see Table 7 for parameter values

t60 – double-density run, parameter combination used for day-time and twilight-time data for ICESat-2 postlaunch releases (ASAS code versions v2.0 and v3.0) see Table 7 for parameter values

t64 – double-density run, alternative best parameter combination for analysis of 2017-Oct version of GLAS-based simulated ATL04 data, see Table 7 for parameter values

t74 – double-density run, parameter combination used for night-time data for ICESat-2 postlaunch releases (ASAS code versions v2.0 and v3.0) see Table 7 for parameter values

threshold – See (3.3.6.3)

threshold bias [offset] – A DDA parameter ($threshold_bias$, T) used in threshold calculations. See (3.3.6.3). For algorithm parameters in general see (3.9) and Table 2d

threshold sensitivity – A DDA parameter ($threshold_sensitivity$, t) used in threshold calculations. See (3.3.6.3). For algorithm parameters in general see (3.9) and Table 2d

C Abbreviations

ASAS – **A**TLAS **S**cience **A**lgorithm **S**oftware. The ASAS is the science algorithm component of the ICESat-2 Science Investigator-Led Processing System (SIPS).

ATLAS – **A**dvanced **T**opographic **L**aser **A**ltimeter **S**ystem, the instrument aboard ICESat-2.

ATL02 Product – Science Unit Converted Telemetry. Photon time of flight, corrected for instrument effects. Includes all photons, pointing data, spacecraft position, housekeeping data, engineering data, and raw atmospheric profiles, segmented into several minute granules.

ATL04 Product – Uncalibrated Backscatter Profiles. Along-track atmospheric backscatter data, 25 times per second. Includes calibration coefficients for polar regions. Segmented into several minute granules.

ATL09 Product – Calibrated backscatter and cloud characteristics. Includes atmospheric layer heights, blowing snow, integrated backscatter and optical depth.

DDA – **D**ensity-**D**imension **A**lgorithm

GLAS – **G**eoscience **L**aser **A**ltimeter **S**ystem

ICESat-2 – **I**ce, **C**loud, and land **E**levation **Sat**ellite **2**, part of NASA's Earth Observing System

MABEL – **M**ultiple **A**ltimeter **B**eam **E**xperimental **L**idar, an airborne simulator instrument for ICESat-2 ATLAS

NRB – **N**ormalized **R**adiative **B**ackscatter

RBF – **R**adial **B**asis **F**unction

SDT – **S**cience **D**efinition **T**eam (NASA)

SIPS – **S**cience **I**nvestigator-Led **P**rocessing **S**ystem

ST – **S**cience **T**eam (NASA)

APPENDIX S: Sensitivity Studies with Figures

7 Sensitivity Studies (for 2013 M-ATLAS Data)

The sensitivity studies are carried out to analyze and illustrate the effects of varying the parameters and prepare for analysis of future ICESat-2 data that may have different characteristics.

7.1 Sensitivity Studies for Single-Density Runs

To demonstrate the effects of the primary parameters, experiments with single-density are given in section (7.1) The following experiments are carried out: Experiment 1: Changing the neighborhood (Figure 20),
experiment 2: changing σ (Figure 21),
and experiment 3: changing anisotropy (for two different neighborhoods, Figure 22).

In detail, the following experiments are carried out:

Experiment 1: Changing the neighborhood r (Figure 20)

Fig. 20a: $\sigma = 5$, $a_m = 10$, r=2, kernel (5,5)

Fig. 20b: $\sigma = 5$, $a_m = 10$, r=3, kernel (7,7)

Fig. 20c: $\sigma = 5$, $a_m = 10$, r=4, kernel (9,9)

Fig. 20d: $\sigma = 5$, $a_m = 10$, r=5, kernel (11,11)

Fig. 20e: $\sigma = 5$, $a_m = 10$, r=6, kernel (13,13)

Fig. 20f: $\sigma = 5$, $a_m = 10$, r=7, kernel (15,15)

Fig. 20g: $\sigma = 5$, $a_m = 10$, r=8, kernel (17,17)

Fig. 20h: $\sigma = 5$, $a_m = 10$, r=10, kernel (21,21)

Experiment 2: Changing σ (Figure 21)

Fig. 21a: r=5, kernel (11,11), $a_m = 10$, $\sigma = 1$

Fig. 21b: r=5, kernel (11,11), $a_m = 10$, $\sigma = 2$

Fig. 21c: r=5, kernel (11,11), $a_m = 10$, $\sigma = 3$

Fig. 21d: r=5, kernel (11,11), $a_m = 10$, $\sigma = 4$

Fig. 21e: r=5, kernel (11,11), $a_m = 10$, $\sigma = 6$

Fig. 21f: r=5, kernel (11,11), $a_m = 10$, $\sigma = 8$

Fig. 21g: r=5, kernel (11,11), $a_m = 10$, $\sigma = 16$

Fig. 21h: r=5, kernel (11,11), $a_m = 10$, $\sigma = 140$

Experiment 3: Changing anisotropy a_m (with r=5 and r=10) (Figure 22)

Fig. 22a: $\sigma = 5$, $a_m = 3$, r=5, kernel (11,11)

Fig. 22b: $\sigma = 5$, $a_m = 3$, r=10, kernel (21,21)

Fig. 22c: $\sigma = 5$, $a_m = 10$, r=5, kernel (11,11)

Fig. 22d: $\sigma = 5$, $a_m = 10$, r=10, kernel (21,21)

Fig. 22e: $\sigma = 5$, $a_m = 20$, r=5, kernel (11,11)

Fig. 22f: $\sigma = 5$, $a_m = 20$, r=10, kernel (21,21)

Fig. 22g: $\sigma = 5$, $a_m = 30$, r=5, kernel (11,11)

Fig. 22h: $\sigma = 5$, $a_m = 30$, r=10, kernel (21,21)

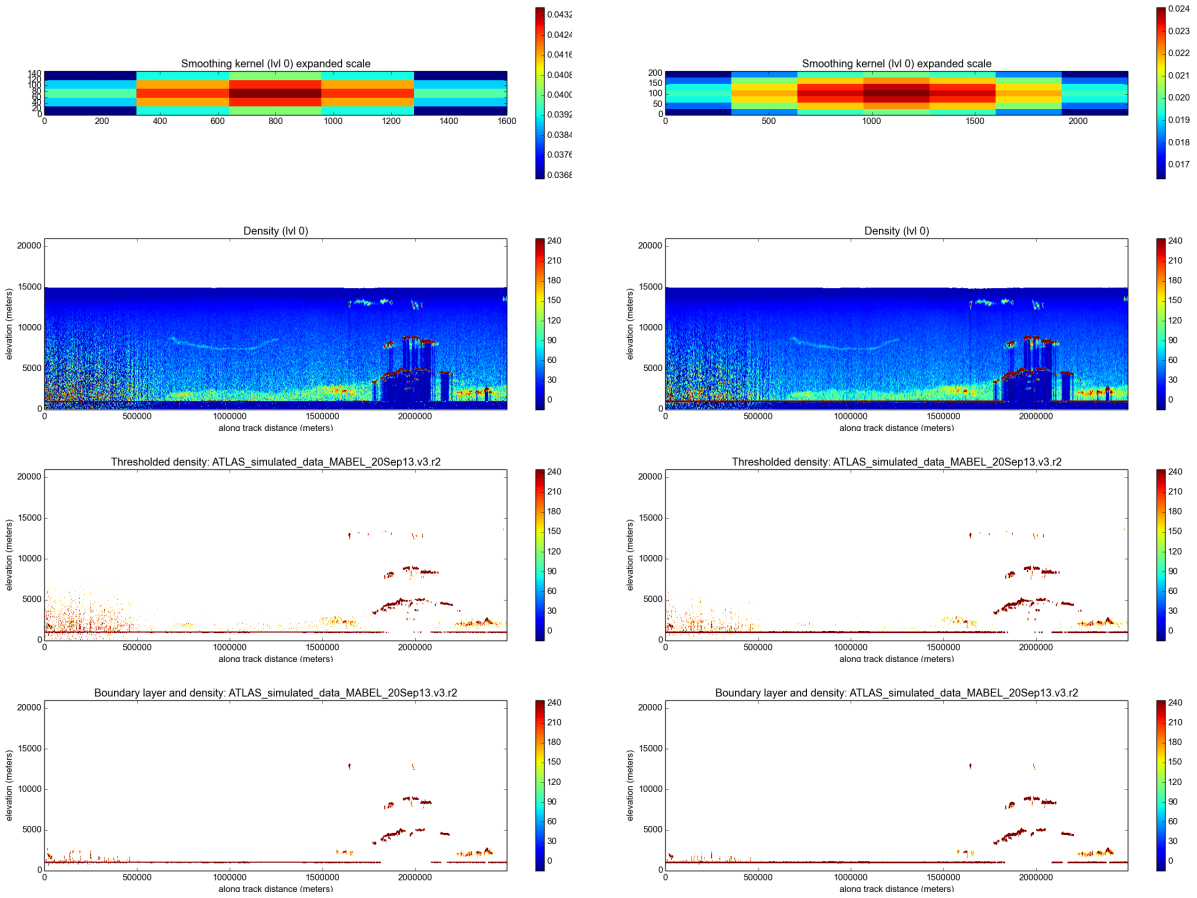
All experiments use the following parameters: cutoff=2 (i.e. 2 standard-deviations), minimal cluster size not removed: 600,

For each experiment and sub experiment, the following resultant figure panels are given from top to bottom:

(1) weight matrix (kernel),

(2) density,

(3) preliminary cloud mask after application of thresholds, with density values within cloud areas, and (4) density in cloud areas, for final cloud mask after application of small-cluster removal.



a) neighborhood = 2

b) neighborhood = 3

Figure 20. Sensitivity Experiment 1: Changing neighborhood

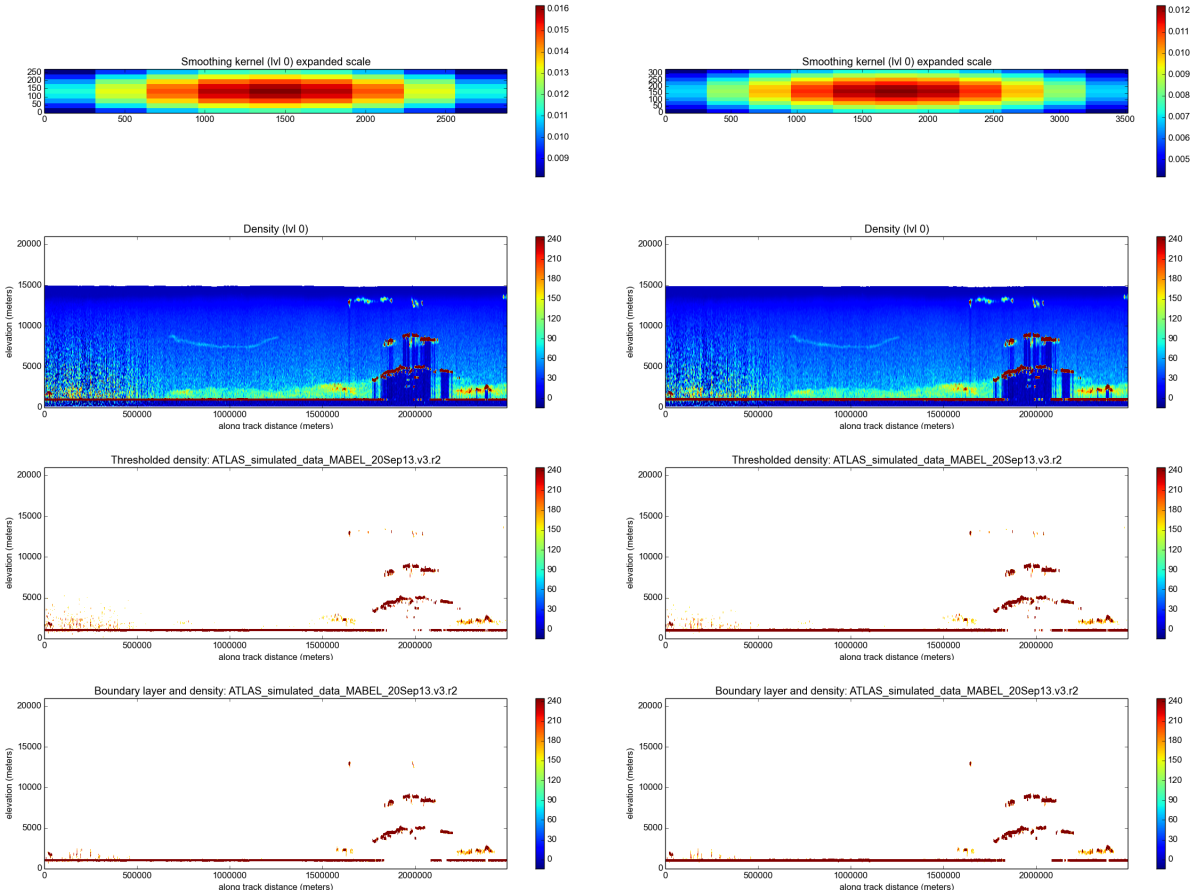
Analysis of day-time data (dataset: ATLAS_simulated_data_MABEL_20Sep13.v3.r2, based on MABEL atmosphere data collected 20Sep13).

$\sigma = 5$

$a_m = 10$

(a) neighborhood = 2 (5x5 kernel)

(b) neighborhood = 3 (7x7 kernel)



c) neighborhood = 4

d) neighborhood = 5

Figure 20, ctd. Sensitivity Experiment 1: Changing neighborhood

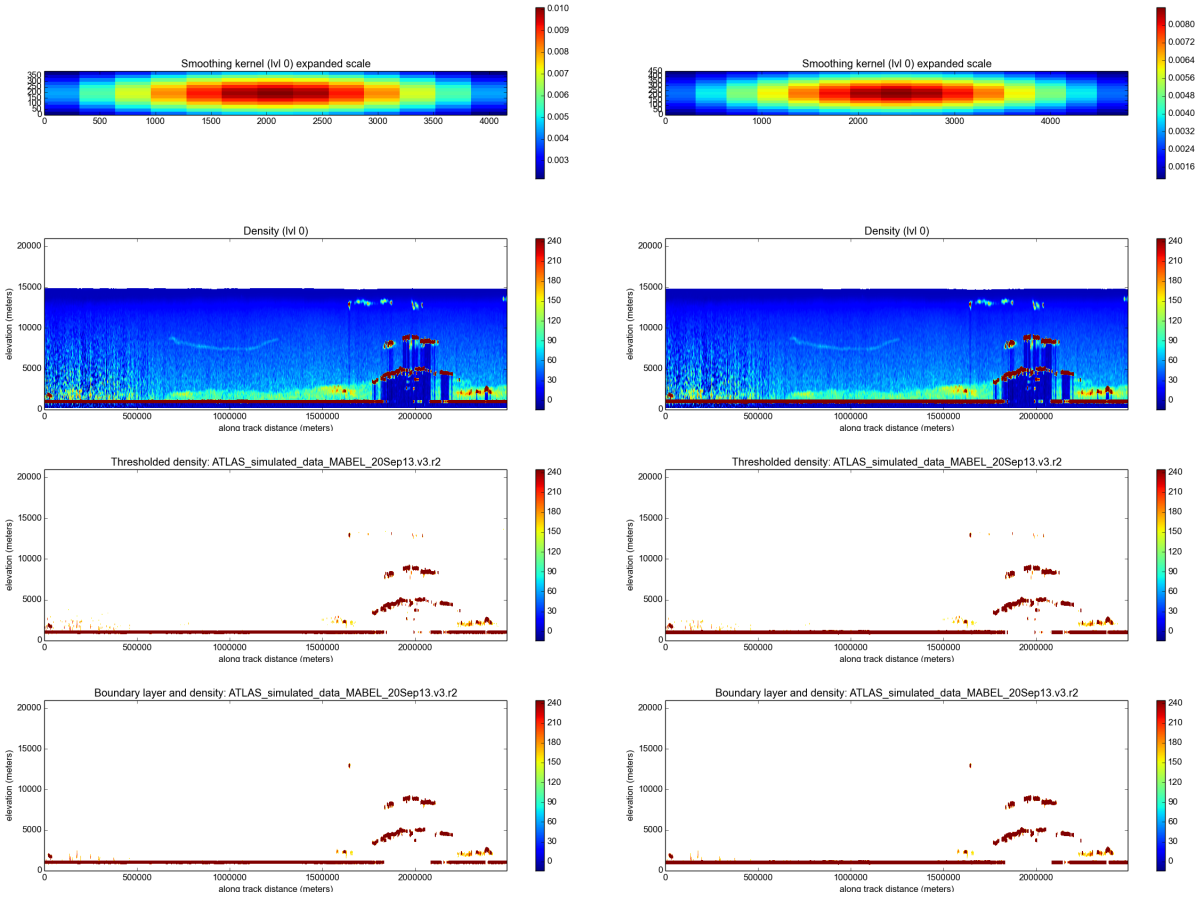
Analysis of day-time data (dataset: ATLAS_simulated_data_MABEL_20Sep13.v3.r2, based on MABEL atmosphere data collected 20Sep13).

$\sigma = 5$

$a_m = 10$

(c) neighborhood = 4 (9x9 kernel)

(d) neighborhood = 5 (11x11 kernel)



e) neighborhood = 6

f) neighborhood = 7

Figure 20, ctd. Sensitivity Experiment 1: Changing neighborhood

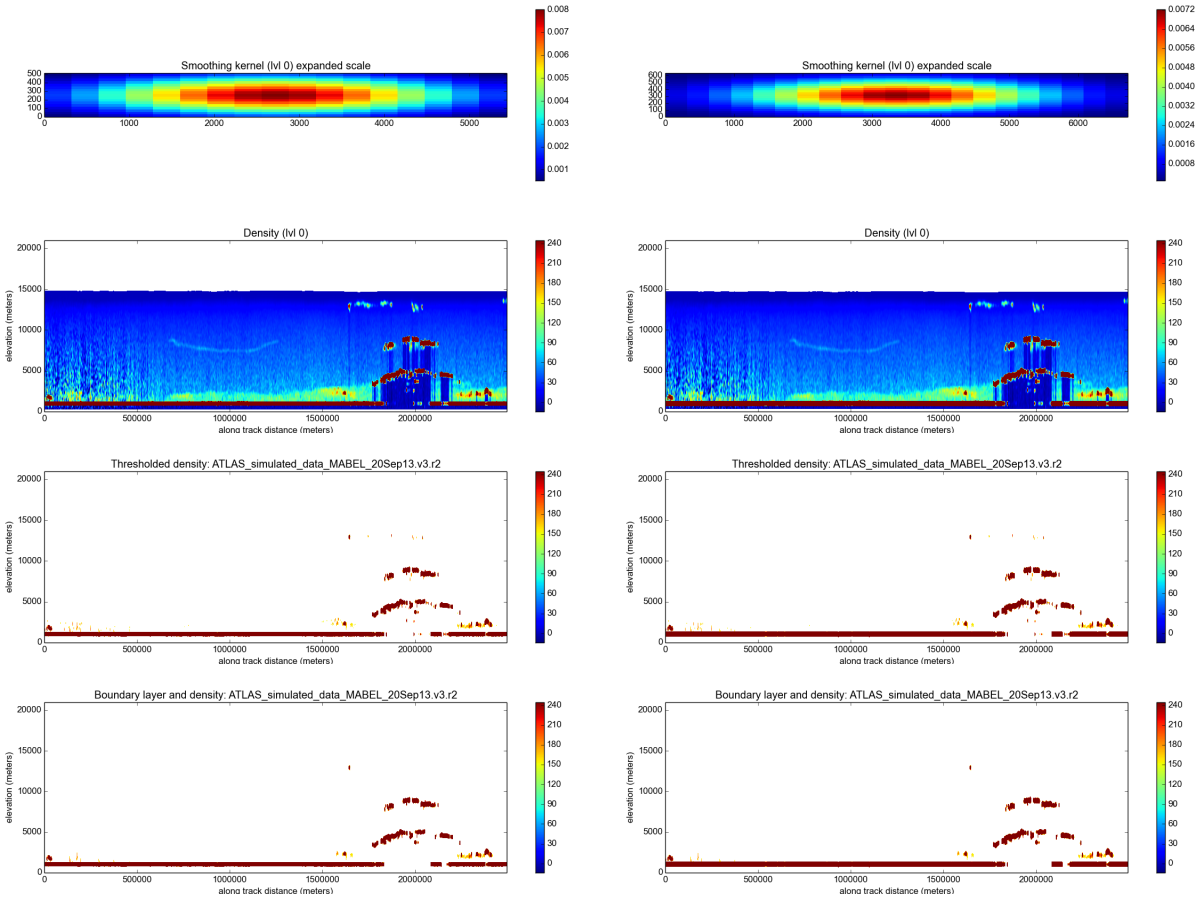
Analysis of day-time data (dataset: ATLAS_simulated_data_MABEL_20Sep13.v3.r2, based on MABEL atmosphere data collected 20Sep13).

$\sigma = 5$

$a_m = 10$

(e) neighborhood = 6 (13x13 kernel)

(f) neighborhood = 7 (15x15 kernel)



g) neighborhood = 8

h) neighborhood = 10

Figure 20, ctd. Sensitivity Experiment 1: Changing Neighborhood

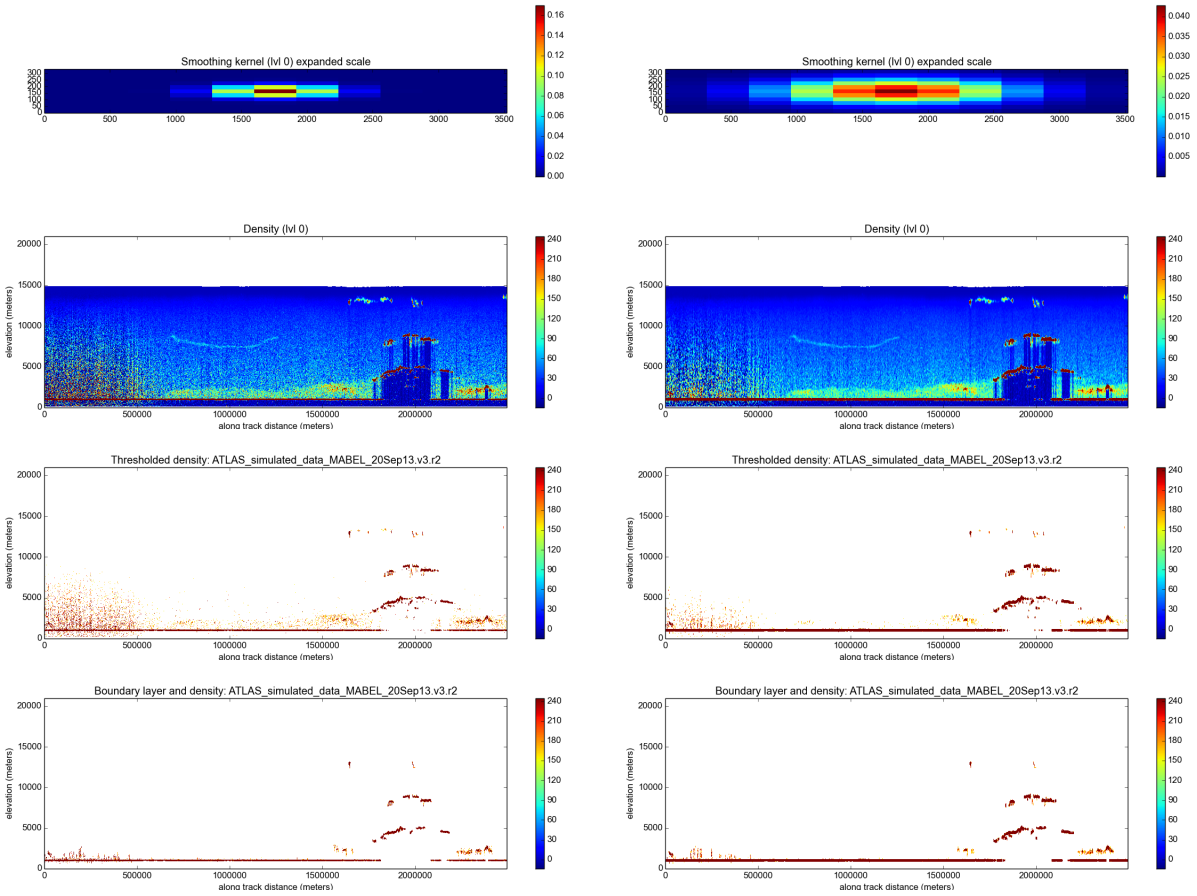
Analysis of day-time data (dataset: ATLAS_simulated_data_MABEL_20Sep13.v3.r2, based on MABEL atmosphere data collected 20Sep13).

$\sigma = 5$

$a_m = 10$

(g) neighborhood = 8 (17x17 kernel)

(h) neighborhood = 10 (21x21 kernel)



a) $sigma = 1$ b) $sigma = 2$

Figure 21. Sensitivity Experiment 2: Changing σ

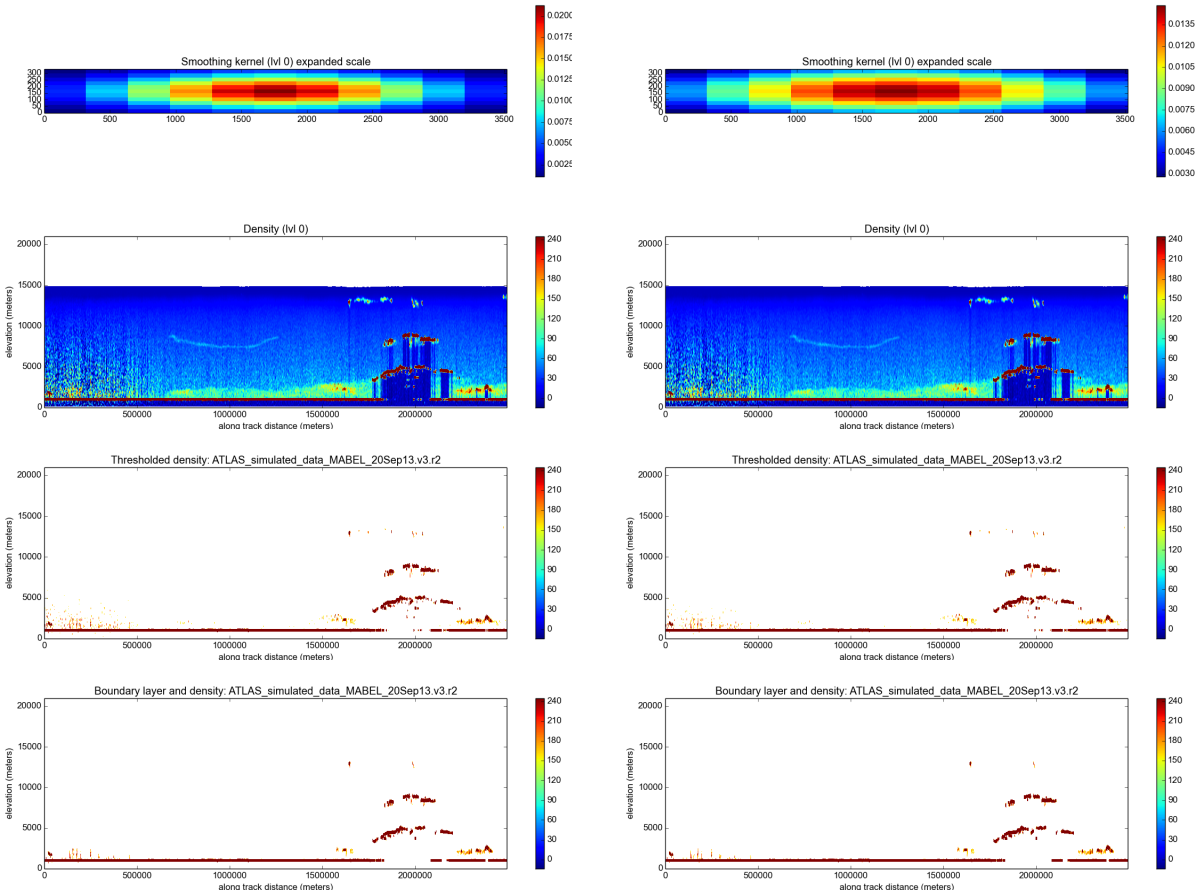
Analysis of day-time data (dataset: ATLAS_simulated_data_MABEL_20Sep13.v3.r2, based on MABEL atmosphere data collected 20Sep13).

neighborhood = 5

$a_m = 10$

(a) $\sigma = 1$

(b) $\sigma = 2$



c) $sigma = 3$

d) $sigma = 4$

Figure 21, ctd. Sensitivity Experiment 2: Changing σ

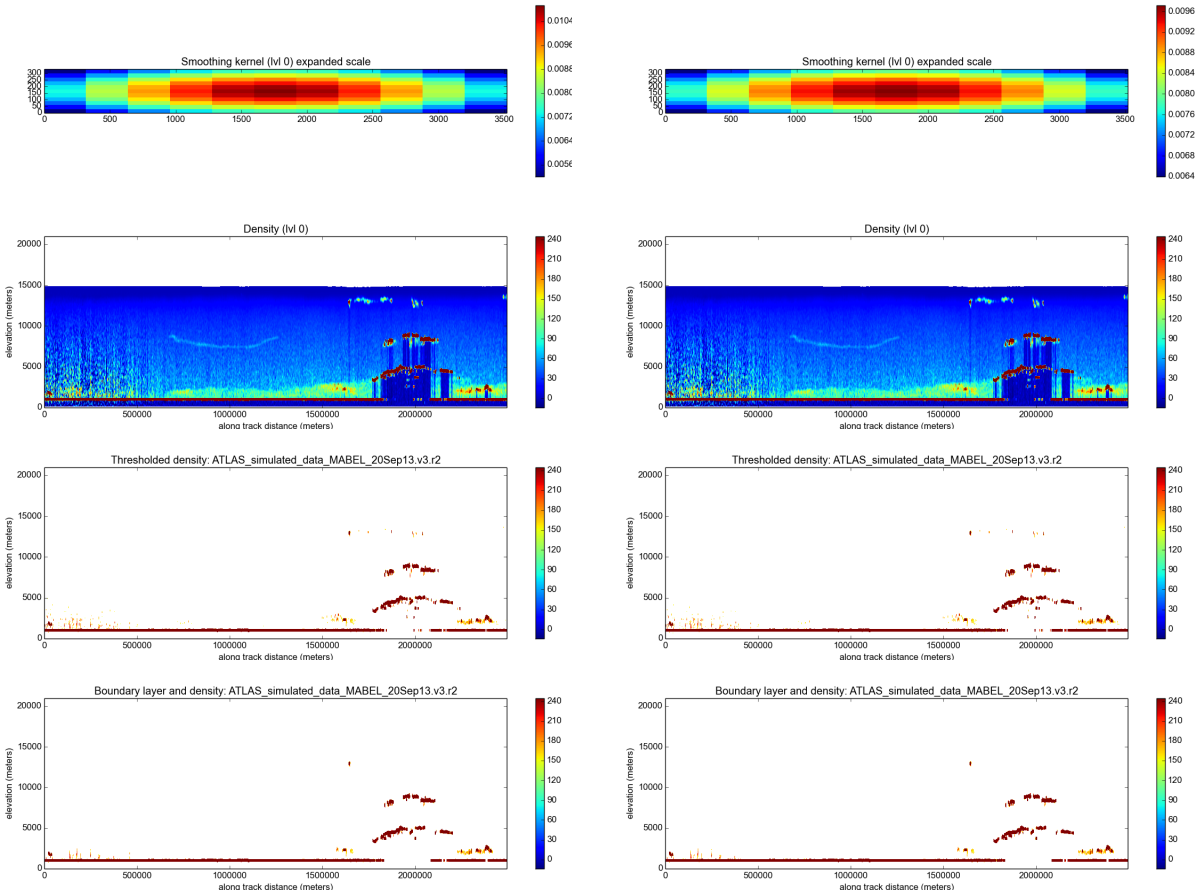
Analysis of day-time data (dataset: ATLAS_simulated_data_MABEL_20Sep13.v3.r2, based on MABEL atmosphere data collected 20Sep13).

neighborhood = 5

$a_m = 10$

(c) $\sigma = 3$

(d) $\sigma = 4$



e) $sigma = 6$ f) $sigma = 8$

Figure 21, ctd. Sensitivity Experiment 2: Changing σ

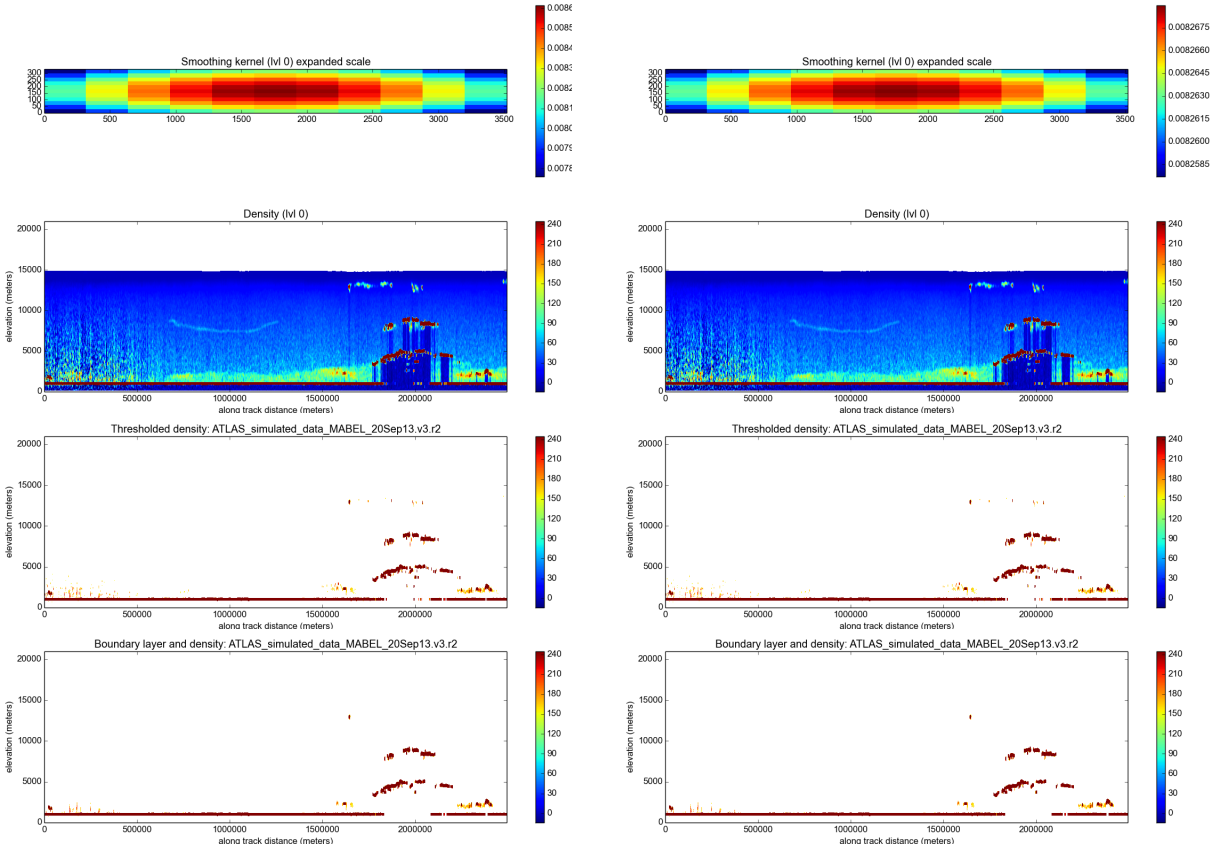
Analysis of day-time data (dataset: ATLAS_simulated_data_MABEL_20Sep13.v3.r2, based on MABEL atmosphere data collected 20Sep13).

neighborhood = 5

$a_m = 10$

(e) $\sigma = 6$

(f) $\sigma = 8$



g) $sigma = 16$ h) $sigma = 140$

Figure 21, ctd. Sensitivity Experiment 2: Changing σ

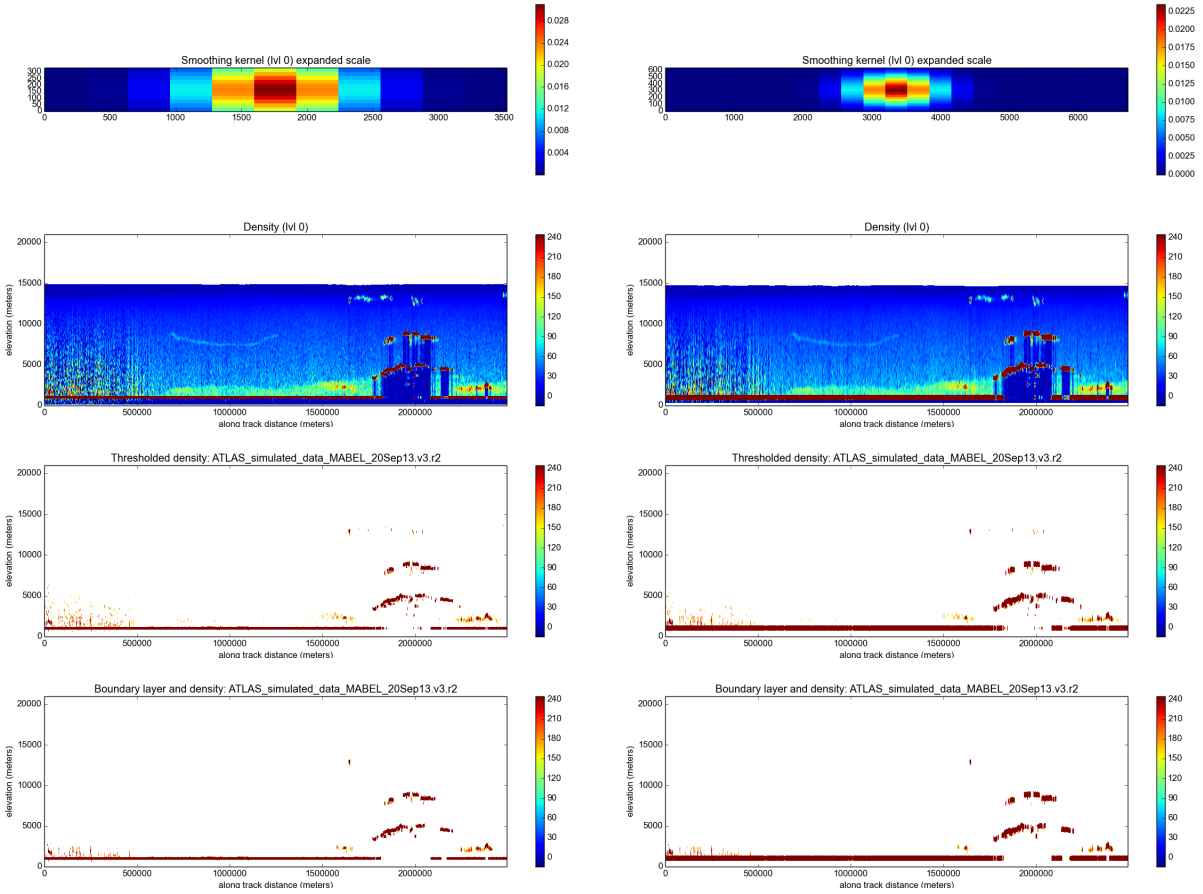
Analysis of day-time data (dataset: ATLAS_simulated_data_MABEL_20Sep13.v3.r2, based on MABEL atmosphere data collected 20Sep13).

neighborhood = 5

$a_m = 10$

(g) $\sigma = 16$

(h) $\sigma = 140$



a) $a_m = 3$, neighborhood = 5

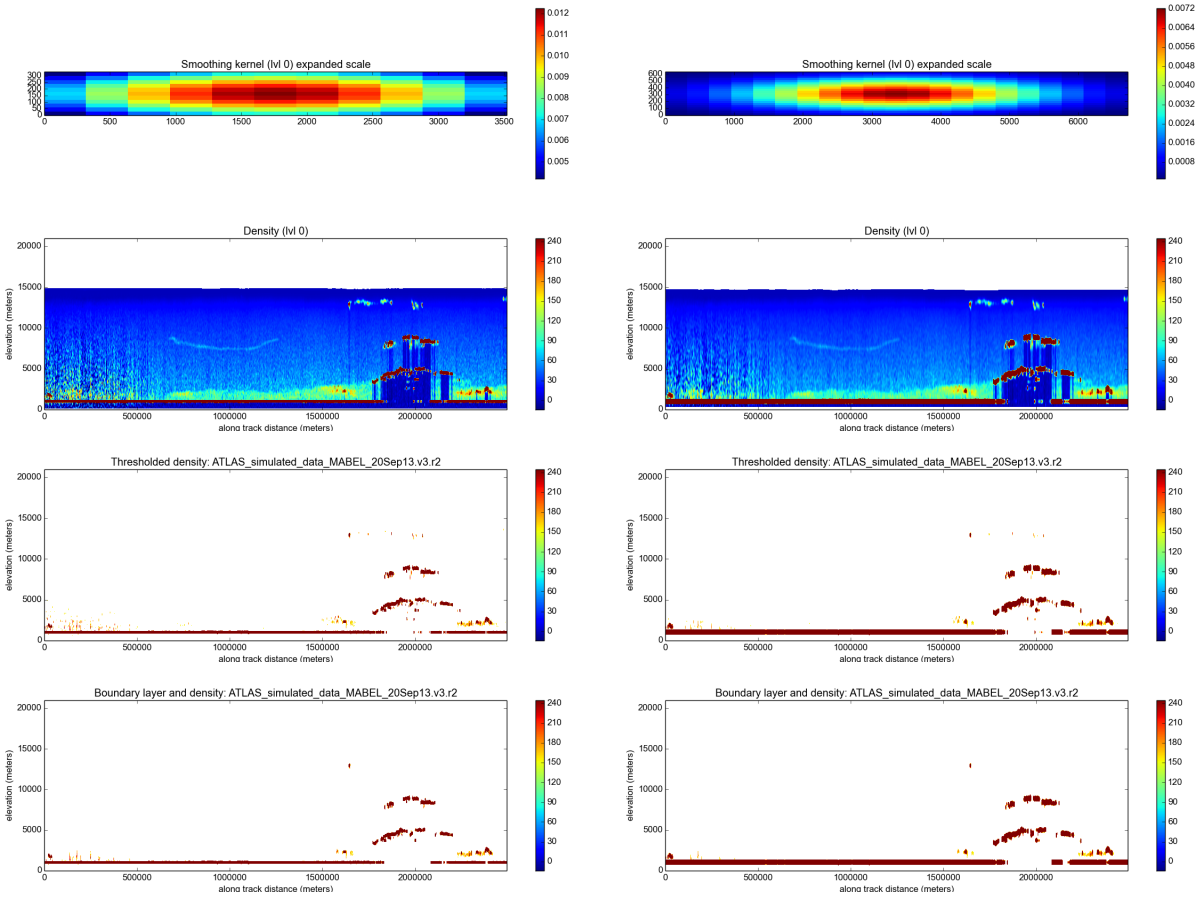
b) $a_m = 3$, neighborhood = 10

Figure 22. Sensitivity Experiment 3: Changing anisotropy (with neighborhood = 5 and 10) Analysis of day-time data (dataset: ATLAS_simulated_data_MABEL_20Sep13.v3.r2, based on MABEL atmosphere data collected 20Sep13).

$\sigma = 5$

(a) $a_m = 3$, neighborhood = 5

(b) $a_m = 3$, neighborhood = 10



c) $a_m = 10$, neighborhood = 5

d) $a_m = 10$, neighborhood = 10

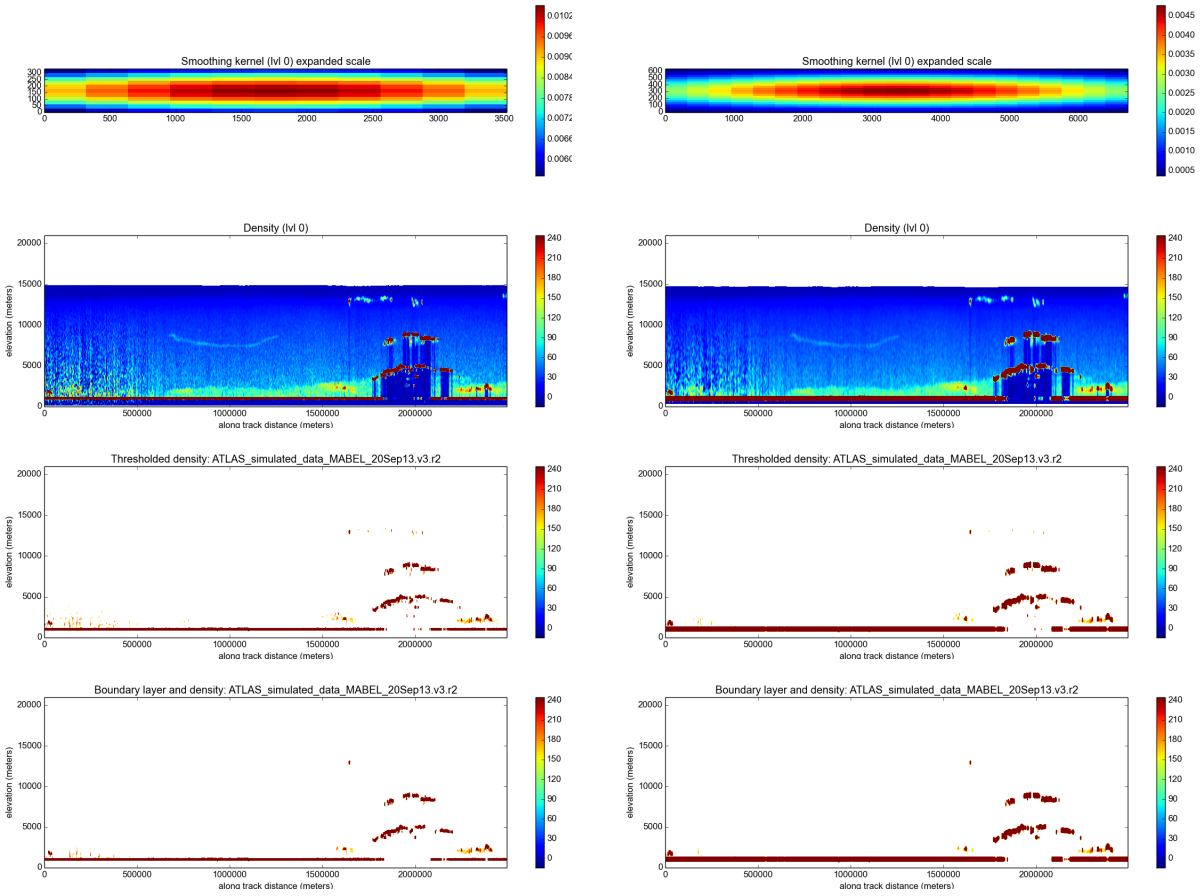
Figure 22, ctd. Sensitivity Experiment 3: Changing anisotropy (with neighborhood = 5 and 10)

Analysis of day-time data (dataset: ATLAS_simulated_data_MABEL_20Sep13.v3.r2, based on MABEL atmosphere data collected 20Sep13).

$\sigma = 5$

(c) $a_m = 10$, neighborhood = 5

(d) $a_m = 10$, neighborhood = 10



e) $a_m = 20$, neighborhood = 5

f) $a_m = 20$, neighborhood = 10

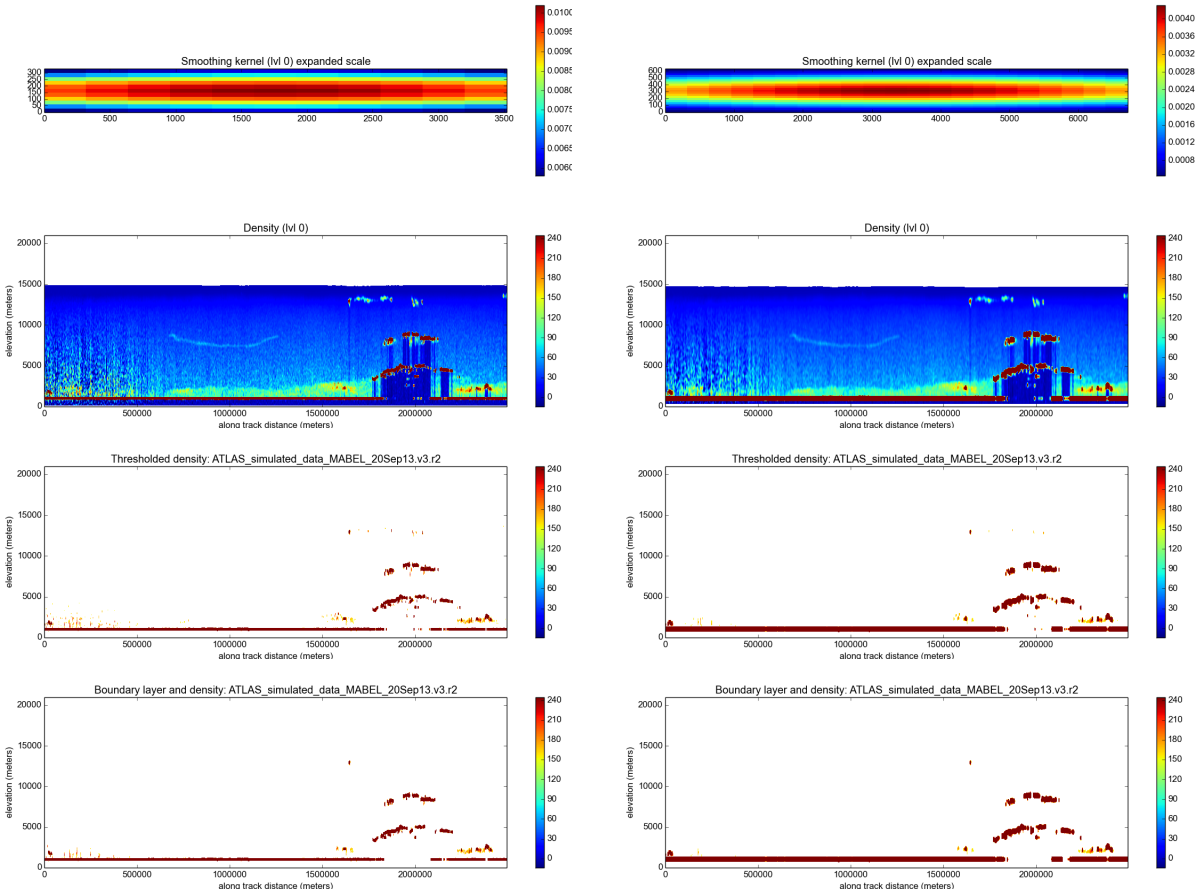
Figure 22, ctd. Sensitivity Experiment 3: Changing anisotropy (with neighborhood = 5 and 10)

Analysis of day-time data (dataset: ATLAS_simulated_data_MABEL_20Sep13.v3.r2, based on MABEL atmosphere data collected 20Sep13).

$\sigma = 5$

(e) $a_m = 20$, neighborhood = 5

(f) $a_m = 20$, neighborhood = 10



g) $a_m = 30$, neighborhood $= 5$

h) $a_m = 30$, neighborhood $= 5$

Figure 22, ctd. Sensitivity Experiment 3: Changing anisotropy (with neighborhood = 5 and 10)

Analysis of day-time data (dataset: ATLAS_simulated_data_MABEL_20Sep13.v3.r2, based on MABEL atmosphere data collected 20Sep13).

$\sigma = 5$

(g) $a_m = 30$, neighborhood $= 5$

(h) $a_m = 30$, neighborhood $= 10$

7.2 Sensitivity Studies for Double-Density Runs

The complete figure series for the double-density run is given in Figure 23;

parameters:

threshold_segment_length(=window) = 20, σ = 3,6, a_m = 10, 20, base_threshold = 70, 0, threshold_sensitivity = 1, 1.

To illustrate the dependencies in the framework of "running density twice", the following experiments are undertaken; parameters listed are parameters changed for density run 2:

Experiment 1: Changing threshold window (Figure 24), (a) window=10, (b) window= 30

Experiment 2: Changing threshold sigma (Figure 25), (a) σ = 5, (b) σ = 7

Experiment 3: Changing anisotropy (Figure 26), (a) a_m = 10, (b) a_m = 30

Experiment 4: Changing base threshold (Figure 27),

˜ (a) base_threshold = -3, (b) base_threshold = 3

Experiment 5: Changing threshold sensitivity (Figure 28),

˜ (a) threshold_sensitivity= 0.9, (b) threshold_sensitivity= 1.1

Experiment 6: Changing minimum cluster size (Figure 29),

˜ (a) size_threshold= 100, (b) size_threshold= 500

For each experiment, the following figures are shown (in 6 panels, top to bottom:)

(1) Kernel of density run 2,

(2) density 2,

(3) threshold of density run 2,

(4) preliminary cloud mask, after threshold application to density 2,

(5) cloud mask after small-cluster removal for density run 2,

(6) combined, final cloud mask from density runs 1 and 2, with density-1 values shown.

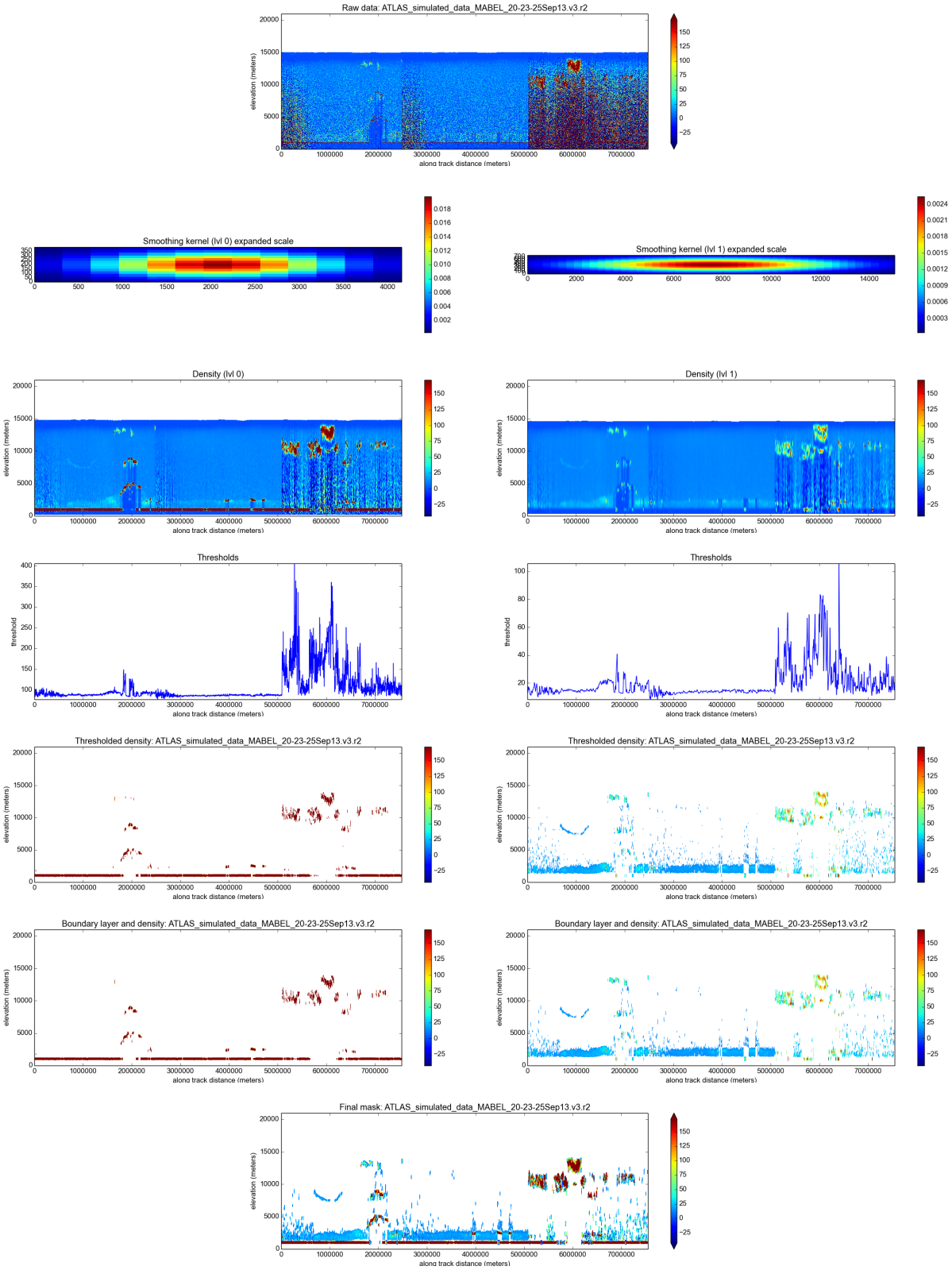


Figure 23. Complete Plot Series for Double Density Run

Analysis of day-time data (dataset: ATLAS_simulated_data_MABEL_20-23-25Sep13.v3.r2, based on MABEL atmosphere data collected 20-23-25Sep13).

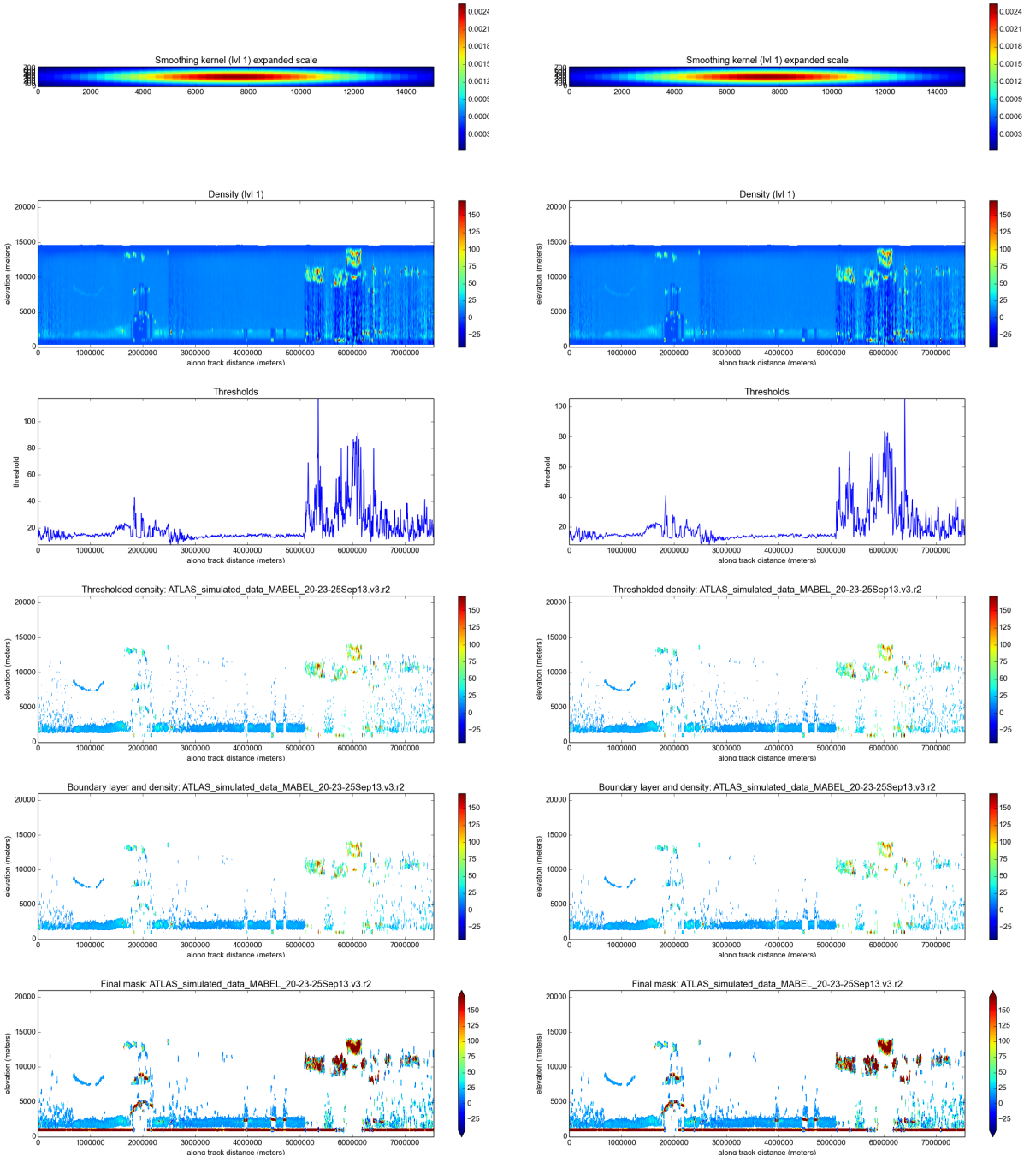
window = 20

σ = 3,6

a_m = 10, 20

base_threshold = 70, 0

threshold_sensitivity = 1, 1



a) window = 10

b) window = 30

Figure 24. Double Density Sensitivity Experiment 1: Changing threshold window

Analysis of day-time data (dataset: ATLAS_simulated_data_MABEL_20-23-25Sep13.v3.r2, based on MABEL atmosphere data collected 20-23-25Sep13).

σ = 3,6

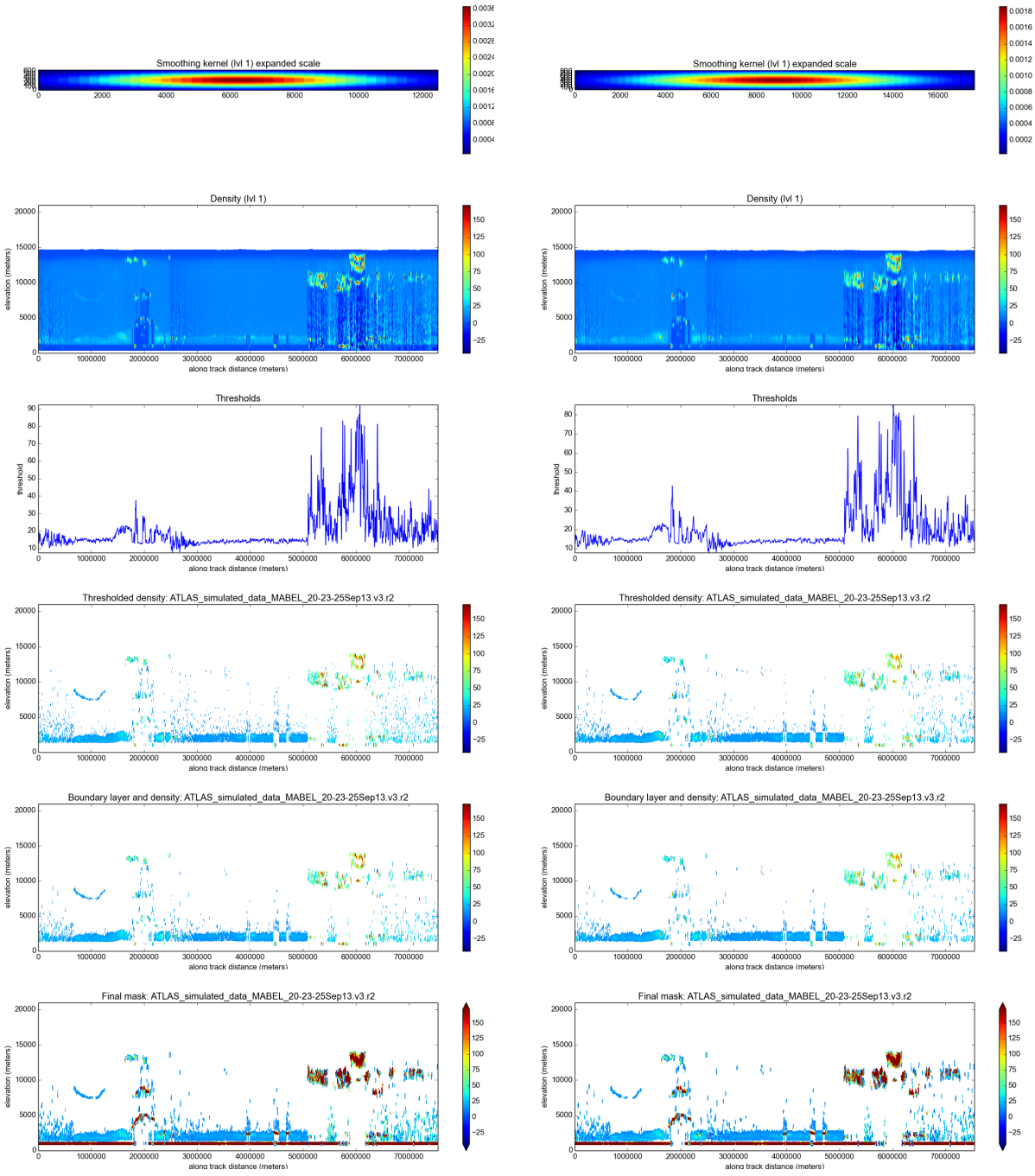
a_m = 10, 20

base_threshold = 70, 0

threshold_sensitivity = 1, 1

(a) window = 10

(b) window = 30



a) $\sigma = 5$ b) $\sigma = 7$

Figure 25. Double Density Sensitivity Experiment 2: Changing sigma

Analysis of day-time data (dataset: ATLAS_simulated_data_MABEL_20-23-25Sep13.v3.r2, based on MABEL atmosphere data collected 20-23-25Sep13).

window = 20

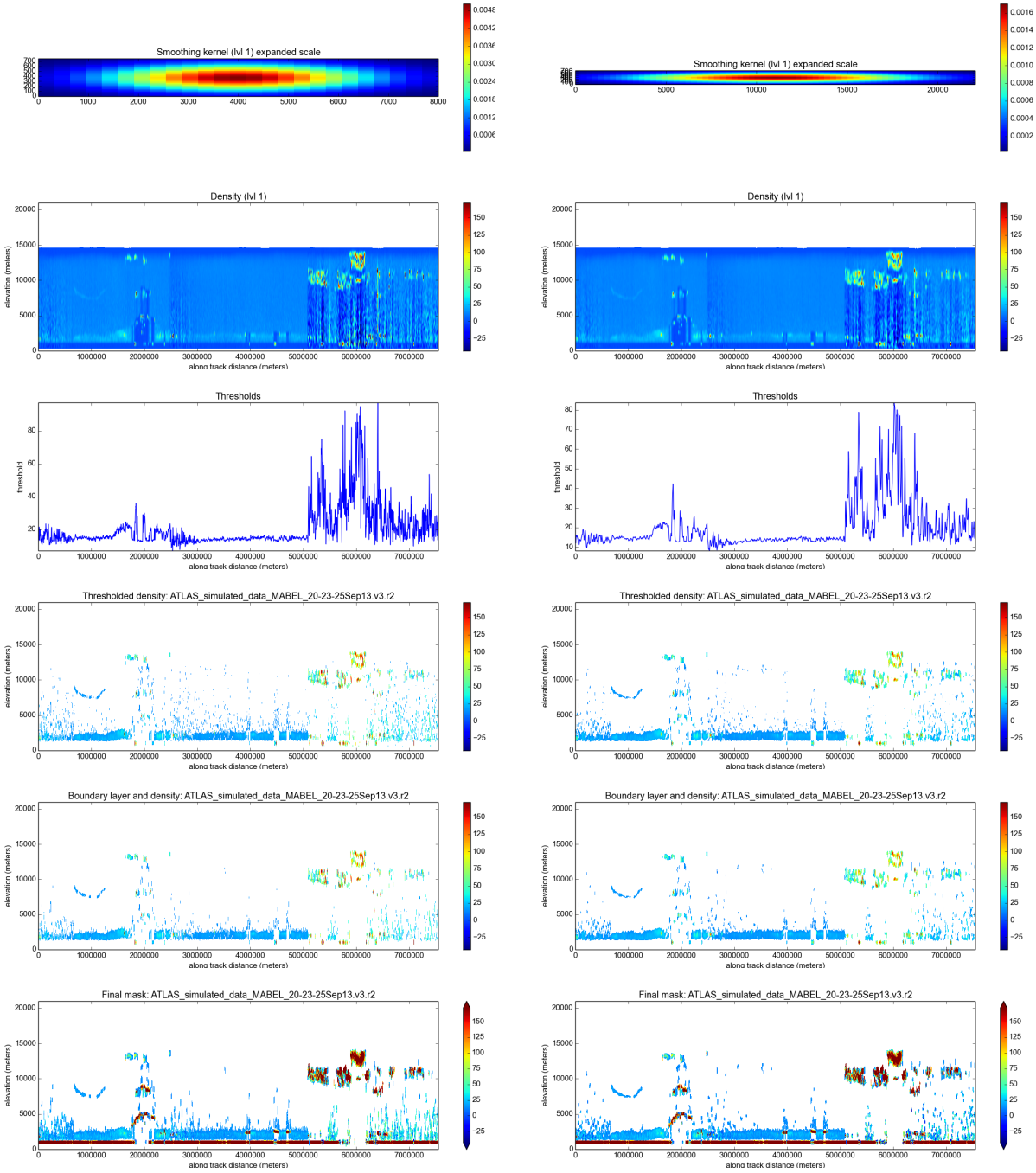
a_m = 10, 20

base_threshold = 70, 0

threshold_sensitivity = 1, 1

(a) $\sigma = 5$

(b) $\sigma = 7$



a) anisotropy = 10

b) anisotropy = 30

Figure 26. Double Density Sensitivity Experiment 3: Changing anisotropy

Analysis of day-time data (dataset: ATLAS_simulated_data_MABEL_20-23-25Sep13.v3.r2, based on MABEL atmosphere data collected 20-23-25Sep13).

window = 20

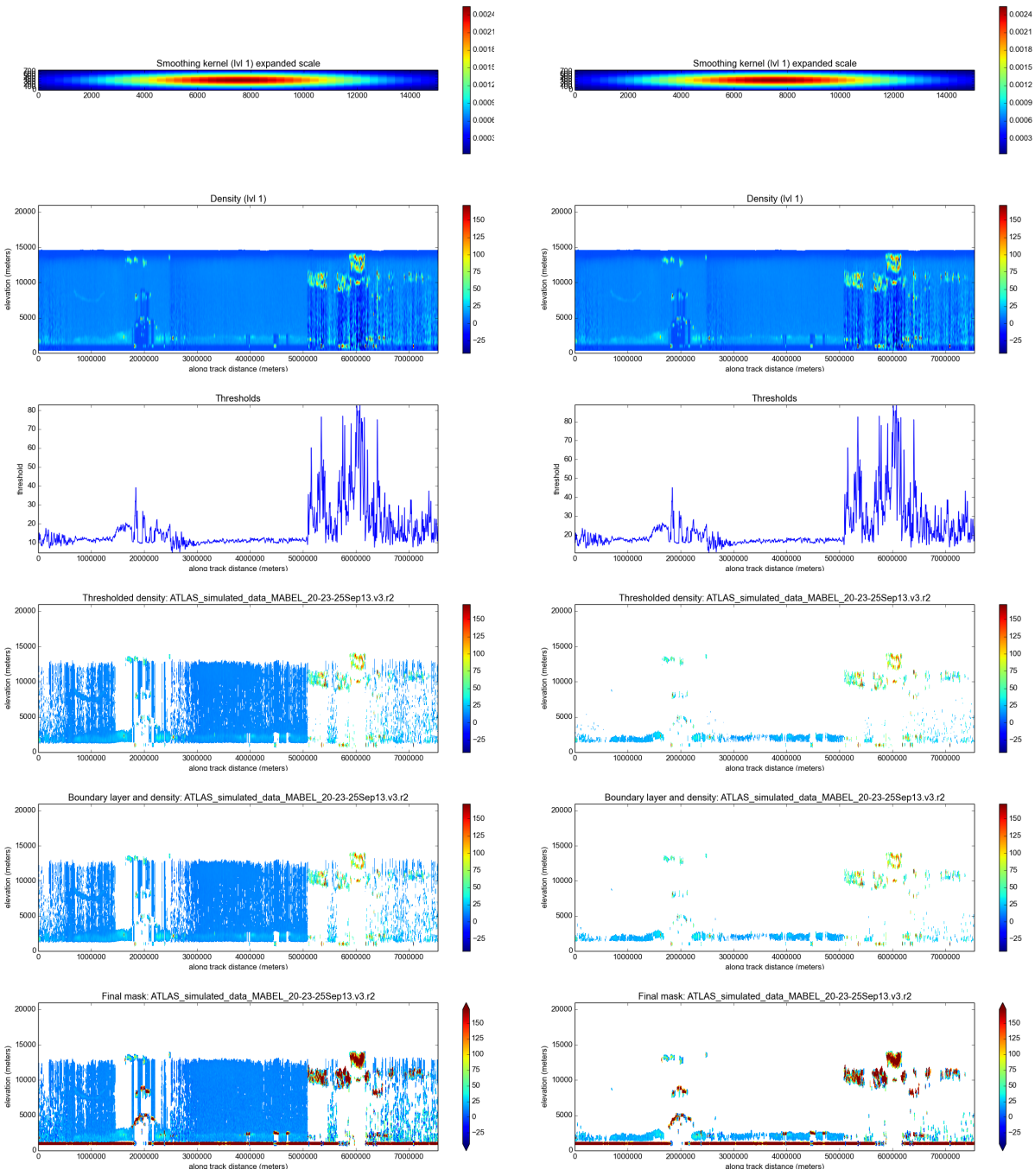
σ = 3, 6

base_threshold = 70, 0

threshold_sensitivity = 1, 1

(a) anisotropy= 10

(b) anisotropy = 30



a) base_threshold = -3

b) base_threshold = 3

Figure 27. Double Density Sensitivity Experiment 4: Changing Base Threshold

Analysis of day-time data (dataset: ATLAS_simulated_data_MABEL_20-23-25Sep13.v3.r2, based on MABEL atmosphere data collected 20-23-25Sep13).

window = 20

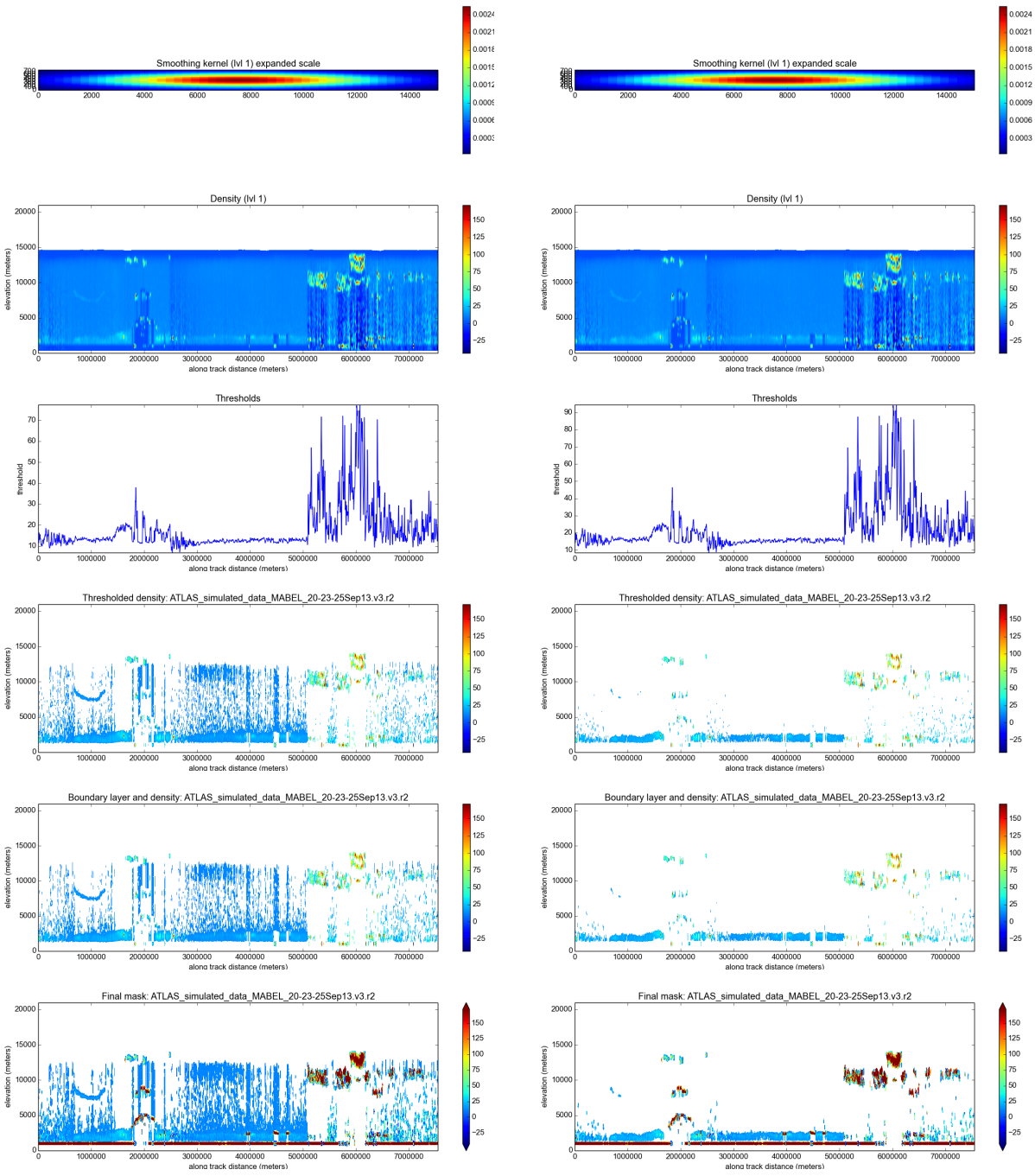
σ = 3, 6

a_m = 10,20

threshold_sensitivity = 1, 1

(a) base_threshold = -3

(b) base_threshold = 3



a) threshold_sensitivity = .9

b) threshold_sensitivity = 1.1

Figure 28. Double Density Sensitivity Experiment 5: Changing Threshold Sensitivity

Analysis of day-time data (dataset: ATLAS_simulated_data_MABEL_20-23-25Sep13.v3.r2, based on MABEL atmosphere data collected 20-23-25Sep13).

window = 20

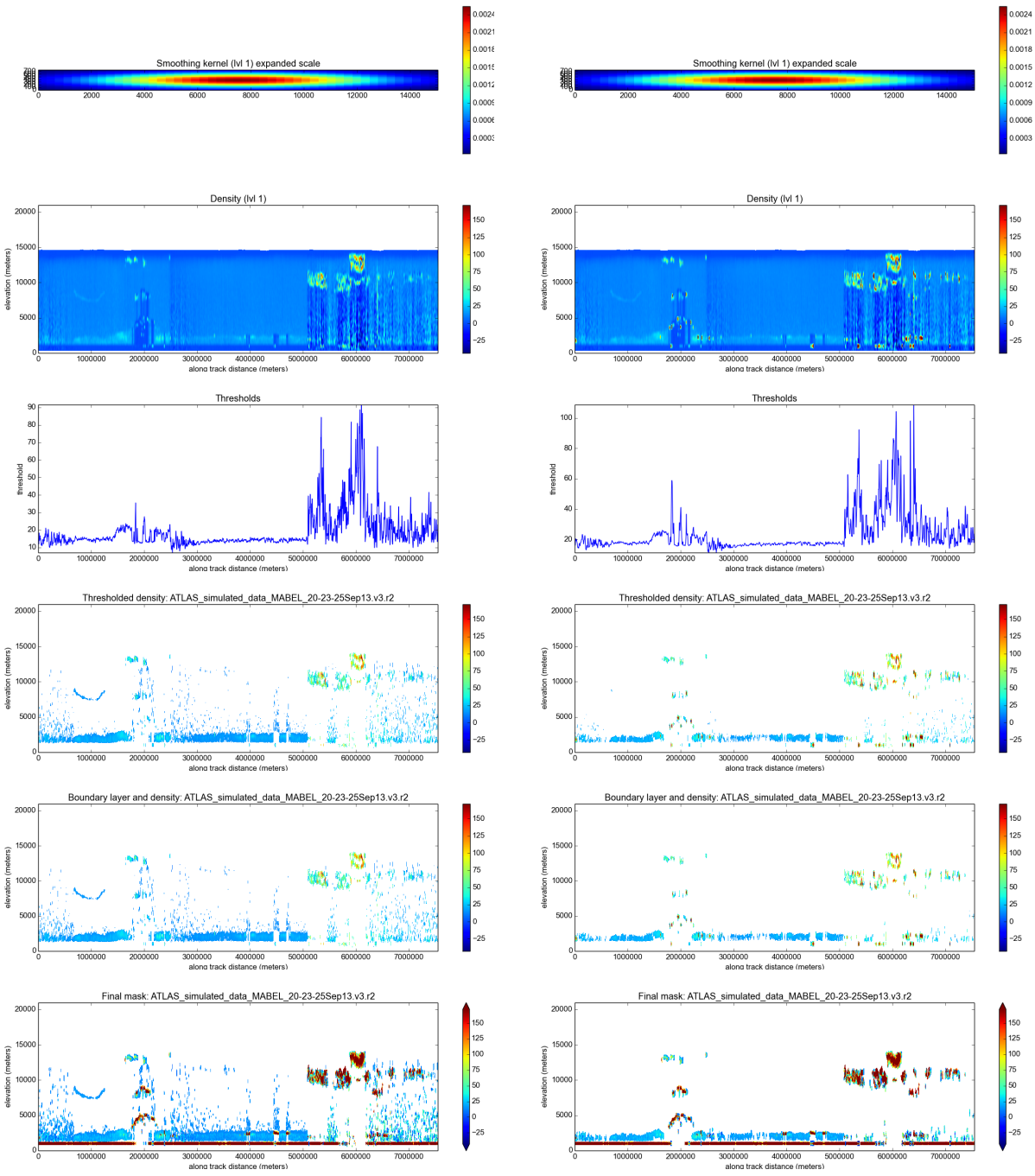
σ = 3, 6

a_m = 10,20

base_threshold = 70,0

(a) threshold_sensitivity = .9

(b) threshold_sensitivity = 1.1



a) size_threshold = 100

b) size_threshold= 500

Figure 29. Double Density Sensitivity Experiment 6: Changing Minimum Cluster Size

Analysis of day-time data (dataset: ATLAS_simulated_data_MABEL_20-23-25Sep13.v3.r2, based on MABEL atmosphere data collected 20-23-25Sep13).

window = 20

σ = 3, 6

a_m = 10,20

base_threshold = 70,0

threshold_sensitivity = 1,1

(a) size_threshold = 100

(b) size_threshold = 500

9 Sensitivity Studies for GLAS-Data-Based Simulated ICESat-2 Data (2016 Version)

Starting from the parameter combinations used in t8 and deemed best, a sensitivity study was carried out systematically, varying each parameter to above and below the parameter combination of t8 (which still remained best). Quality assessment (what is best?) is carried out by creation of a movie that shows layer boundaries from the DDA superimposed on the original GLAS data. The movie can be accessed under [url]. Here, an illustration of the effect of changing parameters in the sensitivity study is presented in Figure 34, which shows the analysis results for a segment with night-day transitions near near 6000 km.

Note auto-adaptive capability of the algorithm at the night-time day-time transition

All examples were run with code version v105. The sensitivity study includes single-density and double-density runs, and parameter combinations that correspond to method A and method B.

(a)

(b)

(c)

(d)

(e)

(f)

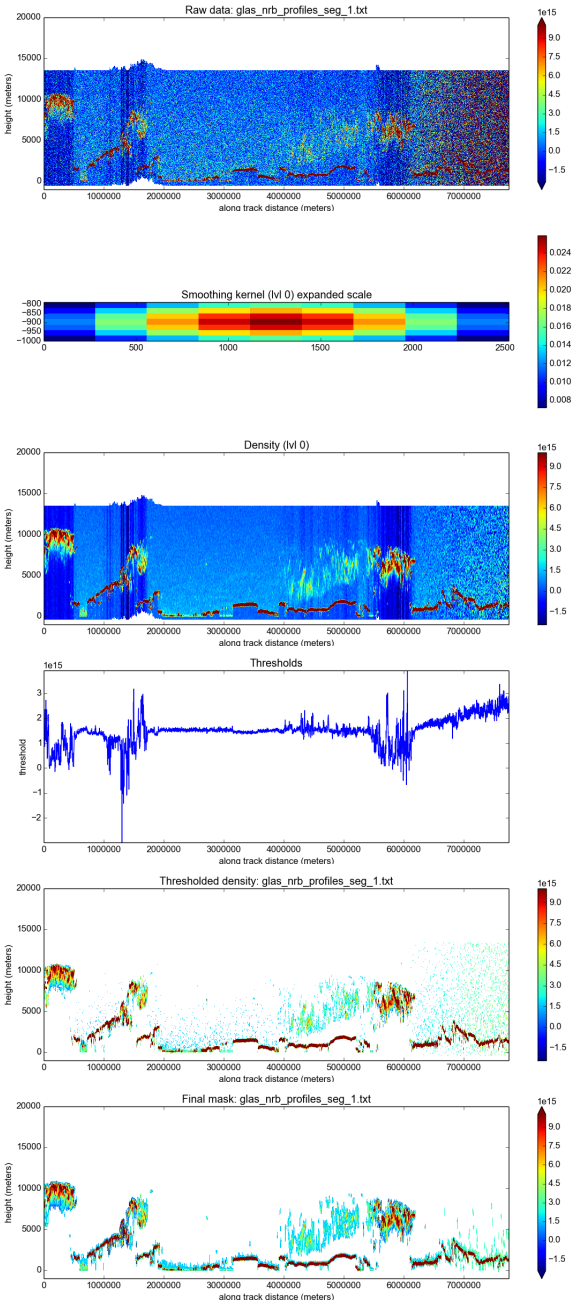


Figure 34-1. Sensitivity study to illustrate effects of parameter changes in method A/B, code version v105, applied to GLAS-based simulated ICESat-2 data. Segment 1 of the 2-orbit data set. min_cluster_size = 300 (t21), other parameters as in (t8).

$\sigma = 3$	cutoff = 1
$a_m = 10$	min cluster size = 300
base_threshold = 60E+13	downsampling = 1
threshold_sensitivity = 0.9	threshold_segment_length = 2
quantile = 0.75	method-B type

(a)

(b)

(c)

(d)

(e)

(f)

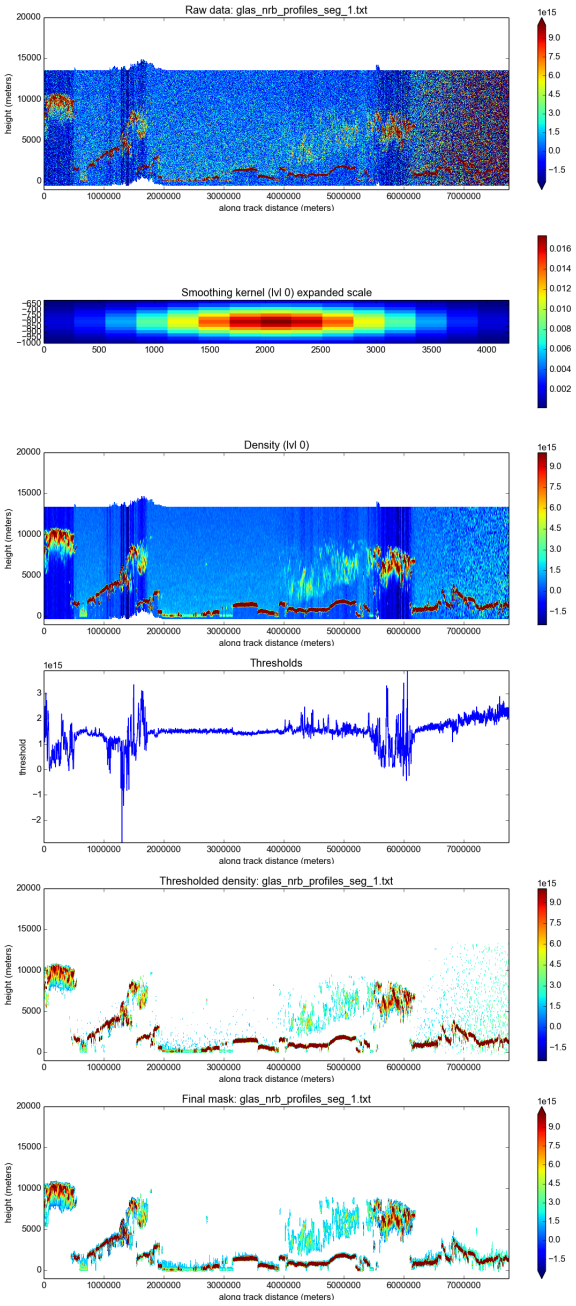


Figure 34-2. Sensitivity study to illustrate effects of parameter changes in method A/B, code version v105, applied to GLAS-based simulated ICESat-2 data. Segment 1 of the 2-orbit data set. cutoff = 2 (t22), other parameters as in (t8).

$\sigma = 3$	cutoff = 2
$a_m = 10$	min cluster size = 600
base_threshold = 60E+13	downsampling = 1
threshold_sensitivity = 0.9	threshold_segment_length = 2
quantile = 0.75	method-B type

(a)

(b)

(c)

(d)

(e)

(f)

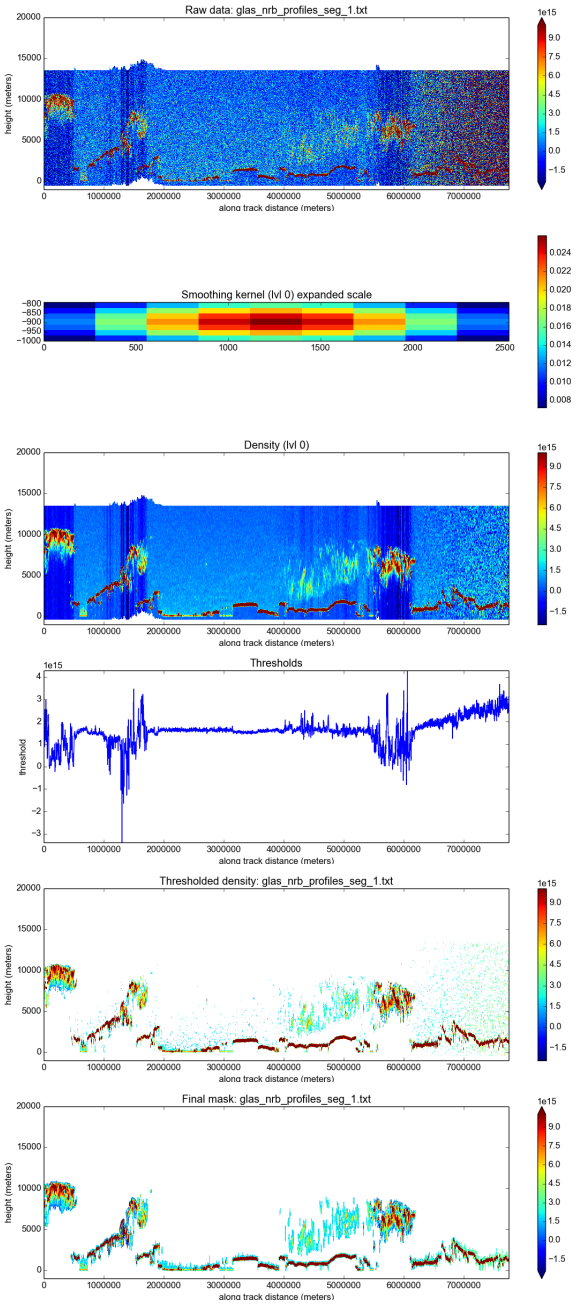


Figure 34-3. Sensitivity study to illustrate effects of parameter changes in method A/B, code version v105, applied to GLAS-based simulated ICESat-2 data. Segment 1 of the 2-orbit data set. threshold_sensitivity = 1 (t23) , other parameters as in (t8).

$\sigma = 3$	cutoff = 1
$a_m = 10$	min cluster size = 600
base_threshold = 60E+13	downsampling = 1
threshold_sensitivity = 1	threshold_segment_length = 2
quantile = 0.75	method-B type

(a)

(b)

(c)

(d)

(e)

(f)

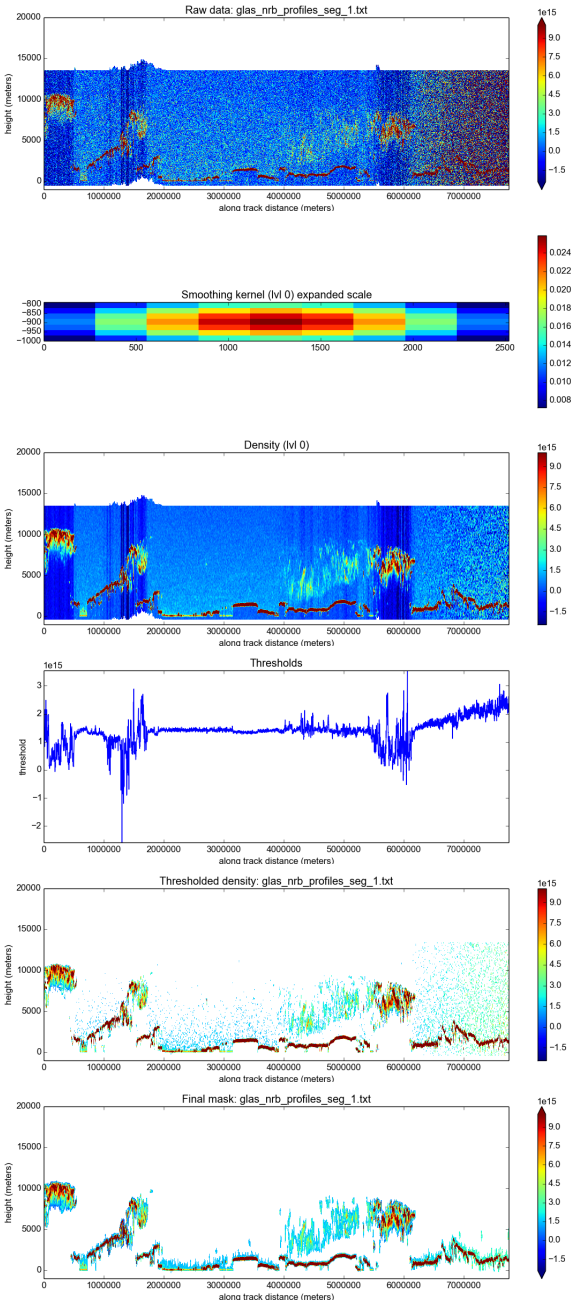


Figure 34-4. Sensitivity study to illustrate effects of parameter changes in method A/B, code version v105, applied to GLAS-based simulated ICESat-2 data. Segment 1 of the 2-orbit data set. threshold_sensitivity = 0.8 (t24), other parameters as in (t8).

$\sigma = 3$	cutoff = 1
$a_m = 10$	min cluster size = 600
base_threshold = 60E+13	downsampling = 1
threshold_sensitivity = 0.8	threshold_segment_length = 2
quantile = 0.75	method-B type

(a)

(b)

(c)

(d)

(e)

(f)

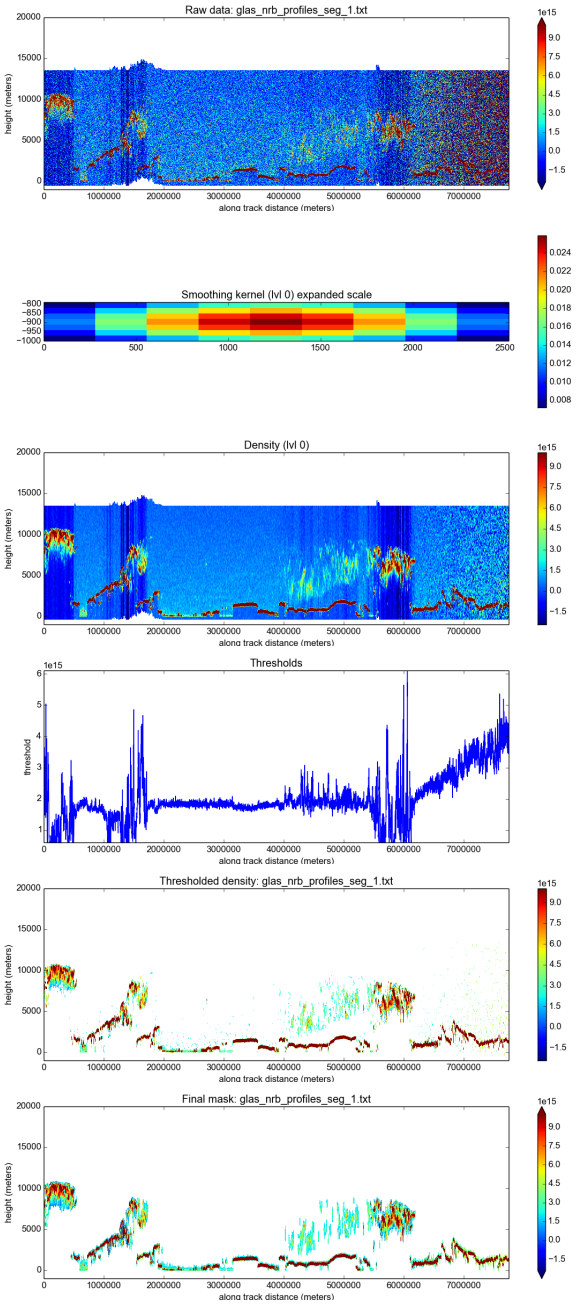


Figure 34-5. Sensitivity study to illustrate effects of parameter changes in method A/B, code version v105, applied to GLAS-based simulated ICESat-2 data. Segment 1 of the 2-orbit data set. downsampling = 5, threshold_segment_length = 0 and quantile = 0.5 (t25), other parameters as in (t8).

$\sigma = 3$	cutoff = 1
$a_m = 10$	min cluster size = 600
base_threshold = 60E+13	downsampling = 5
threshold_sensitivity = 0.9	threshold_segment_length = 0
quantile = 0.5	method-B type

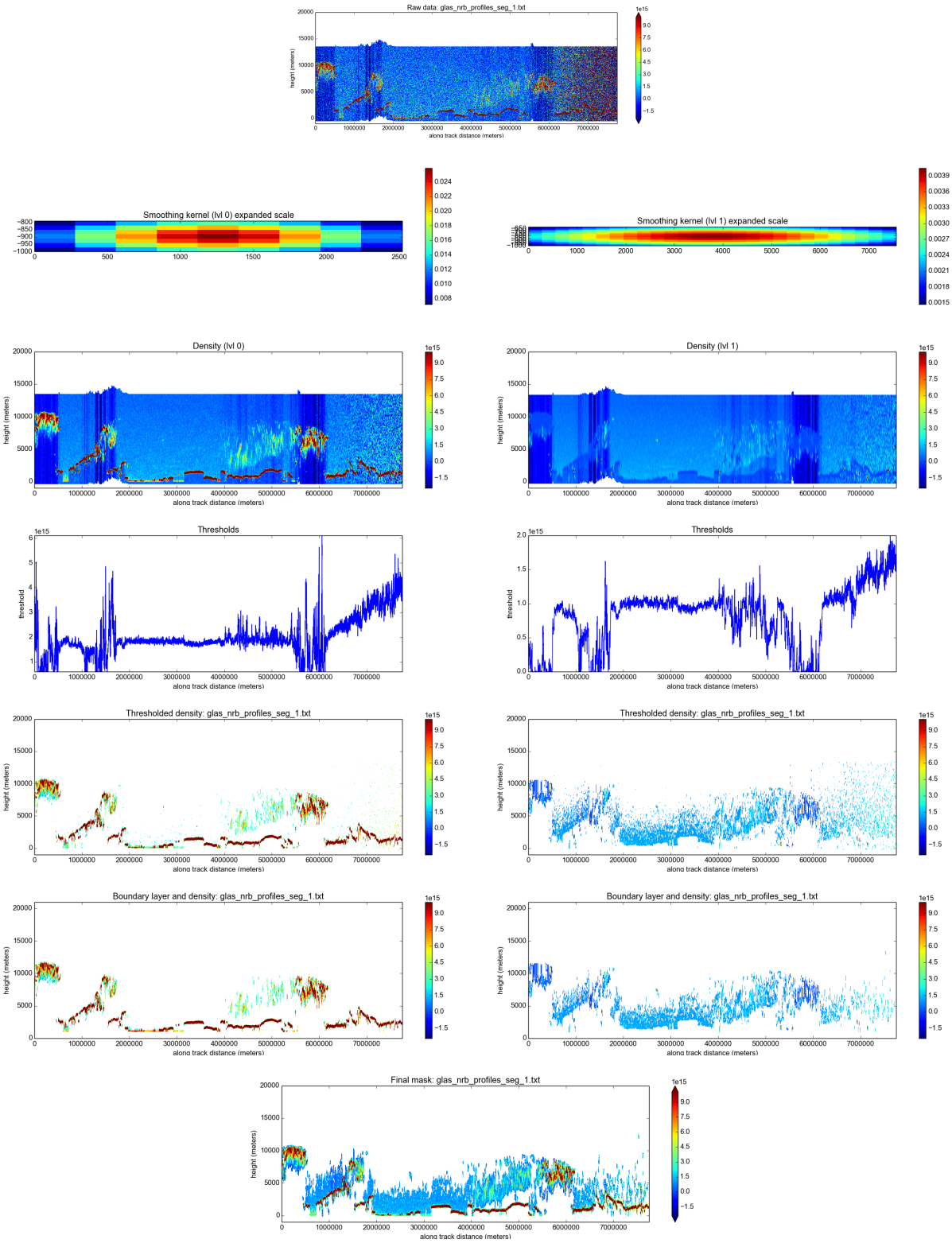


Figure 34-6. Sensitivity study to illustrate effects of parameter changes in method A/B, code version v105, applied to GLAS-based simulated ICESat-2 data. Segment 1 of the 2-orbit data set. Double density, downsampling = 5, threshold_segment_length = 0 and quantile = 0.5 (t26), other parameters as in (t8).

σ = 3,6	cutoff = 1,1
a_m = 10,20	min cluster size = 600,600
base_threshold = 60E+13,0	downsampling = 5,5
threshold_sensitivity = 1,1	threshold_segment_length = 0,0
quantile = 0.5,0.5	method-A type

(a)

(b)

(c)

(d)

(e)

(f)

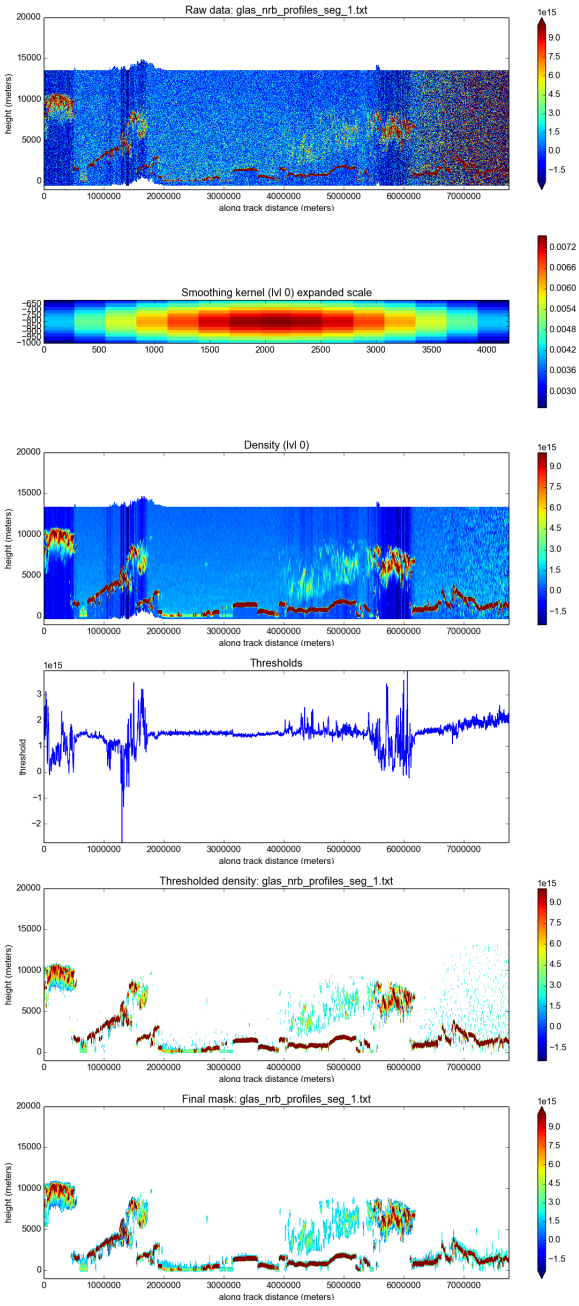


Figure 34-7. Sensitivity study to illustrate effects of parameter changes in method A/B, code version v105, applied to GLAS-based simulated ICESat-2 data. Segment 1 of the 2-orbit data set. sigma = 6 (t28), other parameters as in (t8).

$\sigma = 6$	cutoff = 1
$a_m = 10$	min cluster size = 600
base_threshold = 60E+13	downsampling = 1
threshold_sensitivity = 0.9	threshold_segment_length = 2
quantile = 0.75	method-B type

(a)

(b)

(c)

(d)

(e)

(f)

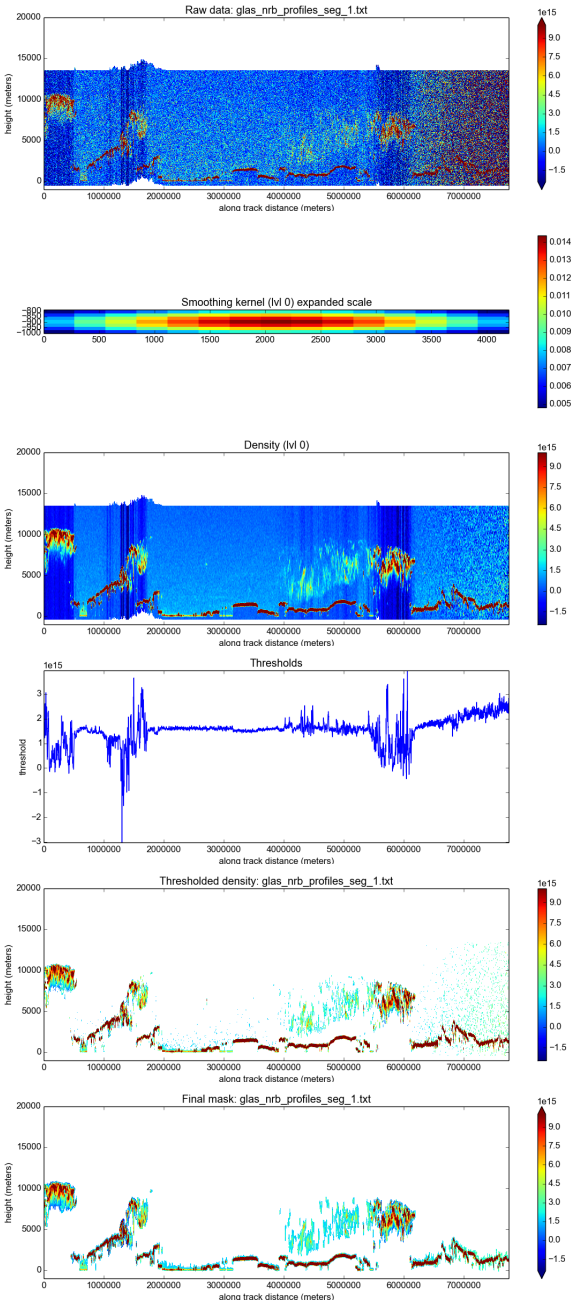


Figure 34-8. Sensitivity study to illustrate effects of parameter changes in method A/B, code version v105, applied to GLAS-based simulated ICESat-2 data. Segment 1 of the 2-orbit data set. a_m = 20 (t12), other parameters as in (t8).

$\sigma = 3$	cutoff = 1
$a_m = 20$	min cluster size = 600
base_threshold = 60E+13	downsampling = 1
threshold_sensitivity = 0.9	threshold_segment_length = 2
quantile = 0.75	method-B type

(a)

(b)

(c)

(d)

(e)

(f)

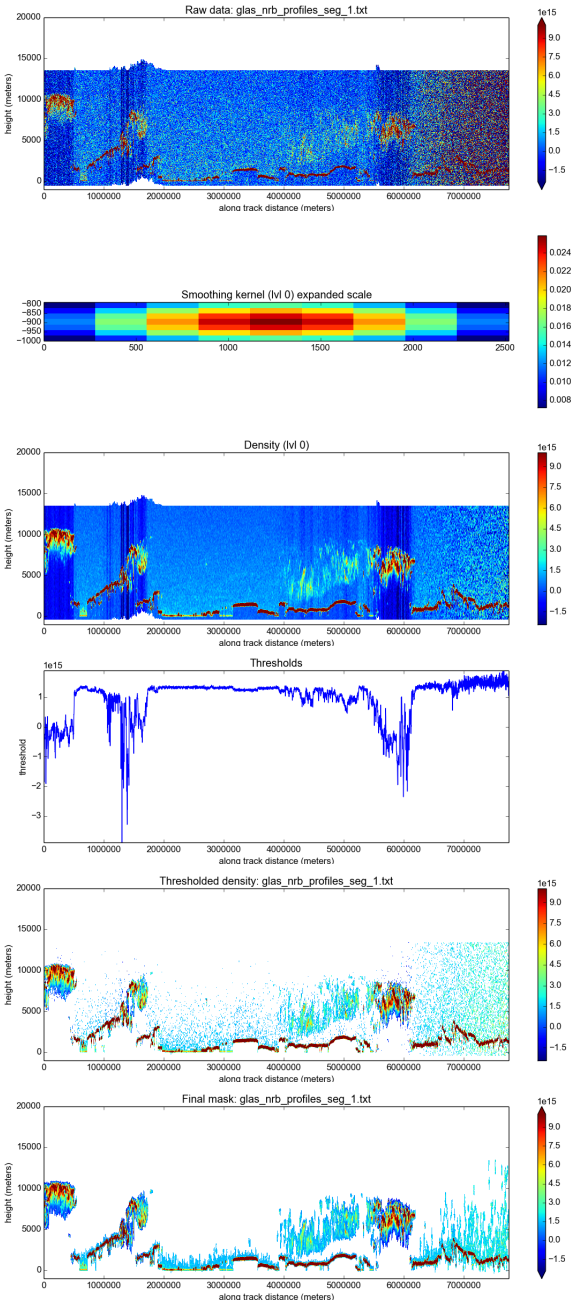


Figure 34-9. Sensitivity study to illustrate effects of parameter changes in method A/B, code version v105, applied to GLAS-based simulated ICESat-2 data. Segment 1 of the 2-orbit data set. q = 0.5 (t6), other parameters as in (t8).

$\sigma = 3$	cutoff = 1
$a_m = 10$	min cluster size = 600
base_threshold = 60E+13	downsampling = 1
threshold_sensitivity = 0.9	threshold_segment_length = 2
quantile = 0.5	method-B type

(a)

(b)

(c)

(d)

(e)

(f)

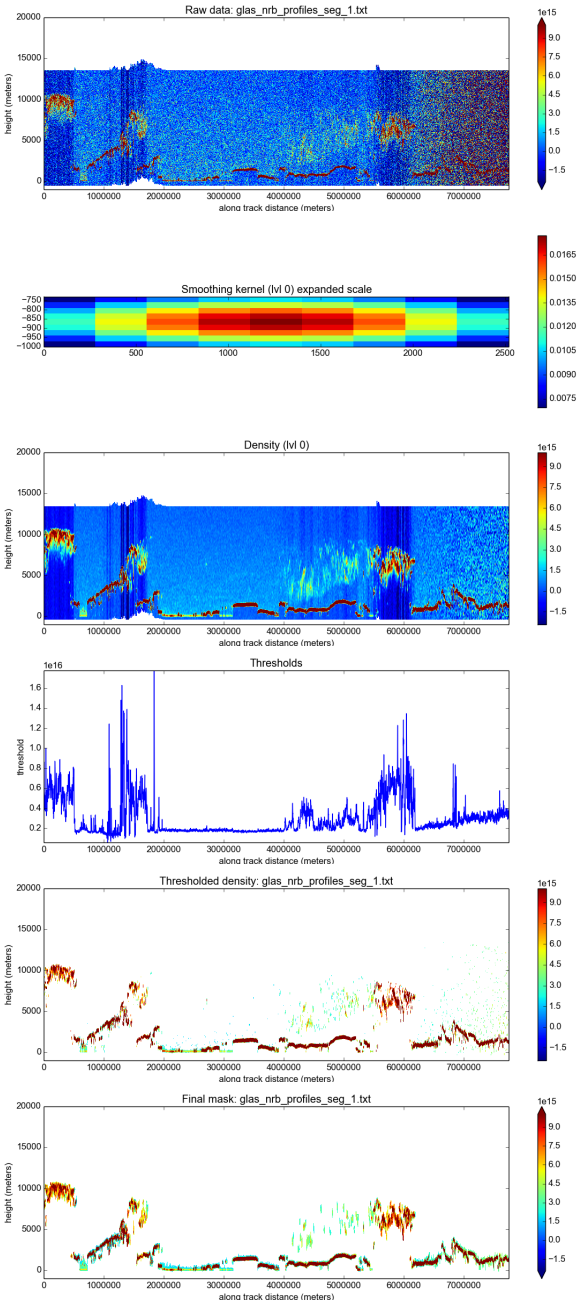


Figure 34-10. Sensitivity study to illustrate effects of parameter changes in method A/B, code version v105, applied to GLAS-based simulated ICESat-2 data. Segment 1 of the 2-orbit data set. q = 0.9 (t31), other parameters as in (t8).

$\sigma = 3$	cutoff = 1
$a_m = 10$	min cluster size = 600
base_threshold = 60E+13	downsampling = 1
threshold_sensitivity = 0.9	threshold_segment_length = 2
quantile = 0.9	method-B type

10 Sensitivity Studies for Analysis of 2017-Oct Version of GLAS-based Simulated ATL04 Data

10.1 Summary, Motivation and Data Sets

Summary. In this section, differences in the characteristics of "GLAS-data-based simulated ICESat-2 data (ATL04) of Oct-2017" data compared to "GLAS-data-based simulated ICESat-2 data (2016 version)" are described. A new sensitivity study is carried out to determine a set of algorithm-specific parameters for auto-adaptive analysis of ATL04 data (with Oct 2017 characteristics.). An important result is that the DDA-algorithm option "running density twice" is required to ascertain correct detection of different types of atmospheric layers during day-time and night-time conditions. As the application of the newly-developed Q/A measure "half-gap confidence flag" (see section (11) Quality Assessment) shows, the layer detection using the double-density runs with the parameter sets (t56) [and (t64)] yields throughout high confidences (mostly 0.8) and somewhat lower confidences where appropriate. – Why two parameter sets at this point? See Section (12) on Testing.

Necessity. Algorithm refinement for the upstream data products, ATL02 and ATL04, and code development and implementation for those products resulted in different characteristics of the NRB values in ATL04, compared to those of the 2016 state-of-the-art simulated GLAS-based ICESat-2 type data. Especially, the energy value in the NRB calculation changed, resulting in NRB values that are almost an order of magnitude larger. The NRB calculation now also includes an ad-hoc identification of optically thick clouds, masking of those clouds and subtraction of everything outside of this mask as "background" (see Part1 of this ATBD). Since this pseudo-background de facto includes thin clouds and aerosols, NRB values can be negative. Because of the division by range-squared, the values can be negative on the order of -1E27.

If we disregard the change of data characteristics and apply the DDA in a single-density run using parameter combination (t8), which was determined best for the 2016 GLAS-data-based simulated ICESat-2 data, we find (see Figure 34-1) that

(a) tenuous cloud layers and aerosol layers are still detected [good]

(b) false positives occur (areas identified as clouds that are likely not clouds) [bad].

These facts require a fresh determination of the set of algorithm-specific parameters. As before, parameters are changed and a new sensitivity study is carried out. Notably, changes in energy levels may also happen after launch, hence it is important to understand the algorithm sensitivity to changes in data characteristics.

In addition, the task of developing confidence measure for atmospheric layer detection as part of quality assessment (Q/A) requires a well-functioning layer detection algorithm, which includes a set of algorithm specific parameters that is matched to the data characteristics.

Data set. In the sensitivity studies, we use a relatively short synthetic data set of 7143 profiles, which includes several sections selected from a 70000 profile data set to include different types of clouds (morphologically complex tenuous cloud, optically thick cloud, aerosol layer) and data from night times and day times. The advantage of using a short data set in sensitivity studies is that visual inspection is possible. Different data situations need to be included to ascertain layer detection with auto-adaptive capabilities. For this study, ICESat-2 type data were simulated based on GLAS data, using the process described in section (9). Data were passed through the SIPS coded processing chain, and the resultant NRB data from product ATL04 are used here. These data are referred to as "GLAS-based simulated ATL04 data (2017-Oct version)". These data are also used in testing of the SIPS code implementation, in comparison of CU code and SIPS code, and in Q/A development. These data constitute the last pre-launch data utilized in algorithm development.

10.2 Single-Density Runs Versus Double-Density Runs

The analysis of the 2016 state-of-the-art GLAS-based simulated data performed well using single-density runs with parameter combination (t8). Some of the previously analyzed data sets required double-density runs (running density twice, see section (3.10)), for instance, early GLAS-based simulated data and 2013 M-ATLAS Data (section (8)). In this section, we perform sensitivity

studies for single-density runs and double-density runs, to analyze the trade-off between the two options for the ATL04 GLAS-based data sets. To recall, the trade-off is as follows:

(a) Single-density runs require less computer time.

(b) The algorithm is computationally inexpensive (mostly linear algebra), hence fast. Therefore double-density runs can be afforded computationally.

(c) A double-density run allows to detect clouds layers of very different spatial characteristics - optically thick, but possibly spatially thin (but not necessarily spatially thin) cloud layers in runs 1. In the second run, the thick cloud layers are ignored and a larger kernel is used to facilitate aggregation of photon counts (or NRB values) over a larger region, which brings out tenuous cloud and atmospheric layers.

10.3 Results and Consequences for Algorithm Applications: Running Density Twice, $t56$, $t64$

In summary, the results of the sensitivity studies are as follows (see also section 3.6 on layer boundaries):

(1) The parameter combination, $t8$, which worked best for the 2016 state-of-the-art simulated GLAS-based ICESat-2 type data, renders ill-defined layer-tops and bottoms especially at day time (right part of the data set). Some false positives appear, especially around the layer boundaries. This indicates that the parameters that determine the threshold function do not match the characteristics of the data any more. The change in NRB value determination (ATL04) requires a new set of parameters.

(2) The parameter combination, $t54$, is the best result for a single-density run, as determined in the sensitivity study. The layer boundaries are much better defined than in the $t8$ run. However, the layer boundaries are still somewhat ragged for night-time data and sub-optimally defined for day-time data. As the sensitivity study shows, it is not possible to retain tenuous clouds, while suppressing false positives, using a single-density run.

(3) This necessitates using a double-density run, which allows to identify optically thick layers in the first run (using a smaller kernel and a very strict threshold function) and, in the second

density run, identify the tenuous clouds, atmospheric layers and most clouds during day-time conditions (using a larger kernel and a less strict threshold function). Notice that tenuous clouds (on the left) are now connected in the vertical direction, except for likely natural gaps, rain (?) falls out of the layer at a possible inversion (aerosols with clouds at the inversion height), but no false positives remain and the layer tops during day-time are smooth. Single clouds are retained during day-time conditions. $t56$ is the parameter combination used in most experiments in October/November 2017 (and deemed best for current state-of-the-art data characteristics in ATL04).

(4) Varying parameters around those of $t56$ and trouble-shooting remaining differences between CU code and SIPS code, we noticed that a smaller cluster size in run1 (200 rather than 300 pixels) retains all good characteristics of $t56$ and appears to slightly improve them. The cluster size of 200 also renders the algorithm more robust (in the sense that all small speckles are already filtered out and first-order cloud layers are more continuous). This is $t64$. Parameters are otherwise the same as in $t56$. Note this may be good to know in testing, going forward, as the CU declustering step and the SIPS declustering step employ similar, but not the exact same function.

In consequence, double-density runs will be used going forward for analysis of ICESat-2 simulated data and ATLAS atmosphere data after launch.

The ATL09 data product will include the density fields from run 1 and from run 2 (see Table 2d).

Details regarding interpretation of the individual sensitivity experiments are given after the figures.

10.4 Sensitivity Studies for Single-Density Runs

The following table shows the parameter combinations that were used for single-density runs:

Parameter Name	Sigma	Anisotropy	Cutoff	Down-sample	Minimum Cluster Size	Threshold Bias	Threshold Factor	Threshold Window	Quantile
t8*	3	10	1	1	600	6E+14	0.9	2	0.75
t32*	3	10	1	1	600	12E+14	0.9	2	0.75
t33*	3	10	1	1	600	10E+14	0.9	2	0.75
t34*	3	10	1	1	600	10E+14	0.9	3	0.75
t35	3	10	1	1	600	6E+14	0.9	3	0.75
t36*	3	10	1	1	600	10E+14	0.9	4	0.75
t37	3	10	1	1	600	6E+14	0.9	4	0.75
t38*	3	10	1	1	600	10E+14	1	2	0.75
t39	3	10	1	1	600	6E+14	1	2	0.75
t40	3	10	1	1	600	6E+14	1	2	0.9
t41	3	10	1	1	600	6E+14	1	2	0.85
t42	3	10	1	1	600	6E+14	0.9	2	0.9
t43	3	10	1	1	600	6E+14	0.9	2	0.85
t44*	3	10	1	1	600	10E+14	1	2	0.9
t45*	3	10	1	1	600	10E+14	1	2	0.85
t46*	3	10	1	1	600	10E+14	0.9	2	0.9
t47*	3	10	1	1	600	10E+14	0.9	2	0.85
t53*	3	10	1	1	600	10E+14	0.9	2	0.75
t54*	3	10	1	1	600	10E+14	0.9	2	0.8

Table 5: **Single Density Runs.** *Denotes an experiment for which results are shown in Figure 35.
(t8) Best parameter combination for analysis of 2016 state-of-the-art GLAS-based simulated ICESat-2 data.
(t54) Best parameter combination for a single density run for analysis of 2017-Oct GLAS-based ATL04 data.

For each run, the following plots are shown:

(a) Raw NRB Data - Valid Bins

(b) Kernel Matrix

(c) Density - Valid Bins

(d) Thresholds Along Track

(e) Density - Thresholded

(f) Density - Declustered

(g) Final Cloud Mask

The final mask in figure panel (g) is derived from the mask shown in panel (f) by application of the algorithm for cloud layers (3-bin rules and inclusion of loner bins).

Explanations of results in Figure 35

(1) **Starting from (t8).** As noted in the summary above, the parameter combination $t8$ renders ill-defined layer-tops and bottoms especially at day time (right part of the data set). Some false positives appear, especially around the layer boundaries. False positives are not limited to day-time conditions.

(2) **Fixing the threshold offset. (t32) with $T = 12E + 14$ and (t33) with $T = 10E + 14$.** The first thing to match is the base threshold, or threshold offset, T, to the new types of NRB data. The value used in (t32) works better and is kept for future runs.

(3) **Investigate the role of the segment length. (t34) with $L = 3$t, (t36) with $L = 4$ and (t53) with $L = 20$.** Here we look into the question: can the raggedy tops of layers, especially of optically thick layers, be smoothed by increasing the segment length? Note this is used in the determination of thresholds - same threshold function per segment, and the total length is $2L + 1$. (T8) uses $L = 2$, hence total length 5. Results for (t34) show that the raggedy edges remain but get a little wider, and even wider for (t36). The effect of the larger segment length on the threshold function is obvious for (t53), but the false positives do not disappear. This indicates that the raggedy edges and false positives cannot be smoothed out using threshold segment length. The visually ragged spots are much larger (wider) than the segment length. Keep $L = 2$.

(4) **Do we really need the threshold-sensitivity factor in the threshold function? (t38) with $t = 1$, also (t44) , (t45), (t46), (t47), (t53) and (t54).** It turns out that it is much harder to achieve a well-working threshold function without using a threshold-sensitivity factor. Experiment (t38) shows this. Other combinations of threshold function parameters were also tried, as documented in Table 5. use $t = 0.9$.

(5) **Setting quantile. (t44) , (t45), (t46), (t47), (t53) and (t54). Quantile 0.75, 0.8, 0.85, 0.9**. Selected $Q = 0.8$. (t44) with $q = 0.9$ - quantile too high, $t = 1$ too high, clouds are missed.

(t45) with $q = 0.85$ - quantile too high, $t = 1$ too high, clouds are missed.

(t46) with $q = 0.9$ - quantile too high, $t = 0.9$, more clouds, but too many tenuous clouds are still missed. No rain falls out of the clouds.

(t47) with $q = 0.85$ - quantile too high, $t = 0.9$, more clouds, but too many tenuous clouds are still missed.

(t54) with $q = 0.8$ - best.

(6) **(t54) is the best parameter combination for a single-density run.** But it does not meet the two requirements (no false positives, no raggedy edges), but retain tenuous clouds, aerosol layers in day-time and night time, keep cloud structure (in left segment), keep rain falling out of the clouds.

This necessitates application of running density twice to meet the cloud detection criteria!

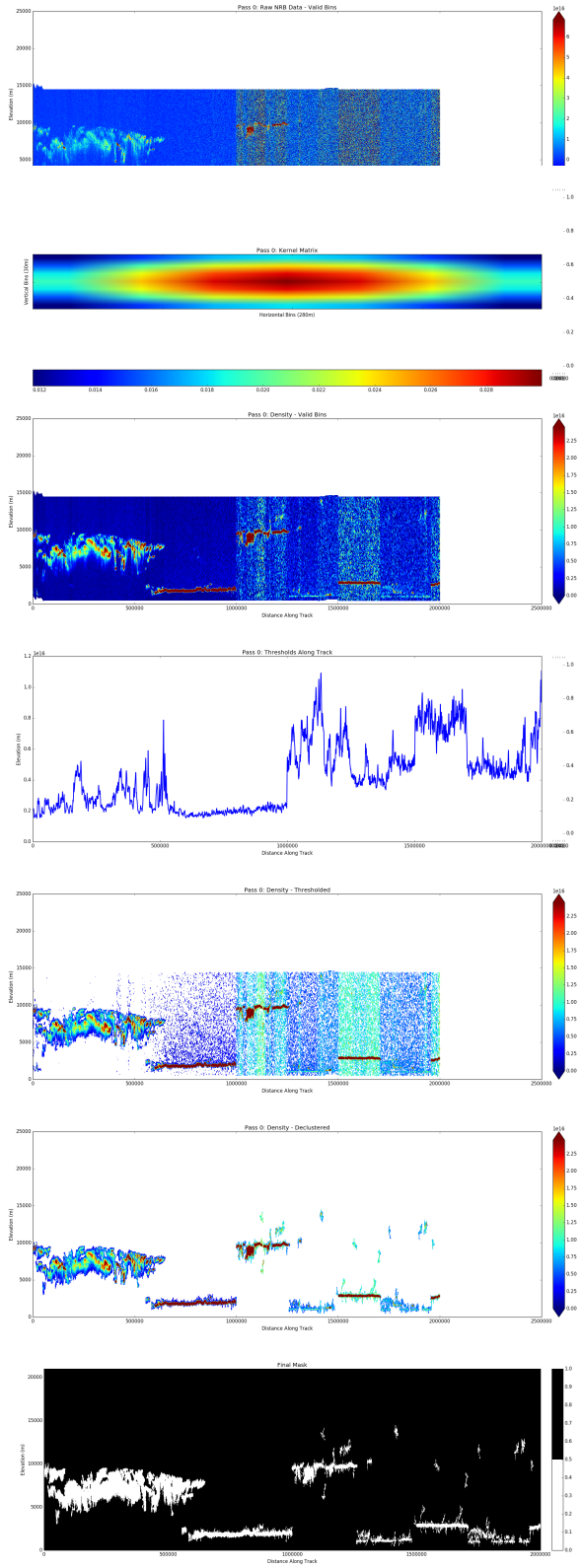


Figure 35-1. (t8) - Analysis of GLAS–data based simulated ICESat-2 data using the DDA.

$\sigma = 3$	cutoff = 1 $a_m = 10$	min cluster size = 600 base_threshold = 6E+14	downsampling = 1
threshold_sensitivity = 0.9	threshold_segment_length = 2	quantile = 0.75	

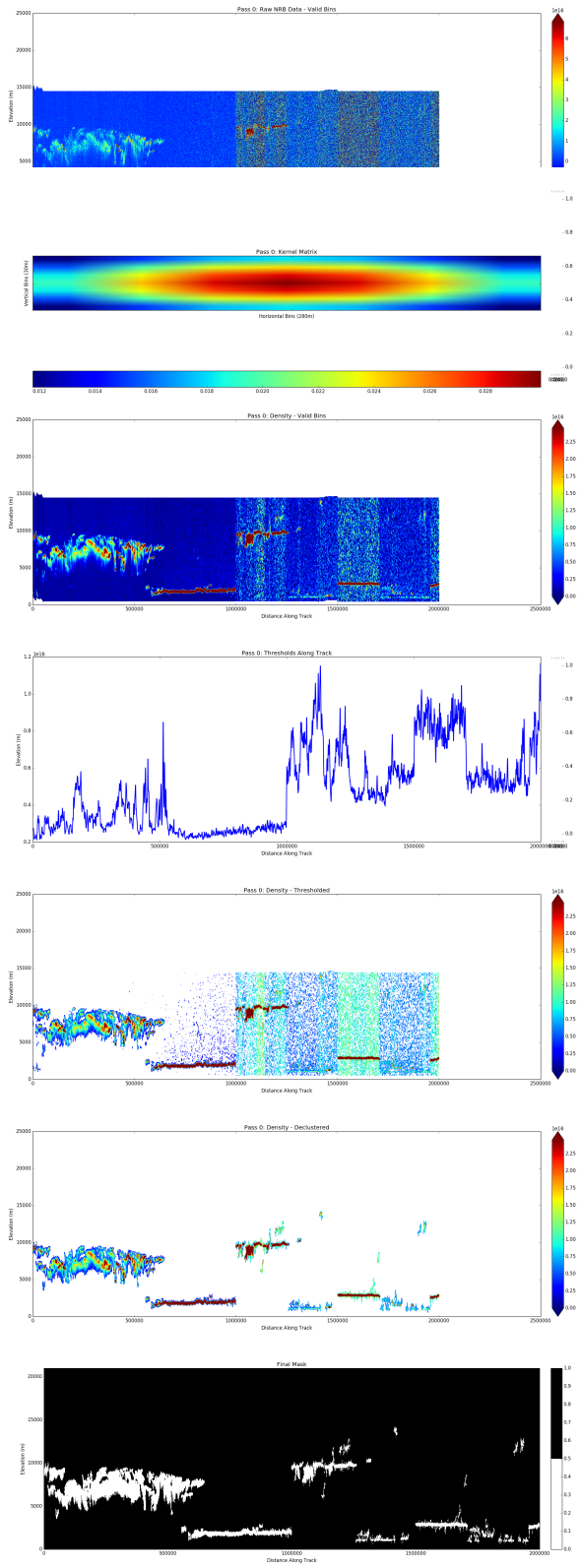


Figure 35-2. (t32) - Analysis of GLAS–data based simulated ICESat-2 data using the DDA.

$\sigma = 3$	cutoff = 1 $a_m = 10$	min cluster size = 600 base_threshold = 12E+14	downsampling = 1
threshold_sensitivity = 0.9	threshold_segment_length = 2	quantile = 0.75	

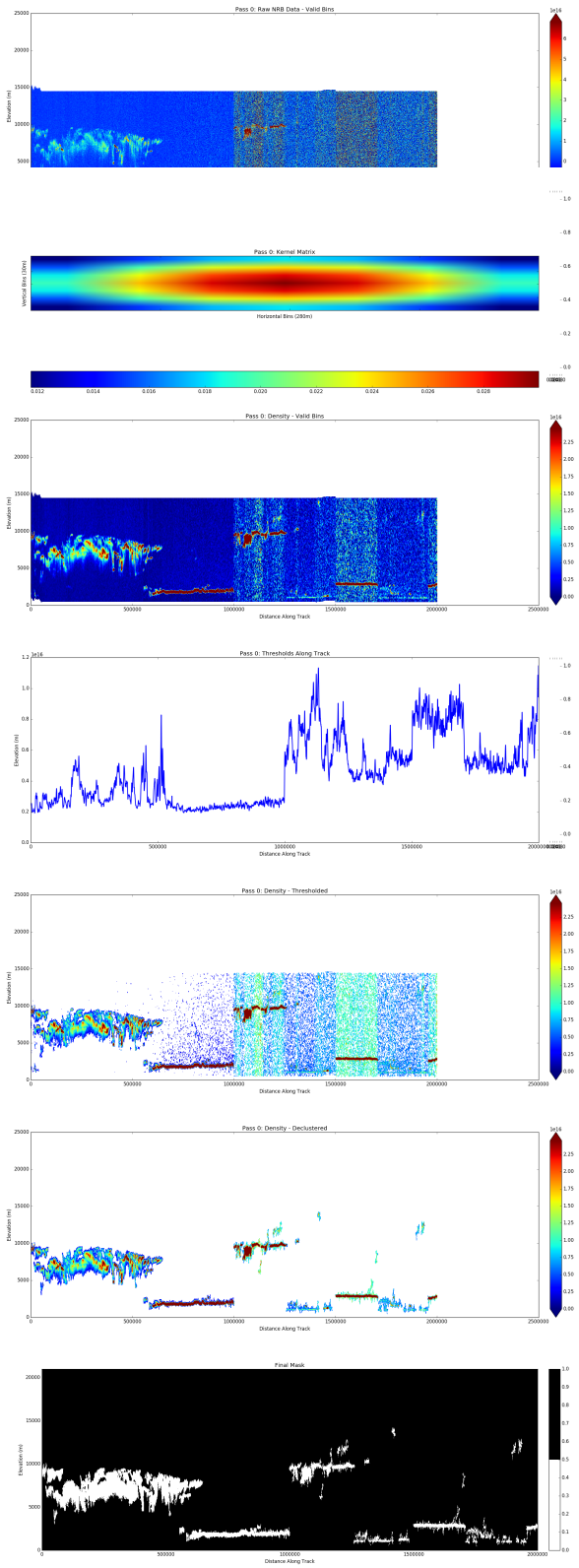


Figure 35-3. (t33) - Analysis of GLAS–data based simulated ICESat-2 data using the DDA.

$\sigma = 3$	cutoff = 1 $a_m = 10$	min cluster size = 600 base_threshold = 10E+14	downsampling = 1
threshold_sensitivity = 0.9	threshold_segment_length = 2	quantile = 0.75	

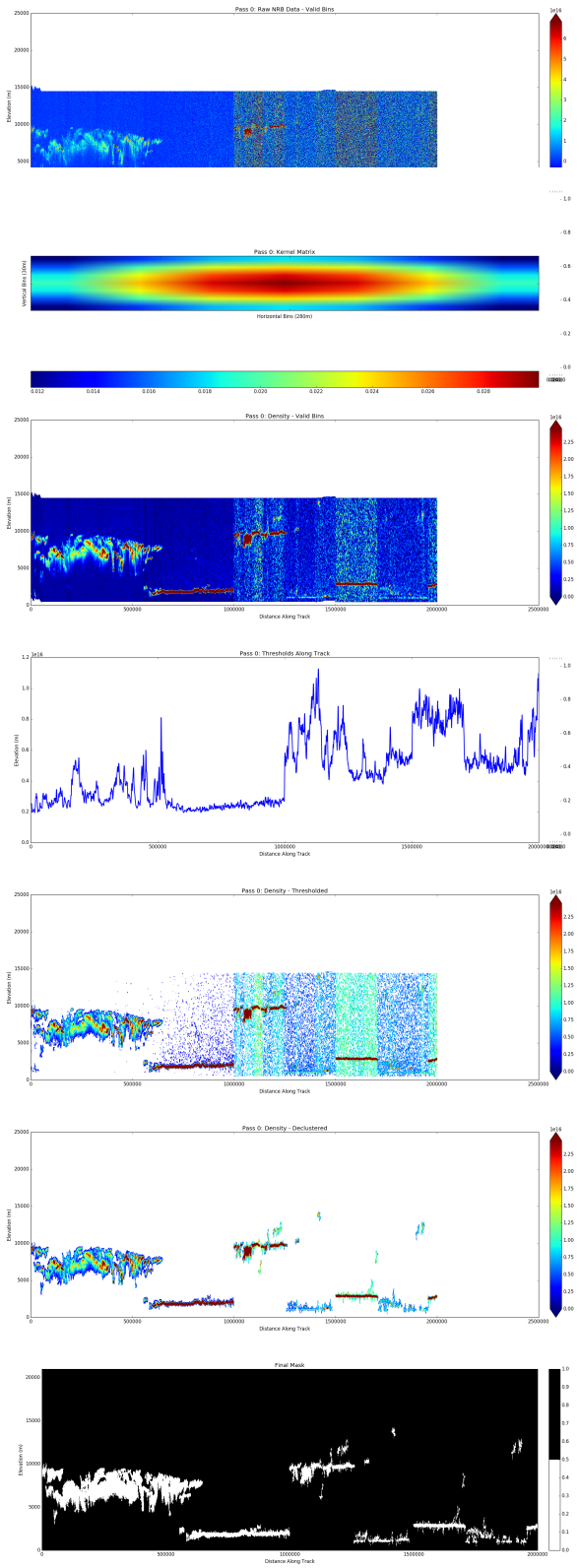


Figure 35-4. (t34) - Analysis of GLAS–data based simulated ICESat-2 data using the DDA.

$\sigma = 3$ cutoff $= 1$ $a_m = 10$ min cluster size $= 600$ base_threshold $=$ 10E+14 downsampling $= 1$ threshold_sensitivity $= 0.9$ threshold_segment_length $= 3$ quantile $= 0.75$

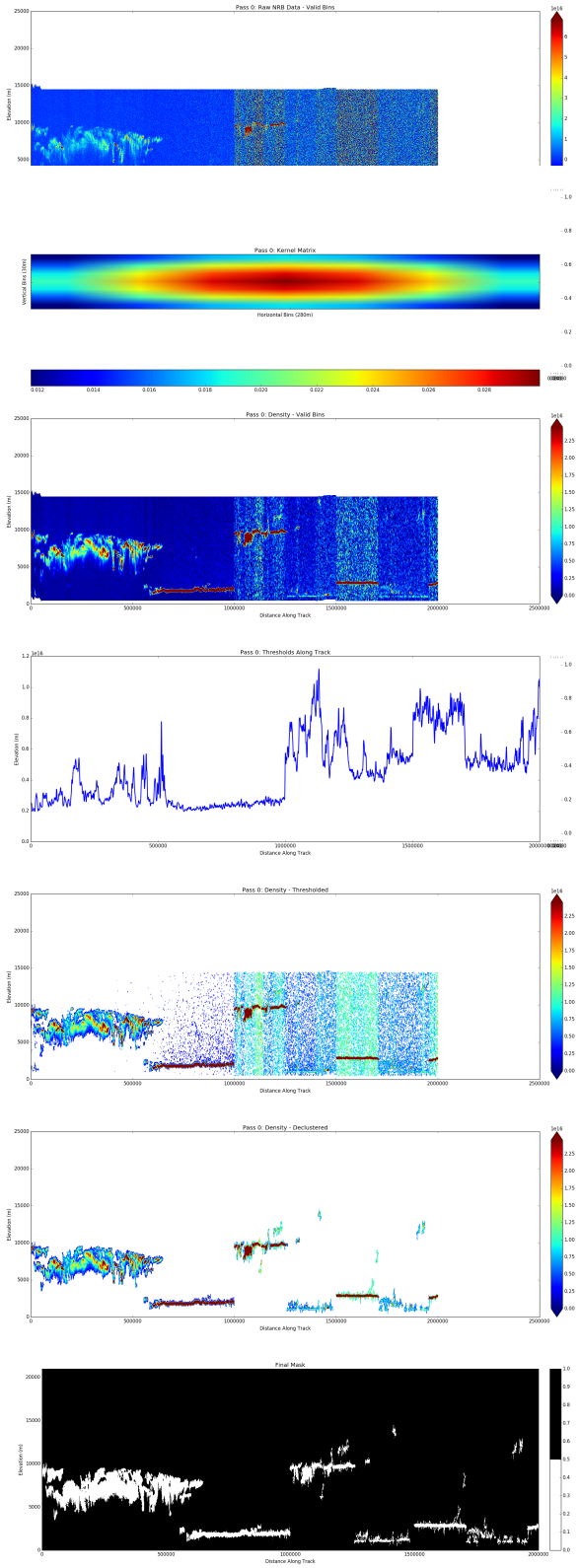


Figure 35-5. (t36) - Analysis of GLAS–data based simulated ICESat-2 data using the DDA.

$\sigma = 3$	cutoff = 1 $a_m = 10$	min cluster size = 600 base_threshold = 10E+14	downsampling = 1
threshold_sensitivity = 0.9	threshold_segment_length = 4	quantile = 0.75	

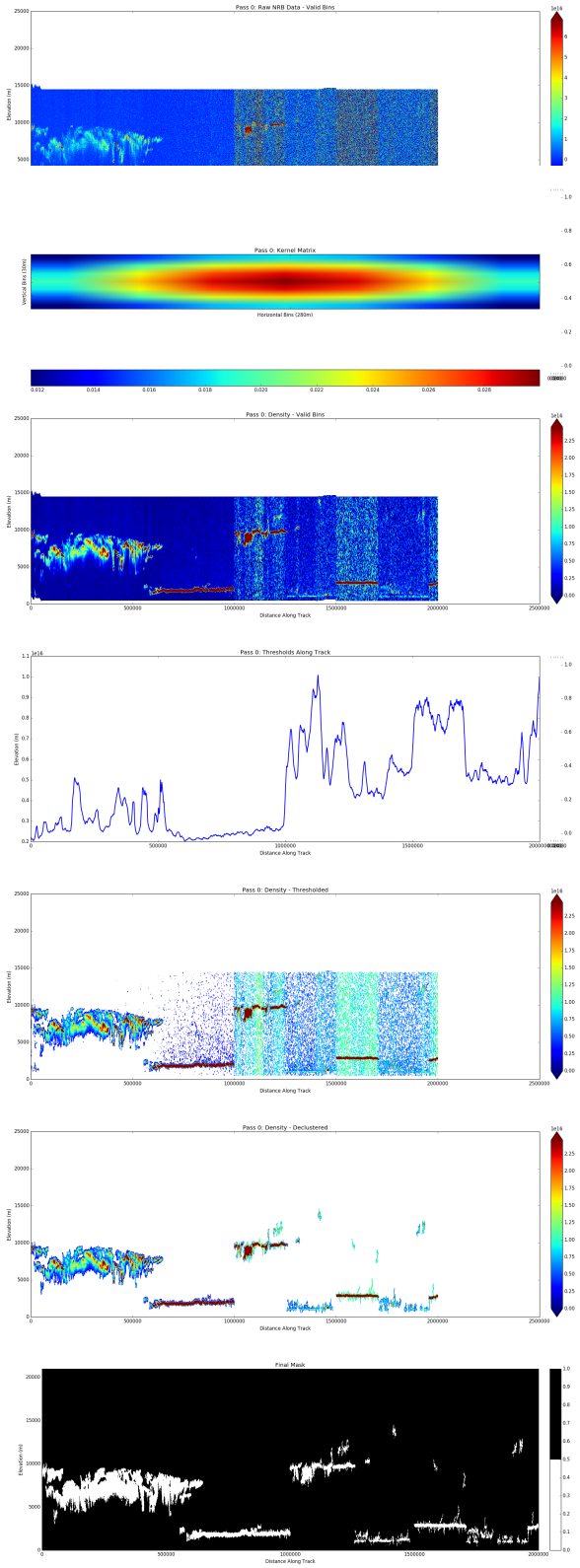


Figure 35-6. (t53) - Analysis of GLAS–data based simulated ICESat-2 data using the DDA.

$\sigma = 3$	cutoff $= 1$ $a_m = 10$	min cluster size $= 600$ base_threshold $=$ 10E+14	downsampling $= 1$
threshold_sensitivity $= 0.9$	threshold_segment_length $= 20$		quantile $= 0.75$

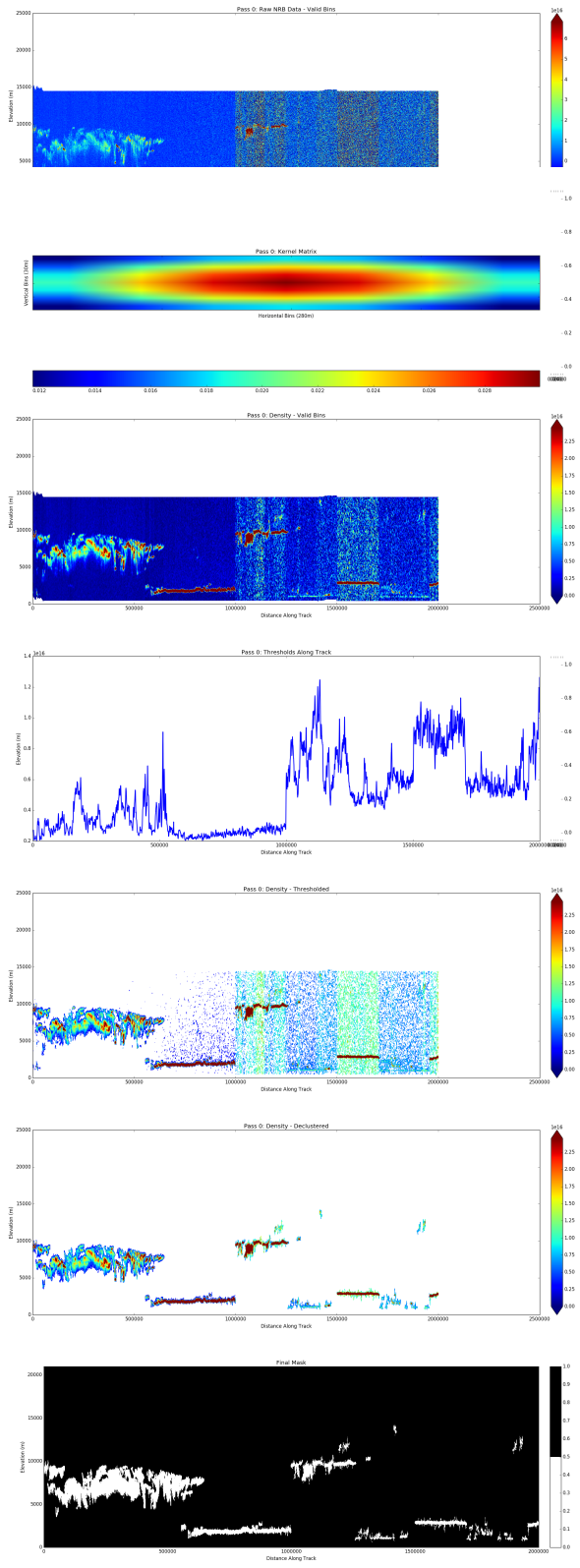


Figure 35-7. (t38) - Analysis of GLAS–data based simulated ICESat-2 data using the DDA.

$\sigma = 3$	cutoff = 1 $a_m = 10$	min cluster size = 600 base_threshold = 10E+14	downsampling = 1
threshold_sensitivity = 1	threshold_segment_length = 2	quantile = 0.75	

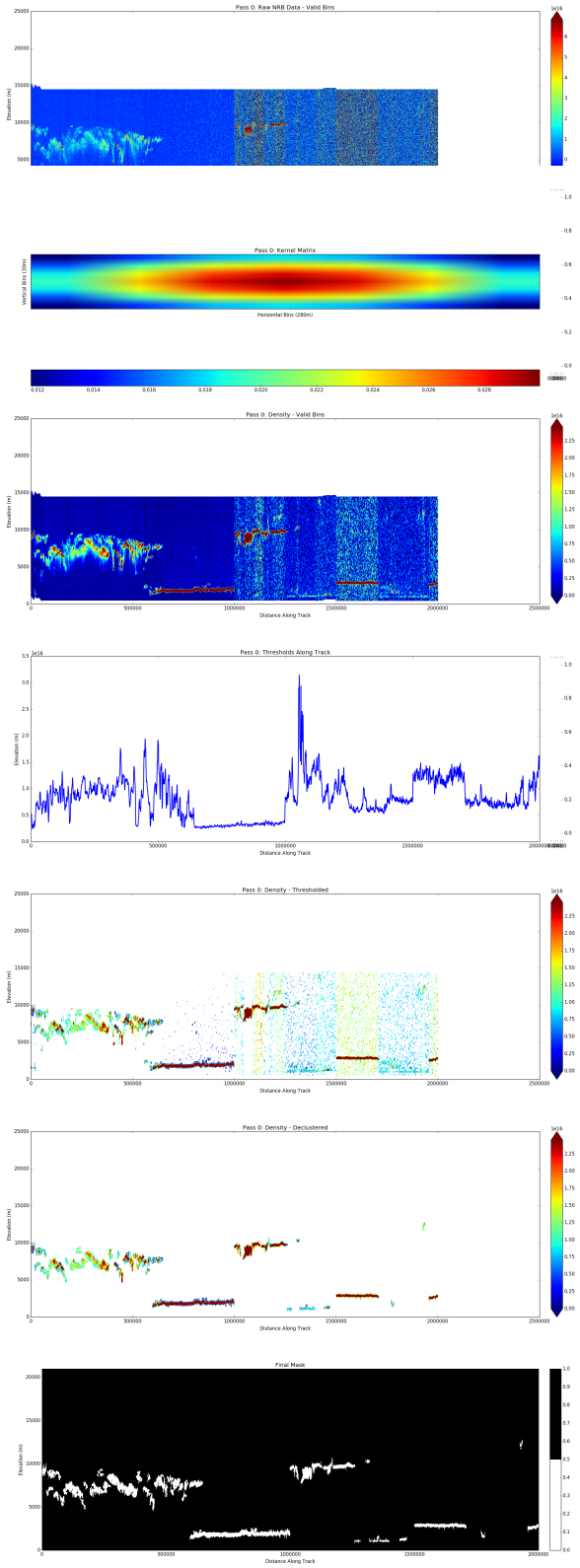


Figure 35-8. (t44) -Analysis of GLAS–data based simulated ICESat-2 data using the DDA.

$\sigma = 3$	cutoff = 1 $a_m = 10$	min cluster size = 600 base_threshold = 10E+14	downsampling = 1
threshold_sensitivity = 1	threshold_segment_length = 2	quantile = 0.9	

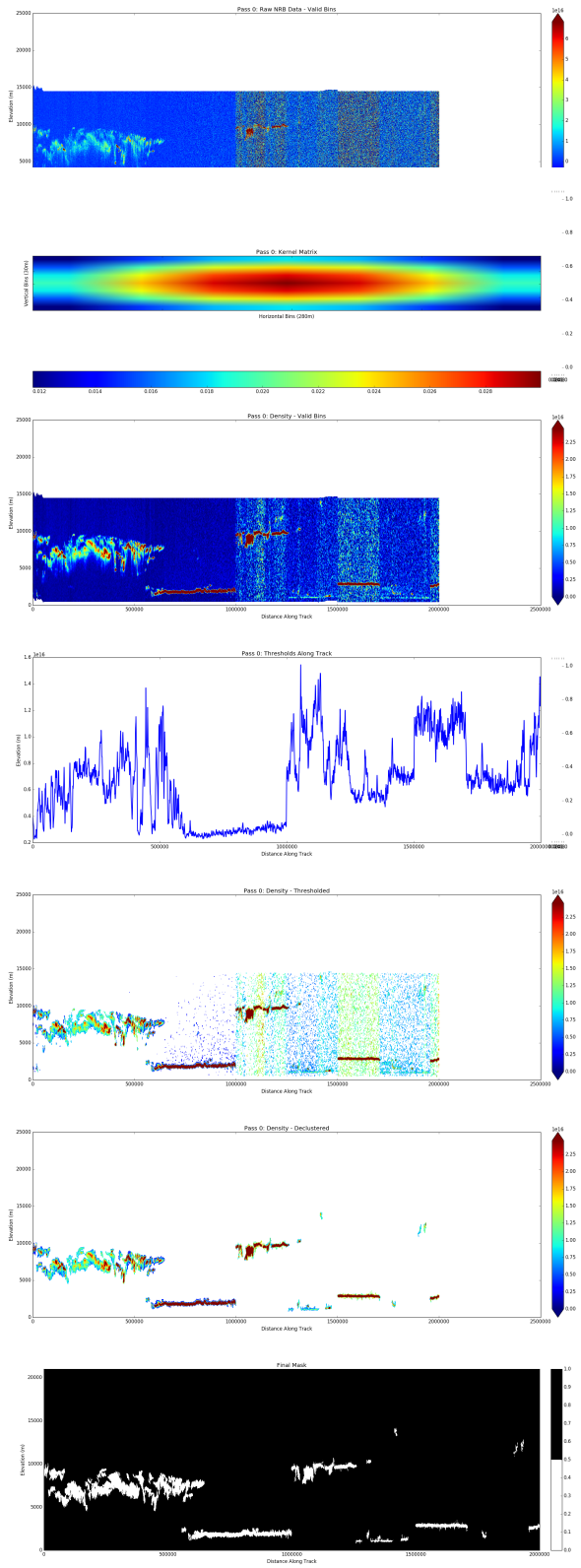


Figure 35-9. (t45) -Analysis of GLAS–data based simulated ICESat-2 data using the DDA.

$\sigma = 3$	cutoff = 1 $a_m = 10$	min cluster size = 600 base_threshold = 10E+14	downsampling = 1
threshold_sensitivity = 1	threshold_segment_length = 2	quantile = 0.85	

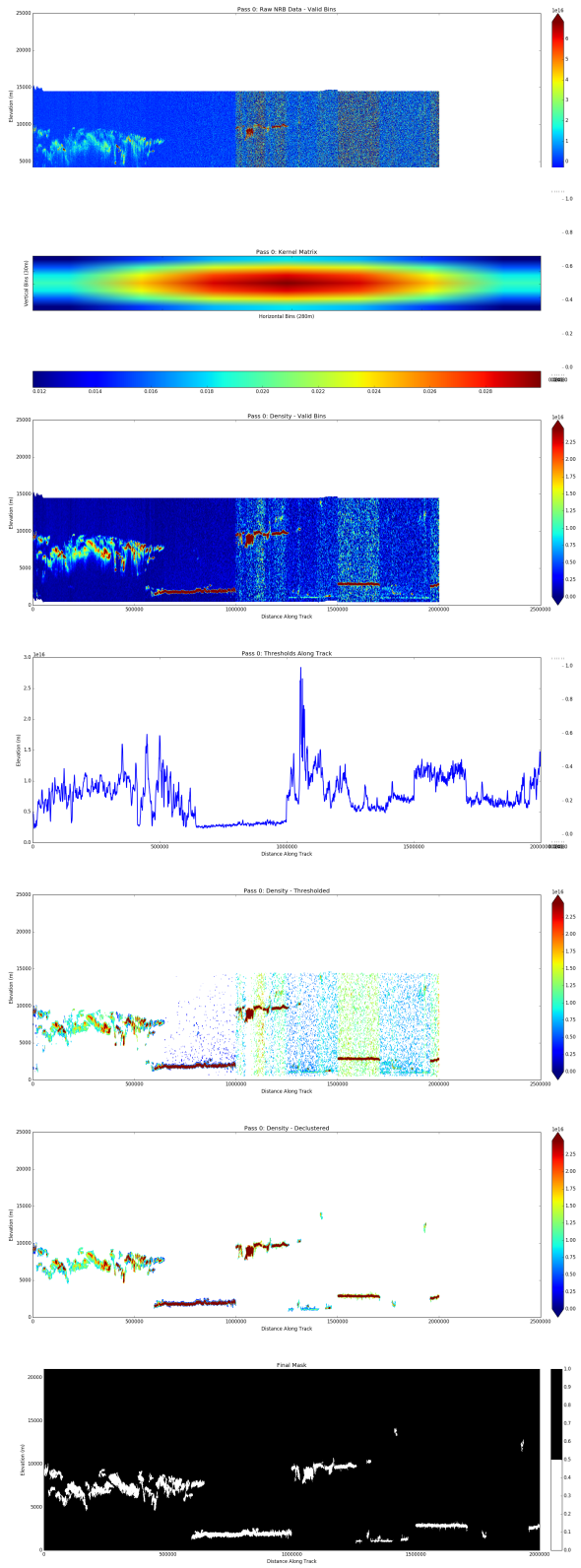


Figure 35-10. (t46) - Analysis of GLAS–data based simulated ICESat-2 data using the DDA.

$\sigma = 3$	cutoff = 1 $a_m = 10$	min cluster size = 600 base_threshold = 10E+14	downsampling = 1
threshold_sensitivity = 0.9	threshold_segment_length = 2	quantile = 0.9	

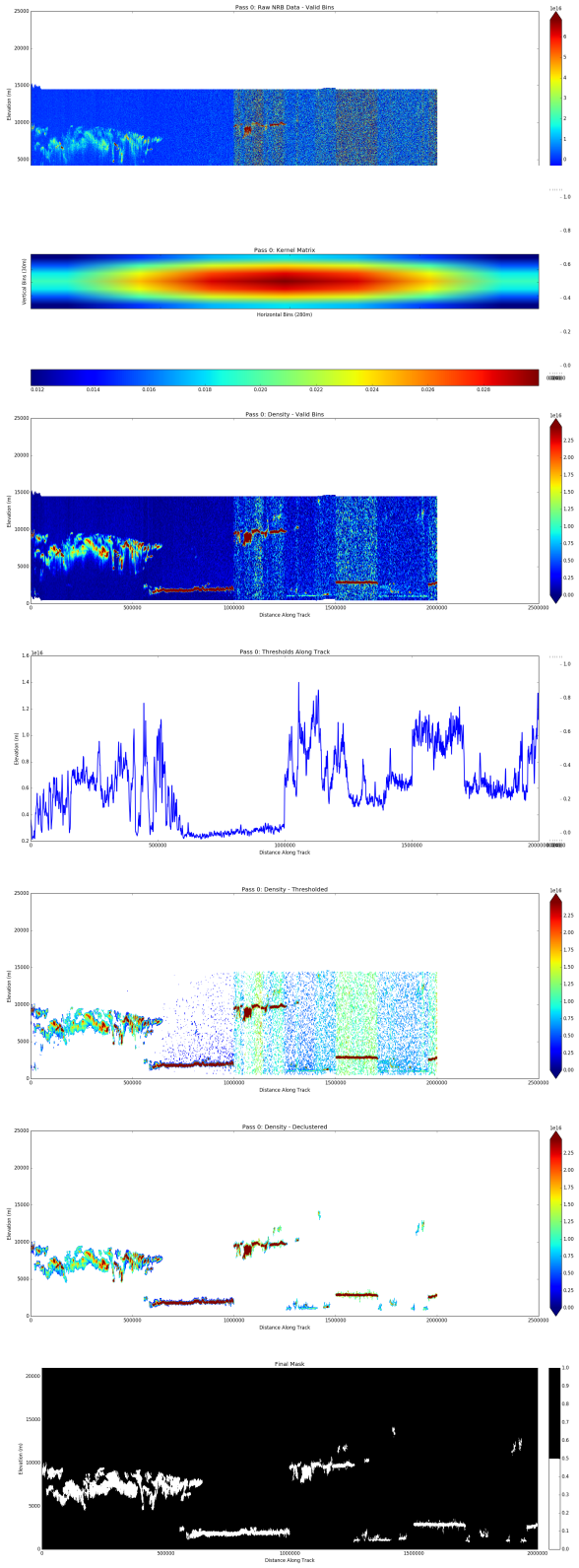


Figure 35-11. (t47)- Analysis of GLAS–data based simulated ICESat-2 data using the DDA.

$\sigma = 3$	cutoff = 1 $a_m = 10$	min cluster size = 600 base_threshold = 10E+14	downsampling = 1
threshold_sensitivity = 0.9	threshold_segment_length = 2	quantile = 0.85	

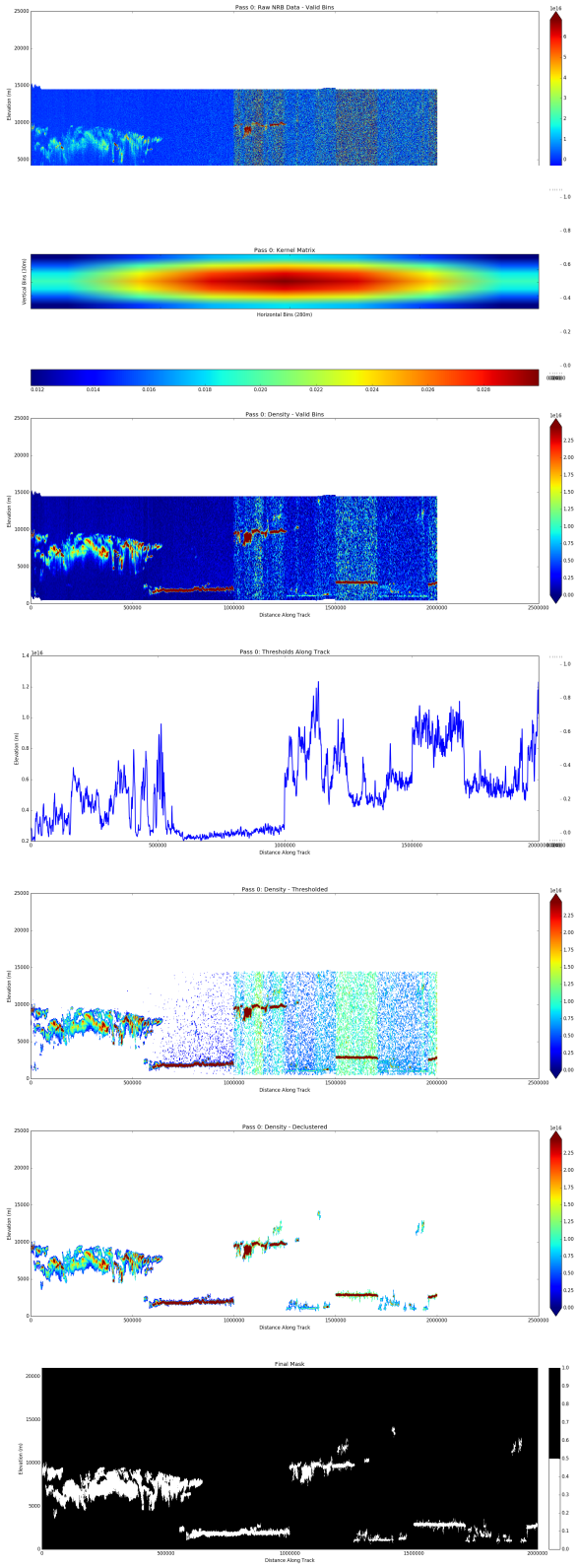


Figure 35-12. (t54) - Analysis of GLAS–data based simulated ICESat-2 data using the DDA.

$\sigma = 3$	cutoff = 1 $a_m = 10$	min cluster size = 600 base_threshold = 10E+14	downsampling = 1
threshold_sensitivity = 0.9	threshold_segment_length = 2	quantile = 0.8	

10.5 Sensitivity Studies for Double-Density Runs

The following table shows the parameter combinations that were used for double-density runs:

Parameter Set Name	Sigma	Anisotropy	Cutoff	Down-sample	Minimum Cluster Size	Threshold Bias	Threshold Factor	Threshold Window	Quantile
t48*	3	10	1	1	600	10E+14	1	2	0.9, 0.7
t49*	3	10	1	1	600	10E+14	1	2	0.95, 0.7
t50	3	10	1	1	600	10E+14	1	2	0.99, 0.8
t51	3	10, 20	1	1	600	10E+14	1	2	0.99, 0.8
t52	3	10, 20	1	1	300, 600	10E+14	1	2	0.99, 0.8
t55*	3	10, 20	1	1	300, 600	10E+14	1, 0.9	2	0.99, 0.8
t56*	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.99, 0.8
t57	3	10, 15	1	1	300, 600	10E+14	0.9, 1	2	0.99, 0.8
t58*	3	10, 20	1	1	400, 600	10E+14	0.9, 1	2	0.99, 0.8
t59*	3	10, 20	1	1	300, 600	12E+14	0.9, 1	2	0.99, 0.8
t60	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.95, 0.8
t61	3	10, 15	1	1	300, 600	10E+14	0.9, 1	2	0.95, 0.8
t62	3	10, 15	1	1	300, 600	12E+14	0.9, 1	2	0.95, 0.8
t63*	3	10, 30	1	1	300, 600	10E+14	0.9, 1	2	0.99, 0.8
t64*	3	10, 20	1	1	200, 600	10E+14	0.9, 1	2	0.99, 0.8
t65*	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.97, 0.8
t66*	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.98, 0.8
t67*	3	10, 20	1	1	150, 600	10E+14	0.9, 1	2	0.99, 0.8

Table 6: **Double Density Runs.** *Denotes an experiment for which results are shown in Figure 36. (t56) Best parameter combination for a double density run for analysis of 2017-Oct GLAS-based ATL04 data. (t64) Alternative best parameter combination for a double density run for analysis of 2017-Oct GLAS-based ATL04 data (declustering size 200 in run 1, otherwise same as (t56)).

For each experiment, the following plots are shown (left column, density run 1 (labeled 0), right column, density run 2 (labeled 1, py convention)):

(a) Raw NRB Data - Valid Bins

(b) Kernel Matrix

(c) Density - Valid Bins

(d) Thresholds Along Track

(e) Density - Thresholded

(f) Density - Declustered

(g) left: Final Cloud Mask (Binary) , right: Final Cloud Mask over Density 1

Explanations of results in Figure 36

(1) Balance the quantiles between density run1 and density run2 **(t48)-(t50).** (t48), (t49) Not good. Keep false positives, make holes in the tenuous cloud layers. Threshold function is bad, quantiles too low. (t50) reaches acceptable quantiles.

(2) Different anisotropies for density run1 and density run2. Also different min cluster sizes. (t56) is the best parameter combination. Notably, (t55), where the threshold sensitivity factors are switched, is a lot worse. (t56) identifies just enough clouds in pass 1, to bring out the more tenuous layers in the second density run. Layers are smooth on top, with rain falling out of the bottom. Note that the small clouds (day-time section) are actually clouds, visible in the density fields. (t63) with anisotropy (10,30) makes the tenuous clouds a little too wide. Use (10,20) for anisotropies.

(3) Different cluster sizes. (t64) with min cluster size 200, otherwise parameters same as (t56). Mean confidence for (t64) is higher (0.801) than for (t56), making this a slightly better parameter combination than (t56) (conf 0.79 on average). (t67) with min cluster size 150: results are not much different from those of (t64). Because more day time data need to be analyzed and because of robustness, keep (t64) as the best run. See the section on "Testing" - the SIPS code uses a somewhat different implementation of the declustering routine, this still needs to be tested. Hence opt for min cluster size of 200.

(4) **Checking intermediate quantiles, (t65) with** $q = 0.97, 0.8$**, (t66) with** $q = 0.98, 0.8$**.** The quantiles of $q = 0.99, 0.8$ indeed yield the best results, evaluated by the same criteria as before. Hence (t56) and (t64) are best. Testing of the declustering routines is still to be done.

In summary, very good results are obtained using double-density runs with parameters (t56)!

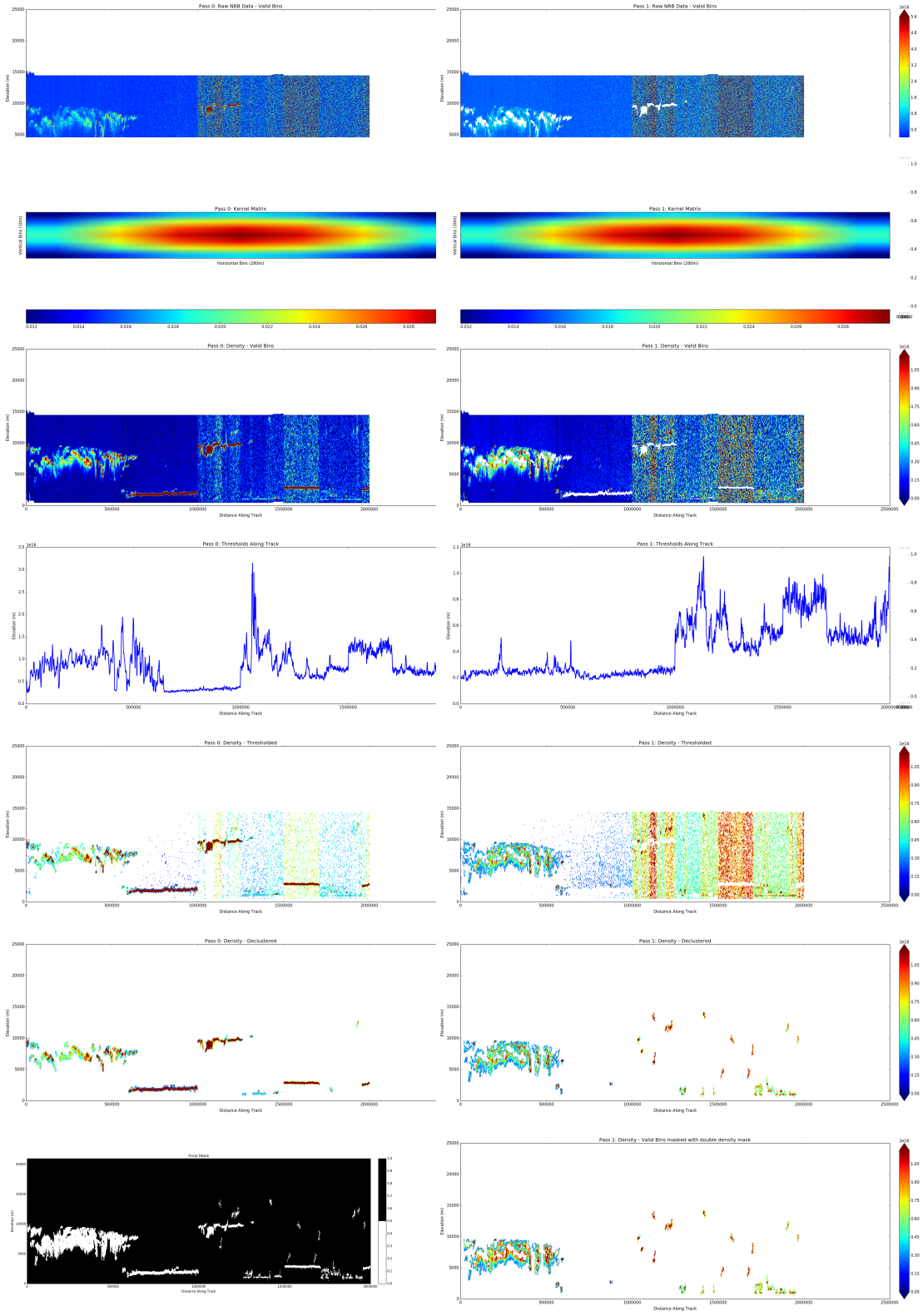


Figure 36-1. (t48) - Analysis of GLAS–data based simulated ICESat-2 data using the DDA.

$\sigma = 3$	cutoff = 1	$a_m = 10$	min cluster size = 600	base_threshold = 10E+14	downsampling = 1
threshold_sensitivity = 1		threshold_segment_length = 2		quantile = 0.9,0.7	

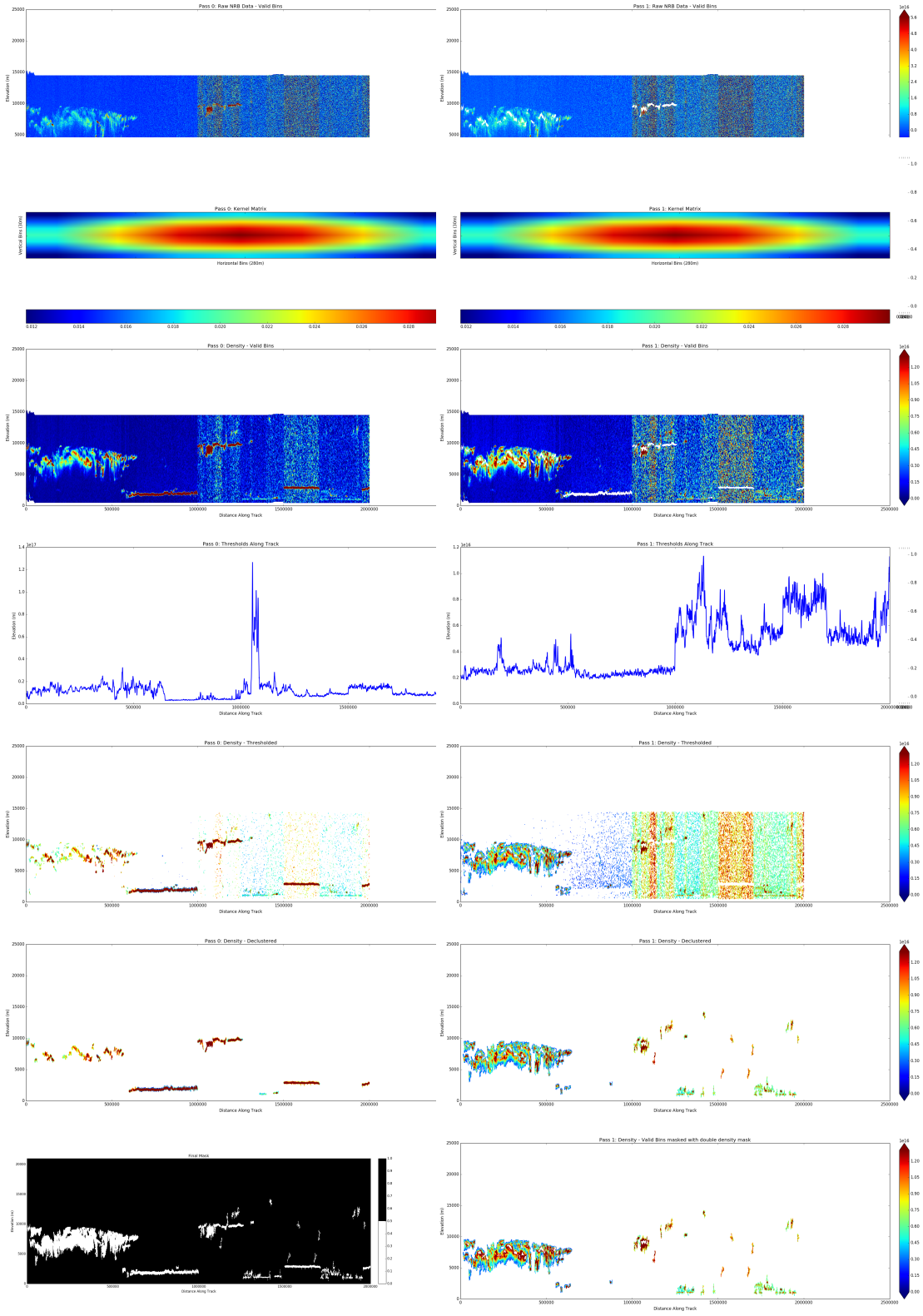


Figure 36-2. (t49) - Analysis of GLAS–data based simulated ICESat-2 data using the DDA.

$\sigma = 3$	cutoff = 1	$a_m = 10$	min cluster size = 600	base_threshold = 10E+14	downsampling = 1
threshold_sensitivity = 1		threshold_segment_length = 2		quantile = 0.95,0.7	

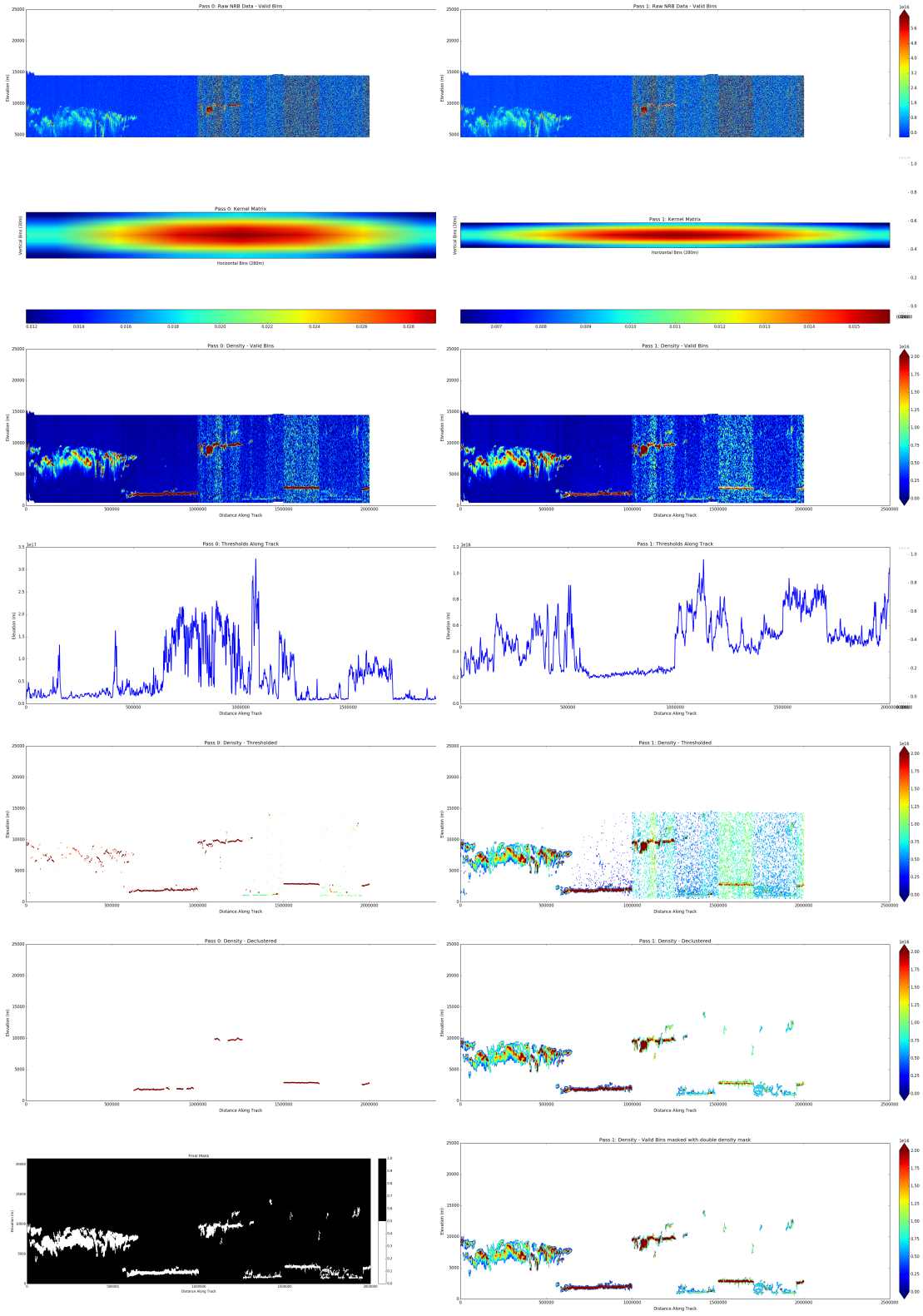


Figure 36-3. (t55) - Analysis of GLAS–data based simulated ICESat-2 data using the DDA.

$\sigma = 3$	cutoff = 1	a_m = 10,20	min cluster size = 300,600	base_threshold = 10E+14	downsampling = 1
threshold_sensitivity = 1, 0.9		threshold_segment_length = 2		quantile = 0.99, 0.8	

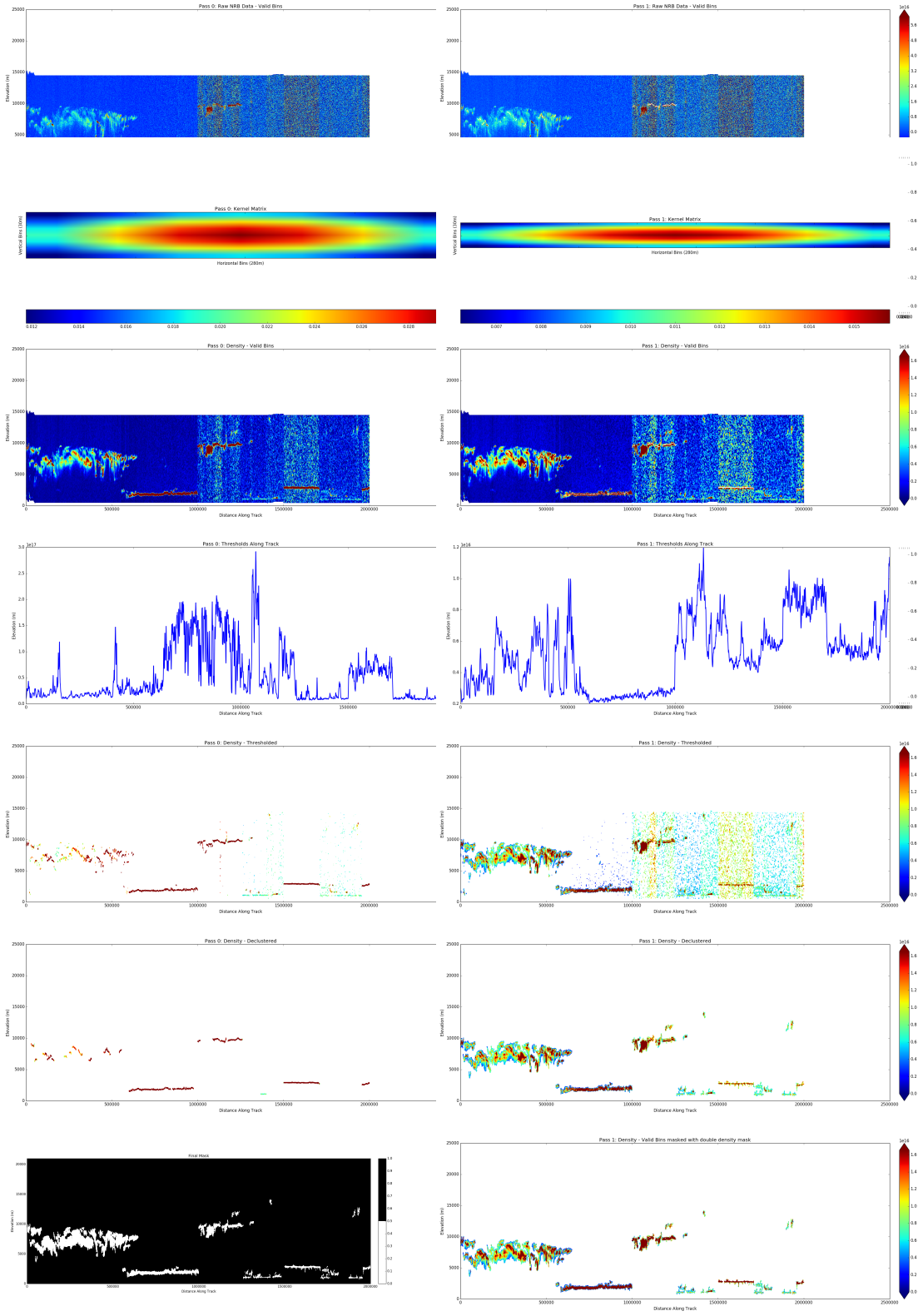


Figure 36-4. (t56) - Analysis of GLAS–data based simulated ICESat-2 data using the DDA.

$\sigma = 3$	cutoff = 1	a_m = 10,20	min cluster size = 300,600	base_threshold = 10E+14	downsampling = 1
threshold_sensitivity = 0.9, 1		threshold_segment_length = 2		quantile = 0.99, 0.8	

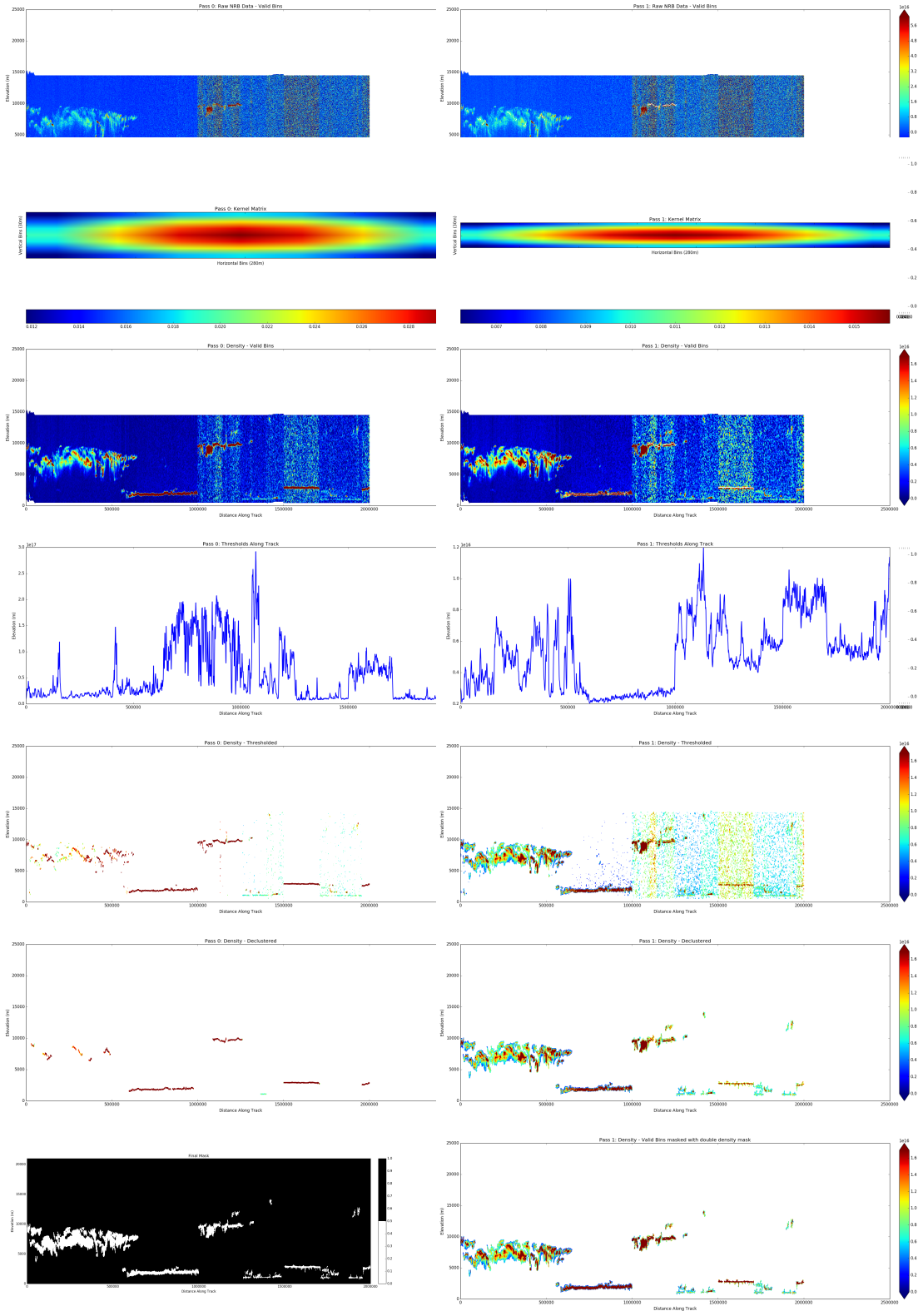


Figure 36-5. (t58) - Analysis of GLAS–data based simulated ICESat-2 data using the DDA.

$\sigma = 3$	cutoff = 1	a_m = 10,20	min cluster size = 400,600	base_threshold = 10E+14	downsampling = 1
threshold_sensitivity = 0.9,1		threshold_segment_length = 2		quantile = 0.99, 0.8	

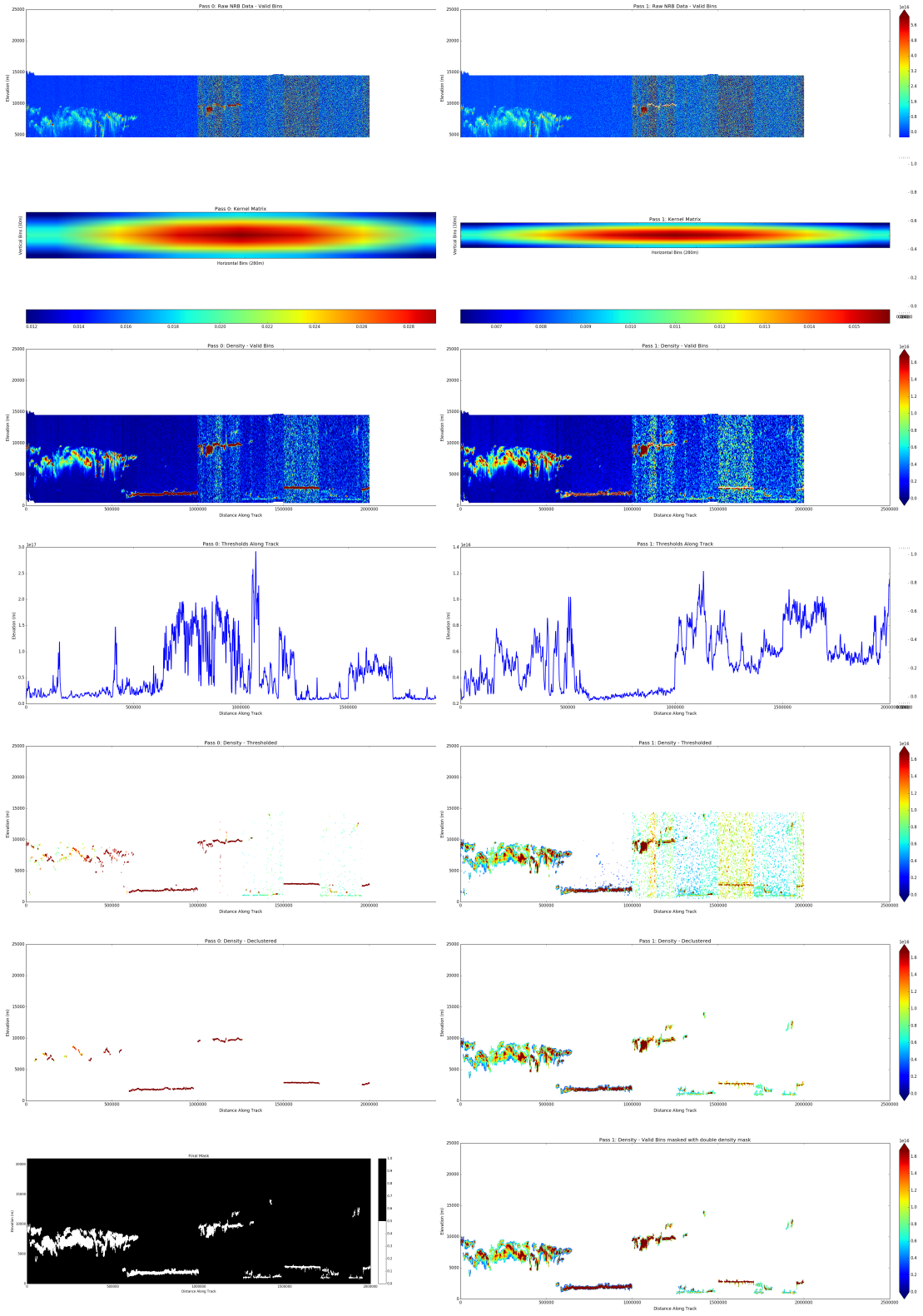


Figure 36-6. (t59) - Analysis of GLAS–data based simulated ICESat-2 data using the DDA.

$\sigma = 3$	cutoff = 1	a_m = 10,20	min cluster size = 300,600	base_threshold = 10E+14	downsampling = 1
threshold_sensitivity = 0.9, 1		threshold_segment_length = 2		quantile = 0.99, 0.8	

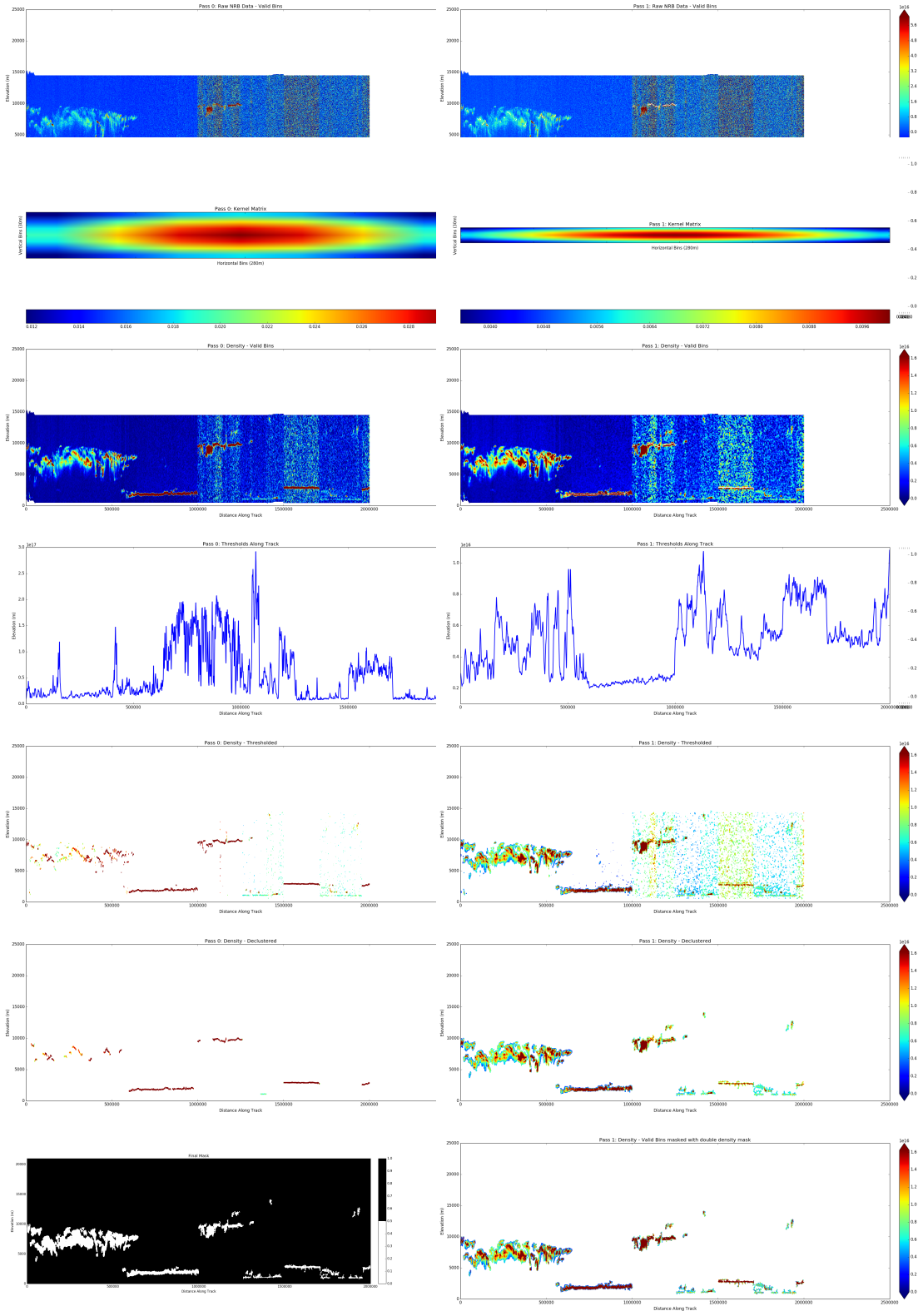


Figure 36-7. (t63) - Analysis of GLAS–data based simulated ICESat-2 data using the DDA.

$\sigma = 3$	cutoff = 1	a_m = 10,30	min cluster size = 300, 600	base_threshold = 10E+14	downsampling = 1
threshold_sensitivity = 0.9, 1		threshold_segment_length = 2		quantile = 0.99, 0.8	

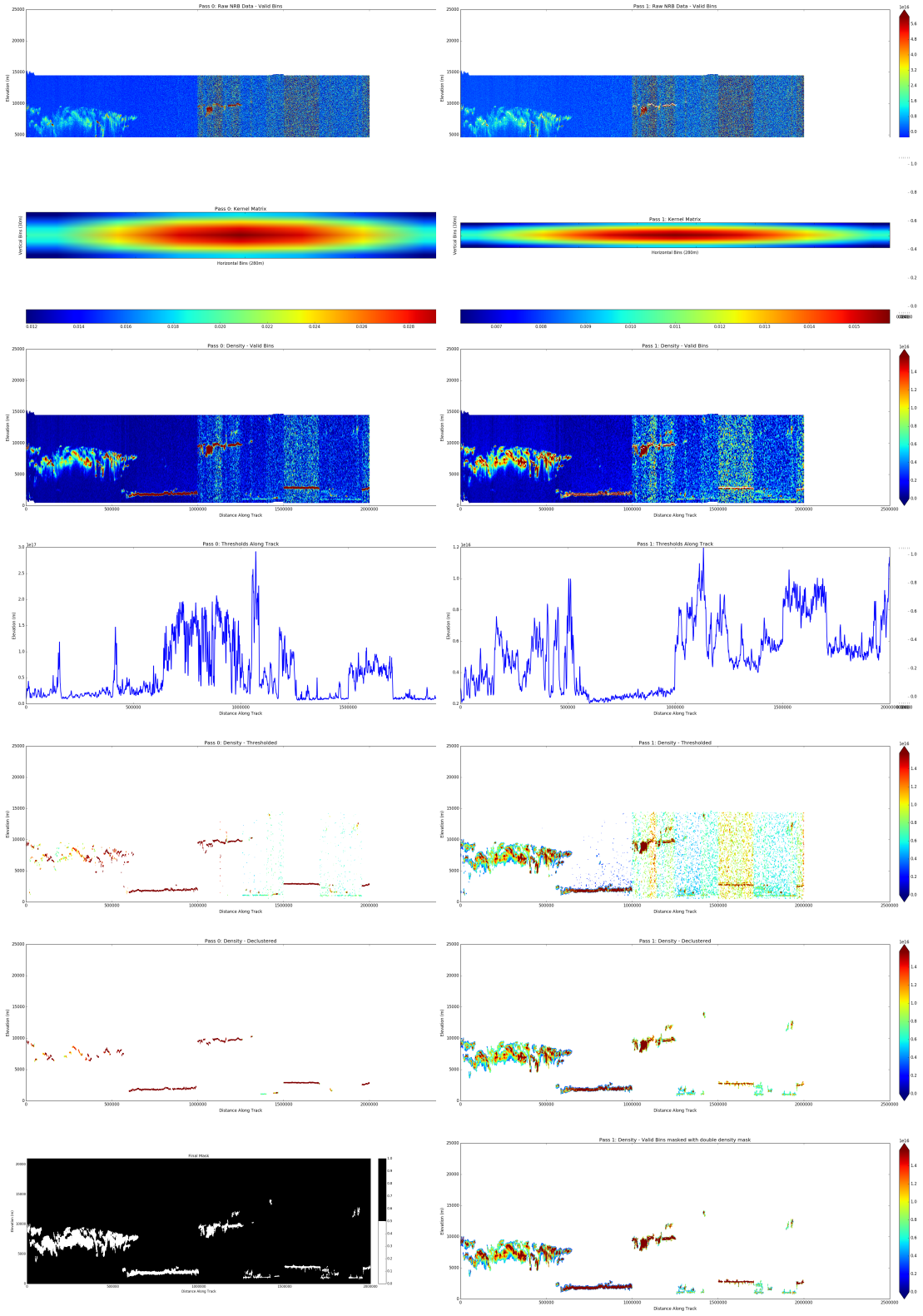


Figure 36-8. (t64) - Analysis of GLAS–data based simulated ICESat-2 data using the DDA.

$\sigma = 3$	cutoff = 1	a_m = 10,20	min cluster size = 600	base_threshold = 10E+14	downsampling = 1
threshold_sensitivity = 0.9, 1		threshold_segment_length = 2		quantile = 0.99, 0.8	

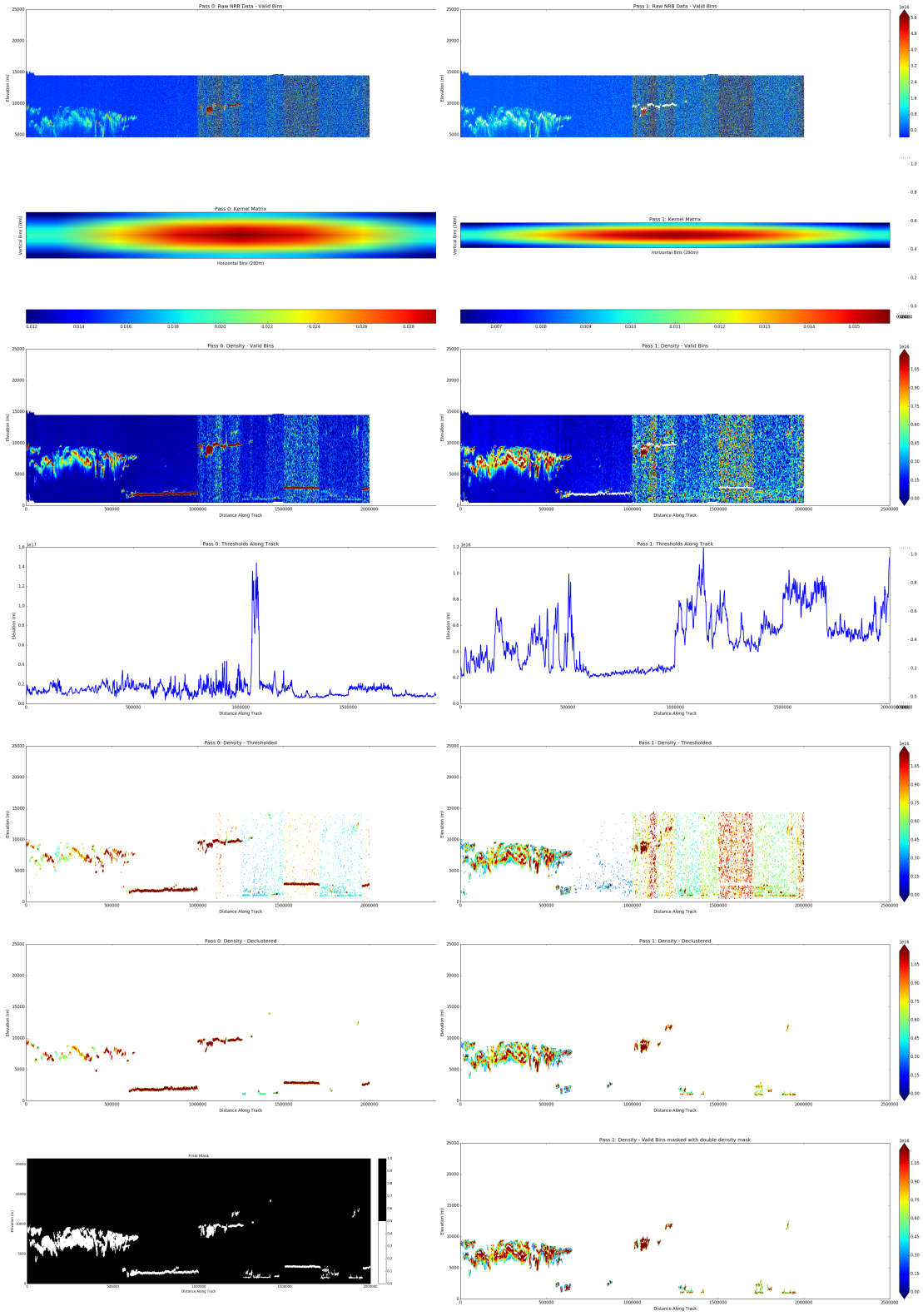


Figure 36-9. (t65) - Analysis of GLAS–data based simulated ICESat-2 data using the DDA.

$\sigma = 3$	cutoff = 1	a_m = 10,20	min cluster size = 300, 600	base_threshold = 10E+14	downsampling = 1
threshold_sensitivity = 0.9, 1		threshold_segment_length = 2		quantile = 0.97, 0.8	

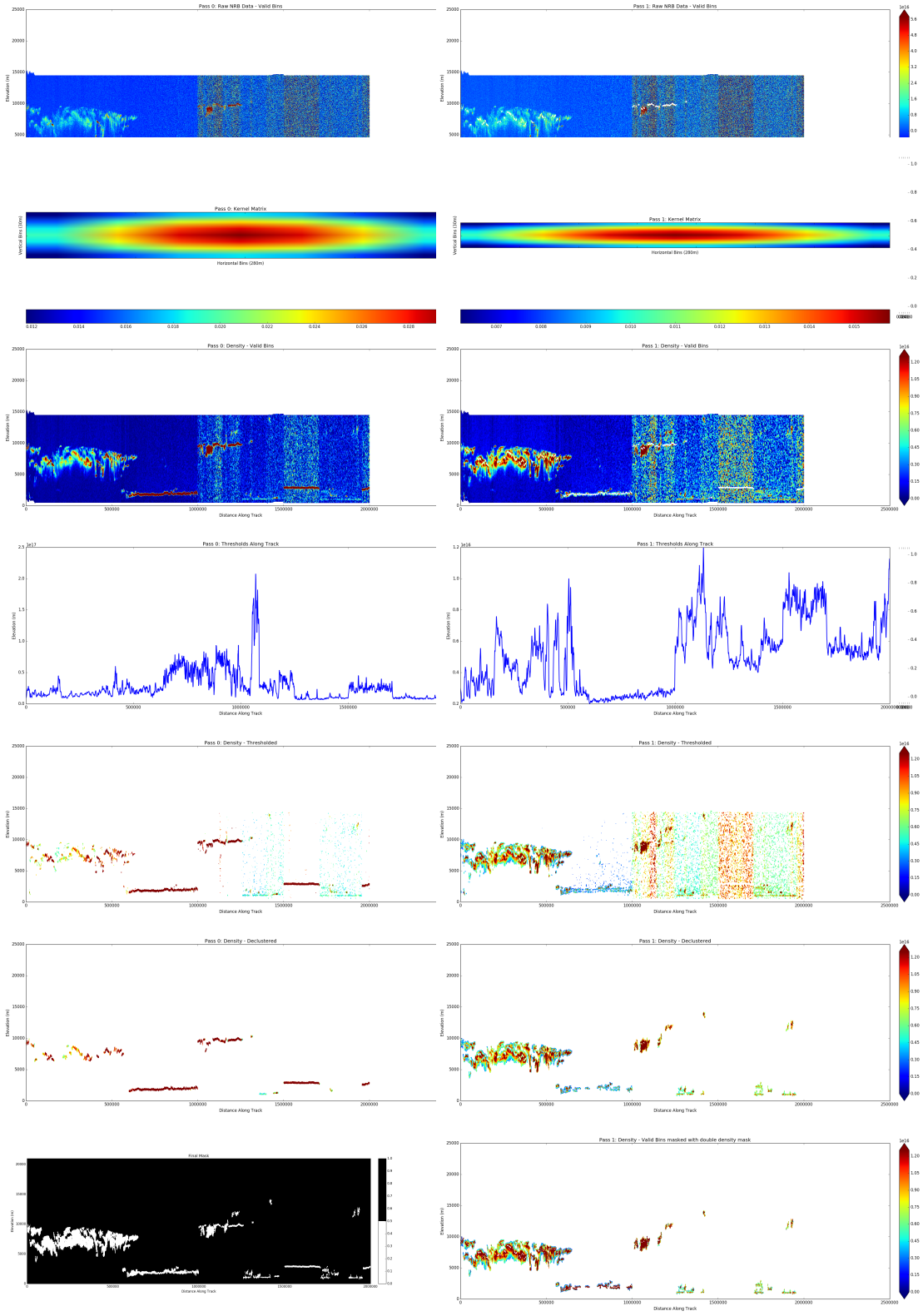


Figure 36-10. (t66) - Analysis of GLAS–data based simulated ICESat-2 data using the DDA.

$\sigma = 3$	cutoff = 1	a_m = 10,20	min cluster size = 300, 600	base_threshold = 10E+14	downsampling = 1
threshold_sensitivity = 0.9, 1		threshold_segment_length = 2		quantile = 0.98, 0.8	

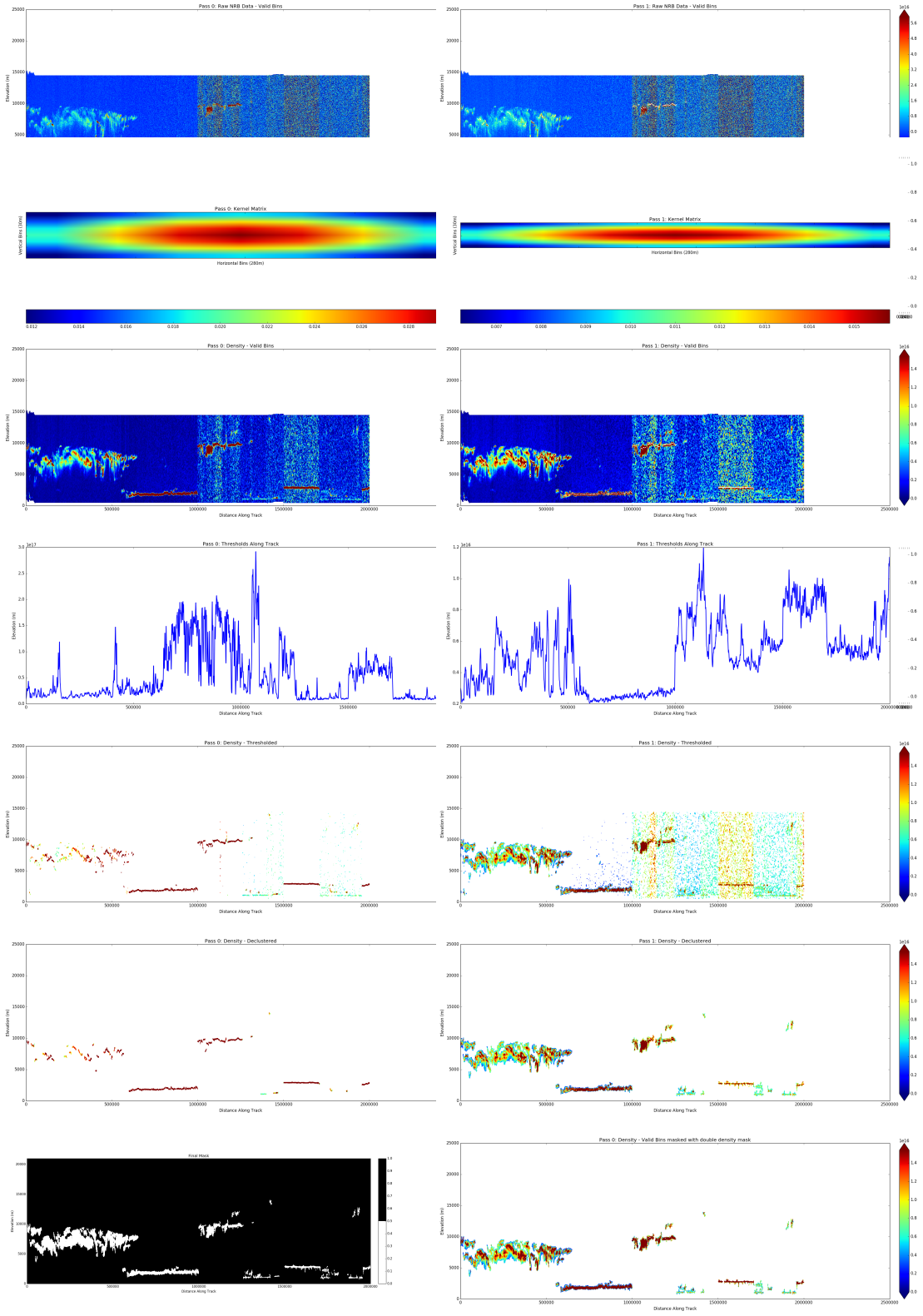


Figure 36-11. (t67) Analysis of GLAS-data based simulated ICESat-2 data using the DDA.

$\sigma = 3$	cutoff = 1	a_m = 10,20	min cluster size = 150, 600	base_threshold = 10E+14	downsampling = 1
threshold_sensitivity = 0.9, 1		threshold_segment_length = 2		quantile = 0.99, 0.8	

14 Analysis of First ICESat-2 ATLAS Data After Launch and Sensitivity Study for ATLAS Atmosphere Data

ICESat-2 was launched on September 15, 2018 from Vandenberg Airforce Base in California on the last Delta-2 Rocket of United Launch Alliance.

14.1 Experiment Setup, Data and Results for Parameter Implementation. t56.

Initial data sets created after launch are version 200. To optimize performance of the algorithm for post-launch data, a sensitivity study was conducted using several parameter combinations. The parameter sets used in this sensitivity study are **t(55) - (t72)**, which are listed in Table 7.

The sensitivity study was carried out on a subset of a composite of several cloud layer characteristics.

`ATL04_20181016022104_02670101_200_01.h5`

is the granule that was used to create

`sensi_subset_20181016022104`

The subset has 8219 profiles.

Result. Following a sensitivity study of the algorithm specific parameters conducted using several parameter combinations, **t(55) - (t72)**, we determined that the optimal parameter combination is $t56$. Notably, this is the same parameter combination that worked best for the last round of pre-launch-data, along with $t64$. Parameter values are given in the Table 7.

Comparison to v200 ATL09s and conclusion (2018-October-23). The sensitivity studies are run by using ATL04 data and the geomath version of the DDA-atmos, along with the Q/A algorithm (which at this time has not been implemented in the SIPS version of the code) to produce ATL09 data. Then those ATL09 results are compared to the matching ATL09 data files (v200), derived by SIPS/ASAS.

The only change required to match $t56$ was

300, 600 for size_threshold1, size_threshold2

The min cluster size is called size_threshold1, size_threshold2 (for density run1, run 2 respectively). In ATL09[rest].h5:

`/ancillary_data/atmosphere/size_threshold1`

Initially used values were 600, 600 . These values are reported on the .h5 file in ATL09. The values are used in code that creates ATL09s from ATL04s.

This parameter change has been made for v201.

Note. NRB value calculation will change in the near future (for v203 or v204) and may require a new sensitivity study.

14.2 Sensitivity Study and Post-Launch Q/A

Sensitivity studies are run for the parameter combinations **t(55) - (t72)** given in Table 7. The parameter sets are defined such that higher and lower values are tested individually for each of the final parameters. If a better combination is encountered in a sensitivity run, then lower and higher values around those last winning parameters are tested in a new sensitivity study. This process is repeated until no better values are found. In critical cases, intermediate parameters are tested as well. Note that Table 7 also includes parameter sets used for sensitivity studies that became necessary after changes to the input data in ATL04.

The following steps are illustrated for each parameter set shown in Figure 41 (by rows)

1. Raw NRB data (valid bins)[ATL04 input data to the DDA-atmos]; Cloud Layer Boundaries over Raw NRB Data
2. Pass 0: Kernel Matrix; Pass 1: Kernel Matrix
3. Pass 0: Density; Pass 1: Density
4. Pass 0: Thresholds Along Track; Pass 1: Thresholds Along Track
5. Pass 0: Density - Thresholded; Pass 1: Density - Thresholded
6. Pass 0: Density - Declustered (Mask 1); Pass 1: Density - Declustered (Mask 2)
7. Pass 0: Final Mask with Density (Combined Mask); QA: Half-Gap Confidence

Note that the code counts *pass 0* and *pass 1*; these are referred to as *density run 1* and *density run 2* in the text.

Parameter Set Name	Sigma	Anisotropy	Cutoff	Down-sample	Minimum Cluster Size	Threshold Bias	Threshold Factor	Threshold Window	Quantile
t55	3	10, 15	1	1	300, 600	10E+14	1, 0.9	2	0.99, 0.8
t56*	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.99, 0.8
t57*	3	10, 15	1	1	300, 600	10E+14	0.9, 1	2	0.99, 0.8
t58*	3	10, 20	1	1	400, 600	10E+14	0.9, 1	2	0.99, 0.8
t59*	3	10, 20	1	1	300, 600	12E+14	0.9, 1	2	0.99, 0.8
t60*	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.95, 0.8
t61	3	10, 15	1	1	300, 600	10E+14	0.9, 1	2	0.95, 0.8
t62	3	10, 15	1	1	300, 600	12E+14	0.9, 1	2	0.95, 0.8
t63*	3	10, 30	1	1	300, 600	10E+14	0.9, 1	2	0.99, 0.8
t64*	3	10, 20	1	1	200, 600	10E+14	0.9, 1	2	0.99, 0.8
t65	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.97, 0.8
t66	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.98, 0.8
t67	3	10, 20	1	1	150, 600	10E+14	0.9, 1	2	0.99, 0.8
t68*	3	10, 20	1	1	300, 600	9E+14	0.9, 1	2	0.99, 0.8
t69*	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.99, 0.7
t70*	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.99, 0.9
t71	3	10, 20	1	1	600, 600	10E+14	0.9, 1	2	0.99, 0.8
t72	3	10, 20	1	1	300, 600	11E+14	0.9, 1	2	0.99, 0.8
t73	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.99, 0.6
t74* †	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.99, 0.55
t75	3	10, 20	1	1	300, 1000	10E+14	0.9, 1	2	0.99, 0.55
t76	3	10, 20	1	1	300, 1500	10E+14	0.9, 1	2	0.99, 0.55
t77	3	10, 20	1	1	300, 2000	10E+14	0.9, 1	2	0.99, 0.55
t78	3	10, 20	1	1	300, 2000	10E+14	0.9, 1	2	0.99, 0.6
t80 †	3	10, 20	1	1	150, 600	10E+14	0.9, 1	2	0.99, 0.55
t81	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.99, 0.45
t82	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.99, 0.3
t83	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.99, 0.5
t84 †	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.99, 0.45
t85 †	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.95, 0.5
t86	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.95, 0.65
t87 †	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.97, 0.55
t88 †	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.97, 0.65
t89 †	3	10, 20	1	1	300, 600	10E+14	0.9, 1	2	0.97, 0.45

Table 7: **Parameter combinations used in sensitivity studies of the DDA-atmos applied to ATLAS data after launch (double density runs).** The table is for using double density runs ($num_passes = 2$). If two values are listed, then value 1 is for density-run1 and value 2 is for density-run1. If one value is listed, this value is used for both density-run1 and density-run2. (*) and (†) denote experiments for which results are shown in a figure.

(*)(1) Sensitivity study for first after-launch data analysis (except (t74)). Results in Figure 41. (t56) Parameter combination used in the official algorithm. Algorithm run on sub-sampled regions in the ATL04_20181016022104_02670101_200_01.h5 data file using returns from profile 2 of the ATLAS beam configuration.

(*)(2) Sensitivity study for after-launch data analysis, for ASAS code v5.0. Results in Figure 43. Algorithm run on sub-sampled regions in the ATL04_20181017T002107_02810101_950_01.h5 data file using returns from profile 2 of the ATLAS beam configuration.

(t69) Best compromise data set used for after-launch data analysis, ASAS atmos code version v5.0 and for the case that the same parameter set is used for day/night/dusk.

(3) Default values for 3 parameter sets are:

day: paramset1=t60, night: paramset2=t74; dusk: paramset3=t60 (see Figure 44). Algorithm run on sub-sampled regions in the ATL04_20181017T002107_02810101_950_01_dda3.h5 data file using returns from profile 3.

† **(4) Sensitivity study for first public release of atmospheric data (release date: May 2019), for ASAS atmos code v5.1. Results in Figure 46.** Algorithm run on sub-sampled regions in the ATL04_20181017T002107_02810101_951_01.h5 data file using returns from profile 3 of the ATLAS beam configuration. Parameter sets are: **day: paramset1=t60, night: paramset2=t87; dusk: paramset3=t60.**

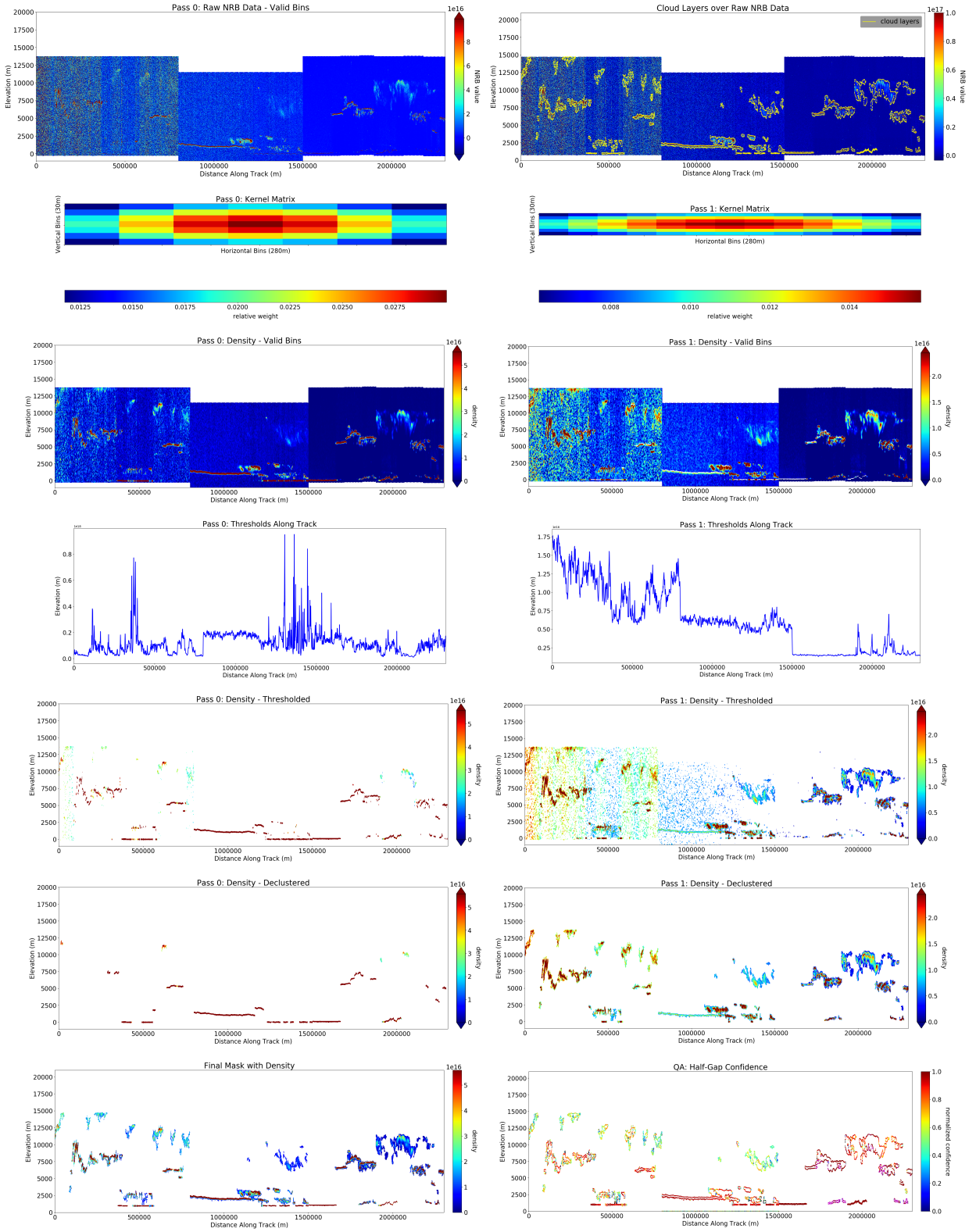


Figure 41-1. Sensitivity analysis of ICESat-2 ATLAS data post-launch: (t56) Parameter combination used in product. Algorithm run on sub-sampled regions in the ATL04_20181016022104_02670101_200_01.h5 data file using returns from profile 2 (prof2) of the ATLAS beam configuration.

σ = 3,3	cutoff = 1,1
a_m = 10,20	min cluster size = 300,600
threshold_bias = 10E+14,10E+14	downsampling = 1,1
threshold_sensitivity = 0.9,1	threshold_segment_length = 2,2
quantile = 0.99,0.8	

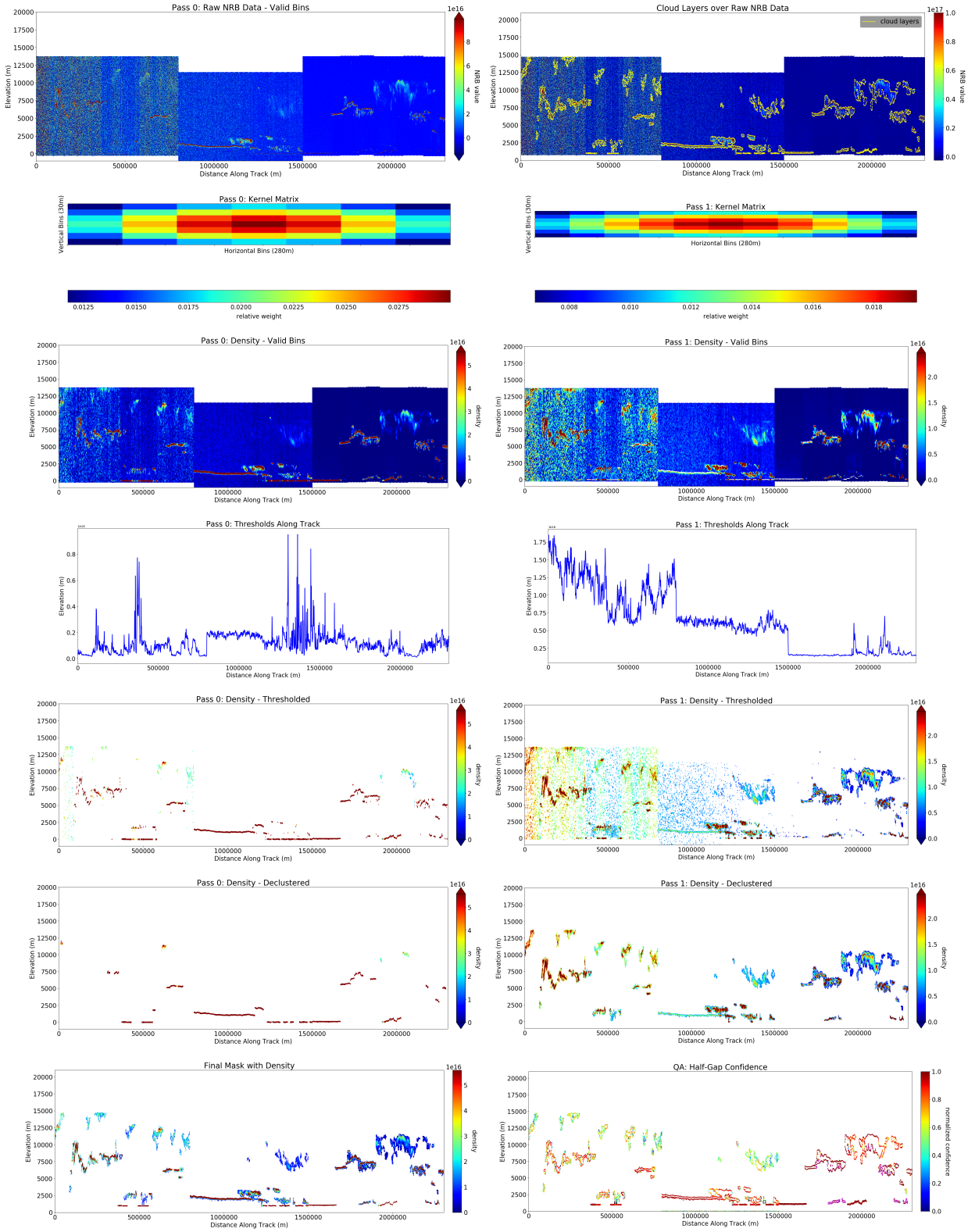


Figure 41-2. Sensitivity analysis of ICESat-2 ATLAS data post-launch: (t57) Smaller anisotropy in pass 2. Algorithm run on sub-sampled regions in the ATL04_20181016022104_02670101_200_01.h5 data file using returns from profile 2 (prof2) of the ATLAS beam configuration.

σ = 3,3	cutoff = 1,1
a_m = 10,15	min cluster size = 300,600
threshold_bias = 10E+14,10E+14	downsampling = 1,1
threshold_sensitivity = 0.9,1	threshold_segment_length = 2,2
quantile = 0.99,0.8	

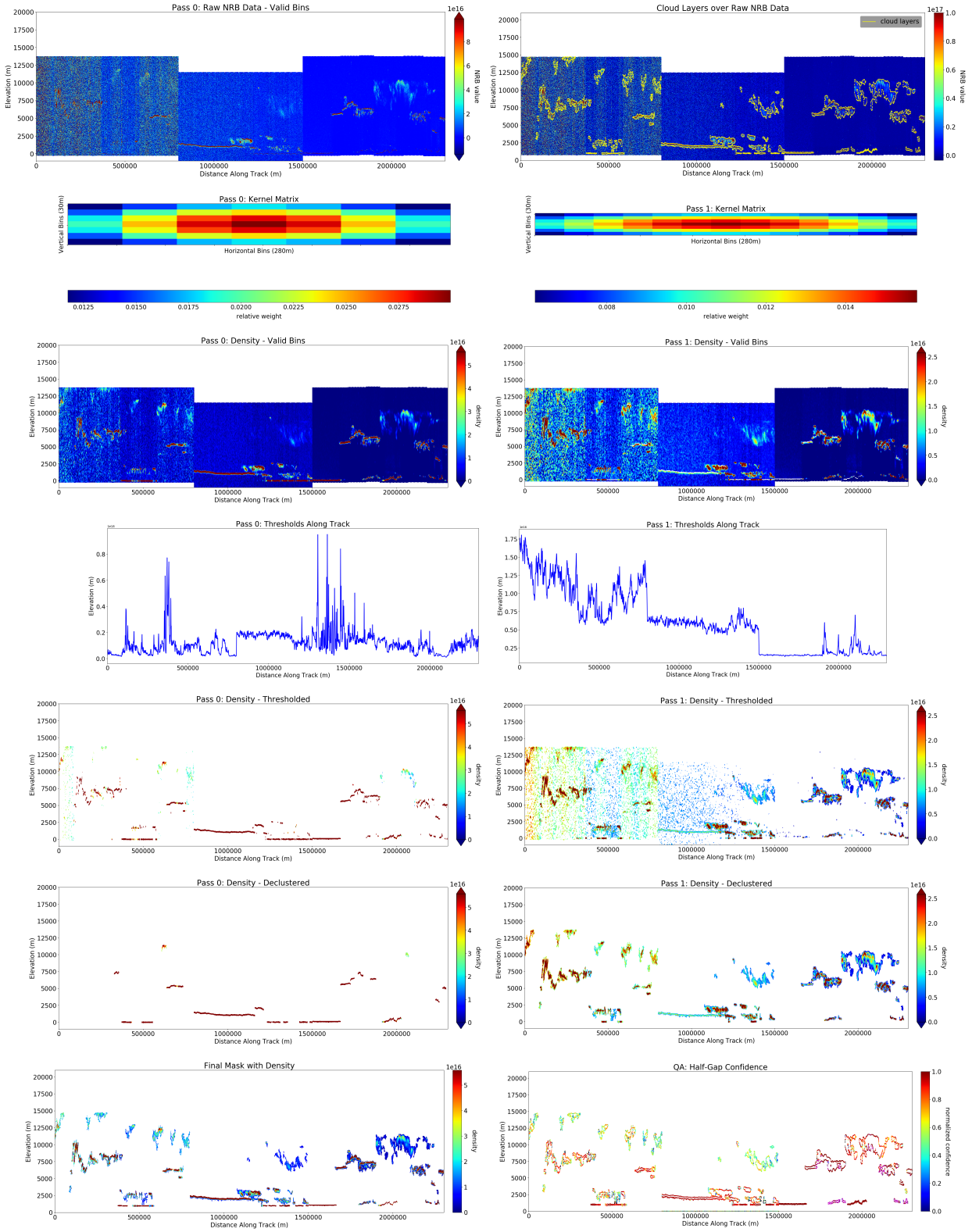


Figure 41-3. Sensitivity analysis of ICESat-2 ATLAS data post-launch: (t58) Larger min cluster size for pass 1. Algorithm run on sub-sampled regions in the ATL04_20181016022104_02670101_200_01.h5 data file using returns from profile 2 (prof2) of the ATLAS beam configuration.

σ = 3,3	cutoff = 1,1
a_m = 10,20	min cluster size = 400,600
threshold_bias = 10E+14,10E+14	downsampling = 1,1
threshold_sensitivity = 0.9,1	threshold_segment_length = 2,2
quantile = 0.99,0.8	

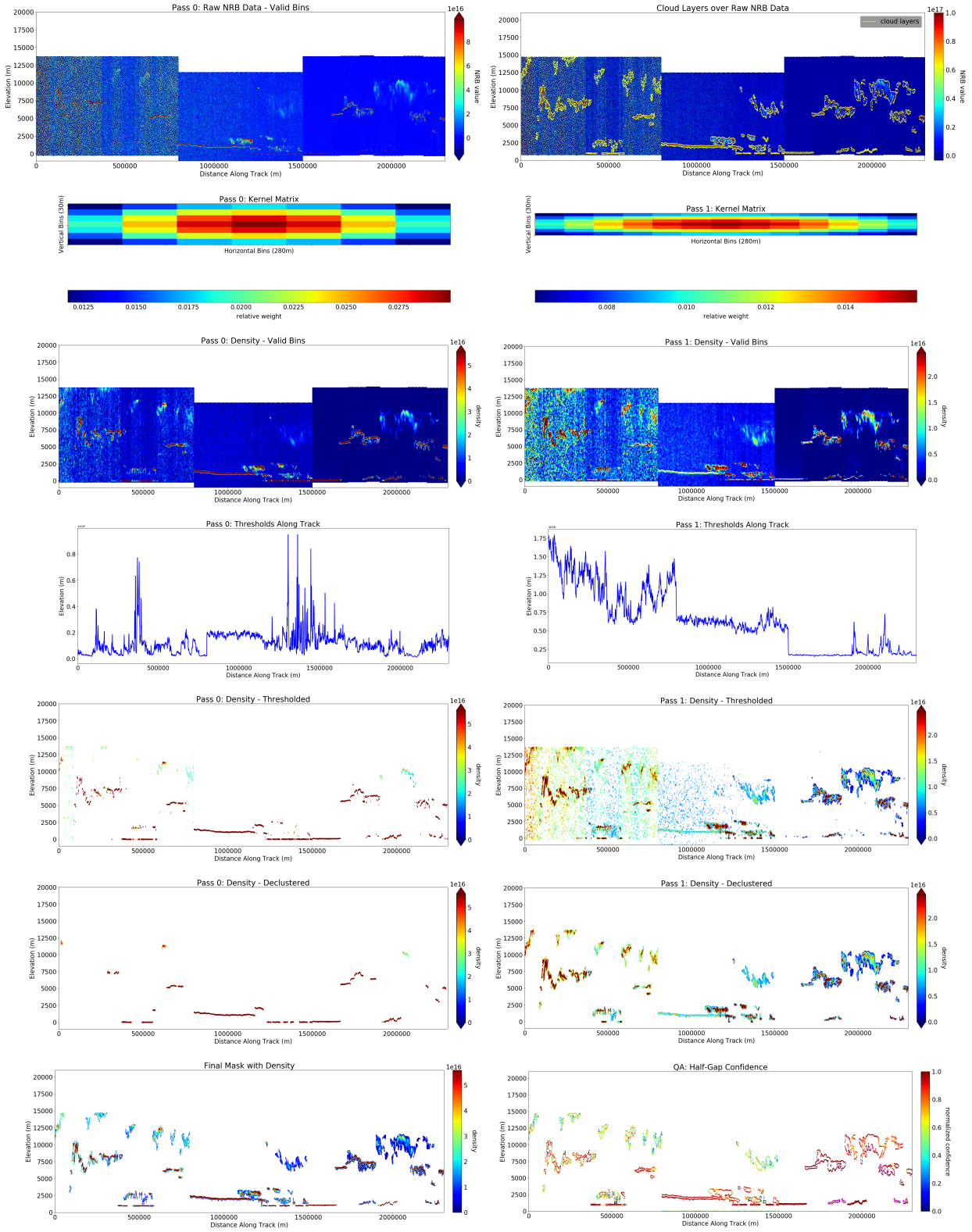


Figure 41-4. Sensitivity analysis of ICESat-2 ATLAS data post-launch: (t59) Larger threshold bias. Algorithm run on sub-sampled regions in the ATL04_20181016022104_02670101_200_01.h5 data file using returns from profile 2 (prof2) of the ATLAS beam configuration.

σ = 3,3	cutoff = 1,1
a_m = 10,20	min cluster size = 300,600
threshold_bias = 12E+14,12E+14	downsampling = 1,1
threshold_sensitivity = 0.9,1	threshold_segment_length = 2,2
quantile = 0.99,0.8	

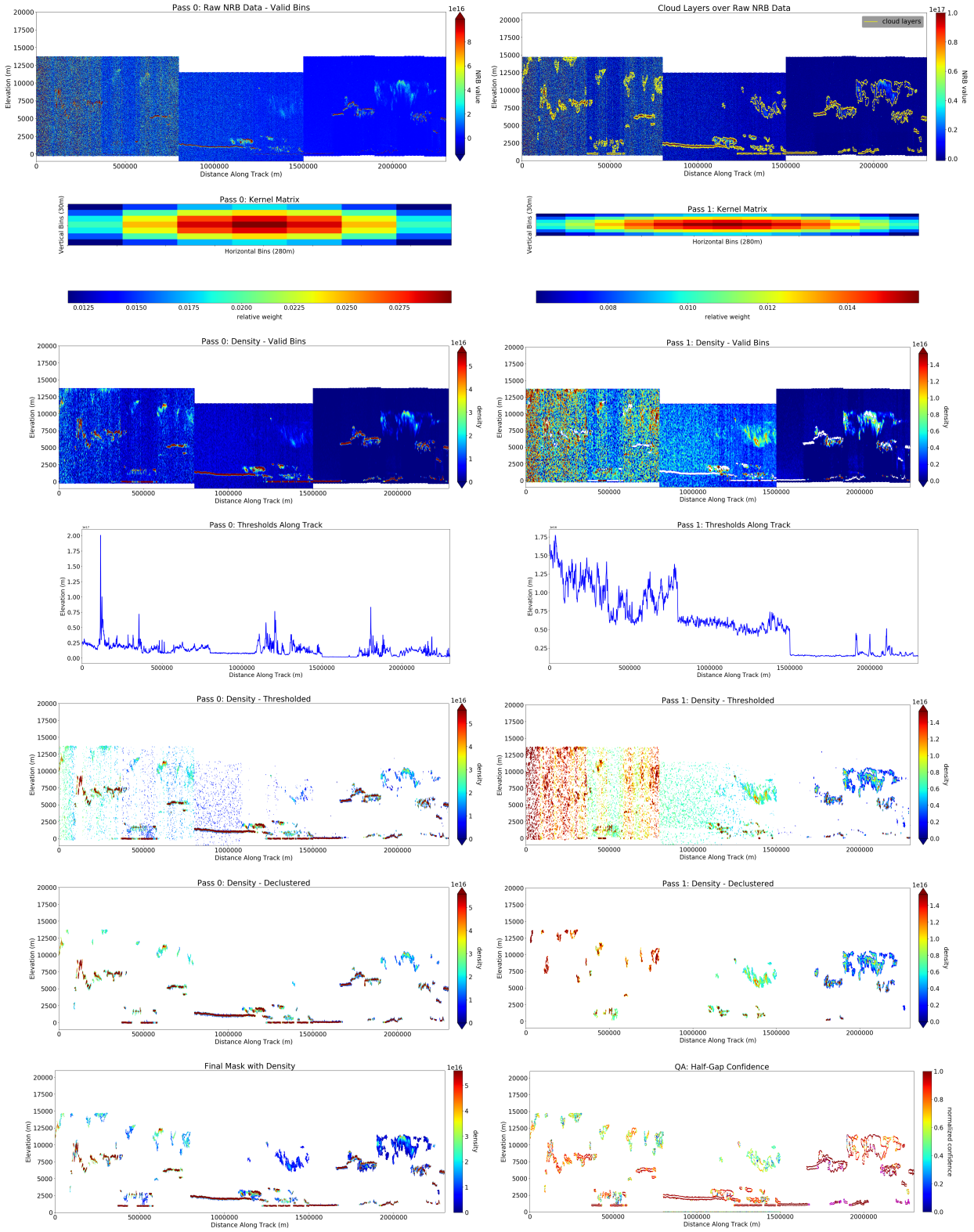


Figure 41-5. Sensitivity analysis of ICESat-2 ATLAS data post-launch: (t60) Smaller quantile for pass 1. Algorithm run on sub-sampled regions in the ATL04_20181016022104_02670101_200_01.h5 data file using returns from profile 2 (prof2) of the ATLAS beam configuration.

σ = 3,3	cutoff = 1,1
a_m = 10,20	min cluster size = 300,600
threshold_bias = 10E+14,10E+14	downsampling = 1,1
threshold_sensitivity = 0.9,1	threshold_segment_length = 2,2
quantile = 0.95,0.8	

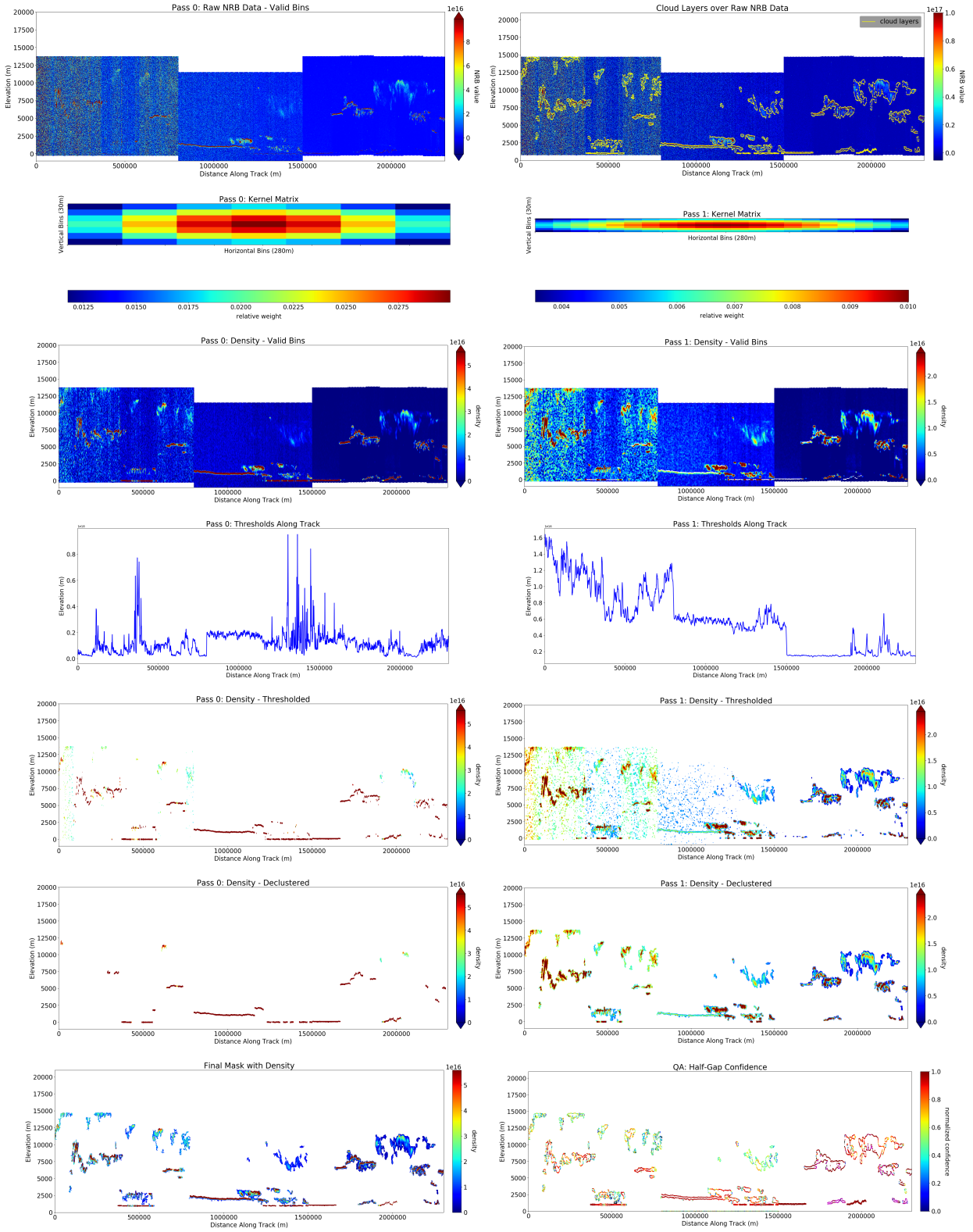


Figure 41-6. Sensitivity analysis of ICESat-2 ATLAS data post-launch: (t63) Larger anisotropy in pass 2. Algorithm run on sub-sampled regions in the ATL04_20181016022104_02670101_200_01.h5 data file using returns from profile 2 (prof2) of the ATLAS beam configuration.

σ = 3,3	cutoff = 1,1
a_m = 10,30	min cluster size = 300,600
threshold_bias = 10E+14,10E+14	downsampling = 1,1
threshold_sensitivity = 0.9,1	threshold_segment_length = 2,2
quantile = 0.99,0.8	

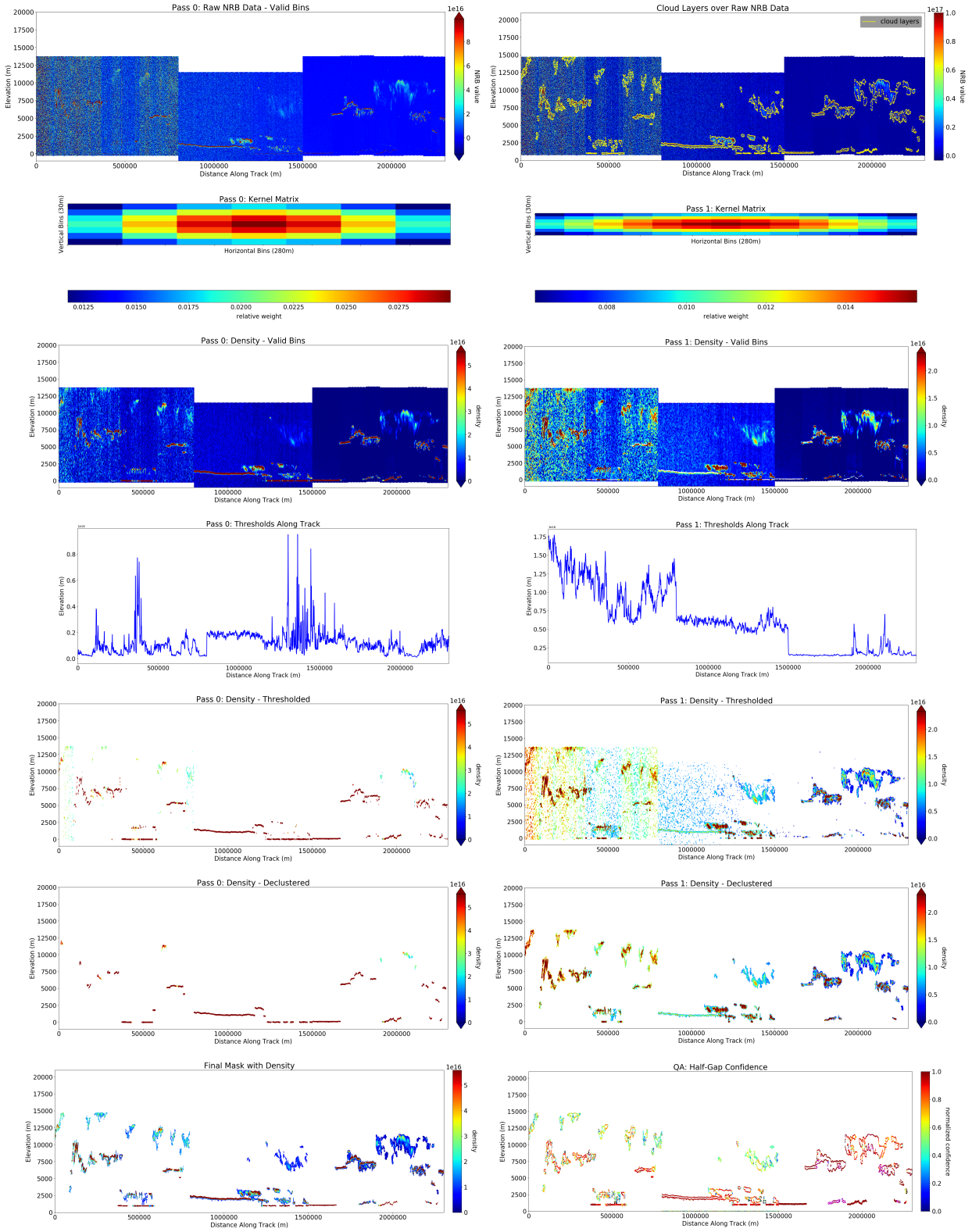


Figure 41-7. Sensitivity analysis of ICESat-2 ATLAS data post-launch: (t64) Smaller min cluster size for pass 1. Algorithm run on sub-sampled regions in the ATL04_20181016022104_02670101_200_01.h5 data file using returns from profile 2 (prof2) of the ATLAS beam configuration.

σ = 3,3	cutoff = 1,1
a_m = 10,20	min cluster size = 200,600
threshold_bias = 10E+14,10E+14	downsampling = 1,1
threshold_sensitivity = 0.9,1	threshold_segment_length = 2,2
quantile = 0.99,0.8	

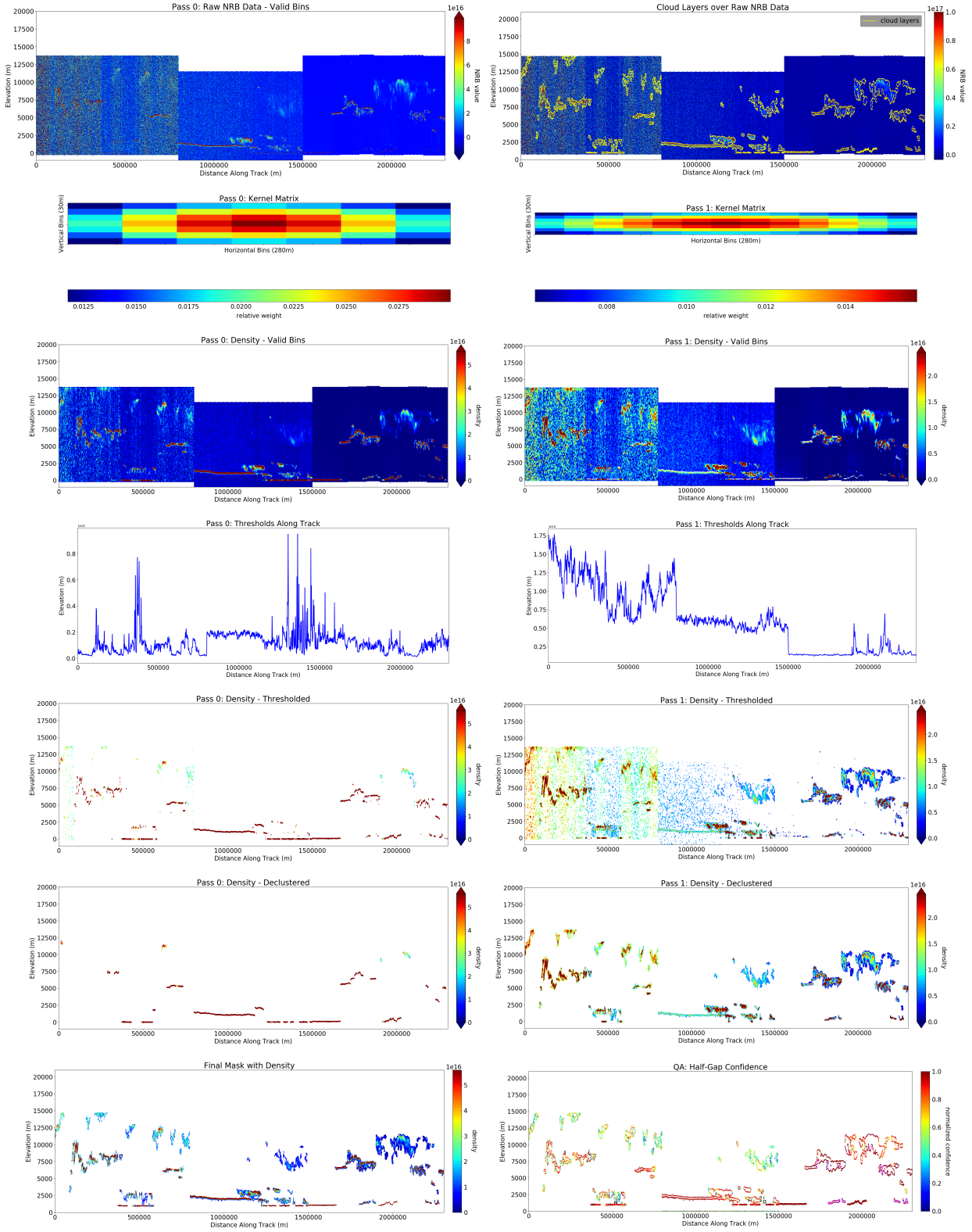


Figure 41-8. Sensitivity analysis of ICESat-2 ATLAS data post-launch: (t68) Small threshold bias. Algorithm run on sub-sampled regions in the ATL04_20181016022104_02670101_200_01.h5 data file using returns from profile 2 (prof2) of the ATLAS beam configuration.

σ = 3,3	cutoff = 1,1
a_m = 10,20	min cluster size = 300,600
threshold_bias = 9E+14,9E+14	downsampling = 1,1
threshold_sensitivity = 0.9,1	threshold_segment_length = 2,2
quantile = 0.99,0.8	

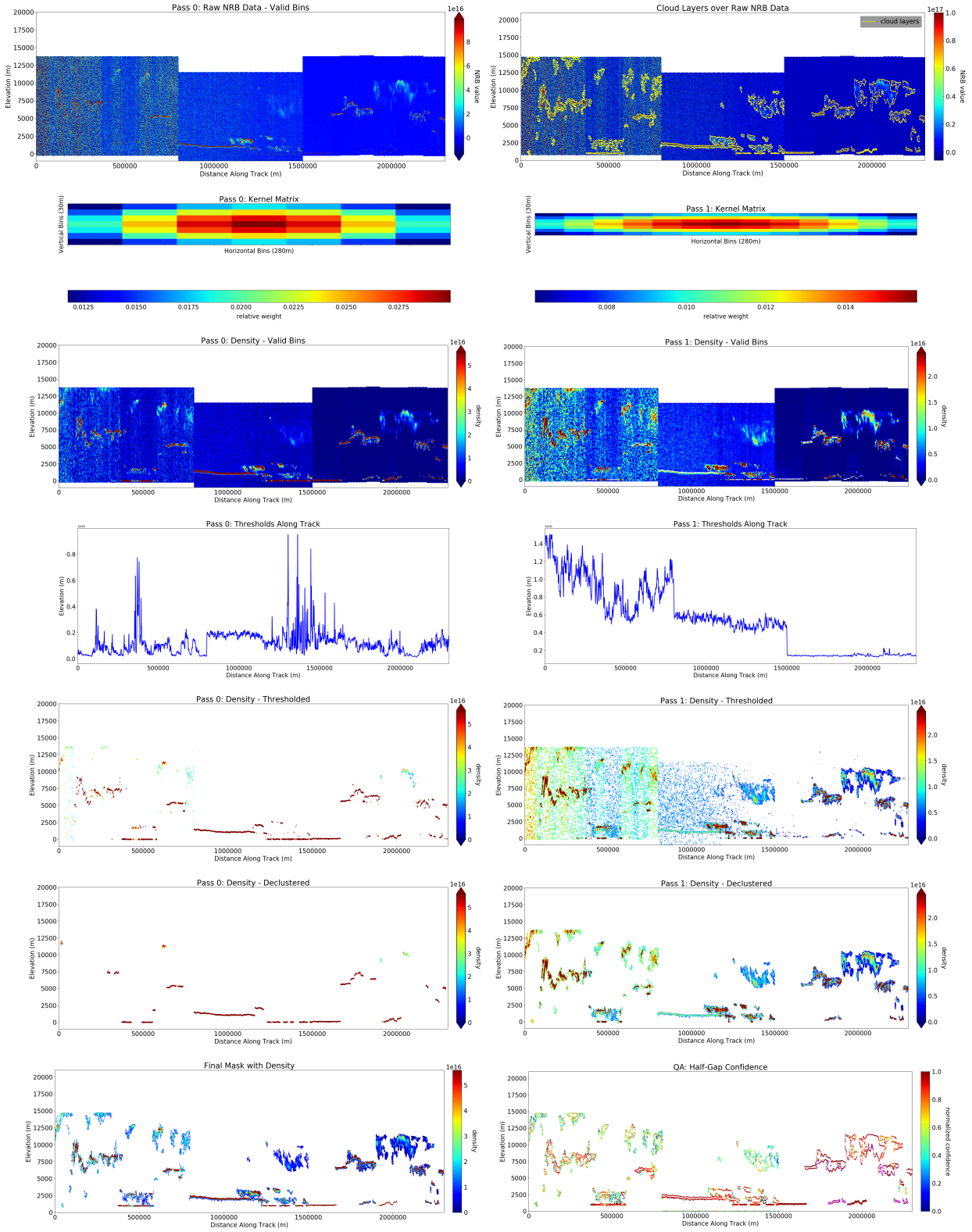


Figure 41-9. Sensitivity analysis of ICESat-2 ATLAS data post-launch: (t69) Small quantile for pass 2. Algorithm run on sub-sampled regions in the ATL04_20181016022104_02670101_200_01.h5 data file using returns from profile 2 (prof2) of the ATLAS beam configuration.

σ = 3,3	cutoff = 1,1
a_m = 10,20	min cluster size = 300,600
threshold_bias = 10E+14,10E+14	downsampling = 1,1
threshold_sensitivity = 0.7,1	threshold_segment_length = 2,2
quantile = 0.99,0.8	

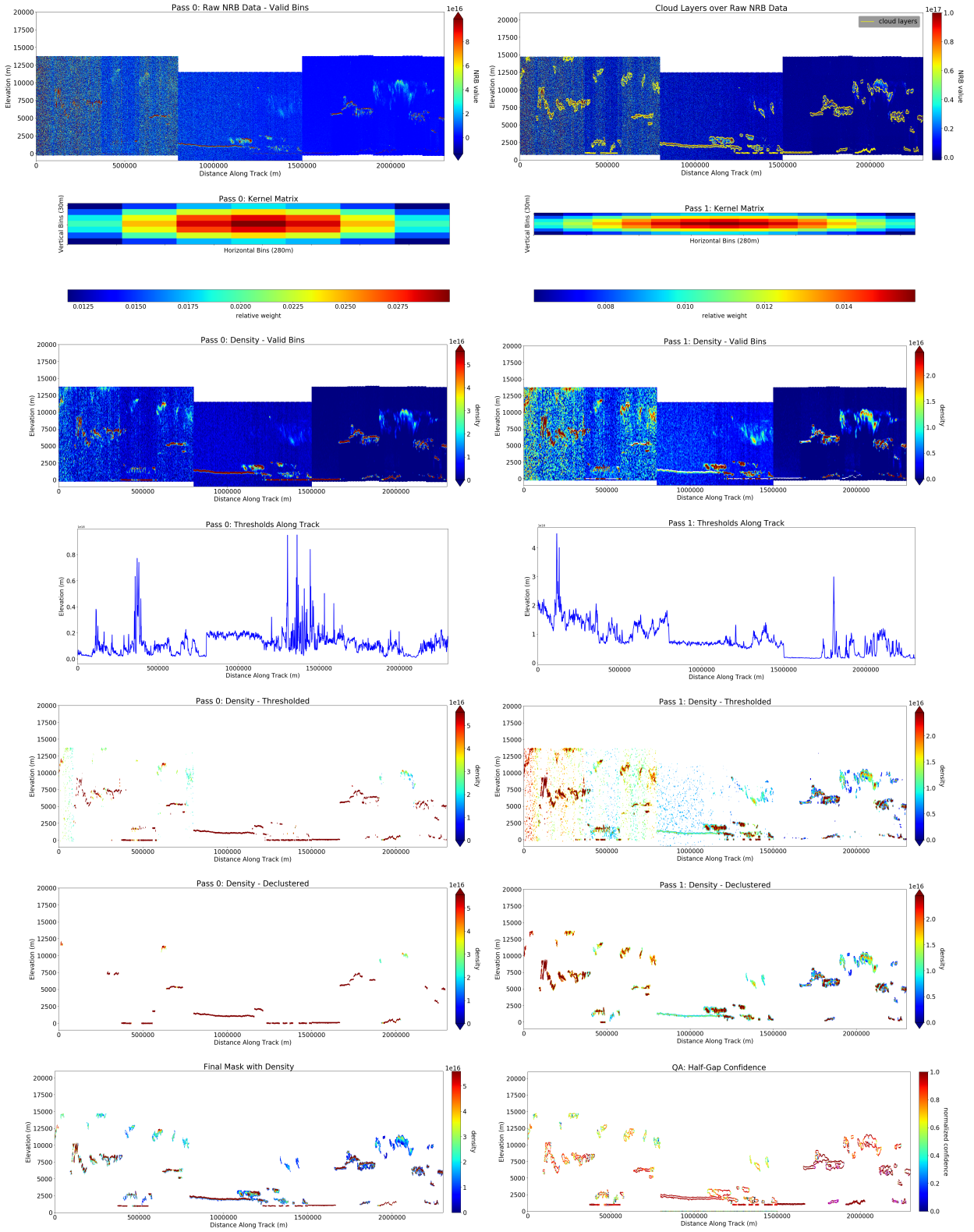


Figure 41-10. Sensitivity analysis of ICESat-2 ATLAS data post-launch: (t70) Larger quantile for second pass. Algorithm run on sub-sampled regions in the ATL04_20181016022104_02670101_200_01.h5 data file using returns from profile 2 (prof2) of the ATLAS beam configuration.

σ = 3,3	cutoff = 1,1
a_m = 10,20	min cluster size = 300,600
threshold_bias = 10E+14,10E+14	downsampling = 1,1
threshold_sensitivity = 0.9,1	threshold_segment_length = 2,2
quantile = 0.99,0.9	

16 Sensitivity Study for Pre-Release Data Version v950, Necessitated by Change in Background and NRB Calculation in ATL04

Creation of a good data product requires improvements of the algorithms and updates of the code used for analysis, which is a process that initially starts from the lower-level products and continues through the higher-level products. Changes of the calculations of background in the atmospheric data in ATL04 have led to changes in calculation of the normalized radiometric backscatter (NRB) values, also in ATL04, on which the detection of atmospheric density, layers and related results from the DDA-atmos is based.

Following such a change in background and NRB data in ATL04 in ASAS code version v5.0 (data labeled pre-release test version v950), a new sensitivity study was run to optimize DDA-parameter sets for the updated ATL04 data. Studies are run for the parameter sets **t(55) - (t74)** given in Table 7, rerunning **(t55) - (t72)** and adding two new parameter sets, **(t73), (t74)**.

The DDA-atmos was run on sub-sampled regions in the data file
ATL04_20181017T002107_02810101_950_01.h5
using returns from profile 2 of the ATLAS beam configuration. Note that the atmospheric data products utilize only the three strong beams and the relationship of beam numbers to profile numbers changes whenever the satellite's measuring array is flipped. Specific information is found in the data products. Beam 2 is always associated with profile 2, but numbers 1 and 3 flip.

Result: (t69) is the best compromise data set used for after-launch data analysis, ASAS atmos code version v5.0 and for the case that the same parameter set is used for day/night/twilight.

The results are reported here, because they are used in following sections to inform later optimizations of parameter sets. The following steps are illustrated for each parameter set shown in Figure 43 (by rows):

1. Raw NRB data (valid bins)[ATL04 input data to the DDA-atmos]; Cloud Layer Boundaries over Raw NRB Data

2. Pass 0: Kernel Matrix; Pass 1: Kernel Matrix

3. Pass 0: Density; Pass 1: Density

4. Pass 0: Thresholds Along Track; Pass 1: Thresholds Along Track

5. Pass 0: Density - Thresholded; Pass 1: Density - Thresholded

6. Pass 0: Density - Declustered (Mask 1); Pass 1: Density - Declustered (Mask 2)

7. Pass 0: Final Mask with Density (Combined Mask); QA: Half-Gap Confidence

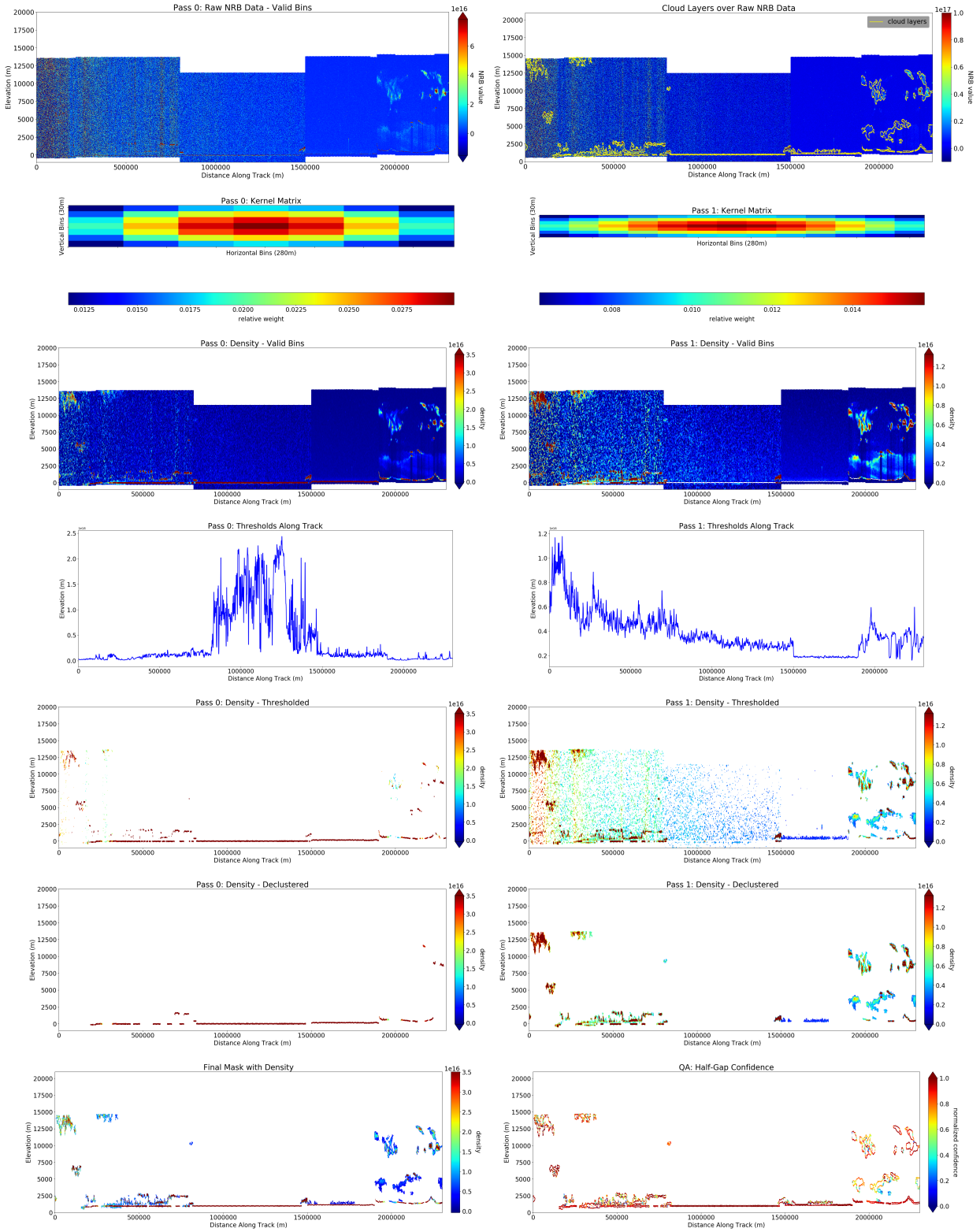


Figure 43-1. Sensitivity analysis of ICESat-2 ATLAS data post-launch: (t56) Parameter combination used in product. Algorithm run on sub-sampled regions in the ATL04_20181017T002107_02810101_950_01.h5 data file using returns from profile 2 of the ATLAS beam configuration.

σ = 3,3	cutoff = 1,1
a_m = 10,20	min cluster size = 300,600
threshold_bias = 10E+14,10E+14	downsampling = 1,1
threshold_sensitivity = 0.9,1	threshold_segment_length = 2,2
quantile = 0.99,0.8	

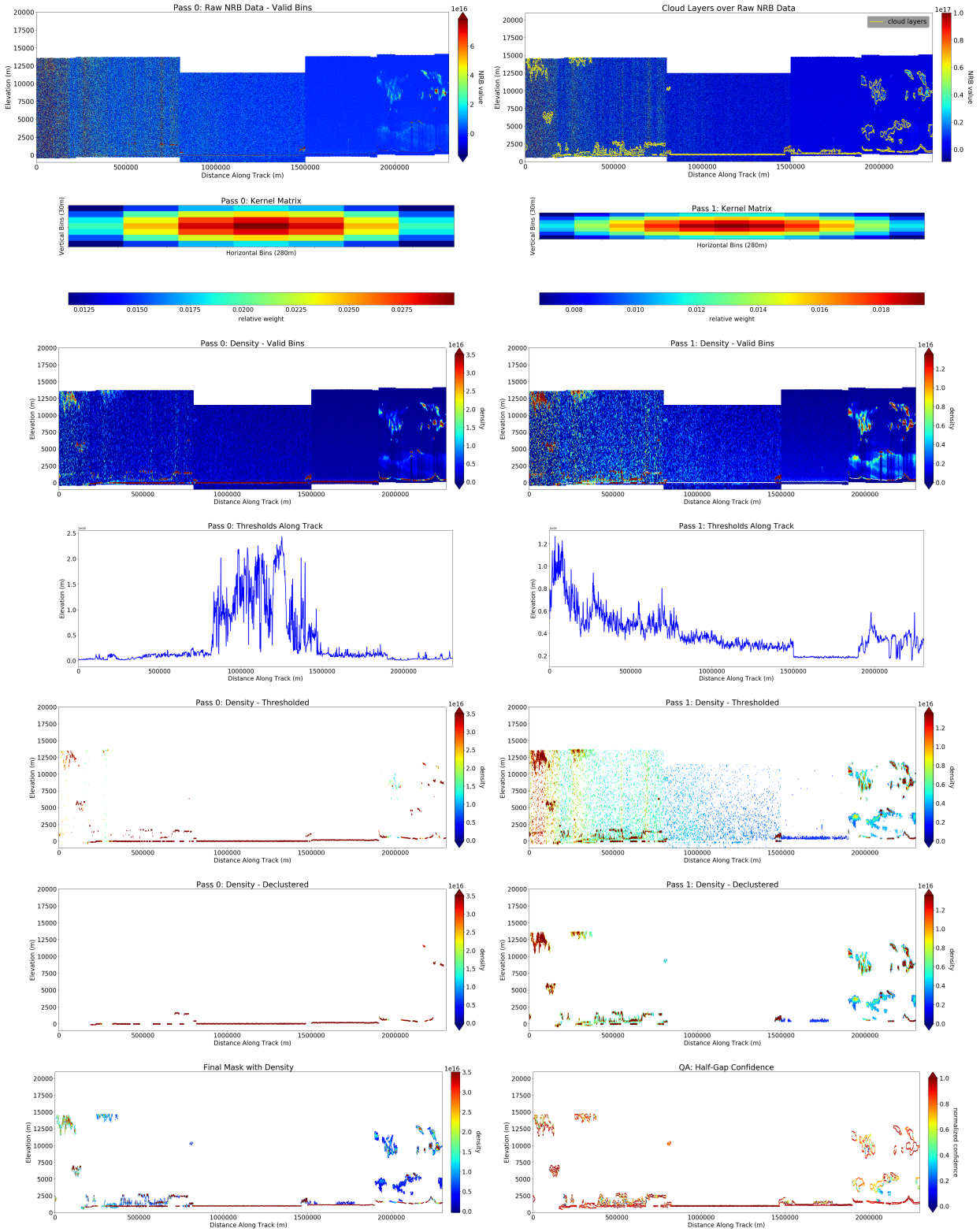


Figure 43-2. Sensitivity analysis of ICESat-2 ATLAS data post-launch: (t57) Smaller anisotropy in pass 2. Algorithm run on sub-sampled regions in the ATL04_20181017T002107_02810101_950_01.h5 data file using returns from profile 2 of the ATLAS beam configuration.

σ = 3,3	cutoff = 1,1
a_m = 10,15	min cluster size = 300,600
threshold_bias = 10E+14,10E+14	downsampling = 1,1
threshold_sensitivity = 0.9,1	threshold_segment_length = 2,2
quantile = 0.99,0.8	

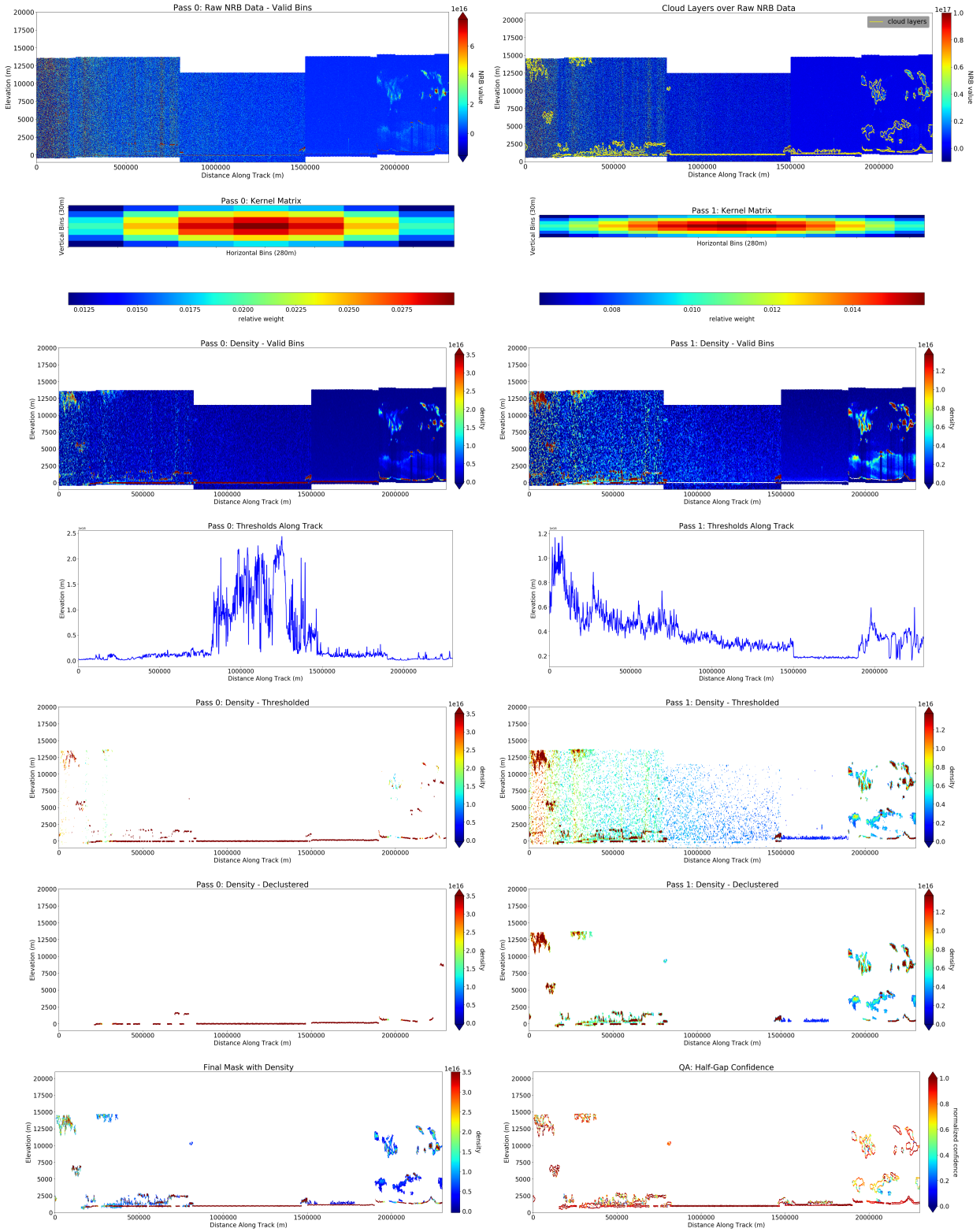


Figure 43-3. Sensitivity analysis of ICESat-2 ATLAS data post-launch: (t58) Larger min cluster size for pass 1. Algorithm run on sub-sampled regions in the ATL04_20181017T002107_02810101_950_01.h5 data file using returns from profile 2 of the ATLAS beam configuration.

σ = 3,3	cutoff = 1,1
a_m = 10,20	min cluster size = 400,600
threshold_bias = 10E+14,10E+14	downsampling = 1,1
threshold_sensitivity = 0.9,1	threshold_segment_length = 2,2
quantile = 0.99,0.8	

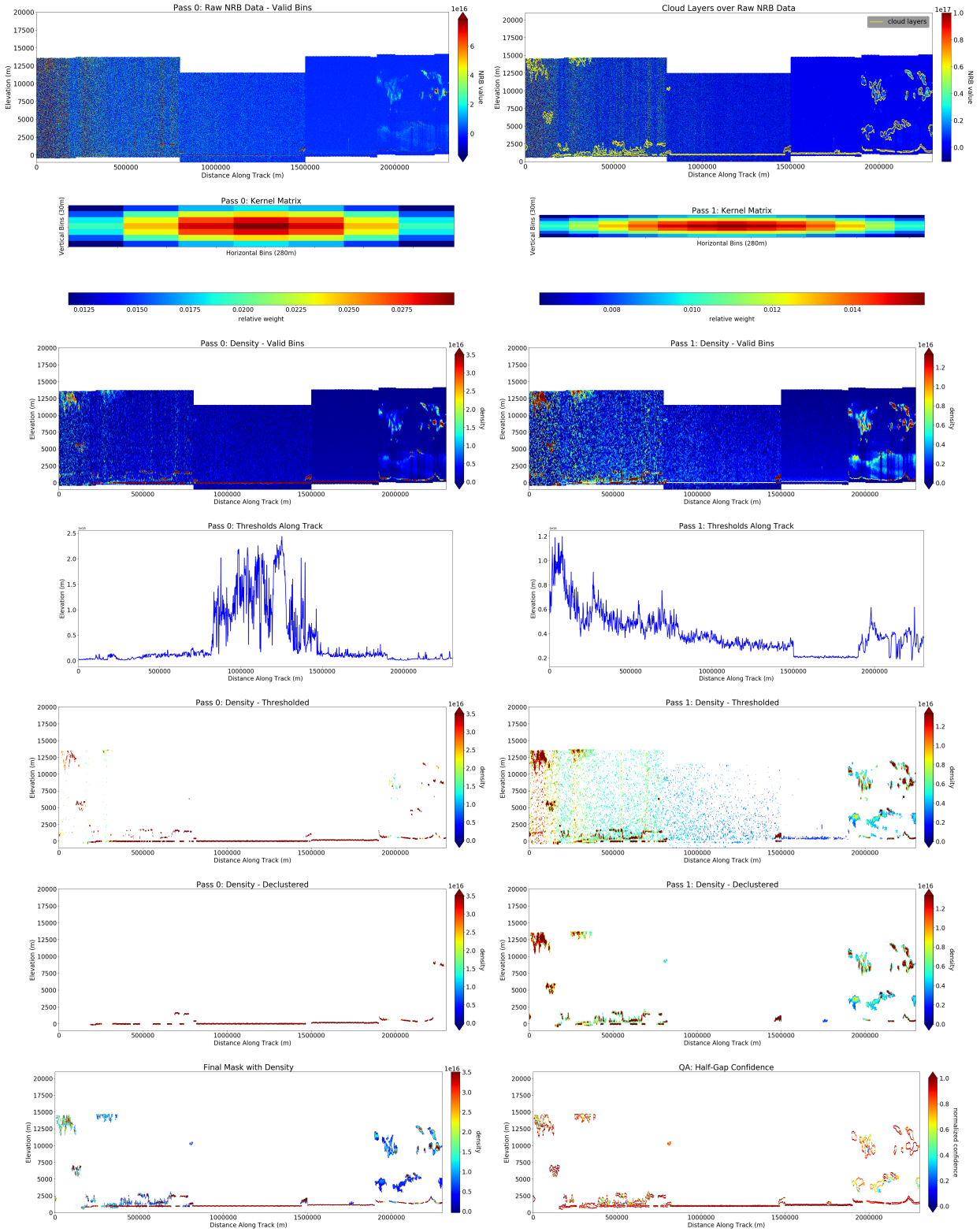


Figure 43-4. Sensitivity analysis of ICESat-2 ATLAS data post-launch: (t59) Larger threshold bias. Algorithm run on sub-sampled regions in the ATL04_20181017T002107_02810101_950_01.h5 data file using returns from profile 2 of the ATLAS beam configuration.

σ = 3,3	cutoff = 1,1
a_m = 10,20	min cluster size = 300,600
threshold_bias = 12E+14,12E+14	downsampling = 1,1
threshold_sensitivity = 0.9,1	threshold_segment_length = 2,2
quantile = 0.99,0.8	

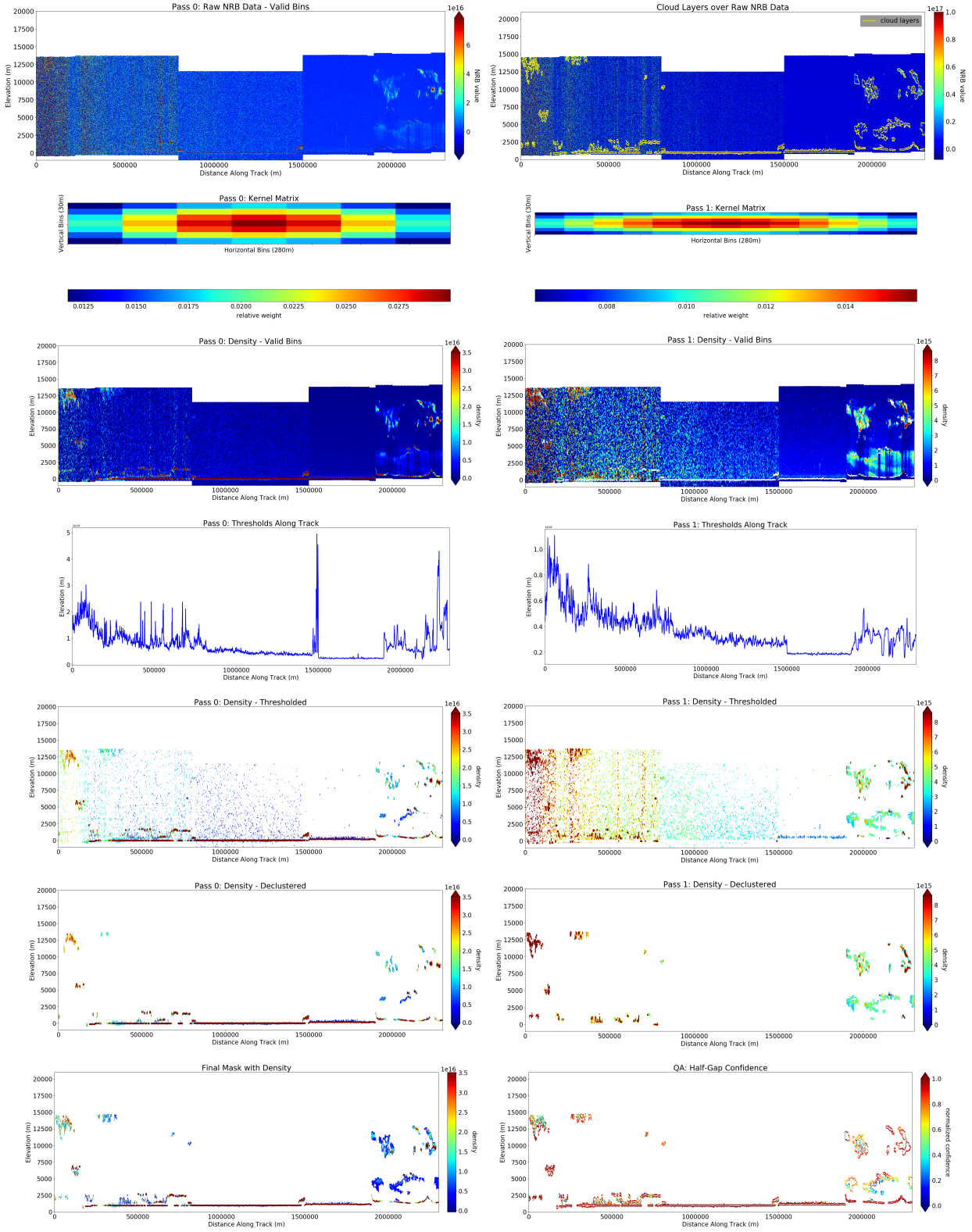


Figure 43-5. Sensitivity analysis of ICESat-2 ATLAS data post-launch: (t60) Smaller quantile for pass 1. Algorithm run on sub-sampled regions in the ATL04_20181017T002107_02810101_950_01.h5 data file using returns from profile 2 of the ATLAS beam configuration.

σ = 3,3	cutoff = 1,1
a_m = 10,20	min cluster size = 300,600
threshold_bias = 10E+14,10E+14	downsampling = 1,1
threshold_sensitivity = 0.9,1	threshold_segment_length = 2,2
quantile = 0.95,0.8	

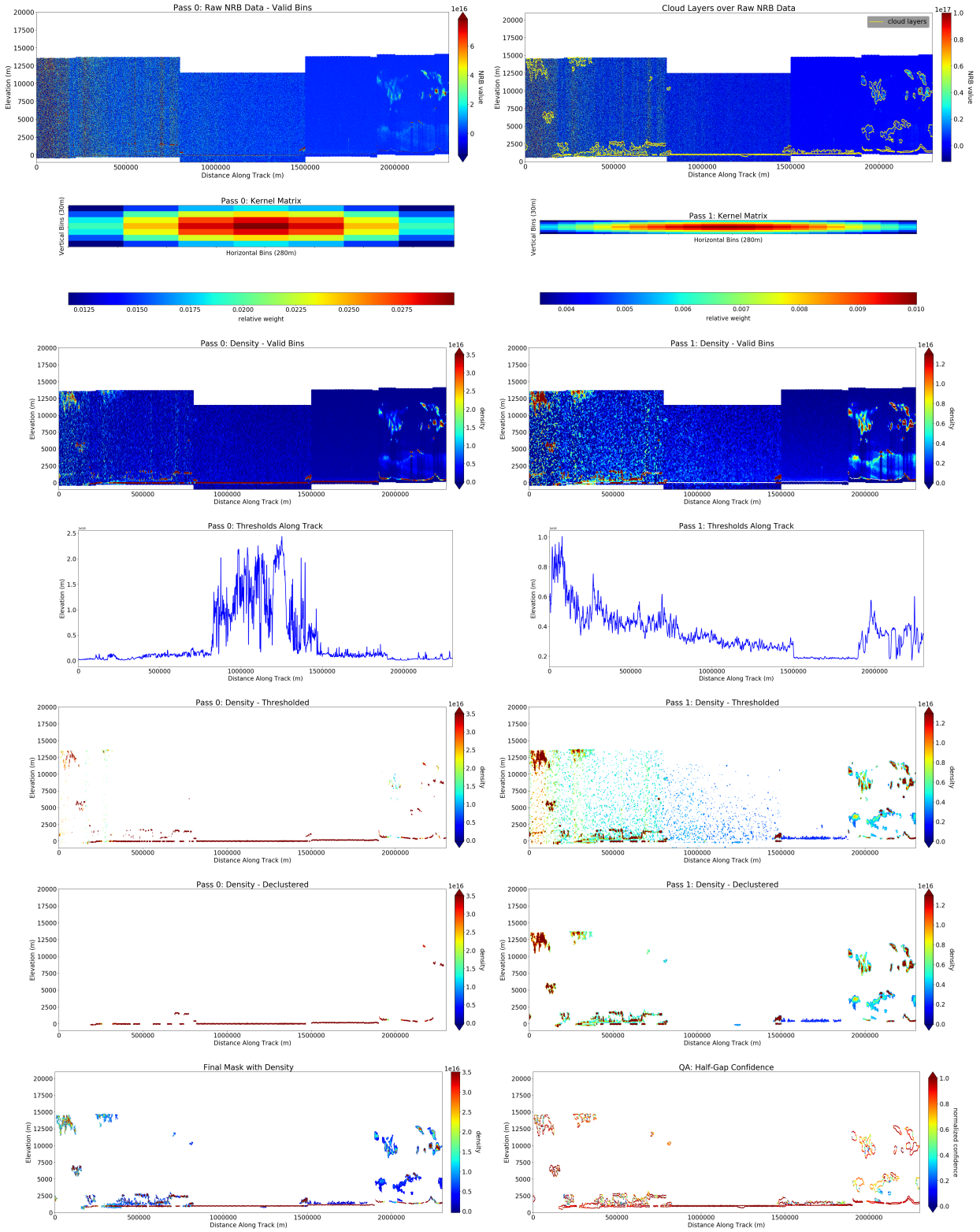


Figure 43-6. Sensitivity analysis of ICESat-2 ATLAS data post-launch: (t63) Larger anisotropy in pass 2. Algorithm run on sub-sampled regions in the ATL04_20181017T002107_02810101_950_01.h5 data file using returns from profile 2 of the ATLAS beam configuration.

σ = 3,3	cutoff = 1,1
a_m = 10,30	min cluster size = 300,600
threshold_bias = 10E+14,10E+14	downsampling = 1,1
threshold_sensitivity = 0.9,1	threshold_segment_length = 2,2
quantile = 0.99,0.8	

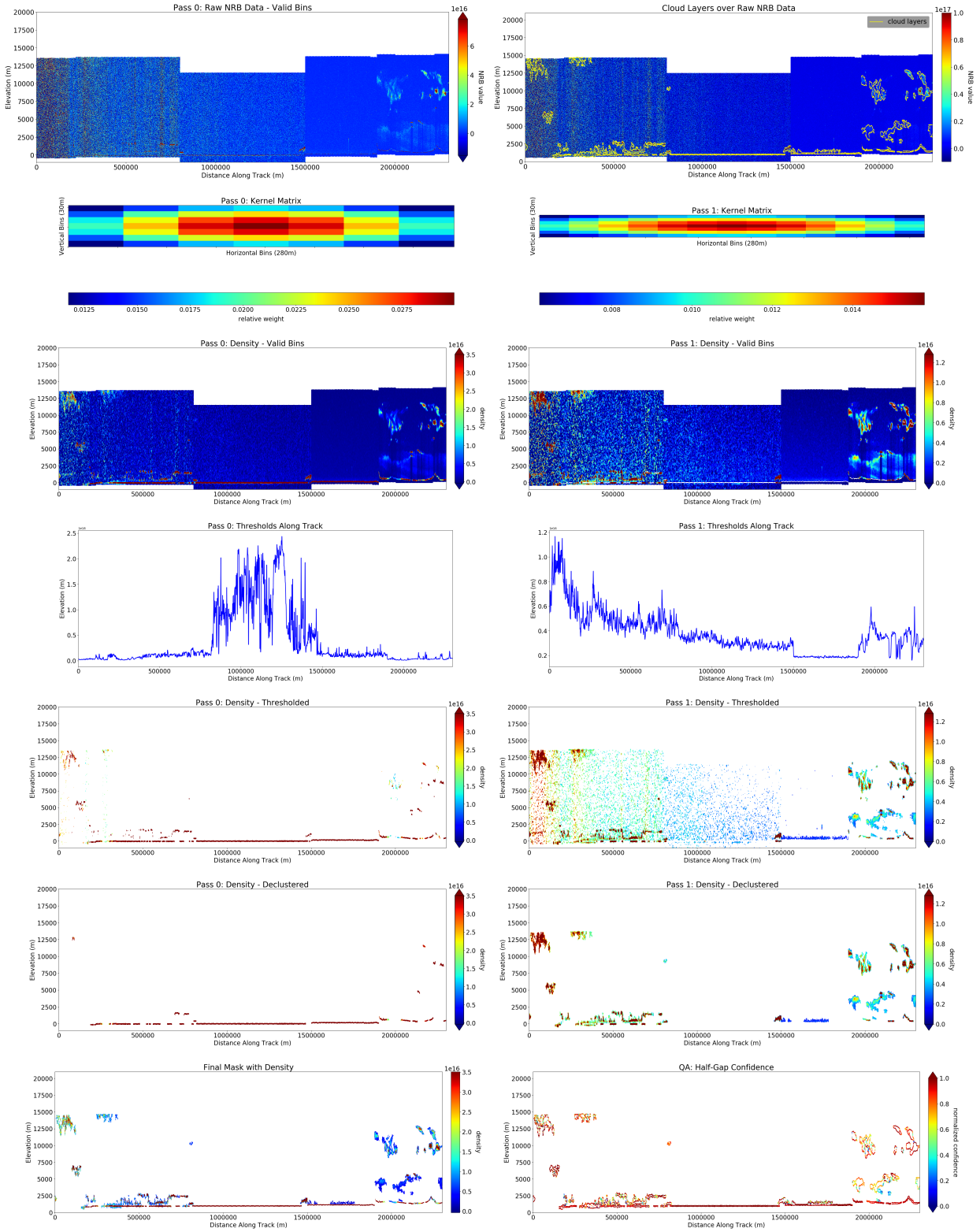


Figure 43-7. Sensitivity analysis of ICESat-2 ATLAS data post-launch: (t64) Smaller min cluster size for pass 1. Algorithm run on sub-sampled regions in the ATL04_20181017T002107_02810101_950_01.h5 data file using returns from profile 2 of the ATLAS beam configuration.

σ = 3,3	cutoff = 1,1
a_m = 10,20	min cluster size = 200,600
threshold_bias = 10E+14,10E+14	downsampling = 1,1
threshold_sensitivity = 0.9,1	threshold_segment_length = 2,2
quantile = 0.99,0.8	

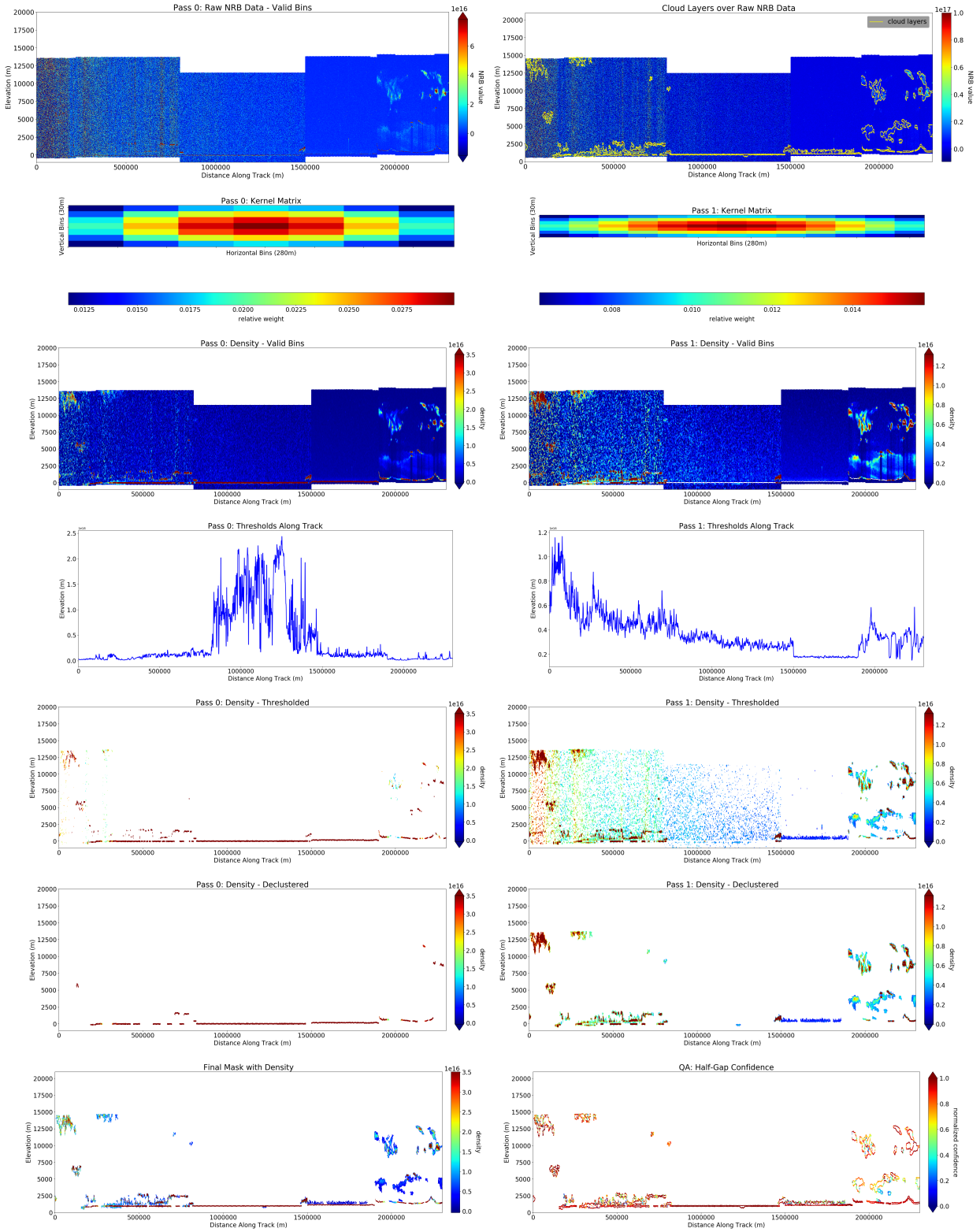


Figure 43-8. Sensitivity analysis of ICESat-2 ATLAS data post-launch: (t68) Smaller threshold bias. Algorithm run on sub-sampled regions in the ATL04_20181017T002107_02810101_950_01.h5 data file using returns from profile 2 of the ATLAS beam configuration.

σ = 3,3	cutoff = 1,1
a_m = 10,20	min cluster size = 300,600
threshold_bias = 9E+14,9E+14	downsampling = 1,1
threshold_sensitivity = 0.9,1	threshold_segment_length = 2,2
quantile = 0.99,0.8	

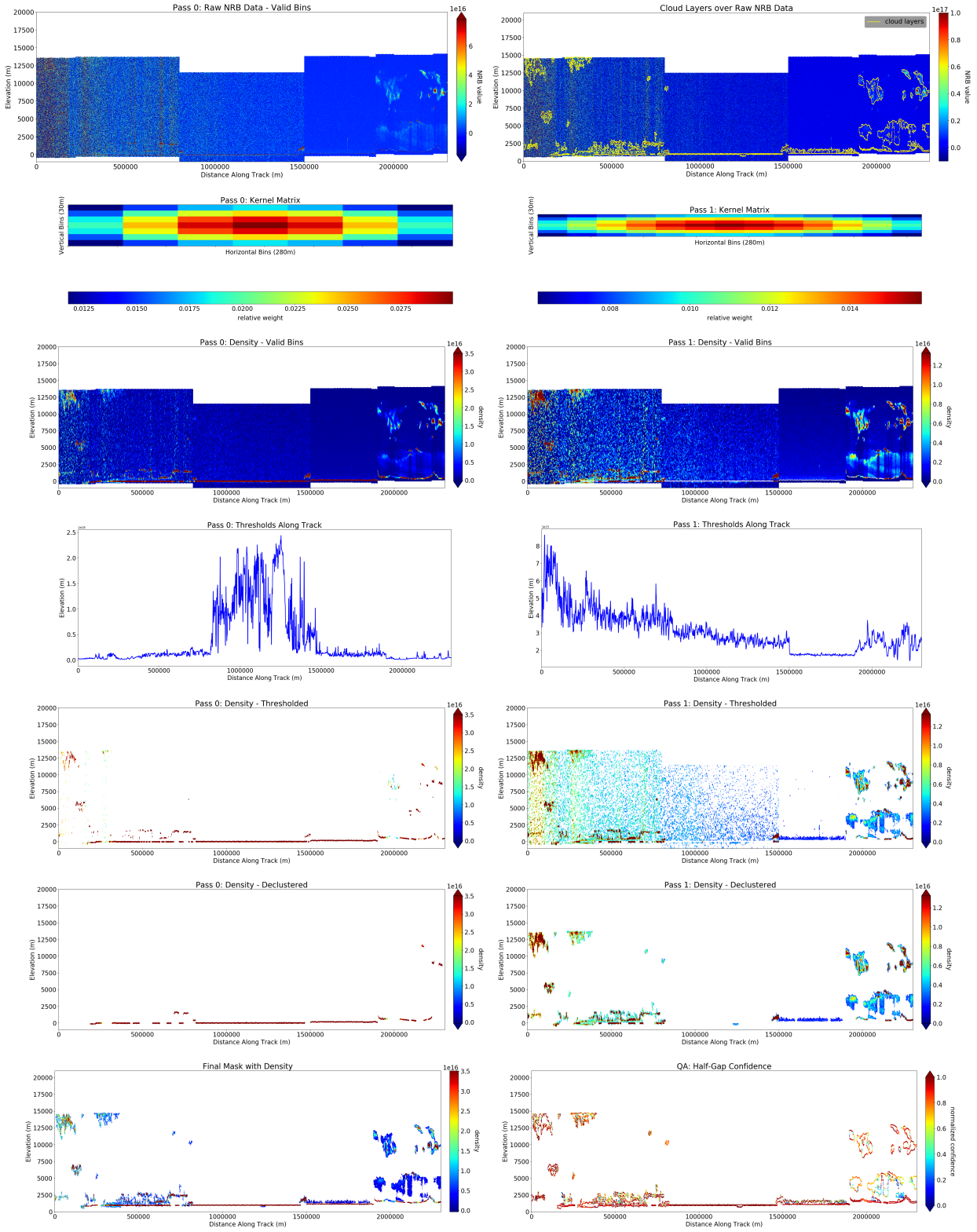


Figure 43-9. Sensitivity analysis of ICESat-2 ATLAS data post-launch: (t69) Smaller quantile for pass 2. Algorithm run on sub-sampled regions in the ATL04_20181017T002107_02810101_950_01.h5 data file using returns from profile 2 of the ATLAS beam configuration.

σ = 3,3	cutoff = 1,1
a_m = 10,20	min cluster size = 300,600
threshold_bias = 10E+14,10E+14	downsampling = 1,1
threshold_sensitivity = 0.7,1	threshold_segment_length = 2,2
quantile = 0.99,0.7	

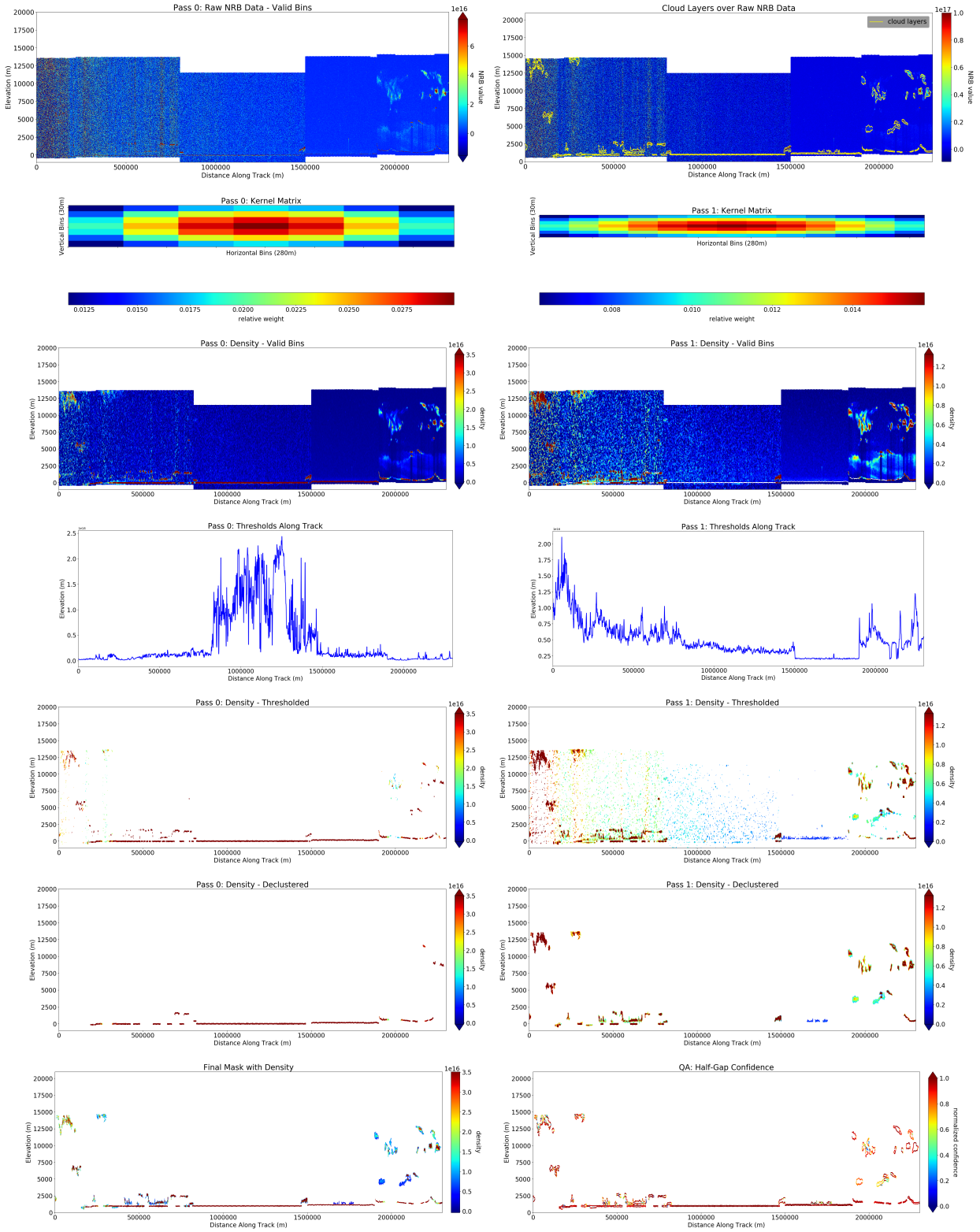


Figure 43-10. Sensitivity analysis of ICESat-2 ATLAS data post-launch: (t70) Larger quantile for second pass. Algorithm run on sub-sampled regions in the ATL04_20181017T002107_02810101_950_01.h5 data file using returns from profile 2 of the ATLAS beam configuration.

σ = 3,3	cutoff = 1,1
a_m = 10,20	min cluster size = 300,600
threshold_bias = 10E+14,10E+14	downsampling = 1,1
threshold_sensitivity = 0.9,1	threshold_segment_length = 2,2
quantile = 0.99,0.9	

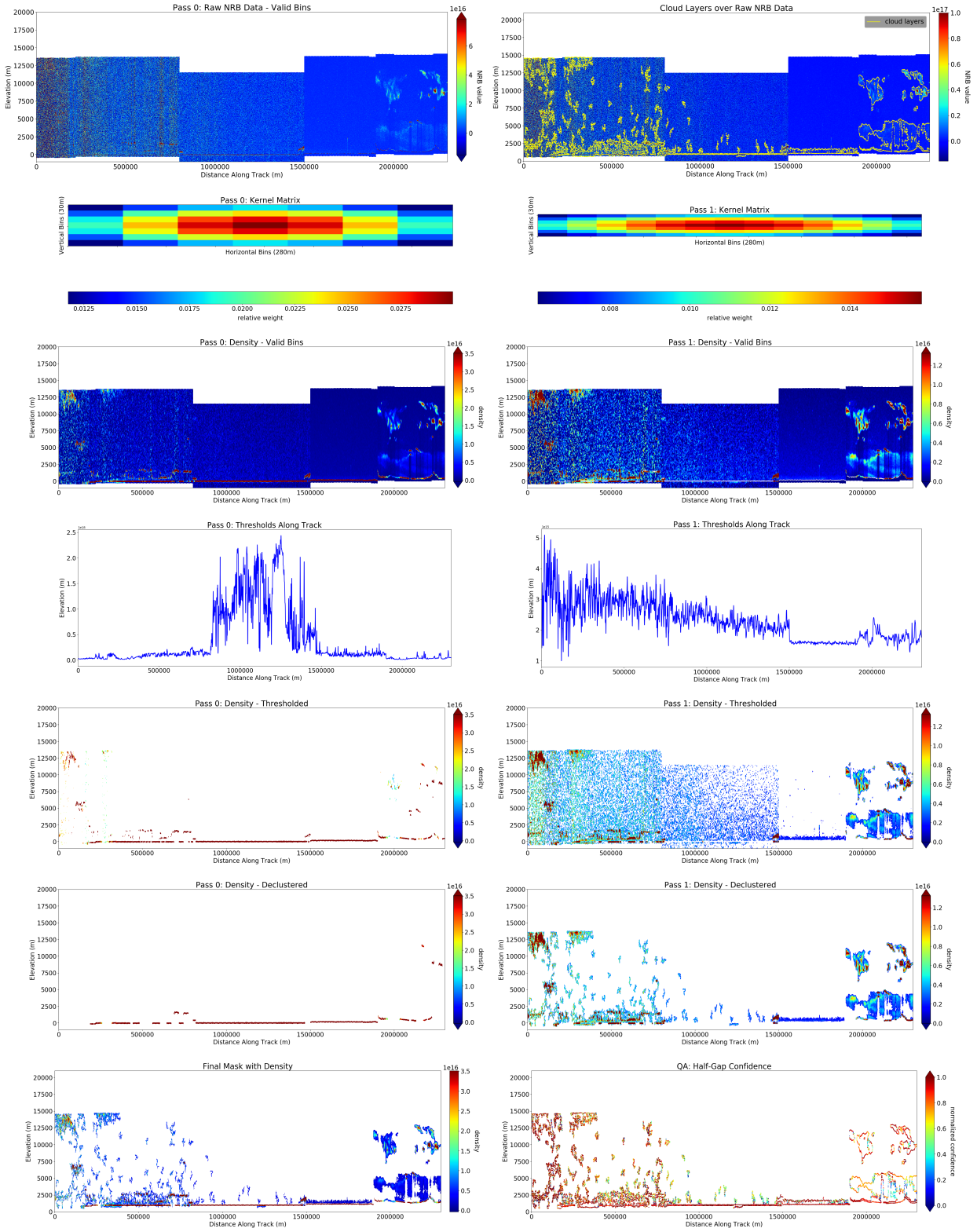


Figure 43-11. Sensitivity analysis of ICESat-2 ATLAS data post-launch: (t74) Smaller quantile in pass 2. Algorithm run on sub-sampled regions in the ATL04_20181017T002107_02810101_950_01.h5 data file using returns from profile 2 of the ATLAS beam configuration.

σ = 3,3	cutoff = 1,1
a_m = 10,20	min cluster size = 300,600
threshold_bias = 10E+14,10E+14	downsampling = 1,1
threshold_sensitivity = 0.9,1	threshold_segment_length = 2,2
quantile = 0.99,0.55	

19 Sensitivity Study to Optimize Parameters for the First Public Release of ICESat-2 Data Products (ASAS code v5.1; 951 data)

Summary. ICESat-2 Atmospheric Data Products will be released together with other data products in the first public release, planned for late May 2019. In preparation for this release, results of ASAS code v5.1 (v951 data) are analyzed in a new sensitivity study. Few changes in the input data in ATL04 resulted in a small parameter change for night-time data, while parameter sets for day-time data and twilight data remained the same. The change of parameters for night-time data is prompted by the analysis of aerosol layers compared to other tenuous layers.

Description

The sensitivity study was run for v951 data, using parameter sets (t74, t80-t89). Parameter sets are listed in Table 7.

Algorithm run on sub-sampled regions in the file
ATL04_20181017T002107_02810101_951_01.h5
using returns from profile 3 of the ATLAS beam configuration. regions subsampled to represent different types of clouds and aerosols. All regions are for night-time data.

The following steps are illustrated for each parameter set shown in Figure 47 (by rows):

1. Raw NRB data (valid bins)[ATL04 input data to the DDA-atmos]; Cloud Layer Boundaries over Raw NRB Data
2. Pass 0: Kernel Matrix; Pass 1: Kernel Matrix
3. Pass 0: Density; Pass 1: Density
4. Pass 0: Thresholds Along Track; Pass 1: Thresholds Along Track
5. Pass 0: Density - Thresholded; Pass 1: Density - Thresholded
6. Pass 0: Density - Declustered (Mask 1); Pass 1: Density - Declustered (Mask 2)
7. Pass 0: Final Mask with Density (Combined Mask); Q/A: Half-Gap Confidence

Results:

(I) Continue to use double-density runs.

(II) Continue to use three parameter sets for day-time/night-time/twilight.

(1) **For day-time: Use paramset1=t60**
(as before, section 17).

(2) **For night-time: Use paramset2=t87**
Use **(t87)** with quantile $q_1 = 0.97$ and quantile $q_2 = 0.55$ for night-time conditions.
Previously used **(t74)** with quantile $q_1 = 0.99$ and quantile $q_2 = 0.55$ for night-time conditions.

(3) **(3) For twilight: Use paramset3=t60**
(as before, section 17).

Reasoning: Parameter set (t87) shifts detection of a more versatile set of clouds into the run-1 group of optically thick clouds and thus enhances the range of clouds and aerosols that can be discriminated in the second density run. While the parameter change appears to be small, the effect is important "down the road" for the detection of layer variability within tenuous layers and aerosols, as opposed to low-level spatial variability in intensity. Using the term "intensity" for values several variables: NRB values and density fields.

See the next section (under construction).

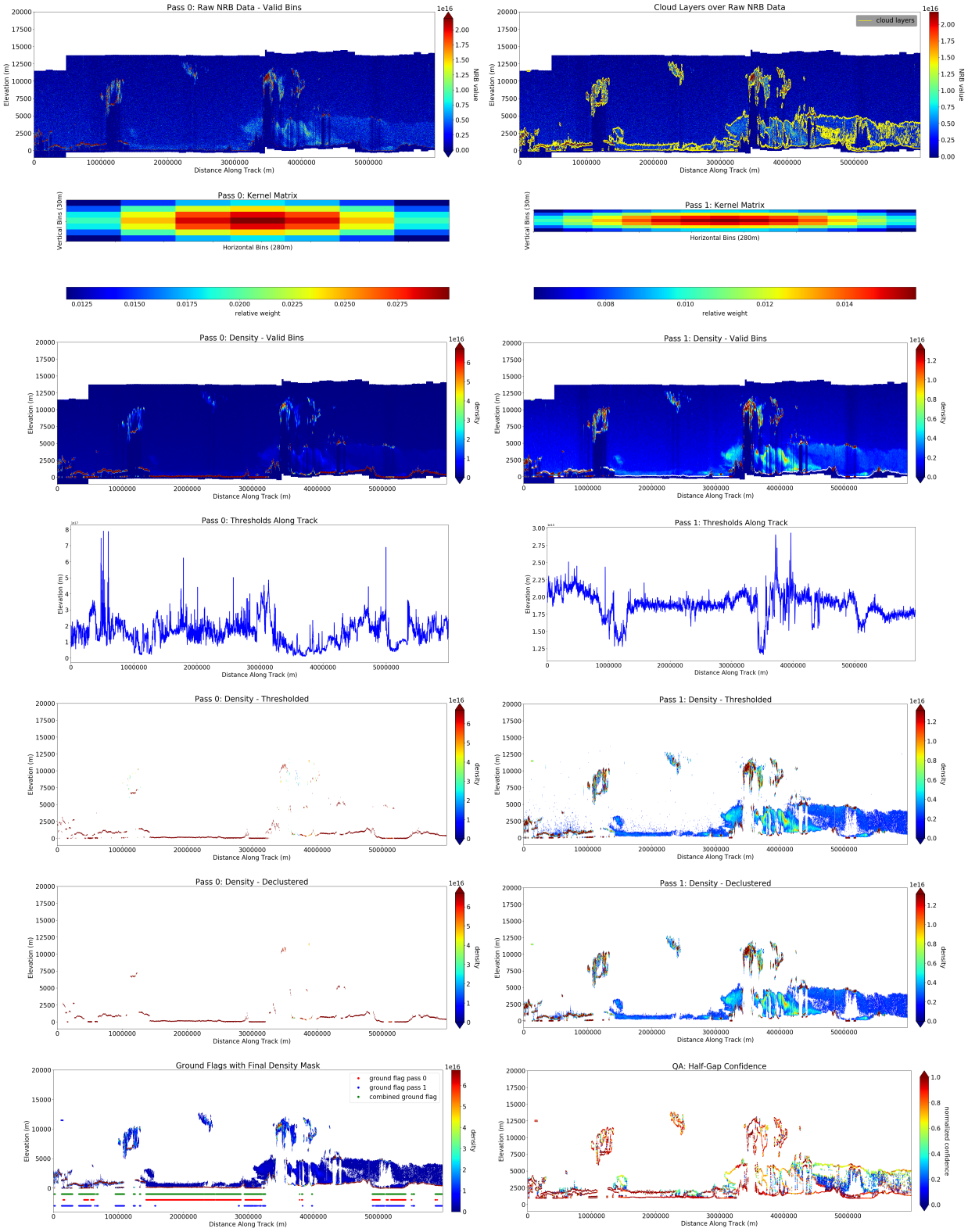


Figure 47-1. Sensitivity analysis of ICESat-2 ATLAS data post-launch (night-time data): (t74) Previous default for night-time section. Algorithm run on sub-sampled regions in the ATL04_20181017T002107_02810101_951_01.h5 data file using returns from profile 3 of the ATLAS beam configuration.

σ = 3,3	cutoff = 1,1
a_m = 10,20	min cluster size = 300,600
threshold_bias = 10E+14,10E+14	downsampling = 1,1
threshold_sensitivity = 0.9,1	threshold_segment_length = 2,2
quantile = 0.99,0.55	

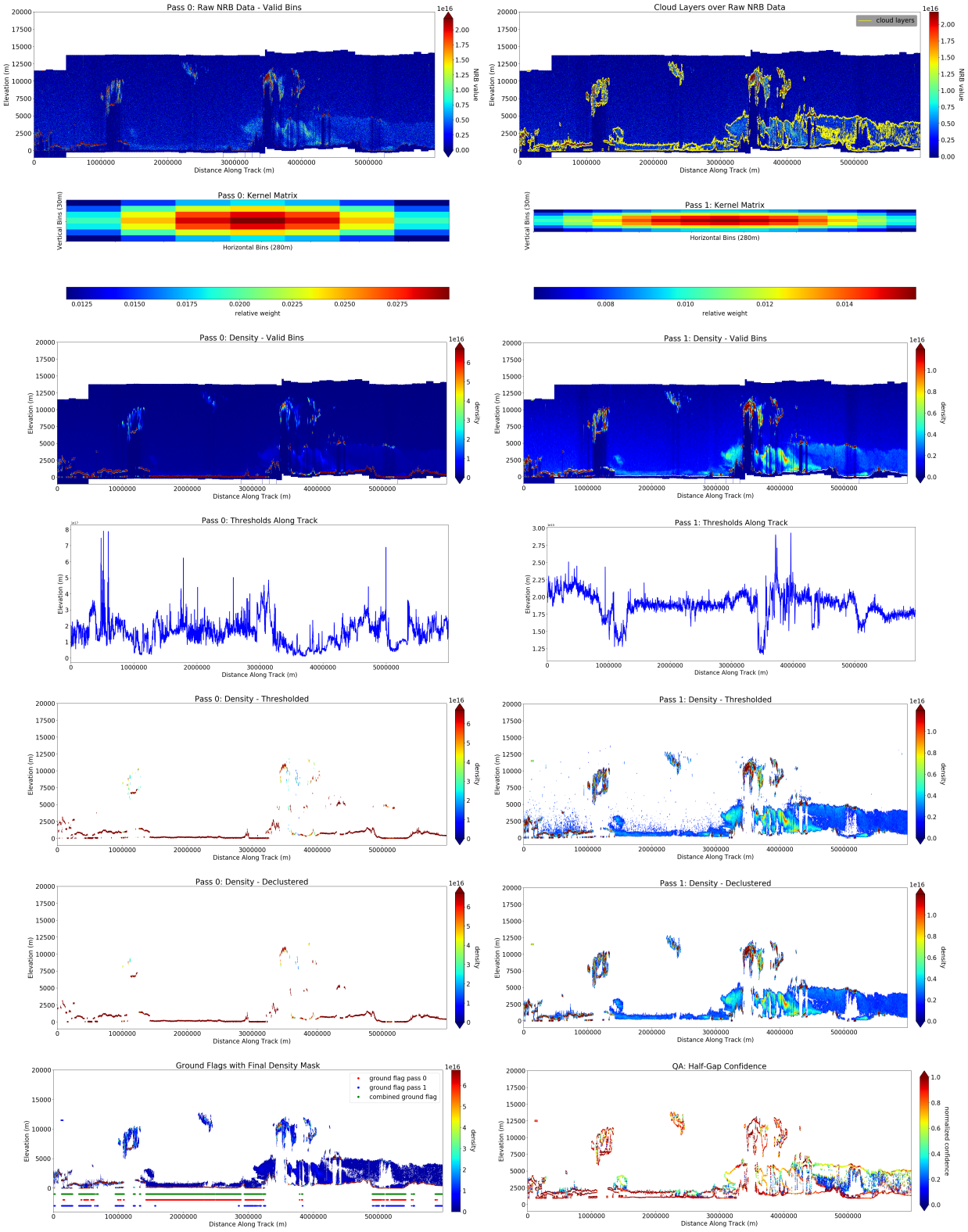


Figure 47-2. Sensitivity analysis of ICESat-2 ATLAS data post-launch (night-time data): (t80) Smaller cluster size in run1 (than in t74). Algorithm run on sub-sampled regions in the ATL04_20181017T002107_02810101_951_01.h5 data file using returns from profile 3 of the ATLAS beam configuration.

σ = 3,3	cutoff = 1,1
a_m = 10,20	min cluster size = 150,600
threshold_bias = 10E+14,10E+14	downsampling = 1,1
threshold_sensitivity = 0.9,1	threshold_segment_length = 2,2
quantile = 0.99,0.55	

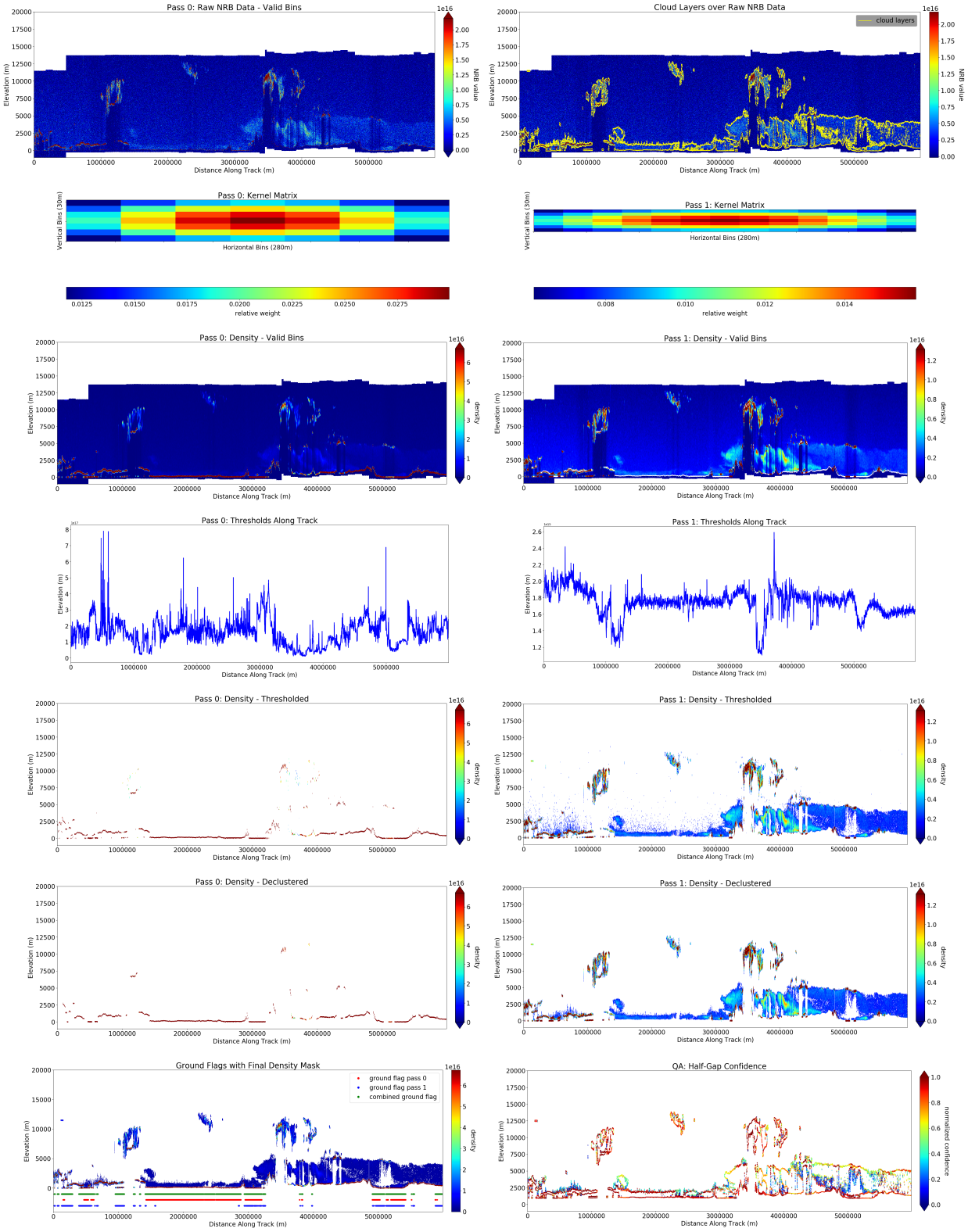


Figure 47-3. Sensitivity analysis of ICESat-2 ATLAS data post-launch (night-time data): (t84) Lower quantile in run2 (than in t74). Algorithm run on sub-sampled regions in the ATL04_20181017T002107_02810101_951_01.h5 data file using returns from profile 3 of the ATLAS beam configuration.

σ = 3,3	cutoff = 1,1
a_m = 10,20	min cluster size = 300,600
threshold_bias = 10E+14,10E+14	downsampling = 1,1
threshold_sensitivity = 0.9,1	threshold_segment_length = 2,2
quantile = 0.99,0.45	

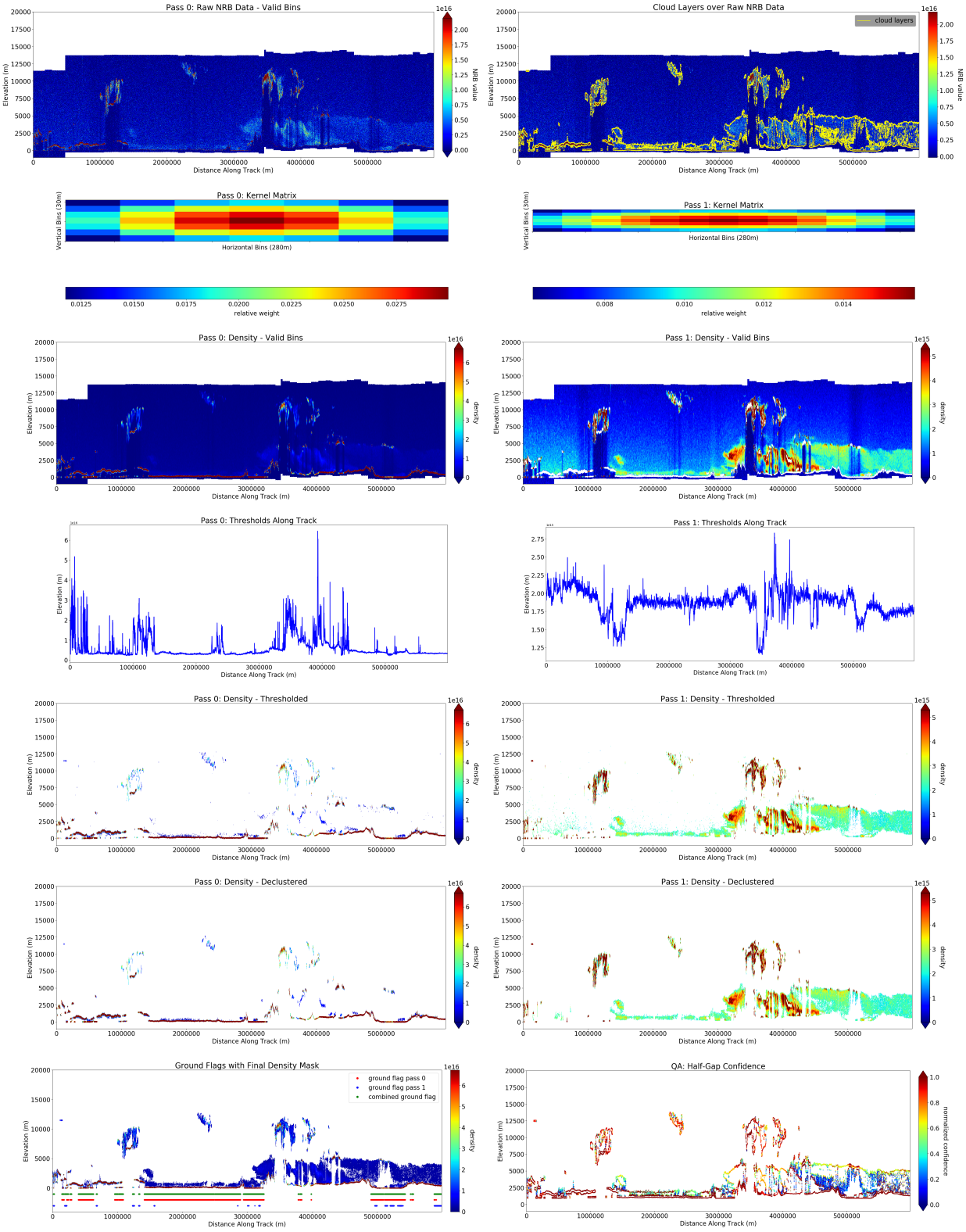


Figure 47-4. Sensitivity analysis of ICESat-2 ATLAS data post-launch (night-time data): (t85) Lower quantile in run2 (than in t74). Algorithm run on sub-sampled regions in the ATL04_20181017T002107_02810101_951_01.h5 data file using returns from profile 3 of the ATLAS beam configuration.

σ = 3,3	cutoff = 1,1
a_m = 10,20	min cluster size = 300,600
threshold_bias = 10E+14,10E+14	downsampling = 1,1
threshold_sensitivity = 0.9,1	threshold_segment_length = 2,2
quantile = 0.99,0.5	

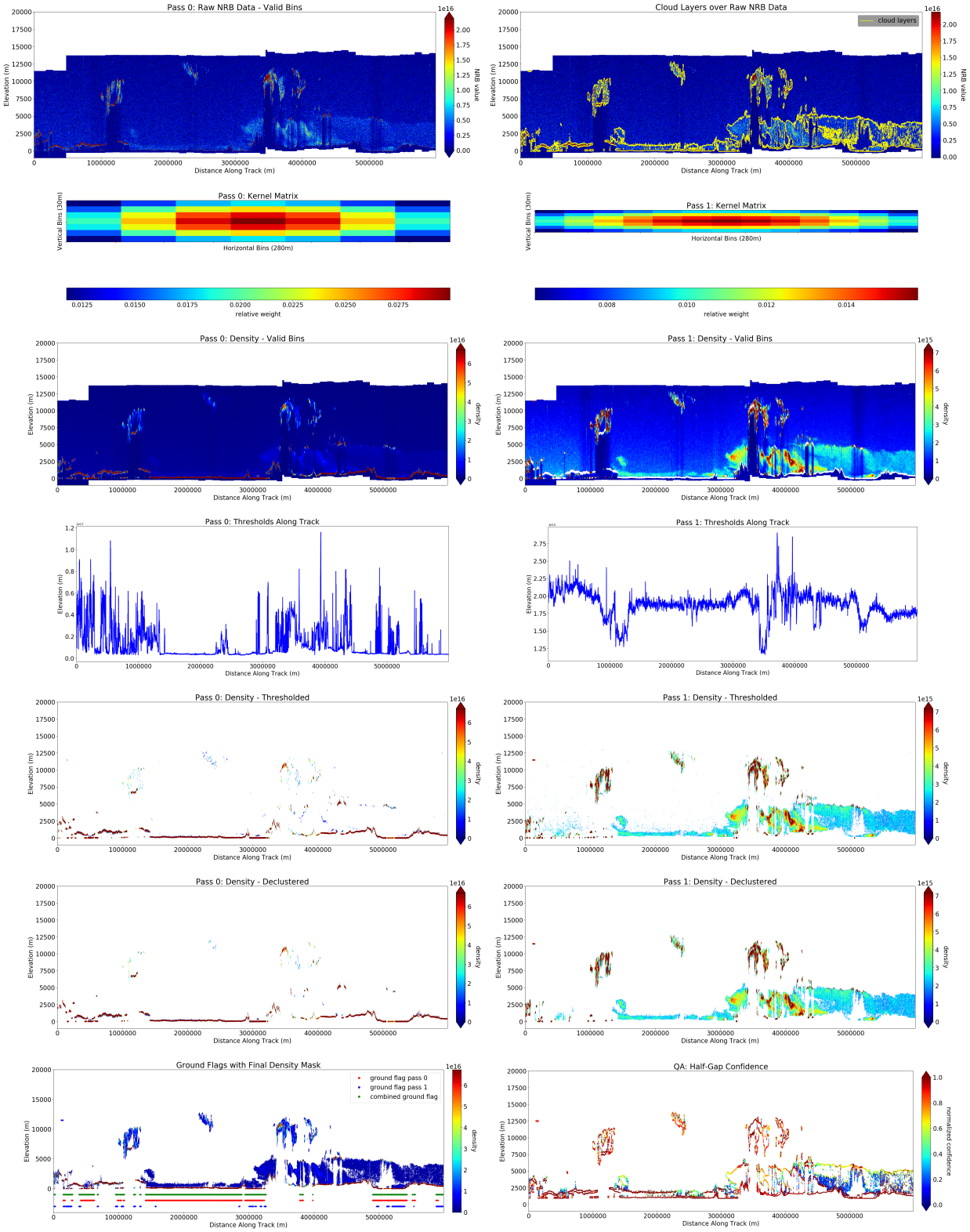


Figure 47-5. Sensitivity analysis of ICESat-2 ATLAS data post-launch (night-time data): (t87) Lower quantile in run1 (than in t74). New default for night-time. Algorithm run on sub-sampled regions in the ATL04_20181017T002107_02810101_951_01.h5 data file using returns from profile 3 of the ATLAS beam configuration.

σ = 3,3	cutoff = 1,1
a_m = 10,20	min cluster size = 300,600
threshold_bias = 10E+14,10E+14	downsampling = 1,1
threshold_sensitivity = 0.9,1	threshold_segment_length = 2,2
quantile = 0.97,0.55	

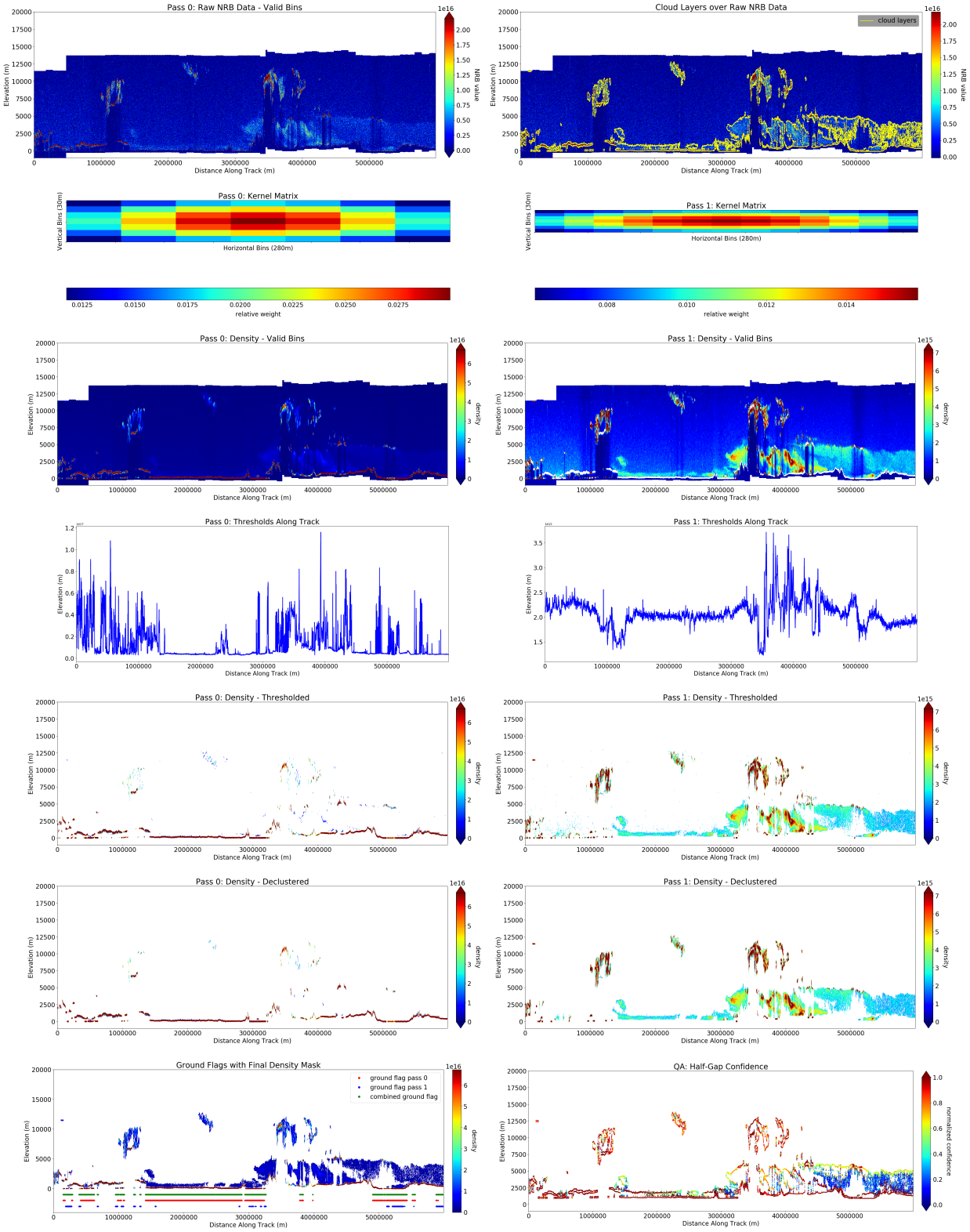


Figure 47-6. Sensitivity analysis of ICESat-2 ATLAS data post-launch (night-time data): (t88) Lower quantile in run1, higher quantile in run2 (than in t74). Algorithm run on sub-sampled regions in the ATL04_20181017T002107_02810101_951_01.h5 data file using returns from profile 3 of the ATLAS beam configuration.

σ = 3,3	cutoff = 1,1
a_m = 10,20	min cluster size = 300,600
threshold_bias = 10E+14,10E+14	downsampling = 1,1
threshold_sensitivity = 0.9,1	threshold_segment_length = 2,2
quantile = 0.97,0.65	

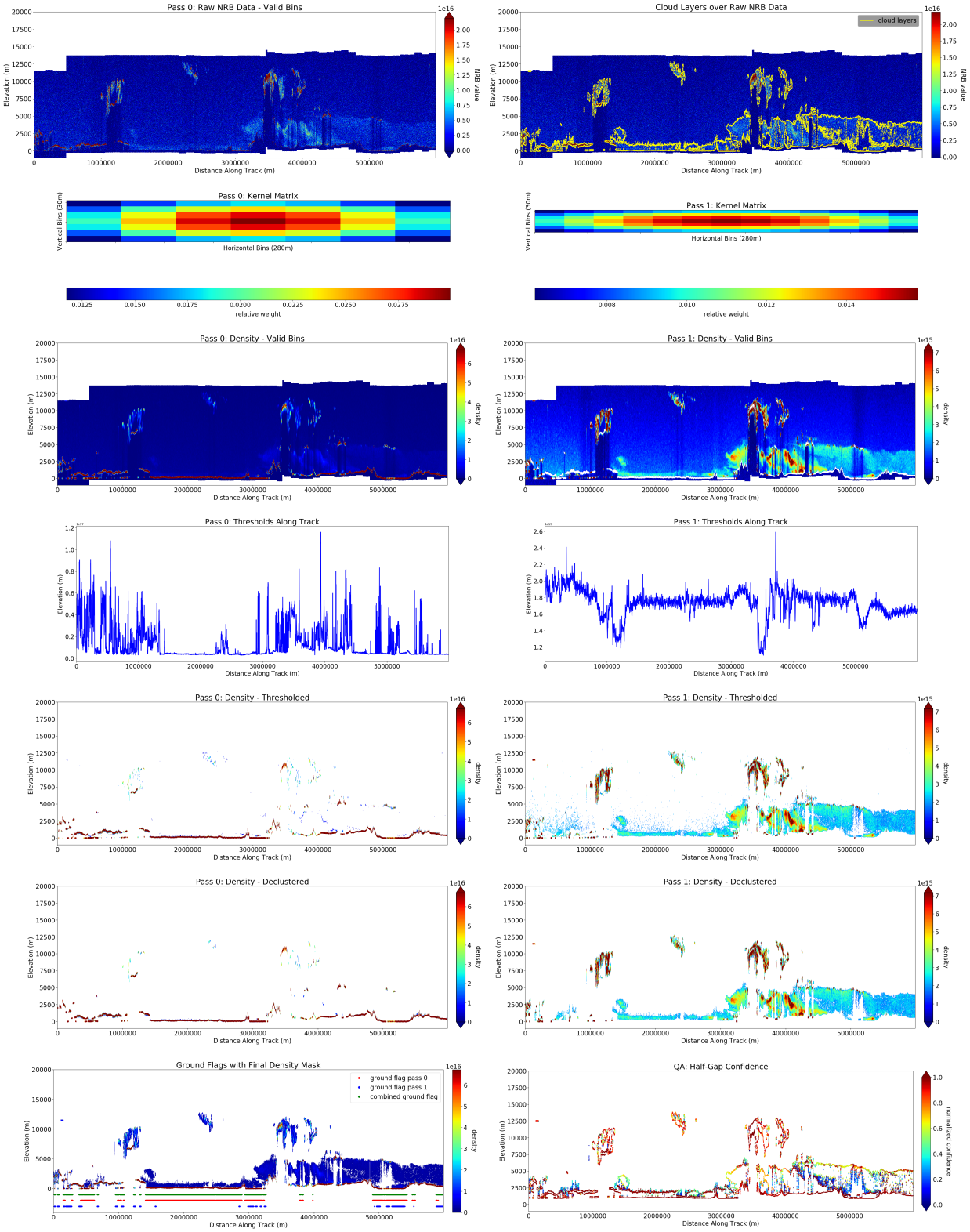


Figure 47-7. Sensitivity analysis of ICESat-2 ATLAS data post-launch (night-time data): (t89) Lower quantile in run1, lower quantile in run2 (than in t74). Algorithm run on sub-sampled regions in the ATL04_20181017T002107_02810101_951_01.h5 data file using returns from profile 3 of the ATLAS beam configuration.

σ = 3,3	cutoff = 1,1
a_m = 10,20	min cluster size = 300,600
threshold_bias = 10E+14,10E+14	downsampling = 1,1
threshold_sensitivity = 0.9,1	threshold_segment_length = 2,2
quantile = 0.97,0.45	