Supplementary Materials: Evaluating a Novel Approach to Detect the Vertical Structure of Insect Damage in Trees Using Multispectral and Three-Dimensional Data from Drone Imagery in the Northern Rocky Mountains, USA

S1. Outbreak history of study area and drone mission flight lines

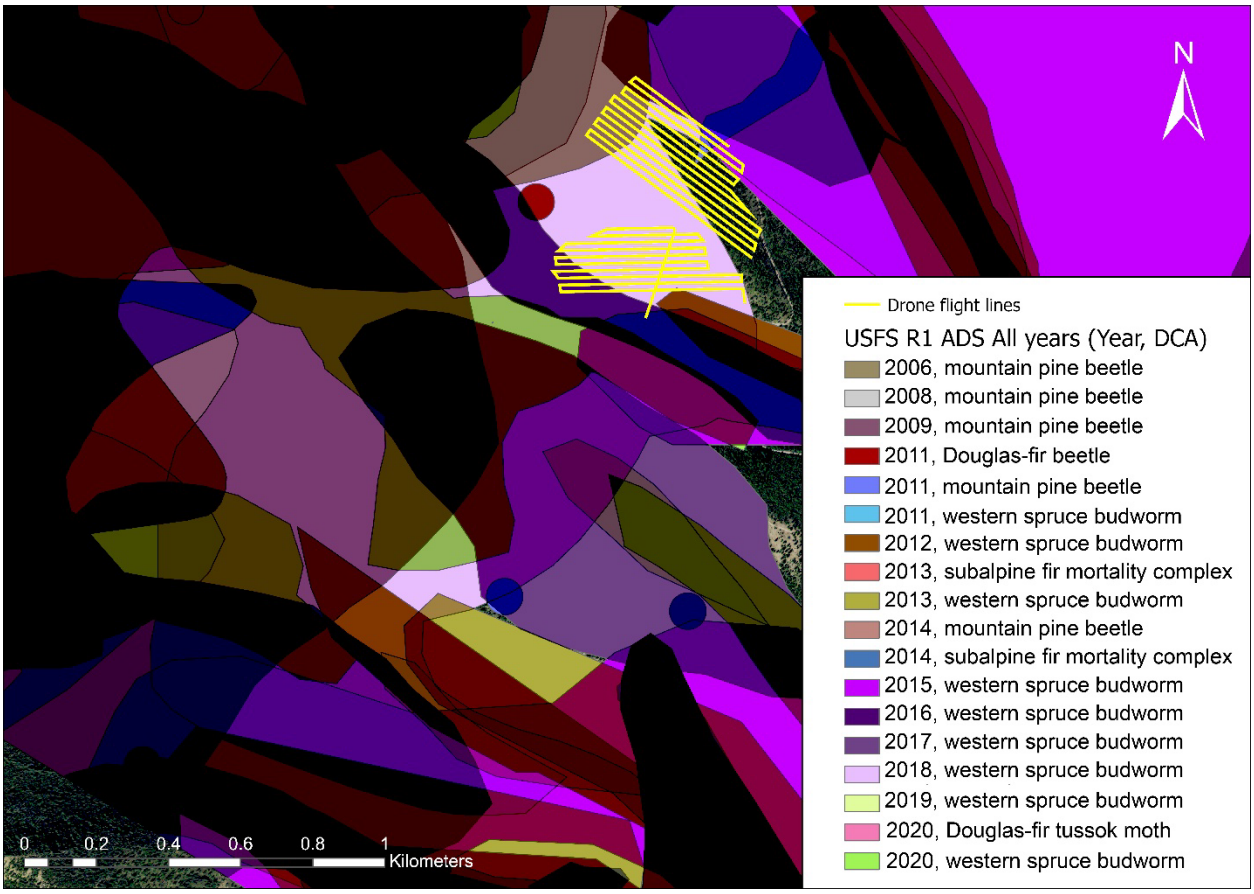

Figure S1. Drone flights (yellow lines) superimposed on polygons from the USDA Forest Service ("USFS") Region 1 ("R1") Aerial Detection Surveys ("ADS") showing insect damage from recent years. "DCA": damage causal agent.

S2. Drone data pre-processing

S2.1. GCP marker corrections and point cloud optimization

Table S1. Estimated positional errors (root mean square error, RMSE) from each processing step by drone mission. RMSE values are estimated using Agisoft Metashape and are accessible via the generation of processing reports. The control points and check points are chosen from the pool of available GCP markers for each drone mission site.

Processing Stage	Mission	X Error (m)	Y Error (m)	Z Error (m)	Total Error (m)
RTK-GNSS error	M1	2.04	0.68	0.48	2.20
	M2	1.66	1.10	1.76	2.66
	Mean	1.85	0.89	1.12	2.43
Final product fit (control points RMSE)	M1	0.12	0.25	0.20	0.35
	M2	0.01	0.01	0.05	0.05
	Mean	0.07	0.13	0.13	0.20
Final product accuracy (check points RMSE)	M1	0.02	0.09	0.03	0.09
	M2	0.01	0.03	0.07	0.07
	Mean	0.02	0.06	0.05	0.08

S2.2. Assignment of multispectral reflectance values to SfM point cloud

The MX-RedEdge sensor records data in five spectral bands (MicaSense, Inc, 2023), and these multispectral reflectances can be assigned to the point cloud using the "*Calculate Point Colors*" option in Metashape [1]. However, the American Society for Photogrammetry and Remote Sensing (ASPRS) [2] specifications for .las format data limit the band data storage to only 4 bands while exporting (R, G, B, and NIR). Therefore, a workaround was implemented to create a .las point cloud file. The point cloud constructed from the multispectral sensor was exported as a .XYZ file (plain text file) from Agisoft Metashape. The .XYZ point cloud file was imported into CloudCompare (version 2.12.4, https://www.cloudcompare.org/, accessed August 2022) and re-exported as a .las file, with the projected coordinate system set to EPSG:26912 (NAD83/UTM Zone 12). During this process, the five-band reflectance data were stored as "*Extra Bytes Variable Length Records (VLR)*" on the .las file. If future updates to the LAS specifications allow for more than four-band information storage, then a streamlined export to the .las file from Agisoft Metashape is suggested.

The SfM point cloud with multispectral reflectance values was imported into R using the `lidR` package [3]. The attributes of each point in the point cloud were stored as a `data.table` format under the `@data` slot of the imported "*LAS-class*" object [3]. The `data.table` package [4] allows for efficient memory allocation and recycling in R and is thus central to data storage and manipulation in the `lidR` package [3]. Since the `data.table` format is an extension of R's native `data.frame` class [4], the `data.table` format was used throughout the project, especially when accessing and updating point cloud data attributes.

S3. Reference data

S3.1. Utilization of individual drone-captured images

The locations of the reference trees from the manually delineated tree crown polygon layer were exported as a point shapefile to Metashape. The imported reference tree location shapefile was converted to "*markers*", and the "*filter photos by markers*" option was used to view individual drone-captured images that contained the reference trees. This backtracking to individual drone images allowed for the location determination and evaluation of the damage along the height of the trees, as the same tree can be seen in multiple images from different angles. These multi-image evaluations of tree damage were used for qualitative assessments of the random forest classification of points in the point cloud and the separation of individual trees into different damage severities.

S3.2. On-screen selection of points representing green, grey, red, and shadow classes on the point cloud data set

The point cloud with tree segmentation was loaded into R, and an "*ID_point*" attribute was programmatically added to the point cloud data set. The "*ID_point*" attribute represented a unique identification number for each point in the point cloud data set.

The point cloud with the "*ID_point*" attribute was imported into CloudCompare with the reference trees polygon layer from ArcGIS (Figure S2a). The "*point-picking*" tool was used to identify the "*ID_point*" attributes of points representing the green, gray, red, and shadow classes within the point cloud of trees from the reference data set (Figure S2b). The "*ID_point*" attribute was recorded in a spreadsheet and imported into R. The point cloud attributes representing the

spectral values and indices of each point were joined with the imported reference data spreadsheet using the "*ID_point*" field. This process resulted in a reference data set of spectral values and indices of points representing the green, gray, red, and shadow classes.

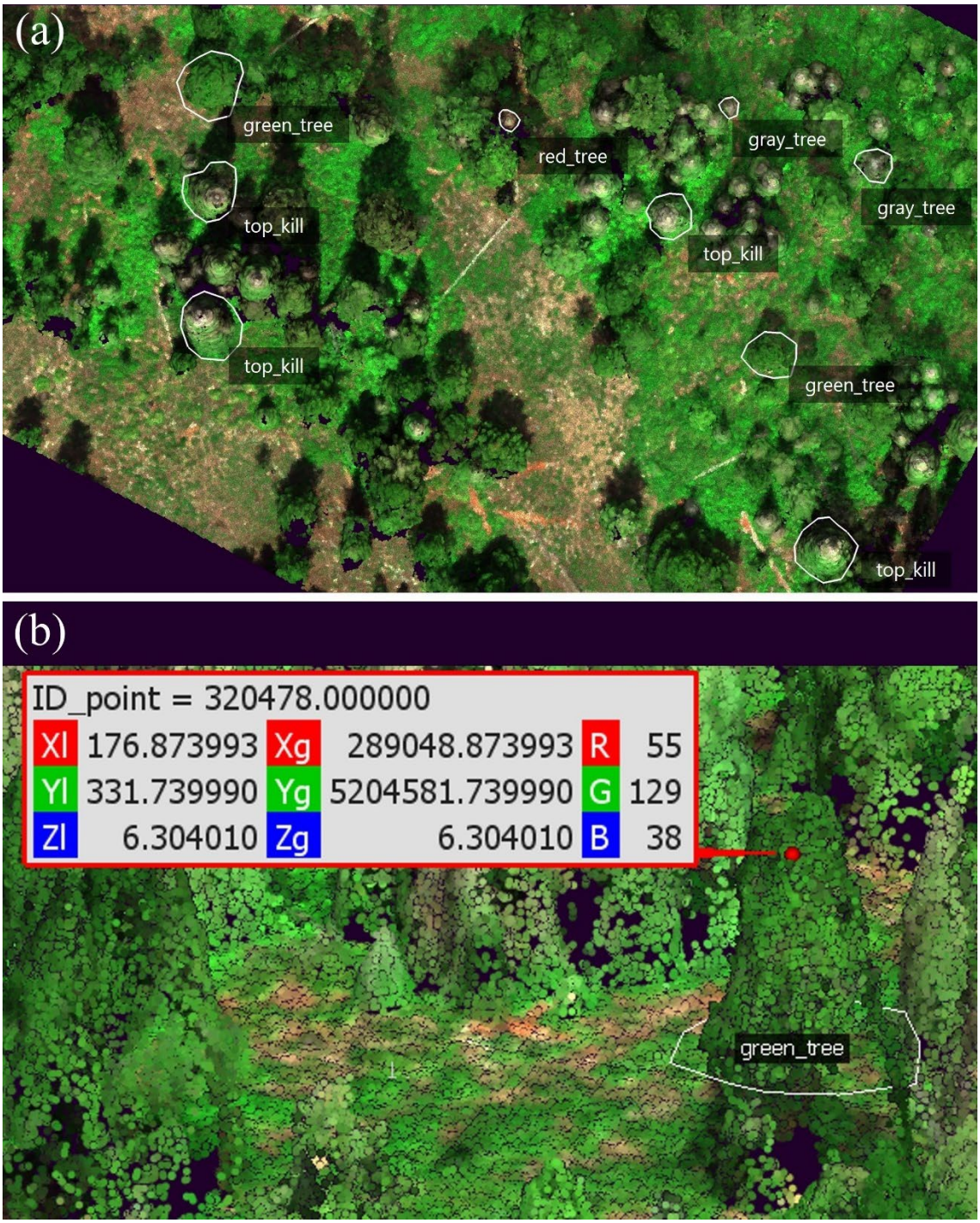

Figure S2. Methods for assembling the reference data set of point clouds using manual point picking in CloudCompare. (a) Top–down view (displayed in CloudCompare) of the subset of the point cloud with the manually identified reference trees (labeled with text). (b) Example of using the "*Point-picking*" tool in CloudCompare to query the "*ID_point*" of the point representing the green class on the reference tree labeled "*green_tree*".

S4. Segmentation of point cloud into trees

S4.1. Ground classification and height normalization of point cloud

The ground/non-ground classification and height normalization of the point cloud required for individual tree segmentation followed recommendations from Mohan et al. [5] and Roussel et al. [6] using the `lidR` R-package [3].

The "*algorithm*" parameter of the `normalize_height` function was set to the K-nearest neighbor with an inverse distance weighting algorithm using the `knnidw` function available in the `lidR` package [3]. The authors of the `lidR` package suggested the `knnidw` algorithm as a suitable compromise between the other two available algorithms [3]. These other algorithms are triangular irregular network (TIN) interpolation, which is computationally fast but has weak estimations of edges with prominent edge artifacts, and kriging interpolation, which provides the best results in terms of representation of the terrain with minimal edge artifacts but is computationally demanding [3].

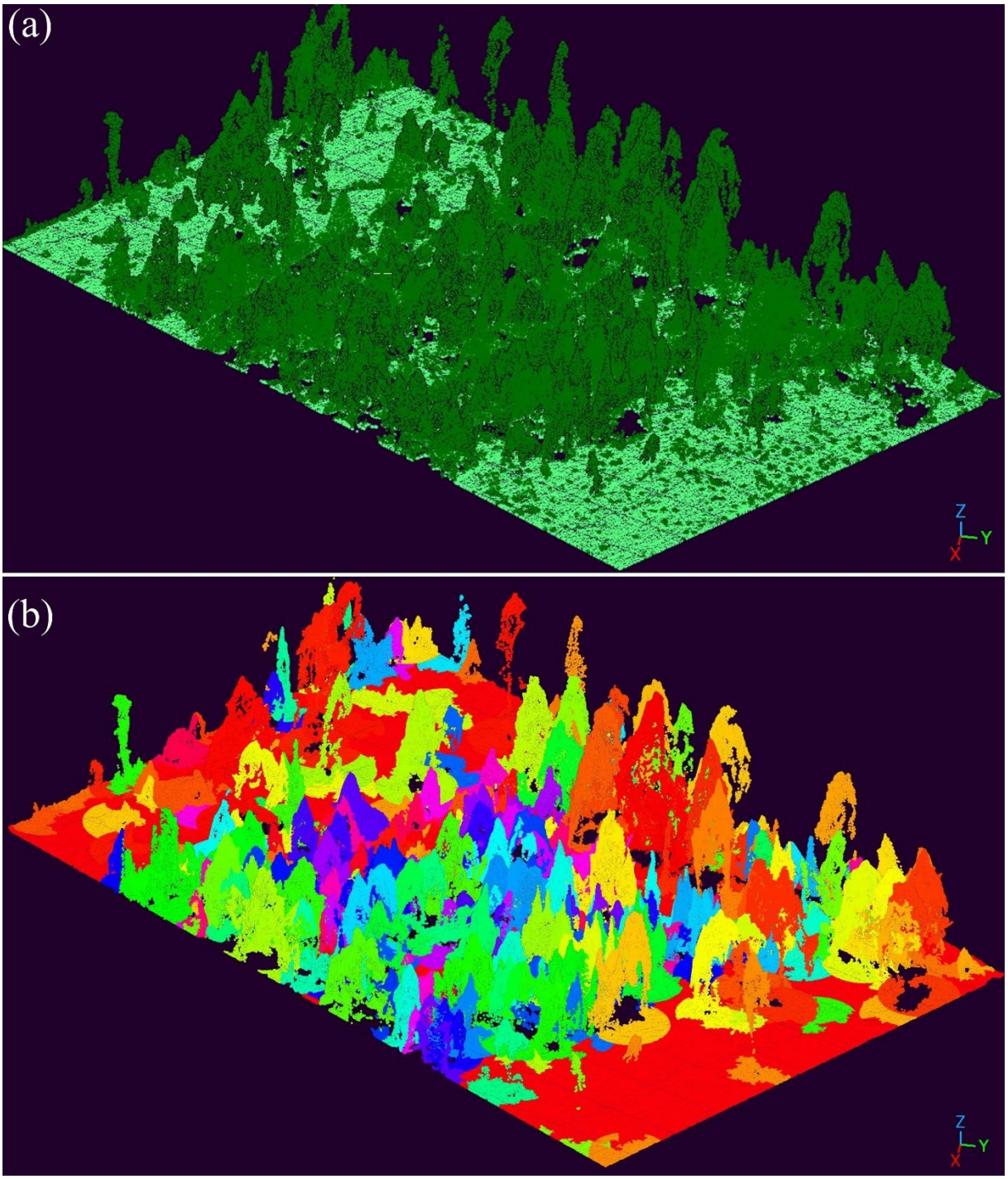

Figure S3. (a) Height normalization for a subset of the point cloud. Points classified as ground are light green and non-ground points are dark green. Visual assessment indicates the successful execution of the ground classification algorithm (ground-classified points predominantly located at lower heights) and height normalization (flat areas demonstrate the

removal of the influence of terrain). (b) Tree segmentation (individual trees are represented in different colors).

S4.2. Tree segmentation algorithms

S4.2.1. Algorithm background and parameters

The `lidR` package consists of various tree segmentation methods that can be broadly categorized into two types: image-based (raster-based) and point cloud-based methods [3]. These methods can be used to perform point cloud segmentation of trees.

In image-based tree segmentation methods, the treetops are first detected using a canopy height model (CHM), and the detected treetops are used to segment the point cloud into individual trees [3]. The CHM was generated using the `rasterize_canopy` function from the `lidR` package [3] and was smoothed using the `focal` function from the `terra` package [7]. A majority of drone-based studies in forestry use a fixed window local maximum filtering (LMF) algorithm for tree segmentation [8–10]. Hence for this project, the `lmf` algorithm with the `locate_trees` function from the `lidR` package [3] was used for tree segmentation. The `lmf` algorithm parameters for a fixed moving circular window were set to a radius of 1.5 m and a minimum height of 1.35 m. The LMF-located treetop points and the smoothed CHM were used as input for the `segment_trees` function to segment the point cloud into tree segments using the `silva2016` method in the `lidR` package [3,11]. The resulting segmented point cloud consisted of a new attribute, `treeID`, allocated to every point that was segmented as a tree object, i.e., every point that was segmented as an individual tree had the same `treeID` attribute value [3].

The `li2012` algorithm starts with the highest point in the point cloud data set and applies a proximity search, where if a second point falls within a radius (set as a parameter), it is

considered to belong to the same tree as the first point, and the algorithm continues to assess all points that fall within the search radius [12]. The algorithm progresses to the next highest point that falls outside of the search radius, considers this outside point as a separate tree, and repeats the search as with the first tree [12]. All points belonging to an identified tree are allocated the same `treeID` attribute [3].

The `crown_metrics` function from the `lidR` package [3] was used for individual tree crown delineation. The `geom` argument for the `crown_metrics` function was set to `concave`.

S4.2.2. Case study to compare tree segmentation algorithms

A preliminary assessment was conducted to determine which tree segmentation algorithm to use for this study. Both tree segmentation algorithms were applied to a subset of the point cloud, and the resulting tree crown polygons were imported in ArcGIS and overlayed on the MS orthomosaic. Qualitative visual assessments and comparisons were conducted to select the better-performing tree segmentation algorithm.

The visual assessment of the image-based [11] and point cloud-based [12] segmentation algorithms indicated a better performance by the point cloud-based algorithm [12]. The image-based segmentation algorithm [11] had difficulty separating tree crowns in areas with a dense forest canopy, failing to detect some of these trees in the subset area (Figure S3a). Conversely, the point cloud-based algorithm [12] had some errors associated with dividing a single tree into multiple trees (over-segmentation) (Figure S3b). Correctly detecting trees but with over-segmentation was preferred to failing to detect the presence of trees, and therefore, the point cloud-based tree segmentation method [12] was used here.

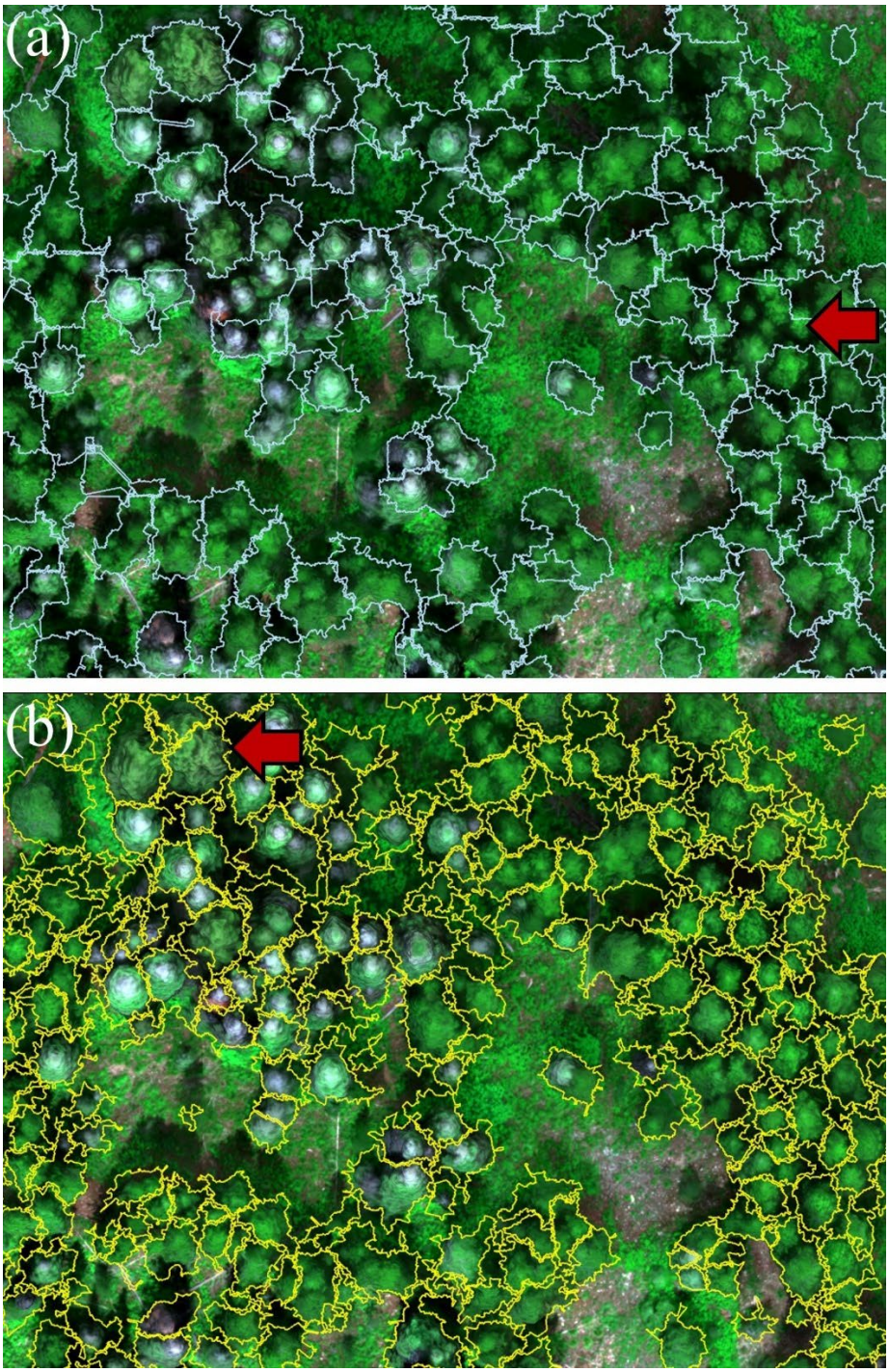

Figure S4. Example of results of tree segmentation from two algorithms. (a) Using the image-based segmentation algorithm (Silva et al., 2016); arrow shows error in detecting trees (no tree detected within polygon indicated by arrow). (b) Using the point cloud-based segmentation algorithm (Li et al., 2012); arrow shows error of dividing one tree into multiple segments (trees).

S4.2.3. Accuracy assessment of tree segmentation

The crowns delineated by the tree segmentation were clipped to a smaller extent to remove the edge artifacts of the orthomosaic (Figure 1). The "*Create Accuracy Assessment Points*" tool in ArcGIS was used to create randomized reference locations (as point features) within the clipped extent of the tree segmentation polygon layer. The input information source was set to the tree segmentation polygon layer such that the randomized reference locations (ArcGIS point feature) inherited the tree segmentation classification results under the "*Classified*" attribute. Randomized points were evaluated with an on-screen assessment to label each point's reference classification under the "*GrndTruth*" attribute. The "*Classified*" attribute of the randomized point cloud represents the predicted class, and the "*GrndTruth*" attribute represents the reference class. A total of 1000 randomized points were qualitatively assessed to build the accuracy assessment reference data set. The "*Compute Confusion Matrix*" tool in ArcGIS was used for the accuracy assessment reference locations (ArcGIS point feature) to produce the confusion matrix and calculate the accuracy metrics.

The second accuracy assessment of the tree segmentation was performed using the reference data set of manually delineated tree crowns (100 trees) on ArcGIS. The crowns from the reference data were overlaid with the crowns delineated by the tree segmentation (Figure S5a). The "*Union*" tool was used to separate the overlapping areas of tree crowns from the reference data set and tree segmentation (area *A*, Figure S5b) as well as the areas that did not overlap (areas *B*

and *C*, Figure S5b). The sum of areas *A*, *B*, and *C* were used to calculate the Sørensen's coefficient (SC; Equation 1) [13].

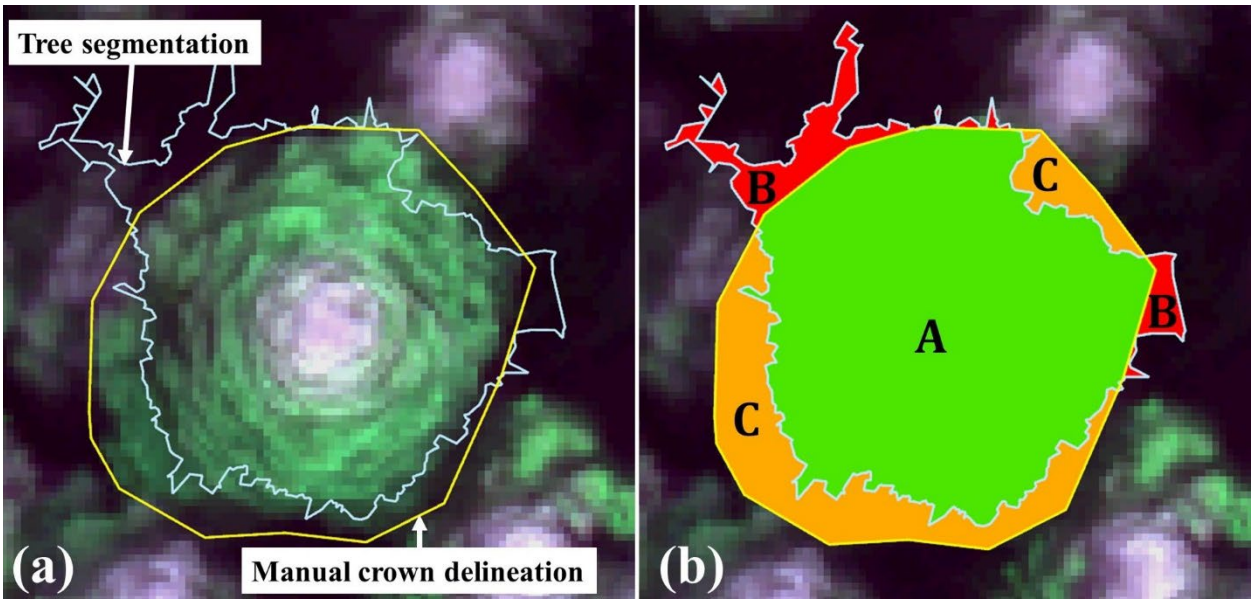

Figure S5. (a) Manually delineated crown polygon (reference data set) overlaid with crown delineation from the tree segmentation algorithm, with true color drone orthomosaic as the basemap imagery. (b). Polygons A, B, and C produced using the "*Union*" tool in ArcGIS with the tree segmentation and manual delineated crowns. "A" is the area identified as the tree crown by the manual delineation and the tree segmentation, "B" is area identified as the tree crown by the tree segmentation algorithm but not by the manual delineation, and "C" is the area identified as the tree crown by the manual delineation but not by the tree segmentation.

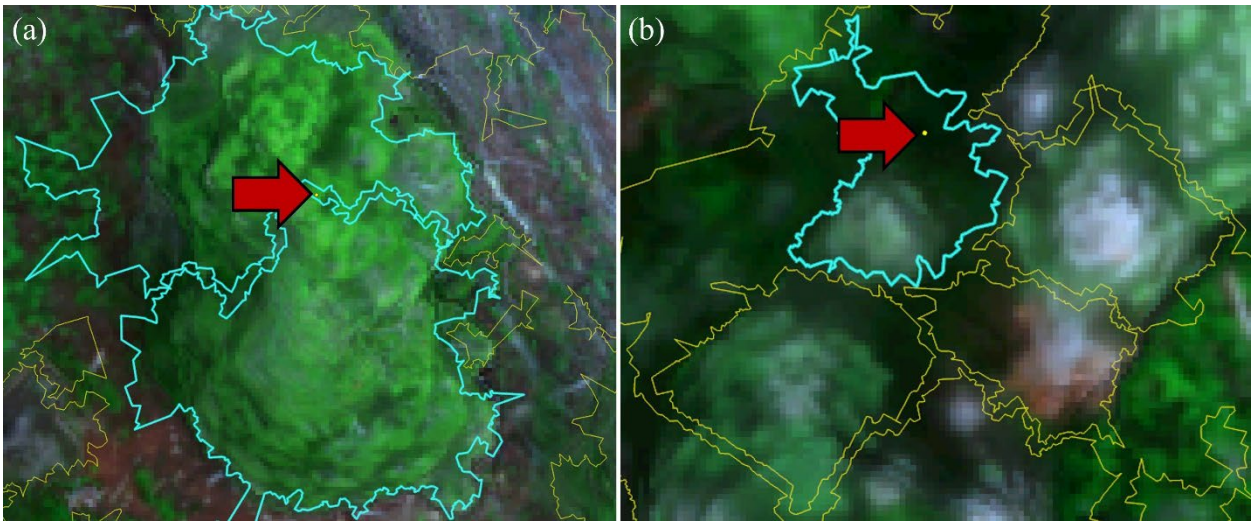

Figure S6. (a) Example from the 352 cases of the "*tree*" class misclassified as the "*not tree*" class (Table 2). The point (reference location, yellow dot indicated by red arrow) falls in between the over-segmented tree crown polygons of the tree of interest. (b) Example from the 31 cases of the "*not tree*" class misclassified as "*tree*" class due to tree segmentation issues. The point falls on the ground or understory vegetation but was erroneously identified as being within the tree crown polygon.

S5. Correlation matrix of predictor variables used in random forest models

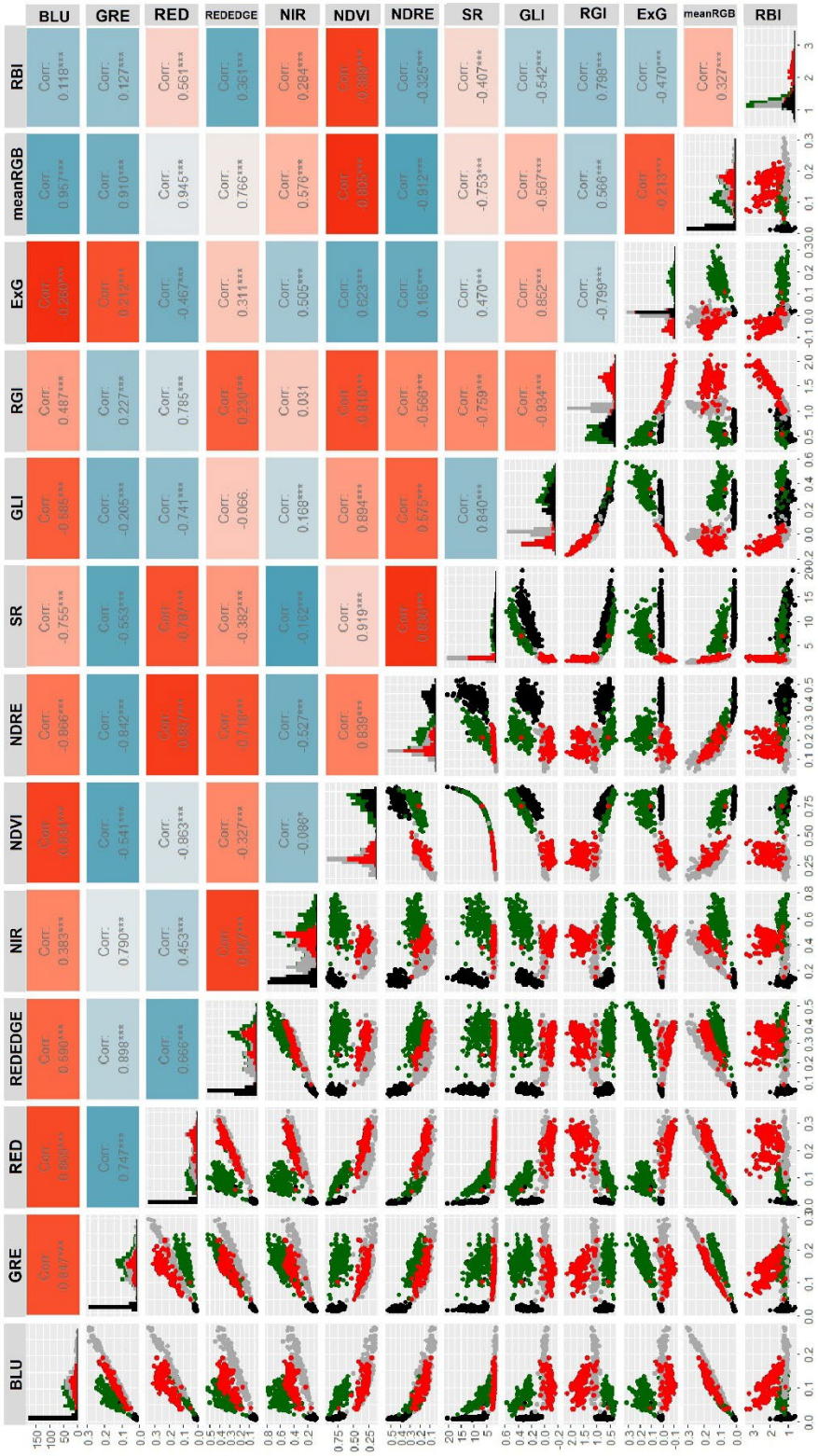

Figure S1. Correlation matrix of predictor variables used in random forest models. The diagonal from top left to bottom right consists of the histograms of classes for each spectral band and index. The colors on the histograms represent the reference data classes: green is healthy, red is red, gray is gray, and black is shadow. The bottom left area below the diagonal contains scatterplots of the reference data classes, with bands and indices plotted on the corresponding axes. The top right area above the diagonal is a heat map ranging from deep blue to deep red that represents the strength of the correlation between the predictor variables.

S6. Evaluation of point classification

Table S2. Top five two-variable and three-variable random forest models.

Rank	Variable	Overall Accuracy (%)	Maximum Pairwise Absolute Correlation Coefficient (|r|)
Two-variable models			
1	RGI + GRE	97.25	0.227
2	RGI + REDEDGE	97.25	0.230
3	RGI + meanRGB	97.25	0.566
4	RGI + BLU	97.00	0.487
5	RGI + NDRE	97.00	0.311
Three-variable models			
1	RBI + GLI + GRE	98.75	0.542
2	RBI + NDVI + REDEDGE	98.625	0.389
3	RBI + NDVI + GRE	98.625	0.541
4	RBI + GLI + meanRGB	98.625	0.567
5	RBI + SR + REDEDGE	98.5	0.407

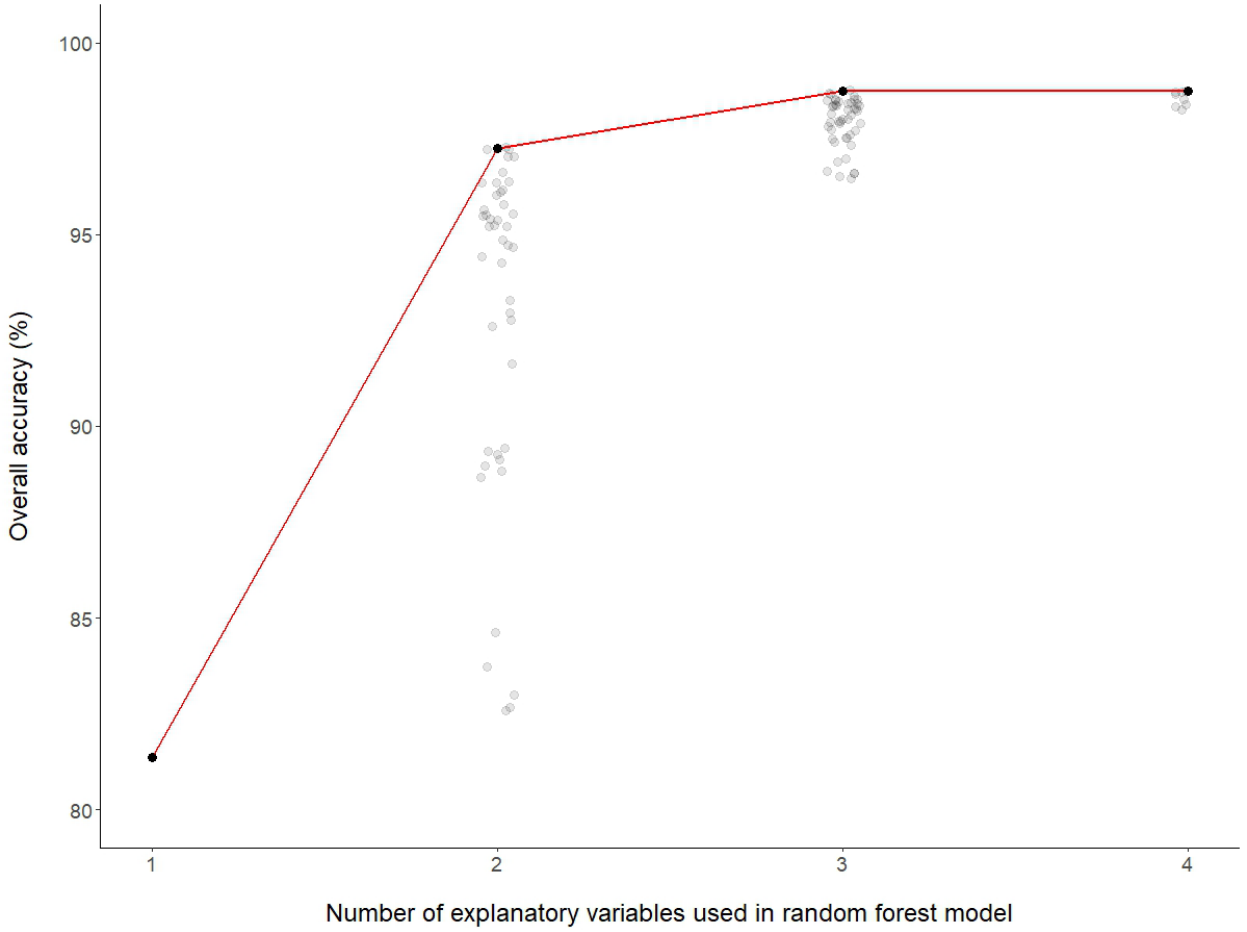

Figure S8. Overall accuracy (OA) as a function of the number of variables used in the random forest (RF) models. Points in black are the highest overall accuracies of RF models from the best subsets' algorithm of explanatory variable combinations with low multicollinearity. The gray points are the overall accuracies from the remainder of the RF models.

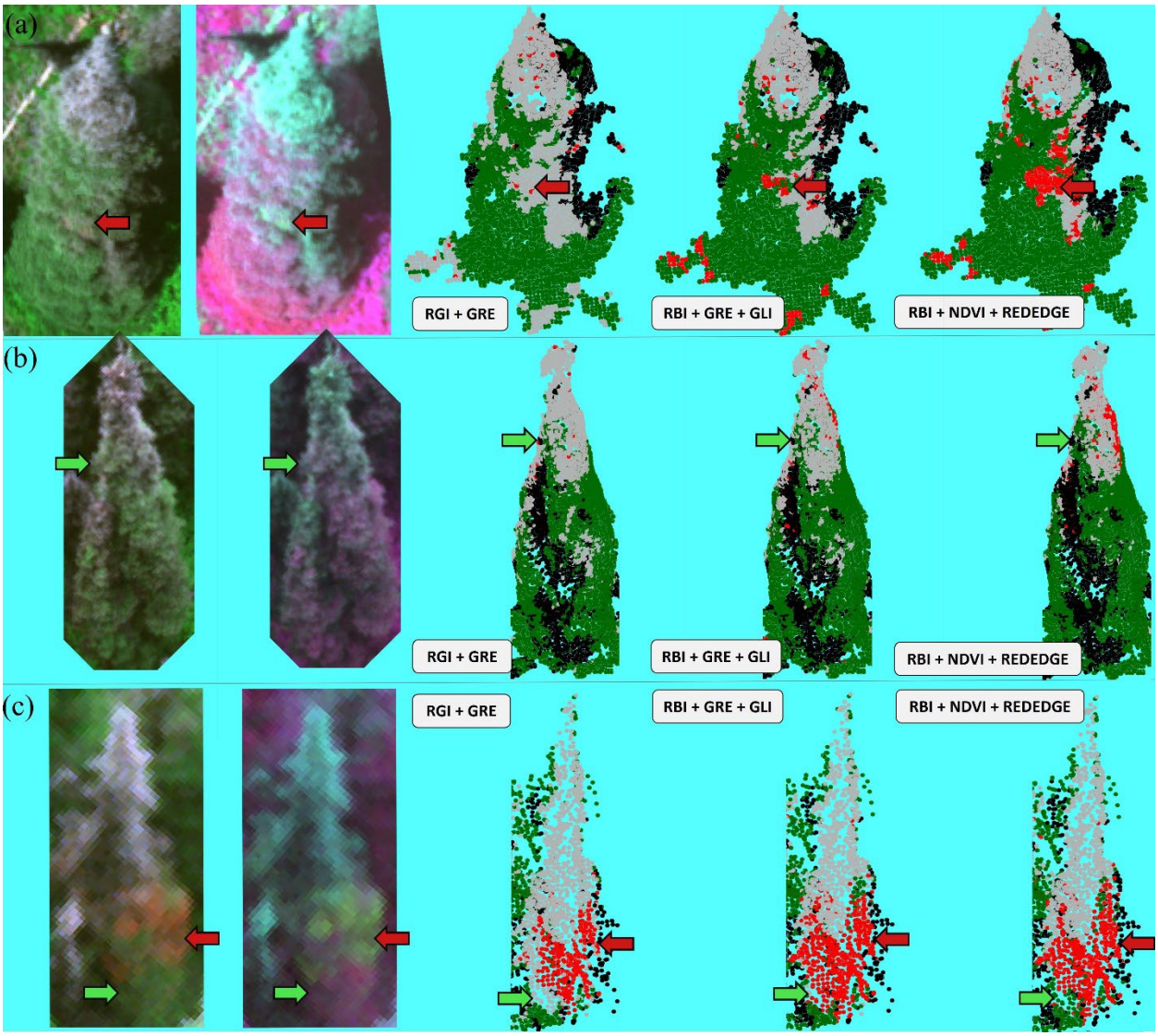

Figure S9. (a), (b), (c) Examples of trees used to evaluate classification models of healthy versus damaged points. Columns (left to right): true color image, false color image (red–green–near infrared), and results from three random forest models (text indicates explanatory variables in models). Green points correspond to green class, red to red class, gray to gray class, and black to shadow class. Red and green arrows indicate damaged (gray and red classes) and healthy areas, respectively, where the second-ranked, three-variable model (RBI + NDVI + REDEGDE model) agrees better with the visual assessment than the other models.

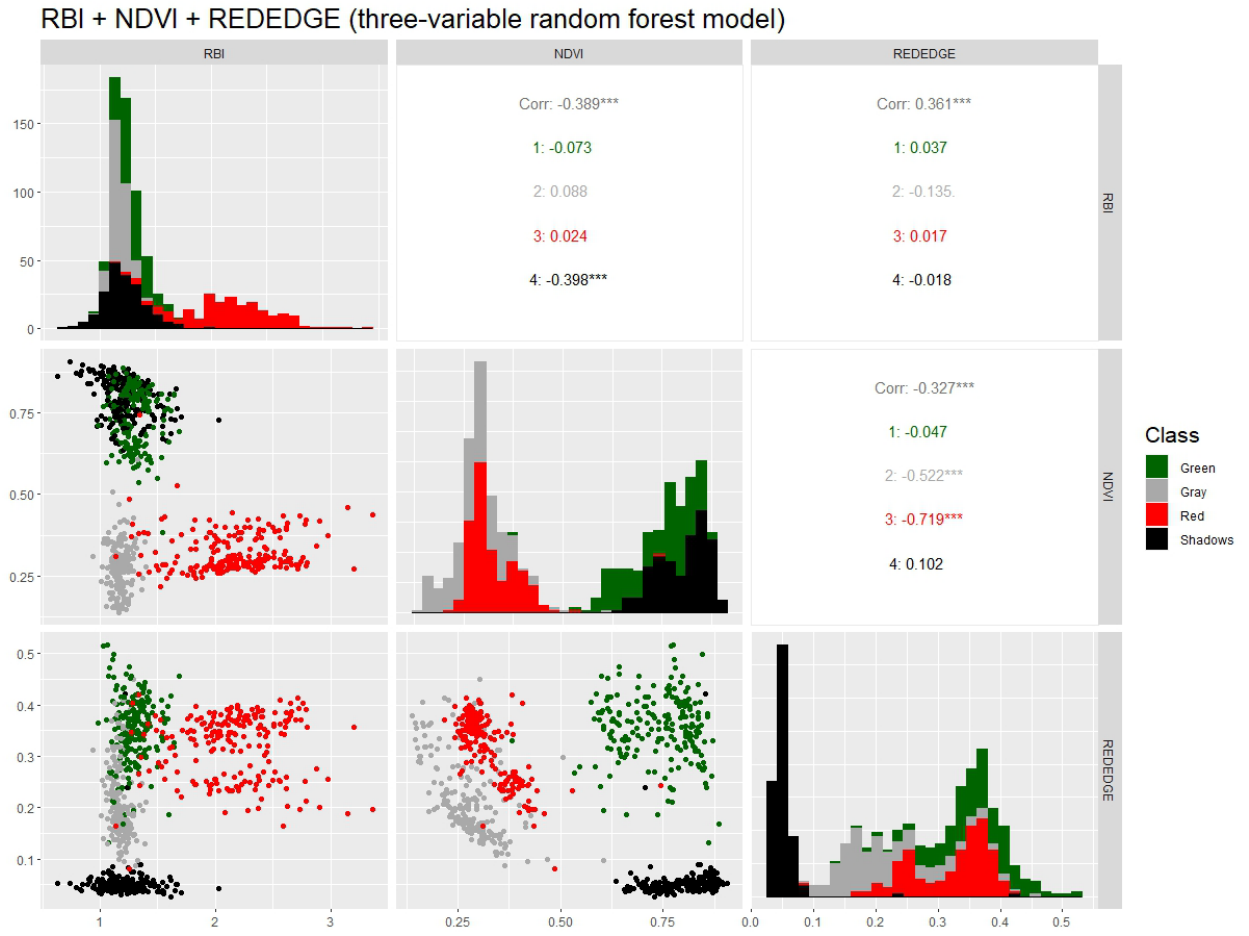

Figure S10. Pairwise plots of predictor variables used to develop the final random forest model (RBI + NDVI + REDEDGE) using the reference data set of points. The diagonal from the top left to the bottom right consists of the histograms of classes for each spectral band and index. The top-right space above the diagonal consists of correlation values for each of the classes in the respective spectral bands and indices. The bottom-left area below the diagonal consists of a scatterplot of reference data classes, with bands and indices plotted on the corresponding axes. The colors of the classes are as labeled in the legend. Figure produced with R function `ggpairs` in the `ggally` package ([14]).

S7. Application of final random forest model to point cloud

The `lidR` R package stores point cloud data as a "*LAS-class*" R object [3]. The attributes of points in the point cloud data were accessed via the `@data` slot of the "*LAS-class*" R object [3] and stored as a `data.table`. A subset `data.table` was created, which consisted of data from columns representing only the reflectance bands (five bands). The vegetation indices (Table 1) were computed and were added as columns to this subset `data.table`. This `data.table` with reflectance bands and vegetation indices was used as the input data to apply the RF model and calculate class probabilities.

The class probabilities for each point from the final RF model were calculated using the `predict` function's "*type* = *prob*" argument [15]. The values returned represent the final RF model's classification probability for each point; these values were added as attributes to the point cloud `data.table`.

The RF model classification and class maximum probabilities were added to the "*LAS-class*" R object using the `@data` slot to manually update the attributes of the point cloud object in R. The "*LASheader-class*" is an object in R constructed following ASPRS LAS specifications [2] that helps define the properties and attributes of point cloud object in R and other software that read .las format files [3]. Therefore, the addition of attributes to the point cloud data set required updating the "*LASheader-class*". The `classify_ground` and `segment_trees` functions add attributes to the `data.table` of the "*LAS-class*" object in R and automatically update the "*LASheader-class*" of the point cloud object [3]. Manually adding the RF model classification and RF model class probabilities as attributes of each point did not automatically update the

"*LASheader-class*". The `add_lasattribute` function was used to update the "*LASheader-class*" with the RF classification attributes and export the classified point cloud as a ".las" file.

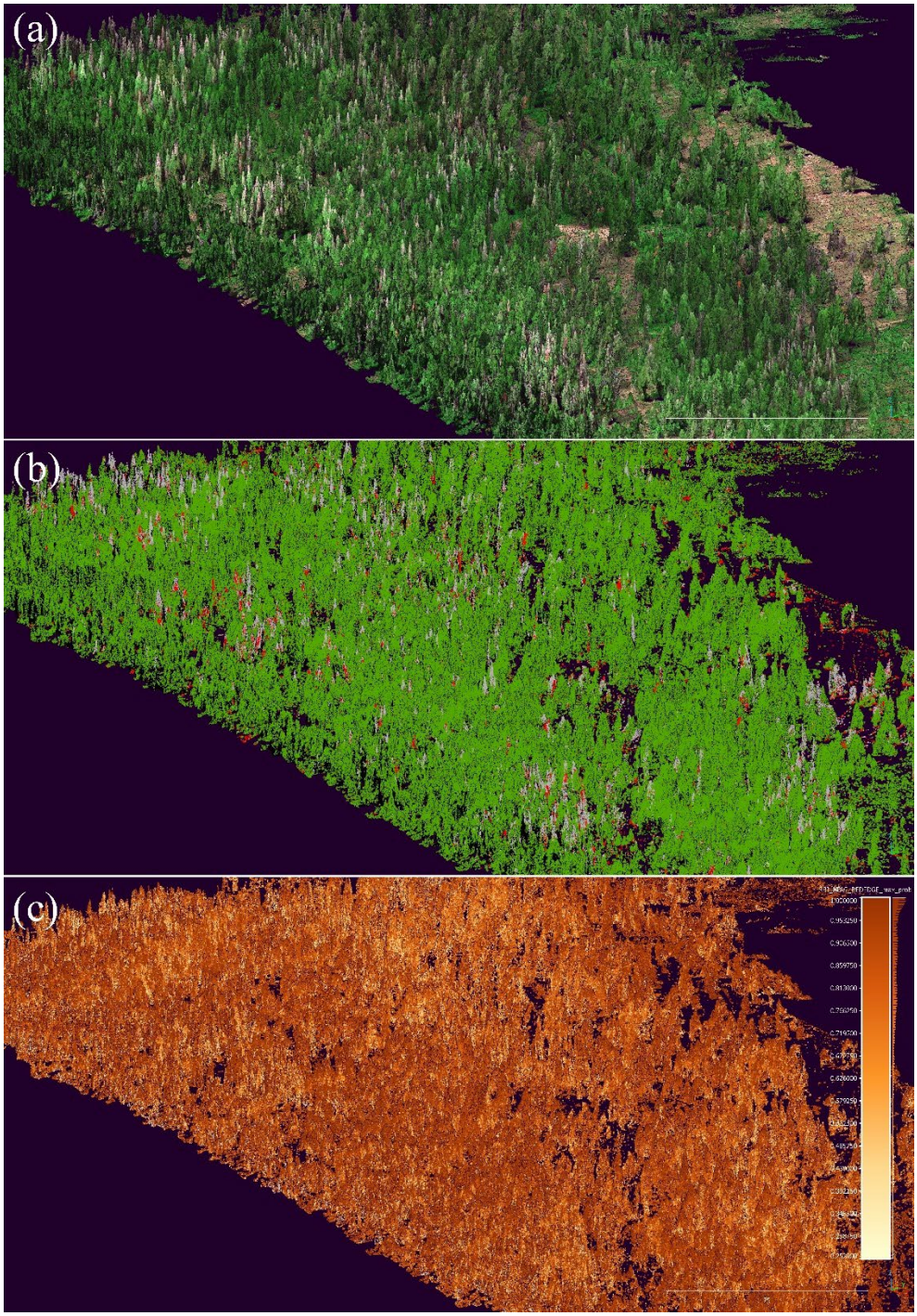

Figure S11. The RBI + NDVI + REDEDGE random forest (RF) model applied to the point cloud data set for the drone site M2. (a) True color rendering of the SfM point cloud for M2. (b) RF classification of SfM point cloud for M2; colors respond to respective classes, with black used for shadow. (c) Probability of each, classified by the RF model; darker brown colors indicate higher probabilities, lighter yellow colors indicate lower probabilities.

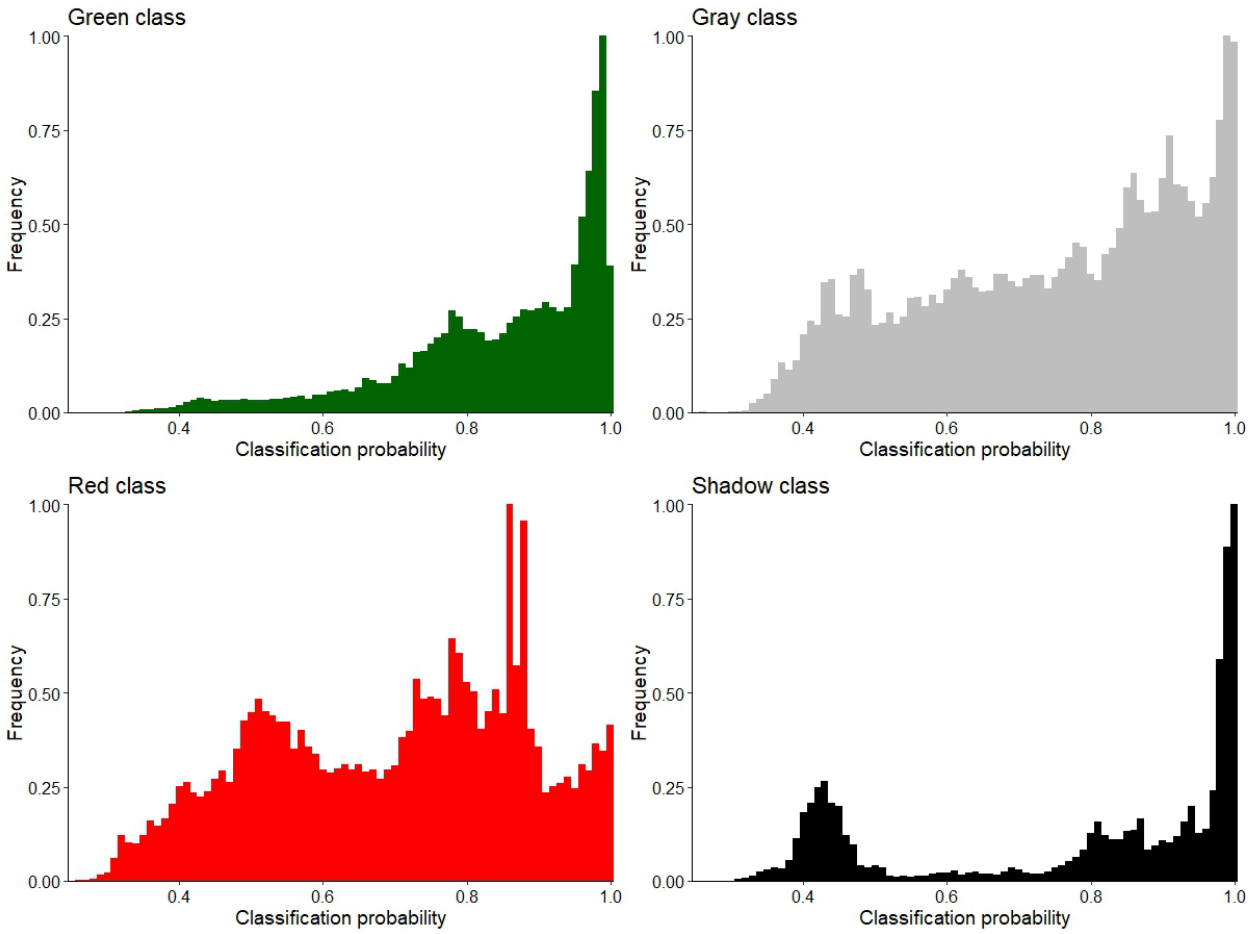

Figure S12. Distributions of classification probabilities of points classified as (a) “green” class, (b) “gray” class, (c) “red” class, and (d) “shadow” class. Frequencies are normalized by the maximum number of points per bin in each histogram to allow for a comparison among classes.

S8. Tree-level damage algorithm

S8.1. Definition of damage severities

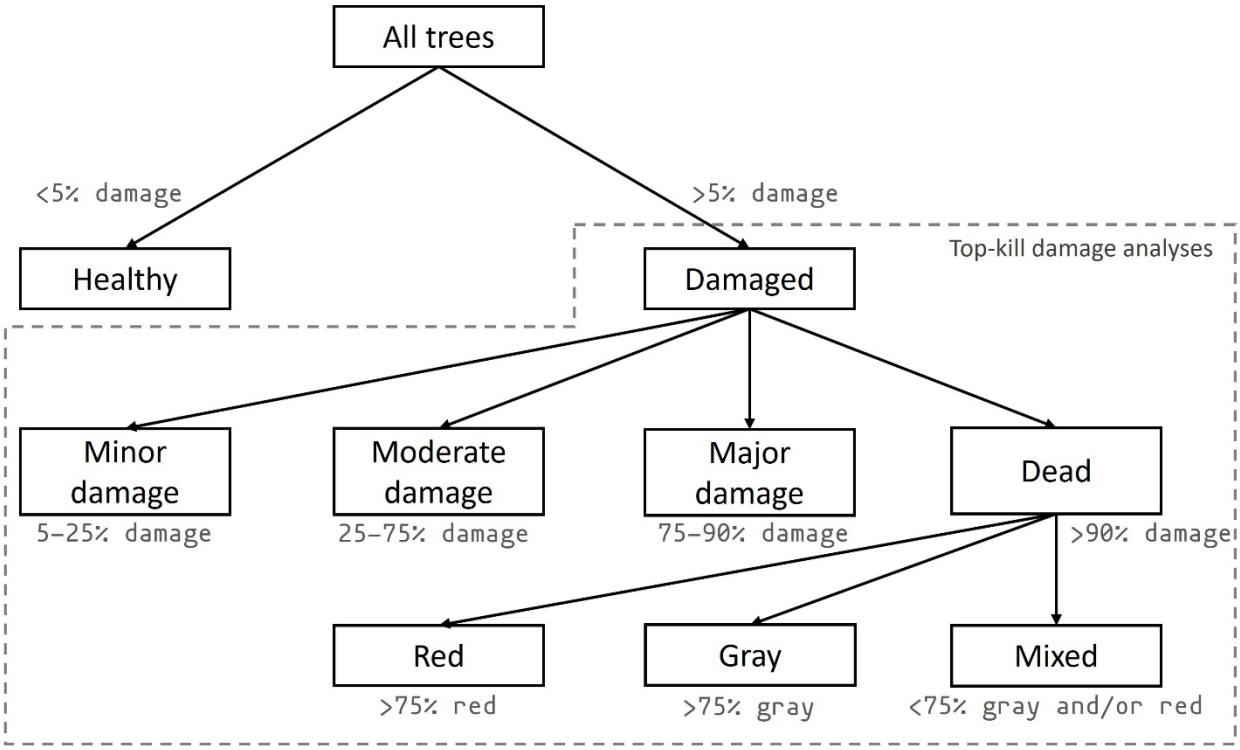

Figure S13. Classification tree used for the separation of trees into healthy, damaged, and different damage severities. "Damage" refers to the sum of red and gray points on each tree. The area with dashed lines indicates the damage severities for which the top-kill algorithms ("*top2bin*" and "*bin2bin*") were applied.

S8.2. Preparation of point cloud for damage assessment algorithm

The tree-segmented classified point cloud file was imported into R as a "*LAS-class*" R object to identify the damage severity of a tree using the `lidR` package [3]. The points classified as ground and shadows were not used in the damage assessment. The ground and shadow points were removed using the `filter_poi` function from the `lidR` package [3].

The `unique` function was used to create a list of unique `treeID` attributes from the `@data` slot of the "*LAS-class*" R object. This list of unique `treeID` attributes was used to iterate individual trees in the imported point cloud data set. The `filter_poi` function from the `lidR` package [3] was used to isolate points for individual segmented trees from the imported "*LAS-class*" R object by setting the "*treeID* ==" argument to the elements of the unique `treeID` list. The damage analysis only requires the height and classification information, and therefore, only the `Z` and classification columns were extracted from the filtered point cloud data set's `@data` slot and stored as a `data.table` for the tree-filtered point cloud data set. Performing a damage assessment on a `data.table` with only two columns is more efficient than using the "*LAS-class*" R object with multiple columns and data slots, especially for iterative operations.

S8.3. Types of tree-level damage algorithms

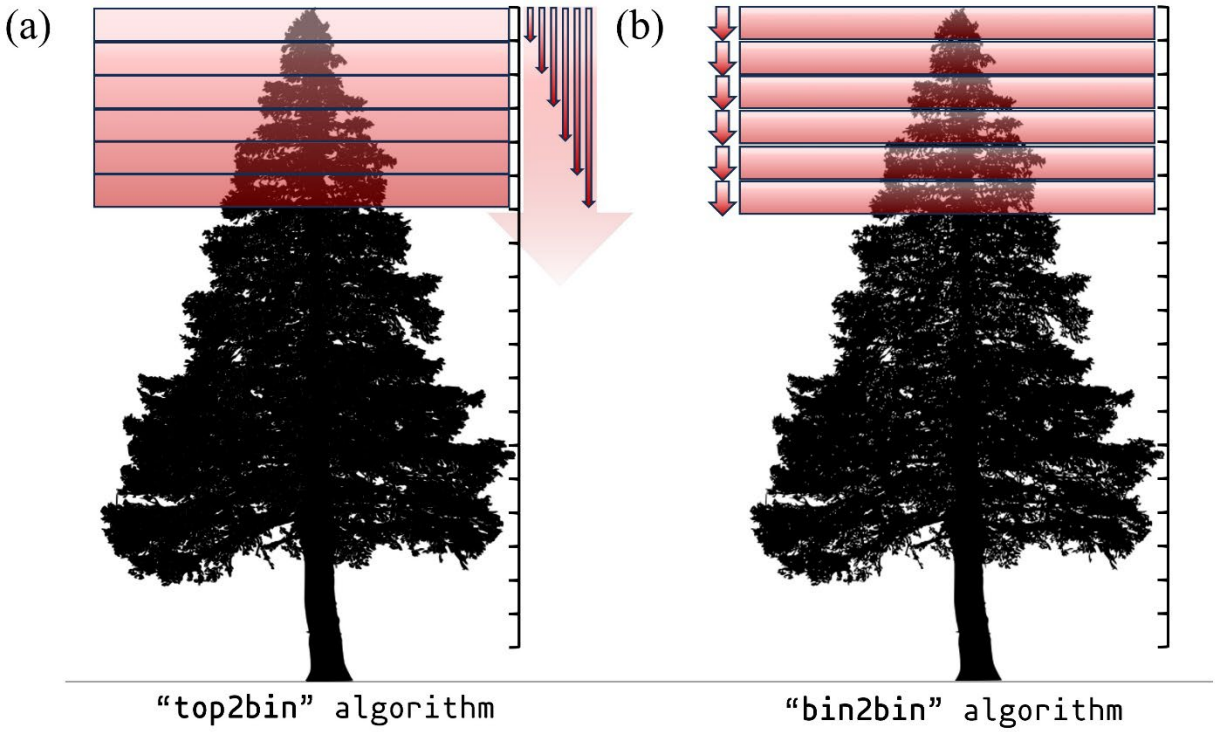

Figure S14. (a) The "*top2bin*" algorithm assesses the percentage of damage points within a set of height bins from the top of the tree to a given height; the algorithm begins at the top of the tree and progresses downward. For example, if a tree is 10 m tall, the "*top2bin*" algorithm assesses damage from 10 m above the ground to 9.75 m above ground (0.25 m bin) for the first set of bins (consisting of one 0.25-m bin), and then assesses damage from 10 m above ground to 9.5 m above ground for the second set (consisting of two 0.25-m bins). The algorithm halts when the percentage of damaged points within a set of bins is below a threshold. (b) The "*bin2bin*" algorithm assesses the percentage of damage points for each height bin beginning at the top of the tree and progressing downward. For example, if the tree is 10 m tall, the "*bin2bin*" algorithm assesses damage from 10 m above ground to 9.75 m above ground for the first bin, and then assesses damage from 9.75 m above ground to 9.5 m above ground for the second bin. The algorithm halts when the percentage of damaged points within a bin is below a threshold.

S8.4. Accuracy assessment of tree-level damage algorithm

The tree polygons with tree-level damage analysis information were imported into ArcGIS, and the polygon layer was clipped to a smaller extent to remove the edge artifacts of the MS and RGB orthomosaics.

The "*Create Accuracy Assessment Points*" tool in ArcGIS was used to create stratified random assessment reference locations (ArcGIS point feature) within the clipped extent. The input information source was set to the tree damage assessment polygon layer such that the randomized reference locations (ArcGIS point feature) inherited the damage severity identified by the algorithm under its "*Classified*" attribute. Randomized reference locations were evaluated with an on-screen assessment, in which each location's reference damage severity was recorded as its "*GrndTruth*" attribute. The MS and RGB orthomosaics were used as the primary source of visual assessment, and individual drone-captured images were consulted for cases that needed additional information. A total of 1000 randomized reference locations were qualitatively assessed to build the accuracy assessment reference data set.

The "*Compute Confusion Matrix*" tool in ArcGIS was used on the accuracy assessment locations to produce the confusion matrix and calculate the accuracy metrics. The confusion matrix from the accuracy assessment on ArcGIS consisted of substantial class imbalances, with most of the reference locations falling in the "*healthy*" class. To address this class imbalance, the attribute table of the accuracy assessment reference locations (ArcGIS point feature) was imported in R, and a bootstrap sampling method was implemented to produce an averaged confusion matrix and computed accuracy metrics [16].

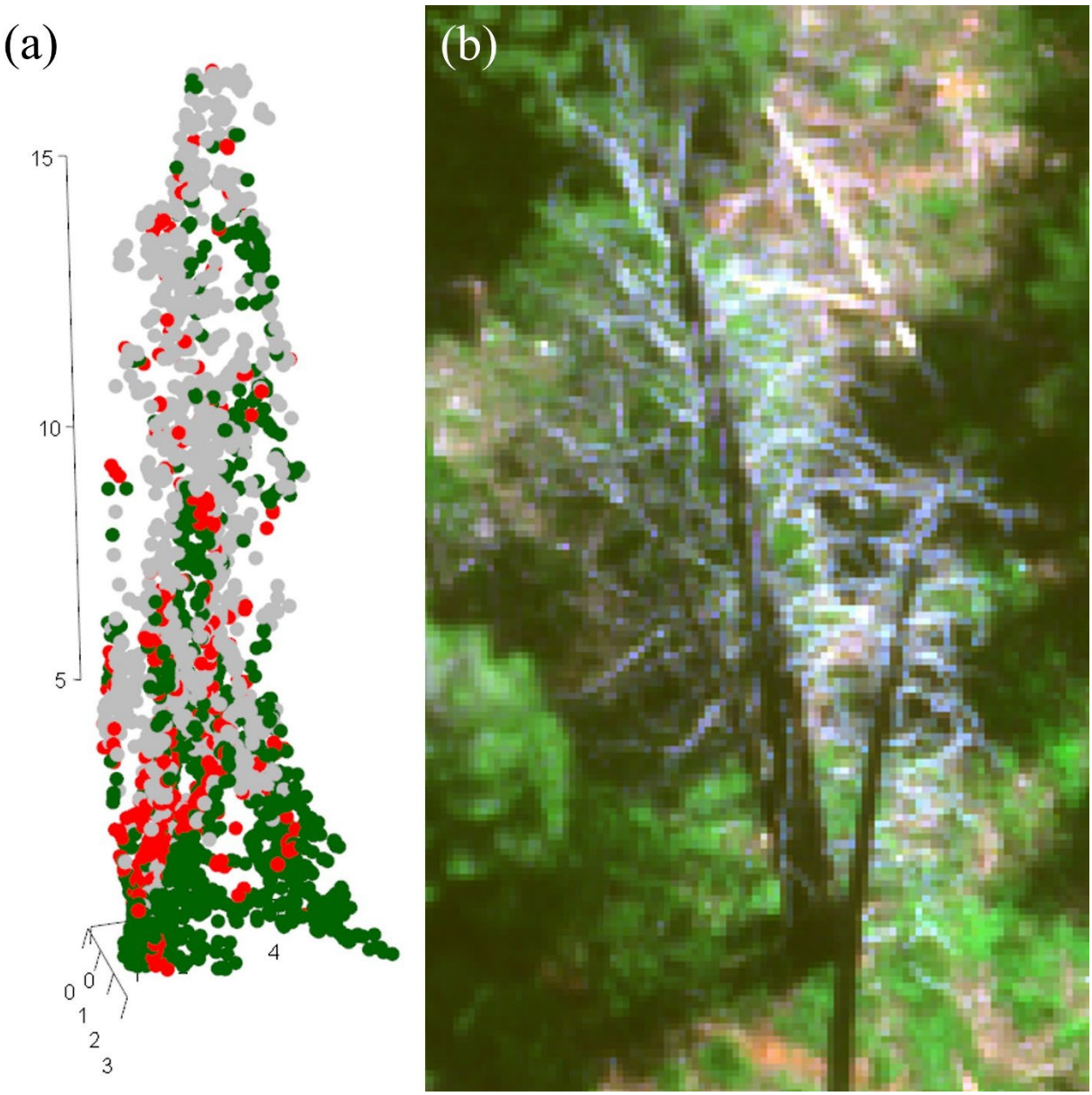

Figure S15. (a) Point cloud of a "*dead*" tree from the reference data set with green classified points on the bottom and middle. (b) Oblique view of the true color image of the "*dead*" tree in (a).

S8.5. Live crown base issue with SfM point clouds

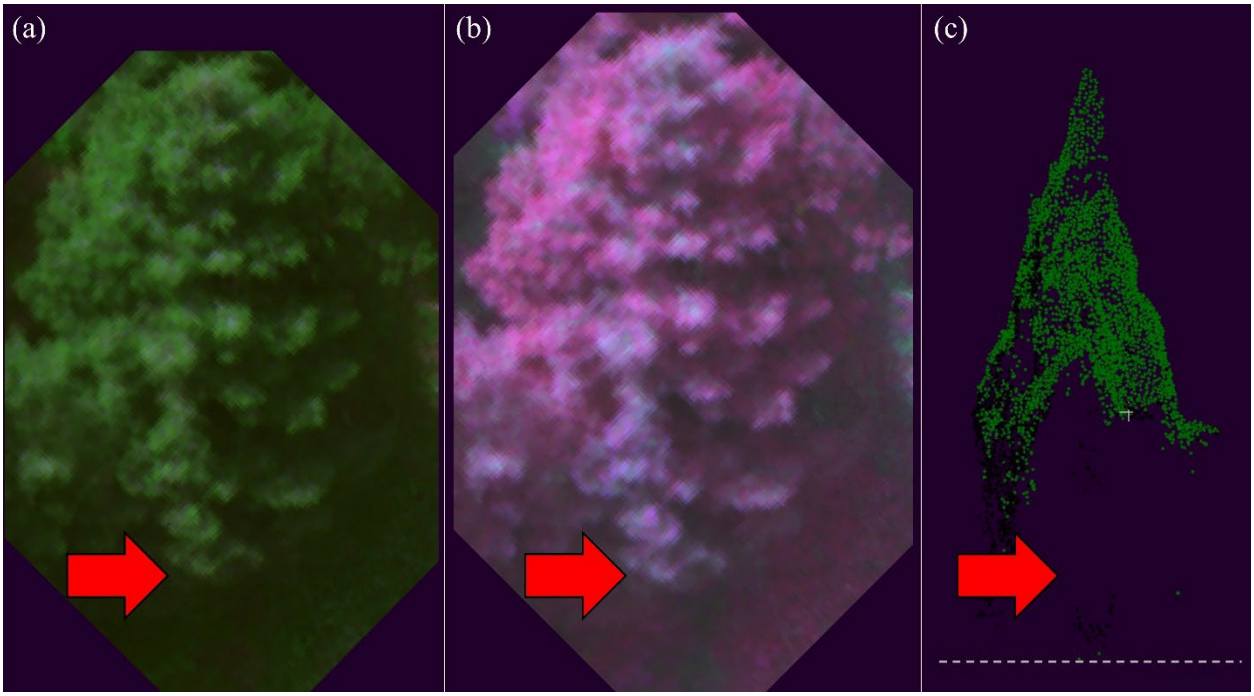

Figure S16. (a) True color and (b) false color drone images of an example tree with a live crown base above the ground (red arrows indicate live crown base). (c) SfM point cloud of the example tree in (a) and (b); red arrow on SfM point cloud shows absence of points, providing inconclusive information for live crown base estimation; gray dashed line represents the ground. The orientation of the point cloud in (c) was adjusted to match the view of the tree in (a) and (b). The placement of the red arrows was an estimation of where the points representing the crown base should be based on the orientation.

References

1. Agisoft Metashape Agisoft Metashape User Manual - Professional Edition, Version 2.0. *Copyright © 2023 Agisoft LLC* **2023**.

2. ASPRS LAS Specification 1.4-R15. *The American Society for Photogrammetry & Remote Sensing, Bethesda, MD* **2019**.

3. Roussel, J.-R.; Auty, D. Airborne LiDAR Data Manipulation and Visualization for Forestry Applications, **2023**, https://cran.r-project.org/package=lidR

4. Barrett, T.; Dowle, M.; Srinivasan, A. Data.Table: Extension of `data.Frame`. **2024**, https://r-datatable.com.

5. Mohan, M.; Leite, R.V.; Broadbent, E.N.; Jaafar, W.S.W.M.; Srinivasan, S.; Bajaj, S.; Corte, A.P.D.; Amaral, C.H. do; Gopan, G.; Saad, S.N.M.; et al. Individual Tree Detection Using UAV-Lidar and UAV-SfM Data: A Tutorial for Beginners. *Open Geosciences* **2021**, *13*, 1028–1039, doi:10.1515/geo-2020-0290.

6. Roussel, J.-R.; Auty, D.; Coops, N.C.; Tompalski, P.; Goodbody, T.R.H.; Meador, A.S.; Bourdon, J.-F.; Boissieu, F. de; Achim, A. lidR: An R Package for Analysis of Airborne Laser Scanning (ALS) Data. *Remote Sensing of Environment* **2020**, *251*, 112061, doi:10.1016/j.rse.2020.112061.

7. Hijmans, R.J. Terra: Spatial Data Analysis. **2024**, https://rspatial.org/.

8. Duarte, A.; Borralho, N.; Cabral, P.; Caetano, M. Recent Advances in Forest Insect Pests and Diseases Monitoring Using UAV-Based Data: A Systematic Review. *Forests* **2022**, *13*, 911, doi:10.3390/f13060911.

9. Ecke, S.; Dempewolf, J.; Frey, J.; Schwaller, A.; Endres, E.; Klemmt, H.-J.; Tiede, D.; Seifert, T. UAV-Based Forest Health Monitoring: A Systematic Review. *Remote Sensing* **2022**, *14*, 3205, doi:10.3390/rs14133205.

10. Guimarães, N.; Pádua, L.; Marques, P.; Silva, N.; Peres, E.; Sousa, J.J. Forestry Remote Sensing from Unmanned Aerial Vehicles: A Review Focusing on the Data, Processing and Potentialities. *Remote Sensing* **2020**, *12*, 1046, doi:10.3390/rs12061046.

11. Silva, C.A.; Hudak, A.T.; Vierling, L.A.; Loudermilk, E.L.; O'Brien, J.J.; Hiers, J.K.; Jack, S.B.; Gonzalez-Benecke, C.; Lee, H.; Falkowski, M.J.; et al. Imputation of Individual Longleaf Pine (Pinus Palustris Mill.) Tree Attributes from Field and LiDAR Data. *Canadian Journal of Remote Sensing* **2016**, *42*, 554–573, doi:10.1080/07038992.2016.1196582.

12. Li, W.; Guo, Q.; Jakubowski, M.K.; Kelly, M. A New Method for Segmenting Individual Trees from the Lidar Point Cloud. *Photogrammetric Engineering & Remote Sensing* **2012**, *78*, 75–84, doi:10.14358/PERS.78.1.75.

13. Legendre, P.; Legendre, L. Numerical Ecology: Developments in Environmental Modelling. *Developments in Environmental Modelling* **1998**, *20*.

14. Schloerke, B.; Cook, D.; Larmarange, J.; Briatte, F.; Marbach, M.; Thoen, E.; Elberg, A.; Toomet, O.; Crowley, J.; Hofmann, H.; et al. GGally: Extension to "Ggplot2" 2023.

15. Cutler, F. original by L.B. and A.; Wiener, R. port by A.L. and M. randomForest: Breiman and Cutler's Random Forests for Classification and Regression 2022.

16. Hardin, P.J.; Shumway, J.M. Statistical Significance and Normalized Confusion Matrices. *Photogrammetric engineering and remote sensing* **1997**, *63*, 735–739.