

PointcloudITD: A software package for individual tree detection and counting

Introduction and Background

The aim of this unit of work was to develop new algorithms and software routines for performing individual tree detection and counting from aerially-acquired 3D pointcloud datasets over plantation forests. This section describes the development of a software package “PointcloudITD” which can be used to detect and map the locations of individual trees, using a machine learning approach to classifying tree locations.

During a previous FWPA-funded research project (FWPA PNC305-1213) a new methodology for individual tree detection was developed by researchers at the Department of Primary Industries, NSW (Forest Science) [Kathuria et al., 2016]. The approach used a two-stage process consisting of maxima-detection and crown classification using a logistical regression model built using hand segmented LiDAR pointclouds. The algorithms were implemented as a series of R scripts which were effective at detecting trees over small areas, but suffered from poor computational performance when applied to larger multi-hectare datasets. The work performed in this current project began as an effort to re-implement these algorithms in software to provide improved computational speed and hence the ability to process larger LiDAR datasets (covering a whole stand or compartment).

Overview of PointcloudITD Software

PointcloudITD builds upon the approach of [Kathuria et al., 2016] by using a two-stage process to detect and map tree locations from pointclouds. During the first stage of processing, candidate tree crown locations are identified in pointclouds by finding local maximas (peaks) in the data that form a set of potential crown/tree locations. The second stage of the process uses a classification model, which is built internally by the software using machine learning and user-provided maps of a small number of tree crowns within plots, to further refine the list of potential candidates into a definitive list of classified crowns. The software is primarily designed to be used with LiDAR measured pointclouds, but can also be used with pointclouds derived from other sources, such as from aerial photogrammetry. The software provides a basic Graphical User Interface (GUI) that allows the user to import pointcloud files in LAS format (support provided by the libLAS library (<http://www.liblas.org/>)), import plot-based stem maps, configure and run different processing steps and export tree maps in a variety of formats including csv, text files and shape (.shp) files.

The workflow model for the software allows a user to import pointcloud data in LAS format, generate canopy normalised points and rasterised canopy height models (or use existing classification built into the pointcloud), determine local maxima points and further distinguish actual crowns/stem locations from this set. In order to produce a classification model that interprets pointcloud data into tree crown labels, the software also provides the ability to load field acquired stem data (from DGPS surveys or other GIS sources), and use this data to develop a stem classifier that can be exported and used on any other lidar dataset. A typical workflow for using PointcloudITD to produce a tree map from a pointcloud dataset consists of the following steps:

- Create a new project and import pointcloud datasets/tiles into the project
- Generate height normalised pointclouds of the forest canopy and rasterised canopy height models in geotiff format
- Detect and map local maximas in the canopy pointcloud
- Extract focal statistics for each local maxima that correspond to features of the local points in the vicinity of each maxima, which are later used to help classify maximas into true crowns
- Load user-provided stem maps and associated plot boundary data that is used by the software to extract training examples of actual stem locations
- Build a classification model from the training examples
- Deploy the training model to the entire pointcloud to detect tree locations, or export this classification model for use across other pointcloud datasets

PointcloudITD is implemented in a combination of the C++ and python programming languages and is designed to work efficiently with large datasets, by processing multiple LiDAR tiles in parallel across the available processing cores on the machine used for running the software. For machine learning, PointcloudITD uses the open-source machine learning software tool kit “scikit-learn” (<http://scikit-learn.org/>) under-the-hood, providing an interface to a variety of different classification algorithms.

Algorithms and Processing Steps

The following sections describe algorithms used within the different processing steps performed by PointcloudITD.

- *Pointcloud normalisation*

Most steps in the processing workflow use points that are normalised to the terrain height, where the z-axis of points represents the canopy height above ground, rather than sea-level elevation or absolute height. The software tool provides the option (a) for the user to specify that the pointclouds being imported are already canopy height normalised points (if, for example, pre-processed using other software tools such as LAStools) or (b) the z-axis of points corresponds to absolute height. Where points are provided with absolute height (option (b)), the software provides the user with a processing option for producing normalised canopy height points as a pre-processing step.

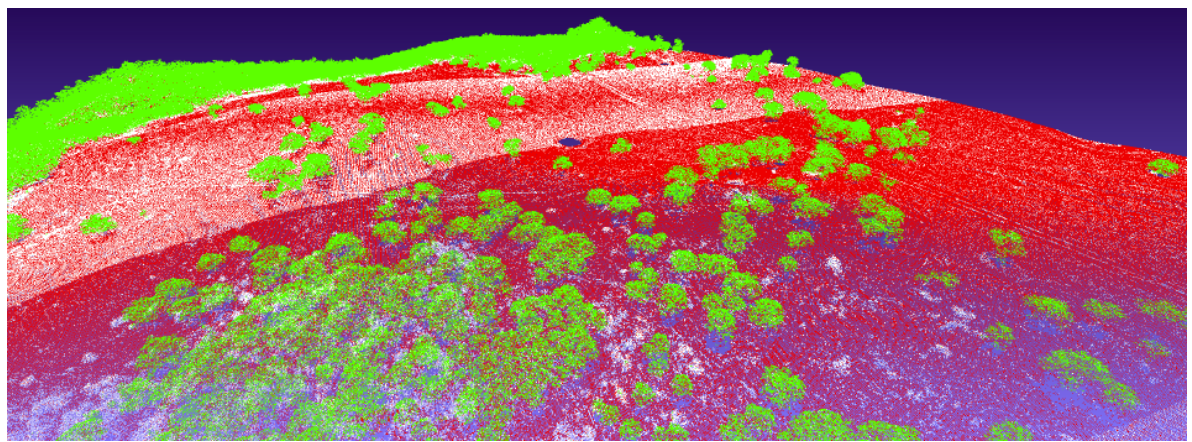
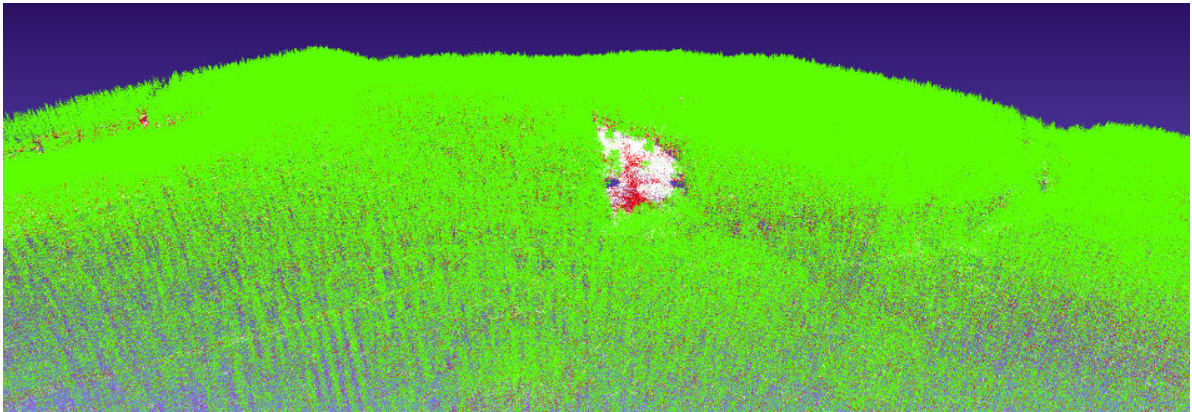


Figure 1. Colour segmented pointclouds (green: canopy, red/white: ground/non-canopy points) using the “Generate Normalised Canopy Points” pre-processing option.

The processing step allows the user to specify whether the pointcloud data contains pre-assigned classifications (such as ground points) that can be used to segment and subtract canopy points from the ground, or if no such classification data is available, the software uses an internal minima-finding strategy to segment ground-points and generate the canopy height model itself. This minima finding strategy is controlled by two parameters, which are user-configurable. The first parameter “LM radius 1” is the minimum distance between local minima points used to represent the ground and the second parameter “LM radius 2” is the maximum horizontal search radius which is used to associate points to their closest ground point, then used for ground height subtraction. Finally, PointcloudITD removes canopy points that have a height above ground smaller than a user selectable threshold.

Once pointcloud normalisation has been performed, PointcloudITD exports separate pointclouds corresponding to the located ground, actual canopy and the original points colour-coded by point type, each as separate files in ply format. Figure 1 illustrates two example colour-segmented pointclouds where green points correspond to canopy, and red/white points correspond to the ground plane/non-canopy objects.

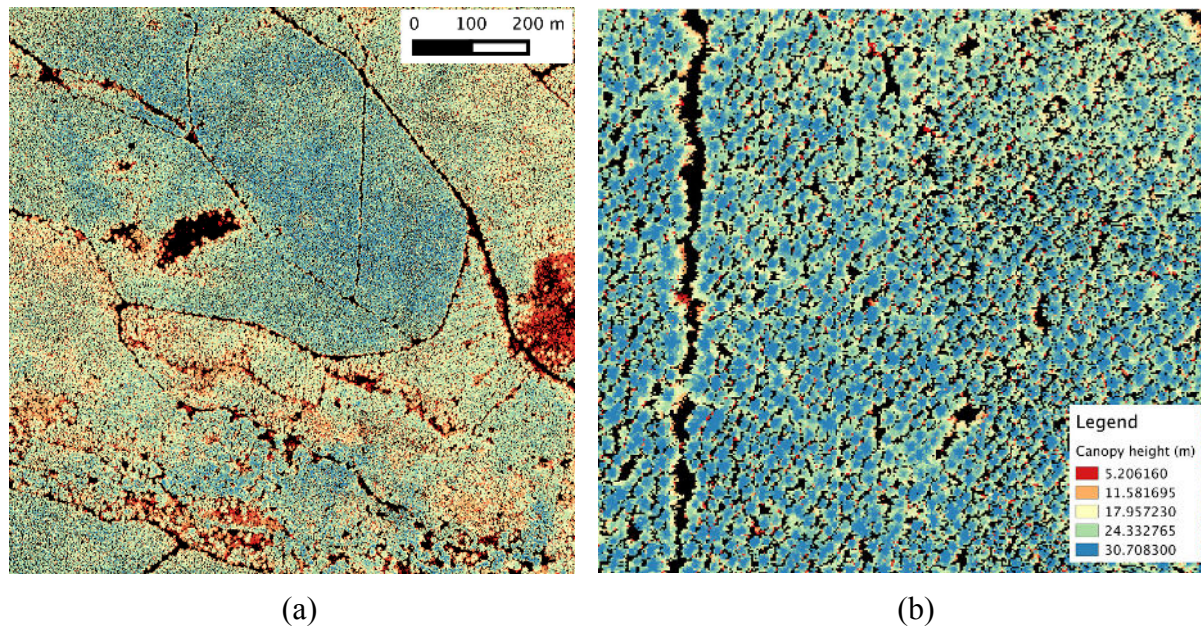


Figure 2. Rasterised canopy height models generated using the “Generate Canopy Height Model (raster)” pre-processing option and viewed in QGIS (an open-source GIS tool). The colours represent canopy height, in this case processed from the LiDAR pointclouds at a 50 cm grid resolution (user selectable).

The canopy height normalised points are then stored in the corresponding project directory and are used for further steps in the processing chain. The software additionally provides the option to generate and export rasterised canopy height models in floating-point geotiff format that are viewable in GIS software (see Figure 2).

- *Local maxima finding*

Once canopy height points are available, a local-maxima finding strategy is used to produce a candidate list of tree-crowns (Figure 3). The user specifies (or uses pre-configured options for) the minimum distance between points (default 2 m) and the software determines the list of all canopy points that are the tallest point within their surrounding area of the radius specified. The algorithm additionally assigns to each local maxima a value representing the maximal radius within which the point continues to have the maximum canopy height, up to a user specified distance (or default value of 15 m). Internally, PointcloudITD uses a grid-based fixed radius near neighbours search (with computational time scaling linear with the number of points) [Bentley et al., 1977] to identify local maxima points. Detected local maxima are exported into several formats including csv, GIS .shp files and as a separate pointcloud file.

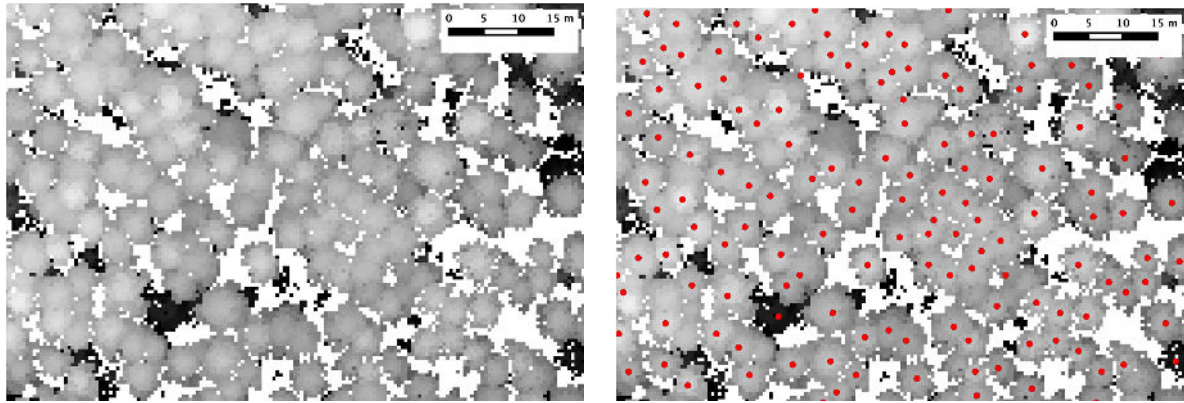


Figure 3. Example detected local maximas (candidate tree-crowns) overlaid with rasterised canopy height model data.

- *Feature/focal statistic calculation*

Later stages in the processing workflow use classification algorithms for selecting tree crowns from the candidate set of local maximas. This process relies on information that can be used to reliably distinguish the actual peak of a crown from other local maxima points that are not the true peak point of the tree. This information is potentially contained in the shape, distribution and density of the pointcloud surrounding the local maxima that makes up the structure of the canopy and stem. In order to provide a summative, quantitative description of this information, suitable for use as a decision/classification variable, PointcloudITD extracts summative statistics and other descriptors from all pointcloud points surrounding each detected local maxima. These sets of values are referred to as “focal statistics” or “features”. During this processing stage, feature variables are calculated for each local maxima using all pointcloud canopy points within a user-selectable radius around the point (default value of 15 m). These feature values for each local maxima are exported in csv and other binary formats, and later used during the classification/refinement process.

Table 1 provides a list of all focal statistics/features calculated by PointcloudITD and made available to the user for later classification stages in the processing workflow. Most focal statistics are calculated based on parameters associated with the local distribution of points in the pointcloud, with a subset of features being calculated based on the local raster information of the canopy height model, corresponding to the surface at the top of the canopy.

Variable	Source	Description
Height	Points	Peak height above ground
Radius still Maxima	Points	Maximum radius to which point is still the maxima
Point Count	Points	Number of points
Hor. Dist. to Tallest	Points	Horizontal XY distance to tallest point
Vert. Dist. to Tallest	Points	Vertical distance to tallest point
Height Rank	Points	Number of points taller than this point
Max. Height	Points	Height of tallest point
Min Height	Points	Height of lowest point
Mean Height	Points	Mean of all point height values
Height Median	Points	Median height value
Height Variance	Points	Variance of point heights
Height Std. Dev.	Points	Standard Deviation of height values
Mean above Mean	Points	Mean height of all points higher than mean
Height Skew	Points	Skew of height distribution
Height Kurtosis	Points	Kurtosis of height distribution
Height Range	Points	Maximum minus minimum height
Height Rel. Range	Points	Height range divided by height mean
Height 10%	Points	10 th percentile height
Height 20%	Points	20 th percentile height
Height 30%	Points	30 th percentile height
Height 40%	Points	40 th percentile height
Height 50%	Points	50 th percentile height
Height 60%	Points	60 th percentile height
Height 70%	Points	70 th percentile height
Height 80%	Points	80 th percentile height
Height 90%	Points	90 th percentile height
RasterCHM std. dev.	CHM Raster	Standard deviation of heights from CHM raster
RasterCHM Gradient Mag.	CHM Raster	Average magnitude of the gradient of CHM raster
RasterCHM Gradient Dir.	CHM Raster	Average direction of the gradient of CHM raster

Table 1. Focal statistics and features calculated for each local maxima and used as variables for the classification/refinement processing step.

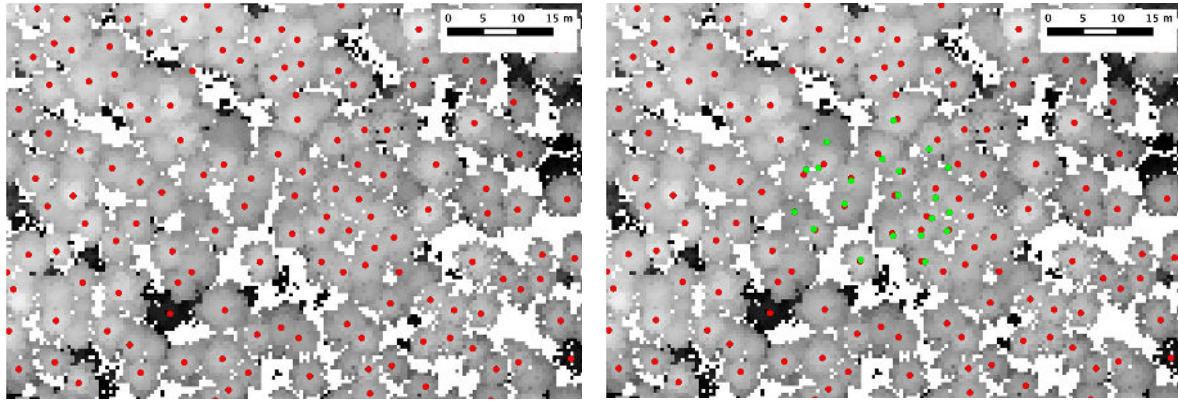


Figure 4. Comparison of detected local maxima and overlaid plot data stem map attained using DGPS measurements in the field. The software makes associations between these points as training instances of actual crowns/stems which are used to train the classification model.

- *Training data association*

Training stem locations are loaded as csv files containing a list of 2D tree stem locations mapped in the field and a corresponding plot number/id. Data on the plot boundaries must also be provided; the software allows for the user to specify plot centers, shape (circular or square) and the dimensions of each plot. The software then takes the plot stem maps and associates detected local maximas using a nearest neighbours association algorithms (with user-definable maximum association distance). All associated local maximas become positive training examples (for the location of a stem/crown) which are later used in classification model training. The remaining un-associated local maximas that lie within the specified sampling plot boundaries become negative training examples (detected maximas, that according to the plot data, do not correspond to actual stems/crowns). PointcloudITD exports crown positions used as positive and negative training examples in .shp format (Figure 4).

- *Classification model training and tree detection*

Once the training data has been loaded, the user can build a stem classification model. The software takes both positive and negative training examples and builds a discriminative classifier based on the associated feature vectors/focal statistics of each positive/negative crown. This stage in the software interfaces to Scikit-learn which provides an interface to a number of user-selectable classification algorithms (a Support Vector Machine (SVM) classifier [Cortes and Vapnik, 1995] is used by default). This stage of the processing performs automatic cross-validation to optimise classifier parameters and provide an estimate of the classification accuracy. The resulting classification model can take as an input the feature vector/pointcloud focal statistics for any other detected local maximas and determine whether this is likely to be a real crown or not. The classification model itself is also exported to a portable file that may be loaded into other projects with other pointcloud sources, lacking plot training data.

Once a stem classifier has been trained, or imported into the current project from an external source, or different project, the software uses the model to refine the list of detected local maximas in the project into a list of detected tree crowns. The resulting list of crown locations, and corresponding canopy heights are then exported in a range of formats including csv files

and GIS .shp files. The resulting output files represent estimated stem locations, from which further analysis can be performed to evaluate tree density and stocking.

PointcloudITD Software Results

Processing Step	Processing time (sec)
Canopy Height Model/Point Normalisation	35.5
Maxima Finding (2.0-15.0 m)	3.5
Focal Statistics Calculation	33.0
Classify Stems	0.8
Total:	72.8

Table 2. Current software processing times per 100 hectares for major steps in processing chain (2.9GHz Intel Core i7 dual-core processor).

- *Computational performance of software*

PointcloudITD was run on a dual-core laptop computer using a pointcloud dataset with approximately 13.2 million points over a 100 hectare area containing approximately 40,000 stems as a benchmark for assessing the computational performance. The software takes approximately 1 minute and 12 seconds for complete processing with most time spent producing a canopy height model from the raw pointcloud data and during calculation of focal statistics used for training and classification (Table 2). These results indicate that a typical forest resource unit of 10,000 hectares could be processed in approximately 1 hour using a two core computer (typical laptop computer) or 18 minutes on a eight core computer (typical high-powered desktop computer).

- *Tree detection accuracy*

Using LiDAR data acquired by HQPlantations over a 47 ha, 1982 Age Class compartment in a *Pinus caribea* var. *Honduras* plantation, the accuracy of tree detection/stem counting from PointcloudITD was assessed. Reference data consisting of stem maps of twenty 0.06 ha plots surveyed using a Trimble GEO7X with an attached Rangefinder were used for classification training and validation. Stem maps for ten randomly selected plots were imported into the software and used to build a classification model, which was then applied to the dataset for the entire 47 ha area. Three focal statistics were used for training; the crown height, “radius still maxima” and the vertical distance to the tallest point.

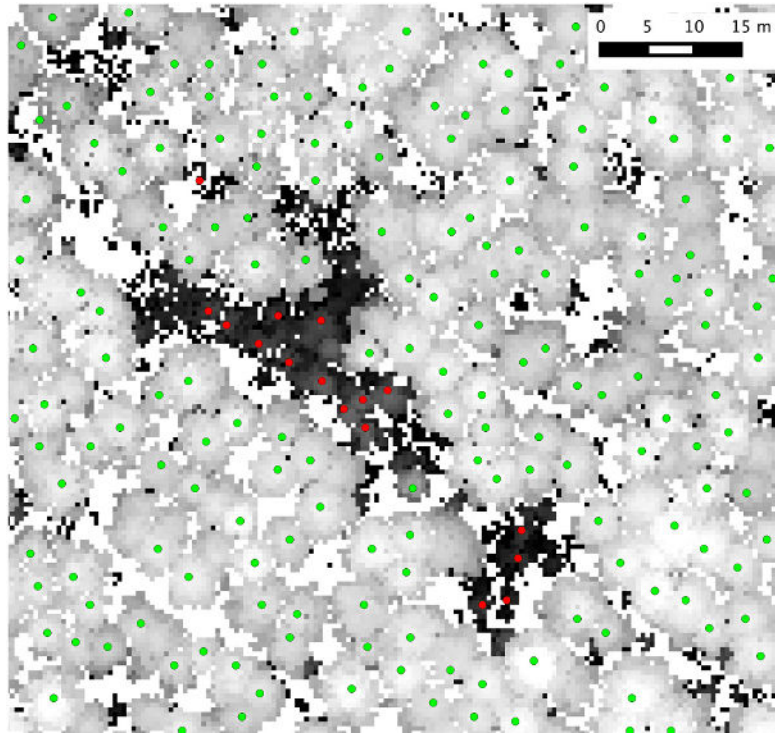


Figure 5. Refinement of detected local maxima into classified tree crowns: all points shown are detected local maxima. Green points are those maxima that have been positively classified as tree crowns based on the machine learning stage, whereas red points are local maximas that have been disregarded based on the classifier output.

Figure 5 illustrates detected tree crowns (in green) overlaid with rasterised canopy height model data. Shown in red and detected local maximas that have been rejected as tree crowns during the classification/refinement stage of processing in the software. During the machine learning phase, positive training examples are used reinforce a classifier model that automatically eliminates false detections based on, in this instance, canopy height (as a feature). Figure 6 shows a comparison between the actual number of stems per 0.06 ha plot (from field-based data) with the detection counts using the classified stem map. Stem detection accuracy was 85% when comparing the sum of true positive and true negative detections to the total number of surveyed stems across the ten plots. The resulting count of detected trees in the validation plots had a bias of 1.7% and RMSE of 7.25%.

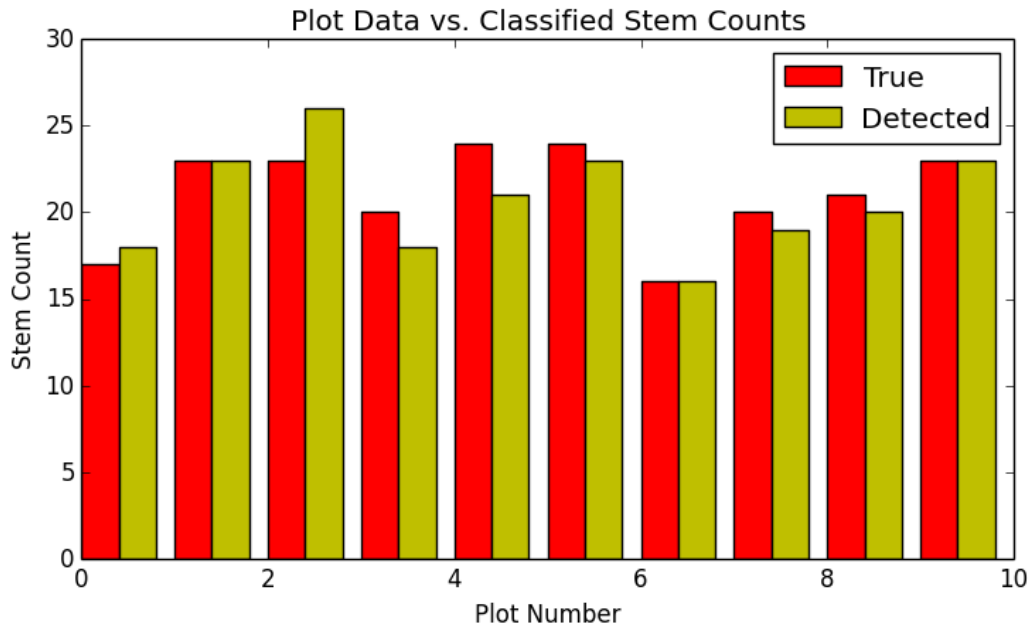


Figure 6. Comparison of number of true and detected stems in 10 validation plots (0.06 ha) using the software tool, where the stem classification model was trained using stem map data from another 10 training plots.

Reference Guide for the operation of PointcloudITD

This section provides more details on the software interface and operation of PointcloudITD including instructions for using the software during a typical workflow.

- *Installation: Windows*

Unzip the software zip file to any location on your machine. Run 'pointcloudITD_GUI.exe' by double-clicking to launch the GUI.

- *Other Suggested Free/Open Source Software*

Pointcloud ITD does not provide functionality for visualisation of results during the different processing steps, however outputs are provided in a project data directory in common file formats which can be accessed by the user. There are a number of suggested, open-source or free cross-platform tools which can be used for visualising results and partial results during the use of PointcloudITD:

- CloudCompare (<http://www.danielgm.net/cc/>): Pointcloud viewer, is compatible with .las and .ply file formats
- Meshlab (<http://www.meshlab.net>): An alternative pointcloud and 3D geometry viewer, compatible with ply
- QGIS (<http://www.qgis.org>): A free Geographical Information System (GIS) environment, capable of viewing, editing and annotating a geodata in a variety of vector and raster formats, including the geotif images and shp files produced by PointcloudITD

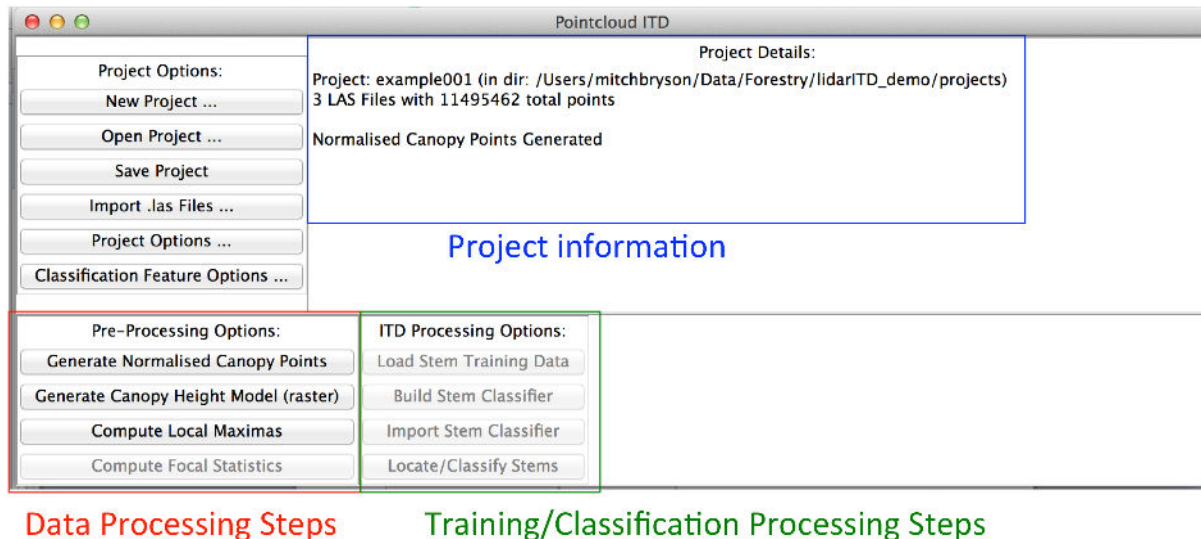


Figure 7. PointcloudITD Software Main Project Window

- *PointcloudITD Main Window*

The software uses a simple graphical interface to perform steps in the processing chain. The main software window (Figure 7) includes options for starting, loading and saving projects, editing processing options and performing processing steps. The user creates a project file, which stores information about the state of the processing and creates a corresponding directory where all processed outputs are saved. Once a project is started, pointcloud files can be imported to the project in LAS format. Multiple pointcloud tiles can be associated with a single project.

- *Starting a New Project*

After opening the software, click on “New Project” in the main window. Use the file dialog to select a location to save the project and all of its working files. Ensure that the project name contains no spaces. PointcloudITD will create a project file and directory at the selected location and name. The project file uses the extension “.itd” and contains information about the state of processing, sources to the original pointcloud data and saved project options. The project file uses an xml format. The project creates a directory in the same location as the project file that has the name “<projectname>_itdfiles”. The outputs of all processing are stored in this directory and accessible to the user. At any time, the project may be saved by pressing the “Save Project” button on the main window. This saves information to the “.itd” project file and should be performed before exiting the software.

- *PointcloudITD project data directory structure*

Data within the project data directory is ordered according to the different processing steps. The sub-directories within the project data directory are:

- “**chm_points**”: contains outputs from “Generate Normalised Canopy Points”
- “**chm_raster**”: contains outputs from “Generate Canopy Height Model (raster)”
- “**localmaximas**”: contains outputs from “Compute Local Maximas”
- “**focalstats**”: Contains outputs from “Compute Focal Statistics”

- “**classifier**”: Contains outputs from “Load Stem Training Data”, “Build Stem Classifier” and “Locate/Classify Stems”.
- *Opening an Existing Project*

To open an existing project, click on the “Open Project” button. Use the file dialog to navigation to the location of the project file and data directory and double-click on the project’s “.itd” file. Once a project is loaded, information about the current state of processing is provided in the project details tab including the number of pointcloud files, total number of points, number of detected local maxima and number of classified crowns.

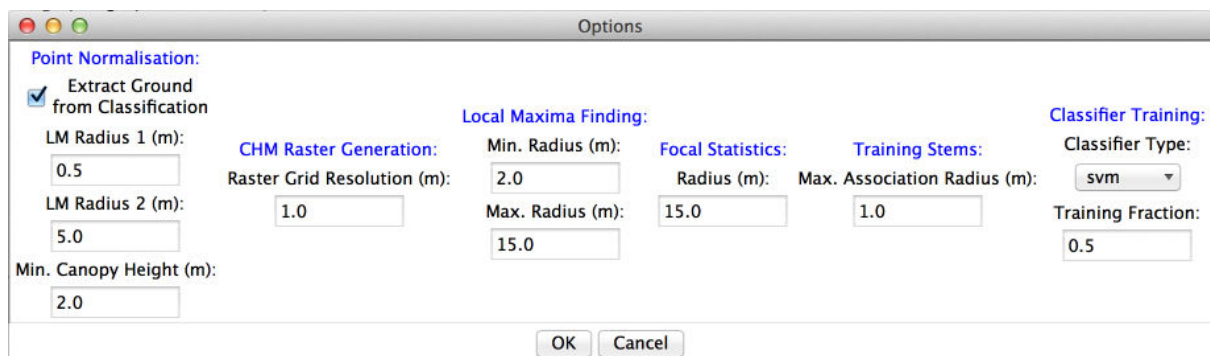


Figure 8. PointcloudITD project options window

- *Project Options*

To configure the options for the project, click on the “Project Options” button on the main window. This opens a separate project options window (Figure 8) that allows the user to configure options associated with the different processing steps. Options are grouped according to the blue text above each group of options:

- **Point Normalisation Options:**
 - **Extract Ground from Classification:** If this option is checked, PointcloudITD will use the classification data contained within the LAS file to locate ground points when normalising points. If unchecked, PointcloudITD will try to find ground points using a local minima finding strategy.
 - **LM Radius 1:** When the “extract ground from classification” option above is checked, this parameter represents the minimum distance between detected local minima/ground points.
 - **LM Radius 2:** When the “extract ground from classification” option above is checked, this parameter represents the maximum distance to which the algorithms searches for a nearby ground point when normalising each canopy point
 - **Minimum Canopy Height:** All canopy points with a height below this threshold are removed from the final normalised canopy pointcloud.
- **CHM Raster Generation:**
 - **Raster Grid Resolution:** Grid resolution to be used when producing rasterised CHM is tif format. These raster files are used mainly for visualisation, and are not used during further processing steps.
- **Local Maxima Finding:**

- **Minimum Radius:** Minimum distance between detected local maxima points. The detected local maxima points are the set of all points for which there exists no taller point within a radius around the point of this value.
- **Maximum Radius:** Each local maxima is assigned a feature value “radius still maxima” which is computed during maxima detection. “Maximum Radius” is the maximum distance out to which the algorithms searches for taller points when calculating this value.
- **Focal Statistics:**
 - **Radius:** When calculating features/focal statistics for each local maxima, radius is the maximum distance to a point for it to be included in the calculations for a given local maxima.
- **Training Stems:**
 - **Maximum Association Radius:** When training points are loaded by the user, a nearest neighbour strategy is used to associate detected local maximas to mapped stems. This parameter represents the maximum distance between training stems and detected local maxima for an association to be made. Detected local maximas with no associations within this radius, but still within the boundaries of the sampled plot area are used as negative training examples (i.e. a detected local maxima that is definitely not a crown peak point/stem).
- **Classifier Training:**
 - **Classifier Type:** The type of classification algorithm used for the classification model. Only the “SVM” (Support Vector Machine) is available in this current version of the software.
 - **Training Fraction:** Fraction of stems to be used as a training set (the rest being used as the test set) for each randomised trial of cross-validation that occurs internally to the software in order to optimise classification parameters and provide an estimate of detection accuracy.

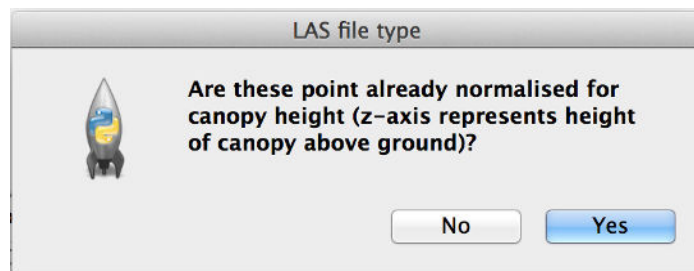


Figure 9. Importing pointclouds: option to import clouds in normalised coordinates

- *Importing pointclouds*

PointcloudITD allows for multiple pointcloud files to be imported into a single project file. PointcloudITD will process pointclouds across multiple files in parallel, reducing total computation time for datasets with large numbers of points. Click on “Import .las files” in the main project window and use the file dialog to select one or multiple .las format files to import into the project. Once files are imported in, the import button can be clicked additional times to load other files from different file system paths. LAS files imported into a project must be compatible with the LAS file format version 1.3 or below, as files coded to a higher version number of the las format are not compatible with liblas. When pointclouds are imported into a project, the user is prompted to indicate if the points have already been height

normalised (z-axis represents height above the ground). Select “Yes” if this is the case or select “No” for points with z-axis coordinate measured in absolute height. For points that are already in normalised form, the software copies the point data into the project and disables access the the first stage of processing “Generate Normalised Canopy Points”, which is no long necessary to perform.

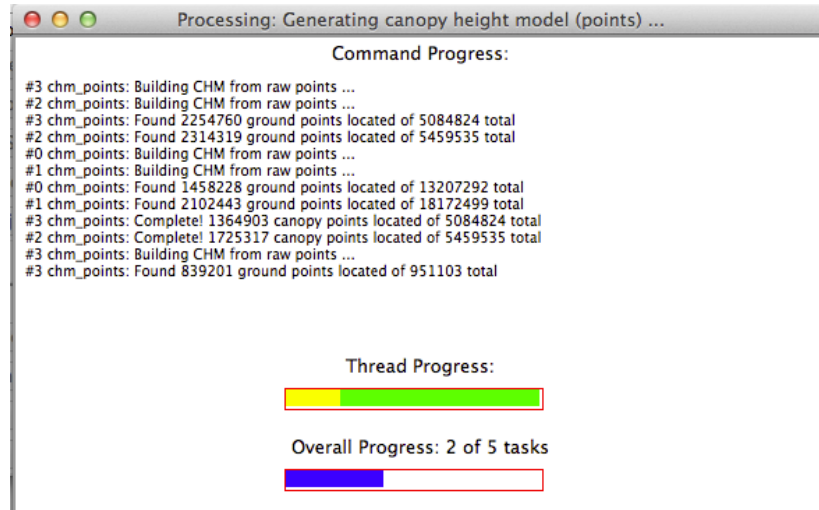


Figure 10. PointcloudITD Processing Progress Window

- *Processing Step: Generate Normalised Canopy Points*

Once pointclouds have been loaded into a project and options configured, the project should be saved by the user. The first stage of processing is to produce a normalised canopy height model for the pointcloud dataset (for datasets which were loaded with absolute z-axis height). Click on the “Generate Normalised Canopy Points” button under “Processing Steps” on the main project window. A processing progress window will open to inform the user of the progress in performing this processing step. Once the processing window is completed, the window will indicate the caption “finished”, at which point this progress window may be closed.

Within the project data directory “chm_points”, the project stores .ply format files of the resulting pointclouds for the canopy, ground points and all points, colour labelled by type. Files are named according to the filename of the original las file the data originates from.

- *Processing Progress Window*

During processing steps, the software provides a processing progress window that provides an overview of the total progress associated with the current processing stage (Figure 10). The software uses a multi-core parallel processing architecture to allocate processing tasks across available processors on a single machine in order to minimise processing/computation time. The window displays information on the overall progress of the current tasks and progress of individual processes. The window also provides a status console for detailed feedback on the status of each process. At anytime, the processing step can be cancelled by closing the progress window. The user will be prompted for confirmation before the processing halts and control returns to the main software window. When the processing for a given task has completed, the progress window will display the caption “finished”, after

which the user can close the progress window in order to move onto the next step of processing.

- *Processing Step: Generate Canopy Height Model (raster)*

PointcloudITD provides the ability to generate and export a rasterised canopy height model (surface of tallest points along the top of the canopy) in geotif format. Click on the “Generate Canopy Height Model (raster)” button on the main window to perform this step. Geotif files are stored in the project directory “chm_raster”. These files are mainly for visualisation purposes, and are not used during further processing steps.

- *Processing Step: Compute Local Maximias*

Once PointcloudITD either has access to previously loaded canopy normalised points, or these points have been generated in the previous processing step, click the “Compute Local Maximias” button under “processing steps” to compute local maxima points. Results are stored within the “localmaximas” sub-directory. The local maximias are saved in .csv, .txt and .ply formats for each separate pointcloud file loaded into the project and in .shp file for (“all_maximas.shp”) for all local maxima points within the whole project.

- *Processing Step: Compute Focal Statistics*

Once local maximias have been computed, click “Compute Focal Statistics” in order to calculate the values for all focal statistics/features for each local maxima. Values for all available variables are computed and saved in .csv format within the “focalstats” sub-directory of the project.

- *Processing Step: Load Stem Training Data*

In order for pointcloudITD to build a classification algorithm that distinguished tree stems/crown peak points from other local maximias in the pointcloud, training examples of points corresponding to both real crowns (positive examples) and example corresponding to local maximias that are not crown peaks (negative examples) are needed. To produce this data, pointcloudITD allows the user to upload data containing stem maps over plots from information collected in the field. PointcloudITD uses the local maximias that are the closest match to the stems within the provided training data as positive examples and all other unassociated local maximias (that fall within the specified plot boundaries) become negative examples.

Training plots should be loaded as one csv file per plot. The first row of the csv file should contain the plot type (either “circle” or “square”), the X (easting) coordinate of the plot center, the Y (northing) coordinate of the plot center and the plot dimension (radius of the plot for a circular plot or length of one edge of the plot for a square plot). Each subsequent row should have four values for each stem that was mapped within the plot: the stem number (1 to N), the X coordinate, Y coordinate and a zero value. Figure 11 illustrates an example training plot csv file (viewed in Microsoft excel and as a raw ascii file).

Once a training file is loaded, the number of positive/negative associations computed from the training data is displayed to the user.

	A	B	C	D
1	circle	460311.088	7185296.55	13.8
2	1	460299.895	7185291.32	0
3	2	460302.266	7185288.89	0
4	3	460304.412	7185284.15	0
5	4	460308.929	7185286.35	0
6	5	460310.171	7185293.47	0
7	6	460317.398	7185293.35	0
8	7	460319.826	7185292.62	0
9	8	460319.148	7185289.01	0
10	9	460322.197	7185288.21	0
11	10	460322.084	7185294.6	0
12	11	460316.495	7185301.09	0
13	12	460311.187	7185296.06	0
14	13	460308.646	7185298.77	0
15	14	460303	7185297.98	0
16	15	460299.443	7185299.79	0
17	16	460305.485	7185303.01	0
18	17	460306.727	7185305.94	0
19	18	460303.847	7185308.15	0
20	19	460312.542	7185310.46	0

(a)

```
circle,460311.0881,7185296.55,13.8
1,460299.8947,7185291.32,0
2,460302.2661,7185288.892,0
3,460304.4117,7185284.149,0
4,460308.9287,7185286.351,0
5,460310.1709,7185293.465,0
6,460317.3981,7185293.353,0
7,460319.8259,7185292.619,0
8,460319.1484,7185289.005,0
9,460322.1974,7185288.214,0
10,460322.0844,7185294.595,0
11,460316.4947,7185301.088,0
12,460311.1872,7185296.063,0
13,460308.6464,7185298.773,0
14,460303.0001,7185297.982,0
15,460299.443,7185299.789,0
16,460305.4845,7185303.008,0
17,460306.7266,7185305.944,0
18,460303.8471,7185308.146,0
19,460312.5423,7185310.461,0
```

(b)

Figure 11. Example stem map training data csv file (a) viewed in Microsoft Excel and (b) in raw ascii form

- *Project options and classification feature selection windows*

The classification feature options window provides the ability for the user to select which specific focal statistic values will be used during classification model training, in order to allow customisation to different datasets (Figure 12). These options should be selected prior to building a classification model. The default values include the height, radius still maxima, vertical distance to tallest point and raster CHM features. The full list of options is described in Table 1.

- *Processing Step: Build Stem Classifier*

Once training data has been loaded and the desired training features/focal statistics configures from the classification features options window, click on “Build Stem Classifier” to train a classification algorithm. A dialog window appears indicating the model is currently being trained, and this dialog automatically disappears when training is finished. The classification algorithm performs internal cross validation to estimate the detection accuracy of the classify (number of true positives and false negatives compared to total local maximas within each plot) and this accuracy is displayed in the project information window. The classifier can be re-generated using new features by opening the classification feature options window, choosing different features by clicking on/off on the selection boxes, clicking OK and clicking “Build Stem Classifier”. Variations in the cross-validation accuracy can provide an indication of the expected performance of the classifier, and can be used to help choose the best sets of classification features to use for a particular pointcloud dataset/ forest environment.

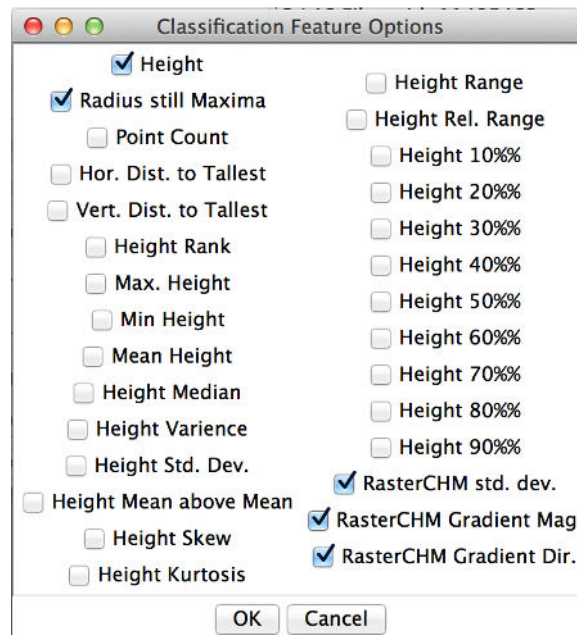


Figure 12. PointcloudITD classification feature options window

- *Importing a stem classifier from another project*

Once a classifier has been built, it can then be imported into other pointcloudITD projects and used to classify points. To import a classifier into a project, click on the “Import Stem Classifier” button on the main window. Use the file dialog to navigation to the location of the project from which the classifier will be imported and select the project’s “.itd” file and click open. Once loaded, the classifier can the be used and the project’s information window should indicate the cross validation performance of the classifier.

- *Processing Step: Locate/Classify Stems*

Once a stem classifier has been built or imported from another project, it may be used to classify the set of local minimas and create a set of refined tree crown points. Click on the “Locate/Classify Stems” button on the main project window. When the process has completed, the list of crowns is stored in “all_classified_crowns.csv” and “all_classified_crowns.shp” in the classifier sub-directory of the project data directory.

References

- [Bentley et al., 1977] J.L. Bentley, D.F. Stanat and E.H. Williams Jr., "The complexity of finding fixed-radius near neighbors", *Information Processing Letters*, 6(6):209-212, 1977.
- [Cortes and Vapnik, 1995] C. Cortes and V. Vapnik, "Support-vector networks", *Machine Learning*, 20-3:273–297, 1995.
- [Eysn et al., 2015] L. Eysn, M. Hollaus, E. Lindberg, F. Berger, J.M. Monnet, M. Dalponte, M. Kobal, M. Pellegrini, E. Lingua, D. Mongus and N. Pfeifer, "A benchmark of lidar-based single tree detection methods using heterogeneous forest data from the alpine space", *Forests*, 6, pp. 1721-1747, 2015.
- [Kathuria et al., 2016] A. Kathuria, R. Turner, C. Stone, J. Duque-Lazo and R. West, "Development of an automated individual tree detection model using point cloud LiDAR data for accurate tree counts in a *Pinus radiata* plantation", *Australian Forestry*, 79, 2016.