Three-dimensional Segmentation of Trees Through a Flexible Multi-Class Graph Cut Algorithm (MCGC)

Jonathan Williams, *Member, IEEE*, Carola-Bibiane Schönlieb, Tom Swinfield, Juheon Lee, Xiaohao Cai, Lan Qie, and David A. Coomes

***Abstract*—Developing a robust algorithm for automatic individual tree crown (ITC) detection from airborne laser scanning datasets is important for tracking the responses of trees to anthropogenic change. Such approaches allow the size, growth and mortality of individual trees to be measured, enabling forest carbon stocks and dynamics to be tracked and understood. Many algorithms exist for structurally simple forests including coniferous forests and plantations. Finding a robust solution for structurally complex, species-rich tropical forests remains a challenge; existing segmentation algorithms often perform less well than simple area-based approaches when estimating plot-level biomass. Here we describe a Multi-Class Graph Cut (MCGC) approach to tree crown delineation. This uses local three-dimensional geometry and density information, alongside knowledge of crown allometries, to segment individual tree crowns from airborne LiDAR point clouds. Our approach robustly identifies trees in the top and intermediate layers of the canopy, but cannot recognise small trees. From these three-dimensional crowns, we are able to measure individual tree biomass. Comparing these estimates to those from permanent inventory plots, our algorithm is able to produce robust estimates of hectare-scale carbon density, demonstrating the power of ITC approaches in monitoring forests. The flexibility of our method to add additional dimensions of information, such as spectral reflectance, make this approach an obvious avenue for future development and extension to other sources of three-dimensional data, such as structure from motion datasets.**

***Index Terms*—biomass, light detection and ranging (LiDAR), remote sensing, vegetation mapping**

Manuscript received March 20, 2019; revised July 22, 2019. This work was supported by the UK's Natural Environment Research Council (NERC) through a PhD studentship held by J. Williams which is a CASE partnership with Royal Society for the Protection of Birds (RSPB) [NE/N008952/1]. Work was further supported by NERC through a grant under the Human Modified Tropical Forests programme of NERC [NE/K016377/1] and through the NERC Airborne Remote Sensing Facility and NERC Data Analysis Node who collected and processed data (project code MA14-14). D. Coomes was supported by an International Academic Fellowship from the Leverhulme Trust. Inventory data was collected as part of an ERC Advanced Grant (T-Forces). X. Cai was supported by the Issac Newton Trust and Welcome Trust. C.-B. Schönlieb acknowledges support from the Leverhulme Trust project on Breaking the non-convexity barrier, EPSRC grant Nr. EP/M00483X/1, the EPSRC Centre Nr. EP/N014588/1, the RISE projects CHiPS and NoMADS, the Cantab Capital Institute for the Mathematics of Information and the Alan Turing Institute. We gratefully acknowledge the support of NVIDIA Corporation with the donation of a Quadro P6000 GPU used for this research. (J. Williams is the corresponding author: email jonvw28@gmail.com)

J. Williams, T. Swinfield and D. Coomes were with the Forest Ecology and Conservation Group, Department of Plant Sciences, University of Cambridge, CB2 3EA, UK which itself is also part of the University of Cambridge Conservation Research Institute (UCCRI), David Attenborough Building, Cambridge, CB2 3QY, UK.

J. Williams, C.-B. Schönlieb, J. Lee and X. Cao were with the Image Analysis Group, Department of Applied Mathematics and Theoretical Physics (DAMTP), University of Cambridge, CB3 0WA, UK.

T. Swinfield was additionally with the Centre for Conservation Science, Royal Society for Protection of Birds, David Attenborough Building, Cambridge, CB2 3QY, UK.

L. Qie was with the School of Life Sciences, University of Lincoln, Brayford Pool, Lincoln, LN6 7TS, UK

I. Introduction

AUTOMATICALLY identifying and mapping trees is a long-standing goal within the field of forest remote sensing, and there is currently particular interest in finding robust solutions for segmenting multi-species stands with complex structures [1]–[6]. With the widespread adoption of Light Detection and Ranging (LiDAR) it is possible to collect data on the three-dimensional (3D) structure of forest stands over hundreds of square kilometers in a matter of hours, commonly in the form of 3D point clouds [7]–[9], from which individual trees can be segmented. Individual tree approaches have the potential to track individual-level changes in response to events such as disease, biological invasion, logging and extreme climatic events (e.g. droughts) thereby gaining a clearer understanding of the processes that generate structural change [10]–[13].

Until recently, most approaches to delineating individual tree crowns (ITCs) relied on converting the 3D structural information contained in LiDAR datasets into rasterised 2D surface models [1], [2], [5]. Typically, these approaches start by finding local maxima which are treated as tree tops, then searching around those peaks to find tree crowns by methods such as the watershed algorithm [14]–[17], region growing [18]–[20], valley-following [21]–[23] or variable-window filtering [24], [25]. Whilst these methods are often very successful at finding the largest trees, particularly in coniferous forests where trees have clearly defined tops, conversion to 2D rasters results in the loss of almost all the information from the understorey, including many smaller trees, which are represented in the full 3D point cloud [5], [6], [20], [26]. For applications such as forest type and cover mapping and above-ground biomass (AGB) estimation these methods can deliver robust results, but for estimating tree size distributions or tree-centric changes this loss of data is problematic [27].

Segmentation approaches that make better use of LiDAR point clouds are becoming more prevalent [5], [6]. The earliest methods worked with voxels: data summarised within 3D grids rather than 2D rasters [28]–[30], but were highly affected by variation in point density [5]. More recently, methods have emerged which produce a full 3D version of the tree by

taking points directly from the point cloud using minimal or no summarisation [6]. Other early approaches made use of common clustering algorithms, such as k-means, to find groups of points thought to be trees [31]–[33]. These produce reasonable results for larger trees, especially in simpler conifer forests where crowns are typically distinct [2]. Such methods are less reliable at finding smaller trees or delineating structurally complex stands [6], such as tropical rain forest and broadleaf temperate forest. These are challenging forests to segment because trees often have overlapping crowns [34] and stands contain multiple layers of trees [4]. To address the multiple layered complexity, methods have been developed which apply an algorithm more than once to point-cloud data, locating and segmenting tall trees, which are then removed from the point cloud before reiterating the algorithm [35], [36]. The latest methods combine a clustering approach with constraints based on the allometry of trees: knowledge of how the size and shape of crowns changes with tree height within a particular forest type [37], [38]. A popular clustering approach has been to apply the mean shift algorithm [39]–[43], for example, AMS3D combines a locally derived allometric relationship between height and crown diameter with an adaptive mean shift algorithm [44]–[46]. A further clustering and allometric approach, Ptrees, combines a multi-scale nearest-neighbour algorithm with analysis of crown geometry [47]. Finally, normalised cut methods, from graph theory, have been applied by using recursive binary cuts (i.e. repeated division of segments into two components), until some stopping point is reached [48], [49]. Because this approach works with a matrix of distances between points in the cloud, it is computationally intensive, and is only applied as a refinement step [39], [48], [50]–[53] or applied to voxelised datasets [48], [50], [54]. Only recently has the use of a Multi-Class Graph Cut approach been applied to the problem of ITC detection to data from European coniferous forests, focusing on Terrestrial LiDAR data and working with voxelised data [55].

This paper aims to provide a method that can simultaneously address a number of common individual limitations and extensions to current segmentation algorithms working with Airborne LiDAR (ALS) point clouds. We introduce a more flexible approach to normalised cuts, with a particular focus on complex tropical forests, which are often on steep terrain. Most point cloud based segmentation approaches require the heights of points to be topographically corrected, where the ground height is subtracted from the heights of non-ground returns [43], [45], [55] This process warps tree crown geometries when the ground is not level and so can introduce artefacts on steep terrain. With the introduction of PTrees, a method to preserve the structure of crowns whilst getting a correct tree height was introduced and addresses this issue for already delineated crowns [47]. PTrees along with the voxelised normalised cut approach in [48] also cluster directly on the un-corrected point cloud. This is uncommon; most other methods cluster on topographically corrected data, including any warping. Our approach, like PTrees, clusters directly on the raw positions of the points relative to each other, though knowledge of the above ground height is used to define a neighbourhood where local point density information is used and, independently of clustering, to apply allometric feasibility checking. Our midway approach preserves overall structure of the point cloud, without introduced artefacts, whilst allowing use of allometric knowledge. which is now commonly used to guide or refine crown delineation. However, it is very common for this process to be developed on local data, or tuned to specific sites and moving to other locations can be difficult [6], [33]. Our approach incorporates the relationship of crown width scaling with tree height, but we use a regional subset of an openly available global database, from [56], to allow easier transfer. Not all methods require allometric data, but this is commonly replaced with assumptions on the geometry of clustering, such as on crown shape [31], [44], [47] or assumptions about stem geometry and size [48]. The approach of AMS3D to automatically build an allometric model from the data is a possible alternative, though this requires an initial crown extraction step which makes assumptions on crown geometry, though this should generalise well [45]. The approach of [36] is a good example of crown detection requiring no allometric knowledge nor strong assumption on crown shape, though this algorithm focuses on a canopy height model (CHM) raster approach to crown detection, only using the full point cloud in one step to detect lower canopy trees.

In this work we develop a novel and flexible approach based upon a Multi-Class Normalised Graph Cut which works directly on the point cloud before topographic correction [57] and addresses the points and limitations laid out above. Our method converts the point cloud into a graph representation, with a vertex for each point. This conversion is based on 3D proximity as well as assessment of the local density of the point cloud. Four parameters allow tuning of the importance of vertical and planimetric distances and vertical and horizontal variations in density. The algorithm then simultaneously splits the graph into clusters of points which are well-connected. The number of clusters is automatically determined based on the structure of the graph produced from the data, and these result in candidate tree crowns. Through a filtering step, using an allometric crown-geometry relationship derived from a regional subset of a global database, the clusters are checked for feasibility and rejected if they don't meet a set of size criteria or contain too few points. From this a set of tree crowns is produced, along with rejected points not yet assigned to a crown. Through a second application of the algorithm it is then possible to address some of the issue of multi-layer canopies. We find that the difficulties of finding obscured lower canopy trees from ALS data remains with our method, but that a double-layer Multi-Class Graph Cut (MCGC) approach produces a good proxy for field inventory of stems in all but the lowest canopy layers. The strength of our approach is inclusion of allometric knowledge in a way that can be easily changed, and requires no local calibration, being based solely on existing database data. In addition to our contribution to address integrating simple allometry and avoiding clustering on topologically corrected data, our method is designed and implemented in a way to make it flexible and open to computational acceleration. We use a numerical approximation to a full normalised cut and downsampling and imputation to reduce the computational time take for our overall pipeline. Flexibility

in the segmentation step is intrinisic with clustering based on a single weighting for each pair. The weights calculation here is one implementation, but this can easily be adapted to include any information available at each point, such as imagery data, characteristics of the LiDAR data acquisition and prior labelling of points if ground truth is known. We don't test this addition here, but it is very simple to add in a manner similar to [49], [58] and the adaptation of AMS3D introduced in [42], such as using our existing algorithm to map hyperspectral imagery to ALS data to guide crown extraction [59]. This extension beyond spatial information is however not present nor the steps to do so explained in many methods [5], [8] and developing methods which have a clear route to this is an important step in individual tree detection.

We also look at addressing the computation of biomass at a plot level. We use MCGC combined with a relationship for tree biomass from local crown measurements to predict total plot biomass as a sum of that contributed by each tree. Many analyses currently focus on summarised data at the plot level, computing metrics such as the distribution of top of canopy height and variation in canopy structure and using these to predict useful forestry attributes such as biomass or canopy fuel mass [60]–[63]. These give good estimates of biomass and carbon density, but lose information on the individual trees found in each plot [27], [64]. Previous work suggests that these methods are preferable to ITC approaches [27] but we show that the MCGC approach to ITC is able to produce improvements in individual-based estimations of hectare scale biomass. This advancement shows the potential of ITC approaches in forest management.

II. Tree Crown Segmentation with Multi-Class Graph Cut

This section outlines the mathematics underpinning a graph cut, summarising the formal work already completed on this [57], [65], [66]. A *graph*, $G = G(V; E)$, is a coupled pair of a set of vertices, V, and edges, $E \subseteq V \times V$, which connect these vertices. Each edge has an associated non-negative weight, $w_{ij} \geq 0$, which represents how closely related the vertices $\{v_i; v_j\}$ are, with larger values representing stronger associations. In this work we use undirected graphs, so $w_{ij} = w_{ji}$. The choice of the weights is application-dependent and indeed crucial for the performance of the graph cut approach. For the tree crown segmentation algorithm proposed in this paper the choice of the weights will be discussed in Section III-B2.

A *Binary Cut* of a graph into two disconnected subsets (A and B), forming a partition of G has an associated cost defined as $\mathrm{cut}(A; B) = \sum_{i;j:v_i \in A;\, v_j \in B} w_{ij}$, which is the sum of weights for all links that bridge the cut. Minimising this cut over possible partitions A and B means that we are looking for a partition which maximises the dissimilarity between the two sets. Similarly a *Multi-Class Graph Cut* of G into k subsets $\{A_i\}_{i \in 1,\ldots,k}$ forming a partition of G has an associated cost defined as:

$$\mathrm{cut}(A_1, \ldots, A_k) = \frac{1}{2} \sum_{i=1}^{k} \mathrm{cut}(A_i, \overline{A}_i), \tag{1}$$

Algorithm 1: Normalised Spectral Clustering [57]

1: Compute the normalised graph Laplacian $L_{\mathrm{sym}} = D^{\frac{1}{2}}(D - W)D^{\frac{1}{2}}$
2: Compute the first k eigenvectors $u_1, \ldots, u_k$ of L_{sym}
3: Form the matrix $T \in \mathbb{R}^{n \times k}$ from $u_1, \ldots, u_k$ by setting $t_{ij} = u_{ij} / (\sum_k u_{ik}^2)^{1/2}$
4: Cluster the rows of T with k-means clustering [66] into clusters $C_1, \ldots, C_k$
5: Return clusters $A_1, \ldots, A_k$ with $A_i = \{j \mid v_j \in V, \text{row } j \text{ of } T \in C_i\}$

where $\overline{A}_i$ is the complement of A_i, so that $\overline{A}_i = \{v_j \mid v_j \notin A_i\}$. Finding a partition $\{A_i\}_{i \in 1,\ldots,k}$ that minimises (1) can lead to qualitatively very good segmentations, but also tends to favour cutting small sets of isolated vertices [67]. Indeed, without additional constraints, a partition where the whole of the graph is contained in one subset and all other subsets are empty is optimal for (1). To overcome this a balanced graph cut is used. This splits the graph into similarly sized subsets (defined by their volume) whilst also trying to minimise linkages between these sets. Here the *Normalised Multi-Class Graph Cut* [57] is used with an associated cost:

$$\mathrm{Ncut}(A_1, \ldots, A_k) = \frac{1}{2} \sum_{i=1}^{k} \frac{\mathrm{cut}(A_i, \overline{A}_i)}{\mathrm{vol}(A_i)}, \tag{2}$$

where $\mathrm{vol}(A)$ is the *volume* of a set of vertices $A \subseteq V$ and defined as $\mathrm{vol}(A) = \sum_{i;j:v_i \in A; v_j \in V} w_{ij}$. The volume quantifies sets with highly globally connected vertices as 'larger' than poorly connected sets. This formulation of the problem enforces segmentation into subsets of more balanced size and reduces the chances of a single dominant cluster, in line with the problem of tree crown segmentation. The trade-off is that it becomes a discrete optimization problem which is NP-complete and difficult to solve directly [57], [65]. To overcome this, an approximation based on trace minimisation of a matrix is used. This is a well-studied problem, and can be solved by spectral clustering methods [65]. The trace minimisation problem that approximates the minimisation problem for Normalised Multi-Class Graph Cut (2) is:

$$\min_{T \in \mathbb{R}^{n \times k}} \mathrm{Tr}(T^{\top} D^{\frac{1}{2}}(D - W)D^{\frac{1}{2}}T), \quad \text{subject to } T^{\top}T = I. \tag{3}$$

where $W = (w_{ij})_{i,j}$ is the matrix of pairwise weights between vertices v_i and v_j and $D = (d_{ij})_{i,j}$ is the diagonal matrix of degrees for each vertex, that is $d_{ii} = \sum_{j:v_j \in V} w_{ij}$ and $d_{ij} = 0$ for $i \neq j$. Moreover, I is the $k \times k$ identity matrix and T is the $n \times k$ matrix used as the basis for spectral clustering, where each row i represents one of the n vertices v_i. Problem (3) is then solved by Normalised Spectral Clustering as per Algorithm 1.

In previous studies, graph-cut based segmentation of tree crowns has been used with recursive binary cuts, or with a Multi-Class Cut with a predetermined number of clusters k

[48], [49], [52], [58], [68]. In this paper, we instead adapt the approach in [69] and allow k to vary. We propose the **Multi-Class Graph Cut (MCGC)** approach where a minimum and maximum number of clusters are set and the eigenvectors up to the maximum number of clusters are computed. For associated eigenvalues $_i$ (where $_1$ is the smallest eigenvalue) the *eigengaps* $_{i+1}$ $_i$ are computed. In the case of a graph with k disconnected components, the first k eigenvectors should have the same value, with a different value for eigenvalues from $k + 1$ onward, causing the largest eigengap to occur between the eigenvalues k and $k + 1$. Extending this principle, we choose a number of clusters which maximises the eigengap. This has been shown to lead to robust and natural segmentations [65], [69]–[71]. The number of clusters is then set by the eigenvalue $_i$ which maximises the eigengap $_{i+1}$ $_i$ within the pre-defined minimum and maximum range, allowing flexibility based upon the data.

III. Methods

A. Field datasets

Data for this paper were taken from a lowland tropical rainforest inventory reserve in Sabah, Malaysia. Sepilok reserve is a 4294 ha protected area within which three distinctive forest types are found: open 80 m tall alluvial forest, dense 60 m tall sandstone hill forest and 30 m tall Kerangas forests on the shallow soils of hill tops. The elevation in this lowland forest ranges from 0 to 250 m a.s.l. [27].

LiDAR: The reserve was surveyed with ALS data on 5th November 2014, using a Leica ALS50-II ALS flown at 1850 m altitude on-board a Dornier 228-201 travelling at 70 ms^{-1}. The sensor emitted pulses at 83.1 Hz with a field of view of 12.0 and footprint of approximately 40 cm diameter with an average pulse density of 7.3 pulses per m^2 giving an average point density of 24.6 points per m^2 across the permanent plots used in this study. A total area of 26 km^2 was covered, including all 9 permanent plots below. The sensor records full waveform ALS, but for this study the data were discretised by the system, with up to four returns per pulse. A nearby Leica base station was used to ensure accurate georeferencing of the data. The data were pre-processed by NERC's Data Analysis Node and delivered in standard LAS format. [27]

Permanent plots: In order to validate the segmentation approach, we compared the outputs with field-measurements. Three permanent inventory plots of 200 200 m are established in each of the three soil types which have been regularly surveyed through out their history; the most recent survey, used for this analysis, was completed in 2013–15. Field inventory data included the diameters of all stems 5 cm in diameter (measured at a height of 1.3 m) and their species identity, mapped to the nearest 10 10 m subplot. Individual stems are not precisely geolocated, but the corners of all 1 ha sub plots of each 4 ha permanent were recorded with differential GPS. Alluvial plots had the largest DBH and lowest stand density (minimum 5.0 cm, median 10.1 cm, mean 15.2 cm, maximum 165.3 cm, stand density 2574 stems ha^{-1}) followed by sandstone plots (minimum 5.0 cm, median 10.0 cm, mean 15.2 cm maximum 142.0 cm, stand density 3894 stems ha^{-1}) with kerangas forest having the smallest stems with the largest stand density (minimum 5.0 cm, median 9.5 cm, mean 13.1 cm, maximum 110.5 cm, stand density 4836 stems ha^{-1}). All have a similar understorey with roughly half of stems being relatively young with a DBH of 10 cm or less and the main difference comes from the size reached by the established trees in the upper layers of the canopy. For some stems, a field-measured height was also recorded. Additionally, in [27], a set of 91 crowns were identified in the LiDAR dataset and subsequently mapped in the field. These include field and LiDAR based measures of height, stem diameter and crown area.

Allometry: Having allometric relationships that estimate crown size for trees of a given size is important for accurate segmentation, for reasons explained below. Allometric relationships were obtained by subsetting the Indo-Malaya region from a global allometry dataset [56]. In total 7,943 trees were included, ranging in height from 1.4 m to 70.7 m. Quantile log-log regression (from the `quantreg` R package) [72] was used to fit 'median' and 'upper boundary' relationships through the data, with the following form: $CD = H$ where CD is crown diameter, H is tree height with both measured in metres and and are fitted. Quantile regression is analogous to least squares regression, which predicts the mean of response for a given predictor value, instead fitting a model for the specified quantile of the response at a given value of the predictor. We fitted models to log-transformed data, to produce power law relationships. The 50th and 95th percentiles of this relationship were computed as follows:

$$\begin{aligned} CD_{IM_{50}} &= 0.251 \quad H^{0.830}; \\ CD_{IM_{95}} &= 0.446 \quad H^{0.854}. \end{aligned} \tag{4}$$

These relationships were then used to predict crown diameter and radius for trees of each height when applying allometry, as explained in the next section. The relationships in (4) are the only allometric models used for delineation, being based solely on a global database. For other biomes or ecoregions one can use a different subset of the same, or a similar, database. We also computed models for all tropical forests and all trees worldwide and found that the estimation of biomass was similar, though slightly less accurate than for the Indo-Malayan model.

The field inventory at Sepilok includes stem diameter but rarely records tree height as needed to estimate tree biomass as explained in Section III-E. To compute the height of trees in the field inventory at Sepilok we used the relationship described by Coomes et al. in [27]. This approach was only used to estimate biomass from the field inventory. Here height, H (m), is predicted as a function of stem diameter, D (cm), as a power law, with a different relationship applied to each soil type as follows:

$$\begin{aligned} H_{Alluvial} &= 2.105 \quad D^{0.679}; \\ H_{Kerangas} &= 4.57 \quad D^{0.461}; \\ H_{Sandstone} &= 4.001 \quad D^{0.527}. \end{aligned} \tag{5}$$

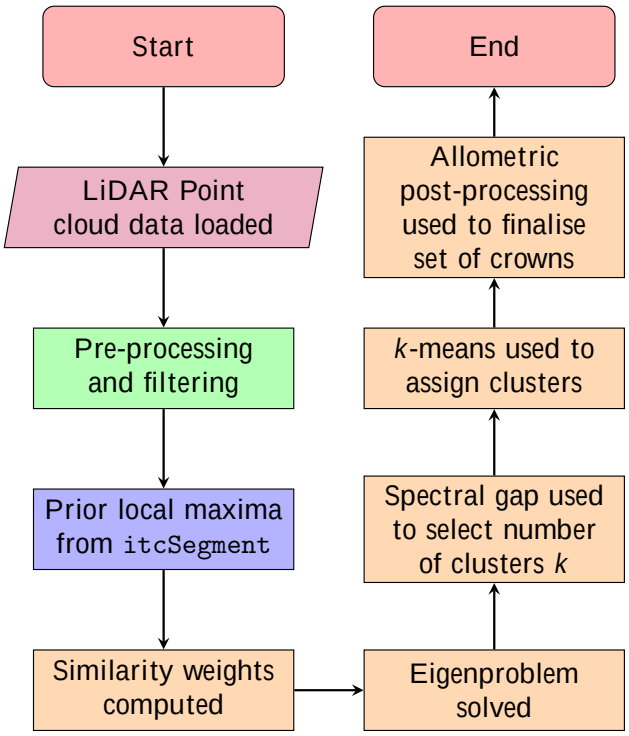

Fig. 1: MCGC tree crown segmentation algorithm. Here the colour of processes reflects the environment they have been coded in: LASTools are marked in green, R code is marked in blue and MATLAB code is coloured in orange.

B. The Multi-Class Graph Cut algorithm

The MCGC approach, summarised in Fig. 1, is explained in detail in this section. LAS data is first pre-processed and then the MCGC algorithm is applied to delineate tree crowns. Further post-processing based on knowledge of tree architectural geometry (i.e. allometric relationships) is then applied to ensure only sensibly shaped crowns are retained. A final double-layer extension is then explained.

1) Pre-processing and prior generation: The point cloud data are first cleaned using the LASTools package[1]. Points marked as noise are removed and the `lasheight` method is used to generate a second point cloud with a model of the ground subtracted from the data points (based on points labeled as ground) to enable computation of above-ground height of points for prior generation.

The accuracy of the graph cut is improved by generating a lower bound on the number of expected tree crowns. This is estimated using the topography-corrected height point cloud and tree allometry information. Here we produced a list of expected tree-top positions computed by itcSegment — a local maximum finder that scales its search window size in relation to tree height used in the R package `itcSegment` [27], [72]. Work presented here is completed using a rasterised canopy height model (CHM) gridded to 0.5 0.5 m pixels. A 50th percentile relationship between height and crown diameter was used to set the size of the local search window based on the height of the CHM at that location. In this study the relevant relationship in (4) was always used in computing the prior. The number of local maxima exceeding a threshold of 5 m then sets a lower bound for the total number of trees to be found by MCGC. This height threshold was set to avoid inclusion ground returns and to avoid over-counting in areas of open canopy, where the search window becomes very small. As itcSegment only works on the top of canopy, it is expected to miss many lower canopy trees and so this number should be an underestimate. The MCGC approach, as outlined in Section II, requires a minimum and maximum number of tree (clusters) per plot to work with. The minimum matching the number of prior 'trees' and a maximum of twice this value was set. This choice was made to avoid forcing over splitting of crowns in the top, dominant layer of the canopy. We justify the suitability of this choice in the discussion in Section V-A1.

2) Graph cut: In what follows, we explain in detail the steps involved in our application of the MCGC graph cut algorithm introduced in Section II. The first step of the graph cut algorithm (implemented in MATLAB) converts 3D co-ordinates into a graph representation for all points above a minimum height of 2 m, to avoid ground returns being included. Here the raw height above sea level values are used to preserve the true structure of the vegetation. Working with topographically corrected data causes crown structures to be warped when the terrain is not flat as points are measured from the ground directly below them and not from the base of their respective tree stem [47]. In this work the 3D information in the LiDAR point cloud was used to construct weights to represent the similarity between points. The performance of graph cut approaches is contingent on the choice of weights, w_{ij}, and these must reflect the structure of segmentation targets to produce good results. Here we introduce a novel approach to assessing the similarity of points in a 3D point cloud for tree detection which avoids the need for prior determination of tree tops or stem locations and only uses the information within the structure of the point cloud. Our process captures knowledge of the basic geometry of the data and of the local density of points and how this relates to tree crowns. The key idea in using the local density of points is that for points near the external boundary of a crown, the local density of points constituting the crown should be higher than that of points outside the crown. This follows from a similar reasoning as used in [44] that trees should form local modes of point density in the data, but we extend this idea to focus on detecting the boundary of a crown. Computing a centre of mass (centroid) for the points in a neighbourhood of a boundary point should result in a point which lies roughly between the centre of the crown and the boundary point on which the neighbourhood is centred. Two nearby points on the boundary of the same crown should have their local centroids in a similar direction relative to the respective boundary points. In contrast, two boundary points on different crowns will have their local centroids located towards the middle of their respective crowns and so relative to the two boundary points, these will be orientated in different directions. Comparing the relative orientation of local centroids with respect to points

[1]https://rapidlasso.com/lastools/

being compared should help distinguish crown boundaries. A simplified example of this principle is illustrated in Fig. 2 and a practical implementation on artificial data is shown in Fig. 3. As demonstrated in Fig. 3d, points on the opposite sides of a crown will also be penalised by this approach. For a single point, high similarities will only occur for links to points on the same side of the same crown and low similarities will result for points in neighbouring crowns and those on the opposite side of the same crown. Moving slightly around the crown, to neighbours of this point in the same crown, we see that points in other crowns still score low similarity, but the high similarity region of the desired crown, and thus the low similarity 'shadow', changes, centring on the new point. Continuing this across the whole crown results in a well-connected set of points from the crown with strong links between mutual neighbours whereas neighbouring but distinct crowns have lower similarity scores to this entire set as no strong link exists to the crown of interest.

To encode the information from the local density of points into the similarity weights, the local centroid for each point was computed. The neighbourhood used was a sphere centred on the point in question with a radius based upon the above-ground height of the point in question. The radius of the sphere was based on the 95th percentile relationship in (4). To avoid the sphere extending beyond the stem location for boundary points in most trees the radius of the sphere was set to be half of the reference radius computed from the allometric relationship. This would capture most of the points in the relevant half of crown without extending too far and being skewed by other trees locally. The centroid of all points in this neighbourhood and the relative vector from the original point to its centroid, denoted by $_i$ for point i, was then computed. In comparing points, the orientations of these relative vectors were compared. Points which belong to boundaries of different crowns would be expected to have vectors with very different orientations. Each of these vectors was broken down into horizontal and vertical components, denoted ^H_i and ^Z_i respectively. Where the orientations suggested membership of different crowns, the similarity value (w_{ij}) was reduced. The process of computation of w_{ij} is outlined as follows (with $w_{ii} = 0;\ \forall i$). Note that it would be straightforward to introduce further similarity comparisons when constructing weights, such as comparing intensity values from imagery, similar to our work in [49], [58], but we have not done so here.

First, a basic similarity based on the distance between each pair of points is computed as

$$w^{\text{base}}_{ij} = \exp\left(-\frac{\|(x;y)_i - (x;y)_j\|_2^2}{\sigma_{XY}^2} + \frac{(z_i - z_j)^2}{\sigma_Z^2}\right); \quad (6)$$

Here, w^{base}_{ij} in equation is separated into a horizontal (first exponential term in (6)) and a vertical component (second exponential in (6)). Each component can have its importance controlled by separate parameters as the vertical and horizontal structure and extent of crowns can vary for different forest types. The horizontal and vertical parameters are set before applying MCGC and are the same for all points ($_{xy}$ and $_z$ respectively). Here $(x;y;z)_i$ are the raw coordinates of point i. An example of this is illustrated in Fig. 3b.

Next, the horizontal angle between the two centroid vectors ^H_i and ^H_j is computed as $_H(i;j)$. When this is more than a right-angle, the basic similarity w^{base}_{ij} is reduced. This reduction is based on the horizontal distance between the points, d_H, so that closer points get a larger reduction in similarity. This effect is normalised by a scale parameter, K_H, which is automatically set to the allometric radius for the tallest point in the data, using the 95th percentile allometry relationship in (4). Similarly, the larger the relative difference in the horizontal components of the centroid vectors, the larger the reduction in the similarity. The parameter W_H controls the overall importance of this modification relative to all other steps in the computation. An example of this reduction is shown in Fig. 3d. With this, the basic w^{base}_{ij} is updated to w^{postH}_{ij} as

$$w^{\text{postH}}_{ij} = \begin{cases} w^H_{ij}; & \text{if } _H(i;j) > \frac{}{2}; \\ w^{\text{base}}_{ij}; & \text{otherwise:} \end{cases}$$
$$w^H_{ij} = w^{\text{base}}_{ij} \quad \exp\quad W_H \frac{K_H}{d_H} \| ^H_i \quad ^H_j \|_2 \ : \quad (7)$$

Finally, the vertical components of the centroid vectors are compared (^Z_i and ^Z_j). When these point in the same direction, no adjustment is made to the score. Where the directions differ, only pairs where these diverge have their weight reduced. Divergence occurs when the taller of the points has a positive Z and the lower point a negative one. This would be expected for points in different crowns of varying height, whereas points at the top and bottom of the same crown would expect to have vertical components that point towards a central point in the crown. As with the horizontal comparison, this reduction is larger for points which have a smaller vertical distance, d_Z, between them. The normalisation for this, K_Z, is set to half of the aboveground height of the tallest point of the data. The reduction is also larger when the relative difference in the vertical centroid vectors is larger. An example of this reduction is shown in Fig. 3e. With this, the final similarity weights, w_{ij}, as shown in Fig. 3c, are computed as

$$w_{ij} = \begin{cases} w^Z_{ij}; & \text{if } ^Z_i \text{ and } ^Z_j \text{ diverge;} \\ w^{\text{postH}}_{ij}; & \text{otherwise:} \end{cases}$$
$$w^Z_{ij} = w^{\text{postH}}_{ij} \quad \exp\quad W_Z \frac{K_Z}{d_Z} \| ^Z_i \quad ^Z_j \| \ : \quad (8)$$

Calculating pairwise weights for every set of two points would be computationally cumbersome, producing a matrix of weights that is far too large to store in a typical computer's memory (16 GB RAM). As an indication, a point cloud of 40,000 points would require 12.8 GB of RAM to be held in memory, ignoring overheads and the need for spare memory to perform the necessary computations, which at the point density of the Sepilok dataset would cover roughly 0.25 ha. To resolve this issue, the Nyström extension is used [73]. Here the complete matrix and relevant eigenvectors are

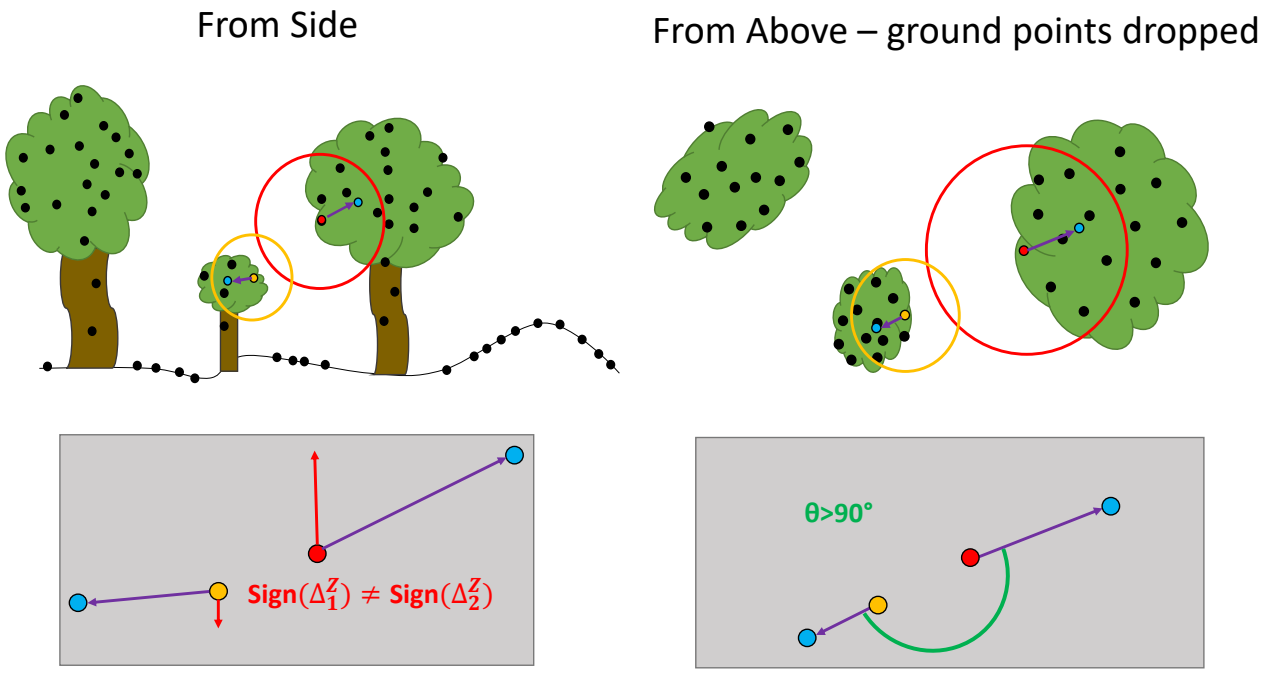

Fig. 2: Simplified illustration of the principle of the local density centroid calculations. Points being compared are red and orange, with associated neighbourhoods shown. Blue points are centroids for each neighbourhood, with centroid vectors in purple. Comparisons of these vectors are then shown in the grey boxes. In this example both comparisons would result in a reduction in the similarity weight between the highlighted points. Here the illustration shows how the horizontal and vertical directional differences in the centroid vectors distinguish the points highlighted as belonging to distinct crowns.

computed for a subset of the points, typically less than 10% of the data. The eigenvectors are then extended to the full dataset through quadrature based on the weights in a manner that produces a robust approximation of w_{ij} [73], [74]. This allows computation of all pairwise weights for the subset and avoids a need for a sparse representation of the pairwise linkages. Computationally, we found computing eigenvectors takes much more time than computing a complete graph (where all points are linked to all other points). Thus the Nyström extension approach is more efficient than trying to compute complete eigenvectors on a sparse version on the graph.

3) Using crown allometry to refine the segmentation: An extension of the basic graph cut algorithm makes use of knowledge of crown radius scaling with tree height to remove improbable trees, by post-processing of clusters identified by the initial graph cut. For a candidate tree in the segmented dataset, a 'maximum' predicted crown radius was taken from look-up table based on its height. The numbers in the look-up table were based on the 95th percentile relationship from the global database in [56] as given in (4). Any crowns which exceeded the range of this dataset were set to match the maximum allometric radius for the dataset. The horizontal top of a tree was approximated as the mean position of all points with aboveground height of 98% or greater of the top point. First candidate crowns that overlapped too much with larger neighbours, both in horizontal extent and vertical overlap were merged. We define excessive horizontal overlap as when either the top or more than 60% of the points of a tree are within the allometric radius of a taller tree. The threshold of 60% was chosen to be a balance between ensuring a majority of points in a candidate crown need to be within the region but also avoiding ignoring all but significantly overlapping candidate crowns. This threshold was set as part of the algorithm and not treated as a parameter. Excessive vertical overlap is when the top quartile of heights in a crown overlap the bottom quartile in a taller crown. This approach was chosen as a non-parametric approach based solely on quartiles. This was also set before analysis and not treated as a parameter. Crowns are only merged if both horizontal and vertical overlap are excessive. Then crowns where more than 5% of points lie beyond the computed maximum radius from the tree top are further trimmed. This cut-off is set to match the radius being used, here being a 95-th percentile relationship. This was also set before analysis, again not being treated as a parameter. Trimming uses hierarchical clustering based on the Euclidean distance between points, using raw heights. Two clusters are produced and the one which includes the tree top is kept, with points in the other cluster added to the rejected points list. Finally crowns which contain too few points were rejected, with a threshold of a minimum of 100 points per crown. This was set to filter crowns which were missing many points and to

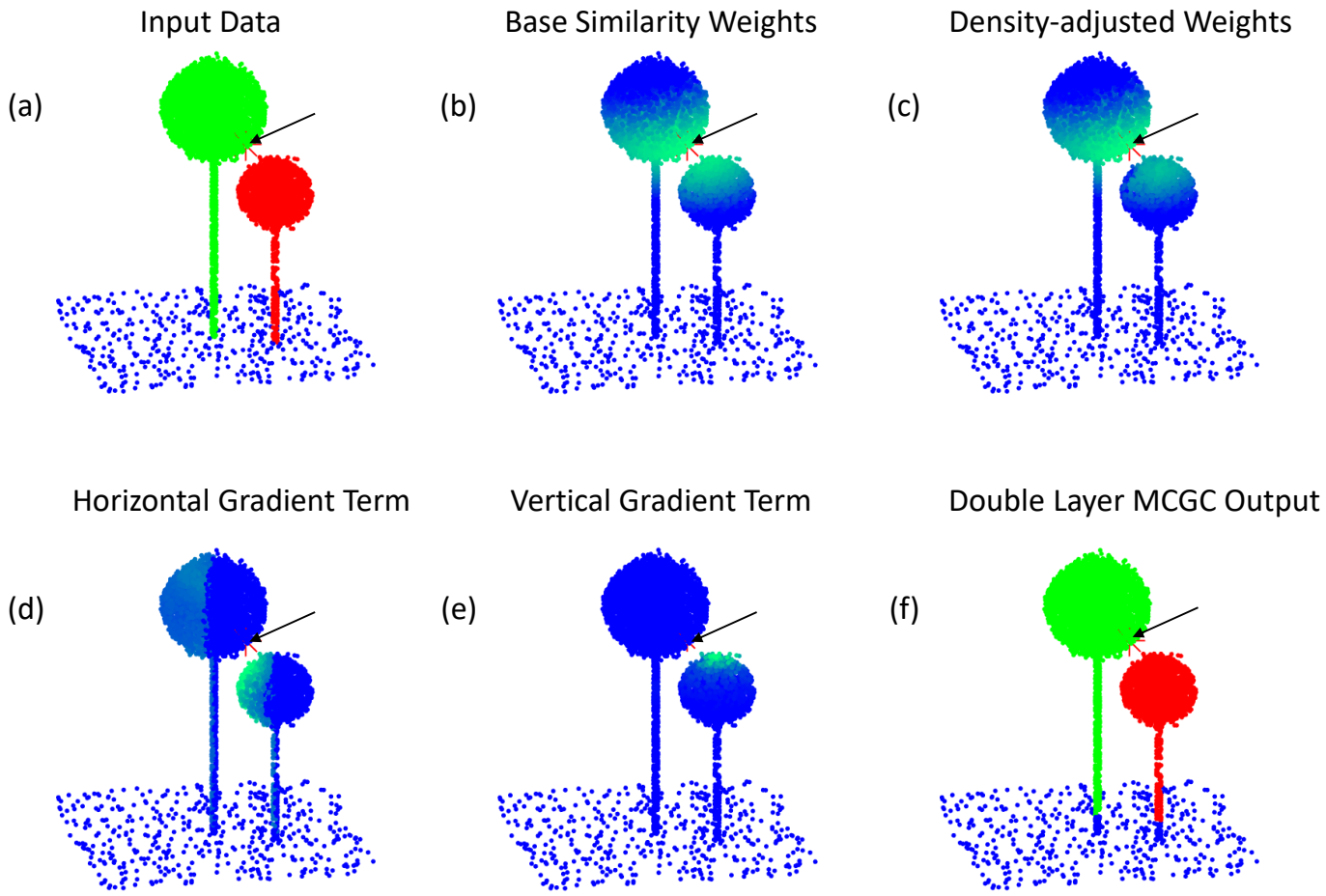

Fig. 3: MCGC applied to artificial trees to explain the role of centroid vector computations in segmentation. The relative effect of applying the centroid adjustments when comparing to the marked point are shown in panels (d) and (e), and this is reflected in the weights as shown in panels (b) and (c), where the closest points on the smaller crown have weaker links than the base weights, with minimal difference made to points in the taller crown. Panel (a) shows two simulated trees with allometry matching that of the 50th percentile for crowns in Indo-malaya where the green tree is 30 m tall and red tree is 20 m tall with blue points being the ground. Panel (b) shows the qualitative distribution of w_{ij}^{base} for each point computed with reference to the crown boundary point highlighted by an arrow in all panels, centred at the red star; higher values are represented by green in the blue-green colour ramp. Panel (c) shows the distribution of w_{ij} once the centroid adjustments have been made. The colour ramp is the same and shows the reduction in similarity to the neighbouring crown whilst preserving linkages to the target crown. (d) shows the effect of the modification based on horizontal components of centroid vectors in (7). Dark blue represents minimal adjustment to the weight and lighter colours represent a greater reduction in weight. (e) shows the same effect as in (d) when looking at the adjustment based on the vertical component of the centroid vectors in (8). (f) shows the output of double-layer MCGC, with each crown highlighted in a different colour, with unclassified points in blue, recovering the true structure in (a) with the exception of the bottom 2 m of the stem, as this was the height threshold set.

avoid the trimming step from causing crowns to be too small. The goal here is to ensure only allometrically feasible crowns are kept, with all other points being marked as not contained within a crown. These points are rejected in this application of graph cut.

4) Detecting lower-canopy trees: The allometric refinement stage leads to there being points which are not assigned to any tree. These can be from trees which are close neighbours of successfully detected crowns where removal of this crown makes delineation easier. Equally many of these points are in the understorey, where point densities are lower due to occlusion and differentiation of crowns is more challenging. A second pass of MCGC is then used to detect tree crowns from the unassigned points. These trees are then added to those accepted in the first application of MCGC to produce the final list of crowns. It is possible to alter the weight parameters to reflect differences in the canopy structure in these lower layers; however we applied the MCGC algorithm with the same set

TABLE I: Values of Parameters Selected for MCGC by Trial and Error

	$_{XY}$	$_{Z}$	W_H	W_Z
Value of parameter	4	2	0.2	0.2

of parameters in the second pass in this study. This simple extension is illustrated in Fig. 4.

5) Selecting parameters: The values of $_{XY}$, $_{Z}$, W_H and W_Z in (6) – (8) control the relative importance assigned to each component in computing similarities between points. To set these, nine of the 36 1-ha plots were used, three from each soil type, comprising 25% of the data in this study. To decide the best choice of parameters from those trialled, the resulting segmentations were manually inspected, and the distribution of detected crown sizes was qualitatively compared to the distribution of field inventory stem sizes for the 9 plots. We chose not to automatically tune the parameters based on biomass estimates as this would be likely to falsely remove any bias, possibly by allowing unrealistic crown delineation. An automatic method to assess accuracy of tree shape could be used for a robust selection with a more detailed field inventory. First the values of $_{XY}$ and $_{z}$ were simultaneously assessed, by trying MCGC with each parameter taking one of a number of values and trying all possible pairings. For this process, the modifications in (7) and (8) were not applied to any weightings. It was expected that the vertical distance would need a smaller parameter, penalising this more heavily, as tropical crowns are often wide and less vertically extended than the typical conic shape of coniferous forests [56]. Analogous parameters in previous work on the use of graph cut on coniferous forests found the horizontal parameter to be the smaller, following the same reasoning [49]. Similarly to the first two parameters, W_H and W_Z were simultaneously trialled in an exhaustive manner, with $_{XY}$ and $_{Z}$ set to their selected values. The final parameters used in this study are listed in Table. I.

C. Improving computational efficiency

As shown in Section II, applying the graph cut algorithm amounts to solving an eigenvector problem. Such problems do not scale well with increasing matrix size. The Nyström extension already reduces the effective matrix size used in the algorithm. However, as the number of points in the dataset increases, the total memory required still increases. To resolve this only with the Nyström extension would require taking a very small subset of points for applying the approximation. Instead, when working with datasets of 1 ha or more a simple method for reducing the workload was developed, by downsampling the data and then imputing the full data results. This can be justified as the dense point cloud is locally correlated – points close in horizontal extent will have similar height values and a subsampled point cloud retains the 3D structure of the dense data. This way the MCGC algorithm is applied in full to a subsampled point cloud and the Nyström extension can be applied at the same proportion of points used in MCGC for datasets over large areas, or from very dense ALS data. The full pre-processed point cloud is downsampled, by random (without replacement) sampling of the data. The Multi-Class Graph Cut is then applied to this subset. Once the crowns are identified, the full dataset is then imputed by a m-nearest neighbour approach. The value of m is set to match the effective downsampling, so if 1/5 of the data is used then $m = 5$ is used for imputation. For a given crown, points could only be added to it in imputation if they lay within the maximum allometric radius of this crown, as computed from the 95th percentile relationship in (4) for the height of this crown. To ensure imputation did not create artefacts, where rejected crowns were now grouped with their nearest valid crown, the DBSCAN algorithm was applied to each final tree [75]. This finds groups of points for which there are at least 10 other points within 2 m. A neighbourhood is then constructed of all neighbouring points which satisfy this property, as well as all points within 2 m of this group. This ensures imputed crowns are formed of a single locally-connected group of points and any points not connected to this were marked as unassigned. In this work we used a subsampling pool of 20% of the data. In the double-layer extension, subsampling and imputation was applied to each application of MCGC, applying it twice in the pipeline. We explored varying or removing subsampling in the second pass of MCGC, as returns in the lower canopy are more sparse owing to occlusion. We found results are generally stable for using 20% of the data or more, with detail set out in Appendix A. In running the full double-layer MCGC algorithm to all 36 one ha plots in this study, the total time taken averaged 66,250 s. This is equivalent to the algorithm taking 30 m 40 s per plot. This timing was completed on a workstation running Windows 7 using MATLAB 2017a. The workstation was equipped with an Intel Xeon E3-1240 V2 CPU, comprising 8 cores running at 3.4 GHz with 16 GB RAM. MATLAB allocates memory smartly to enable calculations to proceed where theoretically they may be RAM-limited. Accordingly, running the MCGC algorithm on a machine with more RAM, or where subsampling is optimised to adapt to the RAM restrictions of a machine has the potential to accelerate the algorithm, but we do not explore that option in this work.

D. Assessment of segmentation accuracy

MCGC was applied to the 36 1 ha plots within the Sepilok dataset. The two versions outlined in Section III-B of the algorithm were tested: (1) **single-layer MCGC** applied a single graph cut to the data followed by the allometric filtering (as per Fig. 1); (2) **double-layer MCGC** extended the results of this by applying a second pass of the algorithm to unassigned points (as per Fig. 4).

To compare the distribution of trees found by the automatic detection methods, the diameter at breast height (DBH) of these were estimated. This metric was previously recorded in the Sepilok data set for all stems, whereas their heights were scarcely recorded. To convert from remotely sensed height to DBH, log-log regression was applied to the 91 trees manually

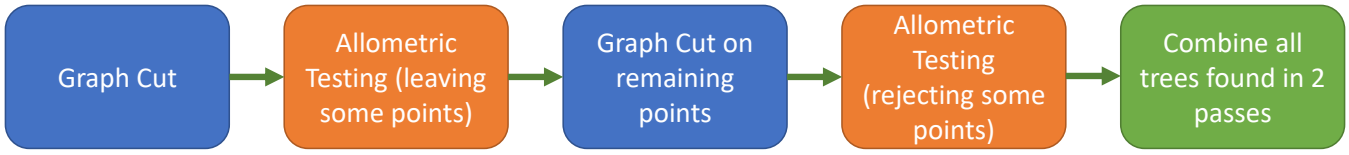

Fig. 4: Pipeline for double-layer approach to MCGC

delineated by Coomes et al. [27]. This results in the following relationship for H (m) and DBH (cm):

$$DBH = 0.252 \quad H^{1.465} \tag{9}$$

DBH for each tree crown detected by the MCGC algorithms was estimated from their above-ground height using this relationship. The results were then compared by grouping stems into a number of diameter classes and comparing the number of stems in the field inventory and the number of stems predicted by MCGC.

E. Predicting biomass

Field surveys are commonly used as a basis for estimating carbon stored in above-ground biomass. Accordingly we computed biomass estimates for the trees in each plot to measure the total estimated above-ground carbon density for each 1 ha plot (ACD, $Mg\,ha^{-1}$). For each plot we compared the total ACD from the field survey to that from each of the MCGC based methods. Here AGB was computed for each tree, and these values were then summed for each plot. Finally a conversion factor of 0.47 was applied to convert from AGB to ACD [76].

For the field survey, the biomass from each tree in the forest inventory was computed according to Chave et al.'s pantropical equation [77]:

$$AGB_{eld} = 0.0673 \quad (WD \quad D^2 \quad H)^{0.976}; \tag{10}$$

where WD is the wood density as taken from the global wood density database [78], [79], D is the stem diameter (cm) and H is the estimated height (m) based on (5). Wood density was mapped to the best taxonomic unit available for each species. If there was no data for a species, then the average for its genus was used, and similarly where there was no data for this the family average was used. If there was no data at family level, an average of all species present in the Sepilok plots was used. The AGB per plot was then computed by summing the contribution from each tree before being converted to ACD.

The ACD contribution of each automatically segmented tree was computed using the following relationship, originally derived by Coomes et al. in [27] from 91 crowns manually delineated and verified in the LiDAR data:

$$ACD_{auto} = 0.268 \quad (H \quad CD)^{1.45}; \tag{11}$$

where H is the aboveground height of the segmented tree (m) and CD is the crown diameter (m) computed from the crown area (CA) as $CD = 2\sqrt{CA/}$. For the MCGC output, the crown area was taken to be the area of the convex polygon enclosing each tree. Comparisons between the ACD estimates from the field inventory and remote sensing methods were compared for each one ha plot. Overall results are then reported via the bias and Root Mean Square Error (RMSE) of the predictions. These were computed as a percentage of the predictions of ACD for the field inventory data. Here a negative bias indicates the remote sensing estimate is an underestimate.

Previous work on automatically detected individual-based AGB estimation notes that often this systematically underestimates biomass as trees in the lowest layers of the canopy are rarely detected [20], [27]. In [27], using the same data as in this study, trees with a DBH below 30cm account for 23% of the total biomass across all plots. It can therefore be informative to apply a simple linear correction factor to account for this. In [27] this was applied both to all plots simultaneously and with a different correction factor for each forest type. Similarly, in [45] the AGB at plot level is computed by applying a multiplicative constant to the sum of estimated AGB for each crown. As we found our overall estimate of ACD from double-layer MCGC was already low-bias overall we chose to only compute a forest-type specific correction factor. This can be justified as the MCGC algorithms here used only a regional allometry which was based on data for the whole Indo-Malaya region in (4) with no local allometric input for the MCGC algorithm. A forest-type specific correction was computed by fitting a linear regression model to the ACD estimates from MCGC and the field inventory for the 12 plots of each forest type, constrained to pass through the origin.

F. Comparison to existing methods

MCGC was directly compared to two existing algorithms as a benchmark of performance. The itcSegment algorithm as implemented in [27] was used as a reference ITC algorithm. This is applied to the rasterised CHM, computed at a 0.5 m resolution. First a moving window local-maxima finder detects tree tops, where the size of the window scales with above ground height in the same manner as our prior generation step does. Finally trees are delineated by a region-growing algorithm taking the local maxima as seed points. Data from this algorithm were processed in the same way as those from MCGC as outlined in Sections III-D and III-E. Estimation of AGB was further compared to an area-based model [80]. This predicts biomass at one ha scale based on metrics of the CHM. The approach uses the general biomass equation from [80] for which ACD is computed for a one ha plot as $ACD_{General} = 3.836 \quad TCH^{0.281} \quad BA^{0.972} \quad WD^{1.376}$, where TCH is mean top canopy height (m), BA is basal area ($m^2\,ha^{-1}$) and WD is

diameter-weighted mean wood density ($g\,cm^{-3}$). To estimate BA and WD from the ALS data the approach used in [27] was applied, where both were estimated as power laws of TCH based on the 36 one ha plots to give an overall model for ACD as $ACD_{General} = 7.37\ TCH^{0.870}$. Following the suggestions in [80] and replicating the approach in [27] a locally-fitted area-based model was also used. This takes the same form as the general model but with the powers and multiplicative factor directly fitted to the Sepilok plots but log-log regression. Further, following the work in [27], BA was predicted not from TCH, but instead from Gap Fraction at a height of 19 m (GF_{19}). This counts the number of pixels in the CHM raster for which the TCH is below 19m as a percentage of all pixels in the raster. Brought together these produce a locally-fitted model for ACD as $ACD_{Local} = 25.93\ TCH^{0.437}\ GF_{19}^{0.209}$.

IV. Experimental Results

A. Tree detection

Single-layer MCGC detects many of the largest trees, but misses intermediate and lower layers of the canopy. The additional detection of trees in double-layer MCGC is evident in the example results for one plot of each forest type shown for each method in Fig. 5. Fig. 6 shows the comparison between the field survey measurements and MCGC-estimated DBH values broken down by bands of diameter and detection rates by diameter class are summarised and compared with the results from using itcSegment in [27] in Table. II. Single-layer MCGC under-estimated the number of trees across all diameter classes (Fig. 6a). This results from a strict set of allometric testing criteria, where clusters that don't pass are rejected. Thus under-estimation of even the tallest trees can be expected. In this case, many of the tallest trees are accounted for (69.8% of trees with DBH>110 cm and 62.8% of those with 90 cm<DBH<110 cm). However, when compared to the same counts for itcSegment, which works only on the top canopy surface, (103.2% and 82.2% respectively) it is clear single-layer MCGC does not account for all trees in the top layer as these are being detected by itcSegment. For all trees with DBH of 90 cm or smaller a single application of MCGC detects fewer than half the number of stems recorded in field inventory, and itcSegment outperforms single-layer MCGC. These trees account for the vast majority of total stems though many will be in the intermediate layers of the canopy. This motivates and justifies applying the algorithm a second time to detect these intermediate layer trees with many of the largest trees already confidently detected.

Once a second pass of MCGC was applied in the double-layer MCGC pipeline (Fig. 6b) the number of trees found increased across all diameter classes. This leads to some overestimation in the tallest classes (128.6% for DBH>110cm and 137.2% for 90 cm<DBH<110 cm) but leads to much better estimates of trees in the intermediate size ranges (87.4% for 70 cm<DBH<90 cm and 56.6% for 50 cm<DBH<70 cm). These detection rates are comparable to, or improvements on, itcSegment for the same size stems (51.8% and 61.9% respectively). Double-layer MCGC still misses most of the trees in the lowest canopy layers (29.9% for 30 cm<DBH<50 cm and 2.2% for 10 cm<DBH< 30 cm). In these classes of stem size itcSegment reports a higher number of stems though itself still misses a large proportion of stems (53.5% and 9.5% respectively). For all diameter classes where DBH<90 cm, adding a second pass of MCGC more than doubles the counts of stems found by single-layer MCGC which means more trees in these diameter classes are found in the second application of MCGC than are found in single-layer MCGC alone.

TABLE II: Summary of Detection Rates by Diameter Class for MCGC and itcSegment with stem count by diameter size class for the field inventory data. A dectection rate of 100% means a count exactly matching that of the field inventory and values above 100% show oversegmentation and values under 100% undersegmentation

	Diameter class (cm)					
	>110	90-110	70-90	50-70	30-50	10-30
Algorithm	Detection Rate (%)					
itcSegment [27]	103.2	82.2	51.8	61.9	53.5	9.5
single-layer MCGC	69.8	62.8	38.9	23.6	12.7	0.7
double-layer MCGC	128.6	137.2	87.4	56.6	29.9	2.2
Stem Count	63	129	247	712	2761	18518

TABLE III: Bias and RMSE for AGB estimates from MCGC, itcSegment and Area-based Modelling for both original model output and once a forest type specific correction factor has been applied to MCGC and ITC and the area-based model has been locally calibrated

	Original		Specifically Calibrated	
Algorithm	Bias (%)	RMSE (%)	Bias (%)	RMSE (%)
itcSegment [27]	-20	26	-1	18
Area-based Modelling [27]	-19	20	0	13
single-layer MCGC	-48	36	3	21
double-layer MCGC	-0	33	-2	18

B. Biomass estimation

Single-layer MCGC underestimated biomass of all three forest types (bias: -48%, Fig. 7a and Table. III) and including a second pass of MCGC removed the overall bias in estimating plot ACD Fig. 8a and Table. III). This is consistent with the comparison of stem counts in Fig. 6a which showed that the single-layer MCGC approach underestimated stem sizes across all classes. Including a second pass of the MCGC algorithm removed the overall bias in the predictions (Fig. 8a). However, this resulted from a tendency to overestimate biomass in the Alluvial forest type matched with small underestimation in both Kerangas and Sandstone forest types (Table. IV). Underestimation can be accounted for by missing trees in the lower canopy. Alluvial forest contains almost all of the tallest trees in Sepilok and so the tendency of double-layer MCGC to over-detect these explains the overestimation of ACD in these plots. Both approaches had a relatively large RMSE (36% and 33%) but this is not unexpected in an individual tree approach to estimation of biomass as big trees contain a large proportion of the total biomass. Small errors in prediction of the largest trees can therefore have a large effect on overall predictions

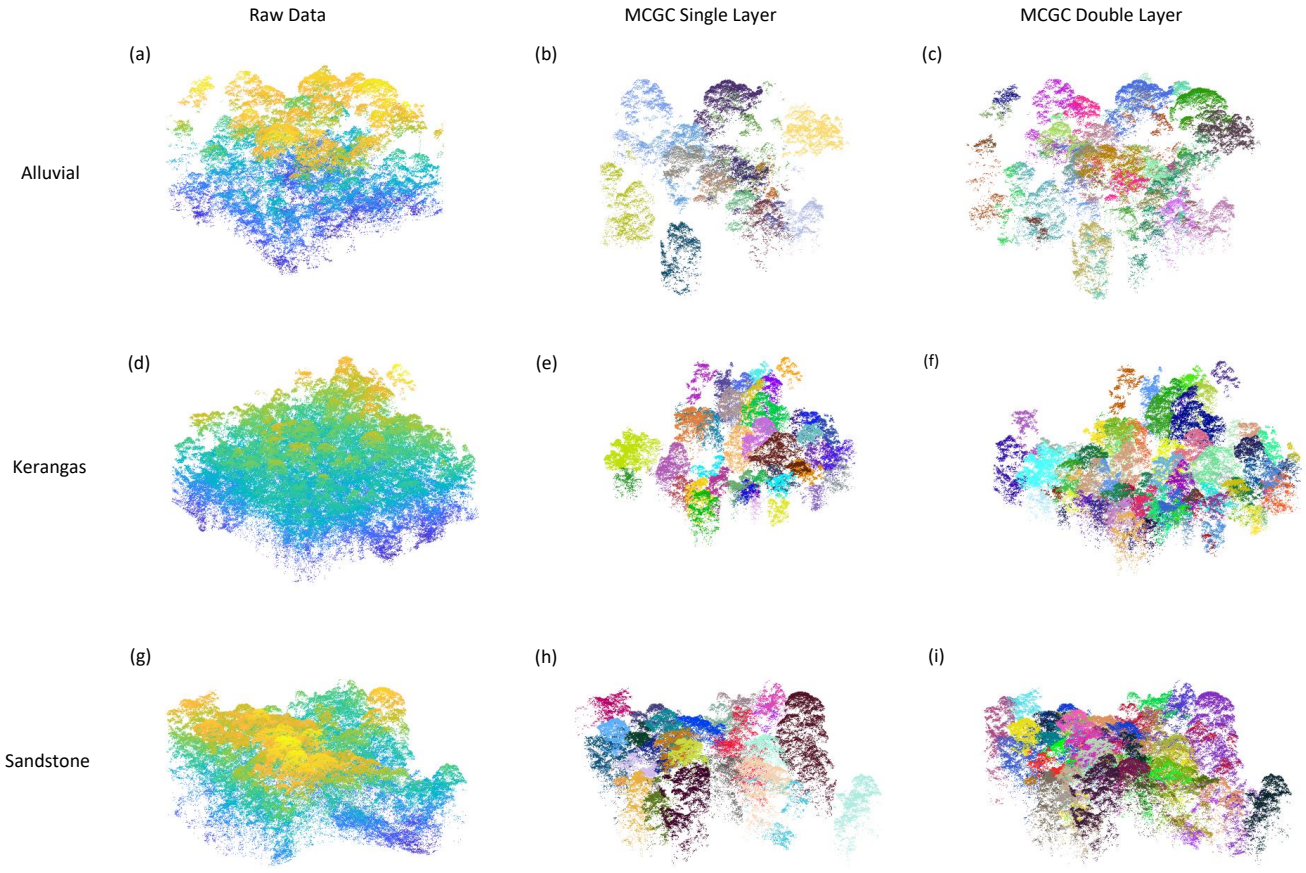

Fig. 5: Example results of applying MCGC single-layer and MCGC double-layer to a 1 ha plot of each forest type. Each row represents a plot of one of the three soil types: Alluvial (a–c), Kerangas (d–f) and Sandstone (g–i). Across each row are the raw data (coloured by height), the output from single-layer MCGC (coloured by crown) and the output from double-layer MCGC (coloured by crown). Points which are unassigned are not included in the MCGC output images and the increased detection of trees can be seen by comparison between the second and third columns: pairs b & c, e & f and h & i.

TABLE IV: Bias and RMSE for AGB estimates from MCGC, itcSegment and Area-based Modelling based on original model output broken down by forest type

	Alluvial Forest		Kerangas Forest		Sandstone Forest	
Algorithm	Bias (%)	RMSE (%)	Bias (%)	RMSE (%)	Bias (%)	RMSE (%)
itcSegment [27]	-14	20	-19	9	-29	18
Area-based Modelling [27]	-3	15	-21	8	-30	13
single-layer MCGC	-26	23	-58	17	-60	22
double-layer MCGC	39	22	-18	14	-21	16

of AGB and ACD. By comparison, in the original study by Coomes et al. in [27], the general area-based approach for estimating ACD, as per [80], produced predictions with a bias of -19% and RMSE of 20% (Table. III). The ITC approach in this study produced ACD predictions with a bias of -20% and RMSE of 26% (Table. III). When looking at the contributions of individual forest types, the methods in the original study produce closer estimates for Alluvial forest types, as a result of the overestimation by double-layer MCGC (Table. IV). In contrast, for both Kerangas and Sandstone forest, double-layer MCGC produces closer estimates to the field inventory than any of the other methods considered here.

Appplying a correction factor for each forest type reduced overall bias and error. For single-layer MCGC the correction factors were larger than 1 as expected given the algorithm underestimated ACD (Alluvial 1.32, Sandstone 2.40, Kerangas 2.30, Table. V). Double-layer MCGC had correction factors that showed an underestimation of ACD in Kerangas and Sandstone forest matched with a similar overestimation in alluvial forests (Alluvial 0.70, Sandstone 1.24, Kerangas 1.21,

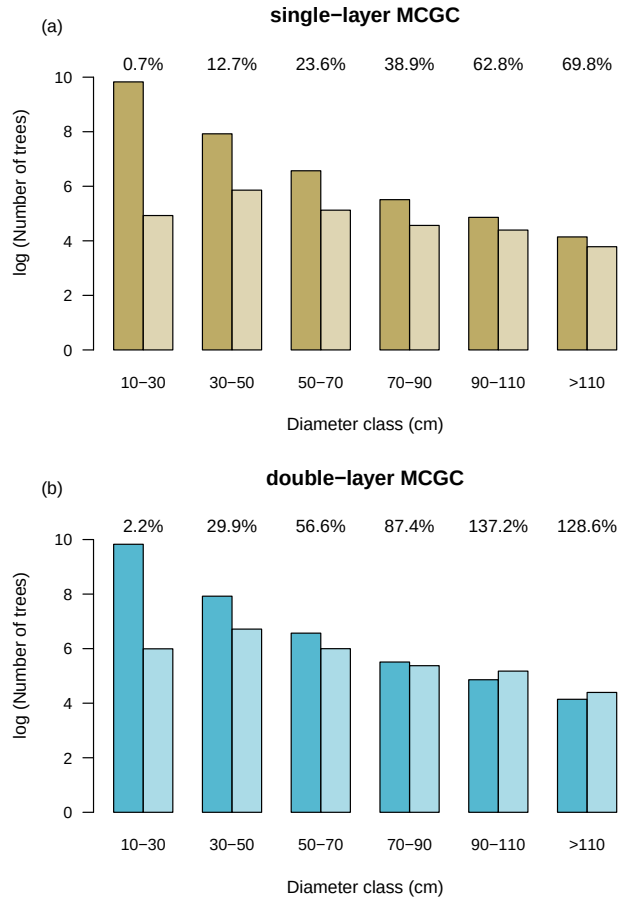

Fig. 6: Comparison of log-transformed numbers of trees by stem diameter for field inventory, left and dark, and each MCGC implementation, right and light. Percentages are given for MCGC as fraction of field inventory. MCGC methods are: (a) single-layer MCGC, (b) double-layer MCGC. Applying the second round of MCGC increases detection rates across all stem size groups and more than doubles the rate of detection in stems of a size of 90 cm or less.

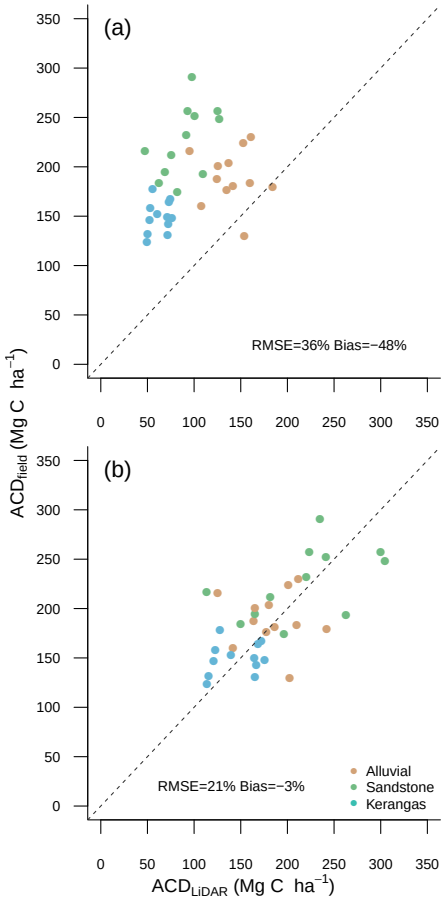

Fig. 7: Comparison of field-estimated above-ground carbon density (ACD_{field}) with estimates based on single-layer Multi-Class Graph Cut (ACD_{LiDAR}) for each of the 36 one-ha subplots. Panel (a) shows the original estimates with a 1:1 line (dashed). (b) shows the effect of applying a linear forest type specific correction factor to ACD_{LiDAR}.

TABLE V: Linear correction factor computed for each forest type when AGB estimates are compared to field inventory

Algorithm	Correction Factor Alluvial Forest	Kerangas Forest	Sandstone Forest
itcSegment [27]	1.07	1.32	1.30
single-layer MCGC	1.31	2.30	2.40
double-layer MCGC	0.70	1.21	1.24

Table. V). Applying this correction removed the bias for single-layer MCGC (3% vs 48%), but only reduced the bias within each forest type for double-layer MCGC as this already had low overall bias (Table. III). In both cases this reduced the RMSE as any bias for each forest type was reduced independently (21% vs 36% and 18% vs 33% for single-layer and double-layer MCGC). The effect of these corrections are similar for itcSegment, removing bias and reducing RMSE, as reported in [27] and shown in Table. III. When choosing to use a locally-calibrated model for the area-based approach the same effects are seen again. Here the model has all parameters for AGB as a function of remotely sensed variables directly fitted to the field inventory data, and the Gap Fraction height is chosen that best predicts Basal Area when compared to field inventory data [27]. Accordingly there is not an obvious analogue of the single correction factor that can be compared as is possible for itcSegment and MCGC (Table. V).

V. Discussion

A. Assessment of MCGC Performance

1) Forest inventory: When comparing the distribution of stem sizes found by single-layer MCGC to the reference field inventory it is clear that across all stem sizes the algorithm did not find all individual trees (Fig. 6a). The initial graph cut segmentation is constrained to find at least as many crowns as a simple maxima finding algorithm. As shown in Table. II, the itcSegment algorithm, which starts with the same local maxima approach finds more than 50% of the count of stems for trees with DBH>30 cm. These totals represent the number

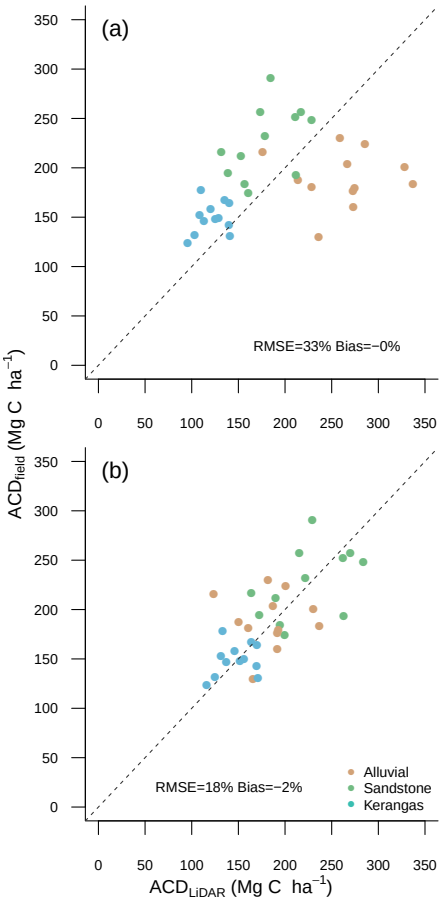

Fig. 8: Comparison of ACD estimates by plot for double-layer MCGC for each of the 36 one-ha subplots. ACD_{field} and ACD_{LiDAR} are the ACD estimates from the field survey and from MCGC applied to LiDAR data respectively. (a) shows the original estimates with a 1:1 line (dashed). (b) shows the effect of applying a linear forest type specific correction factor to ACD_{LiDAR}.

of local maxima initially found as itcSegment is constrained to delineate one tree for each local maximum. From this single-layer MCGC then found fewer stems than itcSegment. The graph cut step is constrained to find at least as many trees as the initial prior number of trees. Therefore the underestimation likely stems from the allometric filtering step. This step applies multiple criteria to each crown, based on knowledge of a regional allometry, rejecting those crowns that do not meet the criteria closely enough. This results in a reduction of the number of crowns, but those that remain are allometrically feasible. Notably, the algorithm finds a greater proportion of crowns relative to itcSegment in the largest classes (DBH>70 cm) than the smaller classes, suggesting that the rejection of crowns may be more common for medium and small trees.

Double-layer MCGC detects more tree crowns in all diameter classes (Fig. 6b and Table. II). These additional crowns are detected in the second application of the MCGC algorithm. The algorithm over-detects the number of trees in the largest diameter classes (DBH>90 cm), though there are fewer trees of these sizes, meaning the over count is not a large number of trees compared to the total count of all stems. As a trade-off, double-layer MCGC is able to detect more than double the number of trees in all small and medium classes (DBH<90 cm) when compared to the single-layer approach. This reduces the under counting in these diameter classes. This results from the 'stripping-off' of the dominant trees to allow the algorithm to better segment the under layers of the canopy. However the algorithm struggles with finding the smallest trees as is a drawback of working with ALS data [27]. The number of returns for these trees is very small and this underlines the importance of alternative approaches, such as using TLS data, for mapping the lowest layers of the canopy. Overall double-layer MCGC is able to detect most crowns for intermediate and large sized trees.

Results from the itcSegment approach in [27] justify the choice of limits for the choice of total tree stems. These limits are set to a minimum of the number of local maxima itcSegment would find and a maximum of twice this. From Table. II it is clear that itcSegment underestimates the number of crowns in all but the largest stem class. Therefore the total number calculated in this way will be an underestimate, though estimates well for trees of DBH>90 cm. Equally, for all stems where DBH>30 cm, itcSegment finds at least 50% of the total number of stems for each diameter class. Therefore doubling the number of local maxima found by the first step of this algorithm should over estimate the number of stems in all diameter classes where DBH>30 cm. This would then make a sensible upper limit for the number of crowns to seek for crowns of this size. For trees where DBH<30 cm all such approaches result in under-counting of stems. These trees are mostly found in the lower layers of the canopy and, as discussed below, this is a persistent problem with ALS data. Thus increasing the upper limit of crowns sought is more likely to lead to worsened over-segmentation of the largest crowns than to aid in finding these elusive smaller trees.

The difficulty of finding trees in the lowest layers of the forest canopy is a persistent problem in analysing ALS data in multi-layered canopies [26], [27], [34]. Here occlusion causes a diminishing number of returns for lower canopy, making it hard to identify the smallest trees in the subcanopy, even when inspecting the data visually [26], [34]. This highlights the importance of taking a combined approach to data collection, pairing ALS data acquisition with other methods such as TLS or field inventory. These methods themselves suffer from limitation in scope of the area they can cover when compared to ALS in the same time frame [7]–[9]. When working on locations of many hectares and larger, being able to automatically estimate the number of stems in the dominant canopy layers is a very useful tool and double-layer MCGC is able to provide this across several distinct forest types in the tropics.

2) Biomass estimation: Single-layer MCGC underestimated 1 ha plot-level carbon density (Fig. 7a). This is consistent with the under-detection of crowns discussed in Section V-A1. The bias of -48% shows that the estimates of biomass are about half of the values calculated based on field inventory, which is not unexpected. The algorithm finds more than half

of the largest trees, which contribute the most biomass. It then finds decreasing numbers of the smaller trees and thus half the biomass is not detected. Applying forest specific correction factors removed most of the bias (Fig. 7b). However, given the initial estimates were heavily biased, we feel this is not an appropriate additional step. Two of the resulting correction factors take values of greater than 2, indicating a doubling of the initial estimate, which would produce large uncertainties in the estimates.

Double-layer MCGC, by contrast, produced estimates of plot carbon density across the whole Sepilok landscape with low bias (Fig. 8a). However, as shown by applying correction factors there is a tendency towards bias in differing forest types. In the original study of this data from Sepilok in [27] neither the ITC nor the area-based approach was able to produce estimates free of bias without local calibration or correction (-20% and -19% respectively). The results from double-layer MCGC do have a slightly larger RMSE of 33% compared to 26% and 20% respectively in the original study. However, combining these with the bias of the methods suggests that double-layer MCGC produces more robust estimates of plot level carbon density across the landscape of Sepilok as a whole than either methods in the original study. This is in contrast to the conclusion there that area-based methods remained the best option. Further, applying a forest type specific correction to the estimates did reduce the relative error to be similar to that for similar corrections in this original study (18% compared to 18% and 13% respectively). However, for the ITC approaches we feel this correction isn't a fair reflection of the algorithm. This correction requires a local calibration to be made and this would only be possible for areas where full field inventory are available and are unlikely to generalise to other regions or biomes. In a similar manner, we feel the fairest way to compare the area-based approach is with the general equation originally used in [80]. Following the work in the original study on the data from Sepilok [27] this approach already requires local fitting of models for basal area and wood density, but the overall form of the equation is not locally fitted. As stated in Section III-E, correction factors for biased biomass estimation can be informative. Often these are greater than 1, where they account for missing contributions from under-segmented crowns, as per the intention in [27]. The need for correction in Kerangas and Sandstone forest arises from MCGC missing lower canopy trees, as discussed in Section V-A1. The need for correction in Alluvial plots arises instead from overestimation of AGB. Based on tree counts in Fig. 6 this is at least partially explained by a tendency to over-segment and over-count the top layer of the canopy which in turn inflates the estimate of biomass.

When comparing models based on the original model outputs, ignoring locally fitted models or forest-type correction factors, double-layer MCGC produces the lowest bias overall across all 36 one ha plots (Table. III). Additionally, double-layer MCGC produces the lowest bias predictions of ACD for both Kerangas and Sandstone forest types (Table. IV). Only in Alluvial forest is double-layer MCGC outperformed by the existing methods, though the potential difficulties in working in this forest type are discussed below. The overall best performing model was still the area-based modelling, once locally fitted. However, in a manner similar to applying a forest-type correction to ITC models, this approach requires local fitting of the overall equation for ACD, a local model for wood density and a choice of the best fitting model for predicting basal area as a function of gap fraction across a range of heights [27], [80]. Thus the fairest and most direct comparison is to look at the output of the most general models, which are also the models most likely to generalise to other biomes and forests (Table. IV). In this comparison, double-layer MCGC performs best across all but the Alluvial forest type. As noted in [27], this forest type is very distinctive. It is dominated by tall dipterocarps, which have narrow stems for their size compared to many other tropical tree families [27]. This leads to Alluvial forest having a high top canopy height, with low basal area. This leads to a low field measured ACD compared to canopy height when compared to other forest type as shown in [27]. MCGC estimates ACD based only on tree height and crown diameter so the unusual nature of Alluvial forest likely contributes to overestimation of biomass by MCGC. This is compounded by MCGC overestimating the number of trees with the largest size (Table. II) which are found in Alluvial plots. Though ALS remote sensing modelling of individual trees is likely to find similar difficulties with Alluvial forest and so developing an allometric model with further field inventory in these plots might be necessary to account for this atypical structure.

Overall, the reduced bias of biomass estimation from automated individual tree modelling across the Sepilok landscape is a powerful tool for the application of the MCGC individual crown detection algorithm. Our work used a regional allometric relationship (for the entire Indo-Malaya region) yet was able to produce initial estimates of biomass, before applying any correction factor, with lower bias than both the individual-tree region growing algorithm and area-based approaches used in [27]. This approach did require the calibrated model from [27] to link remote sensing measurements to biomass, though this is common for any such approach to use allometry to link crown or stem size to biomass. It is worth noting that the overestimated Alluvial forest type has a particularly tall and unusual canopy structure [27]. Many trees in this forest type sit towards the extreme end of the data in [56]. It could therefore be advisable to develop a more specialised allometry for this forest type for use in the allometric feasibility checking step. Similarly any atypical forests in a given region may be better approached with a more specific allometric relation than the very general region-wide model used in this work. This would be further justified by this being the only forest type for which double-layer MCGC does not produce the best estimate of ACD across the method compared in Table. IV.

B. Qualitative comparison with point cloud approaches

Though we were able to compare our approach to two existing methods to estimate ACD at the plot level, a more informative comparison would be to refer to existing point cloud tree detection algorithms. Unfortunately, accessing the existing implementation or implementing these is often tricky

and we consider a direct quantitative comparison beyond the scope of this work. Instead we qualitatively compare MCGC based on results for some of these approaches in other works and a comparison of the concepts and parameters used across methods and their ability to be transferred across study sites and biomes. MCGC aims to simultaneously address some potential individual limitations of other crown detection methods, focusing on tropical forests. As outlined in Section I, the goals were to work as much as possible with the raw point cloud and incorporate allometry without a need for local field inventory, whilst also remaining flexible and efficient.

*1) **Topographic correction**:* To the best of our knowledge, only the PTrees [47] and voxelised normalised cut in [48] cluster directly on the point cloud, without topographic correction. Our MCGC approach works on a hybrid of the raw and topographically corrected data. In computing linkages, the raw data is used to compare distances between points and to compute local density, only the radius for this computation uses topographic correction. Topographic correction is then used in the allometric feasibility checking, but only after the clustering is complete. This hybrid ensures warping (as discussed in [47]) from topographic correction doesn't influence the segmentation of candidate tree crowns. Thus, where terrain is steep, such as mountain environments, or tropical forests in valleys, we expect MCGC along with PTrees and an implementation of the approach [48] to fare best. However, only MCGC and PTrees work with the actual point cloud, with the work in [48] working on a summarised voxelised version of the data. This also works in a recursive binary approach, which is likely to cause splitting of subcanopy trees as discussed in [49]. The simplicity of PTrees, growing crowns by top-down k-nearest neighbours based on locally highest points, is a strong point of this method. However, extension of each label is based on a single initially labelled point which can introduce artefacts when extended across the full point cloud depending on how each crown grows. MCGC instead builds full models for each crown on subsampled data and then only uses imputation to extend these crowns to the full data. For these methods, clustering on data before topographic correction should improve accuracy on challenging and steep terrains, but feel combining this with a more advanced segmentation is a step forward that was yet to be addressed. MCGC progresses this goal with a hybrid point cloud approach, allowing conservation of geometry when clustering, but also allowing use of crown geometry and allometry.

*2) **Allometric and geometric constraints**:* Allometric or geometric constraints are common, yet often essential, to most cutting-edge approaches to point cloud based tree detection. These allow use of knowledge of typical crown size and shape to aid delineation, often at a local calibration cost. Approaches requiring local crown knowledge are likely to transfer less well. We believe a strength of MCGC is the use of a single allometric model fitted on an independent database. This simple yet effective relationship can be altered to move to other biomes and ecoregions. There are approaches that require no allometry, allowing easy transfer. Typically these are CHM raster approaches, where the regularity of the data allows simpler rules for defining crown boundaries. Using the point cloud, this regularity is lost, necessitating geometric or allometric constraints to find boundaries. Attempting to bridge this, [36] delineates on a CHM raster, only using the point cloud for each crown to look for understorey trees, only assuming a minimum of point density as a function of height will occur between trees. This requires no allometric or geometric constraints but only uses the point cloud in a refinement step. The full point cloud approach that we feel requires fewest geometric assumptions is in [31], which clusters trees based on k-means with only one geometric assumption; weighting horizontal and vertical distances differently. This is chosen for pine crown geometry from field data in the study and so may need adapting to different biomes. PTrees includes a greater number of geometric constraints on the crowns, but doesn't make use of local allometric data [47]. Crowns are scored based on four geometric criteria. This approach shouldn't need parameters tuning across biomes, but the validity of the scoring criteria may vary when moving. Stem locations are instead used in [48], which are based on assumptions about stem geometry and the height distribution of ALS returns. These assumptions are likely to hold in their original setting of coniferous stands, but the complex canopy in the tropics is likely to impair these. The recent meanshift approaches often make assumptions about crown geometry. In [39] the bandwidths of the kernel are set empirically based on field data. Similarly, in [42] bandwidths are set by hand, varying by forest type. In [40], a parameter relates the ratio of horizontal to vertical bandwidths of the kernel which is defined based on a parametric model for crown shape, requiring 5 parameters. These assumptions on tree geometry should not depend on the local allometry of trees, but as mentioned in many of these works, may only be valid for certain biomes. The alternative is use of allometric data about the biome or stand itself. In [44], which introduced the meanshift approach, an initial uniform kernel meanshift is iterated until the data falls into distinct layers. The heights of these strata are then used for the kernel, linking width as a ratio of height. The kernel is also biased in the positive z direction. The work in [45] builds on this with an adaptive kernel size that scales with height. This is derived from the data by a process using gradient based filtering and then geometric constraints on a CHM of the data to extract a subset of confidently identified crowns. The boundaries of these are used to extract point cloud data, and further geometric assumptions used to calculate the effective depth and width for each crown. Linear regression is then used to relate these to the height above ground to set scaling models for the kernel. Similarly in [41], after an initial segmentation, candidate crowns are grown by moving down vertical slices of the point cloud. From these crowns diameter estimates are used for horizontal bandwidth in each crown, with a tuneable ratio of this for the vertical bandwidth. These Allometric models require fewer strong assumptions on the geometry of crowns or stems in clustering, instead scaling clustering according to allometry. However, the automated allometry approaches require assumptions on geometry to derive the allometric relationships. In contrast, MCGC incorporates allometric scaling, in a way that avoids local measurement or complex derivation. We simply take a regional subset of a freely available global

database from [56] and use this to fit a power law of crown diameter as a function of height as in (4). This process can be repeated for any biome or ecoregion without the need for local data collection nor any further assumptions to be made about the data. This information is then used to guide the prior estimate of tree tops, the computation of local centroids and for allometric checking. The process of allometric checking does itself require assumptions, as discussed below, but this doesn't affect the clustering step. We believe the approaches of [45] and [41] to be clever and adaptive, but the reliance on prior assumptions of the crown geometry used in these derivations of allometry may need to adjust across regions, whereas our approach only requires use of a different subset of the same global database. We believe this potentially simpler approach to use of crown allometry to be a strength of MCGC.

*3) **Implementation, parameters and computation**:* One drawback of point cloud methods is their relative computational complexity and intensity. We compare methods based their number of parameters, the effort required to implement them and their computation complexity. We focus on the parameters of methods as a proxy for complexity, as accessing implementations for direct comparison and timing is hard. Implementation of most of the methods, including ours, is time consuming. Thus, we have released open-source software available at https://github.com/jonvw28/MCGC to reduce the effort to use MCGC. Some methods may be easier to implement, such as [31] and [47], based on the number of steps, but we feel that the work in [44], [45], [48] and [36] would be similarly difficult to implement owing to either the number of steps, or custom approaches. It is difficult to compare speeds of methods without standardised data, tasks and computational equipment. We expect methods using simpler clustering approaches, such as [31] and [36] to be the fastest, with methods using additonal checking steps slower, as in [47]. We then expect iterative clustering methods with computation at each step to be slower still, such as mean shift in [44] and [45]. Methods using matrix decomposition should take the longest as this operation is costly, such as graph cut in [48] and our work. The use of Nyström extension and acceleration in MCGC drastically reduces this disadvantage to keep our method competitive with other more complex clustering approaches. This is justified by the analysis in Appendix A which shows that the matrix step scales well. Although our method could be tricky to implement, the avaiability of the source code, along with computational acceleration places it fairly in competition with other advanced clustering approaches, such as AMS3D in [45].

The explicit parameters and implicit assumptions that are made by our method are detailed in Table. VI. There may appear to be many settings to choose, especially when contrasted with other methods that claim to only require one parameter. In practice, many of these are set to a standard value and not treated as a parameter, having paralogues in other methods, showing how common this large number of apparent parameters is. The assumptions in Table. VI mostly match similar assumptions in other methods. The limit for the number of crowns set by the prior was set independently of analysis and we would suggest this need not change across biomes. This is similar to the threshold cut value in [48] and [39], though our approach doesn't require explicit tuning. The variables K_H and K_Z are automatically set from the highest aboveground height of the data. Whilst this is adaptive, it is automatic, and is analogous to the process of initial vegetation layer stratification by automated segmentation in [44]. The radius for local centroid computation is set by the allometric lookup table. This is analogous to the assymetric kernel masking and distance computation of AMS3D in [44] and [45]. Similarly, the minimum proportion of points that must be within the allometric radius matches the percentile of the allometry. Both of these assumptions result from allometric relationships, being included in Table. VI for completeness. For computing tree top locations, the cut-off is 98% of the top height, used only to increase robustness by averaging over the tallest points, similar to using a smoothed CHM, such as in [36]. The maximum horizontal and vertical overlap between crowns are assumptions as explained in Section III-B3. These are analogous to the vertical cut-off of stems in [48] and the assumption on the crown base height in terms of a gaussian of heights in [45]. Similar are the testing of crown geometry in [47] and the valley finding approach to split canopy layers in [36]. Some explicit parameters, available to users in our implementation, take a set value and exist in most methods. Our prior uses a CHM grid with 0.5 m wide cells. The same parameter is either manually set, or assumed in [31], [36], [45], [48], or an analogous fixed bandwidth is often used in an initial segmentation as in [41], [44]. We use a height cut-off to reduce noise in the prior and to exclude ground returns in segmentation, but these are fixed. Most methods have a minimum height as in [31], [36], [39], [42], [44], [45], [47], [48] or use last returns to distinguish ground [41] and though [40] has no analogue, it finds that some vegetation segments include ground returns. MCGC also sets the minimum number of points in a crown and has 2 parameters for the DBSCAN step. These are set based on the point density, with minimum points equivalent to the number of returns expected for the footprint of a crown of 1 m radius (once rounded). The DBSCAN method uses a fixed neighbourhood size of 2 m, giving a projection-weighted radius of roughly 1 m in two dimensions, with the number of points set to be 10% of that for the minimum points. these could be tuned but we use two simple rules from point density, based on the points projected onto a crown of radius 1 m and then 10% of this for DBSCAN. These tests of size and connectivity are often used as a step in crown delineation methods, with minimum size used in [39], [47], [48] and connectivity used in [40]. These should be set based on point density relative to a reference point density of 24.6 points per m^2 in this study.

MCGC Parameters which aren't pre-set or dependant on point density consist of 4 parameters for the graph cut, the allometric relationships and subsampling. The four weight parameters constrain the importance of vertical and horizontal distances and local density variations ($_{XY}$, $_Z$, W_H and W_Z in Table. VI). These need to be chosen for each biome, but are likely to transfer well within these. In [31] there is one parameter weighting vertical distance relative to horizontal. In [48] there are also parameters weighting 4 components of the

TABLE VI: Summary of parameters and assumptions of MCGC used in this work. y denotes values which would not be expected to be changed to move to other regions or plots and were not subject to optimisation. z denotes values which were based on point density and should be scaled for other plots relative to the point density in this study of 24.6 points per m^2 See Section V-B for explanation of *.

Parameters		Assumptions	
Parameter	Value	Assumption	Value
Subsampling factor	5	maximimum number of trees relative to prior	2
Crown allometry for prior	2 coefficients in (4)*	K_H, Horizontal scale factor in w_{ij}	set by data
Grids size for CHM used in prior	0.5 m †	K_Z, Vertical scale factor in w_{ij}	set by data
Height cut-off of trees in prior	5 m †	Radius of ball for local centroid in w_{ij}	set by data
Height cut-off for points in MCGC and imputation	2 m †	Cut-off of height for tree-top location averaging	98% of top height
$_{XY}$ in w_{ij}	4	Min proportion of points within allometric boundary	95%*
$_Z$ in w_{ij}	2	Max proportion of points overlapping in allometry testing	60%
W_H in w_{ij}	0.2	Max vertical overlap of quantiles in allometry testing	25%
W_Z in w_{ij}	0.2		
Crown allometry for w_{ij} and feasibility checking	2 coefficients in (4)*		
Minimum points in a crown	100 ‡		
DBSCAN epsilon and number of points	2 m, 10 ‡		

graph cut. In [44] the bandwidth of the kernel needs heights in three strata and a conversion for horizontal extent, totalling four parameters, with two bounds set on the kernel extent. Extending this, [45] defines the kernel by two linear relationships of allometry, with the same restriction on extent. These relationships are derived from the data, becoming instead implicit parameters. Other Mean shift approaches require very similar bandwidth and kernel parameters: there are explicitly four parameters in the first mean shift in [42], with three bandwidths in the second pass; there are explicitly 2 parameters in the first mean shift in [41] with an adaptive bandwidth used later based on rough crown sizes; the bandwidth in [40] scales with height, requiring one parameter, but the kernel is a parametric model, requiring three explicit parameters; [39] only uses two parameters for meanshift, but also uses a normalised cut, requiring 4 parameters. In [47], instead there is an assumption that clustering should work down in order of point height, effectively imposing a bias towards tree tops. There is no obvious parallel in [36], but this method only uses the point cloud as a refinement step. We argue that MCGC requires comparable, or fewer, segmentation parameters to be set or tuned than most point cloud methods. Some CHM point cloud hybrid methods, or earlier clustering approaches, require fewer parameters, but these are not full point cloud approaches. The allometric relationships in Table. VI, and in (4), each appear to require two parameters. In practice, these are derived from the database in [56]. For any given region, the subset of this data should be used to update the models. The prior uses a 50th percentile regression model, and the only explicit parameter which should change is the percentile used to defined allometric extent. Here we use the 95th percentile. The database is global, so relationships can be set for any biome. Therefore despite having 4 implict parameters, in practice, once the biome is set, only the percentile for allometric checking should be considered changeable. This is comparable to the parameterisation used in other allometric methods. The allometry model in [45] has two implicit parameters, but this is automatically derived from the data whereby many assumptions or parameters are needed: first a gradient thresholding filter using seven step sizes with a threshold parameter, then testing the results of this for compactness and upper and lower bounds for height to width ratio, needing 3 more parameters. Ptrees in [47] uses scores of crown size, circularity, orientation and regularity. Although only one of these uses an explicit parameter, these are four strong assumptions. All three methods are comparable in effort for allometry. Finally, subsampling is explored in Appendix A. As a parameter this is something user can choose but is only used to accelerate the method and is something that should be considered an additional parameter as opposed to a core tuning parameter. We believe this addition to be rare in other similar approaches to working with the full point cloud, meaning it is not possible to draw parallels. Only [47] does any kind of imputation of results, but this is based solely on tree apices as opposed to MCGC imputing from pre-delineated subsampled data in which the structure of the data is used to find full trees, not just tree tops.

*4) **Biomass estimation and direct comparisons**:* Estimating plot level biomass is not common with automated tree delineation but is key for forest inventory monitoring. We now contrast Section V-A2 to AMS3D in [45], where hectare scale biomass was computed for a tropical lowland forest in Panama, which is a similar to Sepilok, though in a different ecoregion. Comparing biases isn't possible as the models in [45] are reported after application of a correction factor. Across the 50 ha study site, models for biomass of trees based on crown height and volume, crown height and area and crown height and DBH (modelled from crown height and area) produced estimates of biomass at one hectare scale with RMSE of 16.9%, 13.8% and 15% respectively. This is best compared to the RMSE values in table IV, as these are each for a single forest type, as is the case in [45]. Double-layer MCGC produces estimates with RMSE of 22%, 14% and 16% for Alluvial, Kerangas and Sandstone forest. This result did depend on a local model to link remote sensing measurements to biomass from [27], but without any additional correction factor. Alluvial forest has the highest RMSE, but this forest type is very distinctive so may not be best for comparison. For the latter two forest types, MCGC RMSE values are comparable with those of AMS3D

showing similar uncertainty, though without testing on the same data it is not possible to draw quantitative conclusions.

MCGC and AMS3D have been directly compared for tree detection in another work [81]. In [81], six 6.25 ha lowalnd tropical forest plots of LiDAR data from French Guiana had their crowns delineated by several methods, including an earlier version of MCGC, itcSegment and AMS3D. In this, MCGC and AMS3D were found to outperform any methods not working with the full point cloud. This study focused on two assessments. The first of these were a set of 1598 crowns that are visible from the top of the canopy which were manually delineated in the field, for which methods were assessed in terms of overlap of automated crowns mapped to each manual crown. In this comparison AMS3D outperformed MCGC, with a higher proportion of crowns exceeding the threshold of Jaccard Index (73.8% compared to 54.3%). In this test MCGC was also comparable to two CHM and raster approaches, but these should do well on top canopy layers. Assessment of all canopy layers was done with an algorithm to pair detected crowns with field inventory stems. Then a model was used to predict DBH for each crown from their height and area. Accuracy was assessed in terms of RMSE for DBH predictions across the largest 5000 trees found by each algorithm. Here MCGC was the most accurate with RMSE of 7.64% compared to 7.67% for AMS3D, which was better than all other methods. When only the trees from the congruence analysis were used AMS3D performed better than MCGC (8.92% vs 9.33%), though itcSegment did better than both (8.41%) suggesting a hybrid approach with this for the top canopy, and a point cloud approach for lower layers might give even better results. Looking at how delineated crowns mapped to field inventory stems, MCGC showed its strength. For the manually delineated dataset MCGC found a match for 67% of crowns compared to 63% for both AMS3D and itcSegment, beating all other methods in the study. Across the full study data, AMS3D delienated the most crowns with an average of 2564 per plot, compared to 1832 for MCGC and less than 1500 for all other methods. However, of these, only 37% of crowns produced by AMS3D matched a stem in the automated algorithm, compared to 82% for MCGC. These percentages were beaten by three of the remaining four methods, but these methods all delineated 75% or fewer of the number of crowns MCGC did, and far fewer than AMS3D. As concluded in [81] overall, AMS3D showed best congruence on reference upper canopy crowns, but MCGC did best when all delineated crowns were mapped to field inventory stems. We conclude that MCGC is competitive with AMS3D in an independent study, with strengths in successful detection of allometrically valid crowns across all layers of the canopy.

C. Possible extensions of MCGC

Double-layer MCGC is a very flexible approach to the problem of tree detection and biomass estimation. One obvious extension, which is in development, is to include spectral imagery data or other data which can be assigned to each point. This can easily be incorporated into (8) in a manner similar to the current terms. With a suitable dataset of an RGB or multispectral point cloud data combined with field inventory it should be possible to refine the delineation of crowns as part of MCGC. Similar modifications to account for return number or intensity are possible, all of which require modification only of the formula for linkage weights. The key strength of MCGC in this regard is the flexibility of the similarity computation. It is possible to alter the w_{ij} term to reflect which factors are felt to be relevant in detecting crowns in any given forest.

Though double-layer MCGC still does not find all lower canopy trees, the balance of trees that it does find give rise to a good estimate of biomass at plot-level. These should combine well for analysis of UAV data, for which the lower layers of the canopy are often obscured and so do not appear in the structure from motion reconstruction. Automatic production of estimates of total landscape biomass from UAV point clouds using only existing database models for tree allometries will be incredibly powerful for low cost, large scale project management. MCGC offers progress towards this capability, and bringing in a spectral element in way similar to our previous work in [49], [58], as is present in UAV point clouds or from co-registering hyperspectral data as in our work in [59], should strengthen this further. A further extension to combine a CHM approach, such as itcSegment, to first remove the top trees from the canopy may improve speed and accuracy of biomass estimation.

VI. Conclusion

Our proposed double-layer MCGC approach to ITC detection and biomass prediction was able to identify the dominant trees of the canopy across three distinct tropical forest types. As shown in an independent benchmark, MCGC has strength in both detection and allometric accuracy of crowns in middle and lower canopy layers. From the crowns of these trees MCGC can estimate hectare-scale biomass and carbon density in tropical forest, advancing previous work suggesting ITC approaches still need development to out-perform simpler area-based modelling. Uncertainty in estimating biomass was similar to that reported for AMS3D, another point cloud method, underlining the potential of such approaches, at the plot, but not yet national, scale. We used a region-wide allometric relationship, which could be applied to any forest in the region. Similar approaches can be used to generate allometric relationships for any biome of interest, using only a previously derived allometric model to then link crown geometry to biomass. This is in contrast to area-based approaches, requiring either field measurements or local calibration.

MCGC also simultaneously incorporates multiple aspects which are not commonly all included in tree crown detection algorithms. MCGC is one of few methods which cluster directly on the raw point cloud, rather than topographically corrected data, which skews trees on sloped ground, though does use this information to guide allometric feasibility. MCGC is able to incorporate allometric constraints through a single relationship of crown height to diameter, which can be derived from existing databases. MCGC is also highly flexible and tuneable through a core set of four weight parameters and an allometric relationship whilst not being specific to any one

forest type, though is able to use locally derived allometric relationships. These are made deliberately open to the user for tuning in our openly available implementation of MCGC and as discussed correspond to a similar number of explicit and implicit parameters in many of the cutting edge methods for point cloud tree detection. All of this together makes MCGC a powerful tool in remote sensing of forests, supported by ability for biomass estimation. MCGC also has potential for easy extension to multi-sensor datasets. Flexibility in the the construction of weights allows any combination of variables to be considered in detecting crowns and in particular this is well-suited to UAV-derived structure from motion data where the point cloud includes spectral data. A similar approach could be applied to the combination of hyperspectral imagery and LiDAR data once datasets have been aligned, meaning MCGC can be quickly adapted to new data sources.

APPENDIX
COMPUTATIONAL STABILITY AND SPEED OF SUBSAMPLING

We assessed the effect of varying the subsample factor in the second pass of MCGC in our approach. Trees in the lower canopy normally produce fewer ALS returns because of occlusion by upper canopy layers. It may then be appropriate to reduce subsampling in the second pass of MCGC to account for the sparser nature of the data available. We tried varying the subsampling in the second pass of MCGC on a single random one ha plot from Sepilok (which was of the alluvial forest type). We subsampled by factors ranging from 2 through to 10 and ran MCGC. This was repeated 100 times for each subsample to see stability of biomass estimates and compute average time taken for processing a one ha plot. The motivation to work only on the second pass was the presumed lower point density, but in principle this should give an idea of the effect of varying subsampling for the complete pipeline.

The time to complete the second pass (and by extension the total time for MCGC) increases with the proportion of points kept (Fig. 9(a)). The graph cut step (up to the k-means step in Fig. 1) grows with the number of points, but only in a range from 30 s to 60 s. The costly step is allometric checking. This is seen by the difference between the curves in Fig. 9(a). The roughly linear trend corresponds to roughly exponential growth of processing time for the complete algorithm. Once 20% or less of the data is used (a subsample factor of 5 or greater) the run time stabilises in the range of 50 s to 85 s and further subsampling has less effect. The costly allometric step is the trimming of over-sized trees. This process occurs sequentially, and so should grow linearly with number of crowns. Instead subsampling affects the number of points considered for each crown. Though not tested here, this should mean run time for allometric testing for a fixed subsample factor should grow linearly with area. Further, as each crown is handled independently, this process can be run in parallel which allows further acceleration, though we have not implemented this. The empirical exponential growth of computational time with effective point density justifies subsampling, as long as results remain stable.

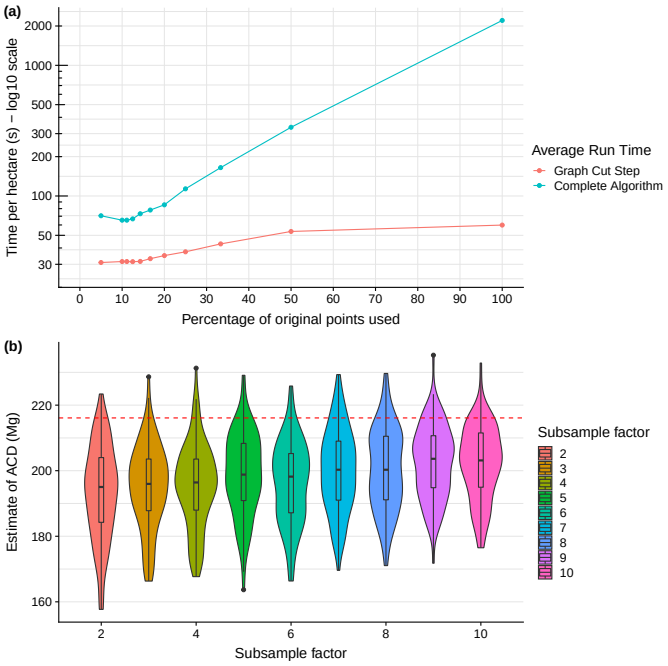

Fig. 9: Comparison of run time and stability of ACD estimation for subsampling in the second pass of MCGC for 100 replicates of a single one ha plot. (a) shows time taken for this pass of MCGC decreases as the percentage of points used reduces. The time axis has a $\log_{10}$ scale. (b) shows the distribution of final ACD estimates for the plot for different subsample factors (a factor of 5 meaning only 1 in 5 points are kept). The red dashed line shows the ACD based on field inventory.

Estimation of ACD showed experimental stability for subsample factors of 5 or lower, using 20% or more of points (Fig. 9b). For this plot, the MCGC double-layer approach slightly underestimates the ACD based on field inventory for all subsampling routines (Fig. 9b), where replication of different randomly subsampled points allows distribution of ACD estimates to be measured. Here the means are roughly 20 Mgha^{1} below reference ACD. The output for subsampling factors of 5 or lower have similar means with distributions with highest support at a similar value of ACD. Once this factor exceeds 5, the distribution starts to show bimodality, before becoming shifted to higher ACD estimates. This likely comes from using too little of the data to reliably build a good map of trees, where sparsity makes density based approaches unreliable. The allometric constraints are also less meaningful as outlier points effectively have greater significance to the overall assessment. This affects the upsampling step and outliers can inflate the apparent extent of crowns when imputing points, leading to an overestimation of biomass.

Combining the results above, we believe a subsampling factor of 5 is justified for the data from Sepilok.This produces speeds at the tail end of the exponential decrease in run time in Fig. 9a, but also sits within the range of values in Fig. 9b for which results appear to be more stable. Further subsampling wouldn't produce a drastic improvement in run

time, but appears likely to risk introduction of artefacts from the subsampling approach. Reducing subsampling shouldn't have a large effect on stability of ACD estimation, but will drastically increase run time.

ACKNOWLEDGEMENTS

We thank members of the NERC Airborne Remote Sensing Facility and NERC Data Analysis Node for collecting and processing the data (project code MA14-14). David Coomes was supported by an International Academic Fellowship from the Leverhulme Trust. We thank Oliver Phillips and Simon Lewis who provided census data collected as part of an ERC Advanced Grant (T-Forces). Thanks to Richard Bryan Sebastian and Reuben Nilus for collecting field data on delineated trees and to Lindsay Banin for collecting data on field measured allometry. We gratefully acknowledge the support of NVIDIA Corporation with the donation of a Quadro P6000 GPU used for this research.

REFERENCES

[1] M. Larsen, M. Eriksson, X. Descombes, G. Perrin, T. Brandtberg, and F. A. Gougeon, "Comparison of six individual tree crown detection algorithms evaluated under varying forest conditions," *International Journal of Remote Sensing*, vol. 32, no. 20, pp. 5827–5852, 2011.

[2] H. Kaartinen, J. Hyyppä, X. Yu, M. Vastaranta, H. Hyyppä, A. Kukko, M. Holopainen, C. Heipke, M. Hirschmugl, F. Morsdorf, E. Næsset, J. Pitkänen, S. Popescu, S. Solberg, B. M. Wolf, and J.-C. Wu, "An International Comparison of Individual Tree Detection and Extraction Using Airborne Laser Scanning," *Remote Sensing*, vol. 4, no. 4, pp. 950–974, 2012.

[3] M. Dalponte, H. O. Ørka, L. T. Ene, T. Gobakken, and E. Næsset, "Tree crown delineation and tree species classification in boreal forests using hyperspectral and ALS data," *Remote Sensing of Environment*, vol. 140, pp. 306–317, 2014.

[4] 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*, vol. 6, no. 5, pp. 1721–1747, 2015.

[5] Z. Zhen, L. J. Quackenbush, and L. Zhang, "Trends in Automatic Individual Tree Crown Detection and Delineation—Evolution of LiDAR Data," *Remote Sensing*, vol. 8, no. 4, 2016.

[6] E. Lindberg and J. Holmgren, "Individual Tree Crown Methods for 3D Data from Remote Sensing," *Current Forestry Reports*, 2017.

[7] Y. Xie, Z. Sha, and M. Yu, "Remote sensing imagery in vegetation mapping: a review," *Journal of Plant Ecology*, vol. 1, no. 1, pp. 9–23, mar 2008.

[8] Z. I. Petrou, I. Manakos, and T. Stathaki, "Remote sensing for biodiversity monitoring: a review of methods for biodiversity indicator extraction and assessment of progress towards international targets," *Biodiversity and Conservation*, vol. 24, no. 10, pp. 2333–2363, sep 2015.

[9] C. Toth and G. Jóźków, "Remote sensing platforms and sensors: A survey," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 115, pp. 22–36, may 2016.

[10] G. P. Asner, R. F. Hughes, P. M. Vitousek, D. E. Knapp, T. Kennedy-Bowdoin, J. Boardman, R. E. Martin, M. Eastwood, and R. O. Green, "Invasive plants transform the three-dimensional structure of rain forests," *Proceedings of the National Academy of Sciences*, vol. 105, no. 11, pp. 4519–4523, 2008.

[11] H.-E. Andersen, S. E. Reutebuch, R. J. McGaughey, M. V. N. D'Oliveira, and M. Keller, "Monitoring selective logging in western Amazonia with repeat lidar flights," *Remote Sensing of Environment*, vol. 151, pp. 157–165, 2014.

[12] K. Y. van Ewijk, C. F. Randin, P. M. Treitz, and N. A. Scott, "Predicting fine-scale tree species abundance patterns using biotic variables derived from LiDAR and high spatial resolution imagery," *Remote Sensing of Environment*, vol. 150, pp. 120–131, 2014.

[13] C. Barnes, H. Balzter, K. Barrett, J. Eddy, S. Milner, and J. C. Suárez, "Individual Tree Crown Delineation from Airborne Laser Scanning for Diseased Larch Forest Stands," *Remote Sensing*, vol. 9, no. 3, p. 231, 2017.

[14] Q. Chen, D. D. Baldocchi, and P. Gong, "Isolating Individual Trees in a Savanna Woodland Using Small Footprint Lidar Data," *Photogrammetric Engineering & Remote Sensing*, vol. 72, no. 8, pp. 923–932, 2006.

[15] B. Koch, U. Heyder, and H. Weinacker, "Detection of Individual Tree Crowns in Airborne Lidar Data," *Photogrammetric Engineering & Remote Sensing*, vol. 72, no. 4, pp. 357–363, 2006.

[16] D. Zhao, Y. Pang, Z. Li, and L. Liu, "Isolating individual trees in a closed coniferous forest using small footprint lidar data," *International Journal of Remote Sensing*, vol. 35, no. 20, pp. 7199–7218, 2014.

[17] T. Liu, J. Im, and L. J. Quackenbush, "A novel transferable individual tree crown delineation model based on Fishing Net Dragging and boundary classification," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 110, pp. 34–47, 2015.

[18] S. Solberg, E. Naesset, and O. M. Bollandsås, "Single Tree Segmentation Using Airborne Laser Scanner Data in a Structurally Heterogeneous Spruce Forest," *Photogrammetric Engineering & Remote Sensing*, vol. 72, no. 12, 2006.

[19] Z. Zhen, L. J. Quackenbush, S. V. Stehman, and L. Zhang, "Agent-based region growing for individual tree crown delineation from airborne laser scanning (ALS) data," *International Journal of Remote Sensing*, vol. 36, no. 7, pp. 1965–1993, 2015.

[20] M. Dalponte and D. A. Coomes, "Tree-centric mapping of forest carbon density from airborne laser scanning and hyperspectral data," *Methods in Ecology and Evolution*, vol. 7, no. 10, pp. 1236–1245, 2016.

[21] F. A. Gougeon, "A Crown-Following Approach to the Automatic Delineation of Individual Tree Crowns in High Spatial Resolution Aerial Images," *Canadian Journal of Remote Sensing*, vol. 21, no. 3, pp. 274–284, 1995.

[22] D. G. Leckie, F. A. Gougeon, S. Tinis, T. Nelson, C. N. Burnett, and D. Paradine, "Automated tree recognition in old growth conifer stands with high resolution digital imagery," *Remote Sensing of Environment*, vol. 94, no. 3, pp. 311–326, 2005.

[23] M. Katoh and F. A. Gougeon, "Improving the Precision of Tree Counting by Combining Tree Detection with Crown Delineation and Classification on Homogeneity Guided Smoothed High Resolution (50 cm) Multispectral Airborne Digital Data," *Remote Sensing*, vol. 4, no. 5, pp. 1411–1424, 2012.

[24] M. Wulder, K. Niemann, and D. G. Goodenough, "Local Maximum Filtering for the Extraction of Tree Locations and Basal Area from High Spatial Resolution Imagery," *Remote Sensing of Environment*, vol. 73, no. 1, pp. 103–114, 2000.

[25] J. Hyyppa, O. Kelle, M. Lehikoinen, and M. Inkinen, "A segmentation-based method to retrieve stem volume estimates from 3-D tree height models produced by laser scanners," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 5, pp. 969–975, 2001.

[26] H. Hamraz, M. A. Contreras, and J. Zhang, "Forest understory trees can be segmented accurately within sufficiently dense airborne laser scanning point clouds," *Scientific Reports*, vol. 7, no. 1, p. 6770, 2017.

[27] D. A. Coomes, M. Dalponte, T. Jucker, G. P. Asner, L. F. Banin, D. F. Burslem, S. L. Lewis, R. Nilus, O. Phillips, M.-H. Phuag, and L. Qiee, "Area-based vs tree-centric approaches to mapping forest carbon in Southeast Asian forests with airborne laser scanning data," *Remote Sensing of Environment*, vol. 194, pp. 77–88, 2017.

[28] S. C. Popescu and K. Zhao, "A voxel-based lidar method for estimating crown base height for deciduous and pine trees," *Remote Sensing of Environment*, vol. 112, no. 3, pp. 767–781, 2008.

[29] Y. Wang, H. Weinacker, and B. Koch, "A Lidar Point Cloud Based Procedure for Vertical Canopy Structure Analysis And 3D Single Tree Modelling in Forest," *Sensors*, vol. 8, no. 6, pp. 3938–3951, 2008.

[30] B. Wu, B. Yu, W. Yue, S. Shu, W. Tan, C. Hu, Y. Huang, J. Wu, and H. Liu, "A Voxel-Based Method for Automated Identification and Morphological Parameters Estimation of Individual Street Trees from Mobile Laser Scanning Data," *Remote Sensing*, vol. 5, no. 2, pp. 584–611, 2013.

[31] F. Morsdorf, E. Meier, B. Allgöwer, and D. Nüesch, "Clustering in Airborne Laser Scanning Raw Data for Segmentation of Single Trees," in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 34, 2003, pp. 27–33.

[32] S. Gupta, H. Weinacker, and B. Koch, "Comparative Analysis of Clustering-Based Approaches for 3-D Single Tree Detection Using Airborne Fullwave Lidar Data," *Remote Sensing*, vol. 2, no. 4, pp. 968–989, 2010.

[33] E. Lindberg, L. Eysn, M. Hollaus, J. Holmgren, and N. Pfeifer, "Delineation of Tree Crowns and Tree Species Classification From Full-Waveform Airborne Laser Scanning Data Using 3-D Ellipsoidal Clustering," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 7, pp. 3174–3181, jul 2014.

[34] J. Vauhkonen, L. Ene, S. Gupta, J. Heinzel, J. Holmgren, J. Pitkänen, S. Solberg, Y. Wang, H. Weinacker, K. M. Hauglin, V. Lien, P. Packalén, T. Gobakken, B. Koch, E. Næsset, T. Tokola, and M. Maltamo, "Comparative testing of single-tree detection algorithms under different types of forest," *Forestry: An International Journal of Forest Research*, vol. 85, no. 1, pp. 27–40, 2012.

[35] C. Véga and S. Durrieu, "Multi-level filtering segmentation to measure individual tree parameters based on Lidar data: Application to a mountainous forest with heterogeneous stands," *International Journal of Applied Earth Observation and Geoinformation*, vol. 13, no. 4, pp. 646–656, 2011.

[36] L. I. Duncanson, B. D. Cook, G. C. Hurtt, and R. O. Dubayah, "An efficient, multi-layered crown delineation algorithm for mapping individual tree structure across multiple ecosystems," *Remote Sensing of Environment*, vol. 154, pp. 378–386, 2014.

[37] L. Jing, B. Hu, T. Noland, and J. Li, "An individual tree crown delineation method based on multi-scale segmentation of imagery," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 70, pp. 88–98, 2012.

[38] B. Hu, J. Li, L. Jing, and A. Judah, "Improving the efficiency and accuracy of individual tree crown delineation from high-density LiDAR data," *International Journal of Applied Earth Observation and Geoinformation*, vol. 26, pp. 145–155, 2014.

[39] N. Amiri, W. Yao, M. Heurich, P. Krzystek, and A. K. Skidmore, "Estimation of regeneration coverage in a temperate forest by 3d segmentation using airborne laser scanning data," *International Journal of Applied Earth Observation and Geoinformation*, vol. 52, pp. 252 – 262, 2016.

[40] W. Xiao, S. Xu, S. O. Elberink, and G. Vosselman, "Individual tree crown modeling and change detection from airborne lidar data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 8, pp. 3467–3477, Aug 2016.

[41] X. Hu, W. Chen, and W. Xu, "Adaptive mean shift-based identification of individual trees using airborne lidar data," *Remote Sensing*, vol. 9, no. 2, 2017.

[42] W. Dai, B. Yang, Z. Dong, and A. Shaker, "A new method for 3d individual tree extraction using multispectral airborne lidar point clouds," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 144, pp. 400 – 411, 2018.

[43] W. Xiao, A. Zaforemska, M. Smigaj, Y. Wang, and R. Gaulton, "Mean shift segmentation assessment for individual forest tree delineation from airborne lidar data," *Remote Sensing*, vol. 11, no. 11, 2019.

[44] A. Ferraz, F. Bretar, S. Jacquemoud, G. Gonçalves, L. Pereira, M. Tomé, and P. Soares, "3-D mapping of a multi-layered Mediterranean forest using ALS data," *Remote Sensing of Environment*, vol. 121, pp. 210–223, 2012.

[45] A. Ferraz, S. Saatchi, C. Mallet, and V. Meyer, "Lidar detection of individual tree size in tropical forests," *Remote Sensing of Environment*, vol. 183, pp. 318–333, 2016.

[46] A. Ferraz, S. Saatchi, C. Mallet, S. Jacquemoud, G. Gonçalves, C. A. Silva, P. Soares, M. Tomé, and L. Pereira, "Airborne Lidar Estimation of Aboveground Forest Biomass in the Absence of Field Inventory," *Remote Sensing*, vol. 8, no. 8, pp. 1–18, 2016.

[47] C. Vega, A. Hamrouni, S. E. Mokhtari, J. Morel, J. Bock, J.-P. Renaud, M. Bouvier, and S. Durrieu, "PTrees: A point-based approach to forest tree extraction from lidar data," *International Journal of Applied Earth Observation and Geoinformation*, vol. 33, no. 1, pp. 98–108, 2014.

[48] J. Reitberger, C. Schnörr, P. Krzystek, and U. Stilla, "3D segmentation of single trees exploiting full waveform LIDAR data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 64, no. 6, pp. 561–574, nov 2009.

[49] J. Lee, D. Coomes, C.-B. Schonlieb, X. Cai, J. Lellmann, M. Dalponte, Y. Malhi, N. Butt, and M. Morecroft, "A graph cut approach to 3D tree delineation, using integrated airborne LiDAR and hyperspectral imagery," jan 2017, arXiv manuscript.

[50] W. Yao and Y. Wei, "Detection of 3-D Individual Trees in Urban Areas by Combining Airborne LiDAR Data and Imagery," *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 6, pp. 1355–1359, nov 2013.

[51] Y. Yu, J. Li, H. Guan, C. Wang, and J. Yu, "Semiautomated Extraction of Street Light Poles From Mobile LiDAR Point-Clouds," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 3, pp. 1374–1386, mar 2015.

[52] B. Yang, W. Dai, Z. Dong, and Y. Liu, "Automatic forest mapping at individual tree levels from terrestrial laser scanning point clouds with a hierarchical minimum cut method," *Remote Sensing*, vol. 8, no. 5, 2016.

[53] L. Zhong, L. Cheng, H. Xu, Y. Wu, Y. Chen, and M. Li, "Segmentation of Individual Trees from TLS and MLS Data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 2, pp. 774–787, 2017.

[54] W. Yao, P. Krzystek, and M. Heurich, "Tree species classification and estimation of stem volume and DBH based on single tree extraction by exploiting airborne full-waveform LiDAR data," *Remote Sensing of Environment*, vol. 123, pp. 368–380, aug 2012.

[55] J. Heinzel and M. O. Huber, "Constrained spectral clustering of individual trees in dense forest using terrestrial laser scanning data," *Remote Sensing*, vol. 10, no. 7, 2018.

[56] T. Jucker, J. Caspersen, J. Chave, C. Antin, N. Barbier, F. Bongers, M. Dalponte, K. Y. Ewijk, I. Forrester David, M. Haeni, S. I. Higgins, R. J. Holdaway, Y. Iida, C. Lorimer, P. L. Marshall, S. Momo, G. R. Moncrieff, P. Ploton, L. Poorter, K. A. Rahman, M. Schlund, B. Sonké, F. J. Sterck, A. T. Trugman, V. A. Usoltsev, M. C. Vanderwel, P. Waldner, B. M. M. Wedeux, C. Wirth, H. Wöll, M. Woods, W. Xiang, N. E. Zimmermann, and D. A. Coomes, "Allometric equations for integrating remote sensing imagery into forest monitoring programmes," *Global Change Biology*, vol. 23, no. 1, pp. 177–190, 2017.

[57] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, aug 2000.

[58] J. Lee, X. Cai, J. Lellmann, M. Dalponte, Y. Malhi, N. Butt, M. Morecroft, C. B. Schönlieb, and D. A. Coomes, "Individual Tree Species Classification From Airborne Multisensor Imagery Using Robust PCA," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 6, pp. 2554–2567, jun 2016.

[59] J. Lee, X. Cai, C. Schönlieb, and D. A. Coomes, "Nonparametric image registration of airborne lidar, hyperspectral and photographic imagery of wooded landscapes," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 11, pp. 6073–6084, Nov 2015.

[60] J. B. Drake, R. G. Knox, R. O. Dubayah, D. B. Clark, R. Condit, J. B. Blair, and M. Hofton, "Above-ground biomass estimation in closed canopy Neotropical forests using lidar remote sensing: factors affecting the generality of relationships," *Global Ecology and Biogeography*, vol. 12, no. 2, pp. 147–159, 2003.

[61] H.-E. Andersen, R. J. McGaughey, and S. E. Reutebuch, "Estimating forest canopy fuel parameters using LIDAR data," *Remote Sensing of Environment*, vol. 94, no. 4, pp. 441–449, 2005.

[62] G. P. Asner, G. V. N. Powell, J. Mascaro, D. E. Knapp, J. K. Clark, J. Jacobson, T. Kennedy-Bowdoin, A. Balaji, G. Paez-Acosta, E. Victoria, L. Secada, M. Valqui, and R. F. Hughes, "High-resolution forest carbon stocks and emissions in the Amazon," *Proceedings of the National Academy of Sciences*, vol. 107, no. 38, pp. 16 738–16 742, sep 2010.

[63] G. P. Asner, J. Mascaro, H. C. Muller-Landau, G. Vieilledent, R. Vaudry, M. Rasamoelina, J. S. Hall, and M. van Breugel, "A universal airborne LiDAR approach for tropical forest carbon mapping," *Oecologia*, vol. 168, no. 4, pp. 1147–1160, 2012.

[64] M. H. Nunes, R. M. Ewers, E. C. Turner, and D. A. Coomes, "Mapping Aboveground Carbon in Oil Palm Plantations Using LiDAR: A Comparison of Tree-Centric versus Area-Based Approaches," *Remote Sensing*, vol. 9, no. 8, 2017.

[65] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[66] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. Springer, 2009.

[67] Z. Wu and R. Leahy, "An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 15, no. 11, pp. 1101–1113, 1993.

[68] J. Lee, X. Cai, C. B. Schoenlieb, and D. Coomes, "Mapping individual trees from airborne multi-sensor imagery," in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, jul 2015, pp. 5411–5414.

[69] A. Little and A. Byrd, "A Multiscale Spectral Method for Learning Number of Clusters," in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, dec 2015, pp. 457–460.

[70] A. Azran and Z. Ghahramani, "Spectral Methods for Automatic Multiscale Data Clustering," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 1, jun 2006, pp. 190–197.

[71] W. Kong, C. Sun, S. Hu, and J. Zhang, "Automatic spectral clustering and its application," in *2010 International Conference on Intelligent*