Table des matières

Remerciements

J'aimerais remercier tout d'abord mon directeur Alexis Achim pour son indéfectible optimisme et son enthousiasme quotidien. Sans cette source de motivation ces trois dernières années auraient sûrement parues plus longues et moins agréables.

Merci ensuite à Martin Béland pour son aide et pour les discussions pertinentes que nous avons eues sur mes travaux et qui ont permis une considérable amélioration des analyses. Merci également à John Caspersen qui a accepté au départ que le projet se déroule à l'Université Laval dans un environnement qui me convenait bien. Je lui suis très reconnaissant d'avoir fait passer mes intérêts avant les siens.

Un remerciement particulier va aussi à David Auty, relecteur officieux de la documentation officielle du package `lidR`. Documenter du code est objectivement un travail pénible et laborieux. Sans lui, je n'aurais pu soumettre le package au CRAN.

Remerciement aussi à Nicholas Coops de m'avoir invité à Vancouver pour travailler dans son labo, mais aussi bien sûr pour m'avoir proposé une bourse importante pour le développement du package `lidR`. Cette bourse a été utilisée pour engager une stagiaire pour continuer le développement de package. Ce projet de recherche a été financé par le projet AWARE (Assessement of Wood Attributes for Remote sEnsing). Je tiens à remercier cette structure pour sa confiance et son appui.

Merci aussi à Tina Cormier de m'avoir invité spontanément à FOSS4G, une conférence internationale annuelle sur le logiciel libre pour le traitement de données géospatiales où j'ai eu la chance de présenter le package `lidR` et de rentrer « pour vrai » dans le milieu du développement de logiciel open source.

Enfin, j'aimerais exprimer ma reconnaissance envers ma famille, mes collègues et mes amis qui ont contribué, à leur manière, à la réalisation de cette thèse. Claude, Manu, Anne, Kaysandra et André, pour avoir partagé tous ces moments avec moi. Jöelle, devenue notre pro d'Inkscape, pour ses illustrations de qualité. Merci à ma maman, qui a patiemment révisé la plupart de mes textes et corrigé tous (oui *tous*!) les "s" au pluriel manquants. Alexis te remercie aussi pour ce travail ascétique. Et bien sûr, un grand merci à Nanou, ma copine et bientôt la mère de mon enfant, qui m'a appuyé dans ce projet en s'expatriant vers cette contrée glacée où nous avons reçu un accueil chaleureux.

Et aussi, bravo et merci à tous ceux qui se sont convertis, au moins partiellement et au terme de ces trois années de croisade, à la sainte trinité GNU/Linux, LaTeX et R.

Avant-propos

Ce document est présenté sous la forme d'une thèse par article. Celle-ci a été conçue selon les critères de présentation adoptés par le comité de programme de 2^{e} et 3^{e} cycle en sciences forestières de l'Université Laval. Les articles suivants, rédigés en langue anglaise, sont inclus dans cet ouvrage. Pour chaque article, j'ai établi les objectifs de recherche, formulé les hypothèses, réalisé les analyses et les interprétations des résultats ainsi que la rédaction.

Chacun des articles a été inclus dans le manuscrit sous la forme exacte avec laquelle il a été écrit, relu et soumis sans aucune modification.

Article 1 — Roussel, J.-R., Caspersen, J., Béland, M., Thomas, S., & Achim, A. (2017). Removing bias from LiDAR-based estimates of canopy height : Accounting for the effects of pulse density and footprint size. Remote Sensing of Environment, 198, 1–16. https://doi.org/10.1016/j.rse.2017.05.032

L'article a été soumis le 1 septembre 2016 et accepté le 24 mai 2017.

Article 2 — Roussel, J.-R., Caspersen, J., Béland, M., & Achim, A. (2018). A mathematical framework to describe the effect of beam incidence angle on metrics derived from airborne LiDAR : the case of forest canopies approaching turbid medium behaviour. Remote Sensing of Environment (in press)

L'article a été soumis le 16 octobre 2016 et accepté le 7 décembre 2017.

Article 3 — Roussel, J.-R., Auty D., & Achim, A. Algorithms and software for ALS in forestry and ecology – A critical review

L'article n'a pas encore été soumis et sera soumis sous peu dans Environmental Modelling & Software.

Alexis Achim, directeur de ce projet de doctorat, est co-auteur des trois articles et m'a conseillé et supervisé dans la rédaction. Martin Béland et John Caspersen, co-directeurs de ce projet de doctorat sont co-auteurs des articles 1 et 2 et m'ont conseillé dans la rédaction. David Auty, co-auteur de l'article 3 m'a aidé dans la relecture attentive du manuscrit. Sean Thomas, co-auteurs de l'article 1, a contribué par l'acquisition des données qui ont servi à l'étude.

Chapitre 1

Introduction

1.1 Les enjeux du LiDAR aérien

Les forêts sont au cœur des préoccupations actuelles à l'échelle planétaire sur les changements climatiques, la déforestation, la séquestration du carbone ou la protection de la biodiversité. Il en découle un besoin grandissant d'informations précises sur la ressource forestière, son évolution et sa durabilité (Koch, 2010; Gillis *et al.*, 2005). À ces enjeux globaux viennent s'ajouter les fonctions économiques, plus locales, de production et de récréation (Bonnet *et al.*, 2011). Le besoin pour les acteurs économiques du secteur forestier, de produire et d'exploiter de façon toujours plus rentable et toujours plus optimisée, pour rester compétitifs, exerce une pression importante sur les entreprises. Ainsi, il existe un vrai besoin de connaître, de cartographier et de caractériser la ressource forestière, et cela passe par la description et la caractérisation dendrométrique de la forêt qui sont des préalables à une bonne gestion (Bonnet *et al.*, 2011).

La description des forêts se concrétise par la réalisation d'inventaires (p. ex. Gaudin, 1997; Gaudin *et al.*, 2005; Gillis *et al.*, 2005) sur des étendues pouvant être vastes. Cela mobilise des ressources humaines et financières importantes (Bonnet *et al.*, 2011). Mais l'inventaire terrain ne suffit plus, car il est trop lent, trop cher et pas assez exhaustif. Le secteur forestier se tourne donc vers des méthodes d'inventaire automatiques ou semi-automatiques à grandes échelles. La télédétection (p. ex. stéréo imagerie, imagerie optique multispectrale) a déjà démontré, au travers de nombreuses études, son potentiel de caractérisation de la ressource forestière (Bonnet *et al.*, 2013). En particulier, le développement du LiDAR (*light detection and ranging*) aérien a ouvert la voie à de nouvelles perspectives (p. ex. Kane *et al.*, 2008; Ioki *et al.*, 2009; Bouvier *et al.*, 2015) et la littérature récente suggère que le LiDAR aérien a le potentiel pour devenir la principale technologie d'inventaire et de cartographie de la forêt.

1.2 Le LiDAR : principe et fonctionnement

La télémétrie (détermination de la distance d'un objet lointain) par laser est une technique de mesure de distance basée sur le délai nécessaire à la lumière pour être renvoyée vers son émetteur.

Le LiDAR est un instrument incontournable de télémétrie active et trouve des applications en topographie (géomorphologie (p. ex. Höfle et Rutzinger, 2011), altimétrie (p. ex. Evette *et al.*, 2014) et bathymétrie (p. ex. Irish et White, 1998), géosciences (risque sismique, météorologie (p. ex. Northend *et al.*, 1966), physique de l'atmosphère (p. ex. Baumgarten, 2010) et sciences de l'environnement (étude de la pollution atmosphérique, agronomie & sylviculture), mais aussi dans l'archéologie (p. ex. Chase *et al.*, 2011), le guidage automatique de véhicules terrestres (p. ex. Schnürmacher *et al.*, 2013; Liu et Deng, 2015) ou spatiaux, ou encore la sécurité routière ou la défense. Une impulsion laser est émise, le temps aller retour de cette impulsion est mesuré et, connaissant la vitesse de la lumière, la distance de l'objet qui a rétro-diffusé l'impulsion peut être calculée.

Le LiDAR aéroporté (ou ALS pour *airborne laser scanning*), est un cas spécifique d'utilisation qui consiste à embarquer le système d'acquisition laser (émetteur et récepteur) dans un avion qui survole un territoire à analyser. De nombreuses impulsions sont émises à haute fréquence et ces impulsions sont rétro-diffusées par le sol où le couvert forestier. Connaissant la distance entre le dispositif d'acquisition et les objets au sol, ainsi que la position exacte du dispositif d'acquisition grâce à un système de positionnent embarqué, il devient possible d'estimer la position exacte des objets qui ont retro-diffusé les impulsions laser et de les cartographier en trois dimensions sous la forme d'un nuage de points.

Outil initialement dédié à la topographie, l'utilisation du LiDAR aéroporté (simplement appelé LiDAR par la suite) dans l'inventaire forestier remonte aux années 1970 (Nelson, 2013). Il permet d'acquérir rapidement un très grand nombre de points dans l'espace qui décrivent la structure horizontale et verticale du couvert forestier (Lim *et al.*, 2003), et ce, de manière extrêmement rapide. Il constitue alors un outil intéressant pour l'amélioration des inventaires forestiers et l'aide à la décision (Gleason et Im, 2012) en produisant des mesures impossibles à réaliser par des techniciens sur place, tant par leur quantité que par leur technicité. Il est ainsi utilisé en foresterie, en écologie (p. ex. Zellweger *et al.*, 2013) ou en aménagement des territoires naturels (p. ex. Bilodeau, 2010) et urbains (p. ex. Mallet *et al.*, 2008). Cependant, le LiDAR ne donne pas de mesures directement exploitables. Il s'agit de nuages de points bruts qui n'ont pas de sens sans post-traitement.

Il est alors indispensable d'apprendre à interpréter ces nuages pour en extraire des informations en développant des algorithmes opérationnels ainsi que des modèles mathématiques permettant de transformer ces nuages de points en données sémantiques et structurées.

1.3 Méthodologies et modèles prédictifs

Pour donner du sens aux données LiDAR, des équipes de recherche issues du monde entier (p. ex. aux USA (Zhao *et al.*, 2009), en Suisse (Kleiner *et al.*, 2010), Finlande (Korpela *et al.*, 2010), Italie (Pirotti *et al.*, 2008), Espagne (Pascual *et al.*, 2008), Canada (Boudreau *et al.*, 2008), Australie (Zhang et Liu, 2013), Angleterre (Donoghue *et al.*, 2007), etc.) œuvrent au développement de modèles mathématiques prédictifs permettant de qualifier, quantifier et analyser la structure des forêts et leurs propriétés biophysiques à partir des nuages de points LiDAR. On distingue deux principaux types d'approches pour construire des modèles prédictifs : (a) l'approche zonale et (b) l'approche individuelle.

1.3.1 Approche zonale

L'approche zonale est simple à mettre en œuvre, et c'est l'approche la plus répandue. On lie une grandeur biophysique d'intérêt Q à des grandeurs X_i extraites du nuage de points. Le lien se fait par modélisation statistique (Holmgren *et al.*, 2003a; Holmgren, 2004; Ioki *et al.*, 2009; Chehata *et al.*, 2009; Zhao *et al.*, 2009; Chen et Hay, 2011; Lim *et al.*, 2014).

Les grandeurs d'intérêts Q sont généralement la hauteur de la canopée, la biomasse, la surface terrière, l'indice de surface foliaire, ou encore le volume marchand des tiges.

Les grandeurs X_i extraites du nuage de points, appelées métriques dérivées, sont des scalaires qui résument en un seul nombre une propriété du nuage du point. Elles peuvent être de plusieurs natures, mais elles sont généralement calculées à partir de la distribution verticale des retours LiDAR (hauteur moyenne de points, hauteur maximale, écart-type de la distribution verticale des points, quantile de la distribution, etc.) et sont souvent de nature statistique. Plus rarement, ces métriques peuvent être calculées à partir de la distribution des intensités des points et sont, dans ce cas aussi, de nature statistique. Dans ces deux cas les métriques sont dérivées de données unidimensionnelles(axe z ou axe i[1]), et ne tirent donc partie que d'une seule dimension sur les nombreuses disponibles. De rares cas de métriques tirant partie de plus de dimensions, et donc d'une plus grande proportion du jeu de données, peuvent être trouvés dans la littérature. On trouve par exemple des métriques comme la rugosité de la canopée qui tire partie des 3 coordonnées spatiales mesurées pour évaluer un indice de complexité structurelle (Kane *et al.*, 2008, 2010),

La modélisation statistique permet ainsi de lier les métriques X_i (variables explicatives) à Q (variable dépendante) par des équations dont les paramètres sont ajustés automatiquement à des données d'inventaire terrain utilisés comme référence.

1.3.2 Approche individuelle

L'approche par délimitation individuelle est moins courante, car elle est techniquement plus difficile et nécessite une plus grande densité de points. La première étape importante de cette approche est la reconnaissance et la segmentation individuelle de chaque arbre à partir du nuage de points. Des approches de segmentations variées ont été proposées (Pyysalo et Hyyppä, 2002; Morsdorf *et al.*, 2004; Reitberger *et al.*, 2008; Pirotti *et al.*, 2008; Reitberger *et al.*, 2009; Kwak *et al.*, 2010; Van Leeuwen *et al.*, 2010; Yao *et al.*, 2012; Vega *et al.*, 2014). Dans un second temps, la méthode consiste, pour chaque arbre, à extraire des métriques dérivées X_i comme la hauteur de l'arbre ou le diamètre de la couronne (Hyyppä *et al.*, 2001; Maltamo *et al.*, 2004; Popescu, 2007; Zhao *et al.*, 2009; Kwak *et al.*, 2010; Yao *et al.*, 2012; Gleason et Im, 2012). Enfin, par modélisation statistique, on

[1] i pour intensité

lie une grandeur biophysique d'intérêt Q aux métriques X_i grâce à des équations allométriques (p. ex. Yang *et al.*, 1978; Laasasenaho, 1982).

1.3.3 Approche par classification

On peut distinguer une troisième catégorie que l'on pourrait nommer **approche par classification**. Elle est de loin l'approche la moins répandue et elle se trouve à cheval entre approche zonale et individuelle en fonction de la façon dont elle est utilisée. Il s'agit d'évaluer à quelle classe appartient un « objet ». Cette approche est utilisée pour la reconnaissance d'essences dans une approche par segmentation individuelle, par exemple, car c'est typiquement un problème de classement (Holmgren et Persson, 2004; Reitberger *et al.*, 2006; Liang *et al.*, 2007; Donoghue *et al.*, 2007; Ørka *et al.*, 2009; Korpela *et al.*, 2009; Weber et Boss, 2009; García *et al.*, 2010; Korpela *et al.*, 2010; Heinzel et Koch, 2011; Vaughn *et al.*, 2011; Yao *et al.*, 2012; Gleason et Im, 2012). On trouve cependant quelques auteurs essayant cette approche dans d'autres contextes comme la détection automatique des feux de forêt (Fernandes *et al.*, 2004), le classement de placettes de forêt en classes de hauteurs (Pascual *et al.*, 2008; García *et al.*, 2011), le classement par tranches d'âge (Weber et Boss, 2009) ou la reconnaissance d'objets d'après une typologie adaptée à un paysage urbain ou semi urbain (Koetz *et al.*, 2008; Chehata *et al.*, 2009). Cette approche est fondamentalement différente des deux autres et repose généralement sur les machines d'apprentissage (*machine learning*) comme les réseaux de neurones, les séparateurs à vastes marges ou les arbres de décisions (Fernandes *et al.*, 2004; Reitberger *et al.*, 2006; Koetz *et al.*, 2008; Chehata *et al.*, 2009; Korpela *et al.*, 2009, 2010; García *et al.*, 2011; Heinzel et Koch, 2011; Zhao *et al.*, 2011; Yao *et al.*, 2012) qui sont des outils mathématico-algorithmiques adaptés aux problèmes de classement.

Ces travaux de recherche montrent qu'il est possible de prédire des informations sur la forêt uniquement en la survolant, moyennant quelques inventaires manuels de calibration. La technologie LiDAR se développe donc afin d'améliorer l'inventaire forestier et de maximiser la rentabilité de l'exploitation. Cependant, l'analyse des travaux académiques montre en réalité que la technologie n'est pas si avancée qu'elle n'y paraît. Les modèles prédictifs développés ne sont pas toujours bons, et lorsqu'ils le sont, ils ne sont généralement applicables qu'à des contextes bien précis. Les modèles statistiques sont en effet extrêmement spécifiques à la zone d'étude dans laquelle ils ont été construits. Ils peuvent même demander une connaissance préalable de la forêt pour être appliqués, ce qui est contraire aux ambitions de cartographie automatique. Ceci est dû à trois niveaux de limitation, à savoir les limitations spatiales, techniques et méthodologiques des modèles.

1.4 Limites des études et des modèles statistiques

Les études présentées dans la littérature, à quelques exceptions près (p. ex. Thomas *et al.* (2006); Hopkinson et Chasmer (2009)), sont menées sur de petits territoires d'expérimentation. Dès lors, les modèles prédictifs créés empiriquement sont localement justes, mais rien ne prouve qu'ils le soient pour d'autres forêts géographiquement et structurellement distantes. Il en va tout particulièrement ainsi de l'approche zonale qui est très peu généralisable (Van Leeuwen et Nieuwenhuis, 2010).

Par ailleurs, un modèle est généralement mis au point à partir d'un unique jeu de données, et donc à partir d'une unique configuration du dispositif d'acquisition. D'une étude à l'autre, d'un jeu de données à l'autre, beaucoup de paramètres peuvent varier tels que la densité de points acquise, l'intensité des impulsions émises, l'altitude du capteur, la vitesse de vol de l'avion, la divergence du rayon laser, la taille de l'empreinte au sol, l'angle maximum d'incidence des rayons, la longueur d'onde du laser, la sensibilité du capteur etc. Ces changements peuvent engendrer des variations dans la structure du nuage de points indépendamment de la structure de la forêt échantillonnée.

Ainsi, on ne sait en fait que peu de choses de la possibilité de généralisation des modèles. D'une part, cette réalité est attribuable à la variabilité naturelle des forêts, mais d'autre part elle est aussi reliée à la variabilité des paramètres d'acquisition. Le problème est le suivant : un modèle empirique M construit à partir d'un jeu de données LiDAR D acquis avec des paramètres P peut-il s'appliquer avec la même précision sur un jeu de données D' acquis dans la même forêt et à la même date, mais avec des paramètres P' ? Si la réponse est non, alors la généralisation des modèles prédictifs ne peut aboutir et il devient nécessaire de recalibrer un nouveau modèle prédictif.

Cette section est dédiée à une démonstration, à travers une revue des limitations qui restreignent les possibilités de généralisation des modèles indépendamment de la variabilité des structures forestières, que la réponse à cette question est effectivement « non ».

1.4.1 Limitations spatiales et taille des inventaires

Tailles des placettes

Pour construire un modèle prédictif empirique, il est nécessaire d'acquérir manuellement des données à partir de placettes d'inventaire qui servent de référence et de calibration. Les modèles ainsi construits sont ensuite appliqués à l'ensemble des placettes non échantillonnées afin d'estimer les grandeurs d'intérêt.

Selon les études, les placettes sont échantillonnées de différentes manières. Par exemple, les tailles varient beaucoup. La littérature présente des placettes dont la superficie varie de 200 m^2 (Popescu *et al.*, 2002; Næsset, 2004a; Donoghue *et al.*, 2007) à 2500 m^2 (Spriggs *et al.*, 2015). Or, la structure de la forêt est dépendante de l'échelle à laquelle on la regarde. Individuellement, chaque arbre est différent, à moyennes échelles on trouve des variations locales de structure, et à grandes échelles la forêt tend vers l'homogénéité.

Gobakken et Næsset (2008) ont étudié l'influence de la taille des placettes sur les prédictions réalisées. Cependant, cette étude caractérise aussi l'influence du nombre de placettes, du type de végétation et de la densité de points. La grande quantité de résultats sans tendance générale est difficile à interpréter. Par ailleurs, les placettes de test ne faisaient que 200 et 300-400 m^2, ce qui est, dans tous les cas, très petit et peu convaincant.

La question de l'effet de l'effort d'inventaire est pourtant primordiale pour une application pratique. Un modèle calibré pour des placettes de 1000 m^2 peut-il être utilisé pour faire une cartographie à l'échelle de 400 m^2 ? Compte tenu de la structure de la forêt, est-il vraiment judicieux de l'observer à une échelle de 400 m^2 ? En d'autres termes, toutes les tailles d'observation se valent-elles et sont-elles comparables ? Sans réponse à ces questions, la problématique de la généralisation se heurte à une inconnue.

Effort d'inventaire

Certaines études calibrent des modèles sur sept placettes (p. ex. García *et al.*, 2010) alors que d'autres en utilisent 150 (p. ex. Spriggs *et al.*, 2015). On peut légitimement se questionner sur l'influence des efforts d'inventaire sur les prédictions et les modèles. Combien faut-il de placettes pour calibrer un modèle convenablement ? Cela dépend de la taille du territoire à couvrir. Même sans preuve, il est raisonnable d'affirmer qu'un modèle basé sur sept placettes ne vaut que pour ces sept placettes et n'est pas généralisable faute de données suffisantes. Pour les autres, la question reste ouverte. À partir de quelle surface de test estime-t-on que le modèle a une chance d'être généralisable à la forêt entière ? à la région ?

Les placettes d'études, quels que soit leur nombre et leur taille, sont échantillonnées dans un espace restreint de l'ordre de quelques kilomètres ou dizaines de kilomètres carrés. Par exemple : Thomas *et al.* (2006) utilisent un inventaire de 1,5 ha répartis sur 314 ha ; Næsset (2004b), 4 ha répartis sur 1000 ha ; García *et al.* (2010), 0,45 ha repartis sur une surface inconnue ; Gobakken et Næsset (2008), 2 ha répartis sur 90 ha ; Lim *et al.* (2008), 2,4 ha répartis sur 72 ha ; Holmgren et Persson (2004), 1,2 ha repartis sur 250 ha ; Holmgren *et al.* (2003a), 2 ha répartis sur 400 ha ; Ioki *et al.* (2009), 0,6 ha repartis sur 64 ha ; Kwak *et al.* (2010), 0,25 ha repartis sur 80 ha ; Spriggs *et al.* (2015), 37 ha répartis sur 32 000 ha. L'échantillonnage, même s'il est assez grand, n'est représentatif que de cette région géographique limitée et uniquement de celle-ci. Un modèle prédictif construit sur un tel jeu de données n'est généralisable qu'avec l'hypothèse que toutes les forêts en dehors de cette région sont identiques ou que le jeu de données est représentatif d'un grand nombre de cas de figures, ce qui est peu vraisemblable compte tenu des dimensions citées. L'approche zonale et l'approche par classification sont tout particulièrement sensibles à ce problème, car basées sur la spécificité locale de la forêt et des données. L'approche individuelle est probablement plus robuste face à ce problème (Van Leeuwen et Nieuwenhuis, 2010).

Conclusion

Les spécificités des modèles à la zone d'étude et la méthodologie d'échantillonnage ne sont que rarement discutées dans la littérature. Holmgren et Persson (2004) insistent sur le fait qu'ils ne savent pas si leur méthode est exportable à d'autres zones d'étude sans en dire plus. Il n'existe, à notre connaissance, aucun travail ayant essayé d'exporter un modèle vers un autre site d'étude. Pourtant, la répétabilité des expériences est l'un des fondements de la méthode scientifique. L'utilisation du LiDAR comme outil de caractérisation de la forêt semble s'affranchir de la nécessité de confirmer les résultats obtenus du fait qu'il s'agisse plus d'ingénierie que de sciences. En effet, l'enjeux est souvent de produire un modèle prédictif faisant de bonnes prédictions, peu importe comment et pourquoi. Or, en l'absence de confirmations, ce sera ultimement l'applicabilité pratique du LiDAR comme outil de caractérisation de la forêt qui souffrira.

À l'inverse, en absence d'études approfondies, on ne peut pas non plus prétendre avoir démontré que les modèles présentés dans la littérature sont réellement inutilisables en dehors du contexte spatial de leur développement. Cette suggestion découle uniquement d'une analyse critique. Toutefois, il est peu probable que les modèles soient réellement exportables, car en plus de la limitation spatiale liée à l'inventaire, les modèles sont soumis aux limites techniques du LiDAR.

1.4.2 Limites techniques et configuration du dispositif d'acquisition

En règle générale les études sont menées avec un seul type d'émetteur/récepteur et avec une seule configuration de ces derniers (c.-à-d. longueur d'onde fixée, angle de balayage fixé, hauteur de survol fixée, fréquence d'acquisition fixée, etc.). En plus d'être spécifiques à une forêt, les résultats et les modèles sont spécifiques à une configuration particulière du dispositif d'acquisition puisque les valeurs retournées en dépendent.

Intensité

Le LiDAR enregistre l'intensité des retours. Plusieurs études ont montré le potentiel de l'utilisation des intensités, car elles sont affectées par la structure de la forêt (Moffiet *et al.*, 2005) qui peut avoir des réflectances variables. García *et al.* (2010); Watt et Wilson (2005); Hall *et al.* (2005) montrent que l'intensité des retours est toujours une variable améliorant les prédictions.

Pourtant l'intensité n'est pas une grandeur stable et elle n'est pas uniquement fonction de la réflexivité de la cible. Beaucoup d'autres éléments influencent les valeurs mesurées (Moffiet *et al.*, 2005; Höfle et Pfeifer, 2007; Poullain, 2013), dont notamment la distance parcourue par les impulsions laser puisque l'intensité du signal tend à décroître par absorbance dans l'atmosphère.

L'équation 1.1 Baltsavias (1999) montre comment la puissance reçue P_r dépend de la puissance transmise P_T, de la distance entre l'objet et le capteur R, de la transmission de l'atmosphère M, du diamètre du récepteur D_r et de la cible D_{tar}, de la réflexivité de la cible ρ et de la divergence du rayon laser γ.

$$P_r = \rho \frac{M^2 D_r^2 D_{tar}^2}{4R^2(R\gamma + D)^2} P_T \tag{1.1}$$

Dès lors, deux survols réalisés le même jour (condition météo identiques), au-dessus de la même forêt, mais à des altitudes différentes donneront deux jeux de données différents incompatibles en ce qui concerne cette grandeur. Aussi, et pour les mêmes raisons de distances parcourues, la topographie peut faire varier les valeurs d'intensité mesurées.

Pour un survol à altitude constante, les régions de plus basse altitude sont plus loin du capteur et les mesures d'intensité plus faibles indépendamment de la structure forestière.

Par ailleurs, les détails techniques nécessaires à la compréhension des valeurs d'intensité retournées sont inaccessibles, dû au fait que les données sont généralement prétraitées par des logiciels privateurs. C'est pourquoi l'intensité est une valeur difficile à manipuler.

Ainsi, un modèle utilisant les valeurs brutes d'intensité n'a de valeur que pour une étude donnée avec un paramétrage bien particulier. Et encore, les sources de variations locales viennent ajouter un bruit important. Il existe des méthodes de correction permettant, a minima, de corriger la valeur d'intensité des variations de distance. C'est ce qu'on appelle la correction de *range* (Höfle et Pfeifer, 2007; Poullain, 2013; Kukko *et al.*, 2008). Toutefois, pour être parfaitement normalisables, des tests de calibration devraient être faits systématiquement. Ces tests de calibration ne sont cependant pas toujours réalisables, car il faut y penser à l'avance (test sur des surfaces planes et homogènes, par exemple). Dès lors, un certain nombre de modèles utilisant les intensités LiDAR ne sont pas généralisables.

Densité de points

La densité de points est à mettre en relation directe avec les coûts financiers (Baltsavias, 1999; Lovell *et al.*, 2005; Gobakken et Næsset, 2008; Jakubowski *et al.*, 2013; Singh *et al.*, 2015). Acquérir plus de points par unité de surface signifie voler plus bas et/ou plus lentement. Ainsi, des études cherchent à évaluer l'impact de la diminution du nombre de points dans le but de diminuer les coûts sans diminuer la qualité des prédictions.

On peut dire qu'il y a un consensus dans la communauté sur le fait que la densité de points n'est pas une grandeur critique (Anderson *et al.*, 2006; Thomas *et al.*, 2006; Gobakken et Næsset, 2008; Lim *et al.*, 2008; Pirotti et Tarolli, 2010; Lovell *et al.*, 2005; Jakubowski *et al.*, 2013) pour les modèles prédictifs construits dans une approche zonale. Nous sommes toutefois en désaccord avec ce consensus qui s'appuie sur des analyses peu convaincantes. En effet, la majorité des études sur la question considèrent des jeux de données originalement échantillonnées à haute densité de points qui sont ensuite réduits artificiellement. Les études ajustent successivement des modèles statistiques sur les données de plus en plus décimées afin de tester si les prédictions perdent en précision.

Il est évident, par définition de ce qu'est une statistique dérivée du nuage de points, que ces variables ne sont pas sensibles à la densité de points. Ou, pour être plus juste, qu'elles ne sont pas sensibles à une réduction artificielle de la densité de points. Par exemple, si la hauteur moyenne de 1000 points est de 10 m alors la hauteur moyenne de 500 points sélectionnés aléatoirement parmi les 1000 originaux est aussi de 10 m, car la distribution reste inchangée. Réduire artificiellement le nuage de points ne peut pas affecter des métriques aussi simples.

Or, les auteurs cités se limitent à tester les effets de la densité de points sur des modèles classiques utilisant de telles statistiques simples. Thomas *et al.* (2006) suggèrent que la densité de points est critique pour la mesure de la taille des couronnes, la détection individuelle des arbres ou la mesure de la fermeture du couvert, car ces mesures reposent sur la caractérisation horizontale des données. En réalité, pour des métriques plus complexes tirant profit de plusieurs dimensions comme la rugosité de la canopée, la question reste ouverte et peu ou pas étudiée. Il n'apparaît pas pertinent de penser que la densité de points n'a pas d'effet. Effectivement, les modèles basés sur la segmentation individuelle *sont* sensibles et dépendants de la densité de points, car ils utilisent la structuration horizontale des points. D'un inventaire à l'autre, si la configuration varie, les modèles ne sont plus valables et nécessitent une recalibration et un nouvel inventaire local.

Angles d'incidence

Le LiDAR réalise un balayage oscillant classiquement entre -20 et +20°. Les rayons atteignent donc le couvert forestier avec des angles différents. L'effet de l'angle d'incidence est peu documenté et les conclusions sont contradictoires. Holmgren *et al.* (2003b) montrent que la répartition des retours changent avec l'angle d'incidence, tandis que García *et al.* (2010) rapportent que l'effet de l'angle d'incidence peut être négligé d'après Coren et Sterzai (2006); Kukko *et al.* (2008). Les études ne sont cependant pas comparables : Holmgren *et al.* (2003b) travaillent sur des forêts numériques générées par ordinateur tandis que Kukko *et al.* (2008) travaillent en laboratoire sur banc de test et Coren et Sterzai (2006) expérimentent sur des routes. Holmgren *et al.* (2003a) ne trouvent pas d'effet de l'angle sur la mesure de hauteur mais trouvent un effet significatif sur la mesure du taux de couverture.

Ainsi, la question sur l'existence même des effets n'est pas claire. Pourtant, il est possible de se convaincre de l'existence de tels effets en imaginant un nuage de points échantillonné avec un angle d'incidence de 89°. Il est alors absolument évident que la répartition spatiale des points serait très différente comparée au même échantillonnage effectué à 0°. Le phénomène étant physique, géométrique et macroscopique, il est nécessairement continu. De ce fait, il y a aussi un effet à un angle de 1°. La question qui se pose alors est de quantifier ces effets qui sont peut-être très faibles et donc invisibles dans le cadre des tests statistiques, mais dont l'existence est réelle et loin d'être négligeable passé un certain angle.

Si les effets existent mais sont difficilement identifiables par une approche empirique, il faut alors construire un modèle physique théorique qui les explique. Par exemple Goodwin *et al.* (2007); Disney *et al.* (2010) ont fait l'hypothèse que la distance parcourue dans le couvert forestier par un rayon oblique était plus grande qu'au nadir. Ainsi, la probabilité de toucher une cible (une feuille ou une branche) est plus grande, ce qui, mécaniquement, implique que les rayons fortement incidents pénètrent moins le couvert forestier. Cependant, les auteurs ne sont pas allés plus loin que la formulation de l'hypothèse.

Altitude du survol

L'augmentation de l'altitude se traduit par une augmentation de la taille de l'empreinte au sol du laser en raison de la divergence du rayon. Une des conséquences est une perte de capacité de pénétration (Thomas *et al.*, 2006) suivie mécaniquement par une diminution du nombre de retours (Goodwin *et al.*, 2006). Les performances des modèles se trouvent ainsi affaiblies (Popescu *et al.*, 2000; Yu *et al.*, 2004). La structuration verticale des points étant modifiée, les modèles basés sur cette structuration s'en trouveront biaisés dans le cadre d'une utilisation avec un autre jeu de données.

L'augmentation de l'altitude, à fréquence d'émission constante, induit une diminution de la densité de points au sol. Ainsi, une variation de densité de points ne vient généralement pas seule, mais est accompagnée d'autres variations comme l'intensité émise ou la taille de l'empreinte du laser. C'est pourquoi la réduction artificielle de la densité de points comme preuve du fait qu'elle n'a pas d'effet sur les modèles prédictifs ne suffit pas à décrire la réalité. Cet argument s'ajoute à ceux proposés à la section 1.4.2.

On notera tout de même que Næsset (2004b) montre, quant à lui, la non-influence de la hauteur de survol. Cependant, la comparaison se limitait à deux survols à basse altitude (540 et 850 m). Par ailleurs, on a montré comment l'altitude avait une influence importante sur les valeurs d'intensités retournées. L'altitude de survol est donc critique aussi pour la validité et la possibilité d'exporter à d'autres contextes une forte proportion des modèles présentés dans la littérature.

Fréquence d'impulsion

Les propriétés des impulsions émises dépendent de la fréquence d'émission. Quand la fréquence d'émission augmente on échantillonne avec une plus grande densité de points, mais l'énergie ou la durée des impulsions diminue (Baltsavias, 1999; Næsset, 2005). Cela crée un signal plus bruité et une perte de capacité de pénétration dans la canopée (Chasmer *et al.*, 2006a). La structuration verticale des retours s'en trouve modifiée, et les modèles basés sur cette structuration s'en trouvent biaisés si les propriétés des impulsions émises varient entre deux inventaires. Ainsi, de nombreux paramètres sont liés entre eux et sont extrêmement difficiles à étudier empiriquement de façon individuelle sans banc d'essais en laboratoire.

Conclusions

La section précédente présentait des limitations spatiales suggérant que les modèles présentés dans la littérature n'ont généralement de valeur que localement, ou tout du moins, qu'il est fort peu probable qu'ils puissent être valables en dehors de la limite spatiale dans laquelle ils ont été construits. Nous ajoutons en plus une limitation technique qui réduit encore plus les possibilités de généralisation des modèles. Selon les choix des statistiques dérivées utilisées et la configuration du LiDAR, les modèles seront plus ou moins facilement réutilisables avec d'autres jeux de données acquis avec des paramètres différents.

En plus de ces deux problèmes s'ajoute parfois un problème de méthodologie entraînant des incohérences au sein même d'une étude.

1.4.3 Limites méthodologiques et exploitation des données

Pour beaucoup d'auteurs, l'objectif visé par le développement de modèles statistiques semblent être d'atteindre des corrélations R^2 proches de 1 et des erreurs RMSE proches de 0 entre les données d'inventaire et les modèles prédictifs. Ceci engendre des incohérences internes. L'étude de Næsset (2004b) illustre bien ce propos. L'auteur teste un certain nombre de modèles et garde les meilleurs pour chaque classe de test. Regardons par exemple un modèle de prévision du volume de bois.

Pour de jeunes forêts survolées à 450 m d'altitude, le meilleur modèle de prédiction du volume dépend de $h_{50,f}$ la médiane des hauteurs des premiers retours et de $d_{1,f}$ et de $d_{9,f}$ des ratios de premiers retours proches du sol et proches de la canopée. Pour la même forêt survolée à 850 m d'altitude, le modèle change complètement de variables explicatives et devient dépendant de $h_{mean,f}$ la moyenne des hauteurs des premiers retours et $d_{5,l}$ un ratio de dernier retour à la médiane des hauteurs des retours. Si la forêt est mature, de faible qualité et survolée à 450 m ou 850 m, les résultats font apparaître encore d'autres variables explicatives. Les modèles prédictifs pour des forêts matures de bonne qualité sont encore différentes.

Il est dès lors inconcevable d'imaginer porter ces modèles dans une autre forêt alors même qu'il n'y a pas de cohérence interne. La recherche du modèle avec le plus grand coefficient de corrélation a pour effet de produire un ensemble de modèles tous différents et non cohérents entre eux qui sont excessivement sensibles à une variété de changements non contrôlables.

Par ailleurs, si chaque type de structure a son modèle, alors la connaissance préalable de la forêt est un pré-requis à l'analyse des données LiDAR. Or, ce n'est pas ce qui est recherché. Illustré par l'un des travaux de Næsset, cette critique est applicable à plusieurs auteurs (p. ex. Hopkinson et Chasmer (2009); García *et al.* (2010); Singh *et al.* (2015); Ahmed *et al.* (2015)). À cette critique il importe d'amener une nuance à l'effet qu'il est tout à fait justifié d'étudier comment la structure de la forêt influence les résultats. En cela, les auteurs ne font pas « d'erreur » à proprement parler. Par contre, leur approche n'est pas applicable dans l'optique d'une utilisation concrète du LiDAR, car celle-ci sous-entend que nous n'avons pas de connaissance *a priori* de la structure de la forêt.

C'est pourquoi cette approche par recherche des meilleurs indicateurs statistiques, appelée de façon péjorative « *kitchen sink approach* », n'a pas vocation à produire des résultats scientifiques dont la communauté d'utilisateurs pourrait tirer profit comme évoqué plus tôt. Au contraire, il s'agit de produire un modèle à usage uniquement valable localement dans le contexte d'une étude particulière. Dans ce contexte, il est donc pertinent de mettre au des méthodes d'analyse plus puissantes et plus généralistes.

1.5 Problématiques soulevées et ambitions

Les modèles de prédiction présentent donc une dépendance spatiale et une dépendance au paramétrage du LiDAR. Ainsi, les travaux académiques ont le plus souvent une valeur locale et ne répondent pas aux problématiques sur de vastes territoires. S'il est nécessaire d'aller sur le terrain ou d'y retourner tous les 10 ans pour réaliser des inventaires locaux à chaque fois qu'on souhaite utiliser le LiDAR comme outil prédictif, la télédétection perd une partie de son intérêt. Au Québec par exemple, le gouvernement fait réaliser des survols LiDAR afin d'estimer la valeur monétaire des parcelles forestières mises aux enchères pour les exploitant. Parfois, ces derniers sont réticents, à faire confiance à ces données jugées trop imprécises. De la vieille école, certains préfèrent aller sur place et se rendre compte par eux-mêmes, les années d'expérience étant encore leur meilleur outil. Ainsi, « *avant que cette technologie puisse être adoptée avec confiance [...] des modèles robustes pouvant être appliqués et validés pour des superficies de forêt vastes et complexes doivent être développés* » (Thomas *et al.*, 2006). Ces modèles doivent, a minima, être indépendants de la configuration du dispositif d'acquisition.

L'objectif général de cette thèse est donc de faire progresser le développement de méthodes généralistes permettant de réaliser des prédictions à grandes échelles sans re-calibration locale au cas par cas en fonction de la forêt et du dispositif d'acquisition. Cela passe par la capacité à s'affranchir des limitations spatiales et techniques en proposant une voie pour ne plus être dépendant du paramétrage du dispositif d'acquisition.

Pour penser à grande échelle, nous devons nous affranchir de la question de l'influence du dispositif d'acquisition et de sa configuration qui rendent potentiellement caduques les modèles de prédiction. Pour lever ces limitations deux approches peuvent être envisagées :

- L'utilisation de métriques stables, c'est-à-dire de métriques qui ne soient ni dépendantes de la densité de points, ni de l'intensité, ni de la taille de l'empreinte, ni de l'angle d'incidence des rayons etc.
- Se doter d'outils théoriques pour « normaliser » les métriques et les recalculer « comme si elles avaient été acquises avec un dispositif standard ».

La variabilité des forêts et des paysages nécessite un grand nombre de descripteurs. Imposer des méthodes d'analyse utilisant uniquement des métriques stables implique la perte d'un certain nombre de descripteurs (peut-être même tous) potentiellement pertinents, et n'est donc pas raisonnable. C'est donc la deuxième solution que nous avons envisagée.

Normaliser une métrique correspond à la recalculer « comme si elle avait été acquise avec un autre paramétrage ». Il est donc nécessaire de définir un paramétrage standard imposant sa référence à tous les autres. Ce standard ne peut être choisi arbitrairement à partir d'un matériel existant, au risque d'être rapidement désuet en plus d'être non objectif. Ce dispositif standard doit ainsi être théorique et nous proposons dans cette thèse le système d'acquisition suivant :

Densité de points : infinie. Cette référence vient assez naturellement. On ne peut pas privilégier une valeur plutôt qu'une autre. Comme la référence 0 n'a pas de sens, c'est donc naturellement que l'infini qui s'impose.

Angle d'incidence : 0° pour tous les points.

Patron de balayage : parfaitement homogène et régulier.

Empreinte : 0 pour tous les points. Le dispositif émet des impulsions de diamètre nul sans divergence.

Ce travail est essentiellement théorique et académique, mais il est légitime d'un point de vue pratique. Plaçons-nous dans un cas idéal où il existerait un modèle M extraordinairement précis et juste. Ce modèle aurait été mis au point avec un paramétrage LiDAR P_1. L'industrie ou le gouvernement possèdent en réalité des données LiDAR avec un paramétrage P_2. On peut se poser la question de la compatibilité de M avec P_2 et cette étude pourra y répondre dans une certaine mesure en proposant de convertir M dans un système de référence P_{ref}.

1.6 Modélisation statistique vs. modélisation théorique

Il semble important, avant d'avancer plus loin, de mettre l'accent sur un point fondamental qui a dirigé cette thèse du début à la fin : la différence entre les approches de modélisation statistique et théorique. On pourrait résumer la différence ainsi : la modélisation statistique (*data driven*) cherche à faire « parler » les données mesurées alors que la modélisation théorique (*hyphothesis driven*) cherche à faire parler les équations. Une modélisation théorique à été choisie dans cette thèse pour son pouvoir explicatif.

La modélisation statistique n'a aucun pouvoir explicatif. Les équations issues de cette méthode d'étude s'ajustent nécessairement aux données par construction mais ne permettent pas de décrire les processus ou les liens de causalité qui sont sous-jacents.

Son rôle est de décrire les données mesurées, et les modèles ainsi construits ne s'appliquent généralement pas à d'autres cas d'études et ne peuvent en aucun cas prédire des choses qui n'ont pas été observées dans les données. C'est pourquoi l'approche zonale, entièrement basée sur une modélisation statistique, est limitée.

A l'inverse, la modélisation théorique cherche à créer un modèle avant même d'observer les données. Les équations sont construites à partir d'hypothèses théoriques potentiellement sans lien avec le sujet d'étude. Si ces équations s'ajustent aux données alors le modèle théorique est potentiellement une bonne description de la réalité physique/biologique et les équations peuvent parler et mettre en évidence des faits qui n'avaient jamais même été observés et dont on ne soupçonnait pas l'existence a priori.

L'introduction de cette thèse illustre un ensemble de limites applicables à la modélisation statistique telle qu'elle est très majoritairement pratiquée. Nous chercherons à l'inverse des relations théoriques déterministes justifiées dans cette thèse.

Chapitre 2

Effet de la densité de points sur la hauteur du couvert

Il a été montré à de nombreuses reprises que la densité de points n'avait pas ou peu d'influence sur les statistiques dérivées d'une analyse par approche zonale (voir section 1.4.2). Cependant, nous avons montré que ces études étaient incomplètes.

Le problème devient intéressant lorsque les métriques ne sont pas des statistiques. C'est le cas, par exemple, de la métrique h_{max}, soit la hauteur du retour le plus haut dans une parcelle donnée. Cette variable n'est pas une statistique, c'est en fait la queue de la distribution et cette valeur peut être largement variable avec la densité de points puisque qu'elle n'a pas la stabilité d'une statistique.

La particularité de la métrique h_{max} est que, selon l'échelle d'observation, cette métrique peut retourner deux objets différents. À l'échelle d'une parcelle (plusieurs centaines de mètres carrés) il s'agit du point le plus haut retourné, et donc approximativement la taille de l'arbre le plus haut. À l'échelle de 1 ou 2 m^2 (ou moins) cette métrique donne accès au modèle numérique de canopée calculé avec l'algorithme le plus simple existant, mais aussi le plus utilisé.

On devine aisément que cette métrique est hautement sensible à la densité de points en considérant ce problème d'un point de vue probabiliste. Le LiDAR permet un échantillonnage discret de la forêt. Dès lors, il existe une probabilité non nulle de ne pas toucher l'objet le plus haut dans une région de l'espace donnée. Moins on acquiert de points, plus il y a une probabilité importante de manquer cet objet, et donc de sous-estimer la métrique h_{max}. Dans le cas théorique où la densité de points serait infinie, la probabilité de trouver cette hauteur maximum serait de 1.

Cette métrique est donc dépendante de la densité et deux acquisitions LiDAR avec des densités de points nominales différentes devraient trouver, en théorie, des hauteurs d'arbres et des modèles numériques de canopée différentes, et tomber en plein dans le problème soulevé en introduction.

L'enjeu est de quantifier cet effet de façon théorique en se basant uniquement sur la théorie des probabilités et sans tenir aucunement compte de la nature biologique du sujet d'étude afin de (1) recalculer la métrique pour la normaliser et (2) s'assurer que le modèle n'est pas empirique et ainsi s'assurer de sa possible application à des échelles potentiellement bien plus vastes que la zone d'étude.

L'article de ce chapitre propose une analyse théorique, probabiliste et multi-échelle de cette métrique. L'abstraction de la nature biologique du sujet d'étude se fait à travers un jeu de dés. Une fois identifié comme un jeu de dés, le problème se résume en effet à un simple exercice mathématique de probabilités. Le modèle théorique ainsi proposé prétend pouvoir accéder, de façon statistique, à la hauteur maximum d'une parcelle ainsi qu'à la hauteur moyenne de la canopée « comme si le jeu de données avait été acquis avec une densité de points infinie ». La confrontation du modèle de jeu de dés aux données empiriques montre une adéquation de la théorie à la pratique.

Nous montrons donc dans ce modèle que la densité de points est un paramètre important sur les métriques non statistiques et/ou qui reposent sur plus d'une coordonnée spatiale à travers une analyse théorique validée empiriquement.

2.1 Résumé

Le LiDAR aéroporté est utilisé dans l'inventaire forestier pour quantifier la structure des parcelles en utilisant un nuage de points tridimensionnel. Cependant, la structure du nuage de points ne dépend pas seulement de la structure de la parcelle de forêt échantillonnée mais aussi de l'instrument d'acquisition utilisé, de son paramétrage et de la manière dont le territoire est survolé. Les variations résultantes au sein et entre les jeux de données (particulièrement les variations de densité de points et de taille d'empreinte) peuvent induire des variations parasites dans les métriques LiDAR comme la hauteur maximum (h_{max}) et la hauteur moyenne de modèle numérique de canopée (C_{mean}). Dans cette étude, nous comparons tout d'abord deux jeux de données LiDAR acquis avec des paramètres différents et nous observons que les métriques h_{max} et C_{mean} sont 56 cm et 1.0 m plus haute, respectivement, lorsqu'elles sont calculées avec un jeu de données à haute densité et petit empreinte. Puis nous présentons un modèle qui explique ces biais observés en nous basant sur la théorie des probabilités qui nous permet de recalculer les métriques comme si la densité de points était infinie et les dimensions des deux empreintes équivalentes. Ce modèle correspond à la première étape dans la mise au point de méthodes pour corriger diverses métriques LiDAR qui sont utilisées en approche zonale pour la prédiction de la structure des parcelles forestières. De telles méthodes pourraient être particulièrement utiles pour le suivi temporel de la croissance de la forêt considérant que les paramètres d'acquisition changent régulièrement entre les inventaires

2.2 Abstract

Airborne laser scanning (LiDAR) is used in forest inventories to quantify stand structure with three dimensional point clouds. However, the structure of point clouds depends not only on stand structure, but also on the LiDAR instrument, its settings, and the pattern of flight. The resulting variation between and within datasets (particularly variation in pulse density and footprint size) can induce spurious variation in LiDAR metrics such as maximum height (h_{max}) and mean height of the canopy surface model (C_{mean}). In this study, we first compare two LiDAR datasets acquired with different parameters, and observe that h_{max} and C_{mean} are 56 cm and 1.0 m higher, respectively, when calculated using the high-density dataset with a small footprint. Then, we present a model that explains the observed bias using probability theory, and allows us to recompute the metrics as if the density of pulses were infinite and the size of the two footprints were equivalent. The model is our first step in developing methods for correcting various LiDAR metrics that are used for area-based prediction of stand structure. Such methods may be particularly useful for monitoring forest growth over time, given that acquisition parameters often change between inventories.

2.3 Introduction

Airborne laser scanning (LiDAR) is a remote sensing technology for characterizing the surface of the earth using a cloud of georeferenced points. A single point records the height at which the emitted light was reflected back to the sensor with enough energy to generate a “spike of intensity”. During the last two decades, the adoption of this technology has increased rapidly, along with the number of applications, particularly in the fields of topography and forest inventory. In the forestry sector, LiDAR has the potential to reduce the need for intensive ground-based measurement of stand structure, making it a valuable tool for “wall-to-wall” forest inventory and mapping (Thomas *et al.*, 2006).

2.3.1 Prediction methods and their limits

The most common approach for describing forest structure is referred to as the “area-based approach” (ABA), because the point cloud is aggregated and summarized into LiDAR metrics that reflect the structure of the forest at the stand level (usually square pixels of 400 m^2) (Woods *et al.*, 2011; White *et al.*, 2013). This method is dependent on plot-based inventory data, which is used for the calibration of statistical models relating LiDAR metrics to variables of interest, such as stand height, stand wood volume, and stand aboveground biomass (e.g. Holmgren, 2004; Ioki *et al.*, 2009; Lim *et al.*, 2014; Bouvier *et al.*, 2015).

The alternative “individual tree based approach” of delineating and measuring individual tree crowns is rapidly gaining in importance (e.g. Pyysalo et Hyyppä, 2002; Morsdorf *et al.*, 2004; Reitberger *et al.*, 2009; Kwak *et al.*, 2010; Yao *et al.*, 2012; Vega *et al.*, 2014). However, despite the decreasing costs of data acquisition and the constant increase of computing power, the ABA remains the most practical approach for large-scale inventories because it needs lower point density and is therefore cheaper. For example, due to the large landbase of the Canadian province of Quebec, the Ministry of Forests, Wildlife and Parks (MFWPQ) has recently made the decision to run a province-wide survey at a low to medium pulse density (~ 2 to 4 pulses/m^2). This will not be sufficient for delineating individual tree crowns in closed-crown forests, so we expect that the ABA will remain relevant for some years to come.

However, one drawback of the ABA is that the statistical models used cannot be generalized in every configuration. For example, when relating two metrics X and Y to a quantity of interest Q by the equation $Q = \alpha X^{\beta} Y^{\gamma}$, the model is not only specific to the forest type being sampled (Van Leeuwen et Nieuwenhuis, 2010; Coomes *et al.*, 2017), because α, β and γ have been estimated using a local inventory, but is also likely to be specific to the LiDAR campaign, because X and Y could be specific to the instrument, its settings, and the pattern of flight.

Beyond the bias potentially included in existing models, the fact that ABA-based descriptions of forest structure cannot be generalized is important because in practice this might limit the usage of LiDAR for wide-scale or multi-temporal inventory surveys in forestry. Datasets acquired from different flights, and often different providers, may not be perfectly compatible. In the operational context of the province-wide survey described above, statistical incompatibility of datasets acquired with different device parameters has been observed in contiguous areas leading to a spatial discontinuities of predictions at the exact boundary of the datasets using a metric derived from the canopy surface model that was expected to emulate a measure of stand height made in classical optical imagery (Ferland-Raymond B. & Lemonde M.-O. – MFWPQ, personal communication).

One way to avoid this issue when implementing the ABA on a large scale is to collect inventory data for each LiDAR survey, and to fit the statistical models separately. However, this is not ideal in the case of two contiguous datasets that share the same forest type. Also, such a solution implies a new ground inventory and a new calibration is necessary for each dataset, which is both time-consuming and costly. An ideal automated approach would involve the development of models that remain stable for any LiDAR settings and could therefore be applied to various datasets sampled at different times and by different providers.

One potential solution to this problem is to develop models using metrics that remain stable when acquisition parameters change. Such considerations are rarely presented in the literature, though Næsset (2004b) reported that the height of first returns did not vary significantly with flight altitude or footprint diameter (footprint size ranged between 16 and 26 cm), while last returns were more sensitive to variation in footprint diameter. The most common practice is to process a large number of candidate metrics and aim for the highest possible goodness-of-fit by automatically selecting the best combination of usually 3 or 4 of them (for model parsimony) to predict a variable of interest. This approach generally includes little consideration for metric stability. Moreover, the intrinsic nature of LiDAR point clouds implies that there are endless possibilities to develop new variants of each metric, a fact that limits the possibility to make general assessments of their robustness.

A second solution is to examine the effect that acquisition parameters have on the structure of the point cloud, and hence on metrics and model predictions. This option has received more attention in the literature, particularly the influence of pulse density on model predictions (e.g. Lovell *et al.*, 2005; Anderson *et al.*, 2006; Thomas *et al.*, 2006; Gobakken et Næsset, 2008; Lim *et al.*, 2008; Pirotti et Tarolli, 2010; Jakubowski *et al.*, 2013). Most of these studies reached the conclusion that pulse density has little or no effect on predictions because many statistical metrics remain stable when pulse density is artificially reduced (by definition of what a statistic is). Some studies concluded that pulse density affects the accuracy of the predictions without necessarily introducing bias (Magnusson *et al.*, 2007; Magnussen *et al.*, 2010; Ruiz *et al.*, 2014). However, metrics such as maximum height and its derivations are not stable because they are not statistics. Models that rely on unstable metrics can yield biased predictions at low pulse densities (e.g. Nilsson, 1996; Næsset, 1997; Evans *et al.*, 2001; Sadeghi *et al.*, 2015) especially for multi-temporal or multi-provider datasets.

Prior studies generally use an empirical (data-driven) approach to test if acquisition parameters have a measurable effect on particular metrics. However, hypothesis-driven efforts dedicated to correcting the bias that such effects may cause have mainly been restricted to the normalization of signal intensity (e.g. Höfle et Pfeifer, 2007; Kukko *et al.*, 2008). This approach can also be used to recompute LiDAR metrics as if they were obtained from an idealize “standard device”. Such a standardization method should yield the same metrics that would be obtained with an infinite pulse density, a null footprint size and a constant scan angle at nadir as it has been achived for signal intensity.

2.3.2 The specific case of maximum height (h_{max}) and derived metrics

In this paper we focus on the metric h_{max} expressed in two different ways. We derive a mathematical model for understanding how bias in h_{max} varies as as a function of pulse density, forest structure, and the scale at which it is computed (the window size). We also examine effect of the footprint size, and a derived metric called C_{mean}, which allows us to further examine the issue of scale dependency.

We examine two sources of variation in pulse density : variation between datasets and variation within datasets. Variation between datasets is mainly attributable to fixed differences in device and flight parameters. Finer scale variation within a single dataset is due to overlaps between flightlines (twice as many pulses per square meter on average), and variation in aircraft speed and attitude (mainly pitch adjustments), which are rarely discussed in the literature. Aircraft pitch adjustments are unavoidable because of the need to maintain the specified altitude. Direction and speed corrections are also common and may result in local variations in pulse density. The local pulse density variations that result from pitch corrections create a clear geometric pattern perpendicular to the flight direction (Figure 2.1). Gatziolis et Andersen (2008) presented a similar pattern and highlighted the fact that its effects on predictions remain unknown.

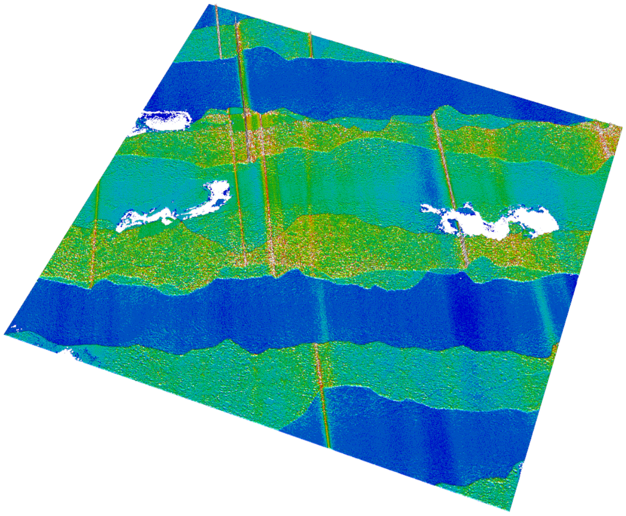


FIGURE 2.1 – Heat map of the variation in pulse density across a 4 km^2 area. Dark blue : low density ; light blue and green : intermediate density ; yellow and red : high density. Variation is due to overlap between adjacent flight lines (running from left to right) and aircraft pitch corrections, which cause the perpendicular stripes (running from top to bottom)

2.4 Methods

2.4.1 Study area

The study area is located within the Haliburton Forest and Wildlife Reserve (fig. 2.2). The forest is a 32 000 ha privately owned property located in the Great Lakes - St. Lawrence Forest Region of central Ontario, Canada (45°13’ N, 78°35’ W). Elevation ranges from approximately 400 to 500 m above sea level. The forest is a mixture hardwoods and conifers typical of northern hardwood forests, and sugar maple (*Acer saccharum* Marsh) is the dominant species, comprising 60% of the basal area. Most of the forest has been managed under selection silviculture for the past 50 years, and was selectively harvested before then. Thus, most of the stands are uneven-aged, with average canopy heights ranging from 20 to 25 m.

2.4.2 LiDAR data

Two separate LiDAR datasets were acquired in August 2009 with an Optech ALTM 3100 system. The first dataset covers the whole 320 km^2 of Haliburton forest (brown in figure 2.2), and was acquired with a standard pulse density (table 3.1). The second dataset is a small area of 68 ha (36 ha of forest, 32 of lake) within Haliburton forest (purple in figure 2.2) that was sampled with a high pulse density (table 3.1) by flying at a low altitude with a higher scan frequency.

The mature forest in this smaller area was sampled with higher density because it encompasses the Haliburton “megaplot” (13.5 ha), which is part of the CTFS-ForestGEO network of long-term forest dynamics research plots (Anderson-Teixeira *et al.*, 2015). The area overflown twice and with large overlaps, which means that on average, each part of this large plot was overflown four times. Pulse density reached 26 pulses/m^2 on average, ranging from 15 to 80 pulses/m^2 (maps of pulse density are given in the supplementary materials fig. S1).

Table 3.1 lists the flight parameters for the two datasets. The acronym “HD” refers to the high density dataset, whereas “LMD” refers to the low to medium density dataset. This information was provided by the data provider as part of the documentation provided with the datasets. The LMD dataset encompasses all of Haliburton, but a subset of this data will be compared to the HD dataset from the megaplot, so in this context we will also refer to this subset as the LMD dataset.

The normalization of the datasets (i.e. the subtraction of the digital terrain model) was done by the provider and we had no access to the raw data. The method was based on triangular irregular network construction from returns classified as “ground”, although we could not obtain further details about the algorithm used to determine point classes.

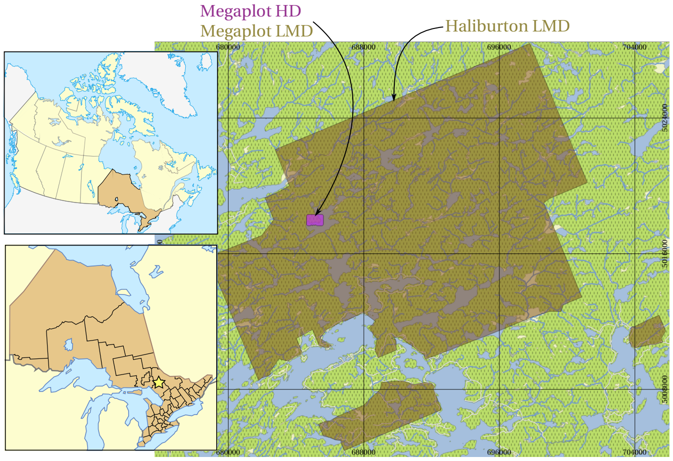


FIGURE 2.2 – Map of study areas. Brown area : low to medium density (LMD) dataset encompassing Haliburton Forest. Purple area : high density (HD) dataset encompassing the megaplot (this area was sampled twice, once at high density and once at low to medium density).

TABLE 2.1 – Flight parameters for the two datasets. HD refers to high density and LMD refers to low to medium density. PRF = pulse repetition frequency.

	LMD	HD
Altitude	1500 m	500 m
Overlap	30 %	50 %
Speed	120 kts	120 kts
Scan Frequency	36 Hz	70 Hz
System PRF	70 kHz	70 kHz
Scan half angle	16 °	10 °
Cross track resolution	0.89 m	0.40 m
Down track resolution	0.86 m	0.35 m
Point density	≈ 2 m^{-2}	≈ 28 m^{-2}
Pulse density	≈ 1.6 m^{-2}	≈ 26 m^{-2}
Footprint size	0.14 m^2	0.015 m^2
Area	30 000 ha	68 ha

2.4.3 Data processing

Data pre-processing

Lakes and wetlands were removed from the datasets to retain only forested areas. To do so, we used geographic data from the latest provincial cartography of Ontario, which matched the location of lakes and wetlands from our LiDAR datasets very closely.

Rasterization

As shown in Figure 2.3, we analysed that data at three nested scales : plot pixel (400 m^2), which is commonly used to compute LiDAR metrics for area-based approaches ; canopy pixel (4 m^2), which were used to compute the canopy surface model as a raster of canopy pixels ; spot pixel (0.14 m^2), which approximate the size of an LMD footprint, allowing us to test the effect of the footprint size on LiDAR metrics.

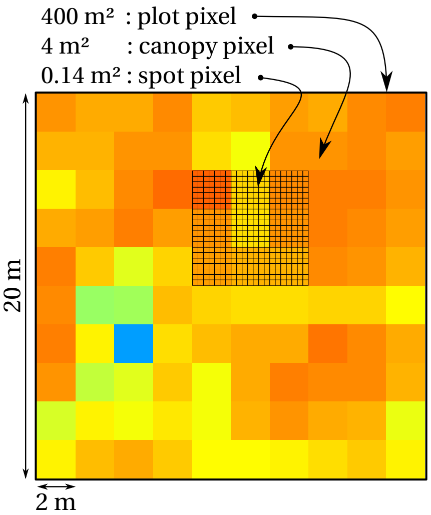


FIGURE 2.3 – Nesting of plot, canopy and footprint pixel. The plot pixel were used to compute LiDAR metrics, canopy pixel were used to compute the canopy surface model, the spot pixel were used to test the effect of the footprint size on LiDAR metrics.

Plot pixels : computing the metrics

For both the LMD and HD datasets, we computed the two metrics (C_{mean} and h_{max}) for each of the plot pixels, as well as a control variable (ρ) :

ρ Pulse density : the number of individual pulses in a plot pixel divided by its area.

C_{mean} Mean height of the canopy surface model : averaged across all the canopy pixels within a plot pixel. A 400 m^2 plot pixel is composed of 100 canopy pixels of 4 m^2, so the mean height is computed from 100 data. The construction of this metric is analogous to that of Ferland & Lemonde referred to in the introduction.

h_{max} Maximum height or the 100th percentile of height, calculated using all returns within a plot pixel.

Canopy pixel : computing the canopy surface model

A canopy surface model was computed for both the HD and LMD datasets using the canopy pixels. We used the "local maximum" algorithm to identify the highest point in each 4 m^2 canopy pixel. This is the simplest algorithm that can be used to compute a canopy surface model, and has the advantage of being amenable to analysis. This algorithm is nothing more than the computation of h_{max} for a smaller window size. Thus, calculating C_{mean} enables us to address the question of scale dependency and the question of metrics inderectly linked to h_{max}. Moreover, the method is identical to that used by the provider to extract the canopy surface model. It therefore corresponds to a product that is used in practice.

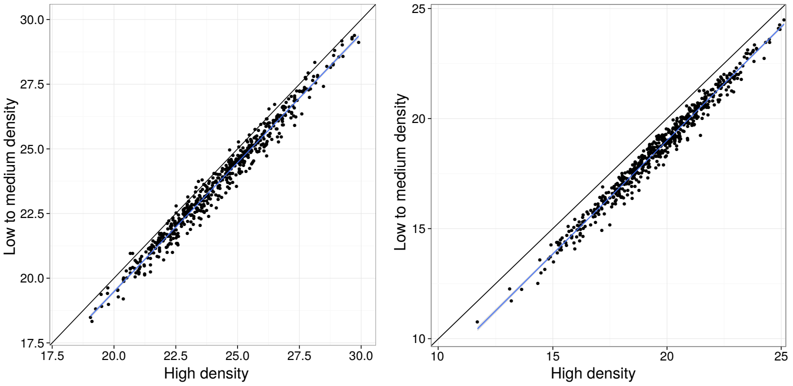
A 2 × 2 m resolution was selected based on the pulse density of the LMD dataset : it was the highest possible resolution beyond which holes would start to occur in the canopy surface model. It was also the resolution used by the provider, but it remains only a choice made among other possiblities.

Footprint pixel : assessing the effect of footprint size

The footprint pixel enabled us to test whether beam divergence causes additional bias when estimating canopy height. Before describing how we assessed the effect of footprint size using the footprint pixels (see section 2.4.7), we must further explain the conceptual framework of our analysis, beginning with a simple observation.

2.4.4 A preliminary observation : comparison of the HD and LMD datasets

To assess the magnitude of bias, we used both the HD and LMD datasets to calculate the height metrics, h_{max} and C_{mean}. The maximum and mean heights were 57 cm and 1.0 m greater, respectively, when calculated using the HD dataset (figure 2.4).



(a) Maximum height (h_{max})

(b) Mean height (C_{mean})

FIGURE 2.4 – Comparison of the height metrics calculated from the HD and LMD datasets, including 586 plot pixels (400 m^2) from the megaplot.

The goal of this study is to identify the sources of this bias, and determine how they can be understood, modelled, and predicted. Using a model based on probability theory, we describe these observations mathematically, as a function of the number of points used to sample a given area. We first describe sampling bias and our model of it from a theoretical perspective. Then, to validate the model, we develop a method for correcting for the effect of pulse density, and show that applying the correction to both the HD and LMD datasets effectively removes the bias, yielding the same height metrics for both datasets. This validation exercise demonstrates that our correction method allows us to recompute the height metrics as if the datasets had been sampled with an infinite pulse density.

2.4.5 A conceptual framework for understanding sampling bias

The complete mathematical development of the model is described in section 3.5. Here, we first present a conceptual framework for understanding various sources of sampling bias, using simple diagrams to illustrate the effects of pulse density, sampling area, and crown shape. Then, we describe our model of sampling bias, and explain how it was validated using the HD and LMD datasets.

Figure 2.5a illustrates how the bias between the observed maximum height ($\widehat{h}_{max}$) and the true maximum (h_{max}), i.e. the actual highest point of the plot, increases as pulse density decreases. When 21 pulses reach the canopy, the observed maximum height ($\widehat{h}_{max,1}$) underestimates the true maximum by the amount Δh_1 = h_{max}- $\widehat{h}_{max,1}$. In contrast, when only 11 pulses reach the canopy (i.e. after removing every second pulse), the observed maximum height ($\widehat{h}_{max,2}$) is even lower, and understimates the true maximum by the amount h_{max}- $\widehat{h}_{max,2} > \Delta h_1$.

Figure 2.5a also illustrates how the bias increases as the area sampled (x axis) decreases (and pulse density remains the same). A plot pixel (400 m^2) includes multiple large trees, so the probability of sampling near the apex of a large tree (near h_{max}) is relatively high, and the observed maximum height ($\widehat{h}_{max,1}$) only underestimates the true maximum by the amount Δh_1. In contrast, when only one large tree is sampled (e.g. between 50 and 100 on the x axis), the probability of sampling near the apex is lower. As a result, the observed maximum height in a 50 m^2 plot underestimates the true height by an even greater amount h_{max}- $\widehat{h}_{max,3} > \Delta h_1$. This bias is even more extreme when using canopy pixel (4 m^2) to compute the canopy surface model.

To account for both of the sources of bias illustrated in Figure 2.5, our model quantifies density-dependent bias at two distinct scales : canopy pixel and plot pixel. The basic form of the model is :

$$h_{max} = \widehat{h}_{max} + \epsilon(\rho) \tag{2.1}$$

where h_{max} is the true maximum value of a pixel (either a canopy pixel or plot pixel) , $\widehat{h}_{max}$ is the observed maximum value of the pixel, and $\epsilon(\rho)$ is the modelled bias computed from the local pulse density (ρ), as described in section 3.5. This basic form also applies to calculating the bias of C_{mean} since this metric is derived from a collection of h_{max} computed in a narrow windows.

Sampling density and sampling area are not the only sources of bias. Comparing figure 2.5a to figure 2.5b demonstrates that the bias in conifer stands is expected to be larger than the bias in hardwood stands, all else being equal. This is because conifers have more co-

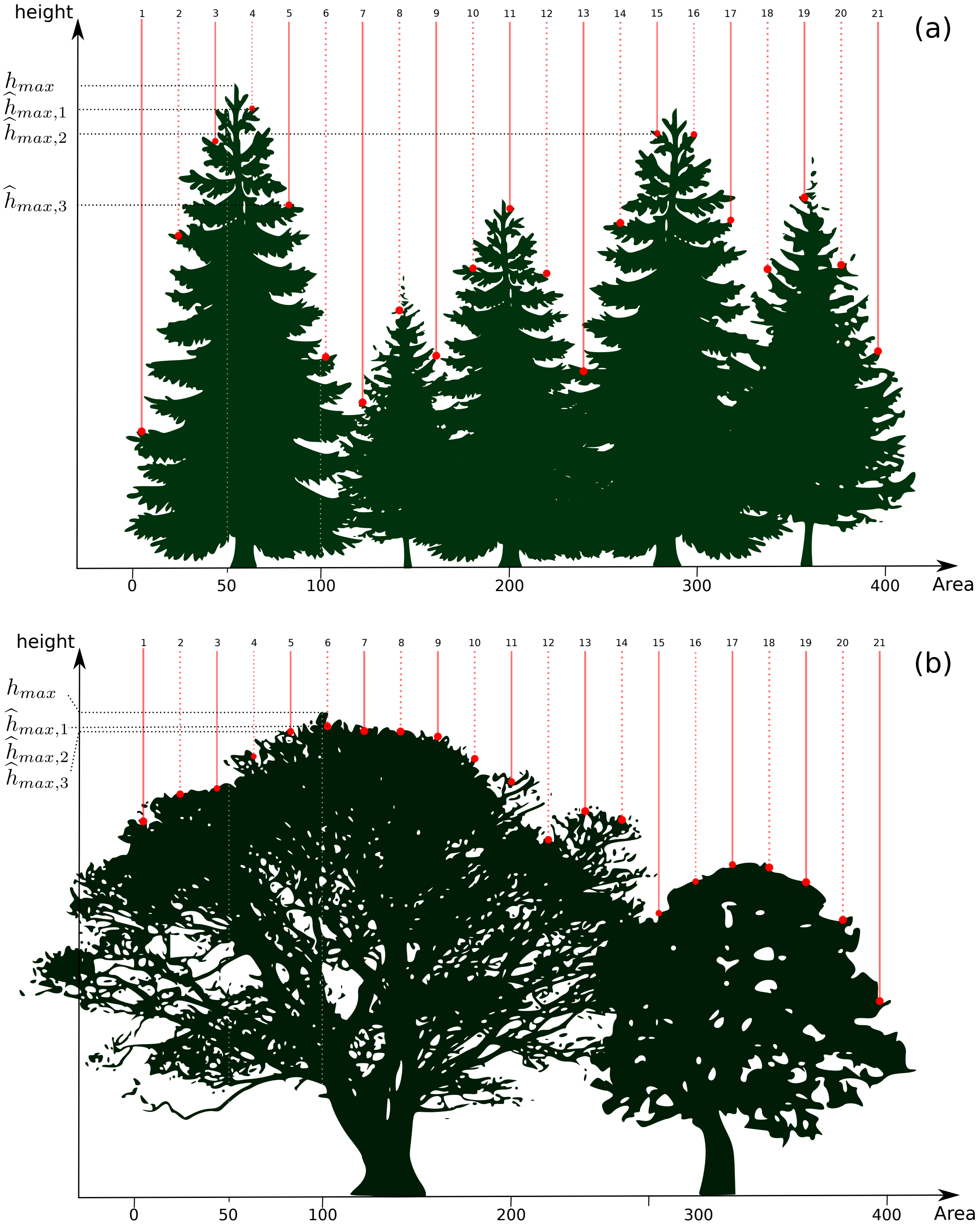


FIGURE 2.5 – Dependence of bias on pulse density, pixel area and canopy shape. h_{max} is the true maximum height, $\widehat{h}_{max,i}$ is the observed maximum height in the following three scenarios : $\widehat{h}_{max,1}$) a pixel (400 m^2) sampled by 21 pulses ; $\widehat{h}_{max,2}$) a pixel sampled by half as many pulses ; $\widehat{h}_{max,3}$) a smaller pixel (between 50 and 100 on the x axis) sampled at the same density as scenario 1. As explained in the text, comparing panels (a) and (b) serves to illustrate how bias depends on canopy shape.

nical crowns, forming a “rougher” canopy with larger variation in the observed maximum height. As described below, our model also accounts for the effect of canopy shape.

2.4.6 Quantifying canopy shape

We used the HD dataset to quantify canopy shape as accurately as possible. As shown in figure 2.6a, we used the original data to calculate the number of pulses that returned from each of many different height intervals, and thereby generated “canopy histograms” that provide both a visual and quantitative assessment of vertical variation in the height of return. The number of returns in each bin reflects the probability that a pulse returns from a given height, so the shape of the histogram reflects the vertical distribution of return heights.

Since our goal is to quantify the bias between the true maximum height and the observed value, our point of reference is not the ground but the true maximum height in a given pixel area. Thus, we standardized the histograms by subtracting the local maximum from the height of each return, such that local maximum equals zero, and all the other returns are negative. This is illustrated both for the plot pixel and the canopy pixel in figure 2.6b and figure 2.6c, respectively.

Because they are examples, the histograms in figure 2.6 were obtained using a subset of the HD dataset, but for the purpose of our analyses we used the entire HD dataset to generate one histogram for each scale (i.e. one for canopy pixel and one for plot pixel). These two histograms were used to quantify the average shape of the canopy in the megaplot, and ultimately the magnitude of bias when estimating the true canopy height, as explained in section 2.5.3.

2.4.7 Validation of the model

Comparing two corrected datasets : HD vs. LMD

We used the model (equation 2.1) to correct the height metrics calculated from the HD and LMD datasets (Figure 2.4). The goal was to validate the model by showing that adding the density-dependent error term to the estimated height yielded the same result for both the HD and LMD megaplots. For the model to be valid, the correction must remove the fixed difference in height between the two datasets (Figure 2.4), resulting in one-to-one relationship between the two datasets. The correction must also increase the goodness-of-fit of the relationship by taking into account secondary sources of density variation within the datasets, such as those attributable to speed and attitude variations.

Comparing corrected flightlines from the same dataset

Secondary sources of density variation were isolated by separating the flightlines of the entire LMD dataset, calculating the metrics for each flightline individually, and comparing the repeat estimates of canopy height obtained from plot pixels that were surveyed in two flightlines (and therefore have two independent estimates). For this analysis, our goal was

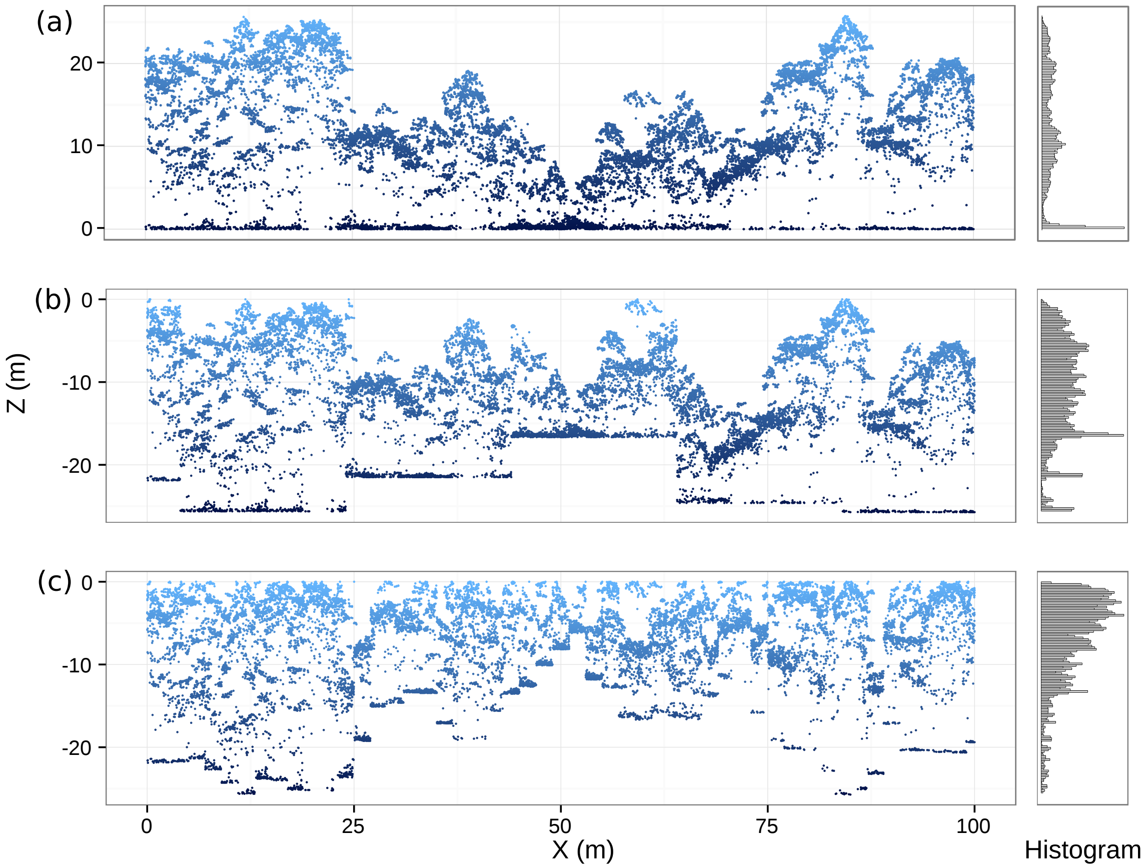


FIGURE 2.6 – Canopy histograms generated using a subset of the HD dataset from the megaplot (a strip of 100×4 m). (a) Original data, (b) standardized at the plot pixel scale, (c) standardized at the canopy pixel scale.

to further validate the model by showing that adding the density-dependent error term to the estimated height reduced the expected bias between pairs of measurements.

For pixels sampled with the same pulse density in adjacent LMD flightlines, the average bias between repeat estimates should be approximately 0. However, there is appreciable variation in pulse density about the mean, which yielded a range of differences in pulse density among the 150 000 plot pixels included in the analysis. Prior to correction, we expected a positive correlation between the difference in height and the difference in pulse density. Adding the density-dependent error term should remove any such correlation, indicating that any residual difference between repeat estimates is unrelated to pulse density.

By applying the correction to the entire LMD dataset, we are assuming that the HD dataset is representative of Haliburton as a whole. In particular, we are assuming that the structure of the canopy in and around the megapot is similar to that of Haliburton as a whole. While the HD dataset does encompass a fairly large area (36 ha), the average canopy shape may differ somewhat, given that the megaplot itself is largely comprised of old-growth forest (20 ha) that has never been harvested.

Footprint size as a potential source of residual bias

We cannot assume that variation in pulse density is the only source of bias, because Hirata (2004) showed that footprint size affects canopy height estimates. Even if the density-dependent correction described above were perfect, there may be appreciable residual bias between the HD and LMD datasets because the footprint of the LMD dataset is approximately ten times larger. Thus, we developed a method for testing whether beam divergence causes additional bias, using the spot pixels that approximate the size of one LMD footprint, yet contain multiple footprints from the HD dataset. The goal of the analysis is to compare the height of the local maximum to an estimated "equivalent height" of one LMD footprint, as explained in section (section 3.5).

2.4.8 Tools used

Data pre-processing and processing was done in the R programming environment (R Core Team, 2015). A purpose-built package named `lidR` was specifically developed for processing LiDAR data (Roussel et Auty, 2017). The source code for implementing our model is provided in the appendix.

2.5 A probabilistic model of bias

2.5.1 Notation

The following notation is used to describe the model :

h_{max} : true maximum height for a given area
$\widehat{h}_{max}$: observed maximum height for a given area
$\bar{h}$: expected (or most probable) maximum height for a given area
$\mathscr{P}(E)$: probability of event E
p **or** P : letters used for a probability
X : a random variable

SI base units are used for numeric application of the model.

2.5.2 Quantifying bias using idealized canopy shapes

Section 2.4.5 described how discrete sampling leads to the underestimation of h_{max}. This section demonstrates how to quantify the underestimation of h_{max} using a probabilistic model. Rather than simply presenting the mathematical derivation of the model, we use diagrams of idealized canopy shapes to illustrate how the bias can be quantified probabilistically.

The probabilistic nature of the underlying sampling process can also be understood by analogy with rolling loaded dice. In particular, when a canopy divided into k-height bins (Fig. 2.6) is sampled with n pulses, the expected maximum height is equivalent to the expected value when rolling a k-sided dice n times. The fact that the k-sides are not equally likely to land face up (in a loaded dice) is analogous to the fact that k-height bins are not equally likely to return a pulse (Fig. 2.6). The probabilistic nature of this sampling process should become clearer after reviewing the four cases below.

2.5.3 A perfectly flat canopy

If the canopy were a perfectly flat surface (Fig. 2.7a), the observed maximum height can only take one value. This canopy shape is represented by a histogram with one bin (shown on the righthand side of fig. 2.7a), indicating that a pulse can return from only one height (h_0) , with a probability (p_0) of 1. In this simple case, the observed maximum height ($\widehat{h}_{max} = h_0$) will always be the true maximum height ($h_{max} = h_0$), regardless number of pulses (n). Thus, the expected value of $\widehat{h}_{max}$, denoted by $\bar{h}$ and expressed as a function of n, is :

$$\bar{h}(n) = p_0^n \times h_0 = h_0 \tag{2.2}$$

The expected value (or most probable value) is the mean value that would be found if we sampled the surface an infinite number of times. Indeed, a computer simulation of the sampling process confirms this simple mathematical result (compare expected and simulated in figure 2.7b), which is hardly surprising in this trivial case, but serves to illustrate that our model captures the underlying sampling process, both in this case and the non-trivial cases discussed further below (for all four cases, we ran 1,200 simulations, including 200 replicates at each of 60 sampling densities).

A flat canopy with one singularity

If we add a singularity to the otherwise flat surface (fig. 2.8), the observed maximum height can take two values. In this case, the canopy histogram (fig. 2.8a) includes two bins at heights h_0 and h_1. If we sample this surface at random with a single pulse, the probability of observing the maximum at h_0 is p_0, which implies that the probability of observing h_1 is $p_1 = 1 - p_0$.

To express the expected value using standard notation, let X be a random variable that takes the value 1 when the pulse returns from height h_1, and 0 otherwise. X follows a Bernoulli distribution, $X \sim \mathrm{B}(p_1)$, so the expected value of $\widehat{h}_{max}$ for a single pulse is :

$$\begin{aligned}\bar{h}(1) &= \mathscr{P}(X = 1)h_1 + \mathscr{P}(X = 0)h_0 \\ &= p_1 h_1 + (1 - p_1)h_0\end{aligned} \tag{2.3}$$

When randomly sampled with n independent pulses, only one has to return at h_1 to find the true maximum height. This process corresponds to n independent iterations of a Bernoulli process, and therefore follows a binomial distribution. Now, let X be a random variable that counts the number of times h_1 is missed. X follows a Binomial distribution,

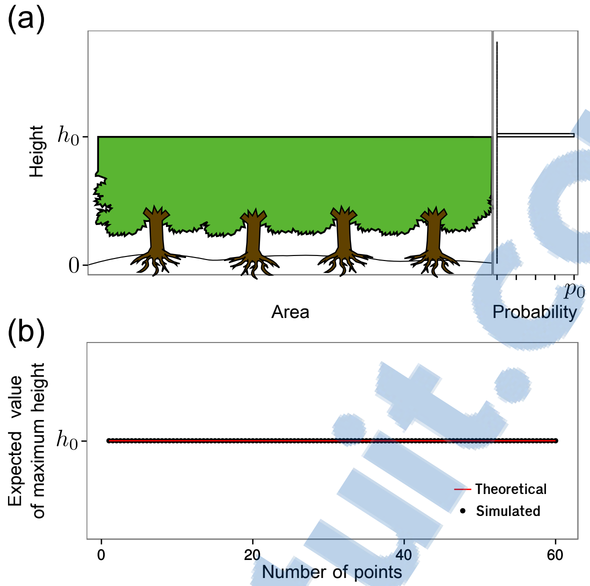


FIGURE 2.7 – (a) Probability of a pulse returning from height h, assuming a perfectly flat canopy. A pulse can only return from h_0 with probability $p_0 = 1$, as shown by the histogram on the right-hand side. (b) The observed maximum height, calculated ($\bar{h}$) and simulated ($\widehat{h}_{max}$) as a function of the number of points used to sample the surface (n) : $\bar{h}$ is the expected value ; $\widehat{h}_{max}$ was simulated by repeatedly sampling from the surface (200 times per density).

$X \sim \mathrm{B}(n, p_0)$, so the probability that h_1 is the observed maximum height ($\widehat{h}_{max}$) is the probability that *at least one* of the n pulses returns at height h_1 :

$$\begin{aligned}
\mathscr{P}(X < n) &= 1 - \mathscr{P}(X = n) \\
&= 1 - \binom{n}{n} p_0^n (1-p_0)^{n-n} \\
&= 1 - p_0^n \\
&= 1 - (1-p_1)^n
\end{aligned} \tag{2.4}$$

The expected value of $\widehat{h}_{max}$, expressed as function of the number of sampling points n is :

$$\begin{aligned}
\bar{h}(n) &= \mathscr{P}(X < n) h_1 + \mathscr{P}(X = n) h_0 \\
&= (1 - p_0^n) h_1 + p_0^n h_0 \\
&= \left(1 - (1-p_1)^n\right) h_1 + (1-p_1)^n h_0
\end{aligned} \tag{2.5}$$

Again, the expected value (or most probable value) is the mean value that would be observed if the canopy were repeatedly sampled with n pulses. Sometimes one or more of the pulses would return from the true maximum height h_1, but sometimes not, so on average there is bias.

As before, this is confirmed by a computer simulation : comparing the expected and simulated values in figure 2.8b shows that in both cases the observed maximum height first increases with pulse density, then approaches the true maximum (h_1) asymptotically. Thus, approximately 40 pulses are required to observe the true maximum height with high probability. At lower pulse densities, one or more of the pulses may return from the true maximum height, but on average there is bias, since many pulses will return at h_0, such that $\bar{h} < h_{max}$.

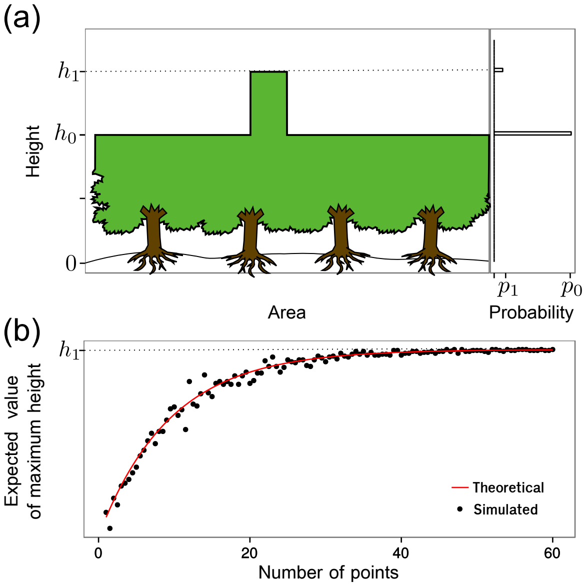


FIGURE 2.8 – (a) Probability of a pulse returning from height h, assuming a flat canopy with one singularity. A pulse may either return from h_0 or h_1, with probabilities p_0 and p_1, as shown by the histogram on the right-hand side. (b) The observed maximum height, calculated ($\bar{h}$) and simulated ($\widehat{h}_{max}$) as a function of number of points used to sample the surface (n) : $\bar{h}$ was calculated using equation 2.5 for the expected value ; $\widehat{h}_{max}$ was simulated by repeatedly sampling from the surface (200 times per density).

Our goal is to quantify the bias shown as a function of pulse density and canopy shape including more realistic canopy shapes. To do so, we must first introduce a more generic form of equation 2.5 that allows to write a generic form of the equation for canopies with more than one singularity. In particular, we need to re-express equation 2.5 using the following notation : $p'_1 = p_1 = \frac{p_1}{p_0+p_1}$ (because $p_0 + p_1 = 1$), then, let P^n_k be

$$P^n_k = 1 - \left(1 - \frac{p_k}{\sum\limits_{i=0}^{k} p_i}\right)^n \tag{2.6}$$

with $k \in \mathbb{N}$ and n still the number of points. We can see that :

$$P^n_1 = 1 - \left(1 - \frac{p_1}{p_0 + p_1}\right)^n = 1 - \left(1 - p_1\right)^n \tag{2.7}$$

Thus, P^n_1 can be substituted for p_1, which is equal to p'_1 in equation 2.3 (single pulse) to obtain equation 2.5 (n pulses) :

$$\bar{h}(n) = P^n_1 h_1 + \left(1 - P^n_1\right) h_0 \tag{2.8}$$

This generic form can also be expanded to quantify the expected maximum value when sampling canopies with more than two heights (see below).

A flat canopy with two singularities

If we add two singularities to an otherwise flat surface (fig. 2.9), the observed maximum height can take three values. In this case, the histogram includes a third bin, representing the probability (p_2) that a pulse returns from h_2, the true maximum height in this case. If we sample this surface at random with a single pulse, the probability of missing the true maximum height is $1 - p_2$. If h_2 is missed, we have now two other possibilities i.e. finding h_1 or h_0. For a single sampling point missing h_2, the probability to find h_1 and h_0 becomes $p'_1 = \frac{p_1}{p_0+p_1}$ and $p'_0 = 1 - \frac{p_1}{p_0+p_1}$, respectively. Because $p'_2 = p_2 = \frac{p_2}{p_0+p_1+p_1}$, $\bar{h}(1)$ can be written :

$$\bar{h}(1) = p'_2 h_2 + (1 - p'_2)\left(p'_1 h_1 + \left(1 - p'_1\right) h_0\right) \tag{2.9}$$

Again, we note that sampling with more than one pulse is a Binomial process. Thus, the expected value of $\widehat{h}_{max}$ can be calculated by substituting each of the probabilities (p'_i) with P^n_i , as we demonstrated for a canopy with one singularity (eq. 2.8). With two singularities, however, such a demonstration would be rather lengthy, so we only provide the final equation for the expected value of $\widehat{h}_{max}$:

$$\bar{h}(n) = P^n_2 h_2 + \left(1 - P^n_2\right)\left(P^n_1 h_1 + \left(1 - P^n_1\right) h_0\right) \tag{2.10}$$

The expected and simulated values in figure 2.9a again show that the observed maximum height increases asymptotically with pulse density, but only 20 pulses are required to observed the true maximum with high probability. The curve is steeper in this case because singularity number 2 is wider, and because singularity number 1 provides another value closer to the real maximum than h_0, as explained in section 2.4.5.

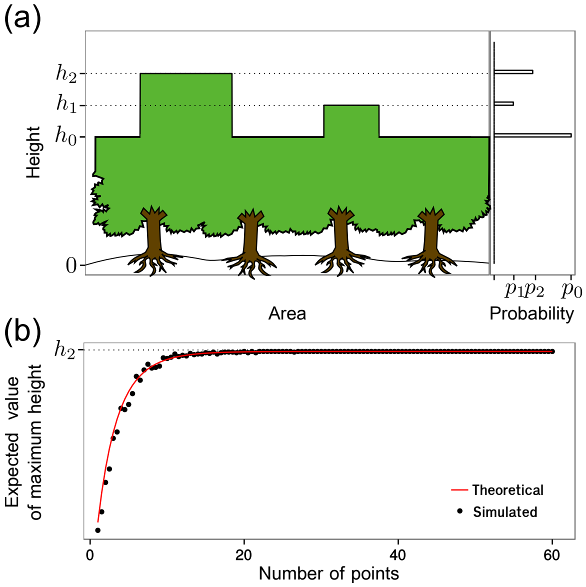


FIGURE 2.9 – (a) Probability of a pulse returning from height h, assuming a flat canopy with two singularities. A pulse may return from three different heights (h_0 to h_2), with probabilities (p_0 to p_2), as shown by the histogram on the right-hand side. (b) The observed maximum height, calculated ($\bar{h}$) and simulated ($\widehat{h}_{max}$) as a function of number of point used to sample the surface (n) : $\bar{h}$ was calculated using equation 2.10 for the expected value ; $\widehat{h}_{max}$ was simulated by repeatedly sampling from the surface (200 times per density).

A continuous canopy

As shown in figure 2.10a), a continuous canopy can be discretized using a histogram with k bins, one for each height (h_i) and probability (p_i), where $i \in [\![0,k]\!]$. When sampled with n pulses, the expected value of h_{max} is :

$$\bar{h}(n) = P_k^n h_k + (1 - P_k^n)\big[P_{k-1}^n h_{k-1} + (1 - P_{k-1}^n)(P_{k-2}^n h_{k-2} + \ldots)\big] \tag{2.11}$$

This equation can be simplified using its recursive form. Let $\mathbb{H}$ be the set of couples height/probability : $\mathbb{H} = \{(h_i, p_i) | i \in [\![0,k]\!]\}$. We can define :

$$H_i^n(\mathbb{H}) = \begin{cases} h_0 & \text{if } i = 0 \\ P_i^n h_i + (1 - P_i^n) H_{i-1}^n & \text{else} \end{cases} \tag{2.12}$$

Therefore :

$$\bar{h}(n) = H_k^n(\mathbb{H}) \tag{2.13}$$

The agreement between the expected and simulated values in figure 2.10b shows that this recursive function can be used to calculate $\widehat{h}_{max}$ for realistic canopy shapes like that shown in figure 2.10a, just as we did for the idealized canopies in figures 2.7a, 2.8a and 2.9a.

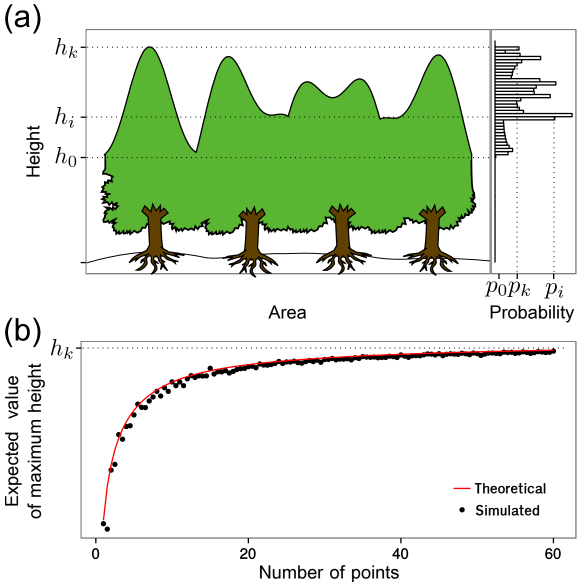


FIGURE 2.10 – (a) Probability of a pulse returning from height h, assuming a continuous canopy that is discretized using k bins, as shown on the right-hand side. (b) The observed maximum height, calculated ($\bar{h}$) and simulated ($\widehat{h}_{max}$) as a function of the number of point used to sample the surface (n) : $\bar{h}$ was calculated using equation 2.13 for the expected value ; $\widehat{h}_{max}$ was simulated by repeatedly sampling from the surface (200 times per density).

Quantifying bias using standardized histograms

Comparing the expected and simulated values has demonstrated that we can calculate $\bar{h}$ for any canopy shape, and that it varies as a function of both canopy shape and pulse density. Since h_{max} is the point of reference (section 2.4.6), the bias e (equation 2.1) is always negative and must be calculated using the standardized histograms. Let $\mathbb{H}_r$ be an histogram standardized with a resolution r :

$$e_r(n) = H_k^n(\mathbb{H}_r) \tag{2.14}$$

2.5.4 The effect of footprint size

Including the recursive function in equation 2.1 isolates a second error term that quantifies the bias associated with footprint size (δ) :

$$h_{max} = \widehat{h}_{max} + H_k^n(\mathbb{H}_r) + \delta \tag{2.15}$$

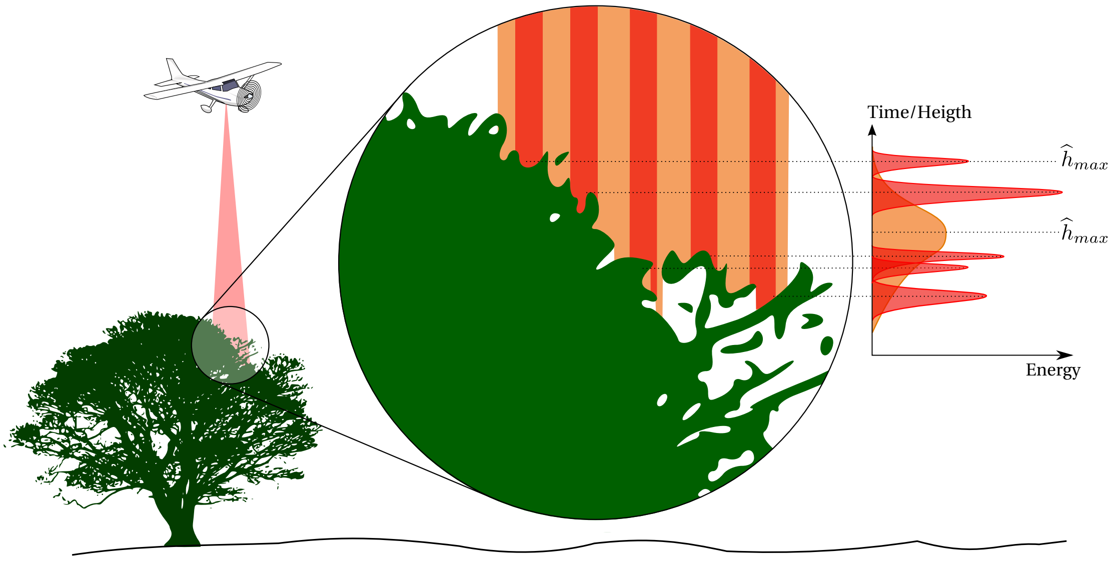


FIGURE 2.11 – The effect of footprint size on the observed maximum height ($\widehat{h}_{max}$). The red columns are small footprints, while the orange colour should be seen as a single column in the background belonging to a footprint that is ten times larger. The larger footprint has a broader waveform because the intensity is integrated over a larger surface area. As a result, the large footprint underestimates the maximum height recorded by the smaller footprints - i.e. the height at which the orange waveform peaks is lower than the highest red peak.

For the HD dataset, we can assume that the footprint is small enough to have a negligible effect, so we fixed δ at 0. However, we did estimate δ for the LMD dataset (as explained further below), since the footprint is ten times larger in the LMD dataset.

Figure 2.11 illustrates why large footprints are expected to underestimate maximum height. The individual red columns represent pulses with small footprints, and the larger orange column in the background represents a pulse that is ten times larger. As shown to the right, the waveform of small footprints is Gaussian with a small standard deviation, so the returned height is rather accurate. In contrast, the larger footprint records a broader waveform because the intensity is integrated over a larger surface area. As a result, the large footprint underestimates the height of the local maximum returned by the smaller footprints (i.e. the height at which the orange waveform peaks is lower than the highest red peak). This phenomenon is described in detail by Hancock *et al.* (2015) and Disney *et al.* (2010). Note that the large footprint is represented by Gaussian distribution, though in practice it may not be Gaussian (see Hancock *et al.* (2015)).

We used the spot pixels to estimate the bias for the LMD dataset, since they approximate the size of one LMD footprint (0.14 m^2), yet contain 3-15 pulses from the HD dataset. Our method consisted of subtracting the height of the local maximum obtained from the HD dataset from an estimated “equivalent height” of the LMD footprint. In other words, we estimated the average distance between the peak of the orange waveform and the peak of the highest red waveform (figure 2.11).

Assuming that canopy reflectivity did not change between the LMD and HD surveys, we estimated the height of the orange peak as the point at which the smaller footprints have returned 50% of their total intensity (figure 2.12). This method probably does not provide the correct value in every case, but by computing it over a large number of spot pixels, it can be expected to provide a good estimation of the average "equivalent height".

The Optech ALTM 3100 system used in this study emits pulses with a beam width (a function of pulse duration) of 1.02 m (Hancock *et al.*, 2015). This width is defined as half the distance between the points at which the power drops below 61% of the maximum. This implies that such pulses are unable distinguish two distinct objects that are located less than 50 cm apart in the direction of the beam axis. Thus, a beam that is 1 meter wide would be unable to distinguish between the first 5 returns in red (figure 2.12, but it would be able distinguish them from the other 3 beams in black, which were therefore be excluded when estimating the equivalent height of the orange beam. The value of δ was assessed for each spot pixel by subtracting this equivalent height from the height of the highest sampling point among the HD pulses it contained.

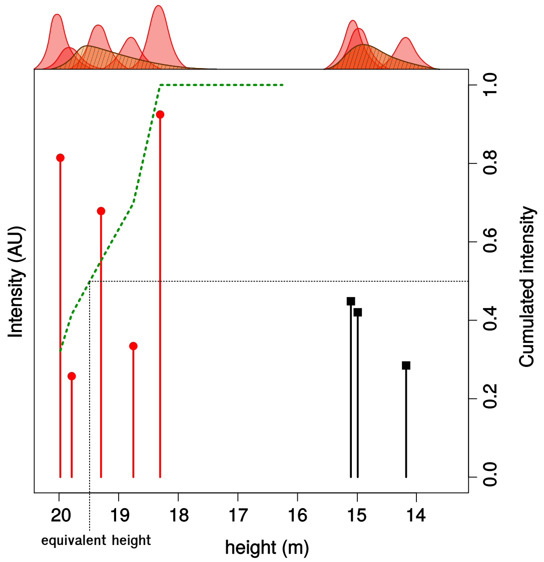


FIGURE 2.12 – Illustration of the "equivalent height" of a large footprint (LMD dataset), as estimated from many of smaller footprints (HD dataset). The vertical lines show the intensity of pulses that returned from two sets of objects, one set that is higher in the canopy (red with round end), and one set that is lower in the canopy (black with square end). The integrals at the top show the corresponding waveforms for the small (plain red) and large (stripped orange) footprints. The peak of the orange waveform is the "equivalent height" of a large footprint, and is estimated as the point at which the smaller footprints have returned 50% of their total intensity (cummulative intensity is shown in dotted green). Note that a large footprint (1 m wide) is only able to distinguish two objects, as indicated by the two orange waveforms.

2.6 Results

2.6.1 Expected value of the bias as a function of the scale of observation

As expected, the average canopy shape differed substantially (figure 2.13) when calculated using the canopy (4 m^2) and plot (400 m^2) scales (histograms are called $\mathbb{H}_4$ and $\mathbb{H}_{400}$). These two histograms must be used in conjunction with equation 2.15 to calculate the difference in bias for these two scales of observations.

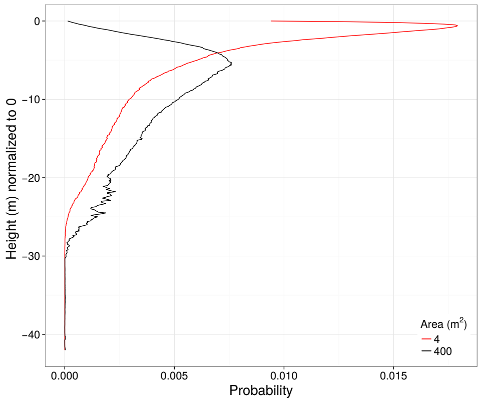


FIGURE 2.13 – Canopy shape in canopy pixels (4 m^2, $\mathbb{H}_4$) and plot pixels (400 m^2, $\mathbb{H}_{400}$).

Results show that there was less bias when estimating the h_{max} of plot pixels (figure 2.14). Approximately 10 pulses/m^2 (4 000 pulses) are required to estimate the h_{max} of plot pixels with reasonable accuracy (mean bias < 10 cm). At 30 pulses/m^2 (12 000 pulses), the bias is negligible.

A higher density of pulses is required for canopy pixels, even though they exhibit less variation in height. Approximately 20 pulses/m^2 (80 pulses) are required to estimate the h_{max} of plot pixels with a mean bias < 10 cm (figure 2.14).

2.6.2 Footprint size

The HD dataset included 1 160 000 spot pixels (0.14 m^2), of which 193 000 included a sufficient number of pulses (4 or more) to compute the equivalent height (eq. 2.15). On average, the equivalent height of an LMD footprint was 16 cm lower than the highest HD footprint (i.e. δ=16 cm).

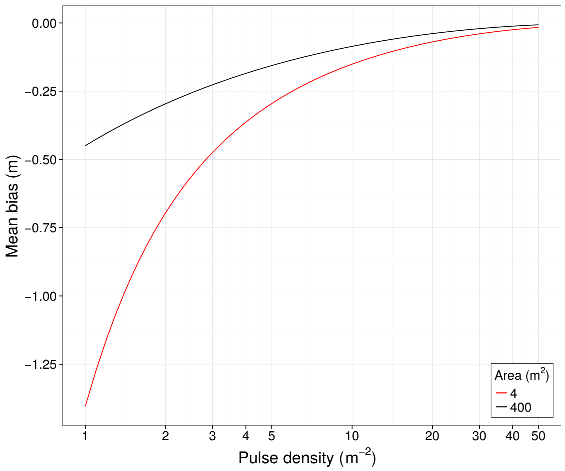


FIGURE 2.14 – Dependence of bias (equation 2.13) on pulse density and area sampled : 4 m^2 in canopy pixels and 400 m^2 in plot pixels.

2.6.3 Comparing two corrected datasets : HD vs. LMD, effect of device configuration

h_{max}

The original bias between the HD and LMD datasets was -57 cm, on average (figure 2.4a). We applied a correction using the black line in figure 2.14 :

$$h_{max} = \widehat{h}_{max} + H_k^n(\mathbb{H}_{400}) + \delta \tag{2.16}$$

After correcting each plot pixel individually in the LMD dataset, h_{max} increased by 56 cm, on average, with a range of 41 cm to 75 cm (figure 2.15). In contrast, h_{max} only increased by 2 cm for the HD dataset, with a range of 1 cm to 8 cm. The bias between the two corrected datasets was reduced to 7 cm, on average.

The original goodness-of-fit (R^2) between the HD and LMD datasets was 0.976. After correcting both datasets, the R^2 was increased to 0.977, and the RMSE of the regression was reduced to 1.17 cm from an initial value of 1.34 cm (figure 2.15).

C_{mean}

As explained in section 2.4.3, C_{mean} is computed using the local maxima from 100 canopy pixels (4 m^2), each of which should be corrected using the red line in figure 2.14. However, correcting each canopy pixel individually is computationally demanding, and would be even more so if the canopy surface model were computed at a higher resolution (in other applications). For this reason, we chose to apply an average correction based on pulse density of the plot pixels. This way, the correction was the same for every canopy

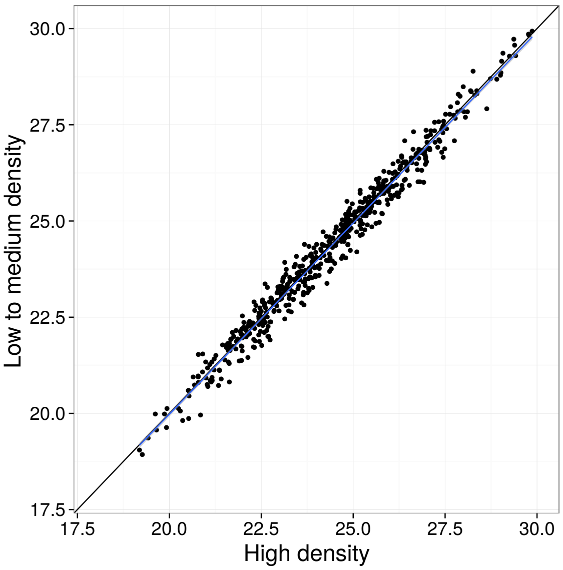


FIGURE 2.15 – Comparison of the corrected maximum heights (h_{max}) from the HD and LMD datasets, including 586 plot pixels (400 m^2) from the megaplot.

pixel and it was computed only once at the plot pixel scale :

$$C_{mean} = \widehat{C}_{mean} + H_k^n(\mathbb{H}_4) + \delta \tag{2.17}$$

The original bias between the HD and LMD datasets was -1 m, on average (figure 2.4b). After correcting each plot pixel individually in the LMD dataset, C_{mean} increased by 82 cm on average, with a range of 30 cm to 1.70 m. In contrast, C_{mean} only increased by 4 cm for the HD dataset, with a range of 2 cm to 14 cm. The bias between the two corrected datasets was reduced to 7 cm, on average.

The original goodness-of-fit (R^2) between the HD and LMD datasets was 0.978. After correcting both datasets, the R^2 was increased to 0.983, and the RMSE of the regression was reduced to 31 cm from an initial value of 36 cm (figure 2.16).

To test the validity of using an average correction for each plot pixel, we also corrected each canopy pixel in the megaplot individually, then repeated the analyses described above. The results did not differ substantionaly from those obtained using an average correction at the plot scale (not shown), so we concluded that an average correction could be applied to the entire LMD dataset.

2.6.4 Comparing corrected flightlines from the same dataset : effect of aircraft attitude

On average, there was no difference between the repeat estimates of mean canopy height (C_{mean}) obtained from LMD flightlines that sampled the same plot pixels twice (Fig. 2.17a). Prior to correction, however, the difference in mean canopy height (C_{mean})

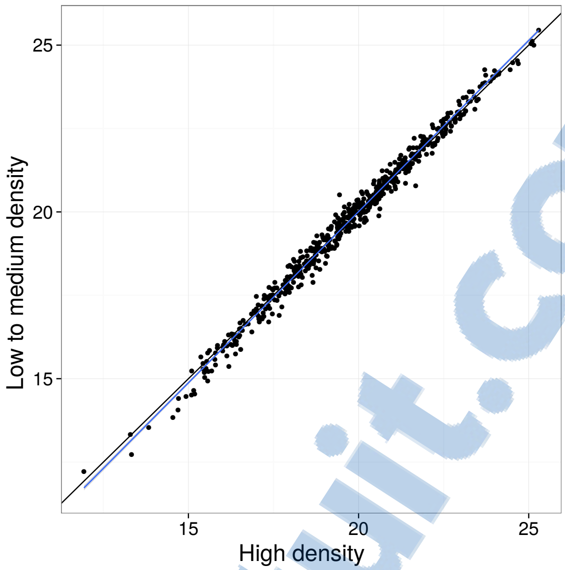
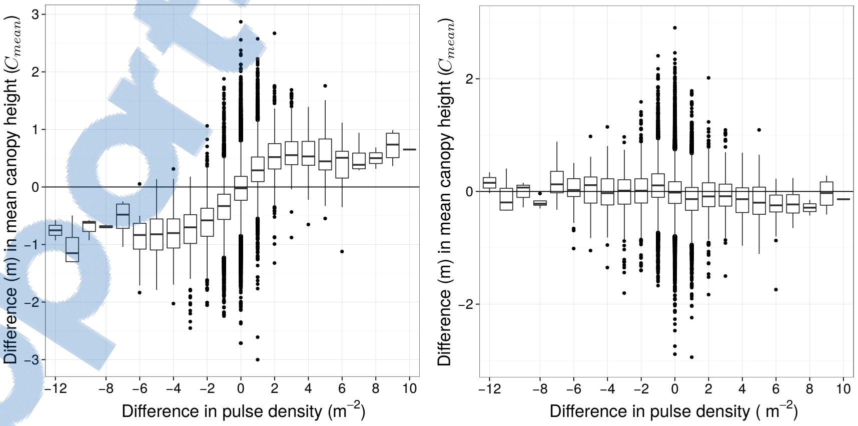


FIGURE 2.16 – Comparison of the corrected mean heights (C_{mean}) from the HD and LMD datasets, including the 586 plot pixels (400 m^2) from the megaplot.

was positively correlated with the difference in pulse density, with the bias reaching more than 50 cm at either extreme. After correction, this correlation was largely removed (Fig. 2.17b), indicating that any residual difference between repeat estimates is unrelated to aircraft speed and attitude.



(a) Before correction

(b) After correction

FIGURE 2.17 – Difference between repeat estimates of mean canopy height (C_{mean}) obtained from LMD flightlines that sampled the same plot pixels twice (150 000 observations). The pixels were binned by the difference in pulse density, and box plots were used to visualize the correlation between the two differences, both before (a) and after (b) correction.

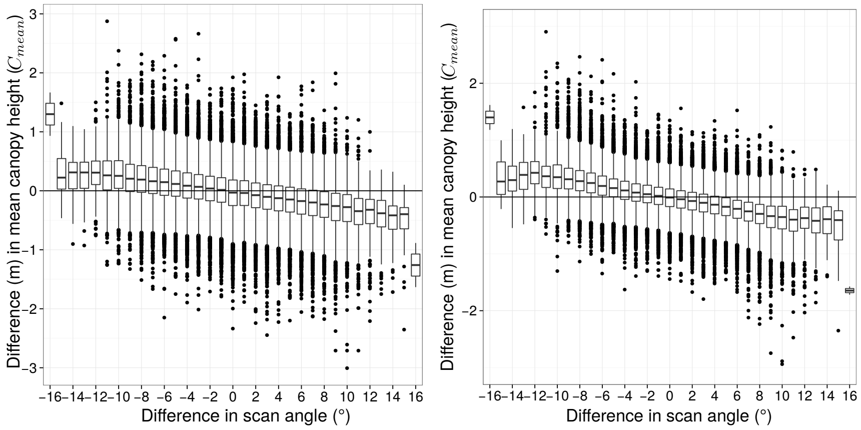
We obtained the same result for h_{max} (not shown), but the correlation was weaker, because h_{max} is less sensitive to the pulse density due to an effect of scale.

2.6.5 Accuracy of quantifying canopy shape with the LMD dataset

All the results above were generated using canopy histograms obtained from the HD dataset. To test whether a LMD dataset could be used instead (as in an application for which only one dataset is available), we repeated all the analyses after using LMD data from the megaplot to generate the histograms. The results were approximately the same, but the residual biases were slightly higher at 13 cm and 20 cm for h_{max} and C_{mean}, respectively (results not shown).

2.6.6 Effect of scan angle

While our method removed most of the bias associated with variation in pulse density, there was considerable residual bias attributable to scan angle. Following the same procedure as that applied in section 2.6.4 for pulse density, we compared plots from separate flightlines based on their mean angle of incidence. Figure 2.18 shows a clear effect of scan angle, which is not accounted for in our correction method. This effect was not significant for h_{max} (results not shown).



(a) Before correction

(b) After correction

FIGURE 2.18 – The effect of scan angle as revealed by the difference between repeat estimates of mean canopy height (C_{mean}). Rasters that were sampled in two flightlines were binned by the difference in scan angle, and box plots were used to visualize the correlation between the two differences, both before (a) and after (b) correction (150 000 observations).

2.7 Discussion

2.7.1 On the usage of h_{max} and C_{mean}

The correction of h_{max} and C_{mean} proposed in this study was derived from the initial question we raised about metric normalization. Our capacity to understand and describe the underlying sampling process was the most important factor determining our choice of metrics for this study. Nevertheless, correcting biases for these metrics is also important in practice. Even if h_{max} can be avoided in predictive models in favour of other less density dependent metrics such as lower percentiles, its use remains common whether it is in a direct or indirect form. The latter can occur, for example when metrics are defined based on a layerization of the point cloud. For example, Woods *et al.* (2008) defined a metric d_n which can mathematically be expressed as :

$$d_n = \int_0^{n\frac{h_{max}}{10}} f(z)dz \tag{2.18}$$

with n being an integer between 1 and 9 and $f(z)$ the probability distribution of points on the z axis. In this article the term "maximum height" is never used and equation 2.18 is not provided, but a careful interpretation of the metric description leads us to state that each d_n is biased because of the indirect use of h_{max}. This can be referred to as a second order usage of h_{max}.

The case of C_{mean} is another example of indirect usage of h_{max}. We found only three other examples of this metric being used in the literature (Ruiz *et al.*, 2014; Asner et Mascaro, 2014; Coomes *et al.*, 2017). However, it remains an interesting metric with potential applicability in the development of predictive models of forest structure. Most metrics used in ABA models are unidimensional metrics derived from the z coordinate only. But with the LiDAR point cloud being, at least, a tri-dimensional dataset it can be argued that it is reductive to use only one of them. Features extracted from the canopy surface model represent an easy and accessible way to extract information from the three spatial coordinates. For example, it can be used to extract information on the texture of the forest canopy. The interest of such metrics derived from canopy surface model is recognized and they have been used in the literature (e.g. Kane *et al.*, 2010; Ruiz *et al.*, 2014; Asner et Mascaro, 2014). However, our study shows that they must also be used and interpreted with caution. The choice of the algorithm used to compute the canopy surface model is not without consequences. The local maximum algorithm is a simple, easily implementable algorithm which is used in the recently developed `itcSegment` R package (Dalponte, 2016). A careful study of the source code for this package shows the canopy is computed using such an algorithm with a linear interpolation and smoothing as post process. Ruiz *et al.* (2014) computed a canopy surface model in the same way except for the use of an inverse distance weighting interpolation.

The availability of such tools implies that users have the possibility to derive various types of metrics from a canopy surface model. As highlighted in the introduction, such metrics are currently being provided in an operational context. In some cases, this may create bias issues which may be corrected using the probabilistic approach proposed in this study. Beyond this, our analysis of the behaviour of h_{max} at different scales provides a case study that may help raise general awareness about the fact that various metrics can be more or less sensitive to device parametrization, forest structure, footprint size, plot size, etc. It also demonstrates that such variations may not always be trivial.

2.7.2 Removing bias from estimates of canopy height

We have shown that our model can be used to remove the bias in maximum height (h_{max}) and the bias in the mean height of the canopy surface model (C_{mean}), both of which are substantial when the LiDAR data is collected at a low to medium pulse density. We have also shown that there is considerable variation in pulse density within a single dataset (figure 2.1), and that the resulting biases can be removed by our model as well.

The asymptotic relationship that we observed between bias and pulse density is similar to that observed by Hirata (2004), who found that the number of trees located using local maxima reaches an asymptote at 10 pulses/m^2 and higher, but decreases sharply below 3 or 4 pulses/m^2. Similar asymptotic relationships have also been described by Jakubowski *et al.* (2013) and Hansen *et al.* (2015). Our model also describes the underlying mechanism leading to a plot size dependency, as found by Hansen *et al.* (2015). However, these authors described the relationships empirically, whereas we modelled it based on probability theory. Furthermore, our model not only provides a mechanistic explanation of the underlying sampling process, but also the means to correct the resulting bias. Our model focuses on two specific metrics, but each of the additional questions raised in these cited references remain driven by the probability theory, and are thus more likely to be understandable in a model rather than from descriptions of local observations.

Our conclusion is that the new metrics obtained from our analytical model are accurate and correspond to what would be computed if the data were sampled with an infinite pulse density. This assertion is reasonable if an error of 10 or 15 cm is considered acceptable. Our results do not suggest that the residual error can be further reduced using our method, but the gain in accuracy compared to using the raw data remains substantial. Our results imply that caution should be used when building predictive models from such uncorrected metrics. It is difficult to generalize their effects on predictions because they depend on multiple factors such as the model used, the plot size, the dataset used, the device settings, the forest type and the model calibration method. However, in a homogeneous and dense hardwood forest the effect is expected to be rather low. Conversely, in a sparse coniferous forest it could be more important.

2.7.3 Effect of footprint size

The effect of footprint size on height bias has received little attention in the literature. One of the few studies Hirata (2004) was conducted in mountainous terrain, and found that large footprints overestimate the maximum height recorded by smaller footprints, the opposite of what we found. However, this result may be specific to mountainous terrain, because it was explained based on geometric considerations related to topography and slopes. Furthermore, the footprint sizes were much larger, reaching 1.1 m^2, nearly ten times larger than the footprint of the LMD dataset. In experimental conditions closer to ours, **?** found a similar effect of underestimating tree heigh of few centimetres.

Our correction method produced good results both for h_{max} and C_{mean}, indicating that it is reasonable to attribute the remaining bias to footprint size. However, the footprint correction should only be seen as a plausible explanation for the residual bias. Whether or not it is the real cause remains debatable. Indeed there could be additional error caused by the non-random distribution of pulses. Our model assumes that pulses are randomly and uniformly distributed in space, but in reality they follow a clear scanning pattern (a seesaw wave). This model assumption, required to make the mathematical development, may have an influence that we believe to be negligible compared to the gain in accuracy that we obtained. Another source of residual bias could come from the unknown pre-processing done by the provider. For example, the point classification step may have differed between the HD and LMD datasets.

2.7.4 Implications for the state of the art

Predicting stand structure and monitoring growth

We have shown that h_{max} and C_{mean} are systematically underestimated unless a sufficiently high pulse density is used to approach the asymptotic values. The density-dependence of LiDAR metrics may limit the applications of the aerial LiDAR technology, especially when two datasets sampled with different parameters need to be joined (different contracts for a large area) or compared (two datasets are sampled at a five-year interval to monitor forest growth).

Using the same pulse density in each inventory is not a solution to this problem, because pulse density changes substantially within a single dataset, as seen in figure 2.1. For example, the model predicts that C_{mean} is 50 cm higher in overlaps where the pulse density is twice as high, whereas h_{max} is 10 cm higher. Homogenizing the pulse density within a dataset could be a good way to avoid this problem, but removing points will introduce more uncertainty. Indeed, a metric can be seen as the single realization of a random variable, which is thus associated with a given uncertainty. A higher point density implies a lower uncertainty. Removing data willfully would not make much sense as it equates to adding noise in otherwise more accurate data. Therefore, a correction of metrics based on a hyphothesis-driven approach appears preferable.

In pratice, it is unlikely that a separate high density dataset would be available to generate the canopy histograms. We propose that this step could be achieved using local areas of high sampling density as a reference. The interface between overlaps and zones where aircraft pitch correction has further increased the sampling density could be used, for example. Since we found that a very high pulse density was not necessarily required (section 2.6.5), this solution should provide satisfying results. For larger areas than that used in our study, we suggest using a moving window to build a correction profile field, which would then be defined in any location. Hence, we consider that several solutions can be implemented to apply our model to any low density dataset acquired over any forest type. However, since our main focus was to present a formal description of the underlying sampling process, providing more specific guidance for practical applications is beyond the scope of this study.

Other high percentiles of height

The 99th percentile is often used instead of maximum height (e.g. García *et al.*, 2010; Singh *et al.*, 2015; Goodwin *et al.*, 2007), because it is both representative of maximum height while being robust to the noise caused by any possible outliers García *et al.* (2011). However, similarly to maximum height, the 99th percentile is also subject to underestimation, but by a smaller value due to the slightly higher probability of sampling it. The same logic of decreasing underestimation applies to the 98th percentile and each subsequent percentile. The bias is expected to decrease until the quantiles eventually become statistically stable. These hypotheses have been empirically tested in other results (fig 2 in supplementary materials) and we found that percentiles become very stable around the 90th percentile in our dataset, but the biases become positive after the 80th. While the methodology and the equations developed in this study could likely be transferred to the higher percentiles, the full mathematical development would undoubtedly be much more complex than for the limit case of the maximum height. The theoretical quantification of this effect for other percentiles is beyond the scope of this study.

2.7.5 Alternatives to the local maximum algorithm

We used the local maximum algorithm because it is the simplest method to compute the canopy surface model, and because it has the dual advantage of being amenable to analysis using probability theory, while also allowing an easy assessment of scale dependency. However, it is not necessarily the most widely used algorithm in practice.

Some algorithms modify the results obtained from the local maximum method (e.g. Popescu, 2007). They consist of computing the local maximum at a high resolution and filling the holes with an interpolation algorithm. Interpolation may render mathematical analysis more difficult to solve, but the preliminary considerations made in section 2.4.5 remain applicable. Obviously, a careful study of the effect of LiDAR parameters on interpolated canopy surface model would still be required.

Triangular irregular networks are also commonly used (e.g. Maltamo *et al.*, 2004; Zhao *et al.*, 2009; Asner et Mascaro, 2014). Metrics derived from this kind of representation may also be unstable, so we can also expect some artefacts and side effects. Thus, a careful study of the effect of LiDAR parameters on the canopy surface models produced by this type of algorithm would also be required.

2.7.6 A more complex issue than usually portrayed

In the majority of cases, the effect of pulse density or scan angle is studied in the literature in an overly simplistic way that does not correctly represent reality. We have shown that the problem is much more complex than what is suggested by a simple artificial reduction of pulse density. In reality, variations of pulse density are accompanied by variations of aircraft altitude, and therefore of footprint size. The footprint size changes the behaviour of the rays, allowing them to penetrate more or less easily into the canopy. Aircraft altitude changes are also be accompanied by variations in other parameters like aircraft speed, scan frequency, and emitted pulse frequency. Such changes modify the sampling pattern over the forest and the shape of the full waveform returns.

Furthermore, the results of empirical experiments are only valid locally, and they do not elucidate the underlying mechanisms. There is therefore a need for mathematical models that provide a mechanistic explanation of the underlying sampling process. Our model was only designed to recompute two metrics while accounting for two effects, and yet the model is still rather complex, despite the fact that we did not have to consider how the beams penetrate the canopy (we analysed only h_{max} at different scales). Modelling the effect of pulse density and other parameters using all returns, for example, would be much more challenging because of penetrating beams.

Admittedly, we chose only two of many possible metrics because they were relatively easy to model. But there remains need to model other metrics, and not only as a function of pulse density, but also of the scan angle, the footprint size, the pulse duration or the scanning pattern.

2.8 Conclusion

The metrics used in an area based approach do not represent absolute values, meaning that they depend not only on forest structure but also the LiDAR device, its settings, and the pattern of flight. As a result, some metrics are systematically underestimated, and we have shown that the magnitude of bias depends on pulse density, canopy shape, observation scale and probably footprint size. Furthermore, we developed a model that explains the observed bias and allows us to recompute the metrics as if the density of pulses were infinite, while also controlling for the effect of footprint size and observation scale.

This is a first step towards developing what we refer to as a standardization method, that consists of recomputing metrics as if they were obtained using a "standard device" and "standard parameters". It follows a similar approach to that currently used to correct for variations in signal intensity within and between datasets. The ultimate goal is to describe the behaviour of all metrics as a function of the most important device parameters, such as pulse density, scan angle, footprint size, pulse duration or emitted energy.

It is important to bear in mind, however, that data providers consider some information to be proprietary, such as the algorithms used to discretize the full waveform signal and classify the points, as well as some details about the sensors. These details inevitably introduce variability that is impossible to model if the information is not made available to the end-users.

2.9 Additional figures (supplementary material)

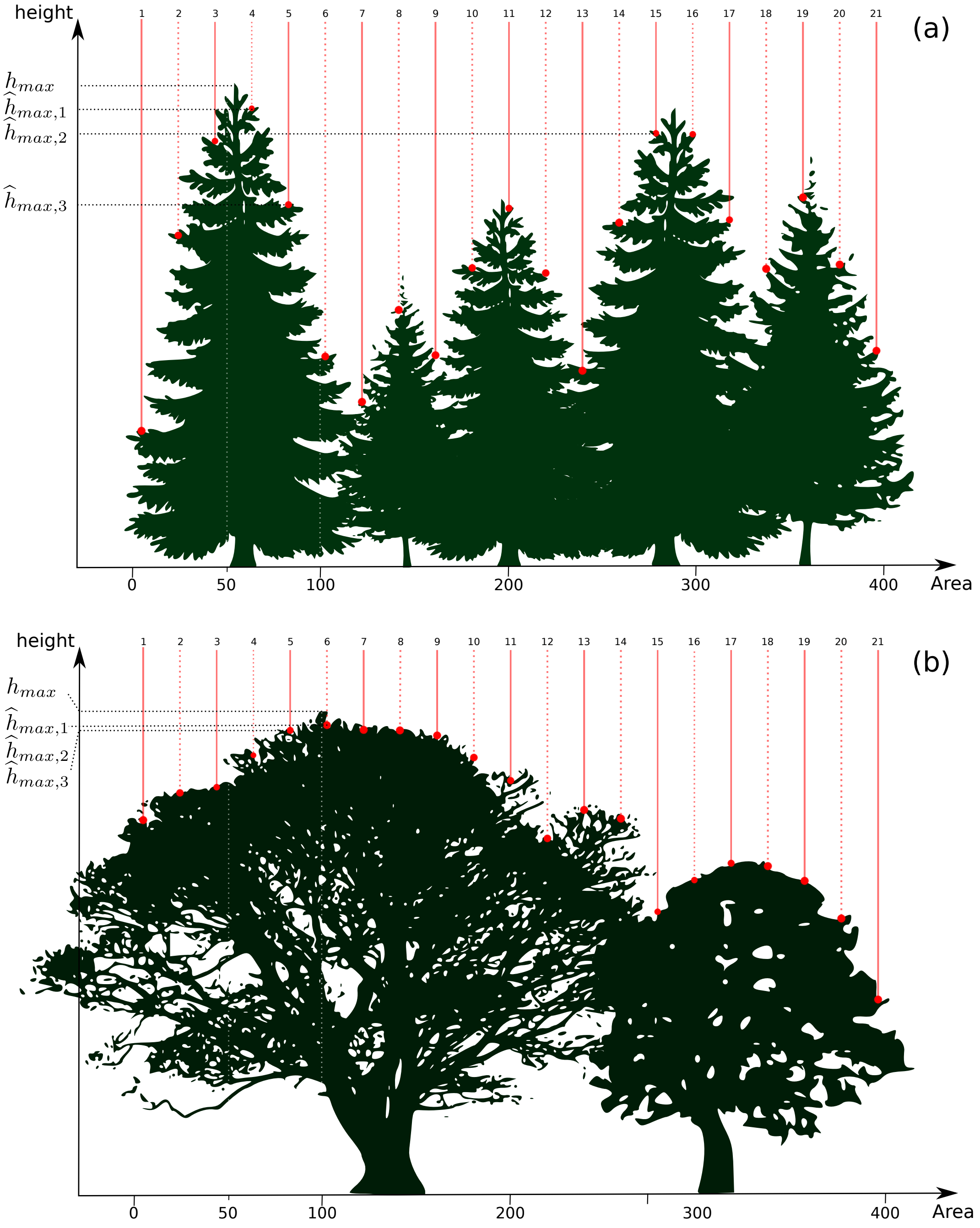
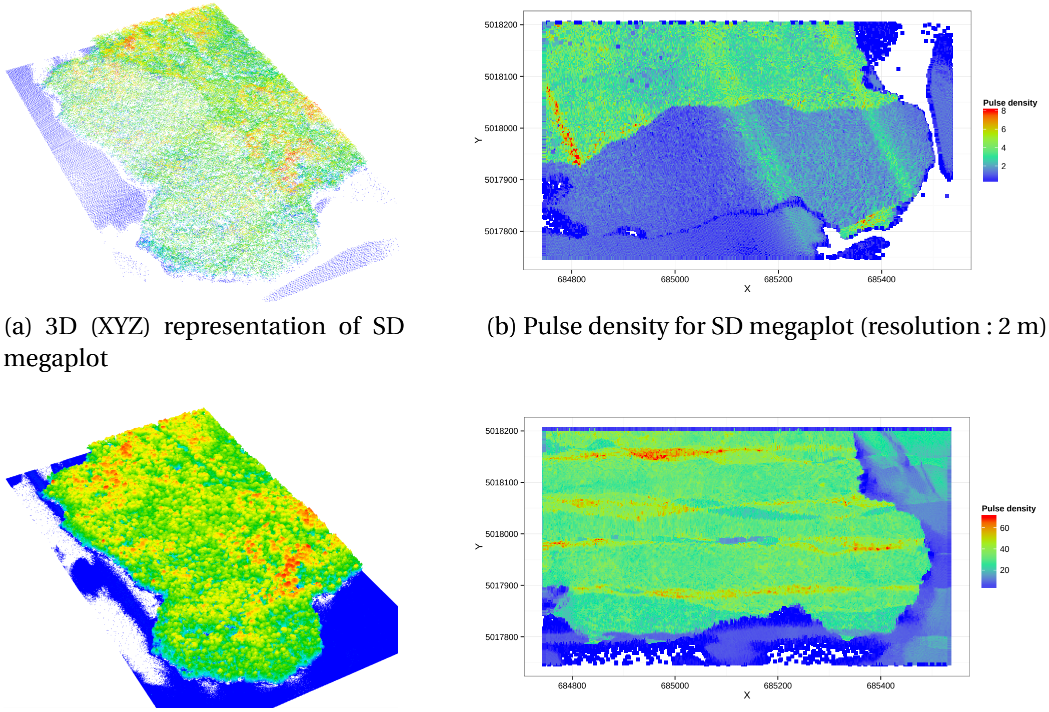


FIGURE 2.19 – Comparison of effect of missing local maximum for two different types of forest. Because the softwood forest have an higher level of roughness, it is expected to have a more pronounced effect.



(a) 3D (XYZ) representation of SD megaplot

(b) Pulse density for SD megaplot (resolution : 2 m)

(c) 3D (XYZ) representation of HD megaplot

(d) Pulse density for HD megaplot (resolution : 2 m)

FIGURE 2.20 – 3D representation of megaplots and plot of the local pulse density.

Chapitre 3

Effet de l'angle d'incidence sur les métriques unidimensionnelles dérivées de z

Le précédent chapitre met en évidence qu'il est possible, avec une approche théorique, de recalculer certaines métriques « comme si elles avaient été calculées avec une densité de points infinie ». Le problème a été résolu pour deux métriques spécifiques, mais des travaux similaires doivent être menés pour d'autres métriques, notamment les quantiles élevés qui sont sensibles à la densité de points pour les mêmes raisons que la hauteur maximale.

L'article 1 conclut sur le fait qu'il existe une dépendance à l'angle d'incidence des rayons pour la métrique C_{mean}. L'article 2 aborde logiquement la recherche d'un modèle théorique de normalisation de l'angle d'incidence pour recalculer les métriques comme si elles avaient été échantillonnées au nadir.

Nous proposons ainsi un modèle faisant quelques considérations physiques simples permettant de recalculer toutes les métriques dérivées de la distribution verticale des points comme si toutes les impulsions avaient été émises au nadir. Pour ce faire, ce deuxième chapitre s'abstrait, encore une fois, du sujet d'étude afin de créer un modèle qui vient en amont de la donnée, indépendamment du contexte forestier, en faisant l'hypothèse qu'une impulsion arrivant avec un angle d'incidence plus important traverse une distance plus grande et a donc une probabilité plus grande de rencontrer un obstacle. Il s'agit en fait de la loi de Beer-Lambert communément utilisée en chimie des solutions. La forêt est alors assimilée à une solution à faible concentration.

Théorique et très généraliste, ce modèle a été confronté à de vraies données et s'avère décrire parfaitement le comportement moyen observé. Ce modèle correspond à un cadre conceptuel idoine pour penser et modéliser les effets de l'angle. Il n'apporte pas de solution définitive à proprement parler ; il apporte une façon d'aborder le problème et met en évidence comment la structure locale de la forêt influence l'effet de l'angle d'incidence.

Bien que capable de faire des prédictions théoriques sur des choses non encore observées, ce modèle ne résout pas la question de la dépendance à l'angle d'incidence de la métrique C_{mean}. En revanche, ce modèle permet de retrouver – sans le chercher – le résultat que h_{max} n'est pas sensible à l'angle d'incidence des rayons. En effet, h_{max} est invariant par la fonction Q décrite dans le modèle.

Bien que la question de la dépendance à l'angle d'incidence de C_{mean} n'est pas résolue dans cet article, nous avons ici un modèle d'effet d'angle très généraliste qui semble pertinent sous certaines hypothèses. Si C_{mean} ne rentre pas dans ce modèle c'est que C_{mean} n'est pas une métrique purement verticale mais qui est indirectement dérivée des 3 coordonnées spatiales x, y, z. Le modèle présenté ici ne rend pas compte des effets sur ce type de variables explicatives. C'est donc un troisième modèle qui doit rendre compte de ce comportement.

3.1 Résumé

Le LiDAR aéroporté est utilisé dans l'inventaire forestier pour quantifier la structure des parcelles en utilisant un nuage de points tridimensionnel. Cependant, la distribution tridimensionnelle des points ne dépend pas seulement de la structure des parcelles échantillonnées mais aussi de l'angle d'incidence des rayons car la probabilité qu'un rayon soit réfléchi par la canopée augmente avec la distance qu'il doit parcourir au travers de la canopée. Ainsi la canopée semble avoir une plus grande densité à mesure que l'angle d'incidence augmente, toutes choses égales par ailleurs. Les variations résultantes entre et au sein des jeux de données peuvent engendrer des biais dans les métriques LiDAR dérivées de la distribution verticale des points. Dans cette étude, nous avons modélisé l'effet de l'angle d'incidence sur la structure verticale du nuage de points pour prédire les biais des métriques dérivées du nuage de points lorsqu'elles sont échantillonnée au-delà de nadir. La comparaison de paires d'observations provenant de différentes lignes de vol (observations hors nadir et à nadir pour les mêmes points) démontrent que le modèle reproduit précisément les biais des métriques observées dans une forêt de feuillus nordiques dont la canopée est relativement continue. Ainsi, le modèle pourrait être utilisé pour corriger les biais de mesures des métriques LiDAR et apporte un cadre mathématique qui pourrait être utilisé pour sélectionner un angle maximum d'incidence des rayons lors d'une acquisition en considérant le compromis entre les coûts d'acquisition et le besoin d'obtenir des mesures non biaisées.

3.2 Abstract

Airborne laser scanning (LiDAR) is used in forest inventories to quantify stand structure with three dimensional point clouds. However, the 3D distribution of the point clouds depends not only on stand structure, but also on scan angle, because the probability for an oblique beam to be reflected by the canopy increases with the distance it must travel through the canopy. Thus, the canopy appears to increase in density as the incidence angle increases, all else being equal. The resulting variation between and within datasets can induce bias in LiDAR metrics derived from the vertical distribution of points. In this study, we modelled the effect of scan angle on the vertical structure of the point clouds to predict the bias of metrics derived from points sampled off-nadir. Comparison with paired observations from different flightlines (off- and at-nadir observations of the same point) demonstrated that the model accurately reproduced the bias of metrics calculated for a northern hardwood forest with relatively continuous canopy. Thus, the model could be used to correct the bias of LiDAR metrics, and provides a mathematical framework that could be used to inform the selection of maximum incidence angle in LiDAR surveys, considering the trade-off between decreasing acquisition costs and obtaining unbiased measurements.

3.3 Introduction

Airborne light detection and ranging (or LiDAR) technology is increasingly being used in the field of forestry as a complement to traditional field inventories. This technology provides forest managers with detailed, continuous information on forest structure that can cover large areas and be processed rapidly with little need for human interpretation. Data processing relies partly on automated algorithms (e.g. Pyysalo et Hyyppä, 2002; Morsdorf *et al.*, 2004; Reitberger *et al.*, 2009; Kwak *et al.*, 2010; Yao *et al.*, 2012; Vega *et al.*, 2014) and partly on empirical statistical models (e.g. Holmgren, 2004; Ioki *et al.*, 2009; Lim *et al.*, 2014; Bouvier *et al.*, 2015). Height and density metrics derived from the point cloud can be used to estimate the horizontal and vertical distribution of vegetation, which have various applications in forestry and ecology (Vauhkonen *et al.*, 2014). For example, forest managers use LiDAR to predict product recovery under different harvest prescriptions (Maltamo *et al.*, 2014).

LiDAR has brought a fundamental improvement in the quality of aerial inventories, which explains the rapid uptake of this technology by practitioners (Popescu *et al.*, 2002; Gleason et Im, 2012). It can be argued, however, that some aspects of this technology are not fully understood. For example, the literature does not provide adequate understanding of the effect of LiDAR sensor parametrization and flight pattern on the three-dimensional structure of the point cloud (Goodwin *et al.*, 2007). Thus, empirical statistical models of stand structure may only be applicable to a single forest, a single device and a single set of acquisition parameters.

While locally calibrated models meet user needs at a given point in time, each new LiDAR survey may require another calibration with new ground data, given that both the device and acquisition parameters change through time. Thus, the use of LiDAR technology for forest monitoring requires a better understanding of how changes in device settings and flight patterns affect the structure of the point cloud. Ultimately, users would benefit from being able to normalize any two sets of lidar-derived metrics as if they were acquired the same way.

While several studies have been dedicated to understanding how the density of emitted pulses affects various metrics and their prediction accuracy (Lovell *et al.*, 2005; Anderson *et al.*, 2006; Thomas *et al.*, 2006; Gobakken et Næsset, 2008; Lim *et al.*, 2008; Pirotti et Tarolli, 2010; Jakubowski *et al.*, 2013), few have examined the effect of incidence angle. Widening the scanning angle allows a larger area to be surveyed more rapidly and at a lower cost (Goodwin *et al.*, 2007; Evans *et al.*, 2009). However, the financial advantage of a wide scanning angle could be offset by significant biases in the derived metrics.

Despite uncertainties regarding its magnitude, the effect of scanning angle is unarguable when one considers the extreme case of a 89° angle of incidence, in which case metrics are obviously biased compared to those obtained from a vertical beam simply because of shadowing effects or because of the increased distance between the top of the canopy and the ground. There is therefore a gradual increase in bias between 0 and 89°, such that the canopy appears to increase in density as the incidence angle increases, all else being equal. Yet, the physical or geometrical phenomena leading to such effects, the magnitude of these effects at a given angle, and their consequences are still largely unknown. Thus, the ongoing debate about the choice of the maximum incidence angle would benefit from a better mechanistic understanding of angular bias, so that the trade-off between cost and prediction bias could be quantified.

Holmgren (2004) recommended limiting the scanning angle to 10° to prevent effects on forest metrics estimates, while Disney *et al.* (2010) proposed limiting it to less than 15° to avoid ground detection problems. However, these suggestions are hardly applicable in the case of large survey areas and in practice the maximum incidence angle is typically ±15-20°, with a will to increase it further.

No consensus can be drawn from studies that have attempted to quantify the effects of incidence angle on LiDAR-derived forest metrics. Holmgren *et al.* (2003b) and Lovell *et al.* (2005) provided simulations of non-divergent beams hitting conical, ellipsoidal or half-ellipsoidal solid (i.e. impermeable) digitally reconstructed trees. With such simplifications of the reality Lovell *et al.* (2005) showed a dependency of the predominant height of the canopy on maximum incidence angle. They pointed out that increasing the maximum incidence angle and keeping all other parameters unchanged leads to an overall decrease in pulse density. The measure of the predominant height was therefore biased, as was mathematically demonstrated by Roussel *et al.* (2017). Within a given scan, Holmgren *et al.* (2003b) showed that the percentiles of height tended to decrease with increasing incidence angle, using a simulation which assumed that trees were solid, impermeable objects. Lovell *et al.* (2005) also highlighted the importance of the incidence angle by showing that maximum tree height retrieval is less accurate at the scanning edges due to a more uneven spacing of LiDAR points. Goodwin *et al.* (2007) improved these simulations by using permeable half-ellipsoidal digital trees. Their results showed that larger incidence angles "*produced a higher number of foliage hits and increased beam interception probability at the forest stand scale*". They also demonstrated that higher incidence angles increase the crown area visible to a LiDAR pulse.

In their comparison of field measurements and LiDAR data, Holmgren *et al.* (2003a) did not find a statistically significant effect of the incidence angle on the estimation of a metric related to the height of dominant trees (i.e. mean of tree heights weighted by their basal area). Similarly, Næsset (1997) showed that the effect of the incidence angle was non-significant in a regression model used to predict forest basal area. These results do not necessarily contradict the previous results because all metrics may not be equally dependent on incidence angle. It is also possible that different effects can compensate for one another, so that some predictive models appear to be insensitive to the incidence angle. Morsdorf *et al.* (2008) also concluded that the effect of the incidence angle is not as evident as it may first appear.

Montaghi (2013) presented an interesting study on the effect of incidence angle, taking advantage of perpendicular flightlines -typically used for strip adjustment and calibration- to compare a large amount of data sampled at nadir and off nadir. Although the overly large number of t-test comparisons (~1300) implies that some of the statistically significant effects reported may only be attributed to random variation (1 out of 20 test at alpha = 0.05). Even with stronger statistical methods, this approach would lead to as many statistical models as metrics, i.e. one for each metric as they each have their own dependency to the incidence angle. Furthermore, these statistical models would be specific to a given dataset, which entails the same data-dependency issue described above.

An important issue with the current understanding of angular bias is that it has been acquired mainly through descriptive approaches. Indeed, mathematical models have rarely been used to explain how and why LiDAR metrics vary with incidence angle. Goodwin *et al.* (2007) and Disney *et al.* (2010) proposed that the probability a beam is reflected by the canopy increases as incidence angle increases, simply because the distance it travels through the canopy increases. Despite the plausibility of this explanation, a formal demonstration is still required.

In this study, we hypothesize that angular bias can be normalized by the distance a beam must travel through the canopy. We suggest a simple physical formalization of this hypothesis to quantify the effect of incidence angle on the vertical distribution of points. We then compare our theoretical model against real data to validate its relevance. Thus, rather than a very large number of empirical models derived for each metric, we propose a single overarching model that applies to all metrics.

3.4 Material and methods

3.4.1 Study area

The study area is located within the Haliburton Forest and Wildlife Reserve. The forest is a 32 000 ha privately owned property located in the Great Lakes - St. Lawrence Forest Region of central Ontario, Canada (45°13’ N, 78°35’ W). Elevation ranges from approximately 400 to 500 m above sea level. The forest is a mixture of hardwoods and conifers typical of northern hardwood forests, and sugar maple (*Acer saccharum* Marsh) is the dominant species, comprising 60% of the basal area. Most of the forest has been managed under selection silviculture for the past 50 years, and was selectively harvested before then. Thus, most of the stands are uneven-aged, with average canopy heights ranging from 20 to 25 m.

3.4.2 LiDAR data

The LiDAR dataset was acquired in August 2009 covering the whole 320 km^2 area of the Haliburton forest. It was acquired with a pulse density of approximatively 2 pulses/m^2 on average. The complete set of parameters is given in table 3.1.

Data pre-processing

The normalization of the dataset (i.e. the subtraction of the digital terrain model) was done by the provider. The method was based on triangular irregular network construction from returns classified as “ground”. Each point was interpolated, which implies that the

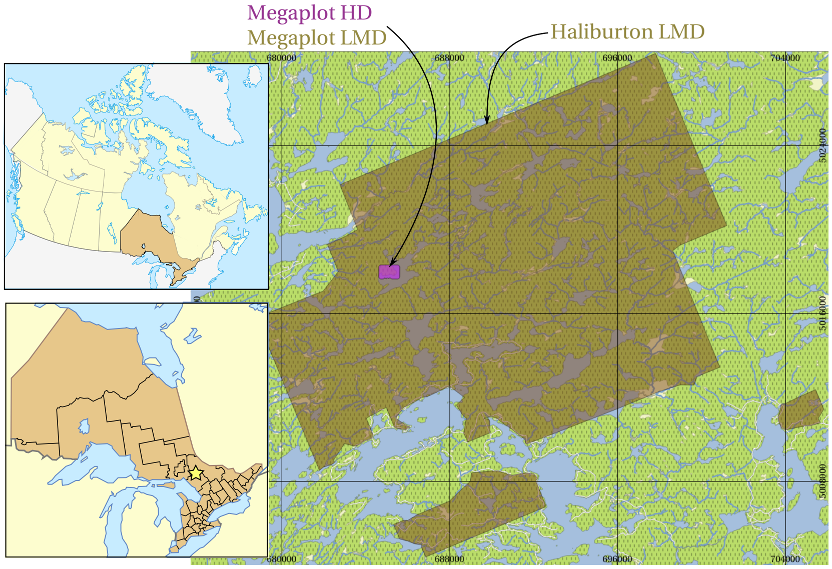


FIGURE 3.1 – Map of study areas. Left panels show the positioning of the study area in Canada. The star indicates the exact location of the study area in Ontario. In the right panel brown areas represent the boundaries of the LiDAR dataset.

TABLE 3.1 – Flight parameters for the datasets. PRF : pulse repetition frequency

Parameter	Value
Sensor	Optech ALTM 3100
Altitude	1500 m
Swath overlap	30 %
Speed	120 kts
Scan Frequency	36 Hz
System PRF	70 kHz
Max. off-nadir angle	16 °
Cross track resolution	0.89 m
Along track resolution	0.86 m
Point density	≈ 2 m^{-2}
Pulse density	≈ 1.6 m^{-2}
Footprint size[a]	0.14 m^2

[a] Beam divergence was not part of the data documentation. The footprint size was given instead.

normalization was not based on a digital terrain model, thereby giving a virtually infinite resolution. Further details about the algorithm used to determine point classes could not be obtained. We did not have access to the raw data.

Lakes and wetlands were filtered from the dataset in an attempt to retain only forested areas. The process was based on geographic data from the latest official cartography of Ontario, which spatially matched very closely with observed lakes and wetlands from our LiDAR datasets.

3.4.3 Conceptual framework

We aim to quantify the changes in the height distribution of returns that result from increasing the incidence angle, and hence both the distance a beam travels through the canopy and the probability that the beam is reflected by the canopy. In particular, we aim to develop a model that reproduces the resulting increase in canopy returns and the corresponding decrease in ground returns. Successful description of this incidence angle effect should enable normalizing the point distribution, and consequently every existing metric derived from elevations within the point cloud (i.e. classical metrics derived from z coordinates) as if all data had been sampled at-nadir.

Our approach consisted of developing a set of two mathematical expressions to predict how a point distribution sampled at-nadir would be altered if it had been obtained from another off-nadir angle. To validate the model, the predicted bias of various LiDAR metrics derived from such an "off-nadir" point cloud was compared to the bias observed between paired observations from different flightlines. Among the infinite number of metrics that could be derived, we chose nine representative metrics that describe various aspects of a distribution. We believe that if our model is capable of predicting the behaviour of these diverse metrics, then it can be considered an adequate description of physical reality.

3.4.4 Metric computation

We rasterized the dataset at the level of a “plot raster” i.e. a 20×20 m pixel, which is a commonly used resolution both in the literature and in applications that map quantities of interest using an area based approach (Woods *et al.*, 2011; White *et al.*, 2013). The flightlines were treated individually to avoid introducing variation related to the existence of overlaps in which rasters were sampled twice. For each plot raster we computed one control metric (the mean absolute incidence angle of the returns) and, using all returns, nine metrics derived from the distribution of elevations in the point cloud :

- The mean height of the returns ;
- The standard deviation of the heights ;
- The coefficient of variation of the heights ;
- The 30, 50 and 70th percentiles ;
- The kurtosis and the skewness of the distribution ;
- The entropy of the height distribution as labelled in the context of information theory. This index often called the “Shannon index” or “Shannon evenness index”

in forestry or ecology applications. van Ewijk *et al.* (2011) name it “vertical complexity index” in a paper dedicated to LiDAR.

These metrics were not chosen for their relevance in forest inventory applications, but instead because they represent classical descriptors of central tendencies, deviation and heterogeneity applicable to any distribution.

The mean absolute incidence angle was based on the scan angle rank i.e. the actual data stored according to LAS format specifications (ASPRS, 2013), which does not represent the real nadir angle. Zero may in reality be off-nadir, depending on aircraft roll angle. However, we did not consider this effect as it would only create negligible noise in our analysis.

3.4.5 Software

Data pre-processing and processing was done in the R programming environment (R Core Team, 2015). A package named `lidR` specifically developed for LiDAR data processing was used (Roussel et Auty, 2017). The R source code to compute the model is given as a supplementary material.

3.5 Model development

3.5.1 Initial considerations

The model relies on probability theory and on three simplifying assumptions : the first two relate to the proportion of energy that is backscattered toward a LiDAR sensor, the third relates to the distribution and orientation of material within the canopy.

A quantity E_0 of energy emitted by a LiDAR instrument will either i) be absorbed by the canopy, ii) be backscattered in any direction other than towards the sensor, iii) be backscattered towards the sensor with insufficient energy to generate a point (when using discrete LiDAR) and, finally, iv) be backscattered towards the sensor with sufficient energy to generate a point. The first three quantities are considered to be “non-contributing”, or “lost” energy, while the last quantity will be referred to as “contributing energy”.

First, we assume that beams have an infinitesimal width and carry only contributing energy. Indeed, we worked with a point cloud that, by definition, resulted exclusively from such contributing energy. Following this assumption, when such a beam encounters an object of the canopy, it can only be reflected towards the sensor and generate a point. Multi-returns were considered to come from multiple beams and we neglected multiple scattering.

Second, we assume that the proportion of contributing energy is not affected by the beam incidence angle. Under this assumption, what changes between two different angles is not the amount of contributing energy but only the distribution of this energy throughout the canopy, and therefore the distribution of the triggered returns, or points.

Third, we assume that at a given elevation, the probability of reflecting contributing energy is proportional to the number of canopy elements, or material density. As we did not have prior information about the vertical distribution of canopy elements within a canopy layer, we assumed a random distribution at any given elevation.

Under these assumptions, the model describes how to recompute a point distribution sampled at one angle "as if it were sampled at another angle". It relies on a set of two equations - the gap fraction profile function as presented in Bouvier *et al.* (2015) and its reciprocal function.

3.5.2 Notation

θ incidence angle
I event "beam interacts with a canopy element"
R event "beam is reflected"
A event "beam is absorbed"
$\underline{X}$ a random variable
$\overline{E}$ complementary event of the event E
$\mathscr{P}(E)$ probability of the event E
p_k probability for a given beam to generate a point in layer k
i_k probability of I in layer k for a given beam
r_k probability of R in layer k for a given beam
a_k probability of A in layer k for a given beam
Δz layer thickness

Gap fraction profile

The gap fraction describes the probability for a beam to reach the ground without encountering canopy elements. Bouvier *et al.* (2015) proposed an equation adapted to point clouds for computing the gap fraction profile as a function of height within the canopy (eq. 3.1). The authors used this equation to define a metric that could be used in a predictive model of biomass. We believe that it has further potential applications for modelling the behaviour of the LiDAR signal. The following lines provide a description of the Bouvier *et al.* equation and its interpretation. The gap fraction profile is defined in equation 3.1 (using the original notations) :

$$P_k = \frac{N_{[0;z]}}{N_{[0;z+dz]}} \tag{3.1}$$

where $N_{[0;z]}$ refers to the number of returns below z, and $N_{[0;z+dz]}$ refers to the number of returns below $z + dz$ with dz the thickness of a layer of forest. The equation expresses the number of laser returns that actually reached the layer $z + dz$ and those that passed through the layer $[z; z + dz]$. P_k represents the gap fraction of the k^{th} layer.

The gap fraction can be interpreted as the probability, for a single beam carrying contributing energy and reaching the layer k, of passing through this layer without interacting with canopy elements. In our model, we consider an alternative event I which expresses the probability i_k that a beam reaching layer k *interacts* with a canopy element in this layer.

$$i_k = 1 - P_k \tag{3.2}$$

As described further below, this probability is related to the height distribution of points, which itself can be interpreted as the probability, for a single beam carrying contributing energy, of generating a point in the k^{th} layer .

Let $p_k(\theta)$ be the probability that a beam generates point in the k^{th} layer at a incidence angle θ. According to equation 3.2, the probability of interacting with canopy elements in layer k is $i_k(\theta)$. If we assume that a beam interacting with a canopy element is always reflected towards the LiDAR sensor with sufficient energy to generate a return (contributing energy), events I and R are equal and $i_k = r_k$:

$$r_k(\theta) = 1 - \frac{\sum_{i=1}^{k-1} p_i(\theta)}{\sum_{i=1}^{k} p_i(\theta)} \tag{3.3}$$

Figure 3.2 illustrates how equation 3.3 can be used to calculate the probability of interaction from the height distribution of points in a hypothetical canopy with 5 layers that generate either 20 or 0 returns when sampled vertically. This example shows that to obtain the same number of points in each layer, the canopy must increase in density as the beams approach the ground. Accordingly, the probability of a beam interacting with a canopy element has to increase from top to bottom. The probability associated with the bottom layer is always 1 because of the presence of the ground (gap fraction is 0).

Figure 3.2 also illustrates how the interaction probabilities can in turn be used to recover the height distribution of points, which is the product of two probabilities - the probability that a beam reaching a layer interacts with it (i_k), and the probability of reaching the layer (i.e. the probability of passing through each previous layer). Thus, the height distribution can be recovered using the reciprocal function, f^{-1} :

$$p_k(\theta) = r_k(\theta) \prod_{i=k+1}^{n+1} (1 - r_i(\theta)) \tag{3.4}$$

with n being the number of layers. Note that for the equation to apply to the case where $k = n$, we added a virtual $n+1$ layer with a probability of interaction of 0 (see 3.9 for explanations and a mathematical demonstration).

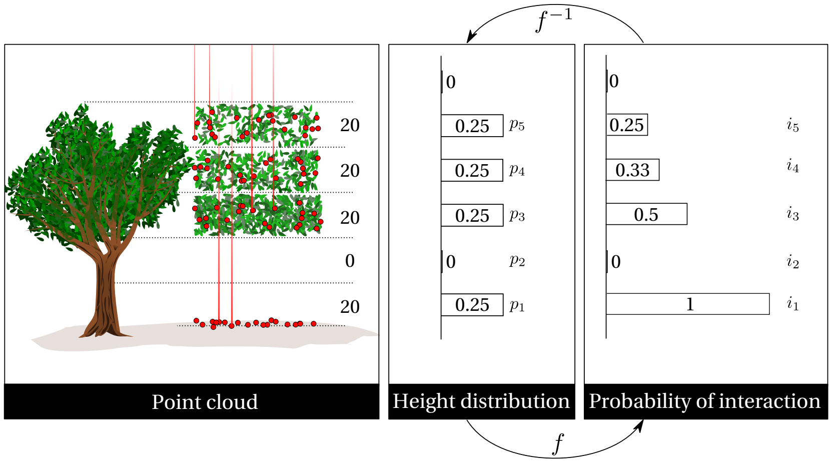


FIGURE 3.2 – Vertical sampling scenario illustrating how the probability of interaction is calculated from the height distribution of points, and vice versa. Function f (eq 3.3) is used to calculate the probability of interaction from the height distribution, and the inverse function f^{-1} (eq 3.4) is used to recover the height distribution of points from the interaction probabilities.

3.5.3 Effect of incidence angle on the height distribution of points

When a beam arrives at an angle of θ degrees, its travel distance through each layer is $1/\cos(\theta)$ times longer than that of a vertical beam. Thus, the probability of interacting with canopy elements increases with the incidence angle, and there is a corresponding change in the height distribution of points (i.e. the probability that a beam generates a point in any given layer).

As demonstrated further below, an approximate value of the probability of interaction can be calculated from $p_k(0)$ by including the factor $1/\cos(\theta)$ in function f^{-1} (eq. 3.4). For $k > 1$, $p_k(\theta)$ is the probability that a beam arriving at angle θ generates a point in layer k :

$$p_k(\theta) = \frac{r_k(0)}{\cos\theta} \prod_{i=k+1}^{n+1} \left(1 - \frac{r_i(0)}{\cos\theta}\right) \qquad (3.5)$$

Thus, the probability of interaction increases in each layer by a factor of $1/\cos(\theta)$, except the ground layer, which by definition always has a probability of interaction of 1.

This incidence angle effect is illustrated in figure 3.3, which shows that increasing the incidence angle from 0 to 45° increases the probability of interaction by $\sqrt{2}$, thereby shifting the expected height distribution upwards (increasing the number of canopy while decreasing the number of ground points).

To demonstrate that dividing by $\cos(\theta)$ in equation 3.5 provides a reasonable approximation of the increase in interaction probability at larger incident angles, we must consider the spatial distribution of elements in the canopy. If we assume that the spatial distribution is random, the number of elements in the path of a beam is a random variable X

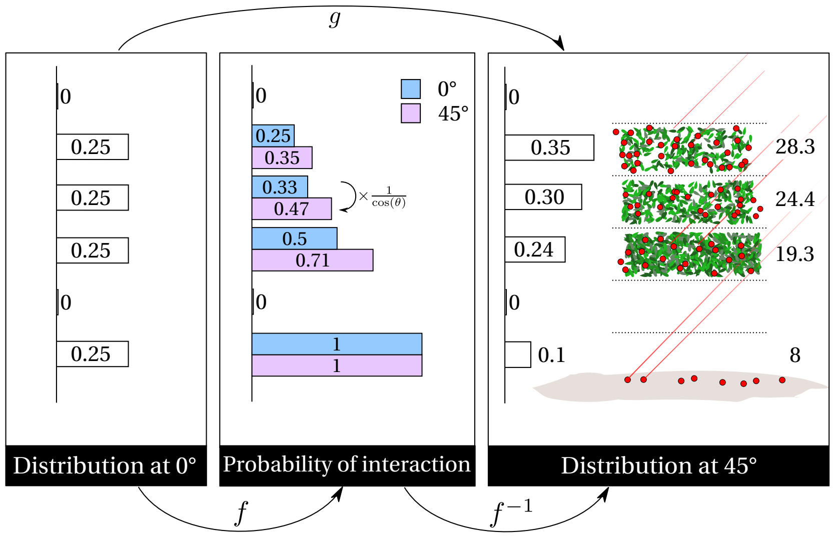


FIGURE 3.3 – Oblique sampling scenario illustrating how the expected height distribution of points sampled at 45° is calculated from the height distribution of points observed when sampled vertically (Fig. 1). The function f is used to calculate the interaction probabilities for each layer from the height distribution of points sampled at 0°. The inverse function f^{-1} is used to convert the interaction probabilities back into a height distribution specifying the probabilities that a beam generates a point in any given layer, given that it enters at an angle of 45°. The function g is the composition of f and f^{-1} ($g = f^{-1} \circ f$).

that follows a Poisson distribution (Nilson, 1971). Thus, the probability, when traveling a distance d through layer k, of encountering n elements is :

$$\mathscr{P}_k(X = n) = \frac{d\lambda_k^n e^{-d\lambda_k}}{n!} \tag{3.6}$$

and the probability of interacting with at least one canopy element in layer k is i_k :

$$\begin{aligned} i_k &= \mathscr{P}(X > 0) \\ &= 1 - \mathscr{P}(X = 0) \\ &= 1 - e^{-d\lambda_k} \end{aligned} \tag{3.7}$$

The quantity λ_k is related to the density of leaves in the k^{th} layer, as well as their orientation and spatial distribution. The meaning of this quantity is not required to solve the problem at hand, though readers can refer to Nilson (1971), Campbell et Norman (1990) or the discussion section for more details.

Since $i_k = r_k$, and the thickness of the layers Δz tends towards 0, we can simplify this expression using the first order Taylor expansion of the exponential near 0 :

$$\begin{aligned} r_k(0) &= 1 - e^{-\Delta z \lambda_k} \\ &= 1 - 1 + \Delta z \lambda_k + O\left((\Delta z \lambda_k)^2\right) \\ &\approx \Delta z \lambda_k \end{aligned} \tag{3.8}$$

For oblique angles, the distance Δz is increased by the inverse of cosine θ, and the probability of intersecting at least one element becomes :

$$r_k(\theta) = 1 - e^{-\frac{\Delta z}{cos(\theta)} \lambda_k} \tag{3.9}$$

Finally, Taylor expansion in the neighbourhood of 0 yields the factor included in function f^{-1} (eq 3.5) :

$$r_k(\theta) \approx \frac{\Delta z \lambda_k}{cos(\theta)} \approx \frac{r_k(0)}{\cos(\theta)} \tag{3.10}$$

Without the first order Taylor expansion the expression would be difficult to manipulate, but it must be noted that the approximation remains valid only if $\frac{\Delta z \lambda_k}{cos(\theta)}$ approaches 0. In other words, the approximation is correct only for thin layers and narrow incidence angles, as discussed in section 3.7.

With this approximation, we can then define a function, g (fig 3.3), using the composition of f and f^{-1} ($g = f^{-1} \circ f$), that enables us to calculate the expected height distribution of points that are sampled obliquely (fig. 3.3), taking into account the distance required to go through each layer. The source code of this function can be found in the supplementary materials.

3.5.4 Observed decrease in number of points per pulse : an unknown effect of incidence angle

We observed that in addition to increasing the probability of interaction, increasing the incidence angle also decreases the number of returns per beam (figure 3.4), which could result in an upward shift in the height distribution of points. Thus, we modified our model to provide an empirical way to account for this effect on the height distribution of points. The model does not provide a mechanistic explanation of this phenomenon, though we do provide two plausible explanations in the discussion section.

Given that a point is a spike of energy, the observed decrease in the number of points per pulse represents a decrease either in the amount of energy that is backscattered to the sensor, or in the amplitude of spikes, some of which become too low to be registered as a point. Thus, the decrease in the number of returns per beam can be represented empirically by allowing foliage to absorb contributing energy, and allowing the probability

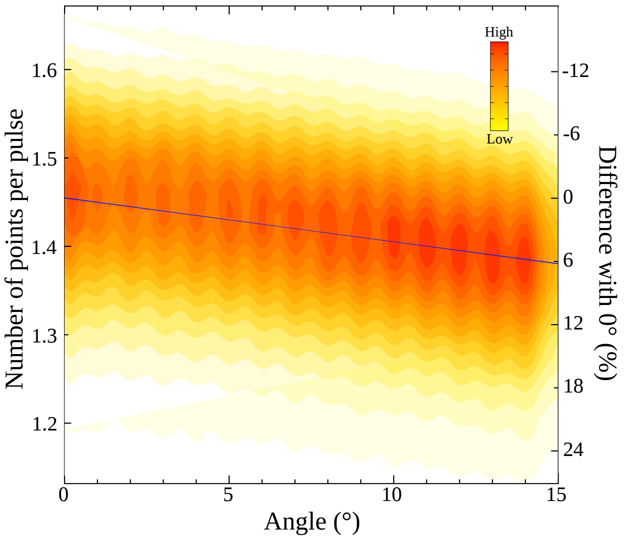


FIGURE 3.4 – Number of points per pulse as a function of the incidence angle modelled empirically using the linear relationship in 600 000 plot rasters covering the all of Haliburton forest. For better readability, the colour scale represents the density of observations. The blue line is the fitted linear regression. The confidence interval is narrower than the line at this scale.

of absorption to vary with incidence angle, similar to the probability of reflection. This implies relaxing the second assumption presented in section 3.5.1.

Let A be the event "beam is absorbed" and $a_k(\theta)$ its probability in layer k at angle θ. Thus far, we have assumed that a beam interacting with the canopy is always reflected ($I = R$), but now the energy is either reflected *or* absorbed : $I = R \cup A$. R and A are two disjoint events, thus :

$$\begin{aligned} &\mathscr{P}(I) = \mathscr{P}(R) + \mathscr{P}(A) \\ \Leftrightarrow\ &i_k(\theta) = r_k(\theta) + a_k(\theta) \end{aligned} \tag{3.11}$$

and equation 3.5 becomes :

$$p_k(\theta) = \left(\frac{r_k(0)}{\cos\theta} + a_k(\theta)\right) \prod_{i=k+1}^{n+1} \left(1 - \frac{r_i(0)}{\cos\theta} - a_i(\theta)\right) \tag{3.12}$$

Now, the height distribution of points depends on both the incidence angle and vertical variation in absorption. While it is unknown how a changes as a function of k, the probability of being absorbed was assumed to be proportional to the probability of being reflected. Thus, we introduced a proportionality function, α, to describe the variability in absorbed vs reflected energy at different incidence angles :

$$a_k(\theta) = \alpha(\theta)\, r_k(\theta) \tag{3.13}$$

While the proportionality function is constant with respect to k, the probability of absorption can vary as a linear function of θ, consistent with the pattern shown in figure 3.4.

Substituting this term into equation 11 leads to :

$$i_k(\theta) = r_k(\theta)(1+\alpha(\theta)) \tag{3.14}$$

Integrating this new expression into equation 3.12 leads to the final form of the expression :

$$p_k(\theta) = \left(\frac{r_k(0)}{\cos(\theta)}\big(1+\alpha(\theta)\big)\right)\prod_{i=k+1}^{n+1}\left(1-\frac{r_i(0)}{\cos(\theta)}\big(1+\alpha(\theta)\big)\right) \tag{3.15}$$

Using the data shown in figure 3.4, we can obtain an empirical estimate of the function α from $\epsilon(\theta)$, the overall probability of being absorbed by the canopy at a given incidence angle, as :

$$\begin{aligned}\epsilon(\theta) &= \sum_{k=1}^{n} a_k(\theta)\\ \epsilon(\theta) &= \alpha(\theta)\sum_{k=1}^{n} r_k(\theta)\\ \alpha(\theta) &= \frac{\epsilon(\theta)}{\sum_{k=1}^{n} r_k(\theta)}\end{aligned} \tag{3.16}$$

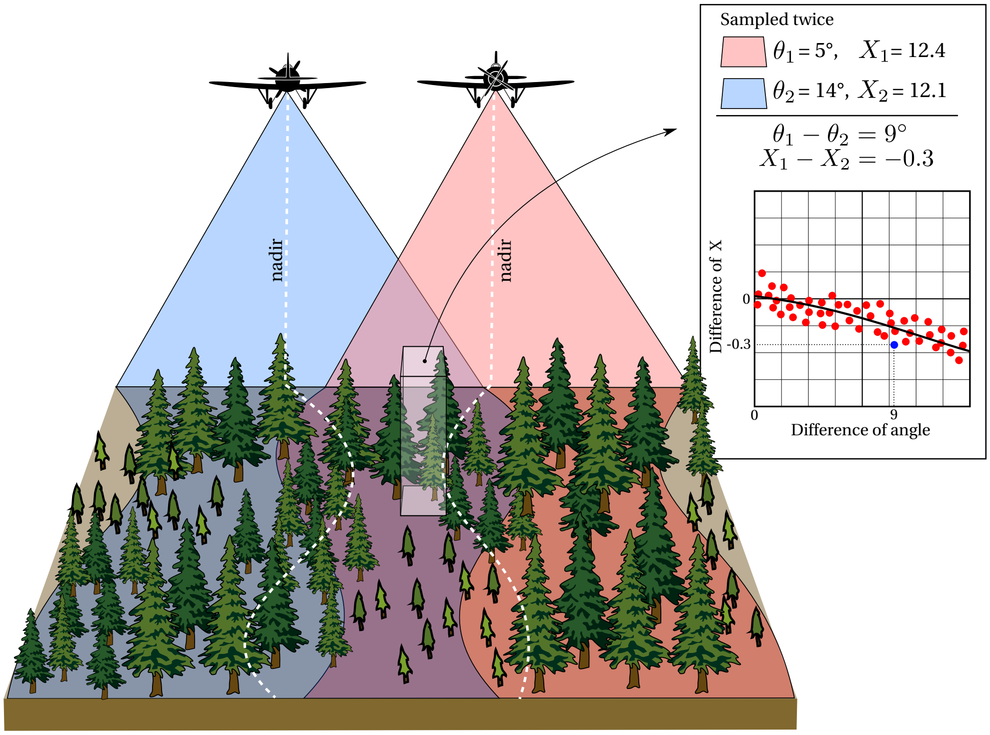
In this equation the only unknown term is the function ϵ. Based on our data and assuming that the relationship is almost linear at least between 0 and 15°, we have :

$$\epsilon(\theta) = \frac{\theta}{15}\epsilon(15) \tag{3.17}$$

with $\epsilon(15) \approx 6\% = 0.06$. Note that ϵ is the only empirical parameter of our model.

3.5.5 Model validation

The bias of LiDAR metrics can be expected to vary locally because they depend on the local forest structure. However, the number of returns available at the plot scale was insufficient to allow us to accurately apply our model to each 400 m^2 raster. Thus, to validate our model, we first predicted the average bias of each metric over the whole study area, then compared it to the average bias observed between paired flightlines. For this reason, our model only captures the average bias observed at the scale of the entire forest, not the plot-scale bias that will remain and be manifest as noise.



nadir
nadir
Sampled twice
$\theta_1 = 5°$, $X_1 = 12.4$
$\theta_2 = 14°$, $X_2 = 12.1$
$\theta_1 - \theta_2 = 9°$
$X_1 - X_2 = -0.3$
Difference of X
0
-0.3
0
9
Difference of angle

FIGURE 3.5 – Illustration of the method used to quantify the bias of LiDAR metrics using 150 000 plot rasters sampled twice from different flightlines. The bias is the observed difference in the LiDAR metrics obtained from the two flightlines (x axis of inset), which varies as function of the difference in incidence angle (y axis of inset).

To implement our model for the whole study area, we first extracted all points sampled at-nadir. Considering the `las` format specification (ASPRS, 2013) in which the incidence angle information is an integer, nadir (0°) corresponds to a incidence angle ranging from -0.5 to 0.5 °. This subset of the data provided an average height distribution of points sampled at-nadir, which was assumed to be representative of the entire forest because the sampling design and the forest structure were completely independent. This reference distribution was then used to recompute the expected height distributions for incidence angles between 0 and 15°, using equations 3.3 and 3.5 or 3.15. Finally, the nine metrics were calculated using each of the distributions, and the expected bias at each angle was calculated as the difference from the reference distribution.

For comparison, we calculated the observed bias by extracting 150 000 rasters centred on (x, y) coordinates that were sampled twice in the overlap of adjacent flightlines. The observed bias was then calculated as the difference between the two values obtained for each metric (fig. 3.5).

3.6 Results

3.6.1 Effect of scan angle on the height distribution of points

As shown in Figure 3.6, the expected height distribution of points sampled at 30° includes more returns in the upper canopy than observed at-nadir, which implies more interactions with canopy elements. This pattern is reversed in the lower canopy due to the conservation of energy. The proximity of the green and the red lines shows that taking into account the reduction of points per beam has a relatively small effect on the expected height distribution of points. In contrast, the difference between the black line and the other two shows that the angular bias described by the probabilistic part of the model is comparatively large.

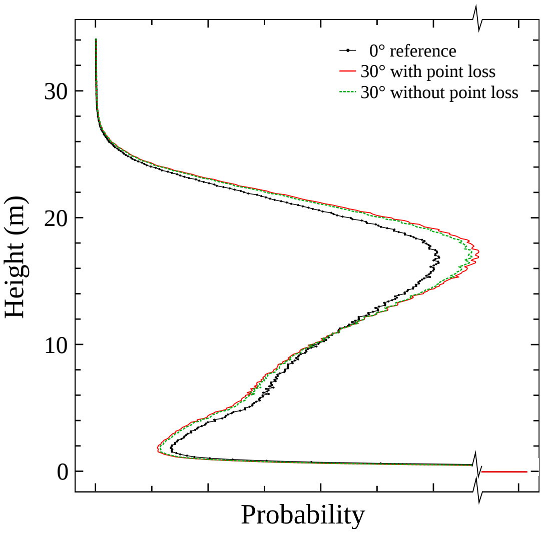


FIGURE 3.6 – The average height distribution of points sampled at-nadir (0°) and the expected height distribution of points sampled at 30°, with and without considering the fewer number of points per beam (calculated using equations 3.3 and 3.15, respectively). An angle of 30° was chosen to visualize the magnitude of the predicted angular bias. The histograms appear continuous because they were computed with 1 cm bins.

3.6.2 Comparison of observed and expected bias

Our model accurately reproduced the bias observed in the data (figure 3.7). Both the sign and magnitude of the bias were correctly predicted for each of the 9 metrics. Including the reduction in the number of points per beam only had a small effect on the predicted bias, but doing so brought the expected value closer to the observed value in every case. The observed bias varied considerably from one plot to the next, as shown by the whiskers on either side of the boxes. This is residual bias that is not captured by the model, because it was not implemented at the scale of a plot raster. As mentioned previously, it was implemented using the average height distribution of all the points in the dataset that were sampled at-nadir.

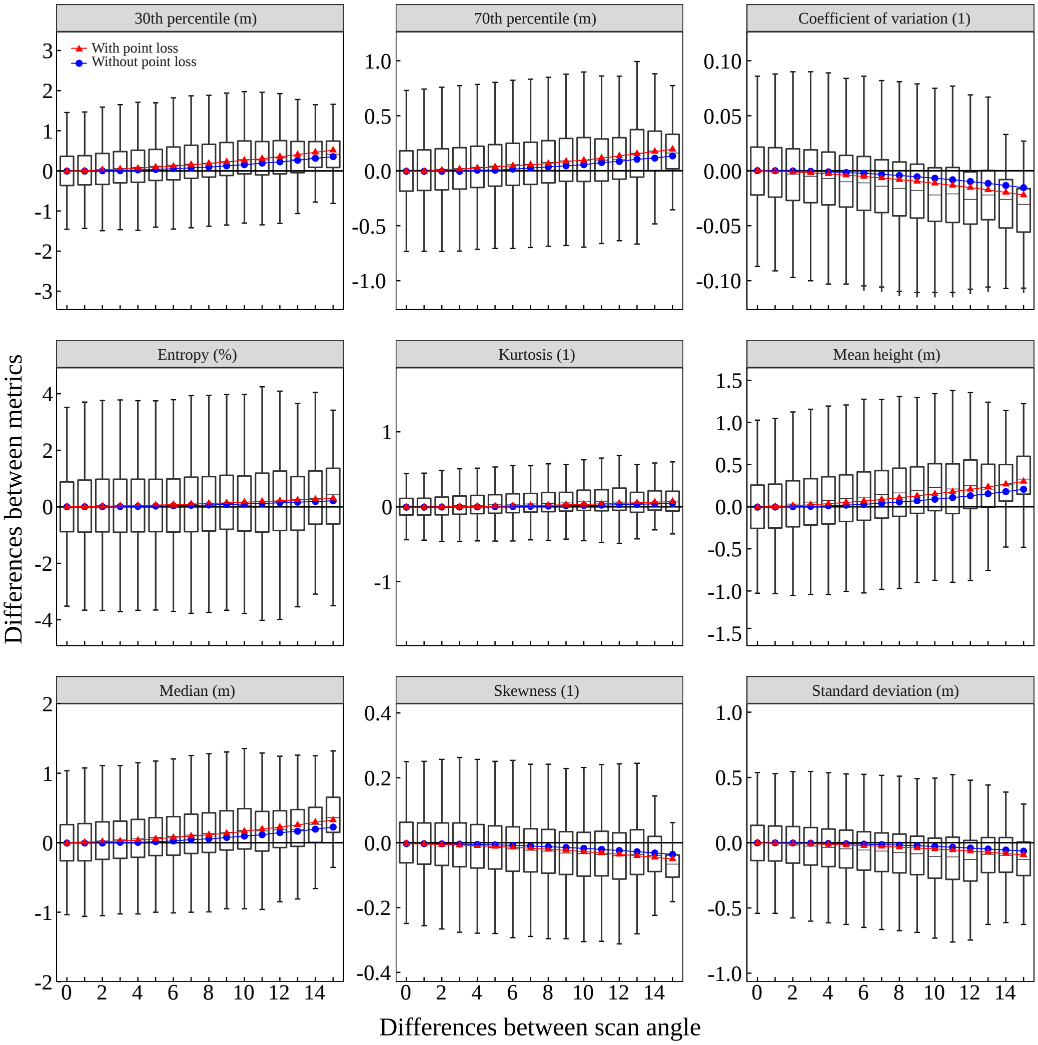


FIGURE 3.7 – Bias observed in the plot rasters that were sampled twice from different flightlines, as shown in (fig. 3.5). The observed bias (boxplots) of the nine LiDAR metrics is compared to the expected bias, both with (red points) and without (blue points) including the reduction in the number of points per beam (calculated using equations 3.3 and 3.15, respectively).

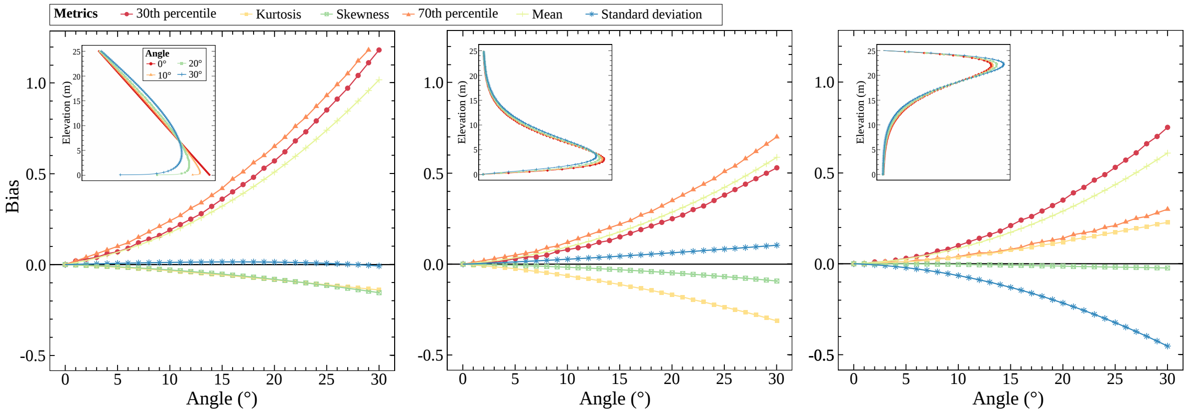


FIGURE 3.8 – Expected bias in three hypothetical stand structures with a maximum height of 25 m. The expected height distribution of points is plotted in the insets, which show (from left to right) a purely theoretical structure in which the number of returns increases linearly towards the ground, and two more realistic structures in which most returns occur in the lower and upper layers of the canopy. The units of the main *y* axis are either in meters or dimensionless (depending on the metric units).

3.6.3 The effect of stand structure on bias

While we were unable to quantify the effect of stand structure empirically, we did calculate the expected bias for three hypothetical stand structures, as shown in Figure 3.8. The stand structure on the left is unrealistic, but serves to show that the biases can be non monotonic with respect to scan angle. In this example the magnitude of the bias in the standard deviation is low, but increases until 15° and then decreases until 30°. Thus, despite its apparent simplicity, the model is able to predict complex patterns of bias.

The next two structures are more realistic as they are derived from modified gamma distributions that produce patterns similar to those observed in our data. Both are symmetric and represent typical stand structures in which most returns occur in the lower or upper canopy (but never on the ground). Comparing the patterns of bias in these two stand structures shows that the sign of the bias can switch from positive to negative (or vice versa), depending on the structure of the stand. The relative magnitude of bias can also switch : in the middle panel, for example, the 30th percentile is more biased than the 70th percentile, while the opposite is observed in the right-hand panel. The only constant pattern is the monotonic increase in bias for the height percentiles and the mean height, which was expected at the model development stage.

3.7 Discussion

3.7.1 Effect on forest resource inventory

We started from the hypothesis proposed by Goodwin *et al.* (2007) and Disney *et al.* (2010) that the increased travel distance through the canopy increases the probability a beam is reflected as incidence angle increase and we modelled that effect. Our model accurately reproduced the observed bias despite our simplifying assumption that canopy elements are randomly distributed in space. This likely reflects the fact that canopies are relatively continuous in northern hardwood forests. By enabling prediction of the bias attributable to incidence angle for various metrics, our model can be used to normalize LiDAR datasets acquired in forests that meet the assumptions made to build the model.

In our study the average overestimation of the mean height reached 40 cm at 15°. Bias of this magnitude could conceivably affect the accuracy of area-based models that use LiDAR metrics to predict other stand-level variables of interest, such as stand height, wood volume or aboveground biomass. However, we cannot make general statements about the influence of bias on the accuracy of area-based models because it is highly dependent on which LiDAR statistics are used. The bias of some metrics may compensate for one another, which could explain the absence of significant effects in the study of Næsset (1997), while bias may be cumulative in other cases. Our study also showed that the effects of incidence angle depend on the forest structure, which implies that the practical importance of the phenomenon described in our study is very site specific.

An important term in the model is the inverse of the cosine, which renders bias nonlinear with respect to incidence angle, indicating that there is a threshold angle beyond which the effect becomes extremely strong. For metrics expressed in meters in our analysis, differences began to reach values larger than one meter at 30°. However, caution must be used in extrapolating to larger angles because of the Taylor expansion that was used to linearize the expression. As the cosine tends towards 0 when θ tends towards 90°, the term $\frac{\Delta z \lambda_k(\theta)}{cos(\theta)}$ tends towards an undetermined limit. Therefore, the Taylor expansion cannot be used for any combination of variables that make $\frac{\Delta z \lambda_k}{cos(\theta)}$ depart too far from 0, in which case the equation is no longer amenable to analysis.

In reality, bias would not reach the maximum value associated with the largest incidence angle due to the overlap between flightlines. Approximately 30% of our area was surveyed in two flightlines, which implies that plots were rarely sampled with a single, large incidence angle. In our dataset, a plot sampled at the maximum angle of 15° in one flightline was likely to have been scanned at a lower angle of about 10°, for example, in a separate flightline. This should limit the effects of the incidence angle in practice. Having points from two amalgamated flightlines would alter their vertical distribution in a way that is a linear combination of the two effects weighted by the respective local point densities (see supplementary materials).

For future practical applications, we recommend that users first determine whether or not the effect of incidence angle can be neglected. This can be achieved by analysing each flightline separately and comparing the metrics obtained from different angles.

3.7.2 Linearity of the observed bias

We quantified the expected bias at a given angle as the difference between two metrics, one of which was calculated using a 0° reference distribution - the average height distribution of points sampled at-nadir (Figure 3.6). In contrast, the two metrics we used to quantify the observed bias were both calculated using rasters that may have been sampled obliquely (fig. 3.5). This implies that a incidence angle difference of 4°, for example, can originate from rasters sampled at 0 and 4°, but also 1 and 5°, 8 and 12°, 10 and 14° etc. The same applies to all incidence angle differences except 15°, which was necessarily the result of a raster sampled at-nadir and another one sampled at 15° because the maximum incidence angle was 15°.

Because our model always used 0° as a reference, our comparison of the expected and observed bias (figure 3.7) is only valid if the effects are linear. Figure 3.7 showed this was not strictly true, but within the range of our observations the angles are small enough, and the effects linear enough, that we consider the comparison to be valid. Using all the rasters sampled in two flightlines was necessary to obtain enough data to highlight the overall pattern hidden in the noise. If we had only used pairs that included one raster sampled at-nadir, the dataset of observed variation would have decreased from 150 000 to only 600 rasters, which was insufficient to show the signal.

3.7.3 Accounting for the reduction of number of returns per beam

Despite the limited magnitude of this effect on the expected bias, the data clearly showed that the number of points per pulse decreases with increasing incidence angle. To our knowledge, this phenomenon has not yet been reported in the scientific literature. One hypothesis is that as incidence angle increases, a pulse is more likely to be intercepted by vertical tree trunks because they become more exposed. Thus, an oblique beam is less likely to have subsequent returns. According to this hypothesis, the average number of points per beam can only decrease as a function of the incidence angle. Moreover the average intensity of first returns should be constant with respect to the incidence angle (if a range correction is applied).

Another hypothesis is that the probability of being absorbed by foliage increases with incidence angle. This can happen if the spatial distribution and orientation of leaves are not random. However, in contrast to the previous hypothesis, this could also lead to an increase in the number of pulses per beam in certain circumstances.

To appreciate why, consider the parameter λ_k that was first introduced in equation 3.6. λ_k is the product of the leaf area density μ, the G function and the clumping factor Ω (e.g. Nilson, 1971; Campbell et Norman, 1990). Thus, λ_k is not only a function of elevation (k) but also a function of incidence angle θ :

$$\lambda_k(\theta) = \mu_k G_k(\theta)\Omega_k(\theta) \tag{3.18}$$

However, we assumed that the spatial distribution of foliage is random, in which case $\Omega(\theta)$ equals unity and (given a spherical leaf angle distribution) $G(\theta) = 0.5$, for any incidence angle. λ was therefore only a function of the elevation (λ_k) in equation 3.6.

If these assumptions are relaxed equation 3.9 becomes :

$$\begin{aligned} i_k(\theta) &= \frac{\Delta z\, \lambda_k(\theta)}{cos(\theta)} \\ &= \frac{\Delta z\, \mu_k\, G(\theta)\, \Omega(\theta)}{cos(\theta)} \\ &= \frac{\Delta z\, \mu_k G(0)\, \Omega(0)\, G(\theta)\, \Omega(\theta)}{cos(\theta) G(0)\, \Omega(0)} \\ &= \frac{r_k(0)}{cos(\theta)} \frac{G(\theta)\, \Omega(\theta)}{G(0)\, \Omega(0)} \end{aligned} \tag{3.19}$$

Comparing the above to equation 3.14,

$$i_k(\theta) = \frac{r_k(0)}{cos(\theta)}(1+\alpha(\theta)) \tag{3.20}$$

we note that what is known as the extinction coefficient K –the G function multiplied by the clumping factor– is the function $1+\alpha$ in our model. The G function depicts the azimuthal angle distribution of the foliage. G can be an increasing or decreasing function of the view angle and K as well. Thus, contrary to the first hypothesis, this second hypothesis allows the bias to be positive or negative i.e. either an increasing or decreasing number of returns per beam.

Because both hypotheses lead to the same mathematical formulation i.e. a factor that multiplies the term $\frac{r_k}{\cos(\theta)}$, further analysis of the reduction in the number of points per beam in other datasets is required to distinguish between them. If decreases in number of returns per pulse are consistently observed at oblique incident angles, then it could be likely attributed to absorption by bark.

In our model, absorption was proportional to the density of canopy elements and the proportionality function (eq. 3.13) was constant with respect to height. This was sufficient to reproduce the behaviour of the data, but in reality it is likely that this coefficient varies between layers. A deeper inspection of the sequence of multiple returns would be necessary to refine our understanding of the reduction in the number of points per beam. In addition to evaluating how this reduction is distributed along the sequence of returns, it would be interesting to examine how it varies with height. For example, Næsset (2009) found that that single echoes tend to occur in the densest parts of the tree crowns. Since our model assumes the forest is *perceived* to be denser off-nadir, the results are therefore compatible.

Over and above all the considerations presented in this section, it must be highlighted that the part of the model accounting for the reduction in the number of returns per beam remains only an empirical add-on. It should not be considered as an intrinsic part of the model that was developed using a hypothesis-driven approach. Instead, we have used this model to propose one way to address the question of point loss. Other explanations can be proposed to explain this phenomenon, such as a loss of energy backscattered due to the increasing path length, for example.

3.7.4 Model applicability

An important limit of our study is that the model was only validated using a single dataset from a northern hardwood forest, and thus for a specific instrument and specific survey settings. Despite this, we believe the model has more general applicability over any type of forest that meets our initial assumptions.

Under our hypothesis-driven approach, our model was derived from a few initial assumptions made about the forest canopy structure and the way energy is spread and backscattered, independently of any site- or device-specific principles. The model is self-contained and does not rely on empirical data. This is the key to justify the empirical validation using a single dataset, which is only deemed to provide a demonstration that the model can be applicable in reality, in one forest type that meets our initial assumptions. The fact the model fits well with our validation dataset provides a good indication that it can offer a plausible representation of the physical reality.

We therefore expect a similar applicability in other forest types that meet the same assumptions. However, with the infinity of forest structures that can be found globally, it would not be possible to provide an exhaustive analysis of the limits of applicability of the model. For this reason, our approach was to provide 1) one example and 2) a source code that enables future users to determine if the model applies or not to their specific context.

Listing all types of forest canopies that could be adequately represented as a set of horizontal turbid layers is beyond the scope of this study. However, we believe the model may be applicable to any closed-canopy forests dominated by broadleaved trees. This includes temperate hardwood forests, but also to tropical humid or even dry tropical forests. Obviously, model applicability remains to be empirically demonstrated in other ecosystems.

3.7.5 Alternative approaches for discontinuous canopies

Our initial assumption that the forest canopy can be represented as a set of horizontal turbid layers would not be valid for forests with a clumped canopy structure such as conifer or savannah canopies. Coniferous forests, for example, exhibit hierarchical clumping structure at different levels (Wenge *et al.*, 1997), and individual conifers are more analogous to large, solid geometrical objects (Li et Strahler, 1985). This suggests that the influence of incidence angle is determined more by geometrical effects than by probabilistic effects.

Our model is analogous to radiative transfer (RT) models because it describes the probability of interacting with components of turbid homogeneous horizontal layers. Another way to model canopies is the geometrical optic (GO) approach, which was first developed for discontinuous conifer canopies that can be represented as an assemblage of three-dimensional, solid objects. The conceptual foundations of both the RT and GO approach were formalized decades ago (e.g. Li et Strahler, 1985; Strahler et Jupp, 1990). To develop a GO model of incidence angle effects discontinuous canopies, the first step would be to transfer existing GO equations to LiDAR applications, similarly to what we attempted in this study with the RT approach. A more advanced approach could even rely on equations that use both GO and RT principles (GORT approach) in the spirit of studies proposed by Wenge *et al.* (1997) or Haverd *et al.* (2012).

3.8 Conclusion

We examined the changes in the height distribution of returns that result from increasing the incidence angle, and hence both the distance a beam travels through the canopy and the probability the beam is reflected by the canopy. We developed a mathematical framework for understanding and predicting the resulting bias of LiDAR metrics, and demonstrated that our model accurately reproduced the bias calculated for northern hardwoods with relatively continuous canopies. The model allows a point distribution sampled at-nadir to be recomputed “as if it were sampled at another incidence angle”.

The model also suggests that the non-random spatial distribution of foliage may be responsible for fewer returns per beam at large incidence angles. Alternatively, this may reflect the fact that oblique pulses are more likely to be intercepted by vertical tree trunks, resulting in the end of a return sequence. Nevertheless, our model predicts the number of points per beam has a small effect on the height distribution of points, compared to increasing the length of the path a beam travels through the canopy.

3.9 Function f^{-1}, basic form

We try to demonstrate that the function f^{-1} can be written :

$$\forall k \in [\![1, n]\!], p_k = r_k \prod_{i=k+1}^{n+1} (1 - r_i)$$

Let’s consider n layers, with the ground layer being layer 1, and the highest layer the layer n. The probability to find a beam travelling thought the layer k is q_k. This beam travelling through layer k interacts with canopy components within that layer with a probability r_k. The probability to generate a point in the layer k is :

$$p_k = r_k \, q_k$$

The probability q_k is the probability that the beam passed through each previous layer without interacting with canopy components.

$$
\begin{aligned}
q_k &= \mathscr{P}(\overline{R}_{k+1} \cap \overline{R}_{k+2} \cap \ldots \cap \overline{R}_n) \\
&= (1 - r_{k+1}) \times (1 - r_{k+2}) \times \ldots \times (1 - r_n) \\
&= \prod_{i=k+1}^{n} (1 - r_i)
\end{aligned}
$$

Then,

$$p_k = r_k \prod_{i=k+1}^{n} (1 - r_i)$$

The case where $k = n$ is a particular case which does not follow the rule because the formula does not make sense :

$$p_n = r_n \prod_{i=n+1}^{n} (1 - r_i)$$

Adding a virtual layer $n + 1$ with a probability r_{n+1} of interaction of 0 solves the issue adding a neutral element into the product :

$$p_k = r_k \prod_{i=k+1}^{n+1} (1 - r_i)$$

Chapitre 4

Algorithmes et logiciels pour le traitement de données LiDAR

Le développement, ou, pour être plus juste, la validation des deux précédents modèles a été permise grâce au développement d'un logiciel dédié à la manipulation de données LiDAR qui s'est réalisé durant le temps de la thèse.

Pour respecter les critères de répétabilité évoqués en introduction, la recherche académique ne devrait être faite *que* grâce à des logiciels libres. En effet, en recherche, chaque étape du développement méthodologique devrait être parfaitement maîtrisé par au moins un membre du groupe de recherche. Ceci n'est *jamais* possible avec du logiciel non-libre aussi appelé logiciel privateur (de libertés [1]). En effet, le logiciel libre permet (a) d'étudier le code source afin de s'assurer du fonctionnement du logiciel et (b) éventuellement de modifier le logiciel pour qu'il s'ajuste à nos besoins particuliers. Le premier point est un pré-requis pour pouvoir analyser des données dans un contexte scientifique, le second point est une nécessité pour pouvoir analyser les données d'une façon nouvelle et non conventionnelle sans avoir à tout reprogrammer à partir de zéro.

Maîtrise du processus d'analyse et possibilité d'ajuster les outils à nos besoins ne sont pas permises par le logiciel privateur. Cependant la majorité des outils actuels sont privateurs et l'offre libre est (très) limitée. De ces deux points principaux naît très tôt dans la thèse la nécessité de développer du logiciel libre. Pour les besoins de la thèse au début, et très vite, devant l'intérêt grandissant de la communauté, pour la communauté. Le package R `lidR` développé pendant 3 ans, a été tout de suite plébiscité par la communauté, alors même que personne n'ait été mis au courant de son existence de façon directe ou indirecte. Seuls, les moteurs de recherche et le bouche à oreille ont permis une certaine notoriété du package, ce qui démontre, au delà des considérations évoquées plus haut, le besoin véritable d'un tel outil au sein de la communauté.

Devant cet engouement, le temps de développement devint de plus en plus important tout au long du doctorat, le nombre de rapports de *bugs* rapportés sur la plateforme d'hébergement du projet ont augmenté, les courriels de questions sur l'utilisation du package

[1] Entre autres la liberté de savoir ce qui est vraiment calculé par le logiciel.

arrivèrent sur une base hebdomadaire... Le package s'est fait connaître et est devenu la référence actuelle pour manipuler des données LiDAR dans R. Cette notoriété, toute relative, est assez importante pour que Nicholas Coops, le chercheur principal du réseau AWARE, propose d'en financer le développement, permettant ainsi d'embaucher une stagiaire. La croissance rapide de la popularité du package a aussi valu une invitation à une conférence internationale sur le logiciel libre. Enfin, elle a aussi mené vers le montage d'une collaboration internationale avec une équipe italienne sous leur propre initiative.

Le développement logiciel représentant la majorité du temps passé à travailler sur ce doctorat, la nécessité de valoriser ce travail s'est faite sentir. Plusieurs chercheurs, membres du groupe AWARE ou non, ont suggéré de publier un article de présentation du package. Cette option n'était au départ aucunement envisagée pour la simple raison qu'il ne s'agit pas de recherche académique, et qu'il apparaît extrêmement prétentieux de chercher à s'auto-promouvoir de cette façon. Si l'outil est bon il sera reconnu par la communauté sinon il disparaîtra avant même de naître. Tel était mon point de vue.

Un consensus a finalement été trouvé pour présenter le package tout en proposant un travail académique. L'article 3 conclut ainsi la thèse par une revue critique et technique de la littérature sur les algorithmes existant pour manipuler des données LiDAR. Cette revue présente le package `lidR` mis en contexte avec la littérature. Et si ce chapitre peut, à première vue, paraître déconnecté des questions de normalisation de la donnée LiDAR, il est en fait directement relié à cette question. En effet, si l'on souhaite traiter la donnée de façon standardisée, il faut des méthodes d'analyses qui ne soient pas spécifiques aux données, incluant, comme nous l'avons montré, un dispositif d'acquisition standard et des méthodes d'analyse théorique, mais aussi une chaîne de traitements algorithmiques claire et bien documentée. Et c'est ce dernier point qui est traité avec une attention particulière dans cette revue de littérature.

4.1 Résumé

Le LiDAR aéroporté est une technologie de télédétection qui est largement utilisée en foresterie et écologie pour suivre, prédire et cartographier des quantités d'intérêts reliées à la biomasse et la faune. La manipulation de données LiDAR, de par leur taille et leur structure complexe, requiert des algorithmes et des logiciels dédiés pour les implémenter. Aussi, les chercheurs ont souvent besoin d'outils pour développer et programmer leurs propres méthodes. Nous avons examiné et évalué de nombreux algorithmes actuellement disponibles et utilisés par les chercheurs et nous avons dressé une liste des logiciels qui les implémentent. En utilisant des exemples simples et des illustrations nous souhaitons sensibiliser la communauté au sujet de problèmes méthodologiques souvent rencontrés dans la littérature scientifique. Enfin nous présentons un programme open-source appelé `lidR` qui permet une manipulation facile des données LiDAR au sein du langage R et qui a été conçu en considérant les problèmes mis en évidence dans cette revue. Cet outil a été développé pour les communautés de recherche en sciences forestières et en écologie.

4.2 Abstract

Airborne LiDAR scanning (ALS) is a remote sensing technology that is widely used in forestry and ecology to monitor, predict and map numerous quantities of interest related to the biomass and wildlife. Manipulation of LiDAR data, due to their size and their structural complexity, requires algorithms and dedicated software to implement them. Also, researchers often need tools to develop and program their own methods. We reviewed and evaluated several algorithms currently available and used by the research community and listed software that currently implement them. With simple examples and illustrations, we raise awareness about methodological issues often found in the scientific literature. We finally present an open-source framework called `lidR` that allows a straightforward manipulation of LiDAR data within the R language, and that was designed in line with the issues highlights in this review. This tool was developed for benefit of the forestry and ecology research communities.

4.3 Introduction

LiDAR (Light Detection and Ranging) technology is currently revolutionizing data acquisition in the natural sciences and engineering. It has many applications in agriculture (e.g. Hämmerle et Höfle, 2014), forest planning (e.g. Bouvier *et al.*, 2015; Spriggs *et al.*, 2015), ecological assessment (e.g. Graf *et al.*, 2009), land surveying (e.g. Tompalski *et al.*, 2016), mapping, urban planning (e.g. Chen *et al.*, 2009; Yu *et al.*, 2010), and even car automation (e.g. Schnürmacher *et al.*, 2013; Liu et Deng, 2015). In the forestry sector, LiDAR has the potential to reduce the need for intensive ground-based inventory and stand structural assessment methods, making it a valuable tool for "wall-to-wall" forest inventory and mapping (e.g. Holmgren *et al.*, 2003a; Næsset, 2005; Van Leeuwen *et al.*, 2010; Vauhkonen *et al.*, 2014; Niemi et Vauhkonen, 2016).

Airborne laser scanning (ALS), using an aircraft-mounted sensor (Vauhkonen *et al.*, 2014), is an increasingly common application in remote sensing for characterizing the topography of large areas of the earth's surface using a cloud of georeferenced points. A single point records the height at which the emitted light was reflected back to the sensor with enough energy to generate a detectable "spike of intensity". Conceptually, this technology can be simply summarized as a way to produce a large quantity of multidimensional data. Inherently, these data contain mainly spatial and discrete information in three dimensions (x, y, z), but also an intensity for each point (a fourth dimension) and the position of each point in the sequence of returns from the same emitted pulse (a fifth dimension). LiDAR datasets also contain metadata both at the point level and at the project level.

4.3.1 Technical challenges of LiDAR data manipulations

There are many challenges in processing such "big data", due to the high quantity of data and the absence of inherent data structures, such as rasters.

Manipulating LiDAR data requires processes with advanced and complex computing algorithms while the sheer quantity of data involve computing resources that often exceed available processing memory (RAM). This implies a strong need of efficient and optimized techniques to process data within a reasonable timeframe. Although this can be achieved through the development of purpose-built software, a potential limitation is that writing efficient routines for the analysis of point cloud data requires technical computing skills.

In the fields of forestry and ecology, bespoke scripts are typically developed outside dedicated software environments by different research teams, or other users, to meet specific data processing needs, or to explore and develop new tools, methodologies or algorithms. These scripts are often written within programming environments such as `R`, `python`, `Matlab`, or other programming languages, depending on individual preferences. The proliferation of software and personalized scripts highlights the need to identify standardized methodologies and algorithms, with the intention of guiding the community towards a more mastered workflow.

4.3.2 The need for a literature review

Our initial assessment of the scientific literature quickly revealed that the methods used to process LiDAR data are often described inadequately, or sometimes not at all, particularly in cases where the algorithmic part of the workflow is not a major concern. We argue that this is a significant barrier to the continued development of LiDAR applications in forestry and ecology, originating mainly from (a) the widespread use of closed-source software that does not allow users to look "under the hood", and (b) a lack of knowledge or interest in the technical aspects of the workflow. The latter issue arises because many scientists and practitioners in these fields are, understandably, more concerned with finding answers to their research questions than with computational complexities.

A purpose of this review is therefore to provide a detailed and accessible assessment of the technical points related to LiDAR data manipulations.

4.3.3 Closed-source software and open-source philosophy

One point we wish to emphasize is the fundamental importance of the free and open-source software (FOSS) and open-format philosophies. The main software tools currently used for LiDAR data manipulation are usually closed-source and non-free (in the sense given by the Free Software Foundation (FSF) i.e. "freedom" not "free of charge").

There is an important political dimension to FOSS philosophy, but here we focus mainly on the technical aspects because they concern every research team, independently of any personal conviction. A fundamental problem associated with the use of closed-source software is that a "black box" is incorporated into the workflow process, so users are not able to study the underlying algorithms to obtain a fuller understanding of their own results. This is a particularly relevant for scientists in all research fields, because opaque methodologies cannot be critically examined during peer review processes. In forestry or ecology contexts, this often leads to uninformative methodological descriptions in published papers, such as *we used the X software to perform the task Y using an internal routine procedure* as an entire description of the process applied.

A purpose of this review is therefore to provide an argumentation in favor the open-source to convince scientific community that this point is not a point to take lightly.

4.3.4 The need for an R package

In forestry and ecology `R` is an extensively used language, but until now there has been no purpose-built package to manipulate LiDAR data in a convenient and efficient way. We developed an open source R framework called `lidR` to perform such tasks and the response of the scientific community has been encouraging. It was used by several research groups even before the first official release, and before we shared any information about the package. Actually the code was publicly available and accessible via search engines. This motivated the further development of the framework for the benefit of all users. The `lidR` package implements several algorithms that are described in the scientific literature and summarized in this review.

A purpose of this review is therefore to present this package and describe how it build in regard of the current state-of-the art.

4.4 Objectives, methods and structure of the review

4.4.1 Objectives

The three main objectives of this review are to : (1) provide a review of the literature focusing on algorithms and technical computing issues pertaining to airborne LiDAR applications in forestry and ecology ; (2) present pedagogical explanations of these issues that strike a balance between accessibility for non-experts and providing useful, in-depth information that remains relevant to more experienced users ; (3) present our open-source `lidR` package and provide an overview of the main algorithms it uses that were derived from the literature.

4.4.2 Methods

In reviewing the literature, we classified articles into two categories : (1) those that simply provided a list of the algorithms used in the study, and (2) those that described at least one algorithm in further detail.

For papers in the first category we focused on the methodologies employed, with the aim of obtaining an overview of the main methods currently used by researchers in our field. However, since the entire corpus related to LiDAR in forestry and ecology contains several thousand articles, it was not possible to produce a fully exhaustive review. Instead, we studied in detail papers that fell into the second category to obtain a deeper understanding of the presented algorithms. In this case we were limited by the numerous instances where new algorithms were presented, but with no implementation methods provided for any software, no source code and no sign of further implementation in subsequent studies. We therefore focused mainly on the algorithms that were available and commonly used.

The motivation for segregating the literature in such a way was to go beyond a simple review of the existing literature and include papers that described algorithms in detail. We actually implemented some of them and studied the source code of open-source software to compare the implementations currently available to the algorithms initially published.

4.4.3 Structure

This paper contains seven sections that present a review of the literature on a given topic, with each written from a different perspective. These changing viewpoints allowed us to avoid repetition when highlighting methodological issues in the use of ALS in forestry and ecological sciences. While we could have approached some of the topics in a given section from a different point of view, this choice of structure allowed us to meet our main objectives without producing an unreasonably long document.

Section 4.5 covers the topic of LiDAR data storage from an optimization standpoint. Here we emphasize the importance of some underlying computer science principles used in LiDAR data processing.

Section 4.6 covers the algorithms used for ground segmentation in ALS within the framework of the ‘free and open-source’ philosophy. By highlighting differences between the methods reported and those that are actually implemented, we aim to illustrate the importance of gaining a better understanding of the algorithms used in processing our own LiDAR data.

Section 4.7 covers the spatial interpolation methods used to compute a digital terrain model, which we look at from the point of view of the importance of producing clear and accurate descriptions in the 'Materials and Methods' sections of scientific papers. We showed that in most of the published articles we read, such methods are at best only partially explained.

Section 4.8 covers the topic of height normalization (subtraction of the terrain from raw LiDAR data), again from the point of view of the need for accurate methodological descriptions. In this case we demonstrated that the production of best-practice guidelines for classical routines would bring important benefits to our scientific field.

Section 4.9 covers the construction of digital canopy models from the point of view of a writing a classical review of existing methods. In this case no major issues were highlighted as those were already portrayed with stronger evidence in other sections.

Section 4.10 covers the use of metrics derived from the point cloud in the area-based approach. We approached this topic so as to emphasize the importance of relying on advanced, recognized methods, and thereby on the efforts wasted in attempting to re-invent them (often unknowingly).

Section 4.11 covers the topic of individual tree segmentation. In this case our point of view was to question the relevance of developing 'new' methods while existing ones are often not tested or used by the community.

Important methodological issues such as the discretization of the full waveform signal or the methods used to clean up outliers in the raw point cloud data are not covered in this review. In the first case, we omitted the topic mainly because it involves complex algorithms that we could not pretend to have sufficiently mastered. The second case can simply be explained by the strict absence of information on this topic in the scientific literature. We also avoided the methods used for several specific tasks, such as species recognition, snag detection or intensity normalization. There are two reasons for these omissions : (1) they are in most cases "in development" and not processes common to many analyses and (2) for pragmatic reasons related to manuscript length. Finally, statistical modelling techniques, data acquisition and hardware, or methods for assessing the accuracy of algorithms are beyond the scope of this review, which is strictly dedicated to the algorithms used to process data.

This paper concludes with section 4.12, which presents the `lidR` package that we continue to develop. We explain how it was designed in accordance with the content of this review i.e. to assemble a wide range of algorithms that represent the current state-of-the-art in LiDAR data processing. We also explain why `lidR` was mainly designed for research purposes, using `R`, which is unarguably the most widely-used software for performing analyses in the fields of forestry and ecology. Due to the rapid evolution of the package, this section may quickly become outdated, but our main objective of providing the community with a core of algorithms published in peer-reviewed journals to explore, test and take advantage of these methods, is expected to be applicable over the longer term.

4.5 Data storage

Reading, writing and storing data are the preliminary steps leading to any analysis. Although not part of the data analysis *per se*, these initial steps are a fundamentally important part of the workflow. In this section we present a short, technical, and didactic review explaining the advantages and drawbacks of existing storage methods that we believe could be useful to the wider LiDAR community.

In terms of functional requirements, the data formats have to provide a solution to (a) store information, (b) read all the data or only a sub-section corresponding to a user-defined geographical zone and (c) share the information with rest of the community. In terms of non-functional requirements, the data formats have to be :

open-document : With free access to its specifications, the community will be able to *store*, *read* and *share* the information.

fast : The format should allow access to data within an acceptable time-frame using personal computers.

efficient : Data storage should use as little memory as possible, taking into account the limitations of generally available computing resources.

Since discrete point clouds have an essentially tabular structure, with one point per row and one coordinate or scalar metadata per column (so-called “tidy” data (Wickham, 2014)), the most trivial open format would be plain text. However, while the plain text format meets our first requirements (i.e. open and readable by anybody), it fails to meet all the other requirements. In fact, using text files to store coordinate and scalar metadata is inefficient in terms of both size and speed of access (fig 4.1).

To understand why, we must consider the nature of a plain text file. As the name implies, this type of file contains only text. The most simple text format currently used in computing is ASCII, which stores every possible character using 7 bits. As a consequence, 7 bits are required in ASCII format to store any digit from 0 to 9. In contrast, 7 bits in usual binary representation allows storage of any number between 0 and 127 (i.e. 2^7 possibilities). For example, storing the number `1234567.89` in ASCII format requires 77 bits (63 bits for the 9 digits, 7 bits for the decimal place separator and 7 bits for the separator between successive numbers, usually a space or a coma), instead of 32 bits in its binary form. This simple example illustrates the inefficiency of using the ASCII format to store numeric values.

With regard to the speed of access to data, binary formats are much more efficient than plain text formats. This is a consequence of how computers read files and how the data are stored. Explained simply, a binary file can be described as a bit-by-bit copy of the internal representation of the data as hosted by a computer using a specific ordering pattern. Computers are inherently able to read numeric values as binary data and this operation is almost instantaneous. In contrast, interpreting a number, for instance `1234567.89`, stored in as text is more difficult. In this case, the computer has to read all characters, translate them all from ASCII into single digits and then interpret the global numeric value. This operation is complex, particularly with float values (decimal numbers), and therefore requires a lot of computing time (fig 4.1).

Because the ASCII format is not appropriate for storage, LiDAR data should be stored in a binary format. Such files can only be interpreted using dedicated software that knows the specific pattern of the bit-by-bit storage and that is able to internally analyse such information. The possibility for users to open, copy, modify, store and share a binary file relies on the fact that it is an open specification. This allows any user to access this type of file and, if necessary, to develop purpose-built software to analyse the data. Conversely, if the format is closed, users become entirely dependent on the owner of the format, who is then free to fully or partially provide, or even deny, access to dedicated software for using their binary file format, usually by issuing commercial licences.

For ALS data the standard file type is the `las` format. This binary format is standardized, and officially and publicly documented and maintained by the American Society for Photogrammetry & Remote Sensing (ASPRS, 2013). Unlike plain text format, it enables LiDAR data to be stored using only a minimum amount of memory in an optimized way. For example, point coordinates can be stored using only 32 bits. Thus, the `las` format provides a standardized way to store and share LiDAR data, which should be used by providers to deliver their data. This format comes in several versions, enabling users to store (or not) extra data or metadata, such as GPS time or an RGB component for each point. It is recommended that all ALS data users should read the official LAS specifications to understand exactly what a `las` file contains. For example, it is possible to store information about the extent of the data within the metadata of `las` files. By reading only the very beginning of the file contents, a user can thus get information on the spatial extent of the file. This allows for very efficient "cherry-picking" of specific regions of interest from among thousands of files, so users can select and read only the files appropriate to their analyses.

Regardless of all the optimizations embedded in the format, `las` files still require a large amount of memory. The requirement to store and share data across the Internet provided the impetus for improved data compression (Pradhan *et al.*, 2005; Mongus et Žalik, 2011)). Since there is currently no official standard to compress `las` files, several different schemes were developed over the last decade, such as "LizardTech LiDAR compressor" (LizardTech), "LAScompression" (Gemma lab) or "zlas" (ESRI). Each provider attempts to become the main reference in terms of file compression, and thus tend to keep their methods proprietary. The philosophy of closed format leads to files that cannot be shared because each format is dedicated to only one type of software, or is dependent on a license fee. However, since Martin Isenburg opened the `LASzip` library (Isenburg, 2013), this format has become the *de facto* standard, since it is free to use and can be freely implemented and supported by any software. The `LASzip` library compresses `las` files into `laz` binary files. It is based on a lossless compression method and files can be read seamlessly like `las` files, since the file can be uncompressed on-the-fly, i.e. as it is read. Moreover, it outperforms previous compression methods both in compression rate (fig 4.1) and compression/uncompression speed (Isenburg, 2013).

The open access and open document philosophy has led to the acceptance of `las` and laz files as mainstream formats, available for the benefit of the wider community, for storing and using LiDAR data. These formats meet all the requirements highlighted above and should be preferred to any other.

The disadvantages of the laz format are mainly the reading time. Indeed, compression implies uncompression, which also needs computation time (fig 4.1). In summary, both the las and laz file formats provide an open-source solution to support our three functional requirements (i.e. to store, read and share data). The las format is much more efficient in terms of allowing fast access to the data, while the laz format is a much more memory-efficient way of storing data.

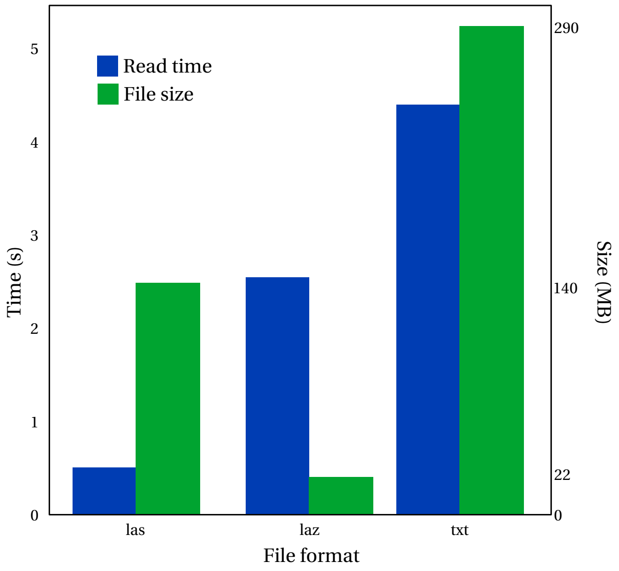


FIGURE 4.1 – Comparison, for a single dataset of 5 000 000 points, of read time and file size for different file formats. This simple example cannot be considered as a benchmark because performance depends on the actual contents of the file and the efficiency of the code used to read the file, among other factors. For example in the R language the function base::read.table reads the text file in 36 s instead of 5 s for data.table::fread. This graph only aims to illustrate the principles explained in this section.

Another solution for data storage is the use of database management systems (DBMS). Such systems can be described as software that provides services for storage, modification and retrieval for data. Data stored within a DBMS are accessible to programs installed on the same computer, or through networks. With these features, DBMS natively support our three functional requirements. Therefore, this storage mode can be considered suitable for ALS data. However, the drawbacks of this solution are that installing, configuring and maintaining DBMS requires skills in computing system administration. Data structures of DBSM are not easy understandable for non-IT users because they mostly require skills in relational models and SQL languages. Therefore this choice, while being technically suitable, is rarely encountered in current forestry and ecology applications.

4.6 Classification of ground points

From the computer science and algorithmic point of view, the segmentation of ground points is not only the first step towards generating a ground surface (Evans *et al.*, 2009; Zhao *et al.*, 2016), but also the most critical step of the workflow (Montealegre *et al.*, 2015b; Zhao *et al.*, 2016). Ground segmentation consists of classifying the point cloud into two categories : (a) the points that belong to the ground and (b) those backscattered by something else. Historically, this step was fundamentally important because ALS was first used for land topography purposes, before being recognized in the mid 70's as a potentially valuable tool for measuring the characteristics of the vegetation (Nelson, 2013).

Considering the large amount of data, this step necessarily has to rely on algorithms that automate the segmentation. Although these algorithms have been subject to decades of development, our review showed that current usage is generally hard to track and not completely mastered by end-users. For example, a common issue is that the segmentation of ground points is usually performed by the data provider (e.g. Næsset et Økland, 2002; Edson et Wing, 2011; Véga et Durrieu, 2011; Hamraz *et al.*, 2016; Roussel *et al.*, 2017), so that end-users either do not have access to or do not provide much information about the applied algorithm. In other cases users usually apply closed-source or undocumented proprietary routines to perform such a task (see section 4.6.2). This implies that the most critical step in LiDAR data processing is usually performed in a "black box" that end-users generally seem to trust, possibly out of habit.

This section aims to (1) review the algorithms and software currently used in the fields of forestry and ecology to perform the segmentation of ground points, and (2) propose ways to tidy up the workflow by improving the description of such methods. Consequently, this section, like others in our review, does not include a comparison of the performance of existing algorithms. Indeed, this task is complex and has been the subject of previous studies (e.g. Zhang et Whitman, 2005; Brovelli et Lucca, 2012; Montealegre *et al.*, 2015a). Our intent was to place our review upstream of such comparisons of algorithms by trying to highlight some of their methodological shortcomings and explain why we believe this step of the workflow deserves to be more accurately mastered by the community.

To achieve this, we reviewed the literature to identify the most commonly used algorithms and the software that propose them. We then dug into the source code and the documentation the software to explain how they actually work. We compared the original descriptions of the algorithms in peer-reviewed articles to their implementation in existing software. In doing so, we hope to raise awareness about the gap between what users report and what they have implemented in reality.

4.6.1 Progressive morphological filters

The Progressive morphological filter (PMF) described by Zhang *et al.* (2003) (700 citations according to Google Scholar), described in 2D in fig. 4.2, is based on a raster generated from the point cloud. The raster cell values correspond to the height value of the lowest point they contain (point-to-raster approach). On this grid surface called G_0 a morphological opening operation is performed using a square structuring element (see appendix 4.14 for detailed explanations) to create surface S_0. S_0 and G_0 are then compared cell by cell and all difference values greater than a set threshold t are removed from the initial grid G_0, which becomes G_1. This operation is repeated iteratively with an increasingly large structuring element and an increasing threshold to make S_1, then G_2 and so on. At the end of process, the lowest points of each remaining pixels are classified as "ground".

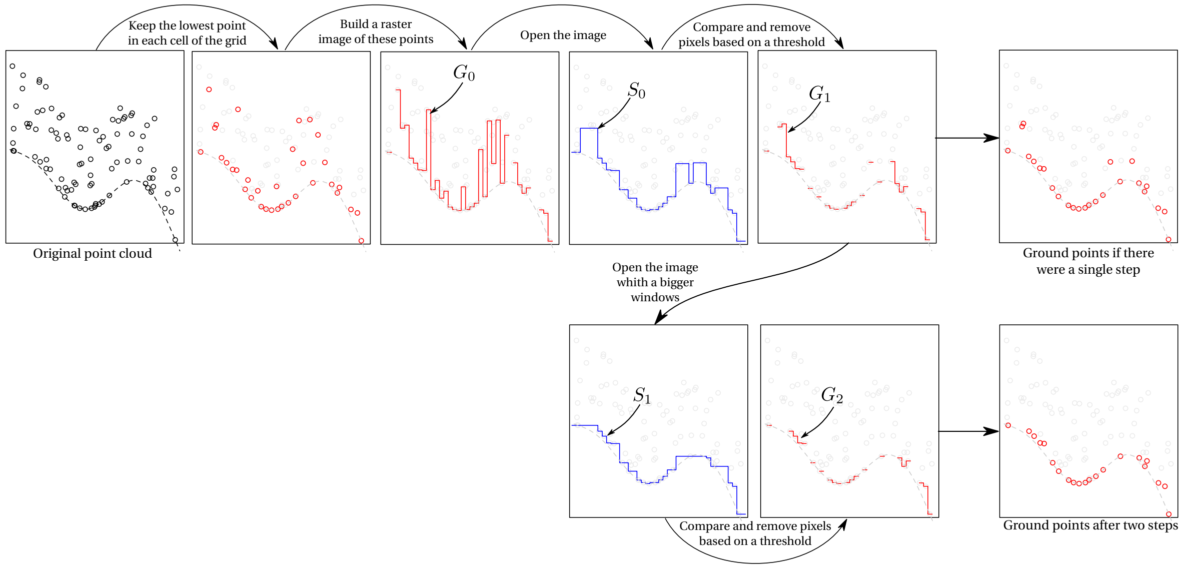

Keep the lowest point in each cell of the grid
Build a raster image of these points
Open the image
Compare and remove pixels based on a threshold
G_0
S_0
G_1
Original point cloud
Ground points if there were a single step
Open the image whith a bigger windows
S_1
G_2
Compare and remove pixels based on a threshold
Ground points after two steps

FIGURE 4.2 – Original Progressive Morphological Filter (PMF) as decribed by Zhang *et al.* (2003) and implemented in SPDlib (Bunting *et al.*, 2011, 2013). Drawn computationally by implementing a 2D version of the algorithm using randomly distributed points.

PMF is likely the most commonly known algorithm with several software implementations, such as in the Point Cloud Library (PCL) (Rusu et Cousins, 2011) and Point Data Abstraction Library (PDAL) (Butler *et al.*, 2016), two famous C++ open-source libraries for point cloud manipulation. It is also provided and recommended in the SPDlib software (Bunting *et al.*, 2011, 2013) and is also proposed in the Laser Information System (LIS) software (Laserdata GmbH, 2017).

Despite the number of implementations, we found only a few referenced uses of this algorithm in forestry or ecology contexts (e.g. Gonzalez *et al.*, 2010; Zhang *et al.*, 2009; Hunter *et al.*, 2013; Sumnall *et al.*, 2016). However, a degree of importance remains conferred on the algorithm by its several open-source implementations, which may explain why it is commonly referred to in methodological comparisons (Zhang et Whitman, 2005, e.g.). The open-source implementations allow computations to be made and compared to what was described in the original paper. This allowed us to illustrate a problem commonly encountered in the scientific literature, which can serve as a simple example of a fundamental issue that has far wider implications than the process of ground segmentation. By inspecting the source codes of PCL, PDAL and SPDlib, we found that the algorithms implemented do not correspond exactly to the original PMF described by Zhang.

The current implementation of the algorithm in SPDlib is very close to the original, with only two minor differences. The first difference relates to the parameters used. In the original article, the size w of the structuring elements used to open the raster is given by a choice of two formulas for each iteration k : $w_k = 2bk + 1$ or $w_k = 2b^{k-1} + 1$. In SPDlib the structuring elements increase following $w_k = 2(b + k) + 1$. The thresholds are also given by $t_k = s(w_k - w_{k-1})c + t_0$ in the original paper, while SPDlib implements $t_k = s(b + k)c + t_0$. s, b ans c being defined in the original paper.

The second difference is the absence of a hole filling procedure for empty pixels (as illustrated in fig. 4.2). Removing pixels that do not meet the set criteria leads to the presence of empty pixels in the raster, which can either be filled by interpolation or be left empty. The first choice was made by Zhang *et al.* (2003), while the second one was made by SPDlib developers.

Because the size of the structuring elements and the elevation difference thresholds are critical to achieve good results when applying the morphological filter method Zhang *et al.* (2003), even such minor changes to the algorithms might bring important effects to the results. Whether these changes represent improvements or not remains an unexplored question in the literature.

The current implementation of the algorithm in PCL and PDAL (described in 2D in fig. 4.3) is very different from the original because it is not based on a raster but on the raw point cloud directly. This choice of the developers was made possible by the fact that morphological operations can be applied either to a raster or a point cloud (see appendix 4.14 for further explanations).

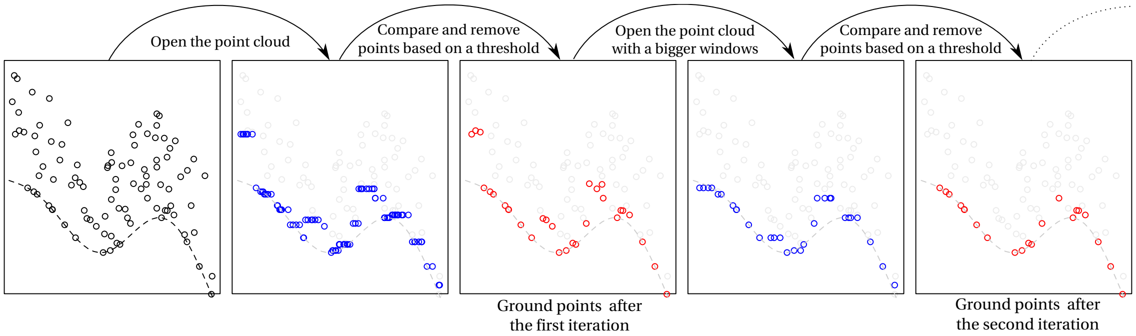

Open the point cloud
Compare and remove points based on a threshold
Open the point cloud with a bigger windows
Compare and remove points based on a threshold
Ground points after the first iteration
Ground points after the second iteration

FIGURE 4.3 – Progressive Morphological Filter (PMF) as implemented in PCL (Rusu et Cousins, 2011) and PDAL C++ libraries. Drawn computationally using the PMF implemented in the `lidR` R package.

This implementation is simpler than the original description because the absence of raster saves the need to define a supplementary parameter to choose its resolution. In addition, no information/point is lost during the point-to-raster process and the question of hole filling does not arise.

For `LIS` the only information we found in the documentation explicitly states that the algorithm does not correspond to the original one described by Zhang *et al.* (2003). Indeed, the documentation clearly states that the module "Filtering" has an *"adaptation of* *the filter proposed by Zhang (2003)"*. However, the software being closed-source, it is not possible to describe the algorithm.

Implementations of published algorithms in software rarely correspond to the exact transcription of what can be found in the original peer-reviewed article. Despite the numerous references to Zhang's PMF algorithm in the literature, the reality of current practice is that, without knowing, most users do not use the original algorithm. This is indicative of a general lack of rigour in the scientific literature when it comes to describing the methods used. We argue that providing a reference to a peer-reviewed paper describing the algorithm is insufficient. At the very least, the software used to perform the algorithm should be specified, and ideally the software should be open-source. As a scientific community we have to make sure that our methods are accurately reported, and therefore reproducible. This is especially true in cases where it is impossible to verify the coding of the algorithm, as shown in the next section.

4.6.2 Progressive TIN Densification

Progressive TIN Densification (PTD) (Axelsson, 2000) (1140 citations according to Google Scholar) is based on triangular irregular networks (TIN), which are described in 2D in fig. 4.4. The first step consists of a rough classification of ground points, which is achieved by keeping the lowest point from each large cells of a raster. The raster cells are defined to be larger than trees, or buildings (e.g. 50 m), for example, to ensure that the lowest points really belong to the ground (the probability of misclassification approaching 0). These rough ground points are then triangulated and for each additional point in the cloud the algorithm computes the angles θ between the 3 ground points of the triangle over which a point is located, as well as the distance d to the plane defined by these three points. A threshold is applied and if both $d < d_{max}$ and $\theta < \theta_{max}$, the points are classified as ground. Once this densification step is done, the process is reiterated with the newly generated ground points until convergence is reached (i.e. further iterations will not generate any new ground points).

The PDT was the most cited ground segmentation algorithm in our literature review (e.g. Liang *et al.*, 2007; Véga et Durrieu, 2011; Montaghi, 2013; Uysal et Polat, 2014; Bouvier *et al.*, 2015; Niemi et Vauhkonen, 2016). The popularity of the algorithm comes not only from its robustness, but is also likely related to its integration in the commercial software TerraScan (Lin et Zhang, 2014; Zhao *et al.*, 2016). Several joint references to this software and this algorithm can be found in the literature. For example, Donoghue *et al.* (2007); Liang *et al.* (2007); Hyyppä *et al.* (2008); Ioki *et al.* (2009); Van Leeuwen *et al.* (2010); Véga et Durrieu (2011); Watt *et al.* (2013); Mora *et al.* (2013); Montaghi (2013); Uysal et Polat (2014); Ahmed *et al.* (2015); Niemi et Vauhkonen (2016) stated that they used TerraScan and they explained that the algorithm used "under the hood" is Axelsson's PDT. Several authors also referred to TerraScan but without providing any information on the algorithm (e.g. Yu *et al.*, 2004; Chasmer *et al.*, 2006b; Li *et al.*, 2012; Racine *et al.*, 2014; Hamraz *et al.*, 2016).

TerraScan being a proprietary software with a closed-source code, it is impossible to

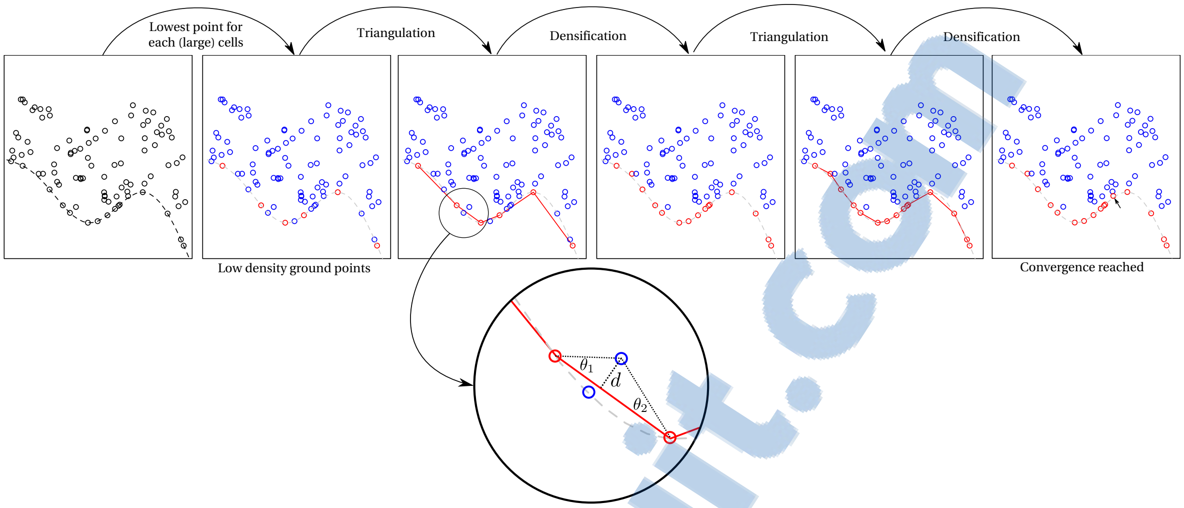


FIGURE 4.4 – Progressive TIN Densification (PDT) as explained in Axelsson (2000). The illustration was drawn computationally by implementing a 2D version of the algorithm. The triangulation is therefore represented as a linear interpolation of two consecutive points and only two angles θ are represented instead of three.

ensure that it actually uses Axelsson's algorithm to perform ground segmentation. Some authors refer directly to the official documentation of TerraScan (Soininen, 2016), but in reality the documentation does not state anything about the use of Axelsson's method. Instead, the documentation broadly describes an algorithm likely using a similar approach to Axelsson's. In contrast, Van Leeuwen *et al.* (2010) stated that TerraScan uses an iterative algorithm that combines filtering and thresholding methods from (Kraus et Pfeifer, 1998; Axelsson, 1999), thereby adding confusion in our assessment.

Hill *et al.* (2017) reported the use of Axelsson's algorithm within the LAStools software. Again LAStools is a closed-source software suite, so there is no way to ensure the correctness of this statement. According to Isenburg, the author of LAStools, the algorithm implemented in LAStools is inspired from the Axelsson's algorithm : *"*`lasground` *uses a variation of the Axelsson 2000 TIN refinement algorithm [...]"* (Isenburg, 2015).

We did not find other references to the use of Axelsson's algorithm in other software. A tutorial from Laserdata GmbH `https://fr.slideshare.net/FredericPetriniMonte/tutorial-ground-classification` let us speculate that LIS uses an hybrid approach based on PDT and a segmentation filter, but we did not find any other occurrence of this information in the documentation. The source code being unavailable, it is again impossible to verify.

The point we wish to emphasize here is our collective lack of capability as researchers to describe and explain accurately the most widely used ground segmentation method in the literature, a step of fundamental importance applied in almost all studies we reviewed. While readers can refer to (Axelsson, 2000) to get an understanding of the original algorithm, it is not possible to verify how it has been implemented.

To ensure reproducibility in a research context, it is important to make a clear distinction between the published algorithm and the software used, and obviously to provide both informations. This point was emphasized in section 4.6.1 and we re-emphasize it here. Issues with hard terrain have been reported for Axelsson's algorithm (Lin et Zhang, 2014; Zhao *et al.*, 2016) and some studies were dedicated to improve it (Zhao *et al.*, 2016, e.g.). But such studies compared the output of TerraScan to their own method or other methods (e.g. Pérez-García *et al.*, 2012; Brovelli et Lucca, 2012; Zhao *et al.*, 2016). Is the comparison meaningful ? For example, the TerraScan documentation does not state anything about 'mirroring points'. This feature, neither described in figure 4.4, nor in our short description, was included in the original paper to help deal with hard terrain and deep slopes. Is this feature actually implemented in closed-source software ? And if not, does it make sense to state that Axelsson's algorithm fails in hard terrain ? It would possibly be both inaccurate and unfair to attribute to the original author a shortcoming of another implementation written by a third party, especially when it is not possible to access it.

Our review led us to the conclusion that almost all reported uses of Axelsson's algorithm in the literature are actually incorrect. This may have broader repercussions because the PDT algorithm inspired several derivative methods such as Lin et Zhang (2014); Pérez-García *et al.* (2012); Zhao *et al.* (2016). We do not mean here to question the quality or the relevance of these studies, but instead highlight that currently accepted practice leads to an overall lack of rigour and consistence that may hinder our ability to truly compare existing methods. In this sense we believe there is a need for more rigour and accuracy in methodological descriptions, which in turn will lead to more reproducible and accurate science.

4.6.3 Hierarchical robust interpolation

Hierarchical robust interpolation (HRI) (Kraus et Pfeifer, 1998) (1250 citations according to Google Scholar) works iteratively. In the first step, a surface is computed with equal weights for all points. This surface runs as an average between all points. Ground points are more likely to have negative residuals, whereas vegetation points are more likely to have small negative or positive residuals. Points above the surface are given a small weight and those below the surface are given a larger weight. These weights p_i are computed from the residuals v_i of a function $f_{g,w}$ in which g and w represent two thresholds computed automatically using an adaptive process for each iteration. Then a new surface is fitted taking into account the weight using a linear interpolation function (Kraus et Mikhail, 1972) and the assigned weights. Points with large weights therefore "attract" the surface. This process is iterated until convergence, or until a given number of iterations have been completed. Upon completion, if a point is vertically above or below the surface within a predefined threshold, the point is classified as ground.

This algorithm has been implemented by the original authors within the SCOP software following the method presented in the original paper. This software now seems to have been superseded by SCOP++, but the official SCOP++ webpage currently redirects users to the website of a private company (`https://geospatial.trimble.com/products-`

and-solutions/inpho from which we could not get further information. Again, investigating how ground points are computed lead us to a dead-end.

FUSION/LDV (McGaughey, 2015), a famous software dedicated to LiDAR data processing in a forest characterization context, use an algorithm based on this algorithm according to the documentation. In this case the documentation is clear and states that the filtering algorithm is *adapted* from Kraus et Pfeifer (1998). The documentation is also clear about what part is adapted : g and w are fixed parameters provided by the user instead of variables estimated internally and dynamically by another algorithm. Despite the similarities between the method this difference is fundamentally important because we have in one case 0 input and 2 variables dynamically computed for each iteration and in the other 2 fixed inputs.

Independently of the question of the relevance of the algorithm, the FUSION/LDV documentation was the clearest about which algorithm was used and what modifications were made to the original. Yet there is no source code to ensure this is what is actually computed under the hood.

Surprisingly, despite the facts that FUSION/LDV is an important software used by many teams in the forestry and ecology fields, and that (Kraus et Pfeifer, 1998) has been cited more than 1200 times, we were not able to find a single clear reference to the use of the HRI in the literature. The only explanation we can provide is – again – that descriptions of methods are often non-rigorous. It seems that users consider, wrongly in our opinion, that ground segmentation is not a fundamentally important part of their workflow.

4.6.4 Multiscale curvature classification

Multiscale curvature classification (MCC) Evans et Hudak (2007) (200 citations according to Google Scholar) was developed for conditions of high-biomass and structurally complex forests. MCC is relatively similar to HRI and detailing their differences goes beyond the scope of this review.

The point we wish to highlight is that despite this algorithm being much less cited than the others, we found several reported uses in forestry and ecology (e.g. Smith *et al.*, 2009; Montealegre *et al.*, 2015b; Boudreault *et al.*, 2015). Leiterer *et al.* (2015) also referred to *"an adaptive multi-scale filter based on that of Evans and Hudak"* without more precision neither on the method nor the software. It was also used in papers dedicated to comparisons between different algorithms (e.g. Tinkham *et al.*, 2012).

The open-source MCC-LIDAR software (Hudak *et al.*, 2013) implements a strict version of what is described in the original paper according to the source code. This is to be expected, since it was developed by (at least) one of the authors of the original paper, but this is not necessarily a proof.

Other implementations in other software may differ slightly from the original method. According to its documentation, GRASS GIS (GRASS Development Team, 2017) uses a modified version of the MCC using a bilinear spline interpolation with Tykhonov regularization instead of a thin-plate spline to construct the intermediate surface. As in FUSION/LDV, the documentation clearly states what is done and what are the differences from the original method (however we did not dig into the sources of this software). The MCC method is also proposed in `Spdlib` in addition to the PMF. According to the source code, it appears to be a strict implementation of the original method (although the complexity of the source code prevents us from being 100% certain that it is a strict implementation by only reading the code).

4.6.5 Other methods

There is a very large corpus of other methods that have been used for ground segmentation. For example Montealegre *et al.* (2015b) reported the use of the "maximum local slope" (MLS) (Vosselman, 2000) from the SAGA GIS software. Lee *et al.* (2010) used an adaptive multiscale filter developed by Kampa et Slatton (2004). Pirotti *et al.* (2013) used their own PMF and inverted the original strategy : the structuring element was progressively decreased instead of increased. Zhang et Whitman (2005) described an "elevation threshold with expanded windows" (ETEW), which is very close to the PMF. We also found an "iterative polynomial fitting" (IPF), but without being able to link it to any references. It is also possible to find many other proceeding papers describing various methods. However, we were not able to find a single implementation or source code of such methods in known and actively maintained software.

4.6.6 Conclusion

The very large corpus of ground segmentation methods yields both opportunities in terms of using algorithms adapted to a particular context and issues in terms of clarity and reproducibility. Our review showed that there are almost as many variations of an algorithm as there are software to implement them. Our objective here was not to question whether the actual implementation is better or worse than the original method. In fact, such an assessment would be very difficult to make because we are rarely able to ensure what is actually computed. This is attributable to the closed-source nature of a large proportion of the dedicated software and to a common lack of detail in the associated documentation.

We believe there is a need to improve the presentation of methods used for the initial data processing steps in LiDAR studies with forestry and ecology applications. The justification for such a change is that currently accepted practices often hinder our real capacity to reproduce studies published in the scientific literature. Indeed, rare were the cases in our review where both the algorithm and the software used to segment ground points were reported, and we did not find any recommendations on how to parametrize such software. Ideally, methods would be made reproducible through the use open-source algorithms, but we understand this is not always possible. We suggest that the minimum information that should be provided include :

— The software and algorithm used to perform each task.

— An acknowledgement of the fact that we cannot ensure which algorithm was used in closed-source situations. When relevant, explanations should suggest that a given algorithm *appears to be inspired* from <author>, but that it is not possible to assess the level of similarity.

— A clear indication of the values (x, y, z) used for each parameter (X, Y, Z).

These obviously do not represent the ideal situation, but it would be an important step forward compared to the currently common practice of stating '*'we used <algorithm> from <author>*", which is generally wrong.

We suspect that one plausible explanation for the lack of detail is that the ground segmentation step is often performed by the data provider rather than the researchers. Ideally, researchers should master all steps of the data processing workflow, but a self-reinforcing cycle currently seems to prevent any progress in this direction. The ground classification step provides a good example of the issue we are facing as researchers. Even in a dedicated study such as ours, finding the existing algorithms, the existing software, the actual implementations, the source code and the documentation was an arduous process. In the absence of guidelines, it is understandable that users find it difficult to chose an appropriate algorithm for a given context (or terrain). Even if users had a clear idea of the algorithm they wish to use, it will be very difficult to identify the software that provides it. The regular user is therefore likely to trust the data provider for this task, or simply to use the algorithm provided in their software of choice without further questions about its relevance.

4.7 Digital terrain model

Generating a digital terrain model (DTM) usually follows the classification of ground points as the second step of LiDAR data analyses. Put simply, a DTM can be described as an "image" of the ground. Over the past decades, methods to generate DTMs have been intensively studied and several algorithms have been proposed for various terrain situations (Chen *et al.*, 2017). DTMs are used for a variety of purposes in practice, such as determination of the catchment basins of water retention and stream flow, or the identification of drivable roads to access resources. It also enables users to normalize the point cloud i.e. subtract the local terrain from the elevation of points to allow a manipulation of point clouds as if they were acquired on a flat surface (see section 4.8).

The construction of a DTM is simply a spatial interpolation of the ground points (see section 4.6) at unsampled locations. The accuracy of the DTM is very important since errors in the DTM will result in errors of tree height estimation (Hyyppä *et al.*, 2008), or more generally in inaccuracies in the measurement of the relative height of any given point relatively to the ground.

There is a wide range of methods that can be used to make spatial interpolation of points, which result from decades of research in mathematics and algorithmic sciences. All such methods are applicable to ALS data but they vary in difficulty of use and they are not necessarily available in dedicated software. Mitas et Mitasova (1999); Chen *et al.* (2017) proposed two good reviews of the different possibilities.

For the same reasons we presented on the fundamental importance of having clear and precise knowledge of the algorithms used to perform the ground segmentation, users should also be aware of the methods used to construct DTMs. This step is often missing or unclear in the scientific papers we reviewed. In more than half of them we were not able to get any information about how the terrain was computed. We believe that the absence of explanations can often result from the authors not knowing themselves. For example Zhao *et al.* (2009, 2011); Edson et Wing (2011); Wing *et al.* (2015); Roussel *et al.* (2017, 2018) used a DTM delivered by the vendor and computed using a proprietary routine with no or vague description of the method. Kwak *et al.* (2010); Jung *et al.* (2011); Tompalski *et al.* (2016); Bouvier *et al.* (2015); Guerra-Hernández *et al.* (2016) did not described the method used to build their DTM. Hill *et al.* (2017) stated that *"The DTM was derived as follows"* and they described the ground segmentation process. Barnes *et al.* (2017) used a rasterized triangular irregular network of ground point but without providing neither the triangulation method nor the interpolation method. Pippuri *et al.* (2012); Hunter *et al.* (2013); Véga *et al.* (2016); Hill *et al.* (2017) used a Delaunay triangulation but also did not provide the interpolation method. Hyyppä *et al.* (2001) used their own built-in method based on a point-to-raster approach (they attributed the elevation of lowest point to each cell of the DTM) followed by an interpolation that *"uses the knowledge of nearby pixels"*. This lack of accuracy is very common in the literature but should be avoided in the interest of making reproducible science.

We had initially planned to draw an overview of the methods used in the fields of forestry and ecology, but this rapidly proved impractical. From the cases in which they were reported, it appeared that there were almost as many methods as papers. Clark *et al.* (2004); Luther *et al.* (2014) used an inverse distance weighted interpolation. Clark *et al.* (2004); Li *et al.* (2012); Zhang *et al.* (2009) used ordinary kriging. Kobler *et al.* (2007) used a newly developed method called repetitive interpolation (REIN). Yao *et al.* (2012) used a bilinear interpolation. Ruiz *et al.* (2014) used an iterative algorithm based on the work of Estornell *et al.* (2011), which itself relies on a modified version of the method presented by Clark *et al.* (2004). Yu *et al.* (2004); Anderson *et al.* (2006) took the mean value of the ground points within each grid cell of a raster and only empty cells were interpolated with real spatial interpolation methods. Pippuri *et al.* (2012); Watt *et al.* (2013); Hunter *et al.* (2013); Véga *et al.* (2016); Hill *et al.* (2017) used a triangulated irregular network (TIN) of the ground points and Watt *et al.* (2013) reported they interpolated the TIN with a linear interpolation. van Ewijk *et al.* (2011); Stereńczak *et al.* (2016) used the ANUDEM method (Hutchinson, 1993), which uses an iterative finite difference interpolation technique. Wang *et al.* (2008) used an active contour algorithm implemented by TreesVis (Weinacker *et al.*, 2004), a software for LIDAR data processing developed by the institute for remote sensing and landscape information systems in Germany.

The main problem is not the wide variety of methods, but the fact that we did not find a single paper stating why a particular algorithm was chosen among other possibilities. Moreover, several authors created their own method without relying on well-known, time-tested spatial interpolation methods. While this is not necessarily bad practice, we suggest such a choice should be backed up by (1) a clear justification of the need for a new algorithm, (2) a description of the method sufficiently clear to be reproduced and (3) a demonstration that the new method outperformed other well-known ones in a particular context.

Another problem is that we did not find a single paper stating the parameters used and how they were chosen. There were also very few papers in which the algorithm used was clear and non ambiguous. Many papers "use a DTM" without providing enough information on its computation. We will attempt to demonstrate here, based on the two most simple methods, how important a clear statement of both the method and the parameters used has non negligible importance.

4.7.1 Example with a triangular irregular network (TIN)

This method is based on triangular tessellation of the ground point data to derive a bivariate function for each triangle, which is then used to estimate the values at unsampled locations. A first source of variation in the DTM comes from the several ways in which such triangulation can be computed (see 4.15 for more details). Yet, as seen in figure 4.5, no interpolation has been performed at this stage. The ground points have been meshed but there is no new data at unsampled locations. Hence, a second source of variation in the DTM comes from the several interpolation options that can be applied. Stating that the DTM was computed using a TIN is therefore not sufficient. Two informations are missing : (a) which method was used to make the triangulation and (b) which method was used to make the interpolation.

Linear interpolation uses planar facets of each triangle to create the interpolation. Used with a Delaunay triangulation (see 4.15), this is the most simple and trivial solution because it involves no parameters. Indeed, the Delaunay triangulation is unique and the linear interpolation is parameter-free. The drawbacks of the method are that it creates a non-smooth DTM and that it cannot extrapolate the terrain outside the convex hull delimited by the ground points since there is no triangle facets outside the convex hull.

Non-linear functions use additional continuity conditions in first-order, or both first- and second-order derivatives, thus ensuring a smooth connection of triangles as well as the differentiability of the resulting surface. The drawback is that it involves a more complex parametrization of the algorithm with possibly several non-trivial parameters to choose. There are many ways to fit non-linear functions (e.g. Akima, 1978) and to chose parameters. Again, a clear description of the methods should be provided, as well as a description of the rationale behind the selection.

4.7.2 Example with invert distance weighting

Invert distance weighting (IDW) is one of the simplest and most readily available methods. It is based on an assumption that the value at an unsampled point can be approximated as a weighted average of values at points within a certain cut-off distance d, or from a given number k of closest neighbours (Mitas et Mitasova, 1999). Weights are usually in-

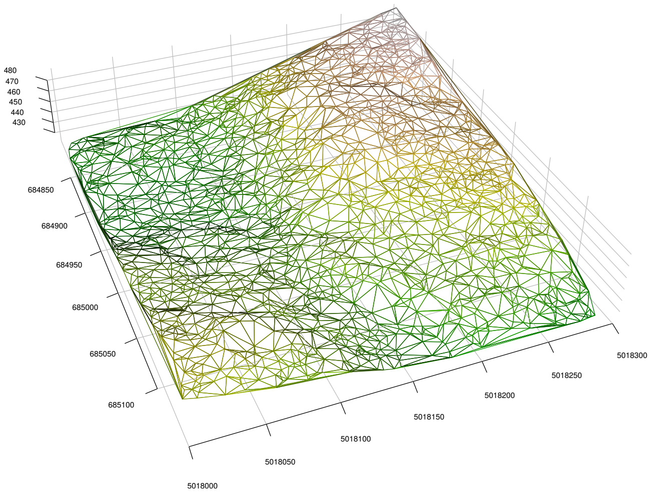


FIGURE 4.5 – Delaunay triangulation of ground point. The ground point are meshed in a unique way but there are still several manners to make the interpolation of the terrain.

versely proportional to a power p of the distance between the location and the neighbour, which leads to the computing of an estimator.

Therefore, the method can be summarized by the definition of two easily explainable parameters (i.e. k or d and p). While this basic method is easy to implement and available in almost any geographic information system (GIS), it has some well-known shortcomings that limit its practical applications. The method often does not reproduce the local shape evidenced by the data and it produces noticeable artefacts, such as local extrema at the location of the data points.

But more important, while a Delaunay triangulation with linear interpolation provides a unique DTM, IDW algorithms can return many different DTMs depending on the neighbourhood definition and the chosen power function p (see fig. 4.6). Thus, stating that the DTM was interpolated using an IDW without stating how the neighbourhood was defined and how the weights were computed is not much better than a complete absence of information.

4.7.3 Conclusion

Spatial interpolation methods is a vast field of statistics and mathematics already well documented and associated with many existing tools and resources. Our review revealed that spatial interpolations methods are used with an almost complete absence of consideration for (1) the relevance of a particular method compared to others and (2) the need to provide the parameters used for the sake of reproducibility.

This suggests that the generation of a DTM is generally perceived, wrongly in our opinion, as a necessary step of the workflow whose details are unimportant in studies using ALS to describe characteristics of the vegetation. The use of a particular algorithm seems to be more dictated by what is implemented in our favourite software than on a set of ar-

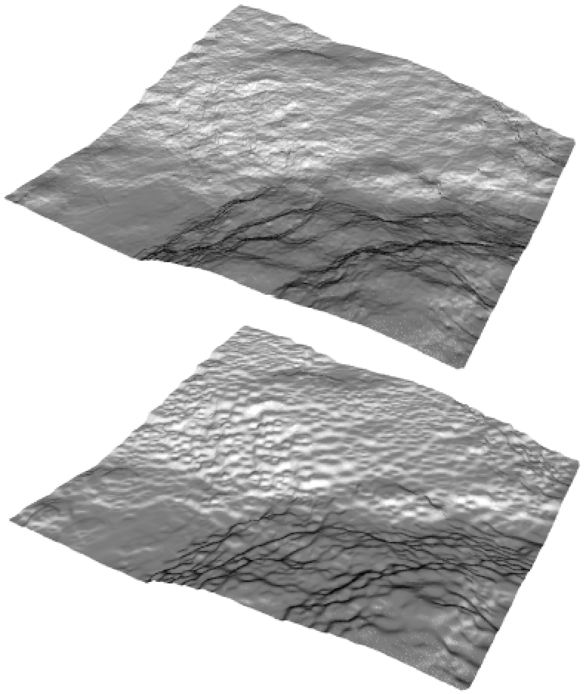


FIGURE 4.6 – Two DTMs computed with an IDW based on the exact same ground points but with different parameters k and p. The magnitude of the RMSE between these two DTM is 65 cm but there are both strictly valid in term of computation method. Thus providing the parameters used to compute the DTMs is extremely important.

guments relevant to the context of the study. For this situation to evolve, we believe users need to be more aware of the existing algorithms and take the habit of providing a complete description of their methodology. More attention should be given to describing substeps and to providing the parameters used to run the algorithms, when relevant. This can be achieved easily by replacing lengthy, often uninformative explanations by the name of a well-known algorithm, the name of the software used and the values of the parameters used as input. For common methods, TIN should come with the name of the algorithm used for the triangulation as well as the name of the interpolation method and its parameters if needed. IDW should come with the definition of the neighbourhood and the value of the power function used. Kriging should also come with the definition of the neighbourhood as well as the choice of the variogram. Finally, regardless of the method used, a justification of the choice of algorithm should always be included.

4.8 Data normalization

As the third step of classical LiDAR data analysis, normalization consists of subtracting the terrain from the point cloud (fig 4.7), thereby enabling its representation on perfectly flat ground. Assigning an elevation “0” to the ground has for advantage to simplify the further analysis of the point cloud over an area of interest. Two methods can be used to achieve this task : (a) using a rasterized representation of the terrain (DTM) (see section 4.7) and (b) interpolating between each point.

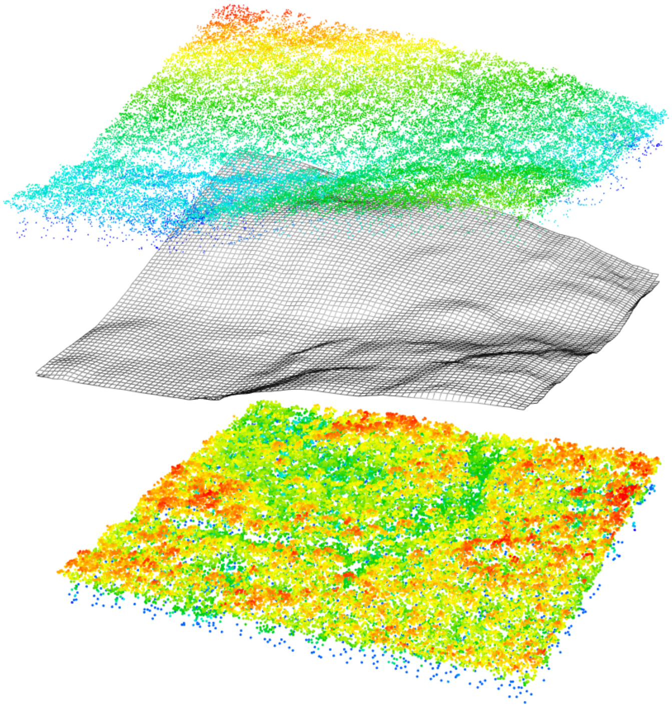


FIGURE 4.7 – Data normalization consists of subtracting the ground to get a reference at the elevation 0. Illustrated here using a Digital Terrain Model raster.

4.8.1 Raster-based normalization

The first and the most common way to normalize the point cloud is to subtract a raster DTM from all points. This method has been widely used (e.g. Wang *et al.*, 2008; Van Leeuwen *et al.*, 2010; van Ewijk *et al.*, 2011; Li *et al.*, 2012; Jakubowski *et al.*, 2013; Ruiz *et al.*, 2014; Racine *et al.*, 2014; Silva *et al.*, 2016) and is very simple and rapid to implement. For each point in the dataset, the algorithm simply has to find the value of the corresponding DTM pixel, and then subtract this value from the raw elevation value of the point. However, a significant drawback of this method is that it inherently leads to inaccuracies due to the discrete nature of the DTM. For this reason, the ground points used as reference (see section 4.6) are not individually normalized at 0. This is because the DTM was created and interpolated using regularly spaced points (see section 4.7), which do not match the actual location of the ground points in the dataset. Therefore, points computed as belonging to the ground are positioned at 0 plus or minus an error. A non negligible consequence of this inaccuracy is that a large number of points are located “under” the ground. This is illustrated in figure 4.8.

Such inaccuracies does not invalidate the method. In practice, the raster format remains the simplest way to store, visualize and share a DTM. However, using this method, we add inaccuracies that are not intrinsic limitations of the capacity to compute the terrain. As a consequence of using raster-based normalization, choices will need to be made regarding negative elevations. Should the negative ground points be removed, or be assigned a 0 value ? And similarly for positive ground points. Then, what about other negative vegetation points ? What does a negative elevation imply ? Obtaining suitable answers to these questions, and the prior choice of a normalization method, should ideally come along with an understanding of the consequences of the intrinsic inaccuracies of the DTM storage format.

According to our literature review, this method is the most commonly applied, and is generally the method used in geographical information system (GIS) software . However, we did not come across any scientific paper or best-practice guide providing suitable answers to the questions highlighted above, or at least raising such questions. Because several studies in forestry and ecology only use the points above 1.37 or 2 meters, the problem is usually invisible. Despite this, we suggest that good practices should involve a clear statement of what was done to deal with the inaccuracies of the raster-based normalization. After browsing through numerous open-source repositories of personal bespoke scripts on `github.com` during the last year, it became clear that enforcing a value to negative points is a common, yet undocumented practice.

4.8.2 Point-based normalization

The second normalization method is based on the interpolation of all points. In this case each ground point is interpolated at its exact position. The DTM is no longer structured as a raster, but as a point cloud that matches exactly the point cloud, which has for effect to remove any inaccuracies attributable to the representation of the terrain. This is illustrated in figure 4.7. The DTM has a virtually infinite resolution and the accuracy of the terrain is the *exact* result of the algorithms used to 1) classify the ground points and 2) to interpolate between them. Using this method, we ensure that every ground point used as reference is exactly normalized at 0, which considerably reduces the number of negative “dummy” points. Some points may still occur below 0 due to the inaccuracies of the interpolation method, the ground segmentation method, or even the inaccuracy of the sampling device (the question of outliers is beyond the scope of this section).

The LAStools software suite, which is extensively used in forestry and ecology, currently normalizes the point cloud this way. Despite this, we found only two studies explicitly stating that the point cloud was normalized using such an algorithm : García *et al.* (2010) used a spline interpolation method and (Khosravipour *et al.*, 2014) used LAStools.

Roussel *et al.* (2018) explicitly describes that the point cloud was normalized using this method, but this step was performed by the data provider. This raises an important question : why is it virtually impossible to find a reference to this method when one of the most important software used in the field normalizes the point cloud using this method ? A large part of the explanation likely lies on the fact that data normalization is generally not explained properly in the scientific literature. From our review, the vast majority of papers either make no mention of this step, or do so in a poor manner that leads to more confusion than information. In some cases the ground segmentation and terrain generation steps are even confused.

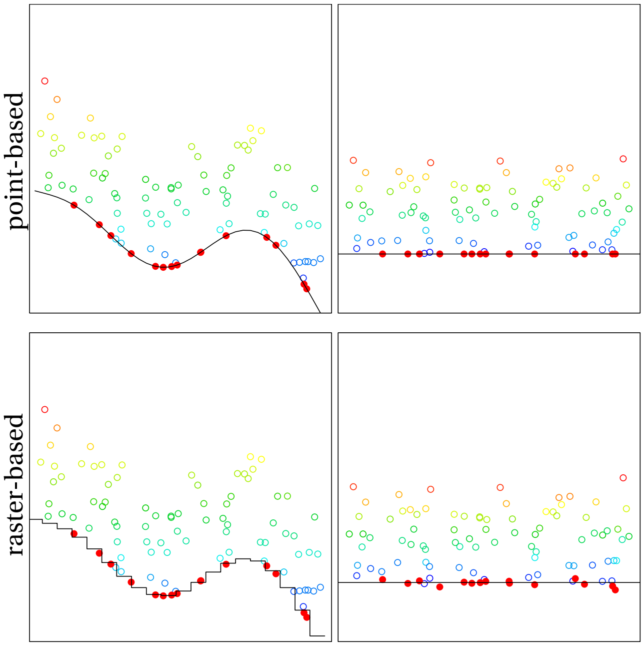


FIGURE 4.8 – The normalization of the data can be done using a raster-based representation of the terrain or using an interpolation passing exactly through each ground point. For both methods the limits of the accuracy depend on the algorithm used to segment ground points and on the algorithm and/or statistical methods used to make the interpolation. However, the raster-based representation adds a supplementary source of error attributable to the data storage format for the digital terrain model.

4.8.3 Limitations of data normalization

The normalization has a lot of advantages for the manipulation of the point cloud but it also involves some drawbacks. Normalization implies a distortion of the point cloud, and therefore of the sampled objects, such as trees, shrubs or even buildings. The problem is exacerbated in highly sloped terrain and for objects with large horizontal size dimensions (fig. 4.9). In this context, Vega *et al.* (2014); Khosravipour *et al.* (2015) manipulated the point cloud without normalization to preserve the geometry of the trees and Alexander *et al.* (2018) studied this effect and its consequences more closely. The key point being to preserve the location of the tree top.

4.9 Canopy Height Model and Digital Surface Model

The Canopy Height Model (CHM) is a digital surface fitted to the top of the canopy. It can be seen as the “canopy version” of the Digital Terrain Model (DTM). The CHM can be used for several purposes, including the segmentation of individual trees, which will

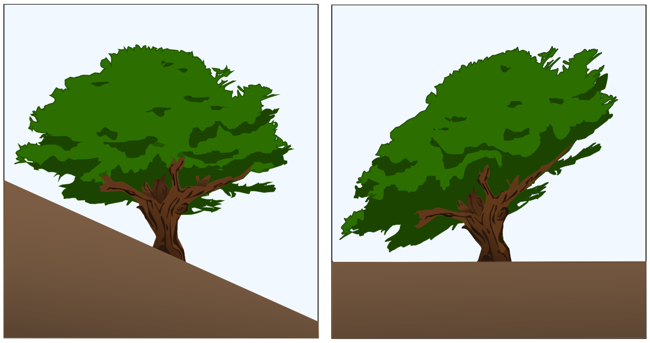


FIGURE 4.9 – Illustration of the effect of normalization on the geometry of objects such as trees located on sloppy terrain. The effect is exacerbated by the slope of the terrain and the horizontal dimensions of the object (inspired from Vega *et al.* (2014)).

be described in more detail in sections 4.10 & 4.11. This section focuses on the multiple algorithms that can be used to compute the CHM.

The term CHM refers to the normalized surface (Ruiz *et al.*, 2014; Popescu, 2007; Hilker *et al.*, 2010), while the term Digital Surface Model (DSM) refers to the non-normalized version of the same surface (Ruiz *et al.*, 2014; Zhao *et al.*, 2009). Several variants also appear in the literature, for example nDSM has also been used to refer to a normalized surface (Diedershagen *et al.*, 2004; Hyyppä *et al.*, 2008). Other notable variants include the term 'Canopy Surface Model' (CSM), (Véga et Durrieu, 2011), 'Digital Crown Model' (DCM) (Hyyppä et Inkinen, 1999) and 'Digital Canopy Model' (DCM) (Hirata, 2004). Further confusion arises from the fact that a DSM does not specify which surface it is referring to, while a CHM does not explicitly state that it refers to a surface. Because these diverse terms can be misleading, here we will use the term 'Digital Canopy Model' (DCM) following Clark *et al.* (2004), which is both consistent with the term 'Digital Terrain Model' (DTM) and, like DTM, is self-explanatory.

Although DCMs are widely used, descriptions in the literature of how they are computed are often weak, obscure or simply missing, for example in Hirata (2004); Kane *et al.* (2010); Diedershagen *et al.* (2004); Zhao *et al.* (2009); Ahmed *et al.* (2015); Hilker *et al.* (2010); Zhang et Liu (2013); Pascual *et al.* (2008). In some cases, the methods rely on proprietary, closed-source software, and thus explanations about computing methods are absent. For example, Pascual *et al.* (2008) stated : *"The raw data (x, y, and z coordinates) was processed into two digital elevation models by TopoSys using as interpolation algorithm a special local adaptive median filter developed by the data provider."* There is a clear "black-box" issue with such descriptions, whereby the lack of information clearly runs contrary to the basic scientific principles of reproducibility and replicability. In an effort to facilitate more detailed descriptions of DCM computation methodologies in future studies, this section will review commonly used methods and describe the currently documented algorithms.

The main algorithms used to create DCMs can be classified into two families (a) the point-to-raster algorithms and (b) the triangulation-based algorithms, with each family containing variations and “tweaks”.

4.9.1 Point-to-raster algorithm

Point-to-raster algorithms are conceptually simple, consisting of gridding the space at a given resolution and attributing to each pixel the elevation of the highest point within this pixel. The algorithmic implementations are trivial and fast in terms of computation time, which could explain why this method has been cited extensively in the literature (e.g. Hyyppä et Inkinen, 1999; Brandtberg *et al.*, 2003; Popescu, 2007; Liang *et al.*, 2007; Véga et Durrieu, 2011; Jing *et al.*, 2012; Yao *et al.*, 2012; Hunter *et al.*, 2013; Huang et Lian, 2015; Niemi et Vauhkonen, 2016; Dalponte et Coomes, 2016; Véga *et al.*, 2016; Roussel *et al.*, 2017; Alexander *et al.*, 2018). This is the default algorithm implemented in (according to the documentation of the closed-source software) FUSION/LDV, LAStools, ArcGIS (ArcGIS, 2016) with the argument that more complex interpolations are unnecessary for ArcGIS.

One drawback of the point-to-raster method is that some pixels can be empty if the grid resolution is too fine for the available point density. Some pixels may then fall within a location that does not contain any points (cf. fig 4.11a), and as a result the value is not defined. This implies a second step of post-processing to fill any gaps using an interpolation method (cf. fig 4.11b). It is at this step that methodologies often diverge, since in the absence of a standard method, different teams often use a range of methods, such as linear interpolation (Dalponte et Coomes, 2016), inverse distance weighting (Véga et Durrieu, 2011; Ruiz *et al.*, 2014; Véga *et al.*, 2016; Niemi et Vauhkonen, 2016) or any other more (or less) documented gap-filling methods. Our review of the literature revealed that information about these methods is often blurred and reduced to the word “interpolation” (e.g. Hyyppä et Inkinen, 1999; Zhao *et al.*, 2009; Popescu, 2007; Liang *et al.*, 2007). A careful inspection of the source code from Dalponte et Coomes (2016) showed a questionable method of iterative interpolation (interpolations of interpolations) that used the mean values of the non-empty neighbouring cells until all the gaps were filled. This method is also described in Brandtberg *et al.* (2003).

4.9.2 Triangulation-based algorithms

Triangulation-based algorithms interpolate the first returns using a triangulation (usually a Delaunay triangulation). Once triangulated, an interpolation within each triangle is used to compute the elevation value for each pixel of the raster.

In its simplest form, this method consists of a strict 2-D triangulation of the first returns. It is difficult to provide an exhaustive list of the studies that have used this method due to the general lack of detail provided in methodological descriptions. However, the method was used at least by Gaveau et Hill (2003); Barnes *et al.* (2017). We also assume that Zhao *et al.* (2009) used a closely related method, although the description is not detailed enough to truly determine whether the triangulation method or the point-to-raster approach was used.

Despite being more complex, an advantage of the triangulation approach is that it cannot leave empty pixels, regardless of the resolution of the output raster (i.e. the entire area is interpolated). However, like the point-to-raster method, it can lead to gaps and other noise in the surface when the number of pixels is abnormally low compared to neighbouring areas, and so-called "pits" attributable to first returns that penetrated deep into the canopy (Ben-Arie *et al.*, 2009) (easily identifiable in fig. 4.11b and 4.11c). Pits may make individual tree segmentation more difficult and change the texture of the canopy in a non realistic way. To avoid this issue the DCM is often smoothed, in an attempt to produce a more realistic surface with fewer pits and less noise (e.g. Brandtberg *et al.*, 2003; Barnes *et al.*, 2017; Jing *et al.*, 2012; Tao *et al.*, 2014). Again, since there is no standard smoothing method and standard routine so individual studies often use different techniques, often with little detailed information on the methodology. (Ben-Arie *et al.*, 2009) presented an interesting "pit-filling" algorithm for post-processing a DCM. We strongly suggest this should be the preferred smoothing method, since neighbouring pixels are used to fill the pits without the pits modifying the values of neighbouring pixels, as normally occurs with other methods.

More advanced algorithms have also been designed that avoid pits during the triangulation step instead of requiring a post-processing step. (Khosravipour *et al.*, 2014) proposed a 'pit-free' algorithm, which consists of a series of Delaunay triangulations made sequentially using points with values higher than a set of specified thresholds. For each threshold, the triangulation network is cleaned of triangles that are too wide, and is then rasterized. The triangulations and rasters are therefore considered to be "partial". In a final step, the partial rasters are stacked and only the highest pixels of each raster are retained (fig. 4.10). The output is a DCM that is natively free of pits without using any post-processing or correction methods. Since this algorithm is available, to our knowledge, only as part of the LAStools software, is more complex to implement and was developed only recently, there are relatively few documented occurrences of its usage. For example, Silva *et al.* (2016); Hill *et al.* (2017); Barnes *et al.* (2017) reported improved performance over other algorithms when it was used as the basis for individual tree segmentation.

A more recent study presented by Khosravipour *et al.* (2016) describes the development of a 'spike-free' algorithm, which uses all returns to build a TIN that ignores the points responsible for 'spikes' or 'gaps' in the meshing. The final production is therefore a Delaunay triangulation of selectively chosen points including first returns, but also some second and third returns. It is probably the most advanced documented algorithm currently available. However, we have not yet found any documented use of this method in the literature.

4.9.3 Minor variations and other approaches

Each of the algorithms described can be implemented with some minor variations to improve the output. To limit the number of empty pixels and pits, an improvement proposed by LAStools consist of replacing each LiDAR return with a small disk. Because the laser beam has a diameter (footprint) it makes sense to consider that it generates disks ins-

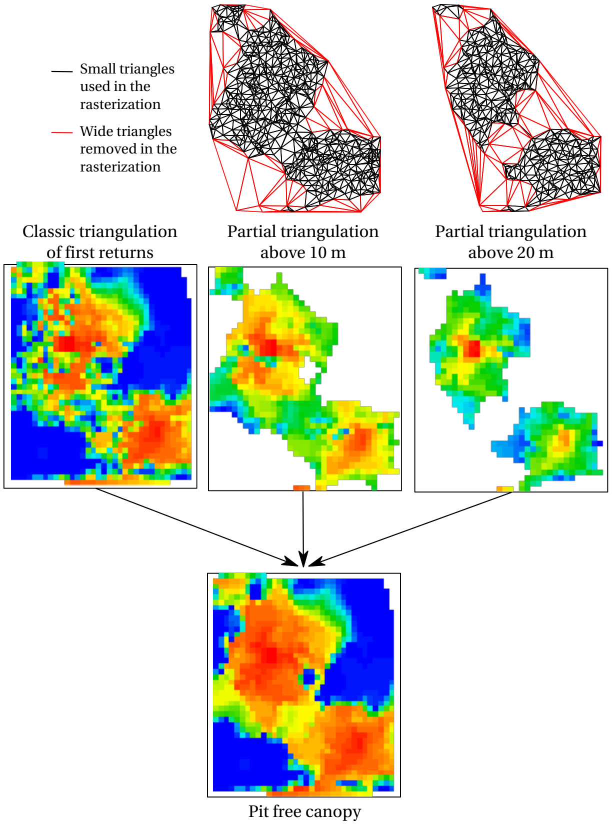

Small triangles used in the rasterization
Wide triangles removed in the rasterization
Classic triangulation of first returns
Partial triangulation above 10 m
Partial triangulation above 20 m
Pit free canopy

FIGURE 4.10 – Illustration of the “pit-free” algorithm using the basic triangulation stacked with two partial rasters at 10 and 20 meters.

tead of points with an area of zero. This “subcircling” adjustment effectively “densifies” the point cloud, and thus reduces the number of empty pixels or pits by naturally smoothing the DCM in a way that would not be possible using a post-processing operation. Such an adjustment can be applied independently of the algorithm used (fig. 4.11d). Our review of the literature did not enable us to retrieve any documented examples of adjustment, but from our discussions with practitioners we understand it is used in practice, at least in an operational context.

Since the DCM, in its native form, is basically a spatial interpolation of the first returns (the Delaunay triangulation followed by a linear interpolation being only one possibility among several others) any spatial interpolation method could be used to generate a DCM. For example Lloyd et Atkinson (2010) proposed a method based on kriging. However, even if we can obtain a result from a spatial interpolation method, it may not always be robust to pits and other noise without an internal mechanism to prevent them. Moreover, the increases in complexity, both at the computational and parameterization levels, makes a method such as kriging harder to implement but without obvious gains.

4.9.4 Conclusion

In summary, there are several methods available to compute a DCM, and these often have to be followed by a post-processing operation. The use of different algorithms will lead to various different DCMs, so the choice of algorithm strongly influences both the quality of the output and the accuracy of further analyses, such as tree segmentation. For these reasons we recommend that the choice of algorithm and any post-processing steps used should be clearly and accurately described in scientific papers. Referenced and documented algorithms should also be preferred over bespoke scripts, which are likely to partly "reinvent" existing methodologies.

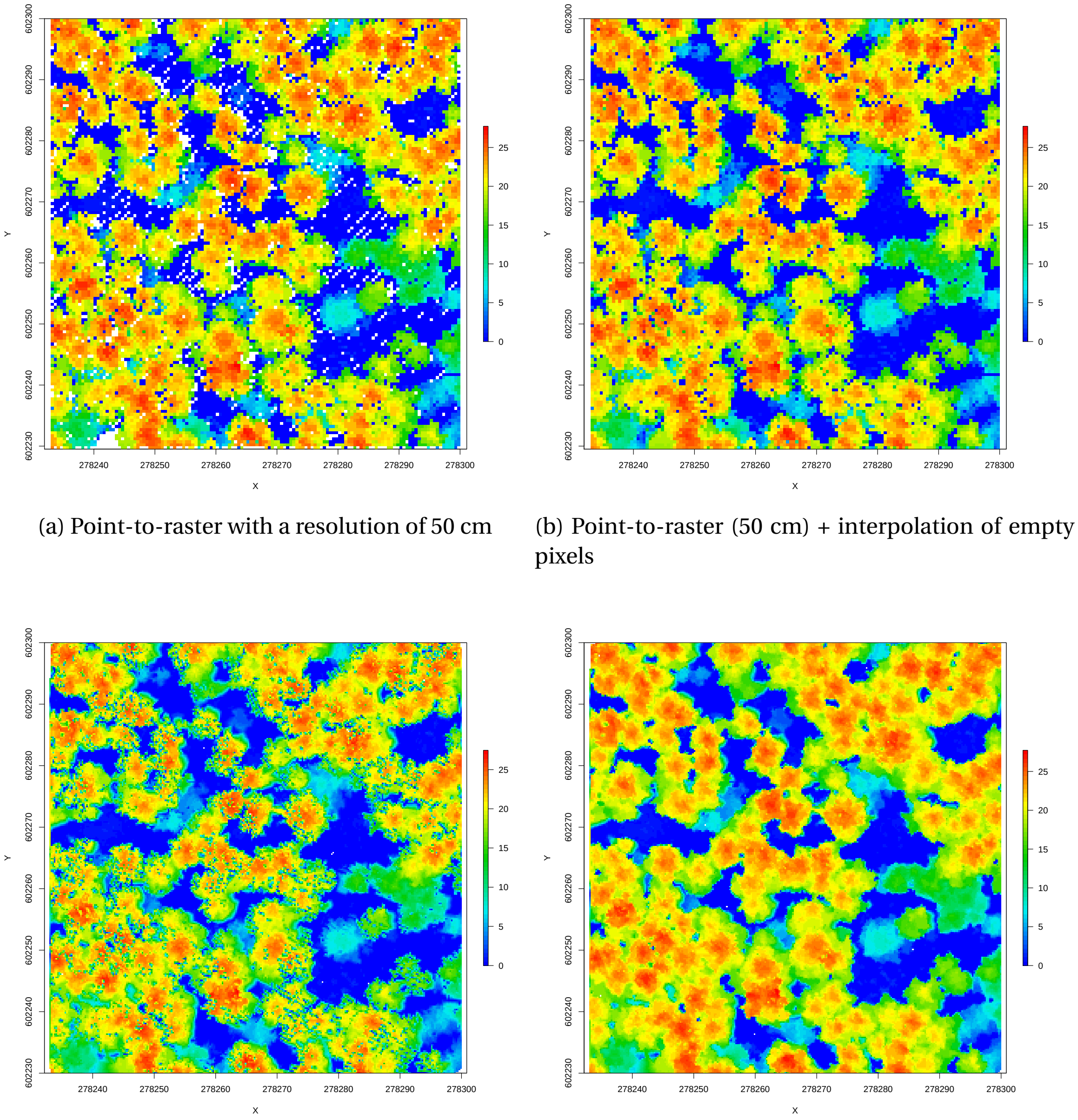
4.10 Developing the area-based approach

The area-based approach (ABA) is a widely used methodology to predict and map values of interest. It is conceptually simple and consists of computing scalars (so-called "derived metrics") that are summarized descriptors of the point cloud structure in a given region of interest (typically a 400 m^2 square or disc). These metrics can then be used as input to statistical models that link ground-based inventory to the structure of the point cloud. Predictions from the models can in turn be applied for each pixel to map a given a quantity of interest.

Derived metrics are usually computed from the z component of the first returns (i.e. height) (**?**) so the statistical models are usually based on a single dimension rather than the three (or even five considering that intensity – see section 4.10.2 – and the position in the return sequence could also be used) available. Therefore, such models use only 15 to 30% of the available data. Due to its relative simplicity, this approach is not associated with significant technical computing issues, so the following sections will focus on methods used to derive some less common metrics that exploit more than the single z dimension of the available data.

4.10.1 Using the three spatial dimensions

As presented in section 4.9, the digital canopy model (DCM) makes use of the xyz dimensions of the point cloud to construct an image of the canopy. Any classical statistical metrics can be derived from the z elevation of the DCM such as the mean height used in Ruiz *et al.* (2014); Asner et Mascaro (2014); Niemi et Vauhkonen (2016); Coomes *et al.* (2017) or any other classic statistic Ruiz *et al.* (2014). In addition, several other highly informative metrics can be derived that make use of the three spatial dimensions of the data. Parker *et al.* (2004); Kane *et al.* (2008, 2010); Luther *et al.* (2014); Blanchette *et al.* (2015) computed a metric called the "rumple index", which is a basically a measure of canopy roughness that can be used as an indicator of the forest successional stage. These studies computed DCMs using different methods but all computed a Delauney triangulation of the resulting raster. They then computed the rumple index as the ratio of the sum of the areas of all triangles to the projected area on the ground. The resulting value is a number between 1 (perfectly flat) and $+\infty$, and is an indicator of canopy structural complexity.



(a) Point-to-raster with a resolution of 50 cm

(b) Point-to-raster (50 cm) + interpolation of empty pixels

(c) Delaunay triangulation of first returns (25 cm)

(d) Khosravipour pit-free + subcircling with 15 cm radius

FIGURE 4.11 – Four DCMs computed from the same point cloud using different methods from each of the two main families of algorithms. (a) Contains empty pixels because of the absence of points in some pixels (The highest point cannot be defined everywhere ; (b) Empty pixels are filled by interpolation, but pits remain ; (c) The resolution was increased without empty pixels, but with many pits due to pulses that deeply penetrated the canopy before generating a first return ; (d) Pit-free with high resolution. The four examples where computed with the FOSS implementations provided by the `lidR` package

Despite its undeniable utility, the rumple index, as computed by these authors, has significant shortcomings because it involves a Delaunay triangulation of a set of points that is perfectly structured as a grid. The Delaunay triangulation is unique for most cases except when there are co-circular points (see 4.15 for more details). For a surface of $n \times n$ pixels there are 2^{n-1} possible Delaunay triangulations, and therefore 2^{n-1} different valid values of the rumple index using such method.

To our knowledge, Seidl *et al.* (2012) were the only authors to use a method based on an algorithm described by Jenness (2004). This algorithm provides a unique value of the surface area for a raster dataset. It is also computationally much faster, so we propose to use this method to compute the rumple index.

The problem may be insignificant in practice because the variability of the metric attributable to the algorithm itself is likely to remain rather small, as indicated by our simulation tests (fig. 4.12), especially when compared to the variability attributable to the choice of algorithm used to compute the canopy model. However, this should ideally not be used as an argument to create methods that can return several different outputs for the same single input. The point we wish to emphasise here is that when a good idea is found to express a value of ecological interest, it is likely that there already an existing and well recognized algorithm to make this computation. Creating a method from scratch will often result in poorer performance.

The rumple index can be seen as a texture index since there is no formal or complete definition of texture (Bharati *et al.*, 2004). Texture indices consist of a set of metrics calculated in an image, which are designed to quantify its perceived texture. Image texture provides information about the spatial arrangement of colours or intensities in a selected region of an image. It has several applications in various fields from video games to medicine, and can also be applied to LiDAR, especially to analyse the DCM. Ruiz *et al.* (2014) used what they called the *edgeness factor* (Sutton et Hall, 1972) as a derived metric from the DCM. Statistical textures were also used in (Pippuri *et al.*, 2012; Niemi et Vauhkonen, 2016).

4.10.2 Using intensity values

The intensity of the points can be considered as the fourth dimension of the point cloud. Several studies has demonstrated the potential of this dimension for which the values are affected by the forest structure (Moffiet *et al.*, 2005). It is rarely used in the ABA because it is a poorly mastered dimension. Indeed, its value is very sensitive to many parameter settings, such as flight altitude, scan angle, emitted energy (which in turn is dependent on emitted pulse frequency and device) (Moffiet *et al.*, 2005; Hyyppä *et al.*, 2008). Therefore, a model based on intensity is generally poorly transferable to larger surveys sampled by different providers using different devices, settings or methods. Even if intensity values have already been used in García *et al.* (2010), their standardization within (and eventually between) point clouds is required before this fourth dimension of the LiDAR data can be used efficiently in the ABA.

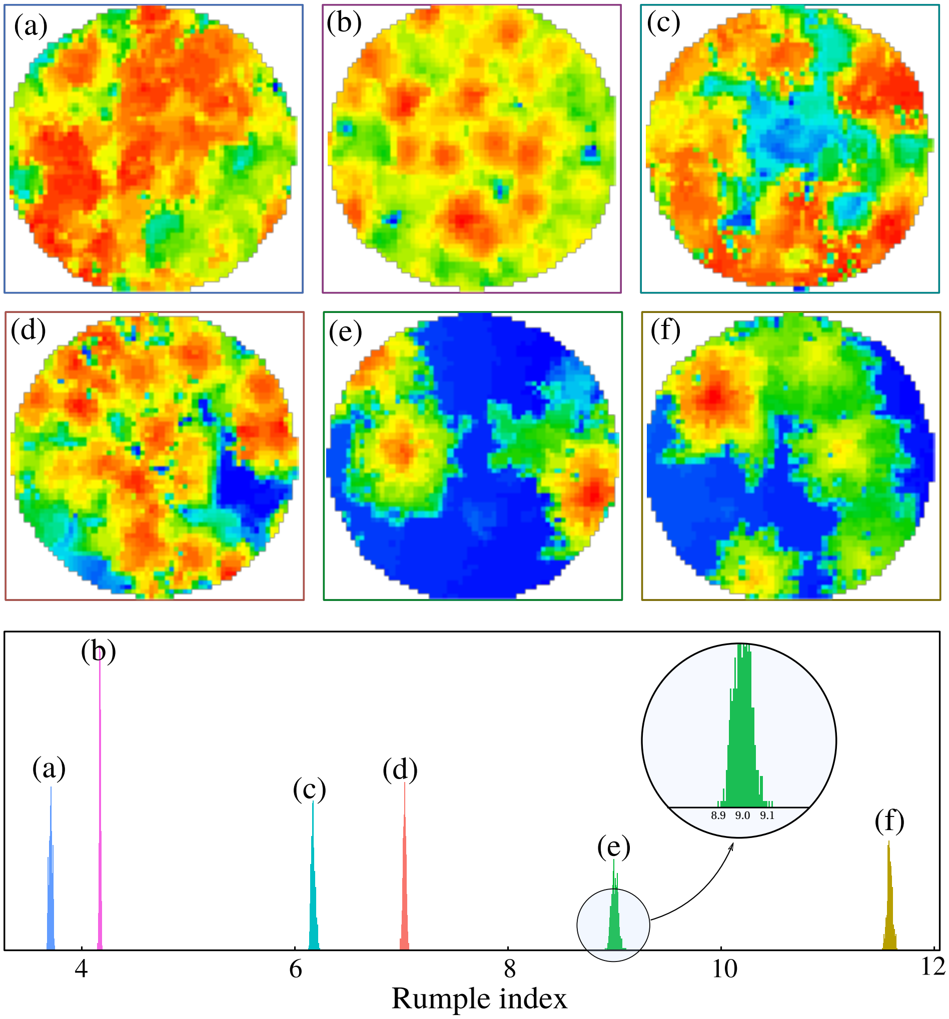


FIGURE 4.12 – Plots (a) to (f) show six digital canopy models computed for circular plots ordered by rumple index. The bottom plot shows histograms of the distributions of rumple indices derived from 500 different Delauney triangulations of the raster.

4.11 Individual tree segmentation

4.11.1 Main concept

Individual tree segmentation has several significant applications in forestry and ecology (Chen *et al.*, 2006; Koch *et al.*, 2006). The main idea is to accurately segment individual trees within the dataset and then extract a database of tree-level metrics. The derived metrics can be any descriptors of the point cloud distribution associated with one tree, or biometric descriptors of the trees such as crown diameter or height. Based on these attributes, and based on some prior knowledge on the tree growth or on allometric equations, it is possible to link the tree metrics to other meaningful values such as wood volume, biomass and species type (Hyyppä *et al.*, 2001; Popescu, 2007; Zhang *et al.*, 2009; Kwak *et al.*, 2010; Yao *et al.*, 2012; Gleason et Im, 2012).

The body of literature on individual tree segmentation is so considerable that it would not be possible to be fully exhaustive in this section. There is a wide range of methods proposed in literature but the range of routines that are actually implemented and available in dedicated software is quite narrow. Developing such routines is a field of research in itself, so there is no standard methods and some research teams work on their own bespoke algorithm and publish them as a "novel approach".

We can distinguish two categories of algorithms i.e. (a) algorithms based on a digital canopy model (DCM, see section 4.9) and (b) algorithms based on the raw point cloud. Another two-fold classification was proposed by (Hamraz *et al.*, 2016) who suggested to distinguish parametric and non-parametric algorithms. In the following sections we preferred the first typology for its simplicity.

4.11.2 DCM-based algorithms

DCM-based algorithms segment individual trees based on an image of the canopy. They are based on regular image processing algorithms that are not specific to point clouds. The choice of algorithm to build the DCM is therefore extremely important (see section 4.9). There are decades of development behind segmentation algorithms used for image processing and computer vision. Because these algorithms have been extensively documented, we describe only the most commonly referred to in the forestry and ecology literature. These belong to the "watershed" and "region growing" families of generic algorithms for image processing, which are not specifics to any kind of image in particular. For this reason, their use for tree segmentation usually implies some forest specific pre- and post-processing of the image given by the DCM.

Watershed

The watershed algorithm treats the image like a topographic map, with the brightness of each pixel representing its height and finds the lines that run along the tops of ridges. Inverting the DCM image lead to catchment basins at the location of each tree and the watershed appears naturally as an pertinent segmentation method. However, in practice, the watershed tend to give over-segmented results due to noise and/or other irregularities such as the differences in tree heights and natural variability of vegetation within tree crowns such branch (Hamraz *et al.*, 2016). To overcome this issue the DCM is usually smoothed in pre-processing (independently of the segmentation algorithm by the way (e.g. Koch *et al.*, 2006; Véga et Durrieu, 2011; Zhen *et al.*, 2013; Dalponte et Coomes, 2016; Silva *et al.*, 2016)).

To improve the segmentation a variation of the regular watershed called "marker-controlled watershed" is classically used to limit the number or regions to segment by specifying the objects of interest with markers. These markers are the tree tops and it implies a first step upstream of the segmentation to find the tree tops. This step can be achieved using a Local Maximum Filter (LMF, see section 4.11.4) algorithm to identify tree tops as markers.

So far, this algorithm have been extensively used in literature to segment trees (e.g.

Pyysalo et Hyyppä, 2002; Mei et Durrieu, 2004; Chen *et al.*, 2006; Kwak *et al.*, 2007; Reitberger *et al.*, 2008; Kwak *et al.*, 2010; Edson et Wing, 2011; Jing *et al.*, 2012; Tao *et al.*, 2014; Barnes *et al.*, 2017; Alexander *et al.*, 2018). Indeed the watershed algorithm is available in any good image processing software natively or using add-on and any programming language have one or more libraries enabling to perform a image segmentation based on the watershed segmentation.

Focusing on software dedicated to ALS data manipulation, we can cite FUSION/LDV (McGaughey, 2015), which according to the documentation uses the watershed segmentation method. Considering that this software is largely used in forestry and ecology, this algorithm is likely to remain widely used for some time. The R package `ForestTools` (Plowright, 2017) uses a marker-controlled watershed using a LMF to find tree tops. In an attempt to promote reproducible science we would have liked to cite more tools but in practice this task is made difficult because often the software and algorithms are not mentioned in scientific papers.

Region growing

This approach to segmentation examines neighbouring pixels of initial seed points and determines whether neighbour pixels should be added to the region based on a given set of constraints. For tree segmentation, local height maxima are used as seed points (figure 4.13).

The marker-controlled watershed is a specific case of region growing algorithms for which the constraint is based on the gradients in the image. Region growing is more generic because the region can be growth based on any criteria. For example, starting from local maxima, Zhen *et al.* (2013) used six conditions based on homogeneity, crown area and crown shape as criteria to stop region growth. Dalponte et Coomes (2016) used a percentage of the local maximum and a user-defined value representing a threshold difference between the local maximum and a given pixel to grow the regions. It is definitively impossible to list all the criteria used in the literature because they are almost as numerous as the number of publications, and descriptions can be unclear. Again, as there are no standard routines, research teams are likely to create their own set of constraints, which means there is an almost infinite number of potential variations of this algorithm.

There are also several ways to grow a region, so stating “growing region” alone provides a weak description. First, there is a choice to make on the connectivity of the structuring element such as 4-neighbours (e.g. Dalponte et Coomes, 2016) or 8-neighbours (Hyyppä *et al.*, 2001; Solberg *et al.*, 2006; Véga et Durrieu, 2011, e.g.) (fig. 4.13a and b) that may lead to different segmentation results. Second, there is a choice to make on the order of the segmentation, with at least two possibilities. Either the regions are all grown simultaneously, or sequentially (fig. 4.13a and c). In the latter case, if a top-to-bottom approach is chosen, the tallest trees impose their shape on the smaller ones.

The region growing algorithm family has been used in several studies (e.g. Hyyppä *et al.*, 2001; Popescu *et al.*, 2002; Solberg *et al.*, 2006; Koch *et al.*, 2006; Véga et Durrieu, 2011; Zhen *et al.*, 2013; Dalponte et Coomes, 2016; Barnes *et al.*, 2017), but is is generally difficult to find clear explanations of the method. We did not find obvious occurrences of material and methods sections stating that the growing region was run using a top-to-bottom approach, but it seems ecologically more pertinent to apply a top-to-bottom method to enable the tallest trees to impose their shape on smaller ones.

Focusing on software dedicated to tree segmentation, Hyyppä *et al.* (2001); Pyysalo et Hyyppä (2002); Maltamo *et al.* (2004) stated they used the commercial software "Arboreal Forest Inventory Tools of Arbonaut", although we did not find any trace of such software and therefore cannot explain what method is used internally. Our review shows that the software "TreeVaW" is more often used (e.g. Popescu, 2007; Popescu et Zhao, 2008; Zhao *et al.*, 2009; Popescu *et al.*, 2011; Zhang et Liu, 2013; Huang et Lian, 2015). According to Edson et Wing (2011) TreeVaW uses a local maximum algorithm coupled with a region growing method. However, we could not find how to download either the software or the source code (if open-source), and again we cannot provide more details about the method under the hood.

An interesting alternative method was proposed by Silva *et al.* (2016), which uses an open-source algorithm implemented in the `R` package `rLiDAR` (Silva *et al.*, 2017). They used a classical LMF algorithm to mark the tree tops, and then isolated each tree using a Voronoi tessellation (Aurenhammer et Klein, 2000) of the tree tops. Then they removed low pixels based on a threshold. The description of the method is elegant but it simply corresponds to a growing region algorithm with no constrains. Indeed, growing circles at constant speed from seed points will result in a Voronoi tessellation. Thus, the method presented by Silva *et al.* (2016) is an unconstrained growing region algorithm.

Zhen *et al.* (2015) proposed a LMF and region growing algorithm in which the growth rate changes with the size of the trees to simulate the competition between them. Thus, dominant trees are expected to impose their shape to some extent on co-dominant, more on intermediate ones and even more on suppressed trees. It could be seen as a fourth case in figure 4.13. We have never seen it used in any studies yet and we have never found any implementations, despite the fact the idea of introducing competition is interesting. However, one must be careful with the chosen constraints because growing regions at different speeds is very similar to multiplicatively weighted crystal-growth Voronoi diagrams Kobayashi et Sugihara (2002), which are not representative at all of tree shapes.

Issues with DCM-based algorithms

According to Li *et al.* (2012), DCM-based methods are not ideal because the DCM can contain inherent errors and uncertainties from a number of sources (see section 4.9). For example, spatial error can be introduced during the interpolation process from the point cloud to the gridded height model, which can decrease the accuracy of the tree segmentation process and of the derived metrics.

In addition, raster images have an inherent scale dependency, which means that a pixel does not always represent the same area. This implies that the same canopy can-

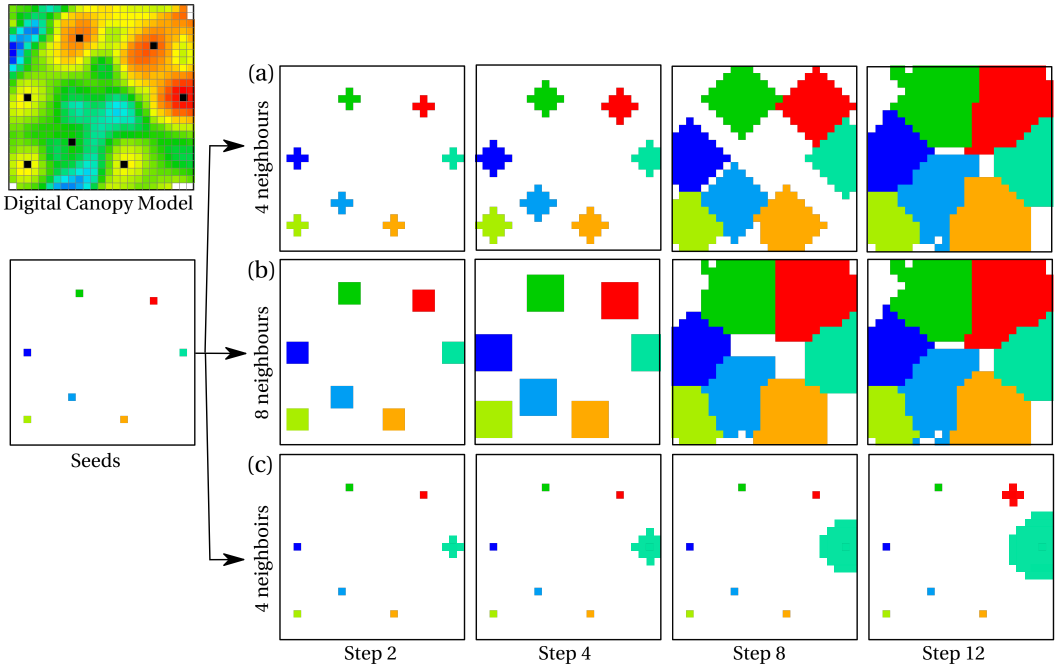


FIGURE 4.13 – Three variants of a growing region algorithm : (a) starting from seed points the regions are all grown simultaneously with a 4-neighbour structuring element until they reach each other or the region stops growing based on a given set of constraints, (b) the same but with an 8-neighbour structuring element, (c) the tallest tree is segmented first by growing a single region until the growth stops based on a given set of constraints, then the region of the second tallest tree is grown based on the same constraints, but the region may also be limited by the first tree that imposes its shape on the others.

not be processed the same way if represented with a coarse resolution image rather than a fine one. Figure 4.14 presents a trivial example showing that the algorithm must be adapted to each resolution. The scale dependency implies that a change of window size is necessary to detect the same tree tops with the LMF algorithm. This example is trivial and can be solved by a simple adjustment based on the known resolution of the image, but in the general case this problem is complex. The mathematical transformations required to rescale the parameters of an algorithm are not so obvious to identify. In computer vision, such scale dependency issues are treated within the framework of the “space scale theory”. Brandtberg *et al.* (2003) presented a segmentation method relying on space scale theory but its implementation would definitively be arduous a user who is not a computer vision engineer.

4.11.3 Algorithms based on the raw point cloud

New methods to segment individual trees directly from the LiDAR raw point clouds have been developed to avoid the scale dependency issue. Indeed, working at the point cloud level simplifies the issue of inaccuracies coming from the DCM. The absence of a DCM also removes the question of the choice of the algorithm to calculate this surface.

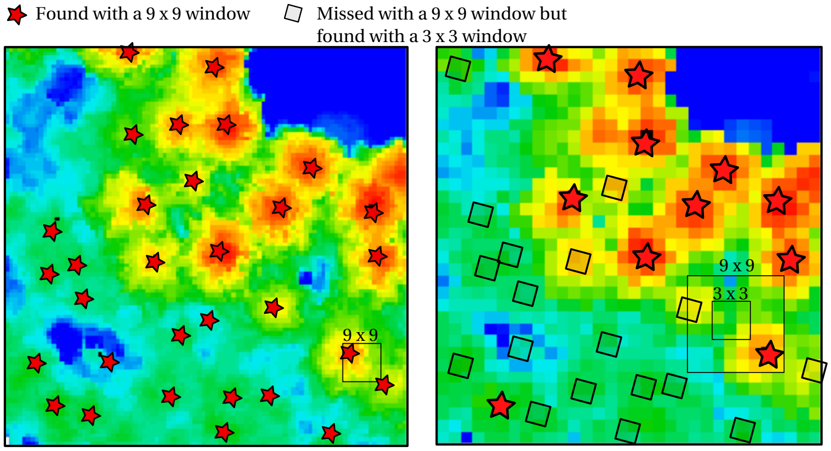


FIGURE 4.14 – Simple representation of the scale dependency issue in image processing. Fine structures are merge at a coarse resolution. To detect tree tops the moving window must be adapted to the image resolution to be able to detect the exact same number of trees. The LMF being a very simple algorithm, the adaptation is trivial and requires only resizing the window, but in general the issue is more difficult to solve.

To facilitate the process, several image-based algorithms available for classical image processing can also be converted to a point-cloud-based version. Section 4.3 and the 4.14 already presented how a morphological operator can be applied to a point cloud. In the same way, the LMF algorithm, for example, can be computed at the point cloud level. Figure 4.15 illustrates how the LMF algorithm can be applied at the point cloud level. The moving window is unique whatever the point cloud density with size expressed in units of the point cloud coordinates.

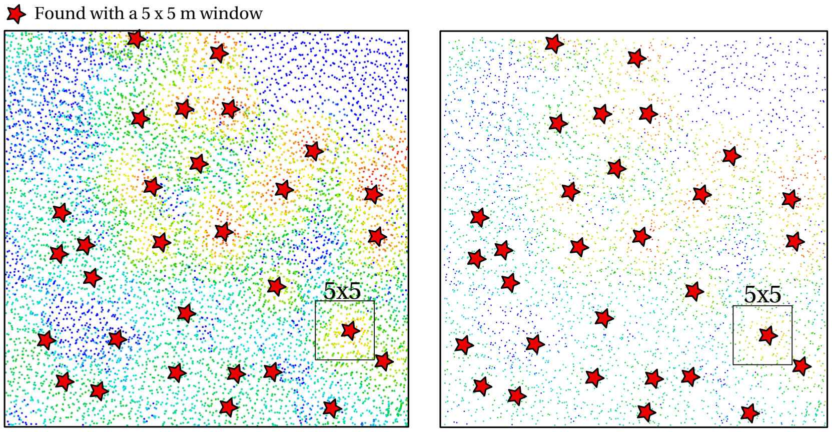


FIGURE 4.15 – At the point cloud level, algorithms are no longer dependent on the density of the point cloud and the scale dependency issue, which implies a variation of algorithm parameters as a function of the density and/or the resolution, is also non-existent. However, the point density may still affect the accuracy of the result.

Li *et al.* (2012) proposed a top-to-bottom region growing method at the point cloud level with a very simple constraint on the euclidean distance between points on the $x-y$

plan. Moreover it does not require to use a LMF. Vega *et al.* (2014) proposed another top-to-bottom growing region algorithm, which is very similar to that propose by Li *et al.* (2012), but with the simple constraint on the euclidean distance replaced by a more complex constraint on convex hull shapes. From the overall maximum of the point cloud, Hamraz *et al.* (2016) looked in 8 directions for the closest local minima to build an 8-point convex hull around the tallest tree. They then removed points belonging to this tree and reiterated the process until there were no more trees to segment. Yao *et al.* (2012) proposed an approach based on a normalized cut. The normalized cut was presented by (Shi et Malik, 2000) as an algorithm for image segmentation that maximizes dissimilarity between the segmented groups and similarity within groups. Gupta *et al.* (2010); Wang *et al.* (2008) also proposed methods based on a clustering approach.

However, to our knowledge, no software implements any of these algorithms based on the raw point cloud. Tao *et al.* (2014) stated that they used the Li *et al.* (2012) algorithm implemented in the Liforest software (`http://greenvalleyintl.com/software/liforest/`). However, we could not find any information or official documentation. The software appears to be closed-source which implies, as already highlighted in section 4.6, that there is no way to study or confirm the algorithms that are used. These development efforts therefore appear to remain at the stage of ideas in scientific papers without any possibility to be implemented by regular users.

4.11.4 About Local Maximum Filters (LMF)

We already mentioned the importance of the LMF to detect tree tops as seed points with DCM-based algorithms. Whatever the segmentation algorithm used, the number of trees detected results directly and only from the LMF algorithm, which has therefore the greatest importance in the tree segmentation process. The identification of individual trees rests on the parameters used to filter these seeds with an LMF algorithm, but also on the algorithm used to compute the DCM and on the parameters used to pre- and post-process this object (smoothing, pit-filling, etc. See section 4.9).

A local maximum is a point or pixel that has a value greater than any of its neighbours, with the neighbourhood being defined by a structuring element (see section 4.6 and 4.14). There are many possible structuring elements, the most trivial being, in the case of an image, the neighbours around a central pixel, and in the case of a point cloud, its equivalent i.e. a square with a given length side (see also fig. 4.14 and 4.15).

In a forest science context, the size and shape of the objects can vary substantially. In a given area of interest, some zones can contain small and dense saplings, while other zones can contain large trees that are sparsely dispersed. A unique structuring element is therefore not necessarily adapted to the entire area of interest. The ultimate LMF algorithm should therefore adapt dynamically its structuring element to the reality around each point or pixel to take into account such variation. We found that two approaches are used almost equally in the literature. For example, Hyyppä *et al.* (2001); Solberg *et al.* (2006); Véga et Durrieu (2011); Dalponte et Coomes (2016); Silva *et al.* (2016) used a structuring element of constant size, while Popescu *et al.* (2002); Chen *et al.* (2006); Zhen *et al.* (2013);

Barnes *et al.* (2017); Alexander *et al.* (2018) described methods to use variable structuring elements adapted to the size of the trees and Kwak *et al.* (2007, 2010) used an extended maxima transformation (a morphological method) to perform this task.

An important problem in the variable structuring element LMF is that it relies on some prior knowledge on the relationship between crown width and tree height, which may not necessarily be available (Jing *et al.*, 2012). Jucker *et al.* (2017) recently combined several sources of data collected worldwide and produced a model of crown allometry that could be of key importance for that purpose. From a computer science point of view, LMFs with variable window sizes also rely on algorithms that are not necessarily available in regular software. Our review of the literature did not enable us to cite a single software that enables the use of such enhanced LMFs. Digging into non-cited software, we found an R package `ForestTools` (Plowright, 2017) that has a very well designed feature to perform an LMF using a user-defined function, which feeds the dynamic computation of the window size.

An interesting point to note is that none of the point-cloud-based algorithms require an LMF as first step which, in a sense simplifies the methods and reduces the number of questions relative to this step providing an identification of the tree while segmenting.

4.11.5 Conclusion

Individual tree segmentation usually relies on segmentation methods resulting from decades of research in computer vision. However, the parametrization of these algorithms suffers from a lack of standardization that leave many options to pre- and post-process the data. This is attributable to two points in our opinion. The first one is biological : as there are different kinds of forest types it is to be expected that different methods will be successfully applied in different contexts. The second point relates to software : if the watershed algorithm is so commonly used it is not because it performs better (it does not), but because it is easy to understand and can readily be implemented from commonly used software. The lack of easy, free and open-source algorithms for individual tree segmentation leads researchers either to use what is available at hand, or to program bespoke scripts for their own needs. This situation explains the over-representation of algorithms that are relatively easy to program or use.

Research in tree segmentation at the point cloud level may open new processing options, but so far the algorithms we reviewed are, in fact, also region growing algorithms coded at the point cloud level. However, the fact they require neither a DCM nor a LMF may lead to different performance in terms of tree detection (in a good or a bad way). In any case, they will be slower to compute because of the quantity of data to process. Their development being currently only “text” format in peer-reviewed journals, the community has not yet taken advantage of these methods. This is why we state that there is a lack of *available* algorithms and software and an over representation of potential methods that cannot be used because of their absence in dedicated software. Our review also led us to believe there is an overly large body of literature presenting “new algorithms”. This is problematic for two reasons : (1) these “new algorithms” are often not “new” at all as demonstrated in this review and (2) there is a large body of existing methods needing to be used and tested before the need to develop new ones can be demonstrated. If current methods are hardly used because of a lack of availability, what is the interest of having more “unavailable” methods ?

4.12 The lidR package in R

One could consider that manipulating LiDAR data in the R environment is not a good idea, and this is a hardly arguable opinion. Understanding the reason for this is beyond the scope of the paper as it requires in-depth understanding of how the `R` language works. Interested readers may refer to section 4.12.4, which provides some hints about the question. However, beyond computer science considerations, the point is that many scientists and research teams (us included) in the fields of forestry and ecology *do* use R to manipulate LiDAR data to try and develop methods and statistical models to predict biometric descriptors of the forest or other quantities of interest. It is in this context that we are developing the `lidR` package (Roussel et Auty, 2017) available on CRAN, which enables users to manipulate LiDAR data in R in an efficient and straightforward manner. This section presents a brief overview of the package with respect to the content of this review.

The following section is based on version 1.4.0 of the package. Due to rapid development and regular updates, some parts of the section may be rapidly become outdated. However, the main ideas should remain relevant in the long term.

4.12.1 Overall approach

The `lidR` package aims to provide tools to manipulate LiDAR data acquired in a forest science context within the `R` environment. The goal is to enable users to try, test and explore methods in a straightforward manner. Thus, `lidR` is not only designed as a toolbox but also as a toolmaker. Indeed, manipulation of data into a programming environment usually implies that users wish to do something that does not exist somewhere else. The goal of a programming language is to create our own processes and tools.

Such goal can be achieved in any language. The efficiency of the C++ language in addition to very good libraries to manipulate point clouds, such as PCL or PDAL, is therefore a very good option to build and develop new methods. However, programming in C++ requires strong skills in computer science and implies a long and complex development process. Conversely, the `R` language enables users to write very complex processes in a few lines of code and requires very little knowledge in computer science. The `lidR` package fully embraced the R approach, providing tools that are meant to be straightforward and easy to use.

The following example represents well what we mean here by straightforward and easy to use :

```r
data = readLAS("lidardata.las")
metrics = grid_metrics(data, user_func, 20)
plot(metrics)
```

These three lines of code compute user-defined metrics in an area-based approach with 20 × 20 pixels on a given `las` file. The output is stored in the well known `data.frame` structure, which can easily be manipulated, even by beginners in R. A simple tweak also enables users to apply the same process, using a multi-core parallelized process over an entire dataset composed of several dozens or even hundreds of files :

```
dataset = catalog("path/to/folder/")
metrics = grid_metrics(dataset, user_func, 20)
plot(metrics)
```

The drawback of such straightforwardness of the R language is the inefficiency of the program both in terms of computation speed and memory usage. The `lidR` package is fast but not "blazing fast". The section 4.12.3 and 4.12.4 will cover these points. The point is that `lidR` is not designed to apply common routines to country-wide datasets. It can perform such a task, but it was not firstly designed for it.

Instead, the development is focused on providing a wide range of easy to use tools to enable R users to manipulate the data and algorithms found in the literature (see section 4.12.2). This is mean to offer the package as a repository of algorithms, and thus provide, as far as possible, a picture of the state of the art. Our thinking is the following : in the absence of implementations of the algorithms published of the literature, nobody will ever be able to criticize, compare, judge or take advantage of this phenomenal amount of work conducted in the field.

Rather than a countrywide data processor, we thus tried to provide a usable and straightforward open-source tool to promote reproducible science and easy development of new methods.

4.12.2 Features design

The LiDAR package covers a wide range of the content covered in this review. First and obviously, `lidR` supports both `las` and `laz` formats both as input and output (I/O) taking advantage of the `LASlib` and `LASzip` C++ libraries (Isenburg, 2013) via the `rlas` package (Roussel, 2017). Thanks to the underlying driver of I/O, `lidR` also supports `lax` files to speed-up spatial queries when reading files (Isenburg, 2012), an important technical point we skipped in this review (see also 4.16.3).

An important point raised in 4.12.1 is the fact that `lidR` is designed to try and explore methods, therefore being a toolmaker. For this reason most of our methods are designed to be highly flexible.

For example, `lidR` offers a progressive morphological filter (PMF) inspired from the Zhang *et al.* (2003) algorithm. The method is a point-cloud-based implementation (see 4.6.1 and 4.14). However, we did not follow Zhang's equations to build the sequence of window sizes and thresholds. Instead, we allowed users to provide any sequence they wish to. This provides a highly flexible tool and enables, for example, the use of decreasing window sizes as suggested in Pirotti *et al.* (2013). The `lidR` package will never dictate what users should do. It only enables users to do, as far as possible, anything possible by providing efficient algorithms under the hood (see 4.12.3 and 4.12.4). However, for interested users, we also made a specific function to compute the parameters using original equations published in Zhang *et al.* (2003).

Another example is linked to the area-based-approach method. The function `grid_metric` seen in section 4.12.1 allows an efficient rasterization of the point cloud and computes any user-defined metrics. The core objective of the function is not only to compute some pre-recorded metrics (many are pre-recorded for convenience and efficiency), but also to allow users to construct something new that does not exist elsewhere. For example a "new" metric could be the mean height of the points weighted by their intensity. This is extremely straightforward to compute in `lidR` :

```r
f = function(x, weight) { sum(x*weight)/sum(weight) }
grid_metrics(dataset, f(Z, Intensity), 20)
```

Several functions are designed like that to provide users complete freedom and autonomy. Tree segmentation algorithms, for example, always separate the local maximum filter (LMF) from the segmentation itself. The two processes being independent, users can rely on our methods or use their own methods, or even their own data computed by other means. The principle we followed is that a function must always perform one and only one task. This provides greater flexibility for the user.

Another important point that drives our development is the ability to use an open-source version of algorithms that do not have open-source implementations, or that to our knowledge do not have any implementations at all. For example, we implemented the Li *et al.* (2012) algorithm for tree segmentation at the point cloud level and we plan in a near future to implement methods presented by Hamraz *et al.* (2016) as well as Vega *et al.* (2014); Yao *et al.* (2012). We implemented an open-source version of the pit-free algorithm (Khosravipour *et al.*, 2016) and we implemented an algorithm from Wing *et al.* (2015) for the detection of snags based on intensity values (thanks to Andrew Sánchez Meador's contribution).

Some less "algorithmic" but not necessarily less useful examples of implementations include a gap fraction profile function as defined in Bouvier *et al.* (2015), a vertical index complexity function as defined in van Ewijk *et al.* (2011), a rumple index function based either on a Delaunay triangulation for sparse points or Jenness's algorithm (Jenness, 2004) for raster-alike structures (see section 4.10).

We obviously plan to implement more methods with the single objective : render published methods available to the community of users, so that they can take advantage of them, criticize them or develop them further.

We attempt, whenever possible, to provide algorithms both at the point cloud and raster levels. This is the case for the LMF (see section 4.11.4) and the rumple index, for example. However, the design of `lidR` focuses mainly on point cloud computations rather than raster computations. This is because several good tools are already available in R to manipulate rasters especially in the `raster` package (Hijmans, 2016).

Also, `lidR` has several tools to automatically extract regions of interest from a large set of files, to apply user-defined functions onto a set of files taking advantage of multiple processors, to segment individual trees, to merge geographic data with point clouds, to decimate point clouds, to display point clouds, etc. The ultimate goal of the development is to enable users to try more things than what is usually proposed in classical software, which are toolboxes with a set of predefined tools. We will not list here all the features of the `lidR` package here, but rather focus on the main concepts that drive its development.

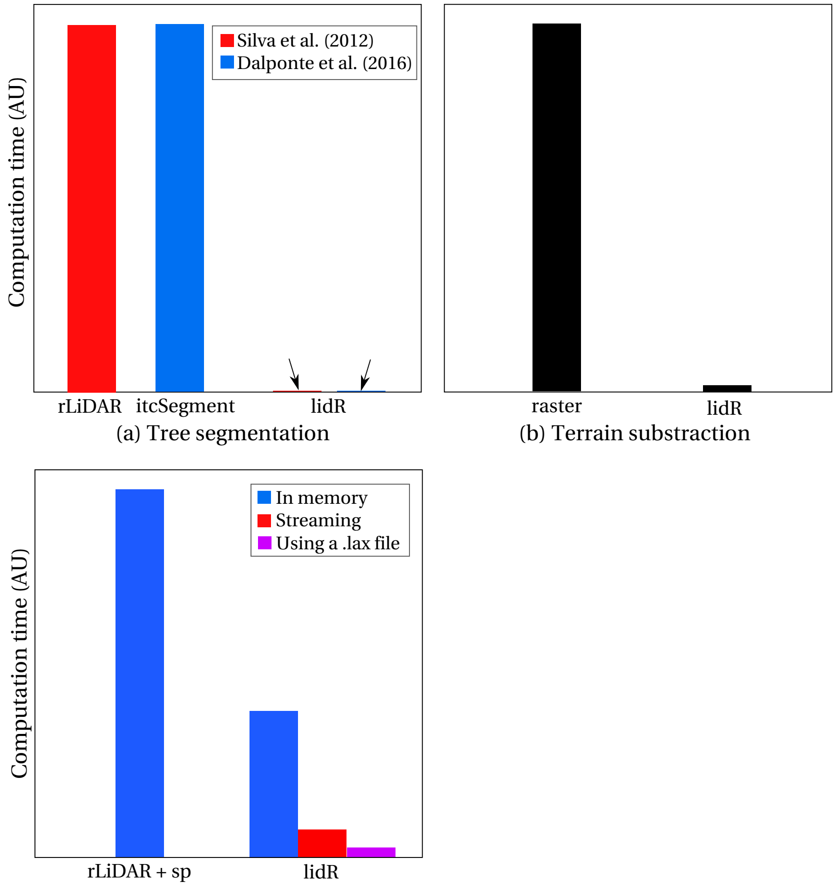
4.12.3 Computation speed

Although it is not our first concern, the development of the `lidR` package makes every effort to optimize computation speed. We want the functions to run in a "convenient" time frame without attempting to build a blazing fast software, a purpose that would be better served by other programming languages. To give the reader an idea of what we mean by "convenient", we compared the computation time of some functions and algorithms available in `lidR` to other ways to obtain the same results with other `R` tools (fig. 4.16).

In figure 4.16a we compared our implementation of the algorithms originally presented in Dalponte et Coomes (2016) and Silva *et al.* (2016) with implementations made by the original authors in, respectively, the `itcSegment` and `rLiDAR` packages. We also compared the subtraction of a digital terrain model (DTM) from the point cloud our in `lidR` package to a common approach in `R` based on the `raster` package (figure 4.16b). Finally, we compared the extraction of a single polygon of few hundred square meters of data from a medium size file (figure 4.16c). In this latter analysis we also compared the 'in memory' way vs. the 'streaming' way (see also section 4.16).

In each case, our algorithms were comparatively drastically faster, to the point that we had to use relatively small samples to keep the bar visible on the graphs. With larger files the difference would have been bigger. However, this actually does not mean that our algorithms are that fast because the other implementations are actually very slow. We regularly make improvements to our algorithms but their current (in version 1.4.0) corresponds to what we consider "convenient". We believe that making them 10 times faster would currently not bring much change to the user experience because for most uses computation times appear as instantaneous to the human mind.

However, this would not hold true any more for larger datasets covering wide areas. Computing during 1 hour or 10 hours is very different than 10 ms and 100 ms. On regional- or country-wide dataset, for example, the computation time may be a limitation. We continuously strive to improve computation speed but at some point the bottleneck lies no longer in the algorithms themselves, but in the interaction between the R environment and the underlying C++ code. This is why in figure 4.16c the 'streaming' way outperform the 'in memory' way by far. The former drastically reduces the interaction between R and the underlying C++ code. This point is technical and further details are provided in section 4.12.4 and in 4.16. Basically, to improve computing speed, one approach could be to do everything at the C++ level, but this would come at the expense of the providing users the ability to interact dynamically with the data at the R level. This is not what we aim for and thus, by design, `lidR` will never be blazing fast. Despite not being our main goal, we make our best efforts to make it as fast as possible for convenience.



(a) Tree segmentation

(b) Terrain substraction

(c) Single polygon extraction

FIGURE 4.16 – Illustration of the computation speed of `lidR` algorithms relatively to other methods in R. Actual computation times can vary a lot as a function of the computer and the operating system. Also, the presented differences in computation time can drastically vary as a function of the size of the dataset. Here we compared methods for small datasets, otherwise the gap would have be much greater and the `lidR` bar invisible. (a) Comparison of two tree segmentation methods developed by the original authors vs. our implementations. (b) Subtraction of a DTM from a point cloud using `raster::extract` and our `lasnormalize` implementation (c) Extracting a single small polygon from a larger file by loading the whole file in R memory, then clipping a subset (in memory) or clipping while reading (streaming).

4.12.4 Optimizations in `lidR`

This section is technical and describes how we dealt with some restrictions imposed by the R language. Without explaining the computer science under the hood, we simply try here to provide the reader a feel of the restrictions imposed by R, and why we introduced section 4.12 saying that the LiDAR data manipulation in R is not necessarily a good idea.

The first restriction is the fact R is a weak typed language that allows to manipulate only 64 bits `double` and 32 bits `int`. The `las` specifications are designed to enable to store LiDAR data using the restricted amount of memory strictly necessary (see section 4.5) using values stored either in 1, 8, 16, 32 or 64 bits. For example, intensity values are stored using 16 bits. At the R level we cannot use 16 bits to store the intensities because such type does not exist. Therefore, we have to use 32 bits, meaning that we use twice more memory than required. Worse, the classification of the points (ground, vegetation, building, water, etc.) is stored using 8 bits in las files, but we have to store it on 32 bits as well so we use four times more memory than required.

Therefore, `lidR` uses approximately twice as much memory to load a dataset than what is really required. This problem is not solvable and is a limitation of R itself. The only solution to solve the problem would be to do everything at the C++ level without providing the user the ability to manipulate the data at the R level. As previously states, this is not our choice, and thus this issue and all the others coming with it represent the irreducible cost of the trade-off between the straightforwardness of the language and its efficiency. And `R` is very straightforward...

To deal with this problem we enable users to load only the relevant fields of the data in R. This step is achieved in a memory efficient manner at the C++ level in the `rlas` package and allows, at the R level, to save a lot of memory by loading only useful data. The following code enables to load the three spatial coordinates and the intensity without losing a single bit of useless memory at the R level.

```r
readLAS(file , select = "xyzi")
```

Another limitation is the way in which LiDAR data is stored. Points are simply stored in a table, which is the best way to enable users to manipulate the data in a classical way in R, an environment in which everything is designed to manipulate vectors and tables. This storage mode is therefore relevant at the R level for R users. However, it is highly inefficient for creating algorithms that have to deal with point clouds. Thus, we regularly have to create a copy of the point cloud at the C++ level to transform it into a vector of points, and then transform back the result into a table. Obviously, this is memory inefficient and time consuming, but we attempted to design our algorithms in a way that reduces this issue as much as possible. Whenever possible, we write efficient algorithms working directly with tables, but sometimes we have to make the transformation to write algorithms that run, for example 100 times faster, than a table-based one, which justifies the memory usage cost. With very large point clouds loaded in the memory this may become a limitation. Again, there are several ways to solve this, but they all imply removing the ability for the user to manipulate the data at the R level.

Another limitation of R is the non-existence of pointers. This issue is strongly related to the previous one. At the C++ level we can create a set of points allocating memory blocks only once on the heap, and we can use pointers to these block to create a subset of the original data allocating only 8 bytes of extra memory per point (the size of a pointer on a 64-bit machine). This not possible at the R level, so a subset of points creates necessarily a copy of the points with no possibility to use an already allocated memory block. This is in addition to the use of twice more memory than what is really required to store the point cloud. Thus, filtering a point cloud at the R level is memory inefficient. To overcome this issue, the `rlas` package takes advantage of the `LASlib` library and allows filtering in reading time (streaming filter) at the C++ level. This enables users to load only the amount of desired data in a memory efficient way. The following code can be used to load the three spatial coordinates and the intensity values for the first returns only without loosing a single bit of useless memory at the R level.

```
readLAS(file , select = "xyzi", filter = "-keep_first")
```

However this is not a solution to all problems and regularly the user will have to create deep-copies of the data. There is no straightforward solution to this problem that would avoid not returning the data at the R level. Another approach could have been to design the package in way that is opposed to all common usages in R, which would create more problems than it solves.

The `lidR` package contains several issues and optimizations like those described here that result from how R works. Describing each one would be beyond the scope of this manuscript, but for each of them we have a solution to reduce the problem without being able to solve it entirely. The great advantage of R is that it enables users to program despite having very little knowledge in computer science. Users of the `lidR` package must realize this comes at a cost. We improve the code almost on a daily basis to limit this issue. This section simply aims to allow the reader to have an overview of the limitations of R to manipulate LiDAR data. We provided here some rough explanations as to why we said that `lidR` is not designed to process country-wide datasets and will never be. Instead, we designed `lidR` to be a good tool for experimentation on small and medium datasets.

4.13 Conclusion

We reviewed the main methods and algorithms currently used to manipulate and analyse LiDAR data in forestry and ecology contexts. We dug into the source code of several software and tools and illustrated many concepts of computer science related to their use. Our review revealed several mistakes and misconceptions made in the current literature.

The current state of the art is relatively tidy. The workflow is almost always the same : ground segmentation > digital terrain model > normalization > area based approach and/or individual tree segmentation, and there is a large body of literature to explore and test new methods and to keep going further. Also, there is no longer a need to prove that the technology works and provides useful data. Considering the huge corpus of LiDAR data and successes, there is no longer a need to introduce studies, as we did in this paper, by saying that “ALS is revolutionizing the way we make science”. It already has.

However, the current state of the art is not accurate. Our initial goal in doing this review was to describe and explain the common methods used, present the underlying computer science concepts and provide some guidelines according to the state of the art. We deviated from this initial goal because the corpus of publications we reviewed did not enable us to achieve this goal, even when coupled with our deeper investigation of existing software and algorithms.

Key information are regularly missing in the material and methods sections of scientific publications. How did the authors segment the ground points ? How did they compute the digital canopy model ? How did they normalize the point cloud ? These are examples of questions that we could too often not answer in a satisfactory manner. When the information is provided, it is regularly poor, non informative, partial and sometimes wrong. The name of the software is often not mentioned, confusion between methods were found and parametrization is virtually never mentioned. This is the main problem raised in this review. By no means do we consider our own research to provide a better example. Indeed, our own publications were cited among the examples of such bad practice. One of the fundamental reasons for this problem is that despite the researchers' best intentions, dataset are often pre-processed by the provider using proprietary software. It is definitively a problem that originates at the hardware level and often we simply have to accept the limitations of what was provided. However, we strongly encourage the community to be as accurate as possible when describing methods because we can do much better. As a first step towards better practice, we encourage authors to at least state clearly the part of the workflow that were not mastered. It is better than nothing and more fair that giving the impression of a mastered workflow.

Our review also left us the impression that there is an overly large body of literature presenting "new algorithms". There are pros and cons to this situation. While it is a good to have the opportunity to take advantage of a wide range of methods, the drawback is that in practice most of the algorithms we found in the literature do not have any implementations. We thus mentioned them very quickly, or even skipped them altogether, because they are just text in publications. As a community, we believe that rather than "new algorithms" we need to use and test existing ones first. To improve this point we suggest that the community should either publish papers presenting new algorithms clearly *or* papers focusing on industrial/ecological results, but not both. The logic behind this is that we cannot perform well at both tasks. Developers should lead the development of new methods and ecology or forestry research scientists should use this to conduct their studies in a solid, reproducible workflow. Indeed, our review revealed that publications that attempt to make both usually fall short of presenting a revolutionizing algorithm, whereas papers dedicated to the first task only tend to present more interesting, "newer" and more robust methods.

Finally, too many algorithm are closed-source. Even if an algorithm is the best, if we cannot explain how it works under the hood it is not useful for us as a scientific community. It can also lead to wrong attributions of authorship, and probably to wrong questioning about accuracy of the algorithm, as emphasised mainly in section 4.6. We strongly encourage authors of methods to provide a source code, or at least a pseudo code within scientific papers. Otherwise, the community will not be able to take advantage of these methods. And for developers, we strongly encourage conscientiousness and accuracy when writing the software documentation because not everybody can read and understand the source code to check what the software actually does.

In light of this situation, we presented a new R framework to process LiDAR data. This framework has been designed to be convenient and to enable users to test new algorithms, new processes, new tools. We focus the development of the package on algorithms from the literature only to produce a simple and straightforward tool enabling users to test and explore LiDAR data in any way they wish. We took care of not inventing anything new. Indeed, we did not create any "new methods", but instead tried to provide a wide range of existing methods found in the literature. Such a repository is meant to help the community to take advantage (or not) of these methods. Additional contributors wishing to include methods that we may have missed to the framework are more than welcome. We expect that this is only the beginning of a project from which the forestry and ecology research communities will hopefully benefit. A lot of further development is still upcoming.

4.14 Mathematical morphology

The field of mathematical morphology has contributed a wide range of operators to image processing, all based upon a few simple mathematical concepts from set theory. This field of mathematics and its applications in image processing are beyond the scope of this paper, but some principles are key to understanding some of the presented material, especially in section 4.6. This appendix aims to explain some key points about morphology. For simplification and because more advanced concepts are not required for this paper we will focus on two operations i.e. erosion and dilation. These are two fundamental operations in morphological image processing on which all other morphological operations are based. They were originally defined for binary images, and were later extended to grayscale images. We show here how these operations can be extended to point clouds.

4.14.1 Erosion and dilation of binary images

The erosion of a binary image I by a structuring element S produces a new binary image I'. Erosion removes pixels located at object boundaries by applying the following rule : *the value of the output pixel in I' is the minimum value of the pixels from I in a given neighbourhood.* The neighbourhood is defined by the structuring element S. This is illustrated in figure 4.17.

Dilation can be seen as the opposite of erosion. Dilation adds pixels on object boundaries applying the following rule : *the value of the output pixel in I' is the maximum value of all the pixels from I in a given neighbourhood.* The neighbourhood is defined, again, by the structuring element S. This is illustrated in figure 4.18

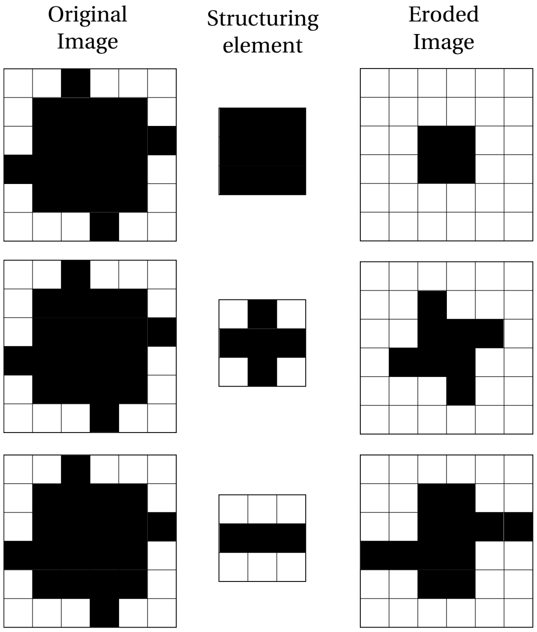


FIGURE 4.17 – Example of erosion using 3 different structuring elements for the same original image.

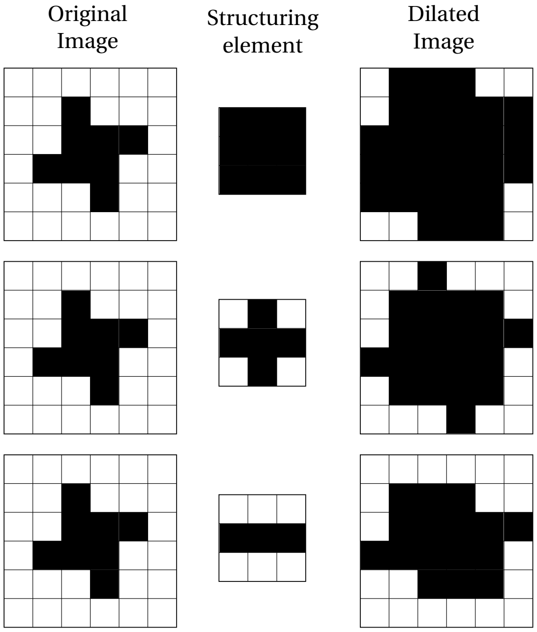


FIGURE 4.18 – Examples of dilation using 3 different structuring elements for the same original image.

4.14.2 Erosion and dilation of non-binary images

As described in the previous section, the morphological operations are not specific to binary images and can be applied to grayscale images applying the same rules. The value of a given pixel in I' is the maximum/minimum value of all pixels from I in a neighbourhood defined by the structuring element. This is illustrated in figure 4.19.

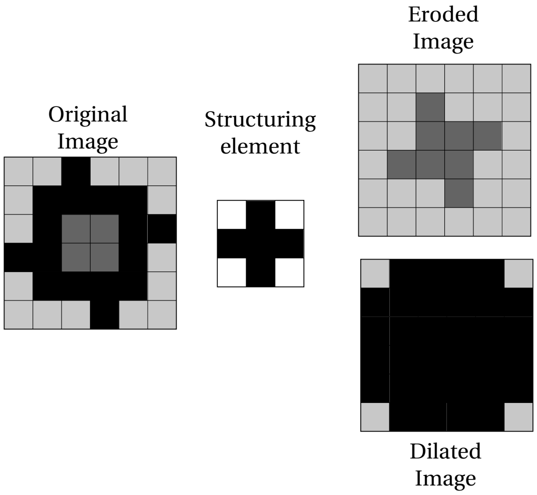


FIGURE 4.19 – Example of erosion and dilation for the same original greyscale image.

4.14.3 Erosion and dilation of point clouds

Finally, the definition of the morphological operations as the minimum/maximum element within a given neighbourhood defined by a structuring element can be applied to a point cloud. The major difference is that the structuring element is no longer restricted by the discrete nature of the images. This is illustrated in figure 4.20 using a disc as structuring element.

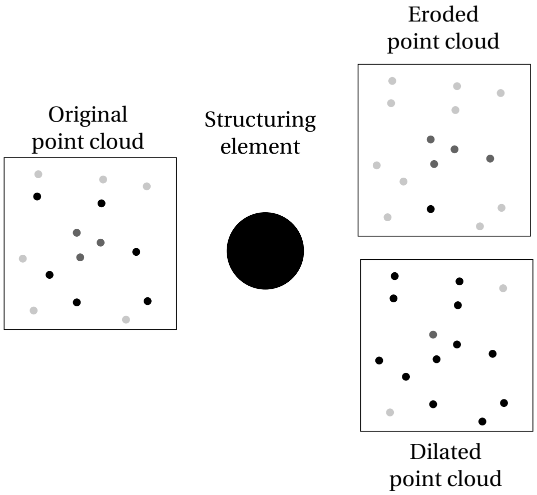


FIGURE 4.20 – Example of erosion and dilation of a point cloud using a disc as structuring element.

4.15 Triangulation

Triangulation is a very important topic in computational geometry. In geometry, a triangulation is a subdivision of a planar object into triangles. It has a very wide range of applications such as 3D modelling, network mapping or the finite element method. Point cloud algorithms make use of it, as seen in section 4.9 with the Khosravipour *et al.* (2014) and Khosravipour *et al.* (2016) algorithms, in terrain modelling as seen in section 4.7, or for ground segmentation with the Axelsson (2000) algorithm. According to our review, it appears pertinent to provide few key points about triangulation. There are several type of triangulations, the most common being the Delaunay triangulation, that have some interesting properties :

Triangulation : if the triangulation does not follow any specific rule of construction it does not have any special properties. For a given set of coordinates there are many different triangulations (fig. 4.21)

Delaunay triangulation : is a special case of triangulation. The Delaunay triangulation abides by some construction rules and maximizes the smallest angle, thereby avoiding “long” triangles. This type of triangulation is unique. The Delaunay triangulation is recognized to be the best triangulation both for its unity and its regularity.

Constrained Delaunay triangulation : forces certain required segments into the triangulation. Because a Delaunay triangulation is unique, a constrained Delaunay triangulation contains edges that do not satisfy the Delaunay conditions. Thus, a constrained Delaunay triangulation differs from the real Delaunay triangulation.

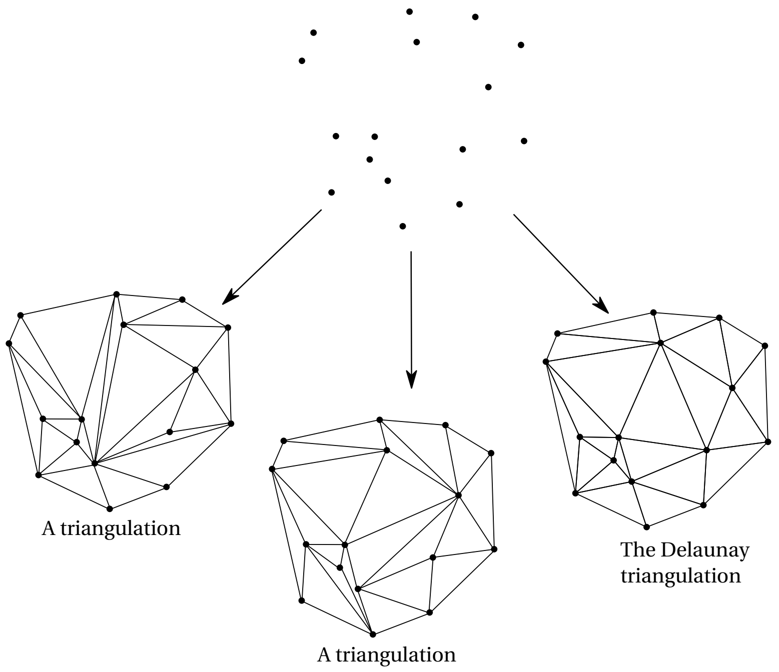


FIGURE 4.21 – Different triangulations of a set of points including the unique Delaunay triangulation

Actually, a Delaunay triangulation is not unique in all situations. It is "almost" unique, but if the point cloud contains co-circular points, i.e. more than three points belonging to the same circle, there are several possible Delaunay triangulations. This is especially true for a triangulation of regularly spaced points (fig. 4.22). The issue is that all triangulations do not have the same properties. For example, the total area of the triangles in 3D may differ from one triangulation to another.

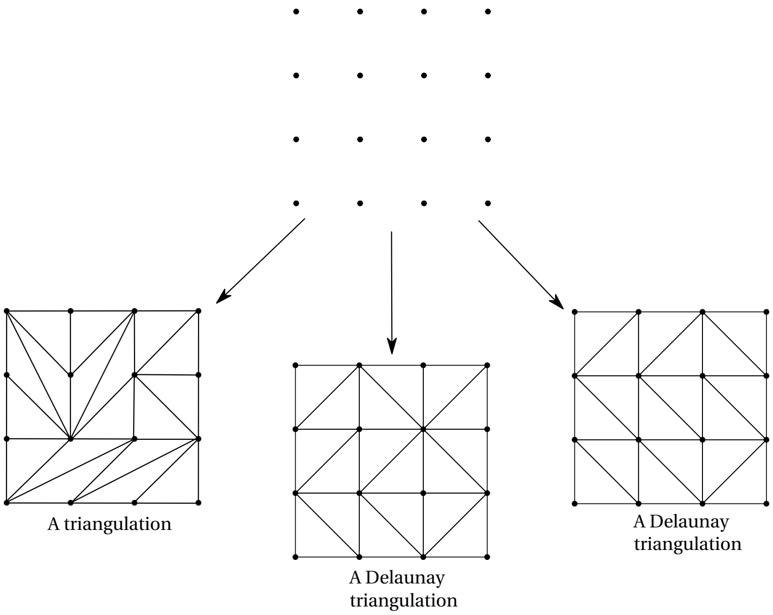


FIGURE 4.22 – Different triangulations of a set of regularly spaced points. There are several valid Delaunay triangulations in such cases.

4.16 'Streaming' computations vs. 'in memory' computations

Section 4.12.3 refers to the difference between a "streaming" computation and an "in memory" computation. This appendix aims to explain the conceptual difference and the computer science under the hood that makes the first one outperform the second one is many cases. For this we will take the example provided in figure 4.16c. Why does extracting a polygon in a streaming way is much faster than in memory, and why does the `.lax` file enable a significant gain.

Lets consider we have a file with *n* points recorded. We also have a polygon *P* for which the extent is contained in this file.

4.16.1 Extracting a polygon the 'in memory' way

This way is the most common that is used every day by everyone. It consists of reading a file and loading its entire content into the processing memory (RAM) of the computer. Our example implies loading the *n* points into the memory. Then, clipping a polygon consists of testing for each point if the point does fall or not into the polygon. If a point falls into the polygon it is added into the output, otherwise nothing happens.

At the computer level, three things happen here. First, an allocation of processing memory for n points. Second, an iteration over each point to test if they are within the polygon. Third, another allocation of memory to store the subset of points. These steps may be computed rapidly using the proper data structure at the C++ level. But we are comparing things that happened at the R level and R is highly inefficient with memory management.

`lidR` relies on both C++ code and R code, always keeping in mind to reduce memory allocations, especially at the R level.For this reason computing speed outperforms methods that are pure R (blue bars in figure 4.16c). However, regardless of the underlying C++ code, there are several interactions at the R level that drastically slow down the algorithm.

4.16.2 Extracting a polygon, the 'streaming' way

Streaming reduces considerably the memory allocation at the C++ level, but more importantly at the R level too. The test of points in a polygon is performed while reading the file, and not after having read the entire file.

At the computer level, several things happen. First, a tiny block a memory is allocated to store a single point, and the first point is read from the file. A test is then performed on this point to determine if it is located within the polygon. If the point falls into the polygon, it is stored into the output, otherwise it is deleted. Then, the second point is read by recycling the memory allocated for the first one. The test is performed again and these steps are repeated for each of the n points sequentially.

At the end of the process the allocated memory consists only of that required to store the points of interest (those that fall into the polygon) plus a single tiny block of buffer memory. This happens entirely at the C++ level, then the R memory is allocated only once when returning the result. Finally, the n points were tested, but we iterate only once over the n points instead of twice in the "in memory" way, one to read the file, one to process the points. Also, the memory allocations are drastically reduced which leads to a good speed-up.

4.16.3 Extracting a polygon, the "streaming" way with `lax` files

A `lax` is a tiny file coming along the `las` file. In our opinion, with `laz` files, these two file types constitute two major contributions of Martin Isenburg to open-source tools for LiDAR data manipulation. A lax file is basically a file that indexes some spatial regions into a `las`. Using the `lax` it is possible find which part of the file contains the bounding box of the polygon, and thus there is no longer a need to read and test the n points contained in the `las` file, but only a fraction of them in the identified subregion of the whole file. Computation time is thus made faster by reducing the number points needing to be tested.

4.16.4 Conclusion

A streaming algorithm may perform drastically faster in some conditions. `lidR` is not a streaming software because we *want to* provide the data to the user at the R level, and because we rely also on other R packages that do not provide streamed version of the algorithms. To be honest, we are not necessarily skilled enough to create streaming algorithms in all cases. Indeed, writing a streaming algorithm is much more difficult and binding, and R was definitely not conceived to work with them.

In practice it is much more complex than what is described here and this section is only a short pedagogic introduction. In `lidR` some function have both an 'in memory' and a 'streaming' version, but not the majority of them. This section aims mainly to explain why, by design, `lidR` will never be blazing fast. At the same time it also explains roughly and partially why `lastools`, which is an entirely streamed software suite, *is* blazing fast and memory efficient and absolutely designed to process country-wide datasets.

Conclusion

Bien que le LiDAR aéroporté ait fait ses preuves comme outil de télédétection, l'étude de la littérature scientifique sur le sujet a permis de démontrer un manque de standards dans les méthodes utilisées pour manipuler et analyser la donnée. Densité de points, intensité émise, sensibilité du capteur, angle d'incidence des rayons, divergence du rayon, choix d'un algorithme de classification des points au sol, choix des méthodes d'interpolation spatiale, choix des logiciels utilisés, choix des algorithmes de discrétisation de l'onde complète sont autant de facteurs pouvant faire varier la distribution spatiale des points échantillonnés, et donc la façon dont la donnée ALS est interprétée et analysée.

Il n'est nullement surprenant ni problématique que des modèles prédictifs valables au Québec par exemple, ne soient pas valables en France parce que la structure et la composition des forêts est très différentes entre ces deux régions géographiques, et même au sein de ces deux régions. Cependant, il serait très problématique que des modèles développés pour une région donnée fassent des prédictions différentes en fonction de certains choix d'acquisition et de traitements, d'où la nécessité d'une chaîne d'acquisition et de traitement standardisée. Une telle approche standardisée n'existe pas à ce jour.

Ce manque existe à deux niveaux : (a) au niveau matériel ou *"hardware"* qui correspond à l'acquisition des données et (b) au niveau logiciel ou *"software"* qui correspond au traitement des données brutes. Ces deux niveaux de standardisation ont été abordés dans cette thèse avec deux approches différentes.

S'il est, en pratique, impossible d'acquérir tous les jeux de données avec le même dispositif d'acquisition et les mêmes paramètres, il est en revanche plausible de chercher à normaliser la donnée et la recalculer « comme si elle avait été acquise avec un dispositif standard ». Nous avons montré dans cette thèse que cela est possible, même si potentiellement difficile. Pour cela, nous avons défini un dispositif d'acquisition standard émettant des impulsions de longueur nulle, de diamètre nul, avec une densité au sol infinie et un angle d'incidence toujours nul. L'enjeu étant, non plus d'utiliser les métriques brutes comme variables explicatives, mais les métriques corrigées et normalisées pour ce dispositif théorique. Ainsi, les modèles prédictifs seraient tous construits de la même façon à partir du même dispositif d'acquisition hypothétique et cela permettrait de comparer des données potentiellement incompatibles.

En se basant sur des considérations physiques et probabilistes simples, nous avons démontré la pertinence de deux modèles mathématiques théoriques pour corriger respectivement certains effets de la densité de points et de l'angle d'incidence des rayons dans le cadre d'une analyse par approche zonale. Ces modèles sont mathématiquement simples, mais leur portée est aussi limitée.

Le premier modèle proposé corrige, en théorie, les effets de densité de points à plusieurs échelles sur n'importe quel type de couvert forestier, mais ne traite que d'une unique métrique. De plus, il nécessite une calibration à partir d'un jeu de données à haute densité représentatif de la zone d'étude. Nous avons cependant montré comment se passer d'un tel jeu de données au prix d'une perte d'exactitude. Le message principal va en fait audelà de la correction d'une métrique. La densité de points *a* un impact sur les métriques et donc sur les modèles prédictifs, soit intrinsèquement sur certaines métriques plus sensibles à cette variation, comme les métriques dérivées de plus d'une coordonnée, soit par voie de conséquence, simplement parce qu'une variation de densité vient toujours avec une variation d'un ou plusieurs autres paramètres. A faible densité de points les effets intrinsèques se font largement sentir dans les chevauchements de lignes de vol (*overlaps*), régions de l'espace où la densité est supérieure. Il s'agit aussi de régions de l'espace où les angles d'incidence des rayons laser sont plus importants.

Le second modèle corrige, en théorie, toutes les métriques unidimensionnelles dérivées de la distribution verticale des hauteurs des retours des effets d'angle d'incidence des rayons, mais est limité à certains types de forêts. Il n'est pas trivial, en pratique, de définir clairement ces types. Le message principal va, en fait, au delà de la correction, et montre que ces problèmes peuvent et doivent être abordés de manière théorique si on souhaite apporter une réponse claire (mais pas forcément définitive). Dans notre cas, nous démontrons l'existence des effets d'angle. Cependant, nous montrons aussi qu'il sont généralement faibles quand les angles sont faibles, démontrant ainsi pourquoi il est difficile de les observer avec une approche statistique. En fait, ils sont même encore plus faibles que ce qui est présenté dans l'article à cause de la superposition de lignes de vol non prise en compte (mais qui est formulée dans le matériel supplémentaire de l'article). Cependant, le résultat principal tient dans le fait que nous démontrons que les effets d'angle sont fonction de la structure locale de la forêt, et donc pas nécessairement négligeables partout.

Ceci ne résout évidemment ni à la question de l'exportabilité des modèles prédictifs locaux basés sur la modélisation statistique, ni à la question ouverte dans l'article 1 portant sur les raisons expliquant pourquoi la hauteur moyenne de la canopée est sensible à l'angle d'incidence. La première question est fondamentalement difficile et nécessite de repenser nos méthodes d'analyse à la racine, ce qui n'est pas vraiment la question étudiée dans ce doctorat. Dans le cadre de cette thèse, c'est le suivi temporel de la ressource, via des acquisitions LiDAR successives qui est abordé. La deuxième question, plus pragmatique, est probablement liée à de la géométrie statistique. À l'instar du modèle d'angle qui ne décrit pas les effets d'angle en forêt résineuse pour des raisons géométriques, les effets d'angle sur C_{mean} sont vraisemblablement géométriques eux aussi.

La piste géométrique a été explorée dans ce doctorat par deux reprises. Dans les deux cas, la piste n'a pas aboutie. Les mathématiques sous-jacentes sont plus complexes et reposent toujours sur des hypothèse très audacieuses sur la distribution spatiale des arbres et leur forme. Statistiques spatiales et projections de forme 3D seront au programme de futurs développement dans ce sens. Nous avons échoué à chaque fois à transférer les modèles d'optique géométriques à notre problématique, mais nous restons convaincus que l'optique géométrique est la clé, ou plutôt la base, de la solution.

Par ailleurs, l'applicabilité réelle des modèles proposés est discutable. Les modèles théoriques visent plus à prouver et expliquer l'existence d'un fait qu'à proposer une solution « prête à l'emploi ». Ce n'était par ailleurs pas le but premier de ces travaux. Les solutions proposées pour une applicabilité pratique sont assez difficiles à mettre en place. Nous mettons en effet en évidence, à chaque fois, que les effets sont dépendants de la structure locale de la forêt ; structure qui n'est accessible que grâce au LiDAR. Le serpent se mord la queue. Il est donc nécessaire de trouver des moyens d'approximer la réalité, quitte à perdre en justesse. L'article 1 propose une bonne solution à cette question, mais difficilement applicable. Quant à l'article 2, une approximation pragmatique doit encore être trouvée. Peut-être en calculant des distributions locales à des échelles intermédiaires entre la placette et la forêt (quelques milliers de mètres carrés).

Ainsi, il s'agit de fort peu de choses devant le chemin restant à parcourir pour prendre en compte plus de paramètres et plus de métriques dans un cadre théorique plus large. Certains effets attendent des développements mathématiques et physiques complexes qui n'ont pas été trouvés dans le cadre de ce doctorat. Une chose est sûre, seule l'approche théorique permettra d'aller plus loin dans les démonstrations généralistes. C'est ce qui se fait pour la normalisation de la coordonnée d'intensité des retours LiDAR et c'est ce qui doit se faire pour les coordonnées spatiales.

Si, dans cette thèse, nous n'avons traité qu'une petite partie du problème, nous pouvons cependant essayer de dresser une image plus large du comportement des métriques dérivées en les classant selon plusieurs catégories, d'abord sur leur nature. Nous proposons deux catégories à cet égard, soit les métriques dérivées qui sont des statistiques, et qui résument avec un nombre la distribution spatiale des retours, et les métriques qui ne sont pas des statistiques et qui ne résument pas l'ensemble de la distribution. Ces dernières peuvent être dérivées à partir d'un unique point ou un objet construit algorithmiquement (dérivé du modèle numérique de canopée par exemple). Ensuite, elles peuvent être catégorisées selon leurs dimensions. Nous proposons deux catégories dans ce cas encore, soit les métriques uni-dimensionnelles qui ne considèrent qu'une coordonnée sur les nombreuses disponibles et les métriques multidimensionnelles tirant profit d'une plus grande proportion de l'information. Les analyses et réflexions proposées au cours de ce doctorat laissent penser que la sensibilité à la densité de points et à l'angle d'incidence des rayons varie fortement entre ces catégories.

Enfin, le troisième article traite de la standardisation logicielle à travers une revue de la littérature relative aux algorithmes utilisés pour traiter les données LiDAR. Si cette standardisation est d'apparence plus simple, elle est en réalité bien plus difficile. Nous avons démontré dans ce chapitre que la littérature scientifique regorge de publications dans lesquelles les méthodes de traitement des données ne sont pas décrites, ou le sont de manière si partielle qu'elles sont au mieux incompréhensibles et au pire fausses. Un grand nombre de publications rapportent des algorithmes peu élaborés inventés de toutes pièces pour les besoins de l'étude sans tenir compte que des méthodes similaires, reposant sur des mathématiques et des algorithmes robustes qui existent déjà et ont fait leurs preuves depuis des décennies. À cela s'ajoute le fait que peu de chercheurs s'intéressent à ce que les logiciels calculent réellement, faisant confiance aveuglement au logiciel. Enfin, les derniers développements algorithmiques pour réaliser certaines tâches propres aux traitements des données ALS ne sont quasiment jamais utilisables par les utilisateurs, faute d'implémentation. Il est dès lors impossible de définir un ou des standards robustes sans une autorité compétente et respectée capable d'éditer des lignes directrices simples, claires et pragmatiques. Et c'est le message principal de ce dernier chapitre. Au regard de la littérature, la communauté des sciences forestières a besoin de recommandations techniques et d'une plus grande rigueur scientifique.

Notre contribution à ce point est mineure. Nous n'avons ni l'autorité, ni la compétence pour éditer de telles recommandations. Néanmoins, la création du package `lidR` va dans ce sens et est l'apport majeur de ces trois ans de travail. Le troisième chapitre, au travers de la revue de bibliographie, explique le développement du package qui, dans les faits, apporte à la communauté scientifique un lieu de développement commun qui apportera bien plus que tous les articles qui auraient pu être publiés si du temps de recherche avait été alloué à la place.

Bibliographie

AHMED, O. S., FRANKLIN, S. E., WULDER, M. a. et WHITE, J. C. (2015). Characterizing stand-level forest canopy cover and height using Landsat time series, samples of airborne LiDAR, and the Random Forest algorithm. *ISPRS Journal of Photogrammetry and Remote Sensing*, 101:89–101.

AKIMA, H. (1978). A Method of Bivariate Interpolation and Smooth Surface Fitting for Irregularly Distributed Data Points. *ACM Transactions on Mathematical Software*, 4(2): 148–159.

ALEXANDER, C., KORSTJENS, A. H. et HILL, R. A. (2018). Influence of micro-topography and crown characteristics on tree height estimations in tropical forests based on LiDAR canopy height models. *International Journal of Applied Earth Observation and Geoinformation*, 65(August 2017):105–113.

ANDERSON, E. S., THOMPSON, J. A., CROUSE, D. A. et AUSTIN, R. E. (2006). Horizontal resolution and data density effects on remotely sensed LIDAR-based DEM. *Geoderma*, 132(3-4):406–415.

ANDERSON-TEIXEIRA, K. J., DAVIES, S. J., BENNETT, A. C., GONZALEZ-AKRE, E. B., MULLER-LANDAU, H. C., JOSEPH WRIGHT, S., ABU SALIM, K., ALMEYDA ZAMBRANO, A. M., ALONSO, A., BALTZER, J. L. *et al.* (2015). Ctfs-forestgeo : a worldwide network monitoring forests in an era of global change. *Global Change Biology*, 21(2):528–549.

ARCGIS (2016). Creating raster dems and dsms from large lidar point collections.

ASNER, G. P. et MASCARO, J. (2014). Mapping tropical forest carbon : Calibrating plot estimates to a simple LiDAR metric. *Remote Sensing of Environment*, 140:614–624.

ASPRS (2013). Las Specification, Version 1.4 – R13, 15 July 2013. Rapport technique. American Society for Photogrammetry & Remote Sensing.

AURENHAMMER, F. et KLEIN, R. (2000). Chapter 5 - voronoi diagrams*. *In* SACK, J.-R. et URRUTIA, J., éditeurs : *Handbook of Computational Geometry*, pages 201 – 290. North-Holland, Amsterdam.

AXELSSON, P. (1999). Processing of laser scanner data—algorithms and applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54:138–147.

AXELSSON, P. (2000). DEM generation from laser scanner data using adaptive TIN models. *International Archives of Photogrammetry & Remote Sensing.*

BALTSAVIAS, E. (1999). Airborne laser scanning : basic relations and formulas. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54(2-3):199–214.

BARNES, C., BALZTER, H., BARRETT, K., EDDY, J., MILNER, S. et SUÁREZ, J. (2017). Individual Tree Crown Delineation from Airborne Laser Scanning for Diseased Larch Forest Stands. *Remote Sensing*, 9(3):231.

BAUMGARTEN, G. (2010). Doppler rayleigh/mie/raman lidar for wind and temperature measurements in the middle atmosphere up to 80 km. *Atmospheric Measurement Techniques*, 3(6):1509.

BEN-ARIE, J. R., HAY, G. J., POWERS, R. P., CASTILLA, G. et ST-ONGE, B. (2009). Development of a pit filling algorithm for LiDAR canopy height models. *Computers and Geosciences*, 35(9):1940–1949.

BHARATI, M. H., LIU, J. J. et MACGREGOR, J. F. (2004). Image texture analysis : Methods and comparisons. *Chemometrics and Intelligent Laboratory Systems*, 72(1):57–71.

BILODEAU, C. (2010). *Apports du LiDAR à l'étude de la végétation des marais salés de la baie du Mont-Saint-Michel.* Thèse de doctorat, Université Paris-Est.

BLANCHETTE, D., FOURNIER, R. A., LUTHER, J. E. et CÔTÉ, J. F. (2015). Predicting wood fiber attributes using local-scale metrics from terrestrial LiDAR data : A case study of Newfoundland conifer species. *Forest Ecology and Management*, 347:116–129.

BONNET, S., TOROMANOFF, F., BAUWENS, S., MICHEZ, A., DEDRY, L. et LEJEUNE, P. (2013). Principes de base de la télédétection et ses potentialités comme outil de caractérisation de la ressource forestière. partie 2. le lidar aérien. *Forêt Walonne*, 124:28–41.

BONNET, S., TOROMANOFF, F., FOURNEAU, F. et LEJEUNE, P. (2011). Principes de base de la télédétection et ses potentialités comme outil de caractérisation de la ressource forestière. I. images aériennes et satellitaires. *Forêt Walonne*, 114:45–56.

BOUDREAU, J., NELSON, R. F., MARGOLIS, H. A., BEAUDOIN, A., GUINDON, L. et KIMES, D. S. (2008). Regional aboveground forest biomass using airborne and spaceborne LiDAR in Québec. *Remote Sensing of Environment*, 112(10):3876–3890.

BOUDREAULT, L.-é., BECHMANN, A., TARVAINEN, L., KLEMEDTSSON, L., SHENDRYK, I. et DELLWIK, E. (2015). A LiDAR method of canopy structure retrieval for wind modeling of heterogeneous forests. *Agricultural and Forest Meteorology*, 201:86–97.

BOUVIER, M., DURRIEU, S., FOURNIER, R. a. et RENAUD, J.-p. (2015). Generalizing predictive models of forest inventory attributes using an area-based approach with airborne LiDAR data. *Remote Sensing of Environment*, 156:322–334.

BRANDTBERG, T., WARNER, T. A., LANDENBERGER, R. E. et MCGRAW, J. B. (2003). Detection and analysis of individual leaf-off tree crowns in small footprint, high sampling density lidar data from the eastern deciduous forest in North America. *Remote Sensing of Environment*, 85(3):290–303.

BROVELLI, M. A. et LUCCA, S. (2012). Comparison of GRASS-LiDAR modules-TerraScan with respect to vegetation filtering. *Applied Geomatics*, 4(2):123–134.

BUNTING, P., ARMSTON, J., CLEWLEY, D., LUCAS, R. *et al.* (2011). The sorted pulse data software library (spdlib) : Open source tools for processing lidar data. *Proceedings of SilviLaser*.

BUNTING, P., ARMSTON, J., CLEWLEY, D. et LUCAS, R. M. (2013). Sorted pulse data (SPD) library—Part II : A processing framework for LiDAR data from pulsed laser systems in terrestrial environments. *Computers & geosciences*, 56:207–215.

BUTLER, H., GERLEK, M. *et al.* (2016). PDAL - Point Cloud Abstraction Library. http://www.pdal.io/.

CAMPBELL, G. et NORMAN, J. (1990). *The description and measurement of plant canopy structure*, volume 31. Cambridge University Press.

CHASE, A. F., CHASE, D. Z., WEISHAMPEL, J. F., DRAKE, J. B., SHRESTHA, R. L., SLATTON, K. C., AWE, J. J. et CARTER, W. E. (2011). Airborne lidar, archaeology, and the ancient maya landscape at caracol, belize. *Journal of Archaeological Science*, 38(2):387 – 398.

CHASMER, L., HOPKINSON, C., SMITH, B. et TREITZ, P. (2006a). Examining the influence of changing laser pulse repetition frequencies on conifer forest canopy returns. *Photogrammetric Engineering & Remote Sensing*, 72(12):1359–1367.

CHASMER, L., HOPKINSON, C. et TREITZ, P. (2006b). Investigating laser pulse penetration through a conifer canopy by integrating airborne and terrestrial lidar. *Canadian Journal of Remote Sensing*, 32(2):116–125.

CHEHATA, N., GUO, L. et MALLET, C. (2009). Airborne lidar feature selection for urban classification using random forests. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 39(Part 3/W8):207–212.

CHEN, G. et HAY, G. J. (2011). A Support Vector Regression Approach to Estimate Forest Biophysical Parameters at the Object Level Using Airborne Lidar Transects and QuickBird Data. *Photogrammetric Engineering & Remote Sensing*, 77(7):733–741.

CHEN, Q., BALDOCCHI, D., GONG, P. et KELLY, M. (2006). Isolating individual trees in a savanna woodland using small footprint lidar data. *Photogrammetric Engineering & Remote Sensing*, 72(8):923–932.

CHEN, Y., SU, W., LI, J. et SUN, Z. (2009). Hierarchical object oriented classification using very high resolution imagery and lidar data over urban areas. *Advances in Space Research*, 43(7):1101–1110.

CHEN, Z., GAO, B. et DEVEREUX, B. (2017). State-of-the-Art : DTM Generation Using Airborne LIDAR Data. *Sensors*, 17(1).

CLARK, M. L., CLARK, D. B. et ROBERTS, D. A. (2004). Small-footprint lidar estimation of sub-canopy elevation and tree height in a tropical rain forest landscape. *Remote Sensing of Environment*, 91(1):68–89.

COOMES, D. A., DALPONTE, M., JUCKER, T., ASNER, G. P., BANIN, L. F., BURSLEM, D. F., LEWIS, S. L., NILUS, R., PHILLIPS, O., PHUAG, M.-H. et QIEE, L. (2017). Area-based vs tree-centric approaches to mapping forest carbon in Southeast Asian forests with airborne laser scanning data. *Remote Sensing of Environment*, 194:(in press).

COREN, F. et STERZAI, P. (2006). Radiometric correction in laser scanning. *International Journal of Remote Sensing*, 27(15):3097–3104.

DALPONTE, M. (2016). *itcSegment : Individual Tree Crowns Segmentation.* R package version 0.2.

DALPONTE, M. et COOMES, D. A. (2016). Tree-centric mapping of forest carbon density from airborne laser scanning and perspectral data. *Methods in Ecology and Evolution*, 7(10):1236–1245.

DIEDERSHAGEN, O., KOCH, B. et WEINACKER, H. (2004). Automatic segmentation and characterisation of forest stand parameters using airborne lidar data, multispectral and fogis data. *International Archives Of Photogrammetry Remote Sensing And Spatial Information Sciences*, 36:208–212.

DISNEY, M., KALOGIROU, V., LEWIS, P., PRIETO-BLANCO, a., HANCOCK, S. et PFEIFER, M. (2010). Simulating the impact of discrete-return lidar system and survey characteristics over young conifer and broadleaf forests. *Remote Sensing of Environment*, 114(7):1546–1560.

DONOGHUE, D., WATT, P., COX, N. et WILSON, J. (2007). Remote sensing of species mixtures in conifer plantations using LiDAR height and intensity data. *Remote Sensing of Environment*, 110(4):509–522.

EDSON, C. et WING, M. G. (2011). *Airborne light detection and ranging (LiDAR) for individual tree stem location, height, and biomass measurements*, volume 3.

ESTORNELL, J., RUIZ, L. A., VELÁZQUEZ-MARTÍ, B. et HERMOSILLA, T. (2011). Analysis of the factors affecting lidar dtm accuracy in a steep shrub area. *International Journal of Digital Earth*, 4(6):521–538.

EVANS, D. L., ROBERTS, S. D., MCCOMBS, J. W. et HARRINGTON, R. L. (2001). Detection of regularly spaced targets in small-footprint LIDAR data : Research issues for consideration. *Photogrammetric Engineering and Remote Sensing*, 67(October):1133–1136.

EVANS, J. S. et HUDAK, A. T. (2007). A multiscale curvature algorithm for classifying discrete return LiDAR in forested environments. *IEEE Transactions on Geoscience and Remote Sensing*, 45(4):1029–1038.

Evans, J. S., Hudak, A. T., Faux, R. et Smith, A. M. S. (2009). Discrete return lidar in natural resources : Recommendations for project planning, data processing, and deliverables. *Remote Sensing*, 1(4):776–794.

Evette, A., Zanetti, C., Cavaillé, P., Dommanget, F., Mériaux, P. et Vennetier, M. (2014). La gestion paradoxale des ripisylves des cours d'eau de piedmont alpin endigués. gestion sécuritaire ou promotion de la biodiversité. *Journal of Alpine Research| Revue de géographie alpine*, (102-4).

Fernandes, A. M., Utkin, A. B., Lavrov, A. V. et Vilar, R. M. (2004). Development of neural network committee machines for automatic forest fire detection using lidar. *Pattern Recognition*, 37(10):2039–2047.

García, M., Riaño, D., Chuvieco, E., Salas, J. et Danson, F. M. (2011). Multispectral and LiDAR data fusion for fuel type mapping using Support Vector Machine and decision rules. *Remote Sensing of Environment*, 115(6):1369–1379.

García, M., Riaño, D., Chuvieco, E. et Danson, F. M. (2010). Estimating biomass carbon stocks for a Mediterranean forest in central Spain using LiDAR height and intensity data. *Remote Sensing of Environment*, 114(4):816–830.

Gatziolis, D. et Andersen, H.-E. (2008). *A guide to LIDAR data acquisition and processing for the forests of the Pacific Northwest*. US Department of Agriculture, Forest Service, Pacific Northwest Research Station.

Gaudin, S. (1997). L'approche typologique et son utilité en foresterie. note = BTSA Gestion Forestière.

Gaudin, S., Theisen, P. et Vanderheeren, N. (2005). Mieux connaître sa forêt grâce à la typologie des peuplements. Centres Régionaux de la Propriété Forestière (CRPF) de Champagne-Ardenne.

Gaveau, D. L. A. et Hill, R. A. (2003). Quantifying canopy height underestimation by laser pulse penetration in small-footprint airborne laser scanning data. *Canadian Journal of Remote Sensing*, 29(5):650–657.

Gillis, M., Omule, A. et Brierley, T. (2005). Monitoring canada's forests : The national forest inventory. *The Forestry Chronicle*, 81(2):214–221.

Gleason, C. J. et Im, J. (2012). Forest biomass estimation from airborne LiDAR data using machine learning approaches. *Remote Sensing of Environment*, 125:80–91.

Gobakken, T. et Næsset, E. (2008). Assessing effects of laser point density, ground sampling intensity, and field sample plot size on biophysical stand properties derived from airborne laser scanner data. *Canadian Journal of Forest Research*, 38(5):1095–1109.

Gonzalez, P., Asner, G. P., Battles, J. J., Lefsky, M. A., Waring, K. M. et Palace, M. (2010). Forest carbon densities and uncertainties from Lidar, QuickBird, and field measurements in California. *Remote Sensing of Environment*, 114(7):1561–1575.

GOODWIN, N., COOPS, N. et CULVENOR, D. (2007). Development of a simulation model to predict LiDAR interception in forested environments. *Remote Sensing of Environment*, 111(4):481–492.

GOODWIN, N. R., COOPS, N. C. et CULVENOR, D. S. (2006). Assessment of forest structure with airborne LiDAR and the effects of platform altitude. *Remote Sensing of Environment*, 103(2):140–152.

GRAF, R. F., MATHYS, L. et BOLLMANN, K. (2009). Habitat assessment for forest dwelling species using lidar remote sensing : Capercaillie in the alps. *Forest Ecology and Management*, 257(1):160 – 167.

GRASS DEVELOPMENT TEAM (2017). *Geographic Resources Analysis Support System (GRASS GIS) Software, Version 7.2.* Open Source Geospatial Foundation.

GUERRA-HERNÁNDEZ, J., GÖRGENS, E. B., GARCÍA-GUTIÉRREZ, J., RODRIGUEZ, L. C. E., TOMÉ, M. et GONZÁLEZ-FERREIRO, E. (2016). Comparison of ALS based models for estimating aboveground biomass in three types of Mediterranean forest. *European Journal of Remote Sensing*, 49:185–204.

GUPTA, S., WEINACKER, H. et KOCH, B. (2010). Comparative analysis of clustering-based approaches for 3-D single tree detection using airborne fullwave LIDAR data. *Remote Sensing*, 2(4):968–989.

HALL, S. a., BURKE, I. C., BOX, D. O., KAUFMANN, M. R. et STOKER, J. M. (2005). Estimating stand structure using discrete-return lidar : An example from low density, fire prone ponderosa pine forests. *Forest Ecology and Management*, 208:189–209.

HAMRAZ, H., CONTRERAS, M. A. et ZHANG, J. (2016). A robust approach for tree segmentation in deciduous forests using small-footprint airborne LiDAR data. *International Journal of Applied Earth Observation and Geoinformation*, 52:532–541.

HANCOCK, S., ARMSTON, J., LI, Z., GAULTON, R., LEWIS, P., DISNEY, M., DANSON, F. M., STRAHLER, A., SCHAAF, C., ANDERSON, K. et GASTON, K. J. (2015). Waveform lidar over vegetation : An evaluation of inversion methods for estimating return energy. *Remote Sensing Letters*, 164:208–224.

HANSEN, E. H., GOBAKKEN, T. et NÆSSET, E. (2015). Effects of pulse density on digital terrain models and canopy metrics using airborne laser scanning in a tropical rainforest. *Remote Sensing*, 7(7):8453–8468.

HAVERD, V., LOVELL, J., CUNTZ, M., JUPP, D., NEWNHAM, G. et SEA, W. (2012). The canopy semi-analytic pgap and radiative transfer (canspart) model : Formulation and application. *Agricultural and Forest Meteorology*, 160:14 – 35.

HEINZEL, J. et KOCH, B. (2011). Exploring full-waveform LiDAR parameters for tree species classification. *International Journal of Applied Earth Observation and Geoinformation*, 13(1):152–160.

Hijmans, R. J. (2016). *raster : Geographic Data Analysis and Modeling.* R package version 2.5-8.

Hilker, T., Hall, F. G., Coops, N. C., Lyapustin, A., Wang, Y., Nesic, Z., Grant, N., Black, T. A., Wulder, M. A. et Kljun, N. (2010). Remote sensing of photosynthetic light-use efficiency across two forested biomes : Spatial scaling. *Remote Sensing of Environment*, 114(12):2863–2874.

Hill, S., Latifi, H., Heurich, M. et Müller, J. (2017). Individual-tree- and stand-based development following natural disturbance in a heterogeneously structured forest : A LiDAR-based approach. *Ecological Informatics*, 38:12–25.

Hirata, Y. (2004). The effects of footprint size and sampling density in airborne laser scanning to extract individual trees in mountainous terrain. *International Archives for Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(8):102–107.

Höfle, B. et Pfeifer, N. (2007). Correction of laser scanning intensity data : Data and model-driven approaches. *ISPRS Journal of Photogrammetry and Remote Sensing*, 62(6):415–433.

Holmgren, J. (2004). Prediction of tree height, basal area and stem volume in forest stands using airborne laser scanning. *Scandinavian Journal of Forest Research*, 19(6): 543–553.

Holmgren, J., Nilsson, M. et Olsson, H. (2003a). Estimation of tree height and stem volume on plots using airborne laser scanning. *Forest Science*, 49(3):419–428.

Holmgren, J., Nilsson, M. et Olsson, H. (2003b). Simulating the effects of lidar scanning angle for estimation of mean tree height and canopy closure. *Canadian Journal of Remote Sensing*, 29(5):623–632.

Holmgren, J. et Persson, A. s. (2004). Identifying species of individual trees using airborne laser scanner. *Remote Sensing of Environment*, 90(4):415–423.

Hopkinson, C. et Chasmer, L. (2009). Testing LiDAR models of fractional cover across multiple forest ecozones. *Remote Sensing of Environment*, 113(1):275–288.

Huang, H. et Lian, J. (2015). A 3D approach to reconstruct continuous optical images using lidar and MODIS. *Forest Ecosystems*, 2(1):20.

Hudak, A., Ruefenacht, B., Domingo, J. D. et Shrestha, R. (2013). *MCC-LIDAR Software, Version 2.1.*

Hunter, M. O., Keller, M., Victoria, D. et Morton, D. C. (2013). Tree height and tropical forest biomass estimation. *Biogeosciences*, 10(12):8385–8399.

Hutchinson, M. (1993). Development of a continent-wide dem with applications to terrain and climate analysis. *Environmental modeling with GIS*, pages 392–399.

HYYPPÄ, J., HYYPPÄ, H., LECKIE, D., GOUGEON, F., YU, X. et MALTAMO, M. (2008). Review of methods of small-footprint airborne laser scanning for extracting forest inventory data in boreal forests. *International Journal of Remote Sensing*, 29(5):1339–1366.

HYYPPÄ, J. et INKINEN, M. (1999). Detecting and estimating attribute for single trees using laser scanner. *The photogrametric journal of Finland*, 16(2):27–42.

HYYPPÄ, J., KELLE, O., LEHIKOINEN, M. et INKINEN, M. (2001). 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*, 39(5):969–975.

HÄMMERLE, M. et HÖFLE, B. (2014). Effects of reduced terrestrial lidar point density on high-resolution grain crop surface models in precision agriculture. *Sensors*, 14(12): 24212–24230.

HÖFLE, B. et RUTZINGER, M. (2011). Topographic airborne lidar in geomorphology : A technological perspective. 55:1–29.

IOKI, K., IMANISHI, J., SASAKI, T., MORIMOTO, Y. et KITADA, K. (2009). Estimating stand volume in broad-leaved forest using discrete-return LiDAR : plot-based approach. *Landscape and Ecological Engineering*, 6(1):29–36.

IRISH, J. et WHITE, T. (1998). Coastal engineering applications of high-resolution lidar bathymetry. *Coastal Engineering*, 35(1):47 – 71.

ISENBURG, M. (2012). Lasindex – spatial indexing of lidar data. https ://rapidlasso.com/2012/12/03/lasindex-spatial-indexing-of-lidar-data/.

ISENBURG, M. (2013). LASzip : lossless compression of LiDAR data. *Photogrammetric Engineering & Remote Sensing*, 79(2):209–217.

ISENBURG, M. (2015). https://groups.google.com/forum/#!msg/lastools/vgEKeV4peVo/PFdRKQDIJOkJ.

JAKUBOWSKI, M. K., GUO, Q. et KELLY, M. (2013). Tradeoffs between lidar pulse density and forest measurement accuracy. *Remote Sensing of Environment*, 130:245–253.

JENNESS, J. S. (2004). Calculating landscape surface area from digital elevation models. *Wildlife Society bulletin*, 32(3):829–839.

JING, L., HU, B., LI, J. et NOLAND, T. (2012). Automated Delineation of Individual Tree Crowns from Lidar Data by Multi-Scale Analysis and Segmentation. *Photogrammetric Engineering & Remote Sensing*, 78(12):1275–1284.

JUCKER, T., CASPERSEN, J., CHAVE, J., ANTIN, C., BARBIER, N., BONGERS, F., DALPONTE, M., van EWIJK, K. Y., FORRESTER, D. I., HAENI, M., HIGGINS, S. I., HOLDAWAY, R. J., IIDA, Y., LORIMER, C., MARSHALL, P. L., MOMO, S., MONCRIEFF, G. R., PLOTON, P., POORTER, L., RAHMAN, K. A., SCHLUND, M., SONKÉ, B., STERCK, F. J., TRUGMAN, A. T., USOLTSEV, V. A., VANDERWEL, M. C., WALDNER, P., WEDEUX, B. M., WIRTH, C., WÖLL, H., WOODS, M., XIANG, W., ZIMMERMANN, N. E. et COOMES, D. A. (2017). Allometric equations for integrating remote sensing imagery into forest monitoring programmes. *Global Change Biology*, 23(1):177–190.

JUNG, S.-E., KWAK, D.-A., PARK, T., LEE, W.-K. et YOO, S. (2011). Estimating Crown Variables of Individual Trees Using Airborne and Terrestrial Laser Scanners. *Remote Sensing*, 3(12):2346–2363.

KAMPA, K. et SLATTON, K. C. (2004). An adaptive multiscale filter for segmenting vegetation in alsm data. *In Geoscience and Remote Sensing Symposium, 2004. IGARSS'04. Proceedings. 2004 IEEE International*, volume 6, pages 3837–3840. IEEE.

KANE, V. R., GILLESPIE, A. R., MCGAUGHEY, R., LUTZ, J. A., CEDER, K. et FRANKLIN, J. F. (2008). Interpretation and topographic compensation of conifer canopy self-shadowing. *Remote Sensing of Environment*, 112(10):3820–3832.

KANE, V. R., MCGAUGHEY, R. J., BAKKER, J. D., GERSONDE, R. F., LUTZ, J. a. et FRANKLIN, J. F. (2010). Comparisons between field- and LiDAR-based measures of stand structural complexity. *Canadian Journal of Forest Research*, 40(4):761–773.

KHOSRAVIPOUR, A., SKIDMORE, A. K. et ISENBURG, M. (2016). Generating spike-free Digital Surface Models using raw LiDAR point clouds : a new approach for forestry applications. *International Journal of Applied Earth Observation and Geoinformation*, 52:104–114.

KHOSRAVIPOUR, A., SKIDMORE, A. K., ISENBURG, M., WANG, T. et HUSSIN, Y. A. (2014). Generating Pit-free Canopy Height Models from Airborne Lidar. *Photogrammetric Engineering & Remote Sensing*, 80(9):863–872.

KHOSRAVIPOUR, A., SKIDMORE, A. K., WANG, T., ISENBURG, M. et KHOSHELHAM, K. (2015). Effect of slope on treetop detection using a LiDAR Canopy Height Model. *ISPRS Journal of Photogrammetry and Remote Sensing*, 104.

KLEINER, L., ROBRA, J. P., GILLIÉRON, P.-Y., SCHAER, P. et MERTINA, C. (2010). Lever de limites naturelles par scanner laser aérien (LIDAR) Evaluation et perspectives dans le cadre de la mensuration cadastrale. *Géomatique Suisse*, 4(EPFL-ARTICLE-148250):136–139.

KOBAYASHI, K. et SUGIHARA, K. (2002). Crystal voronoi diagram and its applications. *Future Generation Computer Systems*, 18(5):681 – 692. ICCS2001.

KOBLER, A., PFEIFER, N., OGRINC, P., TODOROVSKI, L., OŠTIR, K. et DŽEROSKI, S. (2007). Repetitive interpolation : A robust algorithm for DTM generation from Aerial Laser Scanner Data in forested terrain. *Remote Sensing of Environment*, 108(1):9–23.

KOCH, B. (2010). Status and future of laser scanning, synthetic aperture radar and hyperspectral remote sensing data for forest biomass assessment. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(6):581–590.

KOCH, B., HEYDER, U. et WEINACKER, H. (2006). Detection of Individual Tree Crowns in Airborne Lidar Data. *Photogrammetric Engineering & Remote Sensing*, 72(4):357–363.

KOETZ, B., MORSDORF, F., van der LINDEN, S., CURT, T. et ALLGÖWER, B. (2008). Multi-source land cover classification for forest fire management based on imaging spectrometry and LiDAR data. *Forest Ecology and Management*, 256(3):263–271.

KORPELA, I., ØRKA, H. O., MALTAMO, M., TOKOLA, T. et HYYPPÄ, J. (2010). Tree species classification using airborne LiDAR – effects of stand and tree parameters, downsizing of training set, intensity normalization, and sensor type. *Silva Fennica*, 44(2):319–339.

KORPELA, I., TOKOLA, T., ØRKA, H. et KOSKINEN, M. (2009). Small-footprint discrete-return LIDAR in tree species recognition. *In Proceedings of the ISPRS*, numéro 2004, pages 2–5.

KRAUS, K. et MIKHAIL, E. M. (1972). Linear Least-Squares Interpolation. *In Photogrammetric Engineering*, volume 38, pages 1016–1029.

KRAUS, K. et PFEIFER, N. (1998). Determination of terrain models in wooded areas with airborne laser scanner data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 53(4):193–203.

KUKKO, A., KAASALAINEN, S. et LITKEY, P. (2008). Effect of incidence angle on laser scanner intensity and surface data. *Applied optics*, 47:986–992.

KWAK, D.-A., LEE, W.-K., CHO, H.-K., LEE, S.-H., SON, Y., KAFATOS, M. et KIM, S.-R. (2010). Estimating stem volume and biomass of Pinus koraiensis using LiDAR data. *Journal of plant research*, 123(4):421–32.

KWAK, D.-A., LEE, W.-K., LEE, J.-H., BIGING, G. S. et GONG, P. (2007). Detection of individual trees and estimation of tree height using LiDAR data. *Journal of Forest Research*, 12(6): 425–434.

LAASASENAHO, J. (1982). Taper curve and volume functions for pine, spruce and birch [pinus sylvestris, picea abies, betula pendula, betula pubescens]. *Communicationes Instituti Forestalis Fenniae (Finland).*

LASERDATA GMBH (2017). Laserdata gmbh. `https://www.laserdata.at/index.html`.

LEE, H., SLATTON, K. C., ROTH, B. E. et CROPPER, W. P. (2010). Adaptive clustering of airborne LiDAR data to segment individual tree crowns in managed pine forests. *International Journal of Remote Sensing*, 31(1):117–139.

LEITERER, R., FURRER, R., SCHAEPMAN, M. E. et MORSDORF, F. (2015). Forest canopy-structure characterization : A data-driven approach. *Forest Ecology and Management*, 358:48–61.

LI, W., GUO, Q., JAKUBOWSKI, M. K. et KELLY, M. (2012). A New Method for Segmenting Individual Trees from the Lidar Point Cloud. *Photogrammetric Engineering & Remote Sensing*, 78(1):75–84.

LI, X. et STRAHLER, A. (1985). Geometric-optimal modeling of a conifer forest canopy. *IEEE Transactions on Geoscience and Remote Sensing*, 23(5):705–721.

LIANG, X., HYYPPÄ, J. et MATIKAINEN, L. (2007). Deciduous-coniferous tree classification using difference between first and last pulse laser signatures. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVI(Part3/W52): 253–257.

LIM, K., HOPKINSON, C. et TREITZ, P. (2008). Examining the effects of sampling point densities. *Forestry Chronicle*, 84(6):876–885.

LIM, K., TREITZ, P., BALDWIN, K., MORRISON, I. et GREEN, J. (2014). Lidar remote sensing of biophysical properties of tolerant northern hardwood forests. *Canadian Journal of Remote Sensing*, 29(5):658–678.

LIM, K., TREITZ, P., WULDER, M., ST-ONGE, B. et FLOOD, M. (2003). LiDAR remote sensing of forest structure. *Progress in Physical Geography*, 27(1):88–106.

LIN, X. et ZHANG, J. (2014). Segmentation-based filtering of airborne LiDAR point clouds by progressive densification of terrain segments. *Remote Sensing*, 6(2):1294–1326.

LIU, X. et DENG, Z. (2015). *A Graph-Based Nonparametric Drivable Road Region Segmentation Approach for Driverless Car Based on LIDAR Data*, pages 431–439. Springer Berlin Heidelberg, Berlin, Heidelberg.

LLOYD, C. D. et ATKINSON, P. M. (2010). Deriving DSMs from LiDAR data with kriging. *International Journal of Remote Sensing*, 23(May 2012):2519–2524.

LOVELL, J., JUPP, D., NEWNHAM, G., COOPS, N. et CULVENOR, D. (2005). Simulation study for finding optimal lidar acquisition parameters for forest height retrieval. *Forest Ecology and Management*, 214(1-3):398–412.

LUTHER, J. E., SKINNER, R., FOURNIER, R. A., VAN LIER, O. R., BOWERS, W. W., COTÉ, J. F., HOPKINSON, C. et MOULTON, T. (2014). Predicting wood quantity and quality attributes of balsam fir and black spruce using airborne laser scanner data. *Forestry*, 87(2):313–326.

MAGNUSSEN, S., NÆSSET, E. et GOBAKKEN, T. (2010). Reliability of LiDAR derived predictors of forest inventory attributes : A case study with Norway spruce. *Remote Sensing of Environment*, 114(4):700–712.

MAGNUSSON, M., FRANSSON, J. E. S. et HOLMGREN, J. (2007). Effects on Estimation Accuracy of Forest Variables Using Different Pulse Dens ... *Forest Science*, 53(6):619–626.

MALLET, C., CHAUVE, A. et BRETAR, F. (2008). Analyse et traitement d'ondes lidar pour la cartographie et la reconnaissance de formes : application au milieu urbain. *In Reconnaissance de Formes et Intelligence Artificielle (RFIA)*, pages 693–702.

MALTAMO, M., EERIKÄINEN, K. et PITKÄNEN (2004). Estimation of timber volume and stem density based on scanning laser altimetry and expected tree size distribution functions. *Remote Sensing of Environment*, 90(3):319–330.

MALTAMO, M., NÆSSET, E. et VAUHKONEN, J. (2014). *Forestry Applications of Airborne Laser Scanning : Concepts and Case Studies.* Springer.

MCGAUGHEY, R. J. (2015). *FUSION/LDV : Software for LIDAR Data Analysis and Visualization.*

MEI, C. et DURRIEU, S. (2004). Tree crown delineation from digital elevation models and high resolution imagery. *Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36:3–6.

MITAS, L. et MITASOVA, H. (1999). Spatial interpolation. *Geographical information systems : principles, techniques, management and applications*, 1:481–492.

MOFFIET, T., MENGERSEN, K., WITTE, C., KING, R. et DENHAM, R. (2005). Airborne laser scanning : Exploratory data analysis indicates potential variables for classification of individual trees or forest stands according to species. *ISPRS Journal of Photogrammetry and Remote Sensing*, 59(5):289–309.

MONGUS, D. et ŽALIK, B. (2011). Efficient method for lossless LIDAR data compression. *International Journal of Remote Sensing*, 32(9):2507–2518.

MONTAGHI, A. (2013). Effect of scanning angle on vegetation metrics derived from a nationwide Airborne Laser Scanning acquisition. *Canadian Journal of Remote Sensing*, 39(sup1):S152–S173.

MONTEALEGRE, A. L., LAMELAS, M. T. et DE LA RIVA, J. (2015a). A Comparison of Open-Source LiDAR Filtering Algorithms in a Mediterranean Forest Environment. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(8):4072–4085.

MONTEALEGRE, A. L., LAMELAS, M. T. et DE LA RIVA, J. (2015b). Interpolation routines assessment in ALS-derived Digital Elevation Models for forestry applications. *Remote Sensing*, 7(7):8631–8654.

MORA, B., WULDER, M. A., WHITE, J. C. et HOBART, G. (2013). Modeling stand height, volume, and biomass from very high spatial resolution satellite imagery and samples of airborne LIDAR. *Remote Sensing*, 5(5):2308–2326.

MORSDORF, F., FREY, O., MEIER, E., ITTEN, K. I. et ALLGÖWER, B. (2008). Assessment of the influence of flying altitude and scan angle on biophysical vegetation products derived from airborne laser scanning. *International Journal of Remote Sensing*, 29(5):1387–1406.

MORSDORF, F., MEIER, E., KÖTZ, B., ITTEN, K. I., DOBBERTIN, M. et ALLGÖWER, B. (2004). LIDAR-based geometric reconstruction of boreal type forest stands at single tree level for forest and wildland fire management. *Remote Sensing of Environment*, 92(3):353–362.

Næsset, E. (1997). Determination of mean tree height of forest stands using airborne laser scanner data. *{ISPRS} Journal of Photogrammetry and Remote Sensing*, 52(2):49–56.

Næsset, E. (2004a). Accuracy of forest inventory using airborne laser scanning : evaluating the first nordic full-scale operational project. *Scandinavian Journal of Forest Research*, 19(6):554–557.

Næsset, E. (2004b). Effects of different flying altitudes on biophysical stand properties estimated from canopy height and density measured with a small-footprint airborne scanning laser. *Remote Sensing of Environment*, 91:243–255.

Næsset, E. (2005). Assessing sensor effects and effects of leaf-off and leaf-on canopy conditions on biophysical stand properties derived from small-footprint airborne laser data. *Remote Sensing of Environment*, 98:356–370.

Næsset, E. (2009). Effects of different sensors, flying altitudes, and pulse repetition frequencies on forest canopy metrics and biophysical stand properties derived from small-footprint airborne laser data. *Remote Sensing of Environment*, 113(1):148–159.

Næsset, E. et Økland, T. (2002). Estimating tree height and tree crown properties using airborne scanning laser in a boreal nature reserve. *Remote Sensing of Environment*, 79(1):105–115.

Nelson, R. (2013). How did we get here ? An early history of forestry lidar. *Canadian Journal of Remote Sensing*, 39(S1):S6–S17.

Niemi, M. et Vauhkonen, J. (2016). Extracting Canopy Surface Texture from Airborne Laser Scanning Data for the Supervised and Unsupervised Prediction of Area-Based Forest Characteristics. *Remote Sensing*, 8(7):582.

Nilson, T. (1971). A theoretical analysis of the frequency of gaps in plant stands. *Agricultural meteorology*, 8:25–38.

Nilsson, M. (1996). Estimation of tree heights and stand volume using an airborne lidar system. *Remote Sensing of Environment*, 56(1):1–7.

Northend, C. A., Honey, R. C. et Evans, W. E. (1966). Laser radar (lidar) for meteorological observations. *Review of Scientific Instruments*, 37(4):393–400.

Ørka, H. O., Næsset, E. et Bollandsås, O. M. (2009). Classifying species of individual trees by intensity and structure features derived from airborne laser scanner data. *Remote Sensing of Environment*, 113(6):1163–1174.

Parker, G. G., Harmon, M. E., Lefsky, M. A., Chen, J., Pelt, R. V., Weis, S. B., Thomas, S. C., Winner, W. E., Shaw, D. C. et Frankling, J. F. (2004). Three-dimensional Structure of an Old-growth Pseudotsuga-Tsuga Canopy and Its Implications for Radiation Balance, Microclimate, and Gas Exchange. *Ecosystems*, 7(5).

PASCUAL, C., GARCÍA-ABRIL, A., GARCÍA-MONTERO, L., MARTÍN-FERNÁNDEZ, S. et COHEN, W. (2008). Object-based semi-automatic approach for forest structure characterization using lidar data in heterogeneous Pinus sylvestris stands. *Forest Ecology and Management*, 255(11):3677–3685.

PÉREZ-GARCÍA, J. L., DELGADO, J., CARDENAL, J., COLOMO, C. et UREÑA, M. a. (2012). Progressive Densification and Region Growing Methods for Lidar Data Classification. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXIX-B3(September):155–160.

PIPPURI, I., KALLIO, E., MALTAMO, M., PELTOLA, H. et PACKALÉN, P. (2012). Exploring horizontal area-based metrics to discriminate the spatial pattern of trees and need for first thinning using airborne laser scanning. *Forestry*, 85(2):305–314.

PIROTTI, F., GUARNIERI, A. et VETTORE, A. (2008). Neural network and quad-tree approach to extract tree position and height from LiDAR data. *In Proceedings of SilviLaser 2008, 8th international conference on LiDAR applications in forest assessment and inventory*, pages 537–543, Heriot-Watt University.

PIROTTI, F., GUARNIERI, A. et VETTORE, A. (2013). Vegetation filtering of waveform terrestrial laser scanner data for DTM production. *Applied Geomatics*, 5(4):311–322.

PIROTTI, F. et TAROLLI, P. (2010). Suitability of LiDAR point density and derived landform curvature maps for channel network extraction. *Hydrological Processes*, 24(9):1187–1197.

PLOWRIGHT, A. (2017). *ForestTools : Analyzing Remotely Sensed Forest Data*. R package version 0.1.5.

POPESCU, S. C. (2007). Estimating biomass of individual pine trees using airborne lidar. *Biomass and Bioenergy*, 31(9):646–655.

POPESCU, S. C., WYNNE, R. H. et NELSON, R. F. (2000). Estimating forest vegetation biomass using airborne lidar measurements. *In Second International Conference on Geospatial Information in Agriculture and Forestry*, Lake Buena Vista.

POPESCU, S. C., WYNNE, R. H. et NELSON, R. F. (2002). Estimating plot-level tree heights with lidar : local filtering with a canopy-height based variable window size. *Computers and Electronics in Agriculture*, 37(1-3):71–95.

POPESCU, S. C. et ZHAO, K. (2008). A voxel-based lidar method for estimating crown base height for deciduous and pine trees. *Remote Sensing of Environment*, 112(3):767–781.

POPESCU, S. C., ZHAO, K., NEUENSCHWANDER, A. et LIN, C. (2011). Satellite lidar vs. small footprint airborne lidar : Comparing the accuracy of aboveground biomass estimates and forest structure metrics at footprint level. *Remote Sensing of Environment*, 115(11): 2786–2797.

Poullain, E. (2013). *Exploitation de l'intensité du signal LASER d'un LiDAR topographique aéroporté pour des environnements littoraux sableux.* Thèse de doctorat, Université de Caen Basse-Normandie.

Pradhan, B., Kumar, S., Mansor, S., Ramli, A. R. et Sharif, A. (2005). Light detection and ranging (LIDAR) data compression. *KMITL Journal of Science and Technology*, 5(3):515–523.

Pyysalo, U. et Hyyppä, H. (2002). Reconstructing tree crowns from laser scanner data for feature extraction. *International Archives Of Photogrammetry Remote Sensing And Spatial Information Sciences*, 34:218–221.

R Core Team (2015). *R : A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria.

Racine, E. B., Coops, N. C., St-Onge, B. et Bégin, J. (2014). Estimating Forest Stand Age from LiDAR-Derived Predictors and Nearest Neighbor Imputation. *Forest Science*, 60(1): 128–136.

Reitberger, J., Krzystek, P. et Stilla, U. (2006). Analysis of full waveform lidar data for tree species classification. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36:228–233.

Reitberger, J., Krzystek, P. et Stilla, U. (2008). Analysis of full waveform LIDAR data for the classification of deciduous and coniferous trees. *International Journal of Remote Sensing*, 29(5):1407–1431.

Reitberger, J., Schnörr, C., Krzystek, P. et Stilla, U. (2009). 3D segmentation of single trees exploiting full waveform LIDAR data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 64(6):561–574.

Roussel, J.-R. (2017). *rlas : Read and Write 'las' and 'laz' Binary File Formats Used for Remote Sensing Data.* R package version 1.1.6.

Roussel, J.-R. et Auty, D. (2017). *lidR : Airborne LiDAR Data Manipulation and Visualization for Forestry Applications.* R package version 1.2.1.

Roussel, J.-R., Caspersen, J., Béland, M. et Achim, A. (2018). A mathematical framework to describe the effect of beam incidence angle on metrics derived from airborne LiDAR : the case of forest canopies approaching turbid medium behaviour. *Remote Sensing of Environment.*

Roussel, J.-R., Caspersen, J., Béland, M., Thomas, S. et Achim, A. (2017). Removing bias from LiDAR-based estimates of canopy height : Accounting for the effects of pulse density and footprint size. *Remote Sensing of Environment*, 198:1–16.

Ruiz, L. A., Hermosilla, T., Mauro, F. et Godino, M. (2014). Analysis of the influence of plot size and LiDAR density on forest structure attribute estimates. *Forests*, 5(5):936–951.

RUSU, R. B. et COUSINS, S. (2011). 3D is here : Point Cloud Library (PCL). *In IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China.

SADEGHI, Y., ST-ONGE, B., LEBLON, B. et SIMARD, M. (2015). Canopy Height Model (CHM) Derived From a TanDEM-X InSAR DSM and an Airborne Lidar DTM in Boreal Forest. *IEEE Journal of selected topics in applied earth observation and remote sensing*.

SCHNÜRMACHER, M., GÖHRING, D., WANG, M. et GANJINEH, T. (2013). *High Level Sensor Data Fusion of Radar and Lidar for Car-Following on Highways*, pages 217–230. Springer Berlin Heidelberg, Berlin, Heidelberg.

SEIDL, R., SPIES, T. A., RAMMER, W., STEEL, E. A., PABST, R. J. et OLSEN, K. (2012). Multi-scale Drivers of Spatial Variation in Old-Growth Forest Carbon Density Disentangled with Lidar and an Individual-Based Landscape Model. *Ecosystems*, 15(8):1321–1335.

SHI, J. et MALIK, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.

SILVA, C. A., CROOKSTON, N. L., HUDAK, A. T., VIERLING, L. A., KLAUBERG, C. et CARDIL, A. (2017). *rLiDAR : LiDAR Data Processing and Visualization.* R package version 0.1.1.

SILVA, C. A., HUDAK, A. T., VIERLING, L. A., LOUDERMILK, E. L., O'BRIEN, J. J., HIERS, J. K., JACK, S. B., GONZALEZ-BENECKE, C., LEE, H., FALKOWSKI, M. J. et KHOSRAVIPOUR, A. (2016). Imputation of Individual Longleaf Pine (Pinus palustris Mill.) Tree Attributes from Field and LiDAR Data. *Canadian Journal of Remote Sensing*, 42(5):554–573.

SINGH, K. K., CHEN, G., MCCARTER, J. B. et MEENTEMEYER, R. K. (2015). Effects of LiDAR point density and landscape context on estimates of urban forest biomass. *ISPRS Journal of Photogrammetry and Remote Sensing*, 101:310–322.

SMITH, A. M. S., FALKOWSKI, M. J., HUDAK, A. T., EVANS, J. S., ROBINSON, A. P. et STEELE, C. M. (2009). A cross-comparison of field, spectral, and lidar estimates of forest canopy cover. *Canadian Journal of Remote Sensing*, 35(5):447–459.

SOININEN, A. (2016). *TerraScan User's Guide*. http://www.terrasolid.com/download/tscan.pdf.

SOLBERG, S., NÆSSET, E. et BOLLANDSÅS, O. M. (2006). Single tree segmentation using airborne laser scanner data in a structurally heterogeneous spruce forest. *Photogrammetric Engineering & Remote Sensing*, 72(12):1369–1378.

SPRIGGS, R. A., VANDERWEL, M. C., JONES, T. A., CASPERSEN, J. P. et DAVID, A. (2015). A simple area-based model for predicting airborne LiDAR first returns from stem diameter distributions : an example study in an uneven aged , mixed temperate forest. *Canadian Journal of Forest Research*, 1350(June):1–42.

STEREŃCZAK, K., CIESIELSKI, M., BAŁAZY, R. et ZAWIŁA-NIEDŹWIECKI, T. (2016). Comparison of various algorithms for DTM interpolation from LIDAR data in dense mountain forests. *European Journal of Remote Sensing*, 49(October):599–621.

Strahler, A. H. et Jupp, D. L. B. (1990). Modeling bidirectional reflectance of forests and woodlands using boolean models and geometric optics. *Remote Sensing of Environment*, 34(3):153–166.

Sumnall, M. J., Hill, R. A. et Hinsley, S. A. (2016). Comparison of small-footprint discrete return and full waveform airborne lidar data for estimating multiple forest variables. *Remote Sensing of Environment*, 173:214–223.

Sutton, R. N. et Hall, E. L. (1972). Texture measures for automatic classification of pulmonary disease. *IEEE Trans. Comput.*, 21(7):667–676.

Tao, S., Guo, Q., Li, L., Xue, B., Kelly, M., Li, W., Xu, G. et Su, Y. (2014). Airborne Lidar-derived volume metrics for aboveground biomass estimation : A comparative assessment for conifer stands. *Agricultural and Forest Meteorology*, 198-199(September 2014):24–32.

Thomas, V., Treitz, P., McCaughey, J. H. et Morrison, I. (2006). Mapping stand-level forest biophysical variables for a mixedwood boreal forest using lidar : an examination of scanning density. *Canadian Journal of Forest Research*, 36(1):34–47.

Tinkham, W. T., Smith, A. M., Hoffman, C., Hudak, A. T., Falkowski, M. J., Swanson, M. E. et Gessler, P. E. (2012). Investigating the influence of LiDAR ground surface errors on the utility of derived forest inventories. *Canadian Journal of Forest Research*, 42(3):413–422.

Tompalski, P., Coops, N., White, J. et Wulder, M. (2016). Enhancing Forest Growth and Yield Predictions with Airborne Laser Scanning Data : Increasing Spatial Detail and Optimizing Yield Curve Selection through Template Matching. *Forests*, 7(12):255.

Uysal, M. et Polat, N. (2014). Investigating performance of airborne lidar data filtering with triangular irregular network (TIN) algorithm. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 40(7):199–202.

van Ewijk, K. Y., Treitz, P. M. et Scott, N. A. (2011). Characterizing Forest Succession in Central Ontario using Lidar-derived Indices. *Photogrammetric Engineering and Remote Sensing*, 77(3):261–269.

Van Leeuwen, M., Coops, N. C. et Wulder, M. a. (2010). Canopy surface reconstruction from a LiDAR point cloud using Hough transform. *Remote Sensing Letters*, 1(3):125–132.

Van Leeuwen, M. et Nieuwenhuis, M. (2010). Retrieval of forest structural parameters using LiDAR remote sensing. *European Journal of Forest Research*, 129(4):749–770.

Vaughn, N. R., Moskal, L. M. et Turnblom, E. C. (2011). Fourier transformation of waveform Lidar for species recognition. *Remote Sensing Letters*, 2(4):347–356.

Vauhkonen, J., Maltamo, M., McRoberts, R. E. et Næsset, E. (2014). *Introduction to Forestry Applications of Airborne Laser Scanning*, pages 1–16. Springer Netherlands, Dordrecht.

VÉGA, C. et DURRIEU, S. (2011). 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*, 13(4):646–656.

VEGA, C., HAMROUNI, a., EL MOKHTARI, S., MOREL, J., BOCK, J., RENAUD, J.-P., BOUVIER, M. et DURRIEU, S. (2014). PTrees : A point-based approach to forest tree extraction from lidar data. *International Journal of Applied Earth Observation and Geoinformation*, 33: 98–108.

VÉGA, C., RENAUD, J. P., DURRIEU, S. et BOUVIER, M. (2016). On the interest of penetration depth, canopy area and volume metrics to improve Lidar-based models of forest parameters. *Remote Sensing of Environment*, 175:32–42.

VOSSELMAN, G. (2000). Slope based filtering of laser altimetry data. *International Archives of Photogrammetry and Remote Sensing, Vol. 33, Part B3/2*, 33(Part B3/2):678–684.

WANG, Y., WEINACKER, H. et KOCH, B. (2008). A Lidar point cloud based procedure for vertical canopy structure analysis and 3D single tree modelling in forest. *Sensors*, 8(6): 3938–3951.

WATT, M. S., ADAMS, T., ARACIL, S. G., MARSHALL, H. et WATT, P. (2013). The influence of LiDAR pulse density and plot size on the accuracy of New Zealand plantation stand volume equations. *New Zealand Journal of Forestry Science*, 43:1–10.

WATT, P. et WILSON, J. (2005). Using airborne light detection and ranging (lidar) to identify and monitor the performance of plantation species mixture. Borås, Sweden.

WEBER, T. C. et BOSS, D. E. (2009). Use of LiDAR and supplemental data to estimate forest maturity in Charles County, MD, USA. *Forest Ecology and Management*, 258(9):2068–2075.

WEINACKER, H., KOCH, B. et WEINACKER, R. (2004). Treesvis-a software system for simultaneous 3d-real-time visualisation of dtm, dsm, laser raw data, multispectral data, simple tree and building models. *International Archives of ...*, pages 90–95.

WENGE, N., XIAOWEN, L., WOODCOCK, C. E., ROUJEAN, J.-L. et DAVIS, R. E. (1997). Transmission of solar radiation in boreal conifer forests : Measurements and models. *Journal of Geophysical Research*, 102:29555—-29566.

WHITE, J. C., WULDER, M. A., VARHOLA, A., MIKKO, V., COOPS, N. C., COOK, B. D., PITT, D. et WOODS, M. (2013). Best practices for generating forest inventory attributes from airborne laser scanning data using the area-based approach Best Practices Guide. *The*, 89(6):722–723.

WICKHAM, H. (2014). Tidy Data. *Journal of Statistical Software*, 59(10).

WING, B. M., RITCHIE, M. W., BOSTON, K., COHEN, W. B. et OLSEN, M. J. (2015). Individual snag detection using neighborhood attribute filtered airborne lidar data. *Remote Sensing of Environment*, 163:165–179.

WOODS, M., LIM, K. et TREITZ, P. (2008). Predicting forest stand variables from LiDAR data in the Great Lakes - St. Lawrence forest of Ontario. *Forestry Chronicle*, 84(6):827–839.

WOODS, M., PITT, D., PENNER, M., LIM, K., NESBITT, D., ETHERIDGE, D. et TREITZ, P. (2011). Operational implementation of a lidar inventory in boreal ontario. *The Forestry Chronicle*, 87(4):512–528.

YANG, R. C., KOZAK, A. et SMITH, J. H. G. (1978). The potential of weibull-type functions as flexible growth curves. *Canadian Journal of Forest Research*, 8(4):424–431.

YAO, W., KRZYSTEK, P. et HEURICH, M. (2012). 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*, 123:368–380.

YU, B., LIU, H., WU, J., HU, Y. et ZHANG, L. (2010). Automated derivation of urban building density information using airborne lidar data and object-based method. *Landscape and Urban Planning*, 98(3):210 – 219. Climate Change and Spatial Planning.

YU, X., HYYPPÄ, J., HYYPPÄ, H. et MALTAMO, M. (2004). Effects of flight altitude on tree height estimation using airborne laser scanning. *In International Archives Of Photogrammetry Remote Sensing And Spatial Information Sciences*, volume XXXVI, pages 96–101.

ZELLWEGER, F., MORSDORF, F., PURVES, R. S., BRAUNISCH, V. et BOLLMANN, K. (2013). Improved methods for measuring forest landscape structure : LiDAR complements field-based habitat assessment. *Biodiversity and Conservation*, 23(2):289–307.

ZHANG, J. X., WU, J. Q., CHANG, K., ELLIOT, W. J. et DUN, S. (2009). Effects of DEM source and resolution on WEPP hydrologic and erosion simulation : a case study of two forest watersheds in Northern Idaho. *American Society of Agricultural and Biological Engineers*, 52(2):447–457.

ZHANG, K., CHEN, S. C., WHITMAN, D., SHYU, M. L., YAN, J. et ZHANG, C. (2003). A progressive morphological filter for removing nonground measurements from airborne LIDAR data. *IEEE Transactions on Geoscience and Remote Sensing*, 41(4 PART I):872–882.

ZHANG, K. et WHITMAN, D. (2005). Comparison of Three Algorithms for Filtering Airborne Lidar Data. *Photogrammetric Engineering Remote Sensing*, 71(3):313–324.

ZHANG, Z. et LIU, X. (2013). Support vector machines for tree species identification using LiDAR-derived structure and intensity variables. *Geocarto International*, 28(4):364–378.

ZHAO, K., POPESCU, S., MENG, X., PANG, Y. et AGCA, M. (2011). Characterizing forest canopy structure with lidar composite metrics and machine learning. *Remote Sensing of Environment*, 115(8):1978–1996.

ZHAO, K., POPESCU, S. C. et NELSON, R. F. (2009). Lidar remote sensing of forest biomass : A scale-invariant estimation approach using airborne lasers. *Remote Sensing of Environment*, 113(1):182–196.

ZHAO, X., GUO, Q., SU, Y. et XUE, B. (2016). Improved progressive TIN densification filtering algorithm for airborne LiDAR data in forested areas. *ISPRS Journal of Photogrammetry and Remote Sensing*, 117:79–91.

ZHEN, Z., QUACKENBUSH, L. J., STEHMAN, S. V. et ZHANG, L. (2015). Agent-based region growing for individual tree crown delineation from airborne laser scanning (ALS) data. *International Journal of Remote Sensing*, 36(7):1965–1993.

ZHEN, Z., QUACKENBUSH, L. J. et ZHANG, L. (2013). Impact of tree-oriented growth order in marker-controlled region growing for individual tree crown delineation using airborne laser scanner (ALS) data. *Remote Sensing*, 6(1):555–579.