

# 12

## Aplicaciones del escaneado láser terrestre en ciencias forestales

**Antonio J. ARIZA SALAMANCA**  
**Juan Alberto MOLINA VALERO**

### Resumen

La aplicación del láser escáner o LiDAR terrestre en silvicultura y ecología forestal es cada vez más popular y efectiva debido al grado de detalle que esta tecnología puede proporcionar. De hecho, ya existen numerosos estudios sobre su uso para proporcionar estimaciones precisas en el ámbito forestal. La mayor parte de los esfuerzos se han dirigido a su utilización en inventarios forestales a nivel de parcela para la obtención de parámetros básicos, como los diámetros y las alturas de las plantas. Sin embargo, su versatilidad y precisión han propiciado la innovación en el campo, llevando al desarrollo de algoritmos que permiten caracterizaciones más complejas e inventarios de mayor detalle. Estas estimaciones se consiguen a partir de cálculos métricos obtenidos del modelado de cortes o secciones de la nube de puntos registrada por el láser escáner. No obstante, el reto sigue siendo la extracción de estos parámetros de forma correcta a partir de los datos en bruto de TLS. En los últimos años se han desarrollado metodologías que pretenden automatizar las fases de adquisición y procesado de datos TLS. El presente capítulo constituye una guía teórico-práctica para la extracción de parámetros de interés forestal a partir de nubes de puntos de LiDAR terrestre, con aplicación en la silvicultura y en la ecología forestal. La primera parte del capítulo se centra en los fundamentos del escaneado láser terrestre. Seguidamente se describen las aplicaciones en silvicultura y ecología forestal. El capítulo termina con un desarrollo de la cadena de procesos que tiene lugar para transformar la nube de puntos en el producto final. Se acompaña este capítulo de un tutorial en R del paquete FORTLS para la extracción de parámetros forestales a partir de una nube de puntos de LiDAR terrestre.

**Palabras clave:** láser escáner terrestre, segmentación, ajuste geométrico, voxelización, FORTLS.

## 1. Introducción

El escaneado láser terrestre o LiDAR terrestre (TLS, en adelante; acrónimo de *Terrestrial Laser Scanning*) es un dispositivo de precisión muy elevada, capaz de trabajar en diferentes entornos y condiciones atmosféricas. Esta versatilidad está permitiendo su uso en múltiples aplicaciones en diversos campos de la ciencia y la tecnología; desde la ingeniería civil y la conservación del patrimonio cultural, hasta la industria del entretenimiento; destacando su gran potencial en el ámbito forestal debido a la medición rápida y no destructiva que hace de los ecosistemas forestales (Liang *et al.*, 2016). El propósito de los TLS es crear una nube de puntos (representación discreta) muy próxima a las superficies continuas de los objetos medidos. Para ello, el TLS se basa en el mismo principio que el LiDAR aerotransportado (Capítulo 10), con la diferencia de que el barrido se realiza desde un equipo ubicado en la superficie.

### 1.1. Principios del láser escáner

Los TLS emiten una señal óptica sobre un determinado objeto (sensores activos) y detectan su reflexión con el fin de medir ángulos, distancias e intensidad de los puntos iluminados. En función de la manera en la que el escáner recibe y/o analiza la señal reflejada, se pueden distinguir entre dos tipos:

- Escáneres láser de cambio de fase.
- Escáneres láser de tiempo de vuelo.

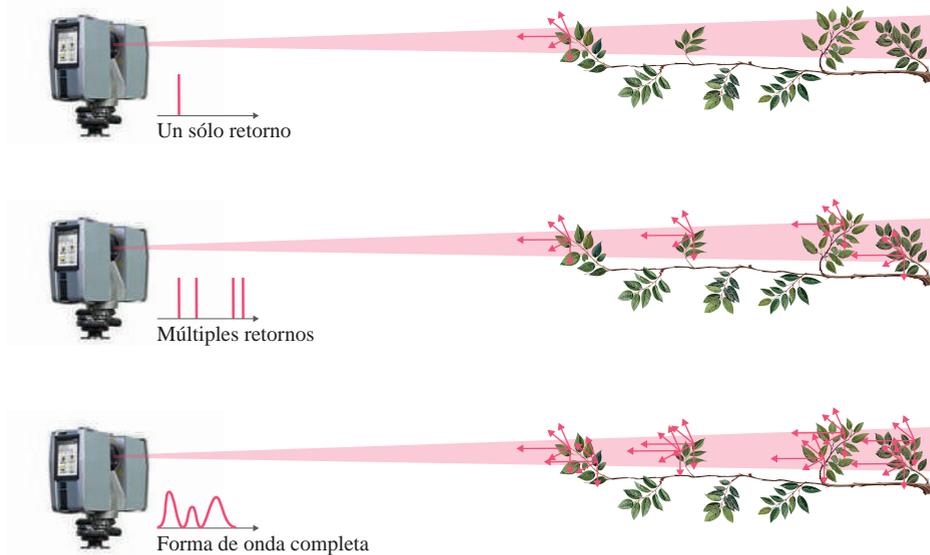
Los TLS utilizados en ingeniería se basan en el principio de tiempo de vuelo. Estos dispositivos miden un intervalo de tiempo entre dos sucesos. Dado que las ondas de luz viajan a una velocidad finita y constante a través de un medio, puede medirse el tiempo durante el cual la luz viaja del dispositivo a un objeto y regresa al mismo. La distancia a dicha superficie puede calcularse mediante la siguiente fórmula (Ecuación 1):

$$\text{Distancia} = ((\text{tiempo de recorrido}) \times (\text{velocidad de la luz})) / 2 \quad [1]$$

En general, hay dos modalidades de detectores LiDAR basados en el principio de tiempo de vuelo: escáneres discretos y escáneres de forma de onda completa (Figura 12.1). Los escáneres discretos registran retornos individuales representando los picos en la curva de la forma de onda. Este sistema puede registrar de uno (Figura 12.1 (sup.)) a cinco (Figura 12.1 (centro)) retornos de cada pulso láser. Una colección de puntos de retorno de LiDAR discretos se conoce como una nube de puntos LiDAR. Los retornos se registran cuando la intensidad sobrepasa un umbral predefinido del sistema.

Los escáneres de forma de onda completa registran la distribución de la luz retornada (Figura 12.1 (inf.)). Esta modalidad puede que contengan más información que los procedentes de sensores LiDAR discretos, pero el procesamiento de datos es complejo y requieren algoritmos para filtrar los datos y extraer la información útil.

Los dos últimos tipos de sensores (Figuras 12.1 (centro) y 12.1 (inf.)) proporcionan información de múltiples profundidades cuando el punto láser no es interceptado totalmente por el primer objeto encontrado, sino parcialmente por varios objetos.



**Figura 12.1.** Modalidades de detectores LiDAR según el número de señales de retorno calculadas para cada dirección y las capacidades de detección del sensor: captura de un solo retorno (sup.) (primer objeto que refleja una parte del pulso láser); captura de múltiples retornos (centro); y captura la forma de onda completa de la señal de retorno (inf.) (basado en Dassot *et al.* (2011)).

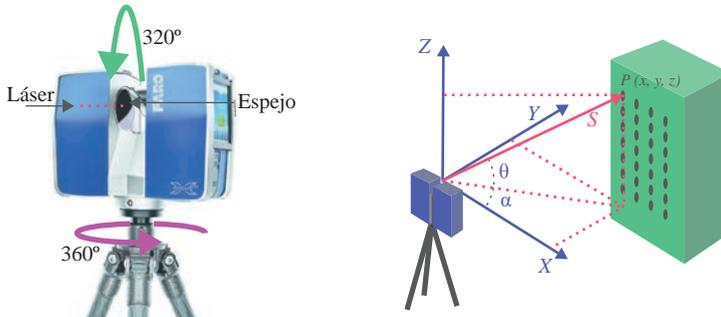
## 1.2. Funcionalidad y principios de medición

El TLS barre todo su campo visual (*Field of View*; FoV) variando la dirección del rayo láser para medir múltiples puntos desde el mismo punto de vista del escáner; bien rotando el propio dispositivo o bien utilizando una unidad de desviación. Generalmente, se emplean tres métodos para desviar el haz hacia una dirección específica:

- Prisma óptico reflectante.
- Círculo de fibra óptica.
- Espejo oscilante.

Este último método es el más utilizado, puesto que los espejos son más ligeros y pueden girar rápidamente y con una gran precisión. El escáner horizontal se denomina *frame scan* y puede tener un FoV desde 40° hasta 360°. El escáner vertical es el *line scan* y puede lograr un FoV desde 40° hasta 320° (Figura 12.2 (izq.)). Por cada señal reflejada, se obtienen dos ángulos correspondientes ( $\alpha$  y  $\theta$ ), la distancia  $\rho$  y la intensidad. Mediante los ángulos se puede definir la posición de cada uno de los puntos de la escena en un sistema

de coordenadas polares, que internamente es transformado a un sistema cartesiano (Figura 12.2 (dcha.)). Con todo, el TLS crea una nube de puntos dónde cada punto queda determinado por su posición (X, Y, Z).



**Figura 12.2.** Funcionalidad (izq.) y principio de medición (dcha.) del láser escáner terrestre (basado en Salehi y Wang (2017) y Xu *et al.* (2018)).

Los dispositivos TLS no sólo capturan información espacial, sino que también pueden adquirir de manera directa la intensidad de la señal devuelta y, de manera indirecta (a través de una cámara integrada o externa), la representación del espacio de color (RGB) de cada punto. Además, algunos equipos integran otros dispositivos auxiliares que permiten obtener imágenes térmicas o imágenes esféricas en alta resolución.

### 1.3. Escaneado láser estático y dinámico

El escaneado láser terrestre se puede dividir en dos categorías: escaneado láser estático y escaneado láser dinámico (Figura 12.3). El caso más común es que el equipo se estacione en un punto fijo (escaneado láser estático), generalmente sobre un trípode. Lo único que se precisa es el propio escáner y un instrumento de referencia que permita obtener coordenadas absolutas, generalmente un GPS diferencial (aunque se puede prescindir de éste y trabajar en coordenadas relativas). En los casi 60 años de historia del escaneado láser terrestre, los sistemas utilizados con mayor frecuencia han sido los escáneres láser terrestres estáticos o de posición fija, dada la creciente exactitud y precisión de la tecnología.

Cuando el proceso de toma de datos se realiza sobre una plataforma móvil, se llama escaneado láser dinámico (SLAM, en adelante; *Simultaneous Location and Mapping*). Estos sistemas requieren la integración en el dispositivo de otros componentes adicionales, tales como una unidad de medición inercial (INS) o un sistema de posicionamiento global (GPS), lo que hace que el sistema sea más complejo. Los datos capturados con un escáner en movimiento son menos precisos que los capturados con el equipo estacionado en un punto fijo, debido a la propagación de errores de posicionamiento dentro de la nube de puntos. Sin embargo, este enfoque ofrece ciertas,

ventajas como la velocidad de captura de datos, la facilidad de uso, el ahorro de costos y la movilidad de su tecnología. Según el refrán “no dejes que la perfección sea enemiga de lo bueno”, un escaneo móvil de alta velocidad y menor resolución puede ser más que suficiente para determinados proyectos (<https://www.faro.com/es-MX/Resource-Library/Article/Motionless-vs-Mobile-Scanning>). Un escaneo móvil basado en SLAM puede capturar un gran espacio de volumen sin necesidad de triangulación GPS o un posterior co-registro de los datos (ver sección 1.3).

Hasta ahora, estas dos categorías (estático y dinámico) han recorrido caminos separados. Actualmente, los avances en los algoritmos de *software* basados en SLAM y las velocidades de procesamiento de datos están ayudando a unificar ambos enfoques.



**Figura 12.3.** Diferencias en el proceso de toma de datos entre escaneo láser estático (izq.) y dinámico (dcha.).

#### 1.4. Aspectos metodológicos

En relación con la calidad de la medida del TLS, cada nube de puntos contiene un considerable número de puntos que presentan errores groseros. Las fuentes de error en el escaneo láser se dividen en cuatro categorías: i) errores instrumentales, ii) errores relacionados con el objeto, iii) errores relacionados con el entorno y iv) errores metodológicos. En aplicaciones forestales, los errores más frecuentes se deben a las condiciones ambientales (por el entorno) y al planteamiento metodológico.

Con respecto a las condiciones ambientales, podemos destacar los siguientes factores:

- Temperatura. Los escáneres láser sólo funcionan adecuadamente cuando se utilizan dentro de un cierto rango de temperatura. Se debe tener en cuenta que la temperatura en el interior del escáner puede ser bastante más alta que la exterior, debido al calor

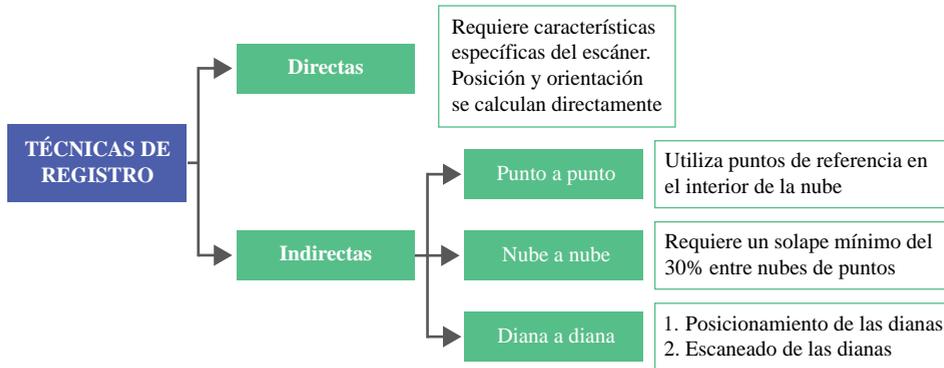
interno o al calor resultante de la radiación externa (radiación solar). Este hecho puede provocar dilataciones puntuales del dispositivo, distorsionando lentamente los datos tomados por el escáner; incluso podría limitar el funcionamiento del dispositivo.

- Humedad ambiental. En situaciones de elevada humedad (p.ej., niebla), pueden formarse gotas de agua por condensación sobre el escáner, afectando gravemente al correcto funcionamiento del dispositivo.
- Viento. Aunque la mayoría de los escáneres láser son muy rápidos, como cada parte de la nube de puntos se toma en un tiempo diferente, cualquier movimiento del escáner o del objeto escaneado distorsiona los datos tomados. Este efecto es especialmente importante cuando planteamos un diseño de posicionamientos múltiples (ver más adelante).

Los errores metodológicos se deben al método topográfico elegido o a la experiencia de los usuarios con esta tecnología. Por ejemplo, a la hora de establecer la resolución, una resolución más alta que la precisión por punto del escáner genera mayor ruido y un incremento en el tiempo de procesado. Otro error frecuente se produce a la hora de posicionar el escáner. Si la distancia entre el escáner y el objeto se encuentra próxima al alcance máximo del dispositivo, los escaneos contendrán mediciones menos precisas y un posible ruido.

Las compañías que fabrican estos dispositivos publican las precisiones de sus equipos. En las especificaciones del fabricante se proporciona el rango de distancias del dispositivo; aunque, usualmente, los valores de precisión y exactitud se indican sólo para una determinada distancia. Como a mayor distancia se produce una mayor dispersión del rayo y, por tanto, una menor intensidad de los retornos, es importante determinar hasta qué distancia se pueden asumir los valores de precisión y exactitud dados en las especificaciones del dispositivo (Ramos *et al.*, 2015). Estas restricciones, junto con las limitaciones por visibilidad, obligan en muchos casos a plantear escaneos parciales desde varios puntos y un posterior co-registro de ellos en un mismo sistema de coordenadas. Este enfoque se utiliza principalmente con dispositivos estáticos. Dependiendo del método de registro seleccionado, también se pueden producir errores. Los posibles errores generados durante la fase de registro también se encuentran en la categoría de errores metodológicos.

En los levantamientos con posicionamientos múltiples, el mismo espacio de volumen se escanea desde diferentes posiciones, produciéndose un solape entre las nubes de puntos capturadas desde cada posicionamiento. Por tanto, se deberán referenciar estos posicionamientos entre sí para obtener las nubes de puntos bajo el mismo sistema de referencia. Existen diferentes formas de llevar a cabo el registro. Las técnicas de registro se pueden clasificar en dos categorías: directas o indirectas. En el registro indirecto no se conocen las coordenadas del punto de posicionamiento. El registro mediante técnicas indirectas se puede realizar de tres formas distintas (Figura 12.4), y todas ellas deben tenerse en cuenta a la hora de planificar el trabajo.



**Figura 12.4.** Técnicas de registro de nubes de puntos procedentes de TLS.

Excepto en el registro “nube a nube”, donde el registro se produce gracias al solape existente entre las nubes de puntos, el resto de las técnicas indirectas requieren del establecimiento de puntos de control. En cada posicionamiento se debe asegurar que se pueden medir varios puntos de control (dianas) que permitan relacionar los diferentes escaneos. Las dianas son identificadas en el interior de la nube de puntos por su forma y alta reflectividad (Figura 12.5). Estas dianas se sitúan en lugares visibles desde varios estacionamientos, es decir, en las zonas de solape, debiendo permanecer fijas durante todo el trabajo.



**Figura 12.5.** Posicionamiento de las dianas dentro del área de acción del láser escáner (izq.) y visualización en la nube de puntos LiDAR (dcha.).

Otro aspecto que se debe considerar cuando se trabaja con dianas es la ubicación de éstas sobre el terreno. Las dianas deben estar distribuidas lo más ampliamente posible, no sólo en las direcciones de los ejes X e Y, sino también en la dirección del eje Z. Algunas configuraciones de las dianas no producen una solución única cuando se realiza

el registro. Por ejemplo, si todas las dianas se encuentran alineadas, tenemos un grado de libertad: la rotación en torno a esa línea. Algunos autores (Alba *et al.*, 2008; Wilkes *et al.*, 2017) indican que, dependiendo de la técnica de registro empleada, debe haber, al menos, cuatro dianas bien distribuidos en la zona de solape (Figura 12.5).

Para el registro directo, la nube de puntos debe estar georreferenciada. Eso se puede realizar de dos formas. En la primera de ellas, la posición y la orientación del escáner se calculan directamente, lo que requiere que el dispositivo cuente con algunas funciones de estación total. En la segunda opción, se fijan una serie de puntos de control (mínimo tres) dentro del espacio de trabajo. Las coordenadas de los puntos de control se capturan con un dispositivo GPS portátil. A continuación, en gabinete, se aplica una matriz de transformación para transformar la nube de puntos de un sistema de coordenadas locales a geográficas a partir de las coordenadas de estos puntos de control. El registro directo no requiere que las nubes de puntos se solapen y es, normalmente, más rápido que el indirecto. Sin embargo, se consiguen mejores precisiones con la georreferenciación indirecta que con la directa.

## 2. Aplicaciones del láser escáner terrestre en ciencias forestales

La correcta gestión de los sistemas forestales requiere de un amplio y pleno conocimiento de su realidad. El TLS posee la capacidad de registrar en detalle la estructura vertical y horizontal de una parcela de bosque. Esta posibilidad ha provocado un enorme interés por la implementación del TLS en la obtención de variables forestales. La mayor parte de los esfuerzos se han dirigido a su utilización en inventarios forestales a nivel de parcela, de los que se obtienen parámetros básicos como los diámetros normales y las alturas (Liang *et al.*, 2016). Sin embargo, su versatilidad y precisión han propiciado la innovación en el campo, llevando al desarrollo de algoritmos que permiten caracterizaciones más complejas e inventarios de mayor detalle (Calders *et al.*, 2020). A continuación, se muestran las principales aplicaciones del TLS en silvicultura y ecología forestal.

En la Tabla 12.1 se muestran solo algunas de las principales aplicaciones. Sin embargo, el conjunto de variables indicado anteriormente permite su aplicación en múltiples ámbitos de la silvicultura y la ecología forestal. Los primeros estudios de aplicación del TLS en el ámbito forestal se centraron en la caracterización de parámetros dendrométricos estándar (diámetro de los árboles y altura total). Su objetivo era demostrar el potencial de esta tecnología para realizar mediciones más rápidas y precisas, en comparación con los inventarios de campo tradicionales mediante útiles clásicos como la forcípula y el hipsómetro (Dassot *et al.*, 2011). Estos estudios pioneros demostraron el potencial del TLS en inventariación forestal. Posteriores casos de estudio evaluaron su capacidad para determinar volúmenes de árboles en pie y la detección automática de fustes. Los resultados de estos trabajos abrieron la puerta a otras investigaciones relacionadas; por ejemplo, la determinación y el cálculo de la biomasa forestal a partir de datos procedentes de TLS.

**Tabla 12.1.** Aplicaciones del TLS en silvicultura y ecología forestal.

Aplicación	VARIABLES	Ejemplos
Inventariación forestal	Densidad de árboles, diámetro del fuste, altura total, área basimétrica y volumen maderable	Moskal y Zheng, 2011; Liang et al., 2016, 2018; Krok et al., 2020; Molina-Valero et al., 2022
Tecnología e industria de la madera	Volumen, diámetros a lo largo del fuste, curvatura, sinuosidad, inclinación, coordenadas del centro de la sección y detección de nudos	Pyörälä et al., 2019; Alvites et al., 2021; Nguyen et al., 2021
Caracterización de la cubierta forestal	Índices de complejidad estructural, índice de área foliar, fracción de cabida cubierta, atenuación de la luz y orientación foliar	Zheng et al., 2012; Zheng y Moskal, 2012; Nunes et al., 2022
Desarrollo de modelos digitales	Modelos digitales de elevación, pendientes y vegetación	Crespo-Peremarch et al., 2020; Klapa et al., 2022

### 3. Procesado de los datos

Las nubes de puntos son datos no estructurados y, por tanto, deben ser tratados para poder extraer información. Por ello, se han desarrollado numerosas técnicas y algoritmos capaces de estimar parámetros forestales a partir de datos procedentes de TLS. Algunas compañías han incorporado estos flujos de trabajo en *softwares* especializados (Tabla 12.2).

**Tabla 12.2.** Ejemplos de *software* para la extracción de parámetros forestales a partir de nubes de puntos de TLS.

Software	Aplicaciones	Enlace
Computree	Preprocesado de las nubes de puntos, desarrollo de modelos digitales, extracción de parámetros dendrométricos estándar, estimación de parámetros de dosel y volumen de madera	<a href="https://computree.onf.fr">https://computree.onf.fr</a>
AMAPVox	Estimación de parámetros de dosel, como fracción de cabida cubierta, interceptación de la luz, densidad de área foliar, entre otros	<a href="https://amapvox.org">https://amapvox.org</a>
LiDAR360	Preprocesado de las nubes de puntos, extracción de parámetros dendrométricos estándar, estimación de densidad de árboles y desarrollo de algoritmos de regresión con datos externos para el cálculo de biomasa, rendimientos, entre otros	<a href="https://www.greenvalleyintl.com">https://www.greenvalleyintl.com</a>
3DFin	Preprocesado de las nubes de puntos, extracción de parámetros dendrométricos estándar	<a href="https://github.com/3DFin/3DFin">https://github.com/3DFin/3DFin</a>
3D FOREST	Preprocesado de las nubes de puntos, generación de modelos digitales, extracción de parámetros dendrométricos estándar y estimación de volumen de madera	<a href="https://www.3dforest.eu">https://www.3dforest.eu</a>

En la Tabla 12.2 se incluyen sólo algunos ejemplos. En concreto, se han incluido *softwares* que permiten realizar flujos completos de procesado; es decir, extracción de parámetros forestales desde la nube de puntos LiDAR. Sin embargo, existen otro tipo de *softwares* que ejecutan etapas iniciales o intermedias del flujo de procesado (ver más adelante), como el registro, georreferenciación, filtrado, etc. Entre los más utilizados se encuentran CloudCompare (<https://www.cloudcompare.org/>), FUSION ([https://forsys.sefs.uw.edu/fusion/fusion\\_overview.html](https://forsys.sefs.uw.edu/fusion/fusion_overview.html)) o ArcGIS Pro (<https://pro.arcgis.com>). Otro factor importante a la hora de seleccionar una de estas herramientas es el tipo de acceso (libre/bajo licencia). En este sentido, cabe destacar los *softwares* Computree, AMAPVox y 3D FOREST entre los más usados de libre acceso. Por el contrario, LiDAR360 o ArcGIS Pro, a pesar de ser muy intuitivos y contar con múltiples funcionalidades, requieren el pago de licencia.

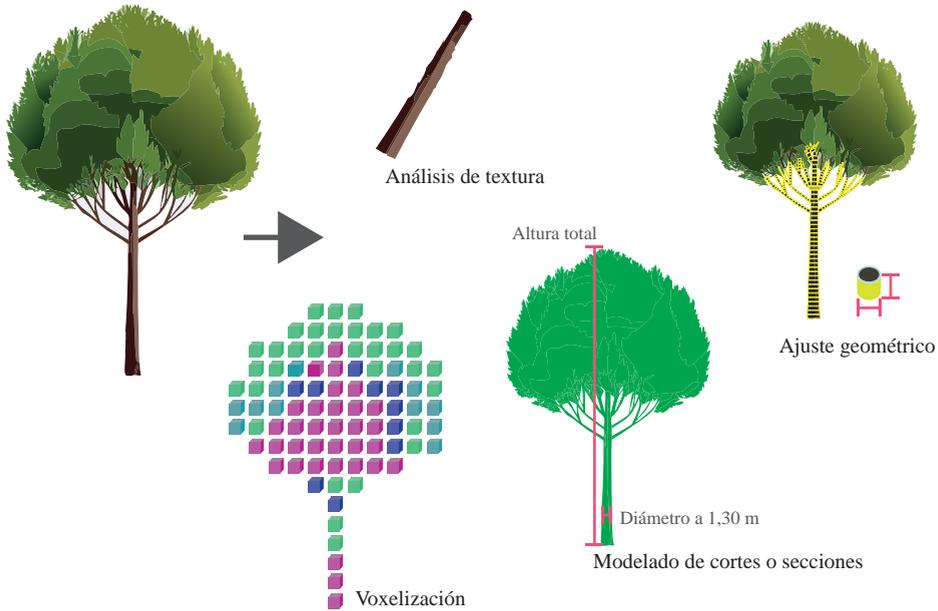
Sin embargo, el procesamiento de una nube de puntos mediante software es, en la mayoría de los casos, manual. Dado que un proyecto puede generar múltiples nubes de puntos y que los conjuntos de datos de un láser escáner comprenden miles de puntos, se necesitan métodos sofisticados para el procesamiento automático. Por este motivo, numerosos algoritmos capaces de extraer parámetros forestales a partir de datos procedentes de TLS se han traducido a lenguajes de programación como R o Python. Trabajar en un entorno de programación permite la automatización de determinados pasos del flujo de procesado e iterar sobre múltiples nubes de puntos de forma simultánea. En consecuencia, se ha producido un notable incremento de paquetes y librerías de estos lenguajes de programación para la visualización y el procesado de datos de TLS. En Ariza-Salamanca y Molina-Valero (2022) se describen algunos de ellos.

### 3.1. Métodos de estimación

En este subapartado se presentan varios métodos de reconstrucción de datos TLS que permiten extraer las métricas de árbol y las características del dosel a partir de la nube de puntos LiDAR (Figura 12.6), así como las herramientas que pueden utilizarse para desarrollarlos. Los métodos de estimación se pueden agrupar en cinco categorías:

- Modelos empíricos
- Modelado sobre cortes o secciones
- Ajuste geométrico
- Voxelización
- Análisis de textura

Excepto en el caso de los modelos empíricos, el resto de los métodos permite la extracción de métricas directamente desde la nube de puntos, sin necesidad de tomar información adicional en campo. La elección de un método u otro dependerá de las necesidades del usuario final. En la Tabla 12.3 se muestran algunos ejemplos de aplicación y las métricas asociadas a cada método. Además de los ejemplos de *software* indicados anteriormente (Tabla 12.2), se mencionan otras herramientas para programación como los paquetes FORTLS, TreeLS, rTLS o ITSM para R y la librería pcl para Python.



**Figura 12.6.** Reconstrucción de árbol y medición: diferentes métodos para extraer métricas de árbol y características del dosel a partir de la nube de puntos LiDAR.

**Tabla 12.3.** Métodos de reconstrucción y medición para extraer métricas de árbol y características del dosel a partir de la nube de puntos LiDAR y herramientas de aplicación. FORTLS: <https://molina-valero.github.io/FORTLS/>; TreeLS: de Conto *et al.* (2017); PCL: <https://pointclouds.org/>; ITSM: Terry *et al.* (2023); rTLS: (Guzmán *et al.*, 2021).

Método de estimación	Herramientas	Métricas
Modelos empíricos	FORTLS; LiDAR360	Diámetro del fuste, altura total, volumen de madera, densidad, área basimétrica
Modelado de cortes o secciones	TreeLS; LiDAR360; Computree; CloudCompare; 3D FOREST	Diámetro del fuste, altura total, parámetros de copa (volumen, área, profundidad, etc.)
Ajuste geométrico	Computree; PCL; ITSM	Diámetro en diferentes secciones (perfiles de fuste y ramas), volumen de madera y ramas, densidad, área basimétrica, altura total, morfología del fuste
Voxelización	Computree; AMAPVox; rTLS	Distribución 3D del material vegetal, topología del árbol, índice de área foliar, procesos de interceptación y transmitancia del dosel, complejidad estructural
Análisis de textura	Computree	Reconocimiento de especies, detección de defectos externos en la madera

### 3.2. Flujo de trabajo

El procesado de los datos hace referencia a la cadena de procesos que tienen lugar para transformar la nube de puntos en el producto final (Lerma *et al.*, 2008). En función de los resultados esperados, esta cadena de procesado será más o menos compleja. El flujo de procesado se puede agrupar en cuatro etapas:

1. Etapa de preprocesado o preparación de los datos.
2. Procesado de la nube de puntos.
3. Extracción de métricas de árbol individual.
4. Extracción de parámetros estructurales a nivel de parcela.

A su vez, cada una de estas etapas está compuesta por una serie de algoritmos que, aplicados de forma secuencial, contribuyen a la consecución de los resultados finales. Qué método se elija dependerá de las especificaciones del proyecto.

#### Preprocesado o preparación de los datos

En esta etapa se incluyen tres procesos comunes en proyectos con TLS: i) la conversión del formato, ii) el registro de múltiples posicionamientos y iii) la georreferenciación. Los dos últimos procesos son opcionales en función de los requerimientos del proyecto.

#### Formato de los datos

Los diferentes tipos de escáner almacenan los datos en una amplia gama de formatos. Para facilitar su procesado posterior, es recomendable que este formato de archivo sea fácilmente accesible y reconocible por una gran cantidad de *softwares*. Para las nubes de puntos 3D generadas por estas tecnologías se utiliza, por lo general, el formato *.las* o su versión comprimida *.laz* como archivo estándar para el intercambio de datos LiDAR (ver Capítulo 12). Estos formatos fueron creados por la *American Society for Photogrammetry and Remote Sensing*. Cada archivo está estructurado al menos en tres partes:

- *Header*. Contiene información de atributos sobre el propio estudio del LiDAR: fecha del vuelo, número de registros de puntos, etc.
- *VLRs* (acrónimo en inglés de *Variable Length Records*). Información opcional como el sistema de referencia de coordenadas, metadatos, etc.
- *Data*. Registro de los puntos tomados y sus atributos, como ubicación (x, y, z), intensidad, número de retorno, etc.

Es necesario conocer los formatos de fichero más comunes y la forma en la que almacena los datos de un escáner láser. En función del formato de los datos, éstos pueden contener más o menos información. Un formato de texto no ordenado como *.xyz* sólo puede contener las tres coordenadas de un punto, mientras que otro formato de texto no ordenado como *.xyzrgb* contiene las tres coordenadas del punto y los valores del espacio de color. Determinadas compañías tienen sus propios formatos de texto (ordenados y no ordenados).

El formato de los datos es determinante para desarrollar ciertos flujos de procesado. Así, para estimar propiedades del dosel, como el índice de área foliar o la interceptación/transmitancia de la radiación, los datos deben contener información específica, como las trayectorias, la información angular o las intensidades. Esta información se utiliza como parámetros de entrada en las ecuaciones desarrolladas para el cálculo de las variables. Como ejemplo, el *software* AMAPVox (método de voxelización, ver Tablas 12.2 y 12.3) estima la densidad de vegetación (PAD, acrónimo de *Plant Area Density*) mediante la aproximación de Beer-Lambert (Vincent *et al.*, 2017; Maeda *et al.*, 2022):

$$\text{PAD} = \lambda_o / G(\theta) \quad [2]$$

Donde  $G(\theta)$  es función del ángulo cenital de la dirección del haz láser, que es la dirección de “disparo” del escáner.

Por tanto, estas estimaciones no podrían realizarse si el formato de los datos no contuviera esta información. Entre los formatos de datos que contienen esta información, encontramos: *.rxp* y *.rsp* (Riegl), *.ptx* y *.ptg* (Leica) o *.xyb* (FARO). Sin embargo, otros métodos de procesado, como el ajuste geométrico o el modelado de cortes o secciones, pueden ejecutarse con la información que contienen formatos como *.las* o *.xyz*. Por ello, el primer paso en la etapa de preprocesado es la conversión de la nube de puntos LiDAR al formato más conveniente para el método de procesado seleccionado.

### Registro de múltiples posicionamientos

En sistemas forestales se trabaja con unidades espaciales relativamente grandes (parcelas, rodales, etc.), por lo que un solo escaneo puede derivar en un muestreo limitado. Esta restricción y las limitaciones por visibilidad (oclusión; Figura 12.7 (sup.)), obligan, en muchos casos, a plantear escaneos parciales desde varios puntos y un posterior co-registro de ellos en un mismo sistema de coordenadas (Figura 12.7 (inf.)).

Cuando se plantea este enfoque de localización de múltiples escaneos, el siguiente paso en la etapa de preprocesado es la unión precisa de estos múltiples escaneos (registro). En apartados anteriores se han descrito las diferentes técnicas de registro (ver apartado 1.4. Aspectos metodológicos). A continuación, se describen herramientas y métodos para realizar el registro mediante técnicas indirectas (Figura 12.4).

Las dianas se utilizan para identificar puntos comunes entre ambos posicionamientos. Existe una amplia variedad de dianas artificiales. En muchas ocasiones estas dianas son suministradas por las compañías fabricantes de dispositivos y están elaboradas con materiales altamente reflectantes. Estas dianas son reconocidas de forma automática en determinados *softwares* de procesado. Además, estas compañías también han desarrollado sus propios *softwares* de registro de nubes de puntos procedentes de TLS, permitiendo realizar el proceso de registro de forma automática. Un ejemplo es el *software* Leica Cyclone REGISTER 360 (<https://leica-geosystems.com/es-es/products/laser-scanners/software/leica-cyclone/leica-cyclone-register-360>). Este tipo de *softwares* proporcionan

flujos de trabajo de registro guiados, que ayudan a los nuevos usuarios y aceleran el procesamiento de los datos para los usuarios experimentados. Las herramientas y los informes de calidad generados eliminan las conjeturas acerca de la calidad del registro y ponen toda la información necesaria al alcance del usuario.

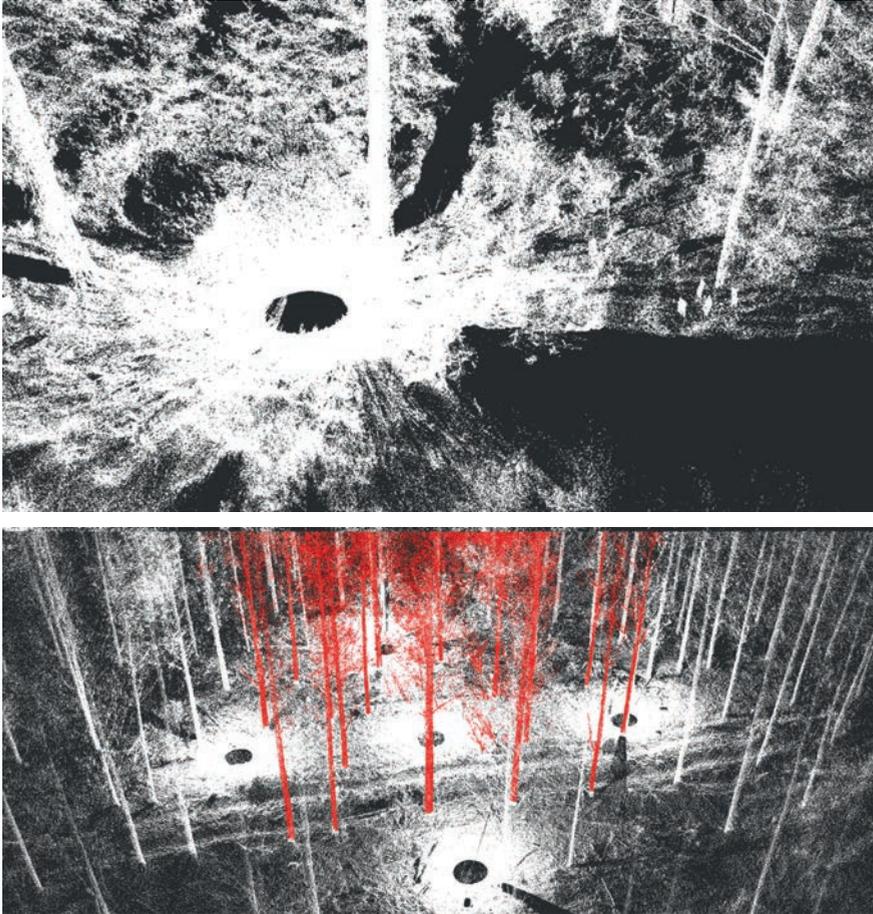
Sin embargo, existen otras alternativas de bajo coste. Se pueden utilizar dianas impresas en papel, que son altamente efectivas siempre que se sitúen en objetos cuya superficie pueda ajustarse a una forma geométrica ideal y sean fácilmente identificables en una visualización de la nube de puntos. Por otro lado, el *software* libre CloudCompare incorpora el algoritmo ICP (acrónimo de *Iterative Closest Point*) que permite realizar el alineado de las nubes de puntos mediante el cálculo de la distancia existente entre todos los puntos de la nube. El algoritmo estima la transformación para alinear ambas nubes de puntos de forma que se minimice el error. Cuando las nubes de puntos no se encuentran bien orientadas y/o posicionadas, el empleo de esta técnica puede producir un efecto “dominó” que propaga los errores y deriva en errores globales grandes. Para posicionar correctamente el conjunto de datos, es necesario que el usuario marque, al menos, tres pares de puntos comunes (dianas) en ambas nubes de puntos. Una vez finalizado el cálculo, el programa muestra una ventana con la matriz de transformación, el error y el solape teórico.

### **Georreferenciación**

Parte de los dispositivos que se encuentran actualmente en el mercado capturan los datos en un sistema de coordenadas local. Este sistema de coordenadas puede transformarse a un sistema de coordenadas geográficas a partir del establecimiento de puntos de control (ver apartado 1.4. Aspectos metodológicos). Un punto de control se define como una ubicación geográfica específica registrada mediante el uso de un sistema de posicionamiento global (GPS). Estos puntos de control se pueden tomar sobre los diferentes posicionamientos del escáner o sobre ubicaciones específicas donde posteriormente se van a fijar objetos perfectamente identificables durante la fase de escaneo.

Existe una amplia gama de *softwares* con paquetes específicos que permiten transformar el conjunto de datos a un sistema de coordenadas geográficas. El *software* libre CloudCompare incluye, entre sus funcionalidades, la herramienta Align (*point pairs picking*). Esta técnica utiliza el archivo que contiene las coordenadas geográficas de los puntos de control como referencia para alinear la nube puntos. La identificación y la selección de los puntos de control en la nube de puntos debe realizarse de forma manual desde la interfaz gráfica de CloudCompare. Esta técnica necesita, como mínimo, cuatro puntos de control para realizar la georreferenciación. El programa muestra el error cometido en cada uno de los puntos, además del error global; una vez finalizado el proceso se muestra en pantalla la matriz de transformación.

La georreferenciación es un paso opcional de la etapa de preprocesado y, por tanto, depende de las especificaciones del proyecto.



**Figura 12.7.** Limitaciones de visibilidad por obstáculos en el entorno del escáner (oclusión) (sup.) y nube de puntos resultado del registro de múltiples posicionamientos (inf.).

## Procesado de la nube de puntos

### Extracción del área de trabajo

La extracción del área de trabajo es una buena práctica, que, además, reducirá el tamaño de las nubes de puntos, ya que éstas suelen ser archivos muy pesados. En consecuencia, se reducirán también los tiempos de computación. Sin embargo, en muchos casos puede ser recomendable dejar un *buffer* para que muchos de los algoritmos implementados puedan detectar mejor los árboles cercanos al borde del área de trabajo. Este paso puede ser desempeñado en *softwares* libres, como CloudCompare y Computree, o en *softwares* comerciales de los fabricantes de dispositivos, entre otras aplicaciones.

## Filtrado

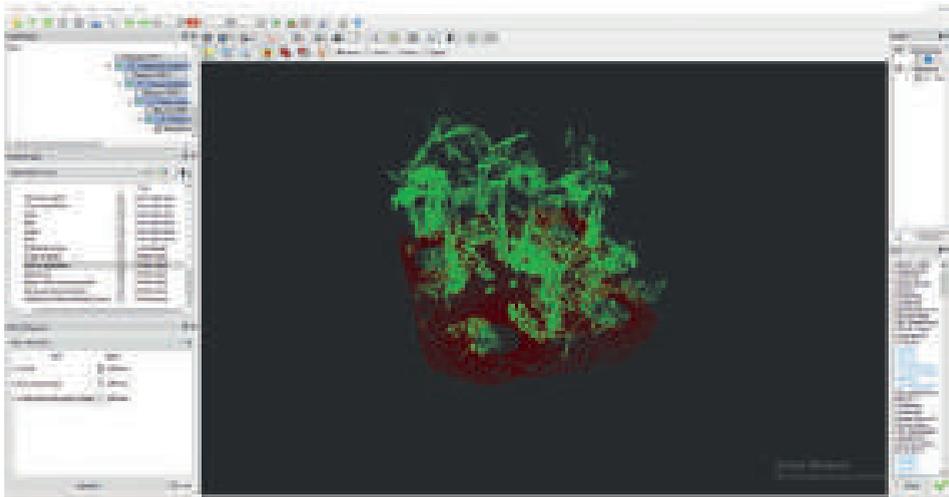
Filtrar las nubes de puntos con objeto de eliminar “ruido” es otra operación que puede mejorar los resultados obtenidos. Hay varios algoritmos que pueden ser utilizados para tal fin, empezando por algo tan sencillo como eliminar puntos repetidos. En cualquier caso, todos ellos conllevarán una reducción de la densidad de las nubes de puntos, lo cual, por lo general, será deseable desde el punto de vista computacional. Algunos de los algoritmos más relevantes son los siguientes:

- Submuestreo de la nube de puntos basado en la distancia. Esta operación elegirá puntos de la nube original de forma que ningún punto de la nube de salida esté más cerca de otro punto que una distancia especificada. Es una operación muy interesante cuando se trata de nubes de puntos registradas o de tecnología SLAM, pues puede servir para obtener una densidad de puntos más homogénea en el espacio. Al respecto, debe tenerse en cuenta que un mismo objeto puede no mostrar una densidad de puntos constante, como ocurre, por ejemplo, en diferentes caras del fuste de un árbol cuando éstas han sido escaneadas desde diferentes estacionamientos. Si los dispositivos no se encontraban a la misma distancia, la precisión de la nube de puntos (y, por ende, la densidad) será distinta en las distintas caras. Después de aplicar este submuestreo, la cantidad de puntos que representan un objeto será más proporcional al tamaño del mismo, funcionando mejor muchos de los algoritmos habitualmente utilizados. No obstante, es importante determinar una distancia mínima adecuada a la precisión de la nube de puntos. Esta operación puede realizarse en CloudCompare.
- Exclusión de puntos atípicos, definidos como aquellos que están más alejados de sus vecinos en comparación con la media de su entorno próximo. Para ello, se considera un número determinado de puntos vecinos más cercanos a un punto focal y se estima la distancia media de dichos puntos a este punto focal. El filtrado consiste en eliminar aquellos puntos que están a una distancia superior a la distancia media más la desviación estándar multiplicada por un valor determinado a definir por el usuario. Este algoritmo se suele encontrar por su nombre en inglés *Statistical Outlier Removal* (SOR) y está disponible, por ejemplo, en CloudCompare, la librería *pcl* de C++ o el paquete de R *VoxR*.

## Clasificación

La clasificación de los puntos pertenecientes al suelo es uno de los pasos fundamentales en el procesado de las nubes de puntos. Hay varios algoritmos desarrollados para ello, siendo uno de los más utilizados el CSF (*Cloth Simulation Filter*; Zhang *et al.*, 2016). Una de las ventajas de este algoritmo es que funciona con nubes de puntos pertenecientes a dispositivos LiDAR con un solo retorno, que son comunes en dispositivos LiDAR terrestres. La idea que hay incorporada en el CSF es invertir la nube de puntos original y, a continuación, dejar caer un “pañó” sobre la superficie invertida desde arriba. Analizando las interacciones entre los nodos de la tela y los puntos LiDAR correspondientes, se puede determinar la forma final

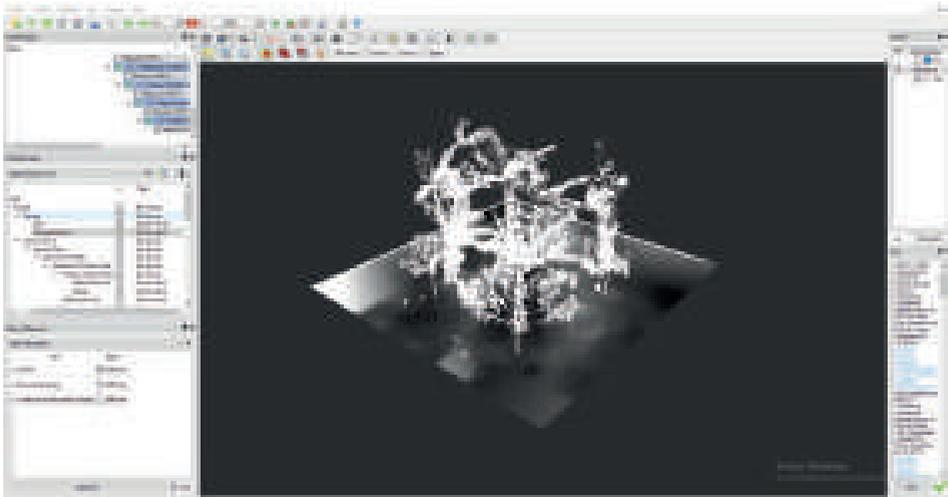
de la tela y utilizarla como base para clasificar los puntos originales como suelo o no suelo. Este algoritmo está incluido en muchas de las soluciones de *software* (CloudCompare, FORTLS, etc.). El resultado es una clasificación binaria que asigna a cada punto una clase, o bien punto de vegetación o bien punto de suelo (Figura 12.8).



**Figura 12.8.** Resultado del proceso de clasificación, en verde se representan los puntos clasificados como vegetación y, en marrón, los puntos de suelo. Interfaz gráfica de usuario del *software* Computree.

### Generación de modelos digitales

Una vez clasificados los puntos pertenecientes al suelo, éstos se pueden utilizar para la generación de modelos digitales del terreno (MDT), que es un paso fundamental para otros pasos del flujo de trabajo (Figura 12.9). Hay varios algoritmos que han sido desarrollados para este proceso, destacando la interpolación de la distancia inversa ponderada (IDW, acrónimo en inglés de *inverse distance weighting*) o la generación de una red irregular de triángulos (TIN, acrónimo en inglés de *triangular irregular networks*). El método IDW supone que los elementos cercanos entre sí son más parecidos que los que están más alejados. Para predecir un valor para cualquier ubicación sin medición (interpolación), IDW usa los valores medidos que rodean a la ubicación de predicción, asignando ponderaciones altas a aquellos más cercanos y viceversa. Estas ponderaciones disminuyen como función de la distancia (de ahí el nombre de distancia inversa ponderada). El método TIN construye una superficie continua mediante una malla triangular basada en una triangulación de Delaunay. Es importante elegir una adecuada resolución para generar el MDT, que en el caso de dispositivos LiDAR terrestres resulta ser de un valor de 20 cm según Liang *et al.* (2018). Estos algoritmos también están incluidos en muchas de las herramientas disponibles (CloudCompare, FORTLS, etc.).



**Figura 12.9.** Modelo Digital del Terreno (MDT) generado a partir de la nube de puntos LiDAR. Interfaz gráfica de usuario del *software* Computree.

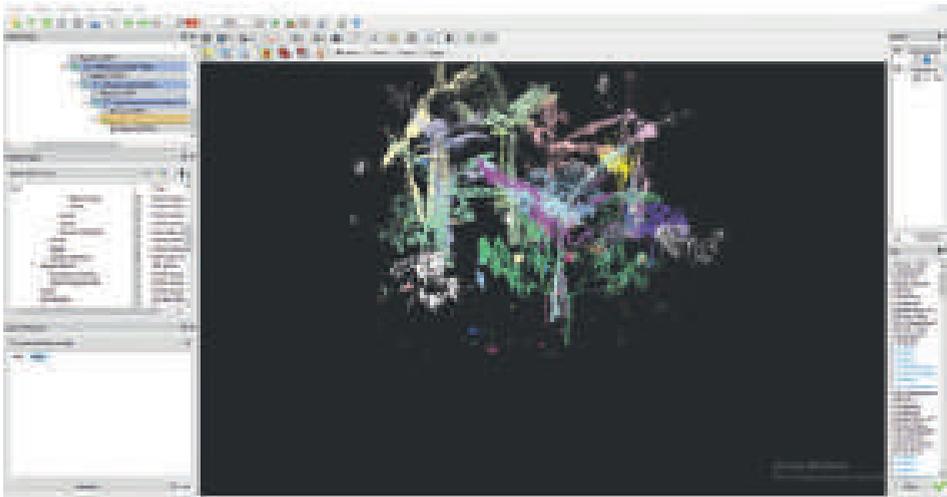
## Segmentación

Para extraer métricas a nivel de árbol individual, cada individuo debe ser tratado de forma individual como un subconjunto independiente del conjunto de datos global. Se han desarrollado diferentes algoritmos que permiten aislar conjuntos de datos de la nube de puntos mediante clasificación. La segmentación de árboles individuales a partir de nubes de puntos procedentes de entornos agroforestales es uno de los pasos del procesado que más tiempo consume, especialmente en parcelas con alta complejidad estructural (elevada densidad de individuos, múltiples estratos, etc.). Se han desarrollado múltiples algoritmos para la segmentación automática de árboles que permiten reducir notablemente el tiempo de procesado (ver Burt *et al.* (2019) y Wang *et al.* (2020)). Sin embargo, el proceso sigue siendo bastante manual y, en función de las características de la parcela, suele generar un gran número de errores (Demol *et al.*, 2022).

Para minimizar los errores en el proceso de segmentación, se recomienda un método que subdivide este proceso en dos pasos. Inicialmente se realiza una segmentación basada en una búsqueda de clústeres (agrupaciones de puntos) puntuales. Este algoritmo produce “semillas” para la segmentación, también conocidas como raíces de árbol detectadas, que componen un punto de inicio para el segundo algoritmo. Como se indica en Bienert *et al.* (2006) y Torresan *et al.* (2018), la segmentación se inicia en un corte horizontal a través de la nube de puntos y a una altura determinada por encima del modelo del terreno. Se recomienda una altura de corte por debajo de 1,30 m de altura, evitando zonas con una elevada ramificación. En el *software* Computree esta técnica puede ejecutarse mediante la herramienta *euclidean clustering operation*. Este proceso identifica clústeres de puntos con un número de puntos superior a un valor umbral indicado. Como resultado de esta segmentación se genera una lista

de objetos con todos los árboles clasificados (semillas para la segmentación) y una lista de objetos rechazados con los clústeres de puntos que no cumplen en el proceso de clasificación (pequeñas ramas y otros elementos que se encuentran en el corte realizado).

A partir de las semillas generadas en el proceso anterior, se realiza la segmentación de los árboles individuales. Para ello, el segundo proceso toma como entrada los puntos clasificados como vegetación (ver apartado de clasificación) y los clústeres “semilla”, extendiendo la clasificación al conjunto de puntos de la nube (Figura 12.10). Un algoritmo que se utiliza con frecuencia para realizar este proceso es el algoritmo competitivo de Dijkstra (Dijkstra, 2022). Este algoritmo encuentra de manera eficiente los caminos mínimos entre los puntos de interés. En este contexto, cada punto de la nube se representa como un vértice en un grafo ponderado, donde las distancias entre los puntos reflejan la proximidad espacial. El algoritmo prioriza la expansión de los caminos más cortos mediante una cola de prioridad, actualizando continuamente las distancias acumulativas. Al introducir la competencia, se incorpora la noción de separación entre segmentos al considerar restricciones o umbrales específicos para la distancia acumulativa. Esto permite la identificación de regiones discretas en la nube de puntos, contribuyendo así a la segmentación precisa en base a criterios de proximidad definidos por el usuario.



**Figura 12.10.** Resultado del proceso de segmentación mediante clústeres “semilla”, aplicando el algoritmo competitivo de Dijkstra. Cada color representa un individuo diferente. Interfaz gráfica de usuario del *software* Computree.

### Extracción de métricas de árbol individual

Se han desarrollado diferentes métodos y aproximaciones para la estimación de parámetros forestales a partir de nubes de puntos procedentes de escaneado láser terrestre (ver Tabla 12.3). En este apartado se resumen algunos de los principales procedimientos que se han desarrollado para cada método.

## Modelado de cortes o secciones

Parámetros como la altura total son fácilmente identificables a partir de la información que contiene cada uno de los puntos que componen la nube. Por ejemplo, para extraer la altura total de todos los árboles presentes en una parcela, basta con determinar el punto de mayor altitud en cada uno de los clústeres generados en el proceso de segmentación (ver apartado anterior). En el caso de parámetros como el diámetro normal o el área basimétrica, la determinación es más compleja y requiere la aplicación de otro tipo de procedimientos.

Simonse *et al.* (2003) desarrollaron un procedimiento que ha sido implementado en diferentes *softwares*, como Computree o CloudCompare. Este procedimiento trata de identificar estructuras circulares, es decir secciones del fuste, mediante la aplicación de una transformación de Hough (Illingworth y Kittler, 1987). En primer lugar, se genera el MDT y, a partir del mismo, se realiza un corte a la nube de puntos a 1,30 cm de altura. Una vez generadas estas estructuras circulares, se determinan parámetros como el diámetro a partir de las coordenadas de los puntos.

## Ajuste geométrico

El método de ajuste geométrico se emplea para extraer parámetros forestales mediante el ajuste de geometrías tridimensionales a las estructuras arbóreas detectadas. Mediante técnicas de optimización, se ajustan formas geométricas, como cilindros o conos, a los perfiles de los árboles. La optimización busca minimizar las discrepancias entre la geometría modelada y los datos LiDAR. De este modo, los parámetros forestales se determinan a partir de la información que contiene cada una de estas geometrías (volumen, diámetro, altura, etc.). Se han desarrollado diferentes metodologías para realizar este ajuste geométrico. En Hackenberg *et al.* (2014) se proporciona una revisión integral que abarca los desarrollos más utilizados. Dentro de las opciones disponibles para *software* libre, se encuentran la rutina *Random Sample Consensus* (RANSAC; Raumonon *et al.*, 2013) implementada en CloudCompare y la de la esfera siguiente (*QSM spherefollowing method*; Hackenberg *et al.*, 2015) para Computree (Figura 12.11).

## Extracción de parámetros estructurales a nivel de parcela (caso práctico basado en el paquete de R FORTLS)

El paquete FORTLS ha sido desarrollado para automatizar el análisis y procesado de los datos generados mediante dispositivos terrestres de tecnología LiDAR (TLS y SLAM) con el principal objetivo de estimar variables de interés en la monitorización forestal. Para ello, FORTLS permite: (i) detectar árboles y estimar algunos de sus atributos (diámetros, altura, etc.), (ii) estimar variables de masa o dasométricas (densidad, área basimétrica, etc.), (iii) generar métricas relacionadas con variables importantes en la gestión y en la planificación forestal, y (iv) evaluar diseños de parcela apropiados sobre la base de la calibración con datos de control (p. ej. datos medidos en campo de manera convencional).



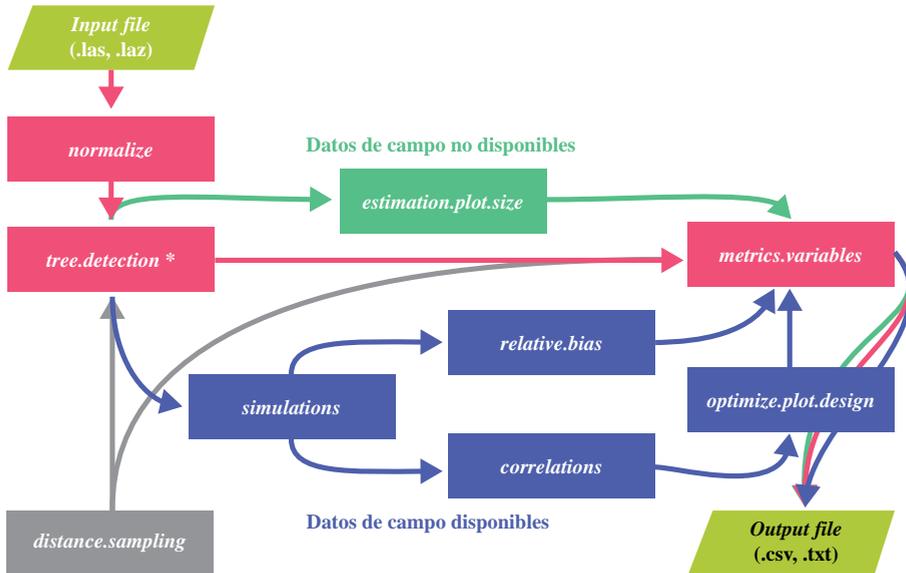
**Figura 12.11.** Reconstrucción de un árbol individual mediante la rutina QSM.

El paquete también incluye metodologías para la corrección de oclusiones cuando se trabaja con escaneos únicos de TLS, reduciendo, así, los tiempos de adquisición y procesado de los datos. Todas estas características confieren a FORTLS un enorme potencial para hacer operativo el uso de la tecnología LiDAR en inventario forestal; permite explorar no sólo los métodos de muestreo convencionales (p. ej. muestreo aleatorio simple o sistemático), sino también las metodologías de inferencia asistidas y/o basadas en modelos que ya son operativas con otros sensores remotos.

En esencia, FORTLS está compuesto por una serie de funciones descritas de forma detallada en el manual y las viñetas disponibles en el repositorio CRAN (<https://CRAN.R-project.org/package=FORTLS>). Además, también se puede acceder a la versión en desarrollo de FORTLS contenida en la plataforma GitHub (<https://github.com/Molina-Valero/FORTLS/tree/devel>). Las funciones que componen FORTLS pueden encadenarse y formar diferentes flujos de trabajo capaces de dar respuesta a las funcionalidades anteriormente especificadas (Figura 12.12). Todas ellas están, o pueden estar, interconectadas, por lo que se recomienda trabajar siempre en un mismo directorio de trabajo, ya que habrá *inputs* y *outputs* dependientes los unos de los otros a lo largo de estos flujos de trabajo e, incluso, durante la ejecución de algunas funciones. Para ello, lo primero que se aconseja es definir tanto el directorio de entrada de los datos como el de salida de los resultados al comienzo de cualquier rutina de trabajo, siendo éstos coincidentes, como ya se ha recomendado.

El objetivo de esta sección no es explicar en detalle los algoritmos desarrollados en FORTLS; éstos pueden consultarse en Molina-Valero *et al.* (2022). Esta sección se centrará en explicar las principales rutas de trabajo que pueden utilizarse de una manera

Lo más sintética posible y didáctica, para lo cual se han incluido fragmentos de código y *scripts* en el repositorio de este capítulo (ver QR al final del capítulo para acceder al mismo). La idea es que el usuario pueda tener unas nociones básicas, y, sobre esta base, que pueda seguir evolucionando en el procesado de nubes de puntos, tanto a nivel de usuario técnico como investigador.



**Figura 12.12.** Funciones de FORTLS y posibles flujos de trabajo. El itinerario rojo muestra los pasos imprescindibles para la obtención de métricas y variables de masa a partir de nubes de puntos generadas con dispositivos terrestres de tecnología LiDAR. El itinerario verde indica la posibilidad de elegir el mejor diseño de parcela cuando no hay datos de campo (o datos de control) para calibrar. El itinerario azul señala el flujo de trabajo necesario para optimizar el diseño de parcela mediante la calibración con datos medidos en campo (o datos de control). La función *distance.sampling* (ruta gris) es un proceso opcional que incorpora metodologías para la corrección de oclusiones en escaneos únicos de TLS. \*Este paso está compuesto por dos funciones principales según la procedencia de las nubes de puntos: (1) escaneos únicos de TLS (*tree.detection.single.scan*), o (2) escaneos múltiples de TLS o tecnología SLAM (*tree.detection.multi.scan*). Hay una tercera función para procesar varias nubes de puntos en serie de forma automática (*tree.detection.several.plots*).

### Variables de árbol individual (o dendrométricas)

Para la obtención de variables de árbol individual, así como de otras funcionalidades, es necesario normalizar la nube de puntos mediante la función *normalize*. El objetivo de este proceso es obtener coordenadas relativas a un punto del plano cartesiano (x, y) especificado. Debido a que FORTLS se desarrolló como una herramienta para

explorar el uso del TLS como un instrumento de muestreo, este punto especificado debe coincidir con el centro de la parcela si se quiere ir más allá de la estimación de variables dendrométricas. Sin embargo, en el caso particular del escaneo único de TLS, es crucial que este punto se corresponda con el de estacionamiento del TLS en campo. De otro modo, algunos de los procesos implementados en los algoritmos de FORTLS podrían dar errores. En cuanto a los datos de entrada conteniendo las nubes de puntos originales que van a ser normalizadas, éstas deberán estar en formato *.las* o *.laz*. En el repositorio (ver QR al final del capítulo para acceder al mismo) se muestra un ejemplo del código que habría que utilizar incluyendo algunos de los argumentos más relevantes.

El output de esta función consistirá en un objeto de datos tipo *dataframe* (guardado por defecto como *.txt* en el directorio especificado en el argumento *dir.result*) con las coordenadas cartesianas (x, y, z), cilíndricas (distancia horizontal o reducida  $\rho$ , ángulo azimutal  $\varphi$ , z) y esféricas (distancia euclídea o real r, ángulo polar  $\theta$ ,  $\varphi$ ) relativas al origen definido en los argumentos *x.center* e *y.center*. Hay que tener en cuenta que, de no especificar las coordenadas del centro de la parcela, la función *normalize* considerará como punto central las medias aritméticas entre los valores máximos y mínimos de las coordenadas x e y [ $x_{centro} = (x_{min} + x_{max})/2$ ;  $y_{centro} = (y_{min} + y_{max})/2$ ]. También habrá una columna (con el encabezado *prob.selec*) donde se asigna una probabilidad de selección a cada punto de 0 o 1. Reteniendo aquellos puntos con una probabilidad de selección de 1, se podrán obtener nubes de puntos reducidas. En el caso de escaneos únicos de TLS, esta reducción de la densidad estará condicionada por la asignación de una mayor probabilidad de retención a puntos más alejados, y de una menor probabilidad a puntos más cercanos, obteniendo así nubes de puntos con densidades isotrópicas. La metodología empleada para la reducción de la densidad de puntos en el caso del escaneo único puede consultarse en Molina-Valero *et al.* (2022). En el caso del escaneo múltiple, esta reducción de la densidad está basada en una selección aleatoria del 50% de los puntos. Aunque se han comentado las variables más relevantes, si se quiere profundizar en el resto de variables obtenidas, se recomienda consultar el manual del paquete u otras fuentes de información.

El *output* de la función *normalize* se utilizará como *input* en las funciones destinadas a la detección de árboles y a la estimación de variables dendrométricas. En caso de utilizar escaneos únicos de TLS, la función que se debe utilizar se llama *tree.detection.single.scan*. Esta función requiere un argumento específico donde se tienen que indicar los parámetros de precisión configurados en el escaneo (*tls.resolution*), bien definidos como la distancia en mm entre dos puntos consecutivos a una distancia concreta del TLS en m, o bien definidos por las precisiones angulares horizontal y vertical en grados sexagesimales [*tls.resolution = list(horizontal.angle = °, vertical.angle = °)*]. Para escaneos múltiples de TLS o tecnología SLAM, la función a utilizar será *tree.detection.multi.scan*, para la que no se requiere especificar la precisión de escaneo. Sin embargo, existe la posibilidad de definir la precisión media de las nubes de puntos resultantes mediante el argumento *tls.precision*; que, por defecto, será de 0,03 metros

(valor apropiado en la mayoría de las situaciones testadas). Variaciones sensibles de este valor (p. ej., 0,05 metros) pueden mejorar la ratio de detección de árboles en casos en los que las nubes de puntos tienen menor precisión, como puede suceder a veces con escaneos procedentes de dispositivos con tecnología SLAM. En el repositorio (ver QR al final del capítulo para acceder al mismo) se muestra un ejemplo del código que habría que utilizar incluyendo algunos de los argumentos más relevantes.

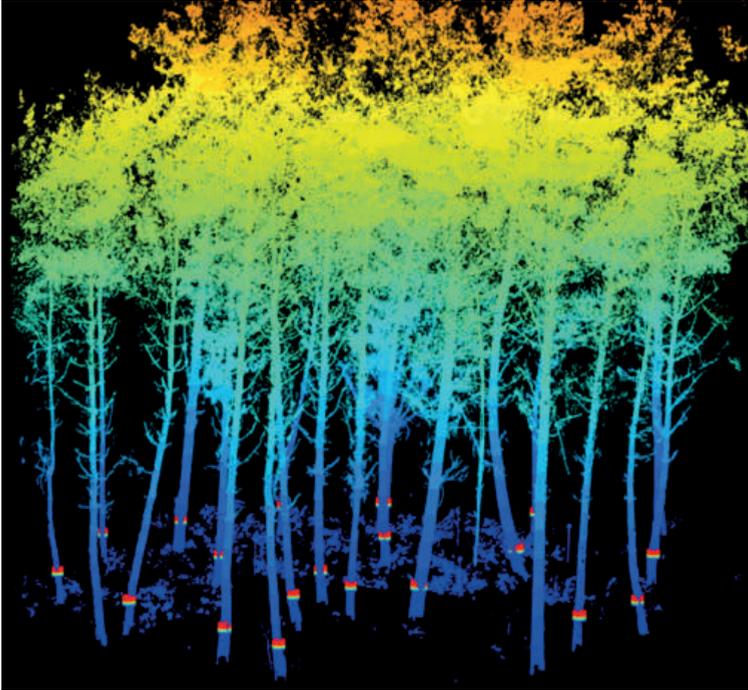
El *output* de estas funciones será un objeto de datos tipo *dataframe* (guardado por defecto como *tree.tls.csv* en el directorio especificado en el argumento *dir.result*) que contendrá, para cada árbol detectado, algunas de las variables dendrométricas más importantes utilizadas en la gestión forestal. En la Tabla 12.4 se especifican las más relevantes y en la Figura 12.13 puede observarse un ejemplo de la salida gráfica tras aplicar las funciones. Además, en el directorio especificado en *dir.result* se guardará un archivo llamado *stem.curve.csv* con los diámetros estimados a diferentes alturas para cada árbol detectado.

**Tabla 12.4.** Principales variables de árbol individual obtenidas con FORTLS.

Variables	Descripción
<i>x, y</i>	Coordenadas (m) del centro del fuste del árbol a 1,3 m de altura
<i>h.dist</i>	Distancia horizontal (m) desde el centro de la parcela hasta el centro del fuste del árbol a 1,3 m de altura
<i>dbh</i>	Diámetro (cm) del fuste estimado a 1,3 m de altura (diámetro normal o a la altura del pecho en el argot forestal)
<i>h</i>	Altura (m) desde la base hasta el ápice del árbol
<i>h.com</i>	Altura (m) del fuste del árbol desde la base hasta un diámetro determinado en punta delgada. Para obtenerla, este diámetro en punta delgada debe ser especificado en centímetros en el argumento <i>d.top</i>
<i>v</i>	Volumen (m <sup>3</sup> ) total del fuste del árbol
<i>v.com</i>	Volumen (m <sup>3</sup> ) del fuste del árbol desde la base hasta un diámetro determinado en punta delgada especificado en el argumento <i>d.top</i>
<i>SS.max</i>	Curvatura del fuste, medida como el máximo sagita (ver Prendes et al. (2022))
<i>sinuosity</i>	Sinuosidad del fuste, estimada como el cociente entre la longitud del fuste y la longitud de una línea recta que une las secciones extremas del fuste (ver Prendes et al. (2022))

### Variables de masa (o dasométricas)

FORTLS incluye una función específica para la obtención de variables y métricas de masa a nivel de parcela. Esta función es *metrics.variables* y funciona para parcelas circulares de área fija, parcelas *k-tree* y parcelas relascópicas (o de conteo angular). Todas ellas representan diseños de parcela habitualmente utilizados en inventario forestal. Para ello, hay que suministrarle como *input* el *dataframe* con los árboles detectados en el



**Figura 12.13.** Árboles de la parcela de escaneos múltiples de TLS (“GaliciaMultiScan.laz”) con una representación en rojo de los diámetros normales estimados para los árboles detectados.

argumento *data* y especificar los diseños de parcela en los que se está interesado (*plot.design* = c(“fixed.area”, “k.tree”, “angle.count”)), así como sus parámetros (radio, k y BAF) mediante el argumento *plot.parameters* (ver el código de ejemplo accediendo al repositorio mediante el QR incluido al final del capítulo). Aquí se recomienda que la distancia máxima (argumento *max.dist*) definida en la función *normalize* sea algo mayor que el radio considerado en la parcela circular, ya que se incrementará la posibilidad de detectar los árboles del borde. En el caso de la parcela relascópica, cuanto mayor sea la distancia máxima establecida en la función *normalize* mejor, ya que, en este caso, la selección de los árboles no sólo se basa en su distancia al centro de la parcela, sino también en el tamaño de los árboles. Habrá que especificar en los argumentos el tipo de escaneo, que por defecto se entenderá como un escaneo único de TLS; en el caso de escaneos múltiples de TLS o de tecnología SLAM sí será necesario especificarlo (*scan.approach* = “multi”). Hay que tener en cuenta que las metodologías implementadas en FORTLS para corregir el efecto de las oclusiones en las estimaciones solamente se utilizarán para la casuística del escaneo único de TLS. En el repositorio (ver QR al final del capítulo para acceder al mismo) se muestra un ejemplo del código que habría que utilizar incluyendo algunos de los argumentos más relevantes.

El *output* de esta función será una lista con hasta tres elementos de tipo *dataframe*, uno por diseño de parcela considerado, que contendrá los valores estimados de las variables (Tabla 12.5) y métricas (Tabla 12.6) de masa. La función también está diseñada para lidiar con varias parcelas si estas están incluidas en los datos de entrada suministrados en el argumento *tree.tls*.

**Tabla 12.5.** Principales variables de masa obtenidas con FORTLS.

Variables	Descripción
<i>N.tls</i>	Densidad de árboles ( $N$ , árboles $\text{ha}^{-1}$ )
<i>G.tls</i>	Área basimétrica ( $G$ , $\text{m}^2 \text{ha}^{-1}$ )
<i>V.tls</i>	Volumen de los fustes ( $V$ , $\text{m}^3 \text{ha}^{-1}$ )
<i>V.com.tls</i>	Volumen maderable hasta un diámetro en punta delgada determinado ( $V_c$ , $\text{m}^3 \text{ha}^{-1}$ )
<i>h.com.tls</i>	Longitud de los fustes maderables hasta un diámetro en punta delgada determinado ( $h_c$ , $\text{m ha}^{-1}$ )
<i>N.hn*</i> , <i>N.hr*</i> , <i>N.hn.cov*</i> , <i>N.hr.cov*</i> , <i>G.hn*</i> , <i>G.hr*</i> , <i>G.hn.cov*</i> , <i>G.hr.cov*</i> , <i>V.hn*</i> , <i>V.hr*</i> , <i>V.hn.cov*</i> , <i>V.hr.cov*</i>	Estimaciones corregidas para $N$ , $G$ y $V$ basadas en las metodologías de muestreo en la distancia. En el caso de $V_c$ y $h_c$ , se añadirá la terminación “.com” (p. ej., $N.com.hn$ )
<i>N.sh*</i> , <i>G.sh*</i> , <i>V.sh*</i> , <i>V.com.sh*</i> , <i>h.com.sh*</i>	Estimaciones corregidas para $N$ , $G$ , $V$ , $V_c$ y $h_c$ según las áreas de sombra generadas por las oclusiones
<i>N.pam**</i> , <i>G.pam**</i> , <i>V.pam**</i> , <i>V.com.pam**</i> , <i>h.com.pam**</i>	Estimaciones corregidas para $N$ , $G$ , $V$ , $V_c$ y $h_c$ basadas en una probabilidad de detección de los árboles según una distribución de Poisson
<i>d.tls</i> , <i>dg.tls</i> , <i>dgeom.tls</i> , <i>dharm.tls</i>	Diámetro normal medio ( $dbh$ , cm) aritmético, cuadrático ( $d_g$ ), geométrico y armónico, respectivamente
<i>h.tls</i> , <i>hg.tls</i> , <i>hgeom.tls</i> , <i>hharm.tls</i>	Altura media ( $h$ , m) aritmética, cuadrática, geométrica y armónica, respectivamente.
<i>d.0.tls</i> , <i>dg.0.tls</i> , <i>dgeom.0.tls</i> , <i>dharm.0.tls</i>	Diámetro normal dominante medio ( $D_o$ , cm) aritmético, cuadrático, geométrico y armónico, respectivamente
<i>h.0.tls</i> , <i>hg.0.tls</i> , <i>hgeom.0.tls</i> , <i>hharm.0.tls</i>	Altura dominante media ( $H_o$ , m) aritmética, cuadrática, geométrica y armónica, respectivamente

\* Variables solamente estimadas para parcelas circulares de área fija y *k-tree* para la casuística de escaneos únicos de TLS. Para más información sobre estas variables ver Molina-Valero *et al.* (2022).

\*\* Variables solamente estimadas para parcelas relascópicas (o de conteo angular) para la casuística de escaneos únicos de TLS. Para más información sobre estas variables ver Molina-Valero *et al.*, (2022).

**Tabla 12.6.** Principales métricas de masa obtenidas con FORTLS.

Métricas	Descripción
<i>n.pts</i> , <i>n.pts.est</i> , <i>n.pts.red</i> , <i>n.pts.red.est</i>	Número de puntos total y estimados correspondientes a las secciones de los diámetros normales de los árboles, tanto para las nubes de puntos originales ( <i>n.pts</i> y <i>n.pts.est</i> ) como reducidas en densidad ( <i>n.pts.red</i> y <i>n.pts.red.est</i> )
<i>P01</i> , ..., <i>P99</i>	Percentiles (m) de la distribución de la coordenada z
<i>mean.arit.z/rho/r</i> , <i>mean.qua.z/rho/r</i> , <i>mean.geom.z/rho/r</i> , <i>mean.harm.z/rho/r</i> , <i>median.z.rho/r</i> , <i>mode.arit.z/rho/r</i>	Estadísticos de tendencia central de las coordenadas z, $\rho$ (distancia horizontal) y r (distancia euclídea): medias aritmética, cuadrática, geométrica y armónica, mediana y moda, respectivamente
<i>var.z/rho/r</i> , <i>sd.z/rho/r</i> , <i>cv.z/rho/r</i> , <i>d.z/rho/r</i> , <i>id.z/rho/r</i> , <i>max.z/rho/r</i> , <i>min.z/rho/r</i>	Estadísticos de dispersión de las coordenadas z, $\rho$ y r: varianza, desviación estándar, coeficiente de varianza, rango, rango intercuartílico, máximo y mínimo, respectivamente
<i>skewness.z/rho/r</i> , <i>kurtosis.z/rho/r</i>	Coefficientes de asimetría y curtosis de la distribución de las coordenadas z, $\rho$ y r
<i>L2.z/rho/r</i> , <i>L3.z/rho/r</i> , <i>L4.z/rho/r</i> , <i>L.CV.z/rho/r</i> , <i>L.skewness.z/rho/r</i> , <i>L.kurtosis.z/rho/r</i>	L-momentos de orden 2 ( $\lambda_2$ ), 3 ( $\lambda_3$ ) y 4 ( $\lambda_4$ ) de las coordenadas z, $\rho$ y r. L-momentos ratios de orden 2 ( $\tau$ , L-CV, coeficiente de L-Variación = $\lambda_2/\lambda_1$ ); 3 ( $\tau_3$ , L-Asimetría = $\lambda_3/\lambda_2$ ); y 4 ( $\tau_4$ , L-Curtosis = $\lambda_4/\lambda_2$ ) de las coordenadas z, $\rho$ y r
<i>median.a.d.z/rho/r</i> , <i>mode.a.d.z/rho/r</i>	Mediana de las desviaciones absolutas respecto de la media aritmética y moda de las desviaciones absolutas respecto de la media aritmética, de las coordenadas z, $\rho$ y r
<i>p.a.mean.z/rho/r</i> , <i>p.a.mode.z/rho/r</i> , <i>p.a.2m</i>	Porcentaje de puntos con valores mayores a la media aritmética y moda para las coordenadas z, $\rho$ y r. Porcentaje de puntos con alturas superiores a 2 m
<i>p.b.mean.z/rho/r</i> , <i>p.b.mode.z/rho/r</i> , <i>p.b.2m</i>	Porcentaje de puntos con valores menores a la media aritmética y moda para las coordenadas z, $\rho$ y r. Porcentaje de puntos con alturas menores a 2 m
<i>weibull.b.z/rho/r</i> , <i>weibull.c.z/rho/r</i>	Parámetros de escala y forma de una función de distribución de Weibull ajustada a los histogramas de las coordenadas z, $\rho$ y r.
<i>CRR.z/rho/r</i>	<i>Canopy relief ratio</i> (media aritmética / máximo) para las coordenadas z, $\rho$ y r

## Métricas

Con la misma función utilizada anteriormente (*metrics.variables*) se obtiene otro tipo de información, que ha sido denominado como métricas por su similitud con las métricas LiDAR obtenidas con los *softwares* comúnmente empleados con datos de LiDAR aerotransportado (FUSION/LDV, LAStools, lidR, etc.). Estas métricas son, en mayor medida, estadísticos descriptivos de las coordenadas ( $z$ ,  $\rho$ ,  $r$ ) de las nubes de puntos normalizadas (Tabla 12.4). Estas métricas se pueden utilizar como potenciales variables predictoras en un modelo, por lo que ofrecen la posibilidad de modelizar el comportamiento de variables respuesta de interés a nivel de masa (p. ej., la biomasa estimada en base a ecuaciones alométricas).

## Rutina para automatizar el flujo de trabajo con varias parcelas

Antes de pasar a la parte de optimización del diseño de parcela, es conveniente hacer un inciso sobre cómo se trabaja cuando hay varias parcelas, ya que será lo más habitual en inventarios forestales. Para ello, se utilizará la función *tree.detection.several.plots*, que es capaz de procesar varias parcelas en bucle. Esta función combina los procesos de normalización y detección de árboles en una sola función. Los *inputs* y *outputs* serán similares a los anteriormente explicados pero extendidos al caso genérico de tener varias parcelas. El argumento para suministrar las nubes de puntos originales (*las.list*) debe contener un vector con el nombre de todas las nubes de puntos que se quieran procesar y estén en el directorio especificado en *dir.data*. Los códigos identificadores de las parcelas se pueden indicar mediante el argumento *id.list*, que deberá contener un vector con los códigos en el mismo orden que el de las nubes de puntos especificadas en el argumento *las.list*. En cualquier caso, este argumento es opcional; en caso de no especificarse, las parcelas serán codificadas con números correlativos desde 1 hasta las  $n$  parcelas. Las coordenadas del centro de las parcelas se especificarán mediante el argumento *center.coord*, al que se le deberá suministrar un *dataframe* con tres columnas ( $id$  = identificador de parcela,  $x = x.center$  e  $y = y.center$ ), con tantas filas como parcelas a procesar. En caso de no especificar el identificador de parcela, la función entenderá que las coordenadas fueron suministradas en el mismo orden que las nubes de puntos en el argumento *las.list*. El resto de los argumentos serán idénticos a los que ya se explicaron previamente en las funciones de normalización y detección de árboles. Si, además, se está interesado en las metodologías de muestreo basado en la distancia para corregir las estimaciones por las oclusiones generadas (Molina-Valero *et al.*, 2022), será necesario el previo uso de la función *distance.sampling*. Estas metodologías suelen utilizarse cuando hay varias parcelas, ya que para una sola parcela muchas veces no se consiguen ajustar las funciones de probabilidad de detección de árboles.

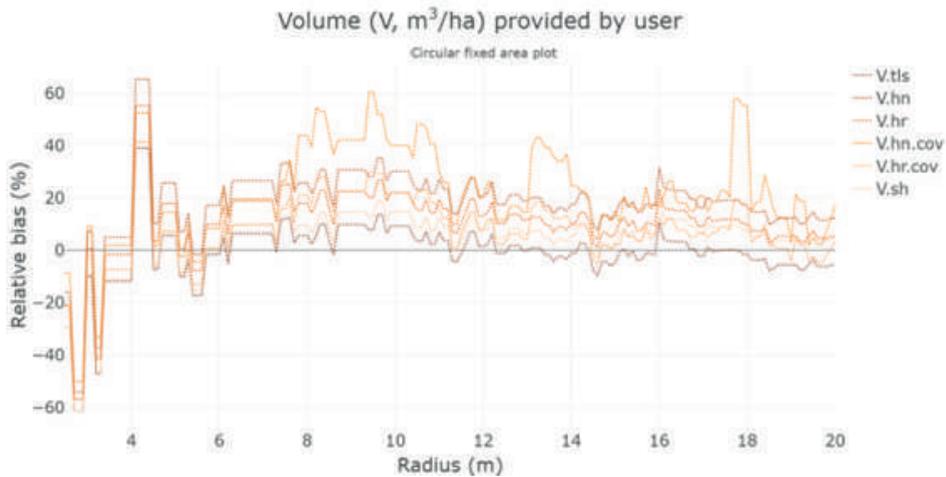
En el repositorio (ver QR al final del capítulo para acceder al mismo) se muestra un ejemplo del código que habría que utilizar incluyendo algunos de los argumentos más relevantes.

## Optimización del diseño de parcela

Una peculiaridad de FORTLS es la posibilidad de calibrar los diferentes diseños de parcela cuando se dispone de datos de control (p. ej., mediciones convencionales de campo). Esto permite determinar diseños de parcela más adecuados en base a una submuestra de datos pareados (control~escáner láser terrestre), tanto a nivel de sesgo relativo como de correlaciones. El primer paso consiste en aplicar la función *simulations* para simular el comportamiento de las estimaciones de variables de masa a medida que aumenta el tamaño de parcela para cada uno de los diseños considerados. Esta función puede tener como *inputs* hasta tres bases de datos, correspondientes a los árboles detectados de las nubes de puntos con FORTLS u otro *software* siempre que esté en el formato adecuado (argumento *tree.tls*), los datos de control (argumento *tree.field*) y, en caso de considerar las correcciones de las estimaciones basadas en las metodologías de muestreo a distancia, el *output* de la función *distance.sampling* (argumento *tree.ds*). La base de datos de control debe contener, al menos, las siguientes columnas: *id*, identificador de la parcela (coincidente con el homólogo de los árboles detectados en la nube de puntos), *tree*, número identificador del árbol; *h.dist*, distancia horizontal desde el centro de la parcela hasta el árbol; *dbh*, diámetro normal del árbol en cm; *h*, altura del árbol en m; *dead*, identificador de si el árbol está muerto (1) o no (NA). A mayores, se puede especificar el volumen (*v.user*) y la biomasa (*w.user*) estimados, por ejemplo sobre la base de ecuaciones alométricas de árbol individual. Otro argumento que es importante especificar es hasta qué valores de radio, *k* y BAF se quieren hacer simulaciones para las parcelas circulares de área fija, *k-tree* y relascópicas, respetivamente (argumento *plot.parameters*). El *output* de esta función será una lista con hasta 3 elementos, uno por tipo de parcela considerado, que contendrá los valores estimados de todas las simulaciones realizadas. Las simulaciones por defecto se realizan a razón de incrementos regulares de 0,1 m, 1 k y 0,1 BAF, hasta el valor máximo especificado para cada uno de estos parámetros. No obstante, estos incrementos se podrán modificar mediante el argumento *plot.parameters* (ver el manual para más información). Igual que en pasos anteriores, estas simulaciones se guardarán de forma automática como archivos *.csv* (uno por tipo de parcela) en el directorio especificado en *dir.result*. En el repositorio (ver QR al final del capítulo para acceder al mismo) se muestra un ejemplo del código que habría que utilizar incluyendo algunos de los argumentos más relevantes.

Una vez que se han realizado las simulaciones, lo primero que se puede evaluar es el sesgo relativo porcentual promedio cometido en las estimaciones de las variables de masa. Este sesgo se estima comparando los datos de control (*tree.field*) con los estimados a partir de las nubes de puntos (*tree.tls*) a medida que aumenta el tamaño de parcela, de modo que valores positivos indican sobrestimación con respecto a los datos de control, y valores negativos lo contrario. Para ello, se utiliza la función *relative.bias*, cuyos principales *outputs* serán gráficos en formato *.html* (guardados en el directorio especificado en *dir.result*) representando la evolución del sesgo relativo (Figura 12.14). En esta figura se muestra un ejemplo para la estimación del volumen de madera basado en

las parcelas procesadas anteriormente, donde puede observarse cómo parcelas pequeñas generan estimaciones más inestables, con picos para los valores de sesgo generalmente más acusados. A su vez, se aprecia cómo a partir de un determinado valor de radio (8 m aproximadamente) el comportamiento empieza a ser más estable, alcanzando zonas en las que el sesgo tiende a cero. Sin embargo, a medida que el radio aumenta ( $> 16$  m), el sesgo empieza a mostrar una tendencia negativa debido a que la ratio de detección de árboles disminuye, generando, por lo tanto, subestimaciones. En resumen, este análisis podría ser útil para elegir el diseño de parcela en el que se comete un menor sesgo y replicarlo con más parcelas medidas con TLS u otros dispositivos.

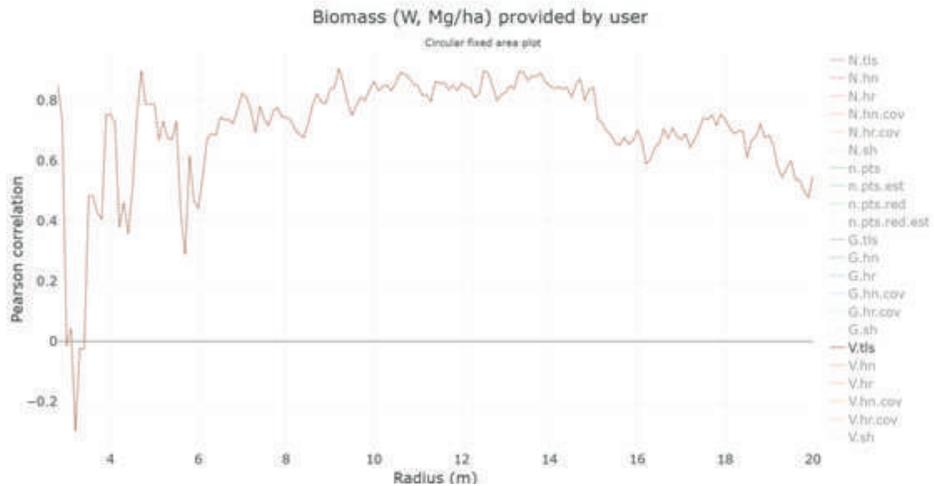


**Figura 12.14.** Sesgo relativo cometido en la estimación del volumen de madera ( $V$ ) incluyendo estimaciones de metodologías que corrigen las estimaciones debido a las oclusiones generadas por los árboles ( $V_{hn}$ ,  $V_{hr}$ ,  $V_{hn.cov}$ ,  $V_{hr.cov}$ ,  $V_{sh}$ ). El eje de ordenadas indica el valor del sesgo relativo (%) y el eje de abscisas el valor del radio (m) de la parcela circular de área fija.

Aunque se ha mostrado el ejemplo de la parcela circular de área fija, se obtendrán resultados análogos para el resto de los diseños de parcela incluidos en FORTLS (*k-tree* y relascópica). En el repositorio (ver QR al final del capítulo para acceder al mismo) se muestra el código utilizado para estimar el sesgo relativo.

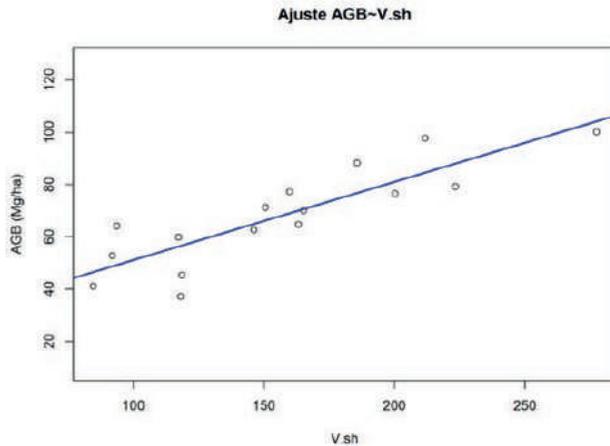
Otra forma de evaluar las simulaciones obtenidas es a partir de las correlaciones con los datos de control. Para ello, se utilizará la función *correlation*, que calculará las correlaciones de Pearson y Spearman entre las variables de interés estimadas con los datos de control y todo el conjunto de variables y métricas obtenidas con FORTLS. Al igual que antes, se generarán gráficos en formato *.html* (guardados en el directorio especificado en *dir.result*) representando la evolución de las correlaciones a medida que aumenta el tamaño de parcela (Figura 12.15). En este caso, se muestra un ejemplo para un muestreo sistemático realizado

en un pinar de *Pinus sylvestris*, ya que para estimar bien las correlaciones es mejor disponer de un gran número de datos. Este ejemplo se basó en dieciséis parcelas medidas y escaneadas con TLS (escaneos únicos ubicados en el centro de las parcelas). Estos datos están incluidos en el paquete FORTLS como datos de ejemplo, y ya se encuentran procesados. El ejemplo aquí mostrado se basa en la correlación entre la biomasa estimada con ecuaciones de árbol individual y las métricas y variables obtenidas con FORTLS. En la Figura 12.15 se ha seleccionado una de las variables (*V.tls*) que mejor correlación mostró con la biomasa; para tamaños de parcela pequeños mostró inestabilidad en las correlaciones de Pearson, y mayor estabilidad a partir de un radio de 9 m, con correlaciones superiores a 0,8 hasta aproximadamente un radio de 15 metros.



**Figura 12.15.** Correlaciones de Pearson a través de un incremento continuo del tamaño de parcela entre la biomasa estimada en base a los datos de control y una de las variables estimadas con FORTLS (*V.tls*). En este caso el eje de ordenadas indica el valor de la correlación de Pearson para la variable de interés biomasa, y el eje de abscisas el valor del radio (m) de la parcela circular.

En el repositorio (ver QR al final del capítulo para acceder al mismo) se muestra el código con el que se obtuvo la Figura 12.15 en base a las dieciséis parcelas de muestreo incluidas en los datos de ejemplo de FORTLS. El estudio de las correlaciones puede ser de gran utilidad a la hora de establecer un diseño de parcela que permita modelizar el comportamiento de las variables de interés con toda la información auxiliar obtenida con FORTLS (variables y métricas). De este modo, a partir de una submuestra representativa de un bosque a inventariar en la cual se tengan datos pareados (control-escáner láser terrestre), se podría ajustar un modelo adecuado para estimar una variable de interés forestal. Estos modelos permitirían implementar técnicas de inferencia asistidas o basadas en modelos (McRoberts *et al.*, 2014), tal y como se aplican con otro tipo de información de teledetección (p. ej., LiDAR aéreo). Un ejemplo para un modelo de regresión lineal simple puede observarse en la Figura 12.16.



**Figura 12.16.** Ajuste de regresión lineal simple entre la variable respuesta biomasa ( $AGB$ ,  $Mg\ ha^{-1}$ ), estimada en base a datos de campo y ecuaciones alométricas, y la variable explicativa volumen ( $V.sh$ ) estimada con FORTLS. En este ejemplo se seleccionaron parcelas circulares de área fija de 10 m de radio.

## Bibliografía

- Alba, M., Roncoroni, F., Scaioni, M., 2008. Investigations about the accuracy of target measurement for deformation monitoring. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* 37, 1053–1060.
- Alvites, C., Santopuoli, G., Hollaus, M., Pfeifer, N., Maesano, M., Moresi, F.V., Marchetti, M., Lasserre, B., 2021. Terrestrial laser scanning for quantifying timber assortments from standing trees in a mixed and multi-layered mediterranean forest. *Remote Sens.* 13, 4265.
- Ariza Salamanca, A., Molina-Valero, J., 2022. Estimación de parámetros forestales mediante LiDAR Terrestre: Aplicaciones en R. <https://doi.org/10.13140/RG.2.2.11947.52007>
- Bienert, A., Scheller, S., Keane, E., Mullooly, G., Mohan, F., 2006. Application of terrestrial laser scanners for the determination of forest inventory parameters. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* 36, 1–5.
- Burt, A., Disney, M., Calders, K., 2019. Extracting individual trees from lidar point clouds using tree-seg. *Methods Ecol. Evol.* 10, 438–445.
- Calders, K., Adams, J., Armston, J., Bartholomeus, H., Bauwens, S., Bentley, L.P., Chave, J., Danson, F.M., Demol, M., Disney, M., Gaulton, R., Krishna Moorthy, S.M., Levick, S.R., Saarinen, N., Schaaf, C., Stovall, A., Terry, L., Wilkes, P., Verbeeck, H., 2020. Terrestrial laser scanning in forest ecology: Expanding the horizon. *Remote Sens. Environ.* 251, 112102.
- Crespo-Peremarch, P., Torralba, J., Carbonell-Rivera, J.P., Ruiz, L.A., 2020. Comparing the generation of DTM in a forest ecosystem using TLS, ALS and UAV-DAP, and different software tools. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* 43, 575–582.

- Dassot, M., Constant, T., Fournier, M., 2011. The use of terrestrial LiDAR technology in forest science: application fields, benefits and challenges. *Ann. For. Sci.* 68, 959–974.
- de Conto, T., Olofsson, K., Görgens, E.B., Rodriguez, L.C.E., Almeida, G., 2017. Performance of stem denoising and stem modelling algorithms on single tree point clouds from terrestrial laser scanning. *Comput. Electron. Agric.* 143, 165–176.
- Demol, M., Verbeeck, H., Gielen, B., Armston, J., Burt, A., Disney, M., Duncanson, L., Hackenberg, J., Kükenbrink, D., Lau, A., Ploton, P., Sewdien, A., Stovall, A., Takoudjou, S.M., Volkova, L., Weston, C., Wortel, V., Calders, K., 2022. Estimating forest above-ground biomass with terrestrial laser scanning: Current status and future directions. *Methods Ecol. Evol.* 13, 1628–1639.
- Dijkstra, E.W., 2022. A Note on Two Problems in Connexion with Graphs, in: Apt, K.R., Hoare, T. (Eds.), *Edsger Wybe Dijkstra*. ACM, New York, NY, USA, pp. 287–290.
- Guzmán, J.A., Hernandez, R., Sanchez-Azofeifa, A., 2021. rTLS: Tools to Process Point Clouds Derived from Terrestrial Laser Scanning. R package version 0.2.3.1. <https://CRAN.R-project.org/package=rTLSI>
- Hackenberg, J., Morhart, C., Sheppard, J., Spiecker, H., Disney, M., 2014. Highly Accurate Tree Models Derived from Terrestrial Laser Scan Data: A Method Description. *Forests* 5, 1069–1105.
- Hackenberg, J., Wassenberg, M., Spiecker, H., Sun, D., 2015. Non destructive method for biomass prediction combining TLS derived tree volume and wood density. *Forests* 6, 1274–1300.
- Illingworth, J., Kittler, J., 1987. The adaptive Hough transform. *IEEE Trans. Pattern Anal. Mach. Intell.* 690–698.
- Klapa, P., Mitka, B., Zygmunt, M., 2022. Integration of TLS and UAV data for the generation of a three-dimensional basemap. *Adv. Geod. Geoinformation e27–e27*.
- Krok, G., Kraszewski, B., Stereńczak, K., 2020. Application of terrestrial laser scanning in forest inventory—an overview of selected issues. *For. Res. Pap.*, 81 (4), 175-194, 1
- Liang, X., Hyypä, J., Kaartinen, H., Lehtomäki, M., Pyörälä, J., Pfeifer, N., Holopainen, M., Brolly, G., Francesco, P., Hackenberg, J., 2018. International benchmarking of terrestrial laser scanning approaches for forest inventories. *ISPRS J. Photogramm. Remote Sens.* 144, 137–179.
- Liang, X., Kankare, V., Hyypä, J., Wang, Y., Kukko, A., Haggrén, H., Yu, X., Kaartinen, H., Jaakkola, A., Guan, F., 2016. Terrestrial laser scanning in forest inventories. *ISPRS J. Photogramm. Remote Sens.* 115, 63–77.
- Maeda, E.E., Nunes, M.H., Calders, K., Moura, Y.M. de, Raunonen, P., Tuomisto, H., Verley, P., Vincent, G., Zuquim, G., Camargo, J.L., 2022. Shifts in structural diversity of Amazonian forest edges detected using terrestrial laser scanning. *Remote Sens. Environ.* 271, 112895.
- McRoberts, R.E., Andersen, H.-E., Næsset, E., 2014. Using Airborne Laser Scanning Data to Support Forest Sample Surveys, in: Maltamo, M., Næsset, E., Vauhkonen, J. (Eds.), *Forestry Applications of Airborne Laser Scanning, Managing Forest Ecosystems*. Springer Netherlands, Dordrecht, pp. 269–292.

- Molina-Valero, J.A., Martínez-Calvo, A., Villamayor, M.J.G., Pérez, M.A.N., Álvarez-González, J.G., Montes, F., Pérez-Cruzado, C., 2022. Operationalizing the use of TLS in forest inventories: The R package FORTLS. *Environ. Model. Softw.* 150, 105337.
- Moskal, L.M., Zheng, G., 2011. Retrieving forest inventory variables with terrestrial laser scanning (TLS) in urban heterogeneous forest. *Remote Sens.* 4, 1–20.
- Nguyen, V.-T., Constant, T., Colin, F., 2021. An innovative and automated method for characterizing wood defects on trunk surfaces using high-density 3D terrestrial LiDAR data. *Ann. For. Sci.* 78, 32.
- Nunes, M.H., Camargo, J.L.C., Vincent, G., Calders, K., Oliveira, R.S., Huete, A., Mendes de Moura, Y., Nelson, B., Smith, M.N., Stark, S.C., Maeda, E.E., 2022. Forest fragmentation impacts the seasonality of Amazonian evergreen canopies. *Nat. Commun.* 13, 917.
- Prendes, C., Canga, E., Ordoñez, C., Majada, J., Acuna, M., Cabo, C., 2022. Automatic assessment of individual stem shape parameters in forest stands from TLS point clouds: application in *Pinus pinaster*. *Forests* 13, 431.
- Pyörälä, J., Kankare, V., Liang, X., Saarinen, N., Rikala, J., Kivinen, V.-P., Sipi, M., Holopainen, M., Hyypä, J., Vastaranta, M., 2019. Assessing log geometry and wood quality in standing timber using terrestrial laser-scanning point clouds. *For. Int. J. For. Res.* 92, 177–187.
- Ramos, L., Marchamalo, M., Rejas, J.G., Martínez, R., 2015. Aplicación del Láser Escáner Terrestre (TLS) a la modelización de estructuras: precisión, exactitud y diseño de la adquisición de datos en casos reales. *Inf. Constr.* 67, e074–e074.
- Raunonen, P., Kaasalainen, M., Åkerblom, M., Kaasalainen, S., Kaartinen, H., Vastaranta, M., Holopainen, M., Disney, M., Lewis, P., 2013. Fast automatic precision tree models from terrestrial laser scanner data. *Remote Sens.* 5, 491–520.
- Salehi, V., Wang, S., 2017. Using point cloud technology for process simulation in the context of digital factory based on a systems engineering integrated approach, in: *DS 87-3 Proceedings of the 21st International Conference on Engineering Design (ICED 17) Vol 3: Product, Services and Systems Design*, Vancouver, Canada, 21–25.08. 2017. pp. 011–020.
- Simonse, M., Aschoff, T., Spiecker, H., Thies, M., 2003. Automatic determination of forest inventory parameters using terrestrial laser scanning, in: *Proceedings of the Scandlaser Scientific Workshop on Airborne Laser Scanning of Forests*. Citeseer, pp. 252–258.
- Terry, L., Calders, K., Åkerblom, M., Bartholomeus, H., Disney, M., Levick, S., Origo, N., Raunonen, P., Verbeeck, H., 2023. Analysing individual 3D tree structure using the R package ITSM. *Methods Ecol. Evol.* 14, 231–241.
- Torresan, C., Chiavetta, U., Hackenberg, J., 2018. Applying quantitative structure models to plot-based terrestrial laser data to assess dendrometric parameters in dense mixed forests. *For. Syst.* 27, 1–15.
- Vincent, G., Antin, C., Laurans, M., Heurtebize, J., Durrieu, S., Lavalley, C., Dauzat, J., 2017. Mapping plant area index of tropical evergreen forest by airborne laser scanning. A cross-validation study using LAI2200 optical sensor. *Remote Sens. Environ.* 198, 254–266.
- Wang, D., Momo Takoudjou, S., Casella, E., 2020. LeWoS: A universal leaf-wood classification method to facilitate the 3D modelling of large tropical trees using terrestrial LiDAR. *Methods Ecol. Evol.* 11, 376–389.

- Wilkes, P., Lau, A., Disney, M., Calders, K., Burt, A., Gonzalez de Tanago, J., Bartholomeus, H., Brede, B., Herold, M., 2017. Data acquisition considerations for Terrestrial Laser Scanning of forest plots. *Remote Sens. Environ.* 196, 140–153.
- Xu, H., Li, H., Yang, X., Qi, S., Zhou, J., 2018. Integration of terrestrial laser scanning and nurbs modeling for the deformation monitoring of an earth-rock dam. *Sensors* 19, 22.
- Zhang, W., Qi, J., Wan, P., Wang, H., Xie, D., Wang, X., Yan, G., 2016. An easy-to-use airborne LiDAR data filtering method based on cloth simulation. *Remote Sens.* 8, 501.
- Zheng, G., Moskal, L.M., 2012. Leaf orientation retrieval from terrestrial laser scanning (TLS) data. *IEEE Trans. Geosci. Remote Sens.* 50, 3970–3979.
- Zheng, G., Moskal, L.M., Kim, S.-H., 2012. Retrieval of effective leaf area index in heterogeneous forests with terrestrial laser scanning. *IEEE Trans. Geosci. Remote Sens.* 51, 777–786.

[Acceso al  
material complementario](#)

