- TESIS DOCTORAL -

Algoritmos para la detección automática de elementos en entornos viales a partir de datos de escáneres láser móviles

[Algorithms for automatic feature detection from Mobile Laser Scanning data]

Carlos Cabo Gómez

Universidad de Oviedo

Minería, Obra Civil, Medio Ambiente y Dirección de Proyectos

Vicerrectorado de Organización Académica

Universidad de Oviedo
Universidá d´Uviéu
University of Oviedo

Vicerrectoráu d'Organización Académica
Vice-rectorate for Academic Organization

RESUMEN DEL CONTENIDO DE TESIS DOCTORAL

1.- Título de la Tesis	
Español/Otro Idioma: Algoritmos para la detección automática de elementos en entornos viales a partir de datos de escáneres láser móviles	Inglés: Algorithms for automatic feature detection from Mobile Laser Scanning data

2.- Autor	
Nombre: CARLOS CABO GOMEZ	DNI/Pasaporte/NIE: [redacted]
Programa de Doctorado: Minería, obra civil, medio ambiente y dirección de proyectos.	
Órgano responsable: EXPLOTACION Y PROSPECCION DE MINAS	

RESUMEN (en español)

Los sistemas de toma masiva de datos, basados tanto en imágenes como en tecnología LiDAR (del inglés, Light Detection and Ranging) recogen información sobre el terreno que, en la mayor parte de los casos, es susceptible de ser interpretada visualmente. Sin embargo, esta inspección visual tiene unos requerimientos de tiempo y recursos que hacen que a menudo no resulte rentable el uso de esta tecnología. Por esta razón existe una clara tendencia a la automatización de la interpretación e identificación de elementos, que se concreta en la creación de algoritmos que realicen estos procesos a partir de imágenes y nubes de puntos. La elaboración de estos algoritmos consiste básicamente en (i) la formulación de abstracciones rigurosas de los elementos buscados (es decir: características y atributos que permiten su diferenciación del resto), y (ii) la parte computacional, que consiste en la transformación de las abstracciones en instrucciones que pueden ser ejecutadas por un ordenador.

Entre los sistemas de toma masiva de datos, cabe destacar el auge en los últimos años de los escáneres láser terrestres, y en especial de los escáneres láser móviles. Estos sistemas recogen cientos de miles de puntos por segundo sobre los objetos situados en el entorno por el que circulan, resolviendo alguna de las limitaciones de los escáneres láser terrestres estáticos y los LiDAR aéreos. Habitualmente, los escáneres láser móviles proporcionan nubes de puntos con precisiones centimétricas, con una densidad y distribución de puntos muy variable (en función de las distancias y los ángulos de incidencia), y que sufren los efectos de los ocultamientos propios de la perspectiva desde un punto móvil y, en general, cercano a la superficie/suelo.

En esta tesis se desarrollan dos estructuras de datos que mitigan los efectos de la gran cantidad de puntos registrados y de la heterogeneidad de su distribución: (i) voxelización, y (ii) nubes de líneas. Estas dos estructuras permiten la creación de versiones reducidas y homogeneizadas de la nube de puntos, y presentan la cualidad de ser reversibles. Es decir, si se transforma la nube de puntos en cualquiera de estas dos estructuras, se genera una versión simplificada sobre la que se aplican los algoritmos de detección automática. Posteriormente, se etiquetan las estructuras simplificadas (vóxeles y/o líneas) y se deshace la transformación para recuperar los puntos de la nube original, que heredan las propiedades o etiquetas asignadas a las estructuras simplificadas.

En base a las dos estructuras de simplificación y homogeneización propuestas, se desarrollan tres algoritmos de detección automática de elementos: (i) un algoritmo de detección automática de postes y los elementos situados sobre ellos (como señales, semáforos o farolas), (ii) un algoritmo de detección automática de superficies verticales (como fachadas, muros o paneles), y (iii) un algoritmo para la determinación automática del límite de la superficie asfaltada en calles y carreteras. Se concretan y especifican para cada uno de los algoritmos (i) las abstracciones utilizadas, (ii) el desarrollo computacional, y (iii) los resultados obtenidos sobre una serie de nubes de puntos de prueba.

En las pruebas de validación, el algoritmo de detección de postes es capaz de detectar más de un 92% de los objetivos, siendo realmente postes el 84% de los elementos detectados. En lo que respecta al segundo algoritmo, más de un 90% de los puntos pertenecientes a cada una de las 27 superficies analizadas son asignados a ellas, y el 99,9% de los puntos asignados a cada superficie, realmente pertenecen a ella. El tercer algoritmo es capaz de determinar correctamente más de un 97% de la superficie asfaltada, y un 99% de la superficie delineada por el algoritmo corresponde a superficie asfaltada real. Los tres algoritmos resuelven muchas de las desventajas de algoritmos anteriores, y los superan en los índices que determinan la calidad de la detección y los errores cometidos.

RESUMEN (en Inglés)

Mass data collection systems, both based on images and LiDAR (Light Detection and Ranging), gather field information that can often be visually interpreted. However, this visual inspection is very time and resource intensive, to a point that frequently it is not worth using this technology. For this reason, there is an increasing trend towards the automatization of feature interpretation and identification processes, which are materialized through the creation of algorithms that perform these tasks from images and point clouds. The design of these algorithms consists of: (i) the formulation of accurate abstractions of the goal features (i.e. attributes and descriptors that allow their identification and separation from other features), and (ii) the computation or implementation, involving the transformation of the aforesaid abstractions into a set of computer instructions.

In the last few years, the use of terrestrial laser scanners have increased substantially, especially with respect to mobile terrestrial laser scanners. These systems gather hundreds of thousands of points per second on the objects and surfaces around them while the mobile platform is moving, and they overcome some of the disadvantages of the static terrestrial laser scanners and aerial LiDAR. Usually, mobile laser scanners provide point clouds with centimetric accuracy, with a highly variable distribution and point density (according to distances and incidence angles to the surfaces and objects). The point clouds are affected by the occlusions caused by the perspective obtained from a specific mobile point, which is usually close to the ground.

In this PhD thesis, two data structures for reducing the effects of the large amount of data and the heterogeneity of the point distribution are developed: (i) voxelization, and (ii) line clouds. These two structures allow the creation of reduced and homogenized versions of the point cloud, and have the advantage of being reversible. That is, the transformation of a point cloud into any of these structures produces a simplified version of the data on which the feature detection algorithms can be applied. Afterwards, the simplified structures (voxels and/or lines) are labelled, and the transformation is reversed in order to recover the original points. These points inherit the labels from the simplified structures.

Three algorithms for automatic feature detection are developed based on the two simplification and homogenization structures: (i) an algorithm for automatic detection of pole-like street furniture objects (e.g. traffic signs, traffic lights or lampposts), (ii) an algorithm for automatic detection of vertical surfaces (e.g. façades, walls or panels); and (iii) and algorithm for automatic delineation of road and street edges. For each one of the algorithms, the abstractions, implementation and validation tests are described and specified.

In the validation tests, the algorithm for automatic detection of pole-like objects is able to identify more than 92% of the goals in the test point clouds, with actual poles accounting for 84% of the detected features. For the second algorithm, the results show that more than 90% of the points belonging to each one of the 27 test surfaces are assigned to them, and 99.9% of the points assigned to each surface are correct. The third algorithm is able to delineate more than 97% of the test road surface, and 98% of the surface delineated by the algorithm belongs to the actual road. The three algorithms overcome many of the drawbacks of previous algorithms, and they surpass their performance in terms of detection and success rates.

A mis padres, hermanos, cuñados, suegros, ahijadas, primos, tíos y consortes, y a ti, abuelito.

A mis amigos, a los que no voy a nombrar aquí por no establecer un orden.

A mis directores de tesis, Celestino y Silverio.

A los finlandeses, que me acogieron como uno más.

A la tertulia del comedor del Indurot.

A ese trozo de cuerda y la gente que me salvó hace un año.

A ti, Cris.

Gracias a todos.

En aquel Imperio, el Arte de la Cartografía logró tal Perfección que el Mapa de una sola Provincia ocupaba toda una Ciudad, y el Mapa del Imperio, toda una Provincia. Con el tiempo, estos Mapas Desmesurados no satisficieron y los Colegios de Cartógrafos levantaron un Mapa del Imperio, que tenía el Tamaño del Imperio y coincidía puntualmente con él. Menos Adictas al Estudio de la Cartografía, las Generaciones Siguientes entendieron que ese dilatado Mapa era Inútil y no sin Impiedad lo entregaron a las Inclemencias del Sol y los Inviernos. En los Desiertos del Oeste perduran despedazadas Ruinas del Mapa, habitadas por Animales y por Mendigos; en todo el País no hay otra reliquia de las Disciplinas Geográficas.

Del rigor en la Ciencia. Borges.

Índice

Resumen/Abstract

Resumen

Los sistemas de toma masiva de datos, basados tanto en imágenes como en tecnología LiDAR (del inglés, Light Detection and Ranging) recogen información sobre el terreno que, en la mayor parte de los casos, es susceptible de ser interpretada visualmente. Sin embargo, esta inspección visual tiene unos requerimientos de tiempo y recursos que hacen que a menudo no resulte rentable el uso de esta tecnología. Por esta razón existe una clara tendencia a la automatización de la interpretación e identificación de elementos, que se concreta en la creación de algoritmos que realicen estos procesos a partir de imágenes y nubes de puntos. La elaboración de estos algoritmos consiste básicamente en (i) la formulación de abstracciones rigurosas de los elementos buscados (es decir: características y atributos que permiten su diferenciación del resto), y (ii) la parte computacional, que consiste en la transformación de las abstracciones en instrucciones que pueden ser ejecutadas por un ordenador.

Entre los sistemas de toma masiva de datos, cabe destacar el auge en los últimos años de los escáneres láser terrestres, y en especial de los escáneres láser móviles. Estos sistemas recogen cientos de miles de puntos por segundo sobre los objetos situados en el entorno por el que circulan, resolviendo alguna de las limitaciones de los escáneres láser terrestres estáticos y los LiDAR aéreos. Habitualmente, los escáneres láser móviles proporcionan nubes de puntos con precisiones centimétricas, con una densidad y distribución de puntos muy variable (en función de las distancias y los ángulos de incidencia), y que sufren los efectos de los ocultamientos propios de la perspectiva desde un punto móvil y, en general, cercano a la superficie/suelo.

En esta tesis se desarrollan dos estructuras de datos que mitigan los efectos de la gran cantidad de puntos registrados y de la heterogeneidad de su distribución: (i) voxelización, y (ii) nubes de líneas. Estas dos estructuras permiten la creación de versiones reducidas y homogeneizadas de la nube de puntos, y presentan la cualidad de ser reversibles. Es decir, si se transforma la nube de puntos en cualquiera de estas dos estructuras, se genera una versión simplificada sobre la que se aplican los algoritmos de detección automática. Posteriormente, se etiquetan las estructuras simplificadas (vóxeles y/o líneas) y se deshace la transformación para recuperar los puntos de la nube original, que heredan las propiedades o etiquetas asignadas a las estructuras simplificadas.

En base a las dos estructuras de simplificación y homogeneización propuestas, se desarrollan tres algoritmos de detección automática de elementos: (i) un algoritmo de detección automática de postes y los elementos situados sobre ellos (como señales,

semáforos o farolas), (ii) un algoritmo de detección automática de superficies (como fachadas, muros o paneles), y (iii) un algoritmo para la determinación automática del límite de la superficie asfaltada en calles y carreteras. Se concretan y especifican para cada uno de los algoritmos (i) las abstracciones utilizadas, (ii) el desarrollo computacional, y (iii) los resultados obtenidos sobre una serie de nubes de puntos de prueba.

En las pruebas de validación, el algoritmo de detección de postes es capaz de detectar más de un 92% de los objetivos, siendo realmente postes el 84% de los elementos detectados. En lo que respecta al segundo algoritmo, más de un 90% de los puntos pertenecientes a cada una de las 27 superficies analizadas son asignados a ellas, y el 99,9% de los puntos asignados a cada superficie, realmente pertenecen a ella. El tercer algoritmo es capaz de determinar correctamente más de un 97% de la superficie asfaltada, y un 99% de la superficie delineada por el algoritmo corresponde a superficie asfaltada real. Los tres algoritmos resuelven muchas de las desventajas de algoritmos anteriores, y los superan en los índices que determinan la calidad de la detección y los errores cometidos.

Abstract

Mass data collection systems, both based on images and LiDAR (Light Detection and Ranging), gather field information that can often be visually interpreted. However, this visual inspection is very time and resource intensive, to a point that frequently it is not worth using this technology. For this reason, there is an increasing trend towards the automatization of feature interpretation and identification processes, which are materialized through the creation of algorithms that perform these tasks from images and point clouds. The design of these algorithms consists of: (i) the formulation of accurate abstractions of the goal features (i.e. attributes and descriptors that allow their identification and separation from other features), and (ii) the computation or implementation, involving the transformation of the aforesaid abstractions into a set of computer instructions.

In the last few years, the use of terrestrial laser scanners have increased substantially, especially with respect to mobile terrestrial laser scanners. These systems gather hundreds of thousands of points per second on the objects and surfaces around them while the mobile platform is moving, and they overcome some of the disadvantages of the static terrestrial laser scanners and aerial LiDAR. Usually, mobile laser scanners provide point clouds with centimetric accuracy, with a highly variable distribution and point density (according to distances and incidence angles to the surfaces and objects). The point clouds are affected by the occlusions caused by the perspective obtained from a specific mobile point, which is usually close to the ground.

In this PhD thesis, two data structures for reducing the effects of the large amount of data and the heterogeneity of the point distribution are developed: (i) voxelization, and (ii) line clouds. These two structures allow the creation of reduced and homogenized versions of the point cloud, and have the advantage of being reversible. That is, the transformation of a point cloud into any of these structures produces a simplified version of the data on which the feature detection algorithms can be applied. Afterwards, the simplified structures (voxels and/or lines) are labelled, and the transformation is reversed in order to recover the original points. These points inherit the labels from the simplified structures.

Three algorithms for automatic feature detection are developed based on the two simplification and homogenization structures: (i) an algorithm for automatic detection of pole-like street furniture objects (e.g. traffic signs, traffic lights or lampposts), (ii) an algorithm for automatic surface detection (e.g. façades, walls or panels); and (iii) and algorithm for automatic delineation of road and street edges. For each one of the

algorithms, the abstractions, implementation and validation tests are described and specified.

In the validation tests, the algorithm for automatic detection of pole-like objects is able to identify more than 92% of the goals in the test point clouds, with actual poles accounting for 84% of the detected features. For the second algorithm, the results show that more than 90% of the points belonging to each one of the 27 test surfaces are assigned to them, and 99.9% of the points assigned to each surface are correct. The third algorithm is able to delineate more than 97% of the test road surface, and 98% of the surface delineated by the algorithm belongs to the actual road. The three algorithms overcome many of the drawbacks of previous algorithms, and they surpass their performance in terms of detection and success rates.

Capítulo I

Introducción y Objetivos

1.1 Sobre la interpretación de los datos cartográficos

La definición de la cartografía, incluso en su sentido más extenso, ha sido ampliamente discutida, y como ocurre con la descripción y acotación de muchas otras disciplinas, es probable que no sea posible una definición consensuada y universal (Jacob y Dahl, 2006; Bagrow, 2010; Taylor, 2013; Koláčný, 2013). Incluso la definición más extendida y simple es motivo de desacuerdo. Valga como ejemplo la definición de las dos acepciones que contempla la Real Academia de la Lengua Española: (i) "Arte de trazar mapas geográficos" y (ii) "Ciencia que estudia los mapas". En ambos casos se incluye el término "mapa" como objetivo y objeto esencial de este campo. Sin embargo, y aunque los procesos cartográficos han sido tradicionalmente dirigidos a la representación de datos con componente espacial, existen tareas y trabajos claramente cartográficos cuyo objetivo final no requiere la representación gráfica ni el uso de mapas (al menos en su versión o forma interpretable por el ojo humano).

Sea requerida o no la representación cartográfica, existen procesos como la toma de datos y la interpretación de los mismos, que han de tener lugar en algún momento previo a la generación del producto final (ya sea alfanumérico o gráfico). Los procesos cartográficos implican siempre de algún modo una categorización de la realidad, por lo que la información tomada en campo ha de ser interpretada. En base a esto, y en función del momento en el que se requiere la interpretación, es posible agrupar en dos categorías los métodos cartográficos según la forma de toma de datos: (i) métodos de toma selectiva, y (ii) métodos de toma masiva. En los métodos de adquisición selectiva de información (como la topografía clásica), la toma de datos se reduce a puntos concretos que se pueden categorizar y agrupar para formar líneas o polígonos, e implican la interpretación de los elementos a cartografiar durante la medición. Es decir, se eligen sobre el terreno una serie de puntos, líneas y/o polígonos, de forma que cada elemento que se pretende representar es simplificado en el momento de la toma de datos, proporcionando una versión interpretada y categorizada de la realidad. Sin embargo, los métodos de toma masiva recogen todos los datos que captan sus sensores, y estos son interpretados y clasificados a posteriori. Pertenecen a este último grupo los procesos de toma de datos LiDAR (del inglés, *Light Detection and Ranging*), de los que resultan nubes de puntos tridimensionales, y los procesos de toma de datos matriciales (imágenes), como los sensores monobanda (Ej. cámaras pancromáticas, o cámaras térmicas), multiespectrales (que recogen información en varias bandas) e hiperespectrales (con un número más elevado de bandas que los sensores multiespectrales). (Amann et al.,2001; Schott, 2007)

Tanto en las nubes de puntos LiDAR como en los sistemas basados en imágenes se recogen todos los datos que el sensor o sensores son capaces de captar. Esta información requiere, en general, una serie de correcciones y transformaciones geométricas, y una posterior interpretación de los objetos y entidades captados. Por ejemplo, una imagen aérea requiere una serie de procesos fotogramétricos para su ortorectificación y georreferenciación, que permiten la toma de medidas sobre ella y la comparación con otras fuentes cartográficas. La ortofotografía resultante se puede considerar un producto cartográfico no interpretado, que puede estar sujeto a simplificación, clasificación e interpretación para su transformación o traducción al lenguaje o simbología cartográficos.

La interpretación y clasificación de los datos provenientes de toma masiva tienen ciertas peculiaridades, y el uso de estos datos no está limitado a los elementos o eventos para los que fue planificada la medición. Por ejemplo, una campaña de fotogrametría, cuya motivación es la elaboración de una cartografía de la red vial de una zona, puede ser usada con posterioridad para la extracción de cualquier otro tipo de datos que hayan quedado registrados en los fotogramas, como una cartografía de vegetación, o como parte de un estudio histórico de cualquier evento. En contraposición a esto, los métodos de toma de datos selectiva, recogen generalmente solo los datos que atañen al objetivo por el que se planteó la campaña de medición.

La interpretación de los datos provenientes de toma masiva es un posproceso que se puede realizar de forma manual por parte de un operador, pero que en muchos casos es susceptible de ser automatizado. Esta automatización de la interpretación de la realidad captada por los sensores de toma masiva de datos recibe el nombre de detección automática de elementos (en inglés, *Feature detection*), y consiste en la computación de abstracciones de objetos, formas o eventos. Es decir, (i) se determinan las características propias de cada elemento que son captadas por los sensores, y que permiten discriminarlos del resto de los elementos que no son objetivo (esta parte constituye la abstracción de los objetos o elementos), y (ii) se establecen los métodos necesarios para la determinación de qué datos cumplen los requisitos establecidos en la abstracción (esta parte constituye la computación) (Pu et al., 2011; Sun et al, 2006; Schnabel et al., 2007). A modo de ejemplo simple: una imagen de satélite que cubre una zona de lagos y bosques contiene datos en una sola banda en la región del infrarrojo cercano. Las superficies de agua profunda reflejan una parte muy pequeña de la radiación solar en el infrarrojo cercano, mientras que las cubiertas vegetales habitualmente reflejan gran parte de la radiación solar en esa parte del espectro. Esta es la abstracción: la correspondencia de píxeles con valores bajos en la imagen a las aguas profundas, y la de píxeles con valores altos en la imagen a la vegetación. La parte computacional (muy sencilla en este caso) consiste en la imposición de un umbral, por debajo del cual, los píxeles corresponderán a aguas profundas y, por encima del cual, se asignarán a cubierta vegetal. Esta determinación de abstracciones de elementos y su computación constituyen el diseño y la implementación de los algoritmos de detección automática, que sustituyen o complementan la interpretación individualizada que podría hacer un operador.

A menudo, la abstracción consiste en la imitación de los procesos de discriminación visuales. Es decir, aquellos procesos que siguen las personas para diferenciar objetos o elementos en base a la forma o color esperados o encontrados en elementos. Sin embargo, en otras ocasiones, la utilización de datos radiométricos (imágenes) o espaciales (nubes de puntos) de objetos permite el establecimiento de abstracciones o reglas discriminatorias de elementos basadas en datos no perceptibles para las personas. Es posible utilizar, como en el ejemplo anterior, datos de sensores que registran más allá del espectro visible, o transformar datos a priori perceptibles por las personas a espacios teóricos para la formulación de abstracciones más complejas que permitan la discriminación de elementos. Ejemplos de este último caso son las transformaciones de análisis de componentes principales, transformada de Hough, Fourier, o Wavelets, aplicables tanto a datos radiométricos (imágenes) como a datos espaciales (nubes de puntos), y que a menudo son utilizadas para la extracción de patrones o formas (Fang et al., 2004; Falkowski et al., 2006; Li et al., 2012; Demantke et al., 2011; Sithole & Vosselman, 2004, Fernández et al., 2007; Ballard, 1981). No obstante, incluso con el uso de datos no perceptibles o de espacios teóricos, la detección automática suele estar basada en una analogía a la percepción visual humana (Morrone & Burr, 1988)

Esta tesis doctoral versa sobre la automatización de la interpretación de datos masivos. En concreto, se desarrollan algoritmos para la detección automática de elementos en de nubes de puntos de escáneres láser móviles.

1.2 Tecnología LiDAR y escaneado láser

El escaneado láser (LS, del inglés *Laser Scanning*), es una técnica de toma de datos masiva, ampliamente utilizada durante las últimas décadas para la cartografía e inventario de áreas urbanas, vegetación, infraestructuras, o levantamientos topográficos de diversa índole. Los escáneres láser proporcionan densas nubes de puntos que cubren el terreno y objetos visibles desde el sensor, y están basados en la tecnología LiDAR, que permite determinar la distancia entre un emisor de un haz láser y un objeto o superficie que refleja su señal (Heritage & Large, 2009). En las décadas de 1970 y 1980, la tecnología LiDAR fue utilizada para la medición de perfiles desde plataformas aéreas para estudios batimétricos y forestales, entre otros (Solodukhin 1978, Nelson et al. 1984, Schreier et al. 1985), y se establecieron los principios básicos del uso de láseres para la teledetección. Sin embargo, no fue hasta la década de 1990 cuando se llevaron a cabo los primeros experimentos con instrumentos de escaneado láser. Los sistemas de medida LiDAR empezaron a ser unidos a sistemas óptico-mecánicos de escaneado/barrido, elementos de posicionamiento global y unidades de medida inercial, constituyéndose así los primeros escáneres láser aéreos (ALS, del inglés Aerial Laser Scanners), (Bufton 1989, Flood y Gutelius 1997, Hyyppä et al. 2008).

Es habitual la diferenciación entre escáneres láser aéreos (ALS) y escáneres láser terrestres (TLS, del inglés *Terrestrial Laser Scanner*). Dentro del último grupo (TLS), hay autores que engloban los escáneres láser móviles (MLS, del inglés *Mobile Laser Scanners*) (Barber et al., 2008; Kukko et al., 2012; Mendenhall, 2011), y hay otros que consideran los MLS una categoría aparte, estableciendo de esta forma tres tipos de LS: ALS, TLS y MLS (Hyyppä et al. 2013). Cuando se hace esta diferenciación en tres grupos, los TLS designan únicamente los escáneres láser estáticos, que suelen ir montados en trípodes (o alguna otra plataforma que garantice su estabilidad durante la medición desde un punto). El modo operativo de la medición con los TLS suele consistir en el emplazamiento del escáner en uno o varios puntos, desde los cuales son visibles los objetos que se desean medir. Si se toman datos desde más de un emplazamiento, se establece una red de puntos de coordenadas conocidas. Estos puntos han de ser identificables en número y distribución suficiente desde cada uno de los puntos de toma. Así, es posible unir con posterioridad los datos escaneados desde cada emplazamiento, de forma que estos estén referidos al mismo sistema de coordenadas (Vosselman y Maas, 2010).

Los MLS tienen la particularidad de que escanean desde una plataforma móvil, sin necesidad de que esta se detenga para hacer barridos. Atendiendo a la los sensores que incorporan y a la forma de calcular las coordenadas absolutas de los puntos medidos desde ellos, los escáneres láser móviles tienen más similitudes con los ALS que con los TLS, ya que incorporan sistemas de posicionamiento global y de medición inercial. Sin embargo, las nubes de puntos tomadas desde MLS tienen mayores similitudes con los TLS que con los ALS, ya que habitualmente están tomadas desde puntos muy cercanos al suelo, y su perspectiva no es cenital, como la que ofrecen los ALS. Esto provoca que las nubes de puntos ALS tengan una cobertura más homogénea del terreno medido. Sin embargo, los TLS y MLS generan nubes de puntos con mayor detalle, haciendo posible la inclusión elementos no visibles desde una plataforma aérea, como fachadas o cualquier otra estructura/forma/objeto vertical o extraplomado (Fowler & Kadatskiy, 2011; Rutzinger et al. 2009)

Los escáneres láser se componen típicamente de los siguientes elementos: (i) unidad de medida/distanciómetro láser (LiDAR), (ii) sistema óptico-mecánico de barrido/escaneado, (iii) unidad de posicionamiento y orientación, y (iv) unidad de control, procesado y almacenamiento (Wehr y Lohr 1999). Los TLS, si se considera un grupo independiente de los MLS, carecen de la unidad de procesamiento y orientación, cuyo cometido es sustituido por el uso de la anteriormente mencionada red de puntos de coordenadas conocidas.

La unidad de medida/distanciómetro láser está formada por un emisor, un receptor, y las ópticas de ambos. El principio básico de los LiDAR es el uso de un láser (del inglés, Light Amplification by Simulated Emission of Radiation) para iluminar un objeto, y un fotodiodo para registrar la radiación reflejada (Schawlow y Townes 1958). La distancia es determinada mediante el procesamiento de la señal, haciendo uso del hecho de que la velocidad de la luz es constante. La óptica del sistema receptor recoge la luz reflejada y la dirige al fotodiodo, que convierte el haz incidente en pulsos eléctricos (Wehr y Lohr 1999). En los sistemas de escaneado láser actuales se utilizan principalmente dos técnicas para la medida de distancias: (i) tiempo de vuelo (ToF, del inglés *Time of Flight*), y (ii) medición de fase (PS, del inglés *Phase Shift*). La técnica ToF utiliza la medición precisa de tiempos para determinar la distancia sensor-objeto a partir de la velocidad de la luz y el tiempo que trascurre desde que el pulso láser sale del emisor, hasta que es captado por el receptor. Sin embargo, las técnicas PS se basan en la emisión continua de señal láser y en la modulación de la amplitud de la señal para calcular las distancias emisor-objeto.

En los LiDAR de tiempo de vuelo se emite un pulso corto de luz (típicamente entre 5 y 10 nanosegundos) hacia el objeto a medir. La distancia entre el objeto y el emisor se obtiene al multiplicar la velocidad de transmisión de la luz por la mitad del tiempo que transcurre entre la emisión de este pulso corto y su recepción por el diodo (tiempo de vuelo). Es decir, el tiempo que tarda la señal en ir desde el emisor al objeto que la refleja, o desde el objeto hasta el receptor. La energía del pulso láser se dispersa alrededor de la línea central del haz por la divergencia que se produce antes de que alcance del objeto. Esta dispersión se traduce en el mayor o menor tamaño de la huella del haz. Así, dependiendo de la geometría del objeto, del tamaño de la huella y del ángulo de incidencia, es posible que el pulso corto emitido sea reflejado en forma de varios ecos, en función de las distancias a las que encontró superficies de reflexión dentro de la huella que forma el haz. Por ejemplo, si un haz vertical encuentra un árbol, y si la frondosidad del árbol y el tamaño de la huella lo permiten, el pulso emitido se verá reflejado en varios ecos. El primero de ellos corresponderá con el primer objeto que reflejó parte del pulso emitido (una hoja o una rama de la copa), y el último eco corresponderá a la reflexión del pulso sobre el suelo (si este llega a él). Existen múltiples soluciones para gestionar la recepción de señales más complejas que la señal emitida, como el registro de un único pulso (el primero o el último), el registro de varios pulsos (dos o más), o el registro de la onda reflejada completa (en inglés, *Full Waveform*), (Wagner et al. 2004, Palojärvi 2003, Jutzi et al. 2003, Wagner et al. 2006)

En los LiDAR de medición de fase se emite un haz continuo con varias longitudes de onda y amplitudes moduladas (Kikuta et al. 1986, Paschotta 2013). Para calcular la distancia recorrida por el haz, se mide la fase de la longitud de onda más corta de las emitidas, con lo que solo resta resolver la ambigüedad del número de ciclos de onda completos en el recorrido del haz. Para esto se utilizan el resto de las longitudes de onda (Wehr and Lohr 1999).

El sistema óptico-mecánico de barrido/escaneado desvía el haz hacia los objetos que se desean medir, de tal forma que la configuración de este sistema determina el patrón de los puntos sobre los objetos medidos. Existen varios métodos para desviar el haz, entre ellos, los más comunes son espejos planos oscilantes, espejos giratorios y el uso de fibra óptica. Las principales diferencias entre los distintos sistemas de barrido del haz se traducen en diferencias en el campo angular de vista, o de escaneado (FOV, del inglés *Field of View*), y en la velocidad de escaneado.

Tradicionalmente, el escaneado LiDAR se ha dividido en dos grandes clases en función del tamaño de la huella del haz sobre el terreno u objetos. Sin embargo, esta clasificación se basaba en los sistemas ALS, siendo considerados escáneres de huella pequeña aquellos con tamaños menores de 1 ó 2 m sobre el terreno. El escaneado no aéreo (TLS y MLS) utiliza tamaños de huella de 2 a 100 mm, que ha pasado a constituir un nuevo grupo (huella muy pequeña) (Anderson 2006, Blair et al. 1999, Brenner et al. 2000, Carbajal y Harding 2001, Harding et al. 2001, Hyde et al. 2005, Lefsky et al. 1999, Wagner 2006, Mallet y Bretar 2009).

1.3 Escáneres láser móviles

Los escáneres laser móviles (MLS, del inglés *Mobile Laser Scanner*) son sistemas formados por varios sensores unidos de forma solidaria, montados sobre una plataforma móvil (automóvil, ferrocarril, embarcación, plataforma aérea, o mochila) (Talaya et al. 2004, Kremer y Hunter 2007, Gräfe 2007ab, Kukko 2009, Kukko 2007, Puente et al. 2011, Alho et al. 2011), que registran datos espaciales del entorno por el que circulan. En general, estos datos consisten en una nube de puntos tridimensional que cubre los objetos y elementos visibles desde los sensores a su paso. Los sistemas MLS constan habitualmente de: (i) uno o varios escáneres láser, (ii) un dispositivo de posicionamiento global (GNSS, del inglés *Global Navigation Satellite System*), (iii) un sistema de medición inercial (IMU, del inglés *Inertial Measurement Unit*), (iv) una estructura calibrada para el anclaje de cada uno de los instrumentos anteriores, y (v) un sistema de procesado y almacenamiento de la información registrada por los sensores. Opcionalmente, los MLS incorporan sensores auxiliares, siendo común que cuenten con cámaras fotográficas, existiendo en la actualidad una clara tendencia a la incorporación de otros sensores auxiliares, como sensores multiespectrales, cámaras termográficas, perfilómetros, etc. La incorporación de nuevos sensores justifica denominación de estos sistemas como sistemas de cartografía móvil (MMS, del inglés *Mobile Mapping System*), más genérico que MLS.

Cada uno de los escáneres láser que incorporan los MLS suele consistir en un distanciómetro que gira sobre un plano (sensores rotacionales) (Barber et al., 2008). Estos sistemas registran puntos sobre un mismo plano (respecto a la posición del MLS), y aportan, para cada punto, dos datos: (i) la distancia sensor-punto registrado, y (ii) el ángulo bajo el que se registró el punto. Sin embargo, existen dispositivos que no usan sensores rotacionales limitados a un plano (Glennie y Lichti, 2010), y otros que limitan a un plano el movimiento de sensores que, de otra forma, permitirían el movimiento libre, como el *RoamerR2* (Kukko et al., 2012), que utiliza un escáner láser terrestre, cuyo movimiento de rotación horizontal es bloqueado). El sistema GNSS que incorporan los MLS permite la asignación de coordenadas absolutas a los puntos escaneados, y la IMU recoge los datos necesarios para corregir las posibles desviaciones del sistema durante su desplazamiento.

Los datos medidos: (i) distancias y ángulos de escaneado, (ii) coordenadas globales del centro de medición, y (iii) desviaciones del sistema en el momento de la toma de cada punto son integrados en una nube de puntos tridimensional en coordenadas globales. La precisión de estas coordenadas es, en general, de uno o varios centímetros en términos absolutos (dependiendo del sistema MLS y de las condiciones de la toma de datos), y de pocos milímetros en la posición relativa de puntos próximos. Estas precisiones implican habitualmente que la geometría interna de los objetos escaneados puede tener precisión milimétrica, mientras que su posición absoluta solo se puede determinar con precisión centimétrica. (Guan et al., 2016; Puente et al., 2013ab; Guan et al., 2013). Una alta precisión interna en la posición de los puntos permite la identificación visual de la mayoría de los objetos medidos por MLS. Por otro lado, los datos MLS suelen ser utilizados para inventario/cartografía urbana, y una precisión centimétrica en la posición absoluta de los objetos cartografiados suele ser suficiente. Sin embargo, cuando se utilizan al mismo tiempo datos MLS obtenidos en distintas campañas de medición, los puntos sobre un mismo objeto procedentes de distintas nubes de puntos pueden tener diferencias centimétricas, lo que puede dificultar la identificación de determinados objetos. Este fenómeno puede producirse también con los puntos de una misma medición, si (i) existen errores de calibración en alguno de los sensores, o si (ii) las condiciones de la toma son adversas, por ejemplo: poca cobertura del sistema GNSS, o si se produce bajo vibraciones o movimientos por encima del límite de detección de la IMU.

En esta Tesis Doctoral, se han utilizado datos de tres MLS diferentes para realizar los tests de los métodos propuestos: (i) Lynx Mobile Mapper system (Teledyne Optech, 2016), (ii) Riegl VMX-250 (Riegl, 2016), y (iii) RoamerR2 (Kukko et al., 2012). Los capítulos 2, 3 y 4 contienen una descripción más detallada de cada uno de estos sistemas.

1.4 Antecedentes en la extracción de elementos a partir de nubes de puntos

El creciente tamaño y complejidad de los datos geométricos que captan los sensores LiDAR ha impulsado el aumento continuo de la demanda de métodos para la extracción automática de elementos (Guan et al., 2016). Las únicas herramientas para la interpretación de las nubes de puntos son el conocimiento previo de las formas y objetos reflejados en la nube, y la posición y distribución de los puntos sobre ellos (Huang &

Meng, 2001). Sin embargo, para hacer posible un uso eficaz de estos datos, a menudo es necesario enriquecer las nubes de puntos con abstracciones y, posiblemente, información semántica, proporcionando una interacción a un nivel más alto. Es decir, definiendo formas, tamaños y demás características de grupos de puntos sobre formas básicas, de manera que (una vez categorizados) permitan la identificación de estructuras más complejas. Estas herramientas son capaces de ofrecer el grado de interacción necesario para identificar ciertas partes de los datos, y por tanto hacer que estos sean más directamente utilizables para modelización de elementos.

Los enfoques de ingeniería inversa tradicional (Kruth y Kerstens 1998, Sarkar y Menq 1991, Liu et al. 2003) pueden suministrar parte de las abstracciones buscadas, pero suelen estar enfocados a la búsqueda de la geometría precisa de los elementos escaneados, y a menudo requiere la interacción intensiva de un operador. El uso de los recursos necesarios para tal interacción no se justifica cuando no se requiere una reconstrucción detallada de los elementos escaneados, sino simplemente la identificación de objetos y la determinación de características básicas, como su posición, o dimensiones no precisas. Esto ocurre principalmente en las nubes de puntos ALS y MLS, cuyo propósito suele ser la cartografía y/o inventario, y no la descripción geométrica rigurosa de las formas recogidas en la nube de puntos (Guan et al., 2016)

Otro problema que surge al tratar con nubes de puntos LiDAR es el enorme tamaño de los datasets (Elseberg et al., 2013), por lo que la eficiencia de los algoritmos infiriendo abstracciones es de suma importancia. Esto llega a convertir en inviables (o, al menos, no rentables) algunos métodos por el tiempo de procesado que requerirían. Varios algoritmos publicados en los últimos años proponen (o utilizan) métodos, estrategias o estructuras de simplificación de las nubes de puntos. Cabe destacar dentro de estas estructuras, las basadas en la imposición de una malla de ortoedros o cubos, como la voxelización, los octrees o los kd-trees (Elseberg et al., 2013; Meagher, 1982; Nuchter et al., 2007). La principal diferencia entre ellas radica en (i) la regularidad de las estructuras utilizadas (tamaño y forma de cubos, vóxeles u ortoedros), y (ii) en la inclusión de órdenes jerárquicos y relaciones de vecindad. De esta forma, los octrees (que generalmente dividen el espacio en cubos) y kd-trees (que utilizan ortoedros iregulares) imponen una jerarquía entre las estructuras que se generan, y establecen y almacenan relaciones de vecindad entre ellos, mientras que la voxelización se reduce a la imposición de una malla regular (sin relaciones jerárquicas ni de vecindad). Si bien, los modelos jerárquicos ofrecen grandes ventajas para el acceso, agrupamiento e identificación de los datos, la generación inicial de estas relaciones tiene altos requerimientos computacionales, y su uso no se justifica cuando el número de búsquedas en la red de cubos u ortoedros se prevé pequeño (Vanderhyde and Szymczak, 2008; Elseberg et al., 2013).

La voxelización, en su versión más simple, es un proceso sencillo y con bajos requerimientos computacionales. Sin embargo, la recuperación de los puntos incluidos en cada vóxel no está contemplada a priori. Es decir, habitualmente, la nube de puntos se voxeliza (se simplifica), generando una versión reducida y por tanto más ágil, pero esta operación supone una pérdida irreversible de resolución, densidad y/o detalle en los datos de entrada (Aijazi et al., 2013, Hosoi and Omasa, 2006, Moskal and Zheng, 2012, Truong-Hong et al., 2013; Wu et al., 2013).

Cabe diferenciar entre los algoritmos de interpretación automática de nubes de puntos (i) aquellos cuyo objetivo es un tipo de elemento u objeto concreto (postes, vehículos, peatones, tendidos eléctricos, etc.) y (ii) aquellos con un objetivo más amplio, que se pueden considerar algoritmos de segmentación o división de la nube de puntos, pues no están dirigidos a una tipología específica. Dentro ambos grupos, pero especialmente en el segundo, es habitual el uso de transformaciones que permiten estimar indirectamente la voluminosidad de los objetos, como el Análisis de Componentes Principales (ACP). El ACP se aplica a grupos de puntos con el fin de obtener unos nuevos ejes coordenados ortogonales que se alineen con las direcciones de máxima variabilidad espacial de un conjunto de puntos. De esta manera, resulta sencillo determinar si un grupo de puntos está distribuido de forma lineal, se adapta a una superficie plana, o si, por el contrario, es una estructura volumétrica. Existen multitud de variantes de este tipo de aproximaciones al problema de la segmentación. Sus diferencias radican habitualmente en (i) el modo de selección de los puntos sobre los que se aplicará el ACP, y (ii) en el tipo de abstracciones que se formulan en el espacio de los nuevos ejes ACP para cada grupo de puntos (Yokoyama et al., 2011; Roggero, 2002; Yu et al., 2015)

Otras aproximaciones a la segmentación de las nubes de puntos están basadas en el "*Region Growing*", la transformada de Hough, o el RANSAC (del inglés, *RANdom SAmple Consensus*), cuyo uso está ampliamente extendido en la detección automática a partir de imágenes y nubes de puntos. Es también habitual el uso de combinaciones de estos métodos, que en ocasiones se unen a otros y/o a transformaciones locales como el ACP o globales como las voxelizaciones, o los árboles jerárquicos de ortoedros (Guan et al., 2016).

El "*Region growing*" aplicado nubes de puntos se basa en la unión de puntos, o de grupos de puntos que fueron a su vez relacionados por cualquier otro método (habitualmente por criterios simples de proximidad). La aplicación de estos métodos permite (en algunos casos) la elaboración de algoritmos robustos ante la presencia de oclusiones que dividen la nube de puntos sobre un mismo objeto. Así, si un objeto queda dividido en dos o más grupos de puntos inconexos, el establecimiento de las condiciones adecuadas para el "*Region Growing*" podría permitir la unión estos grupos (Rabbani et al., 2006; Vanco et al., 2000; Leonardis et al., 1995; Schnabel et al., 2008)

La transformada de Hough tiene su origen en una patente estadounidense de 1962 (Hough, 1962), y desde entonces su uso se ha ido extendiendo y generalizando para la búsqueda de formas y patrones en imágenes y nubes de puntos (Ballard, 1981; Woodford, 2014). El método se basa en la transformación de los datos de origen a un espacio paramétrico, donde se evalúa el número de veces que se cumple un patrón determinado. En su caso más sencillo (coincidente con la formulación original), si se buscan líneas rectas en un espacio bidimensional, se analizan para cada punto los parámetros de todas las rectas que pasan por él. En el espacio paramétrico de las rectas, se estudia el número de veces que se repite un conjunto de parámetros concreto; en este caso dos parámetros, al tratarse de rectas. Los pares de parámetros que más se repiten, o los que se repiten más de un número preestablecido de veces, se considerarán rectas. Este método puede ser utilizado con cualquier forma parametrizable, aunque su rendimiento se ve muy afectado por el número de puntos y, principalmente, por el número de parámetros o grados de libertad. Por esta razón, es práctica común entre los algoritmos que utilizan la transformada de Hough (i) la simplificación de las formas buscadas, y/o (ii) la división del proceso en varias transformaciones menores (con menos parámetros) (Rabbani y Van Den Heuvel, 2005)

Aunque existen varios algoritmos que combinan la transformada de Hough y RANSAC (Tarsha-Kurdi et al., 2007), su forma de afrontar el problema de la identificación de formas predefinidas es prácticamente opuesto. El método RANSAC fue inicialmente propuesto por Fichler y Bolles en 1982, y, al igual que la transformada de Hough, se utiliza para buscar formas parametrizables y predefinidas. Cuando es aplicado a nubes de puntos, parte del mínimo número de elementos (puntos) necesario para definir la forma buscada. Estos puntos son elegidos repetida y aleatoriamente dentro de la nube, y se compara la situación del resto de puntos respecto de la forma ajustada en cada paso. Se definen una serie de condiciones de salida del proceso, y, finalmente, se elige la forma con más puntos dentro de un margen de tolerancia preestablecido. Se trata de un método robusto ante la presencia de valores atípicos (*outliers*), pero que presenta inconvenientes en la elección de umbrales y condiciones de salida, y, en su versión más simple, no es capaz de ajustar varias formas presentes en un mismo conjunto de datos (Douillard et al., *2011*; Raguram et al. 2008, Schnabel 2007)

Los métodos generales (de segmentación global de la nube de puntos) tienen un objetivo más ambicioso que los métodos concretos (aquellos dirigidos exclusivamente a la detección de un tipo de objeto o elemento), ya que no limitan su búsqueda a una sola tipología. Sin embargo, los métodos generales requieren habitualmente una clasificación posterior de los elementos segmentados, y suelen estar basados en la extracción de formas básicas. En este sentido, es frecuente el uso de algoritmos generales de segmentación como paso previo a la aplicación de algoritmos de búsqueda de elementos concretos (Pu et al., 2011; Yokoyama et al., 2011).

Existen algoritmos que integran la segmentación general de la nube de puntos y la clasificación de los elementos segmentados. Estos métodos, a menudo basados en el aprendizaje automático, presentan como inconvenientes principales (i) la necesidad de adaptar los objetos de entrenamiento a los datos que se vayan a utilizar (lo que puede eliminar su carácter universal y automático), y (ii) sus resultados, más pobres en general que los métodos concretos que detectan cada uno de los objetos por separado (en términos de tasas de detección y corrección de los datos detectados). (Lehtomäki et al., 2016; Golovinskiy et al., 2009).

La detección automática de elementos concretos en nubes de puntos es un campo que se ha explotado ampliamente en los últimos años; principalmente aplicado a nubes de puntos de sensores ALS y TLS estáticos. En lo referente a los algoritmos para la extracción automática de elementos concretos a partir de nubes de puntos MLS, se están produciendo grandes avances, aunque resulta evidente la necesidad de mejora y afinamiento de la detección de muchos grupos de objetos (Guan et al., 2016).

Existe un amplio espectro de elementos cartografiables que despiertan el interés de los grupos de investigación y del mercado del software comercial de tratamiento de nubes de puntos (que demanda la mejora del rendimiento y precisión de los algoritmos de detección automática). La configuración y precisión actual de los sensores MLS y el interés de sus usuarios están principalmente dirigidos a la creación de inventarios en entornos viales. Es decir, catálogos cartográficos con los elementos constructivos relevantes del entorno por el que pueden circular la mayor parte de los MLS. Dentro de los elementos cartografiables en el entorno vial, cabe destacar (i) los postes y objetos situados sobre postes (farolas, semáforos o señales, entre otros), (ii) los edificios y construcciones, (iii) la propia superficie asfaltada de la calle o carretera por la que circula el MLS, (iv) los bordillos, o (v) las marcas viales.

En la bibliografía específica de la detección automática a partir de datos MLS, es posible encontrar multitud de algoritmos que abordan el tema para cada uno de estos grupos de elementos, pero aún existe un gran margen de mejora en muchos de ellos en lo referente a (i) su universalidad (es decir, su posible aplicación en distintos ámbitos, ambientes o entornos), (ii) su grado de automatización, requiriendo muchos de ellos supervisión de los resultados, adaptación de parámetros a cada caso concreto, o generación de grupos de aprendizaje, (iii) necesidad expresa en muchos de ellos de una segmentación manual previa a su aplicación en una nube de puntos, (iv) restricciones de volumen de datos que son capaces de tratar, (v) robustez de las abstracciones (resistencia de la formulación básica del algoritmo a la presencia de artefactos y/o ausencia de puntos por ocultamientos), (vi) rendimiento computacional excesivamente bajo, y (vii) las tasas de detección y de corrección de los elementos detectados, que en muchos casos son demasiado bajas como para justificar el uso de los medios y la tecnología MLS.

En esta tesis se desarrollan algoritmos para la detección automática de postes, superficies, y superficie asfaltada. En los Capítulos II, III y IV son consideradas y discutidas de forma pormenorizada todas las deficiencias y posibles mejoras de estudios previos que tratan la detección automática de estos tres elementos, así como el desarrollo algorítmico y computacional, las pruebas realizadas a los métodos propuestos, y los resultados obtenidos.

1.5 Objetivos

Se establece como objetivo inicial (y previo al principal de creación de algoritmos de detección automática) el desarrollo de estructuras de simplificación y homogeneización de la nube de puntos. Estas estructuras han de proporcionar una versión de la nube de puntos, que:

- Suponga una simplificación/reducción sustancial del volumen de datos.
- Esté compuesta por elementos (no necesariamente puntos) que supongan una ordenación u homogenización de los datos originales.
- Permita la agilización de los procesos de detección y/o la formulación de nuevas abstracciones.
- Sea reversible. Es decir, que la detección automática pueda hacerse sobre los elementos simplificados, que estos puedan ser etiquetados, y que sea posible extraer de forma rápida y sencilla los puntos de la nube original que corresponden a cada elemento.

Se establece como objetivo principal de la tesis el desarrollo de tres algoritmos de detección automática que, en base a las estructuras de simplificación y homogeneización desarrolladas, extraigan, delineen y/o identifiquen los siguientes elementos a partir de nubes de puntos de escáneres láser móviles:

1. Los postes en entornos viales, y los objetos sobre ellos, tales como farolas, señales, semáforos, postes desnudos, o incluso árboles en entornos urbanos. El algoritmo debe ser capaz de mejorar los resultados y rendimiento de los algoritmos existentes, y (i) ha de ser capaz de detectar elementos con forma de poste con independencia de las estructuras unidas a ellos, (ii) no ha de requerir datos de entrenamiento, (iii) no ha de hacer asunciones iniciales a cerca de la ubicación de los postes, y (iv) ha de ser independiente de la geometría del escaneado (ángulo y frecuencia de escaneado) y la estructura de los datos, de forma que solo sean necesarias las coordenadas de los puntos de la nube.
2. Las superficies captadas desde el escáner láser móvil, como fachadas de casas o edificios, paneles, o muros, sean completamente planas o no. Los requerimientos iniciales son: (i) la detección no ha de estar limitada a superficies estrictamente planas (ha de ser capaz de detectar ciertas superficies regladas o con cierta curvatura), (ii) ha de ser capaz de detectar superficies con independencia de su inclinación, (iii) ha de ser completamente automático (no ha de requerir datos de entrenamiento), y (iv) no ha de requerir más datos que las coordenadas de los puntos de la nube MLS y el momento en el que fueron registrados.

3. La delineación de superficies asfaltadas en calles y carreteras. El algoritmo ha de funcionar exclusivamente con datos geométricos, es decir: coordenadas de los puntos de la nube MLS, momento en el que fueron registrados, y trayectoria del vehículo MLS. El método ha de ser capaz de resolver situaciones adversas, como irregularidades y discontinuidades en la superficie asfaltada (como grietas, gravilla, o zonas parcheadas), y la presencia de bordes poco netos, o nítidos, en términos geométricos.

Los algoritmos de detección automática han de ser probados con nubes de puntos MLS, contemplando la presencia de las situaciones adversas más comunes en la detección de cada elemento para demostrar su robustez. Así mismo, se ha de evaluar la calidad de la detección (tasas de detección y de corrección de la detección)

Los requerimientos específicos de cada uno de los tres algoritmos de detección automática están impuestos en base a deficiencias y limitaciones de los estudios publicados hasta la fecha que se consideraron susceptibles de mejora. La consecución de estos requerimientos, unida a la mejora de los resultados de otros estudios, supone el reto fundamental de esta tesis.

Capítulo II

Detección de Postes

[*An algorithm for automatic detection of pole-like street furniture objects from Mobile Laser Scanner point clouds*]

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

An algorithm for automatic detection of pole-like street furniture objects from Mobile Laser Scanner point clouds

C. Cabo [a,*], C. Ordoñez [a], S. García-Cortés [a], J. Martínez [b]

[a] Department of Mining Exploitation, University of Oviedo, 33004 Oviedo, Spain
[b] Department of Natural Resources and Environmental Engineering, University of Vigo, 36310 Vigo, Spain

ARTICLE INFO

Article history:
Received 14 May 2013
Received in revised form 16 October 2013
Accepted 22 October 2013
Available online 25 November 2013

Keywords:
Urban
Simplification
Detection
Laser scanning
Algorithms
Mobile

ABSTRACT

An algorithm for automatic extraction of pole-like street furniture objects using Mobile Laser Scanner data was developed and tested. The method consists in an initial simplification of the point cloud based on the regular voxelization of the space. The original point cloud is spatially discretized and a version of the point cloud whose amount of data represents 20–30% of the total is created. All the processes are carried out with the reduced version of the data, but the original point cloud is always accessible without any information loss, as each point is linked to its voxel. All the horizontal sections of the voxelized point cloud are analyzed and segmented separately. The two-dimensional fragments compatible with a section of a target pole are selected and grouped. Finally, the three-dimensional voxel representation of the detected pole-like objects is identified and the points from the original point cloud belonging to each pole-like object are extracted.

The algorithm can be used with data from any Mobile Laser Scanning system, as it transforms the original point cloud and fits it into a regular grid, thus avoiding irregularities produced due to point density differences within the point cloud.

The algorithm was tested in four test sites with different slopes and street shapes and features. All the target pole-like objects were detected, with the only exception of those severely occluded by large objects and some others which were either attached or too close to certain features.

1. Introduction

Accurate urban cartography is being increasingly demanded for several purposes in city management (Sahin et al., 2012; Gröger and Plümer, 2012; Shi et al., 2008). Three-dimensional models are now widely used for street and traffic control (Buch et al., 2008; Ranzinger and Gleixner, 1997), where the identification and accurate determination of the location and shape of certain street furniture elements is crucial. The presence of elements such as traffic lights, traffic signs, lampposts, utility poles or street trees has a huge impact on street and road planning, safety and maintenance, as they have a critical role in the general city management and in the road and street visibility studies for traffic management purposes (Escalera et al., 2010). In addition, the presence of aforementioned objects is used and needed for vehicle and pedestrian navigation and for driver assistance (Zin et al., 2007). Most of these street furniture objects either contain a pole or are entirely shaped like a pole. For instance, lampposts are often pole-like objects (i.e. shaped like a pole), traffic lights are usually placed in a pole-like structure and street trees often have a pole-like trunk.

Three-dimensional city mapping has been carried out in the last two decades using several methods in order to achieve accurate spatial models of the volumetric elements present in urban environments (Frueh and Zakhor, 2003; Haala and Brenner, 1999; Holopainen et al., 2011; Zhou and Neumann, 2013). Surveying, photogrammetry and remote sensing have been widely used, but the emergence and popularization of LiDAR technologies have produced a wide range of new techniques and applications (Gonzalez-Aguilera et al., 2013). The LiDAR technologies most commonly used for urban mapping can be divided in: (i) Airborne Laser Scanning (ALS), (ii) Terrestrial Laser Scanner Systems (TLS) and (iii) Mobile Laser Scanning (MLS) Systems.

Airborne Laser Scanning (ALS) has been used since the early 1990s. This method produces an adequate point density for the extraction of large urban features (i.e. building footprints or vehicles), but is often not enough for smaller or vertical elements. Moreover, the scanning angle does not allow the adequate measurement of points on vertical surfaces (Boulaassal et al., 2007). Terrestrial Laser Scanner Systems (TLS) are able to provide a much higher point density and do not have the angle limitation for

[*] Corresponding author. Tel.: +34 696333979.
E-mail addresses: carloscabo.uniovi@gmail.com, carloscabogomez@hotmail.com (C. Cabo).

0924-2716/$ - see front matter © 2013 International Society for Photogrammetry and Remote Sensing, Inc. (ISPRS) Published by Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.isprsjprs.2013.10.008

vertical objects detection that the ALS has. However, measurements from TLS are usually affected by occlusions in urban environments, and time-consuming scanning from different locations is needed (Dold and Brenner, 2006).

Mobile Laser Scanning (MLS) systems operate with the same principles as ALS, but are usually deployed in a vehicle, such as a van or a car. However, MLS systems produce a denser 3D point cloud than ALS and they use more adequate scanning angles for the measurement of vertical features. Furthermore, MLS systems avoid some of the occlusions that affect TLS, due to the movement of the scanning device and the fact that MLS systems usually use more than one sensor that operate in different scanning planes (Puente et al., 2012; Tao, 2000; Vaaja et al., 2011).

The distribution of the points from laser scanning systems (especially MLS) is usually heterogeneous and the amount of data is generally extraordinarily large. Therefore, and in order to reduce the processing times and the complexity of the datasets, the point clouds are often simplified before the use of an algorithm for feature extraction. In some cases (Yokoyama et al., 2011), the largest features present in the point cloud (i.e. ground points and/or buildings) are removed manually. Alternatively, a segmentation technique based on a surface growing algorithm, which groups points according to their connectivity and coplanarity, can be used (Pu et al., 2011; Vosselman et al., 2004). Another option is to apply a segmentation to the sweeps of the MLS, eliminating groups of points that are not compatible with the section of the target elements (i.e. pole-like objects) (Lehtomäki et al., 2010). More recently, (Puttonen et al., 2013) applied different distance-sensitive sampling methods to the original point cloud and tested them for pole-like objects detection using the algorithm from (Lehtomäki et al., 2011).

Storage and compression of a vast and dense point cloud from TLS is performed in (Elseberg et al., 2013) using octree structures, where the point clouds are stored in a volumetric hierarchical space (Meagher, 1982). Nevertheless, in many cases, it is not necessary to build categorized structures (Elseberg et al., 2013) and a grid of volumetric units (i.e. voxels) is used (Aijazi et al., 2013; Hosoi and Omasa, 2006; Moskal and Zheng, 2012; Truong-Hong et al., 2013; Wu et al., 2013). In some of these voxel spaces, the topologic relations are not initially established, so conditions of neighborhood/proximity that would represent one third of the total processing time (Vanderhyde and Szymczak, 2008) can be analyzed in subsequent stages within the regions or groups of voxels of special interest.

It is only very recently that a few studies have started to address pole-like object extraction from MLS in urban environments. In 2010 (Lehtomäki et al., 2010) used a method which looked for pole sections within each sweep of the MLS. The selected features from different scan lines were linked together and their isolation was checked using two cylinders (see (Brenner, 2009)).

In 2011, (Yokoyama et al., 2011) used iterative smoothing in order to obtain skeleton structures and a subsequent identification of the pole-like objects using principal component analysis. (Pu et al., 2011) used both a shape-based method, and a process based on the identification of horizontal sections of the pole-like features avoiding their extreme segments.

Golovinskiy et al. (2009) propose methods for urban features recognition (including poles) developed from an initial selection of potential objects based on point densities. The candidate objects are separated from the background, and they are characterized according to their spatial context and configuration. Finally, the features are classified comparing their characterization with labeled data from a training dataset.

More recently, (Wu et al., 2013) developed a method that uses an orthogonal, but non regular voxel space for tree detection in urban environments. The algorithm is based in the detection of the sections of trees that match with the expected diameter at breast height (DBH) (i.e. 1.2–1.4 m from the ground). A neighborhood search is applied to the sections that fulfill the requirements at DBH in order to extract the tree trunk and the estimation of other morphological parameters.

The aim of this work is to develop a new methodology for identification of pole-like street furniture objects from MLS data, which is more general and able to improve the performance of the existent methods: (i) able to detect pole-like objects with independence of the structures attached to them. It is a frequent situation that poles are joined together through tree branches or other features. Some methods like (Pu et al., 2011) or (Yokoyama et al., 2011) use 3D connected components labeling before the pole extraction, so they are not able to detect connected poles separately. (ii) Does not require training data. Machine learning based methods, such as (Golovinskiy et al., 2009) imply the time-consuming collection of training data, in addition, the fact that a model works with a training dataset, does not necessarily guaranty that adequate results could be obtained with other data, and they can suffer from overfitting problems (Ling, 1995; Zhang, 2000). (iii) No initial assumptions are taken about the position of the poles. In (Wu et al., 2013) it is assumed that all the targets (i.e. trees) are placed at the same height from the ground, and that they have a pole-like section at breast height (i.e. 1.2–1.4 m). (iv) The algorithm has to be independent of the scanning geometry (i.e. scanning angles and scanning frequency) and structure of the data (i.e. only the *XYZ* coordinates of the points are needed). The method from (Lehtomäki et al., 2010) fulfills the previous requirements, but is dependent of the scanning geometry, as it is based in the use of sections of the poles, which imply a limit in the tilt angle of the sweeps. An indexation of the points (i.e. sweep id. and point index within the sweep) is also needed as input for this algorithm.

The proposed methodology is based on very simple geometric principles and consists in the initial creation of a simplified version of the original point cloud through voxelization, the subsequent analysis of its horizontal sections, and the final identification of the poles and the structures attached to them in the voxel space.

Four test sites from different environments are used in order to test the algorithm. Three of them are from a narrow street in the city center, and they contain different common features in urban environments (i.e. large buildings, junctions, roundabouts, small parks, parked vehicles or bins). The fourth test site is a long street which contains many peri-urban structures, such as commercial centers, large parks, bridges and some road elements).

2. Methodology

The method proposed for the detection of street furniture pole-like objects from MLS data consists of three consecutive steps:

1. Voxelization of the point cloud space through codification. This stage allows to simplify the analysis and to reduce the computing cost of the subsequent operations.
2. Two-dimensional analysis of horizontal sections of the voxelized point cloud. At this stage, the candidates are identified by the properties of their sections.
3. Tridimensional reconstruction of the selected features from the previous 2D analysis.

2.1. Voxelization

Point clouds from MLS measurements are habitually very large and the distribution of the points is usually extremely heterogeneous. For that reason a simplification is often necessary. Consequently, we developed a voxelization method that allows the generation of a reduced version of the original point cloud. In that way, a reduction of the computing requirements is performed by fitting all the data in a regular tridimensional network that does not represent a significant information loss if the scale is correctly chosen. However, the original data is preserved and directly related to the reduced version.

The space is divided in a regular tridimensional grid. As a result, every single point is inside a volumetric cubic unit (i.e. voxel), and the only information which is stored is: (i) the coordinates of the centroids of the voxels containing at least an original point and (ii) the number of points which are inside each voxel. In that way, the original point cloud is significantly simplified using a spatial discretization and just storing the data needed in the following stages, although the initial data is preserved.

Our voxelization method is based on the codification of the *XYZ* coordinates of the scanned points in a single code and its subsequent decoding. The code is an integer number of 12 digits, consisting of three four-digit components. Each component represents one of the three *XYZ* coordinates. The coordinates are transformed to voxel units by dividing by the voxel size, and reduced to the origin of the point cloud using the minimum *XYZ* values (Eq. (1)). Then, the three components are assembled in the single code. An example of codification can be found in the first table of Fig. 1.

$$\begin{aligned}&\text{Code} : xxxxyyyyzzzz\\&xxxx = \text{integer}\frac{X - Xmin}{VoxelSize}\\&yyyy \text{ and } zzzz \text{ are similarly calculated}\end{aligned}\qquad(1)$$

Eq. (1): Codification of voxel centroids coordinates.

The values of the 12-digit code are stored in a single vector. Its length is equal to the number of points in the original point cloud (N). Given that original points belonging to the same voxel have the same code, vector elements are sorted by their value. In that way, all the points from the same voxel are placed together (see Fig. 1). From the vector containing sorted values, the unique elements (i.e. values of codes which appear at least once, but with no repetitions) are extracted and transferred to another vector. The length of the new vector (n) is equal to the number of voxels containing one or more points.

At this stage, two vectors (i.e. correspondence vectors) are generated in order to relate each original point with its voxel and *vice versa*. See Fig. 1 for a graphic example. Vector *M* contains *n* elements and stores the indices of the first point of each voxel. However, vector m contains *N* elements and stores the indices of the voxels that correspond to each original point.

Finally, the codification process is reverted. The 12-digit codes are split, and the data needed for the subsequent stages in the voxel space is calculated: (i) the coordinates of the centroids of the voxels containing at least a point and (ii) the number of points inside each voxel. In the same way, using the correspondence vectors (i.e. *M* and *m*), each voxel is directly related to its points, and each point is related to its voxel, so the points inside any voxel could be directly extracted if desired.

2.2. Two-dimensional analysis

Once the voxelization has been performed, the tridimensional grid is divided in horizontal slices in order to identify the structures that correspond more likely to the horizontal section of a pole. Regarding the properties of the target pole-like objects, some assumptions are established at this point: (i) the area of its sections has to be small, and (ii) a pole section has to be isolated (i.e. almost no point should be detected at its immediate surroundings).

The two-dimensional analysis is carried out in three stages: (i) Segmentation of the connected horizontal elements, (ii) selection of elements by a maximum area criterion and (iii) selection of elements by an isolation criterion. The result of this process is a set of 2D segments associated to a *Z* coordinate that are potentially part of a pole.

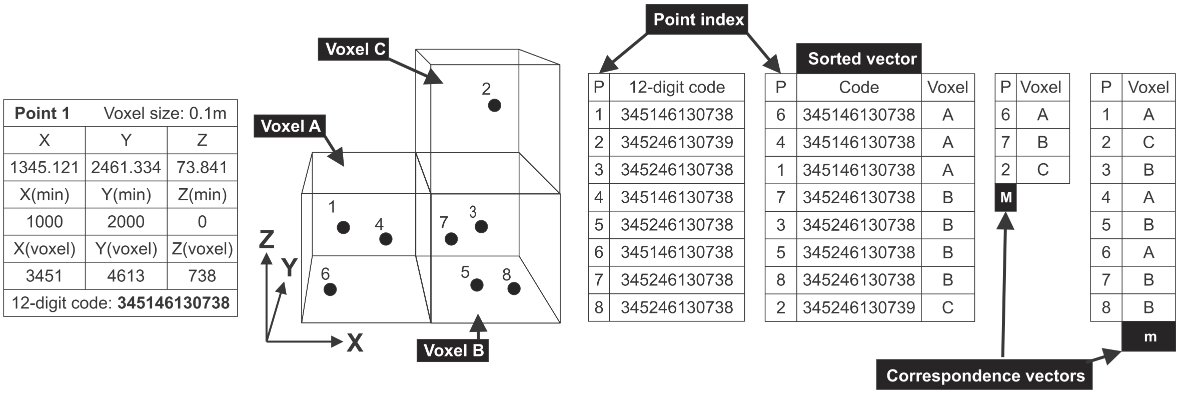
2.2.1. Segmentation of the connected horizontal elements

Each of the 2D sections is analyzed individually. A segmentation process based on the connectivity of the voxels is then implemented and the adjacent voxels are identified, grouped and stored. The segments are usually composed either by voxels from the same object or by voxels from objects attached to them. Hence, as shown in Fig. 2, the purely pole-like parts of a pole are expected to be separated from the rest.

2.2.2. Selection of elements by maximum area

A dimensional restriction to the size of the 2D segments (i.e. sets of connected voxels) obtained during the previous segmentation is applied. In order to be considered part of a pole, the area of its horizontal sections has to be limited; only the groups of voxels whose area is below a threshold are selected. Other restrictions regarding the shape of the section could be established at this point (e.g. the maximum length of the major axis, or a circularity ratio), but they are implicit in the next stage of the process (see Section 2.2.3 case C), where a maximum radius is set from the centroid of each section.

A threshold dependent on the voxel size is chosen, and the groups of voxels from the 2D sections are classified in 2 groups:



Point 1	Voxel size: 0.1m	
X	Y	Z
1345.121	2461.334	73.841
X(min)	Y(min)	Z(min)
1000	2000	0
X(voxel)	Y(voxel)	Z(voxel)
3451	4613	738
12-digit code: **345146130738**		

P	12-digit code
1	345146130738
2	345246130739
3	345246130738
4	345146130738
5	345246130738
6	345146130738
7	345246130738
8	345246130738

P	Code	Voxel
6	345146130738	A
4	345146130738	A
1	345146130738	A
7	345246130738	B
3	345246130738	B
5	345246130738	B
8	345246130738	B
2	345246130739	C

P	Voxel
6	A
7	B
2	C

P	Voxel
1	A
2	C
3	B
4	A
5	B
6	A
7	B
8	B

Fig. 1. Example of the voxelization process. In the first table, the 12-digit code is calculated for point 1. A graphic schema of the position of 8 example points is shown further right, as well as the code of the points, the sorted vector, the voxel assignation for each point and the correspondence vectors.

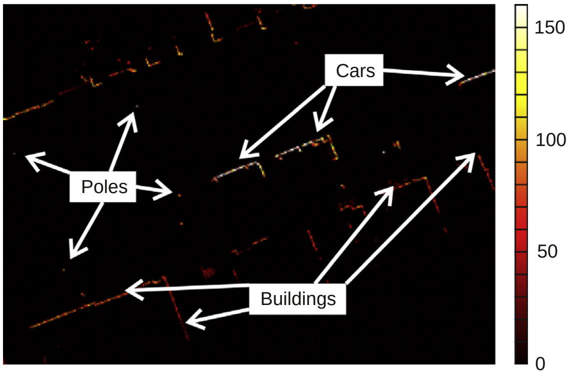


Fig. 2. 2D horizontal section of the tridimensional grid. Color represents the number of points stored in each voxel. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

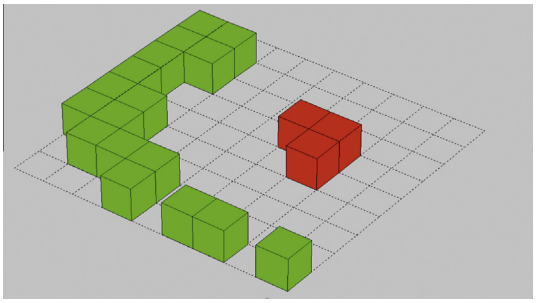


Fig. 3. Segmentation of horizontal sections by area. Red: voxels from a pole section. Green: voxels from building sections. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

(i) possible pole sections and (ii) non-pole sections. As shown in Fig. 3, a big group of pixels (i.e. bigger than the threshold) belongs to a large feature, not to a pole, but a small-size group, could be either part of a pole or part of a larger structure. Features with theoretically large horizontal sections (e.g. buildings or vehicles), could be affected by occlusions and present gaps or lack of continuity in their 2D representation. Thus, the features rejected at this stage are expected not to belong to a pole, and the selected features must be checked in subsequent stages.

2.2.3. Selection of elements by an isolation criterion

The second assumption described in Section 2.2 establishes that a pole section has to be an isolated element (i.e. no other elements should be found at its close surroundings). In order to check this condition, at this stage the centroids of the selected sections are calculated, and from that point, a neighborhood-proximity analysis is established for each pole section candidate.

From the centroid of each candidate section, two search radii are set: (i) an inner radius, within whose influence (i.e. Inside the area closer to the centroid than the radius distance) all the points of the section are supposed to be and (ii) an outer radius that defines a ring, in combination with the inner radius, where no point is expected (see Fig. 4).

Three possible situations are expected *a priori* (see Fig. 4): (a) In the case of the section of a pole (or any other object with a similar section) all the voxels are located within the influence of the inner radius, and no point is detected in the outer ring. (b) A large object, such as a building or a vehicle, should generate continuous sections with a number of connected voxels much bigger than the threshold set in the previous stage. Nevertheless, mainly due to occlusions, small gaps can give rise to patches with an area smaller than the threshold, but surrounded by voxels in the outer ring. (c) Sections from a medium-sized object (e.g. small vehicles or bins), or isolated patches from large objects resulting from big occlusions. Those sections can have a number of voxels lower than the threshold, but a shape not compatible with the horizontal section of a pole. The inner radius limits the major axis of the group of pixels and therefore, joined to the maximum area (see Section 2.2.2), the shape of the section.

Although, theoretically, no point is expected in the outer ring, individual points could be located at the surroundings of a pole section. Using the information stored in each voxel (i.e. regarding the number of original points placed within its limits), a threshold is set, allowing a certain number of isolated points in the outer ring. The number of points located in the ring is calculated and a section is rejected when the limit is surpassed. (Lehtomäki et al., 2010) applied a similar process in order to check the isolation of the preselected poles. The method used two concentric cylinders and compared the percentage of points inside each one of them.

2.3. Tridimensional analysis

A tridimensional neighborhood analysis is applied to the group of voxels that had been selected in the two-dimensional analysis. The sections that were not discarded in previous stages are now joined to their neighbors. All the voxels that share a face, an edge or a corner are grouped together.

In order to eliminate isolated sections (i.e. sections without neighbors and/or vertical connection), a threshold that limits the minimum height is set. The structures with a vertical length bigger than the limit are stored as poles, and the remaining groups are ignored because of their lack of vertical continuity or height (see Fig. 5).

The final result of the application of the pole detection algorithm is a set of voxel-structures that define the pole-like objects. Hence, they are likely to be classified and identified subsequently. These structures are directly related to the original point cloud from the MLS through the correspondence vectors obtained during

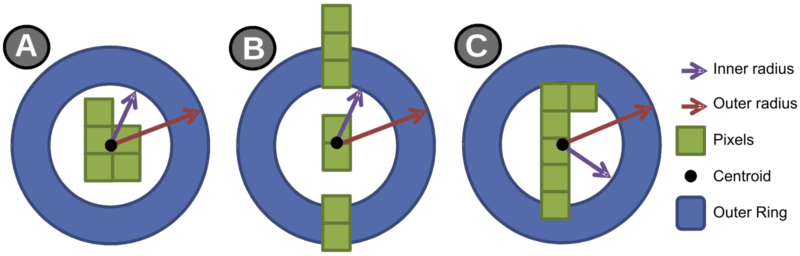


Fig. 4. Diagram of the outer ring search in a horizontal section. Cases: (a) Pole section, (b) a section of a wall with occlusions and (c) false positive detected by the presence of voxels in the ring.

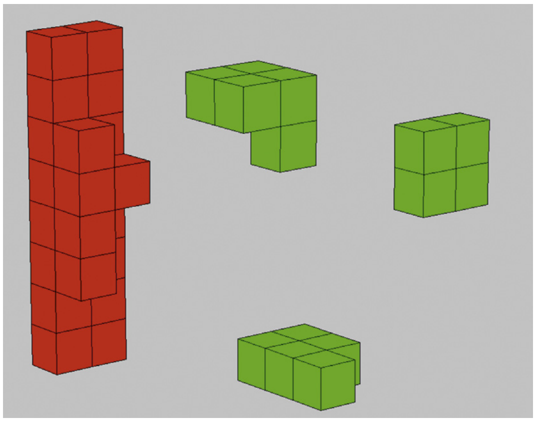


Fig. 5. Vertical union of filtered horizontal sections. Red: Group that exceeds the minimum height (Pole representation). Green: Isolated sections. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Manufacturer specifications for the Lynx Mobile Mapper.

Maximum range	200 m, 20%
Range precision	8 mm, 1σ
Absolute accuracy[*]	±5 cm (1σ)
Laser classification	IEC/CDRH class 1 eye-safe

[*] To meet the absolute accuracy, the GPS data must be of sufficient quality.

the voxelization process. Therefore, points inside each voxel can be recovered if desired.

3. Test case

3.1. Mapping data

The proposed algorithm was tested using two MLS datasets acquired with The Lynx Mobile Mapper system (Optech, 2013). The system is based on two LIDAR sensors mounted on a vehicle, an inertial measurement unit (POS LV 520) produced by Applanix, which consists of a 2-antenna heading measurement system, and an inertial navigation system (Puente et al., 2013). Table 1 contains some technical specifications of the MLS system.

The LIDAR sensors are located in the rear part of a van. Each one registers points in a plane at 60° to the horizontal and 45° to the longitudinal axis of the vehicle (i.e. driving direction) with a 360° field of view.

A point cloud is generated by combining the data from the LiDAR sensors and the inertial measurement unit. Its spatial resolution depends on the scan frequency of the LiDAR heads, that varies from 80 to 200 Hz, and the pulse repetition rate (PRR), ranging from 75 to 500 kHz. Both vertical and horizontal resolutions are influenced by the speed of the vehicle and the distance to the measured objects.

Two different datasets (Dataset A and Dataset B) from the city of Ourense (North-West of Spain) were used. Same settings were applied to the sensors in both datasets (i.e. A scan frequency of 200 Hz and a PRR of 500 kHz), and from the two of them, four different test sites (TS) were selected: TS-A1, TS-A2, TS-A3 and TS-B.

3.1.1. Dataset A (Test sites A1, A2 and A3)

The measurements from Dataset A were made along a 1.1 km long street in the city center (*Rúa do Progreso*). It is a narrow street (22 m on average) which widens in some parts because of parks, junctions and roundabouts (there are buildings on both sides of 70% of the street).

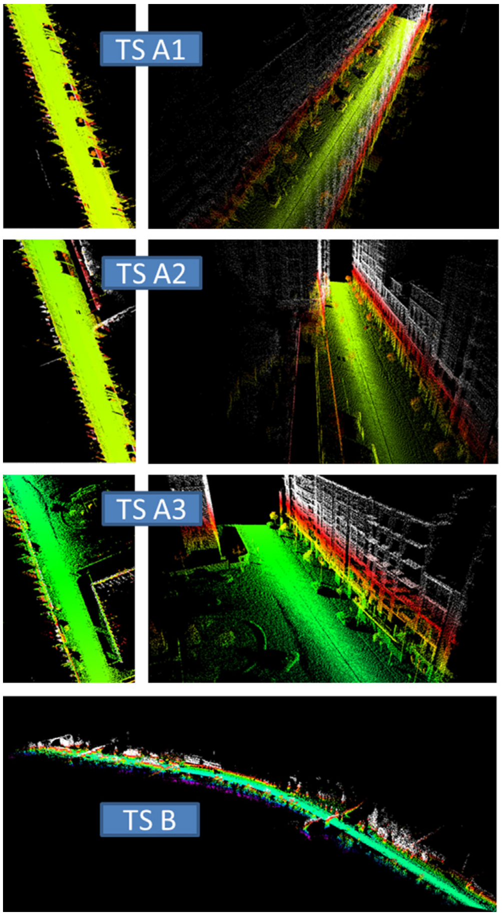


Fig. 6. Footprint and perspective view of the test sites.

In order to test the algorithm separately in different urban environments, three test sites (TS) were selected from Dataset A, all of them with a high density and diversity of poles and with different urban features (see Fig. 6): TS-A1 is a street section completely surrounded by buildings, TS-A2 contains part of a park and TS-A3 contains a big junction and part of a roundabout. The slope of the terrain also influenced the election of the test sites: TS-A1 is almost flat (i.e. the biggest height difference on the ground surface of that segment of the street is 0.5 m). However, in TS-A2 and TS-A3 the height difference is, respectively, 1.3 m and 3.5 m.

The length along the trajectory of each one of the "A" test sections is 100 m, and the number of points per sensor and TS is between 4 and 4.6 million. In the three test sites, features such as cars, vans, telephone boxes, banks, street billboards, trees, lampposts, traffic lights, traffic signs, large and small bins, motorcycles, bushes, street walls, columns and pedestrians were present.

3.1.2. Dataset B (Test site B)

Test site B comprises the whole Dataset B, which contains 41.5 million points from a street 820 m long *(Rúa Ribeira Sacra)*. The street goes along the Miño river and has many peri-urban and road features such as car parks, commercial centers, bridges (with heights between 7 and 30 m), open parks or guardrails (see Fig. 6). The pole like features present in the dataset are lamp posts, traffic signs, trees and bare poles. The street is curved and fairly flat (i.e. 2% slope on average), although there are pole-like objects at different height levels (e.g. four lamp posts are located over a bridge, 6 m above the ground level of the street).

3.2. Reference data

In both datasets, the reference data (i.e. location and description of the target poles) was collected in the field, by checking the position of the pole-like street furniture objects and their subsequent identification in the point cloud. The target poles are street furniture pole-like objects located on the streets: all types of lamp posts, traffic signs, traffic lights, bare poles, trees with a trunk diameter smaller than 30 cm, and other street furniture objects containing a pole (i.e. a pole based telephone box in Dataset A). Objects located more than 35 m away from the edge of the street were not considered.

Most of the target poles are detectable through simple visual inspection of the original point cloud. Nevertheless, the pole-like part of some objects is not visible, although some other features allow their identification in the point cloud. For instance: (i) In Dataset A, the trunk of a tree is completely occluded by a vehicle (i.e. there are no points of that trunk in the data), however, its branches are perfectly visible and its position is determinable. (ii) In Dataset B, a lamp post is surrounded by a bush, and the pole part of the signal is not identifiable, but the lamp itself is visible and its position is established analogously to example (i). In that way, the location of all the target poles detected visually in the field was determined (i.e. TS-A1: 25 poles, TS-A2: 23, TS-A3: 23 and TS-B: 122).

3.3. Algorithm settings

The same settings, from now on referred as ‘standard settings’, were applied to the 4 test sites (Table 2) in order to detect the pole-like street furniture present in all the point clouds. Applying these standard settings, the objects which fulfill the following three conditions are expected to be detected: (i) with a diameter smaller than 30 cm, (ii) a height over 1.2 m and (iii) with not more than 3 sparse points at their surroundings in each horizontal section (i.e. within a distance to the centroid greater than 15 cm and smaller than 45 cm).

Table 2
Settings applied to the four test sites.

Voxel size	0.1 m
Max. section area	0.06 m^2 (6 voxels)
Inner diameter	0.3 m (3 voxels)
Outer diameter	0.9 m (9 voxels)
Points allowed in the ring	3
Minimum height	1.2 m

3.4. Results

In Dataset A, the algorithm was applied to the data from Sensor 1 in the three TS independently. Moreover, in order to compare results, data from Sensor 2 was also used in TS-A1. In Dataset B (Test site B), the algorithm was applied to the points from both sensors at once.

As explained in Section 2, during the voxelization a reduced version of the data is created by fitting the original point cloud in a sparse 3D matrix, where only the cells or voxels containing points store information. In the three test sites from Dataset A, although the distributions of the original point clouds are heterogeneous, the reduction ratio of the voxelized version (i.e. the difference between the number of points in the original point cloud and the number of not empty voxels in the reduced version) is similar for all the test sets (see Table 3). In Dataset B (TS-B), using the data from both sensors, the reduction ratio of the voxelized version is similar to the one obtained in the “A” test sites, although this ratio is slightly smaller using the data from just one of the sensors.

3.4.1. Test site A1

As shown in Fig. 7 and Table 4, using the data from Sensor 1, in TS-A1, 21 out of 25 pole-like street furniture objects were identified. In addition to these, six other non-target poles were detected. All of the six additional poles were inside the footprint of the buildings. They were thin columns and/or vertical bars which were visible through the large windows of the commercial storefronts located along the street (see Fig. 8B).

All the four non-detected poles are trees. Three of them were not identified because of the occlusion by vehicles and bins. Due to this, the length of the part of the pole still detectable by the sensor was smaller than the threshold (i.e. <1.20 m) in two of those cases, and in the other one the pole was not visible at all (Fig. 8A). The remaining non-detected pole was affected by the presence of a pedestrian at its close surroundings, so too many points were detected inside the outer ring of some of its horizontal sections (see Fig. 8C).

Using the data from Sensor 2, two more street furniture objects were identified in the TS-A1. The rest of the detected features in this case were the same as for Sensor 1. Those two objects were not occluded by vehicles from Sensor 2, but two other trees were still undetected: (i) the one completely hidden (for both sensors) by a vehicle and (ii) the one affected by the presence of a pedestrian. As a result of the angle and point of view change from one sensor to the other, three more non-target poles were detected through the windows of the commercial storefronts comparing to Sensor 1.

Table 3
Compression of the voxelized version (1-number of voxels containing data/number of original points) % after voxelization using different voxel sizes for all the test sites. Bold values show the results obtained using the voxel size applied in the test sites (10 cm).

Voxel size (cm)	TS A1 Sensor 1 NOP: 4450203		TS A1 Sensor 2 NOP: 4330630		TS A2 Sensor 1 NOP: 4649906		TS A3 Sensor 1 NOP: 3988431		TS B Sensor 1 NOP: 21486735		TS B Sensor 1 + 2 NOP: 41541540	
	Voxels ×1000	Comp (%)	Voxels ×1000	Comp (%)	Voxels ×1000	Comp (%)	Voxels ×1000	Comp (%)	Voxels ×1000	Comp (%)	Voxels ×1000	Comp (%)
1	4413	0.8	4306	0.6	4617	0.7	3965	0.6	20032	6.8	38244	7.9
2	4171	6.3	4108	5.1	4362	6.2	3802	4.7	19016	11.5	33317	19.8
5	2396	46.1	2453	43.4	2470	46.9	2222	44.3	12182	43.3	17892	56.9
10	**880**	**80.2**	**902**	**79.2**	**916**	**80.3**	**831**	**79.2**	**6563**	**69.5**	**9184**	**77.9**
20	264	94.1	269	93.8	274	94.1	245	93.9	3357	84.4	4292	89.7
50	49	98.9	50	98.8	51	98.9	46	98.9	1018	95.3	1125	97.3
100	13	99.7	14	99.7	14	99.7	13	99.7	302	98.6	304	99.3

NOP: number of original points.
Comp: compression.

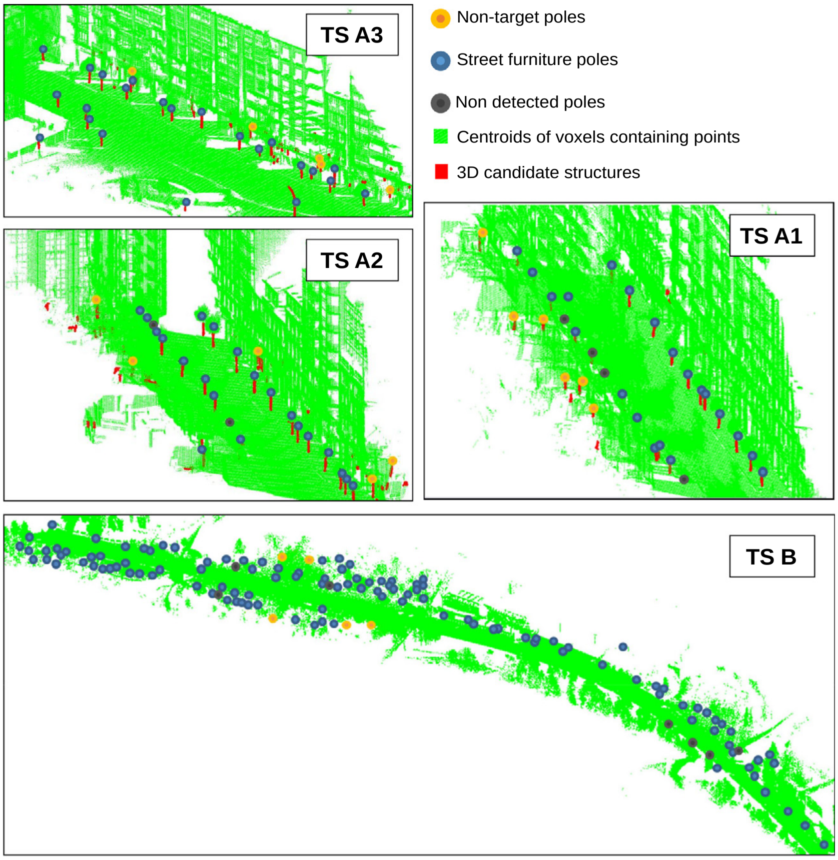


Fig. 7. Results of the application of the algorithm in the four test sites.

3.4.2. Test site A2

From a total of 23 pole-like street furniture objects, 21 were detected. The two non-detected objects (i.e. a traffic sign and a tree) were occluded by a bin and a vehicle respectively (see Fig. 8D).

Another two poles were detected within the limits of the buildings footprint, with the same characteristics as the non-target poles detected in Test Site A1 (see Fig. 7 and Table 4).

3.4.3. Test site A3

All the 23 pole-like street furniture objects located in Test Site A3 were successfully identified (see Fig. 7 and Table 4). Five non-target poles were detected within the surrounding buildings, under the same conditions as the ones detected in TS-A1 and TS-A2.

3.4.4. Test site B

Using the data from both sensors at once, and the standard settings (see Section 3.3), 115 out of 122 target poles were detected in TS-B (see Fig. 7 and Table 4). The pole of a lamp post, located over a bridge, was occluded for both sensors by the parapet of the bridge. Two traffic signs were too close to the guardrail, and the isolated pole-like part of them was not big enough to be detected using the standard settings (see Fig. 8E). Another traffic sign was not detected by the algorithm because it was too close to the wall of a bridge (see Fig. 8G). A bare pole and a lamp post (see Fig. 8F) were surrounded by bushes, and a tree had its trunk covered with ivy, thus the points from their sections were not isolated, and therefore not detected (see Section 2.2). In addition, five non-target poles were detected. All of them were long bare branches of big trees (see Fig. 8H for a graphic example).

3.5. Discussion

3.5.1. Setting parameters

The setting parameters for the proposed algorithm are configurable, and the correct selection of them directly influences the reliability of the results and the speed of the data processing. The voxel size is the most important parameter, since it directly affects the compression of the voxelized version (see Table 3), and the other configurable settings indirectly, as they are set forth in voxel units. The voxel size is the minimum unit that can be used when the settings are established. In fact, the values of the parameters have to be integer multiples of the voxel size (e.g. if the voxel size is 0.5 m, the minimum height of a pole could be set as 0.5 or 1 m, but not 0.7 m, as 0.5 m is the minimum unit). For these specific test cases, the voxel size was established at 0.1 m as (i) it produces a high reduction ratio of the voxelized version (70–80%. See Table 3), and (ii) it is small enough to allow the correct configuration of the rest of the parameters. Even using a large voxel size, a minimum of four voxels is needed in each inner horizontal section in order to ensure the inclusion of all the points from a section of a pole, and the outer diameter has to be at least two voxels bigger than the inner one. In that way, using, for instance, a voxel size of 0.3 m, the inner and outer diameters should be at least 0.6 m and 1.2 m respectively.

The other setting parameters shown in Table 2 are aimed at detecting street furniture objects which either contain or are shaped like a pole, avoiding the detection of false positives. Therefore, (i) a minimum height of 1.2 m was set, eluding to the selection of small pole like objects, such as small bars from rail guards or bollards, (ii) an inner diameter of 0.3 m and a maximum surface of 0.06 m^2 (see Section 2), which delimitates the horizontal section

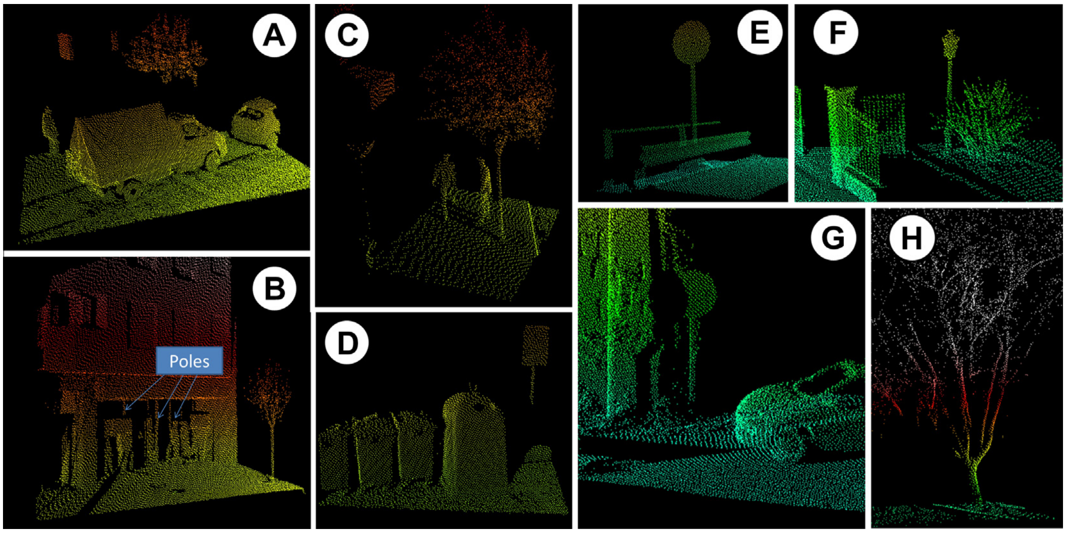


Fig. 8. A: Total occlusion of a pole (TS-1). B: Features detected inside the buildings through big windows in Dataset A. C: Pedestrian close to a tree (TS-A1). D: Partial occlusion from bins (TS-A2). E: Sign post close to the guardrail in TS-B. F: Lamp post surrounded by a bush. G: Sign post close to a wall. H: Almost-vertical bare branches of a tree in Dataset B.

and prevents the selection of wider features like bins, street billboards or pedestrians, and (iii) an outer diameter of 0.9 m that is set out in order to extract the isolated objects, and to avoid the selection of small patches from large features (e.g. walls or vehicles) affected by occlusions.

3.5.2. Performance of the algorithm

Using the proposed settings, all the pole-like street furniture objects are detected, with the only exception of those which are affected by (i) severe occlusions from large objects (i.e. poles occluded by vehicles or large bins in Dataset A, and a lamp post which was not visible because of the parapet of a bridge in Dataset B), or by (ii) the existence of other features in their close surroundings (i.e. a tree undetected in TS-A1 because of the presence of a pedestrian nearby, and some poles in TS-B surrounded by bushes or too close to guardrails or walls). In addition to these, no non-target poles were detected within the street limits in Dataset A. The only poles detected in the test sites from Dataset A, which were not a target (i.e. street furniture pole-like objects), were located inside the footprint of the buildings on both sides of the street. Note that these objects are, in all the cases, pole-shaped and they were detected exclusively through the large windows of commercial storefronts, and no other non-target objects were detected outside the buildings footprint. Consequently, the detected poles which are not street furniture objects, can be automatically eliminated by delimitating the buildings footprint. Some methods for automatic building footprint extraction from MLS data have already been developed (Rutzinger et al., 2011).

In Dataset B, five almost-vertical bare branches from trees were identified as poles (see Fig. 8H). The detection of those non-target pole-like objects could be avoided by using a larger outer diameter, although it could imply the misdetection of some target poles which are close to other features.

Alternatively, the general parameters could be changed and adapted to the datasets. For instance, lowering the minimum height, more target poles would be identified, but making the conditions less restrictive, more small poles would be detected through the windows of buildings in Dataset A. The number of undetected street furniture objects can be reduced by using a smaller outer diameter. For instance, the tree which was undetected because of the proximity of a pedestrian in TS-A1 could be separated from it (Fig. 8C). However, in Dataset B, another pole would remain undetected as it is completely surrounded by bushes.

The length of the horizontal footprint of the detected poles can be restricted in order to avoid the detection of slanting objects. However, in this test case, this restriction was not applied because we aimed to detect even inclined poles such as, for example, the tree trunk shown in Fig. 9.

3.5.3. Comparison with previous methods

Comparing the results with those obtained using other methods is not straightforward: (i) the targets of some of the previous methods are not exactly the same. For instance, (Wu et al., 2013) searched exclusively for trees, and the targets for (Golovinskiy et al., 2009) were all kind of urban features (not only pole-like objects). (ii) Some of them analyzed separately different kinds of poles and were focused in recognition and classification (Pu et al., 2011), and (iii) the test sites and the sensors are different (i.e. the point and pole distribution and density are different in all the datasets).

In (Lehtomäki et al., 2010, 2011), considering reference data (i.e. obtained exclusively through visual inspection of the point cloud), the completeness of the detection ranged between 69.7% and 77.7% for pole-like objects closer than 30 m to the MLS system, and between 71.6% and 83.5% for those closer than 12.5. The detection rates were lower considering visual field inspection (67.1% and 76.4% respectively). The correctness of the method was 86.5–95.1% for poles closer than 12.5 m to the trajectory and 81–86.5% for those up to 30 m away from the sensors.

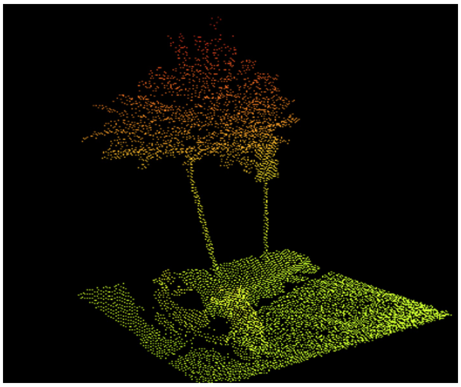


Fig. 9. Slanting pole-like object detected in TS-A1.

Table 4
Number and type of poles detected in the four test sites (A1-3 and B).

Test site	Sensor	Traffic lights	Traffic signs	Trees	Lamp posts	Telephone boxes	Bare poles	Total detection (ratio and completeness [%])	Non-target poles (number and correctness [%])
		(objects in the TS/detected objects)							
A1	1	–/–	4/4	17/21	–/–	–/–	–/–	21/25 (84.0)	6 (77.8)
A1	2	–/–	4/4	19/21	–/–	–/–	–/–	23/25 (92.0)	9 (71.9)
A2	1	1/1	6/7	13/14	1/1	–/–	–/–	21/23 (91.3)	2 (91.3)
A3	1	3/3	7/7	6/6	6/6	1/1	–/–	23/23 (100)	5 (82.1)
B	1 + 2	–/–	20/23	31/32	55/57	–/–	9/10	115/122 (94.3)	5 (95.8)
							Average:	92.3	83.8

Completeness (%): [Target poles detected]/[Total number of target poles].
Correctness (%): [Target poles detected]/([Target poles detected]+[Non target poles detected]).

A combined identification and recognition study of different categories was performed in two test sites in (Pu et al., 2011). The detection rate using one of the datasets was 86.9% for the pole category, and 63.5% for the trees, and 60.8% of poles and 29.5% of the trees in the other test site. The percentage of false positives ranged from 14.9 to 63.9 for the poles, and from 14.3 to 15.4 for the trees. In (Yokoyama et al., 2011), the recognition rate of target pole-like objects was 63.9%, and (Golovinskiy et al., 2009) detected 65% of the reference objects. The algorithm from (Wu et al., 2013) detected all the trees except one in two test sites. Correctness was higher than 98%, and the completeness was 100%.

Although the results comparison is quite complex, the values of completeness and correctness obtained with our algorithm are significantly higher than those from most of the previous methods (see Table 4). Only (Wu et al., 2013) detection rates (where the targets were exclusively trees) are higher. However, (i) their method designed for its use in flat terrain, although they mention an alternative for non-flat terrain, based on the transformation of the vertical trajectory into a flat line, and (ii) all the trees must be at the same height from the ground (even using above alternative).

In addition to the aforementioned comparison, the algorithm detects poles despite the presence of large structures attached to them, or the connection of the structures joined to the poles. Many target poles which were successfully detected in the two datasets were joined together (mainly through the branches of the trees). Some previous methods, such as (Pu et al., 2011; Yokoyama et al., 2011) seem unable to detect connected poles, as they use tridimensional connected components labeling before the pole identification. Furthermore, the method is fully automatic, thus no previous collection of training data is needed, whereas methods like (Golovinskiy et al., 2009) require a set of training objects.

Although the two datasets used in order to test the algorithm had the same scanning geometry (i.e. same sensors and scanning parameters were used in both cases), given the nature of the proposed algorithm, there is no reason why the method would not be able to work and achieve good results with any scanning geometry. The algorithm is completely independent of the angle of the sweeps, and does not need any indexation or special order of the points, as it works with the raw point cloud and only the *XYZ* coordinates are needed. However, (Lehtomäki et al., 2010, 2011) (i) have certain restrictions regarding the tilt angle of the sensors (i.e. it is based on the analysis of the sections of vertical poles), and (ii) requires the identification of the different sweeps and the order of each point within them.

4. Conclusions

In this study, a new algorithm for automatic identification and extraction of pole-like street furniture objects from MLS data is developed and subsequently tested in four test sites (TS). An initial simplification process is performed by fitting the original point cloud in a regular voxel space. This simplification method allows the use of data from almost any MLS system, as it transforms the original point cloud and fits it in a regular grid, thus avoiding irregularities produced due to point density differences within the point cloud. A two-dimensional analysis is then applied in order to detect pole-like sections according to their shape and area. Finally, the pole-like objects are reconstructed in 3D.

The method was tested in four test sites from two different datasets in urban and peri-urban environments. An average completeness of 92.3% (ranging from 84 to 100 in the four TS), and a correctness of 83.8% (from 71.9 to 95.8%) were achieved. Only some pole-like objects occluded by large features, or too close to other objects, remained undetected.

From both the results obtained in the test sites and the thorough analysis of the methods, we can conclude that this algorithm is more general and improves the performance of the existing methods by: (i) Detecting pole-like objects joined to other features or connected to other poles, (ii) it is fully automatic and does not require the use of training data, (iii) no initial assumptions about the relative location of the poles are needed, and (iv) it is independent of the scanning geometry and it only needs the *XYZ* coordinates of the original points.

Acknowledgments

This paper has been funded by Project BIA2011-26915 of the Spanish Ministry of Science and Innovation. CCG is in receipt of a “Severo Ochoa” PhD Grant provided by FICYT-Government of Principado de Asturias.

The authors thank the two anonymous reviewers for helpful comments.

References

Aijazi, A.K., Checchin, P., Trassoudaine, L., 2013. Segmentation based classification of 3D urban point clouds: a super-voxel based approach with evaluation. Remote Sensing 5 (4), 1624–1650.

Boulaassal, H., Landes, T., Grussenmeyer, P., Tarsha-Kurdi, F., 2007. Automatic segmentation of building facades using terrestrial laser data. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 36 (Part 3/W52), 65–70.

Brenner, C., 2009. Extraction of features from Mobile Laser Scanning data for future driver assistance systems. In: Sester, M., Bernard, L., Paelke, V. (Eds.), Advances in GIScience. Springer, Berlin Heidelberg, pp. 25–42.

Buch, N., Orwell, J., Velastin, S.A., 2008. Detection and classification of vehicles for urban traffic scenes. In: Visual Information Engineering. 5th International Conference, July 2008, pp. 182–187.

Dold, C., Brenner, C., 2006. Registration of terrestrial laser scanning data using planar patches and image data. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 36 (Part 5), 78–83.

Elseberg, J., Borrmann, D., Nuchter, A., 2013. One billion points in the cloud – an octree for efficient processing of 3D laser scans. ISPRS Journal of Photogrammetry and Remote Sensing 76, 76–88.

Escalera, S., Pujol, O., Radeva, P., 2010. Traffic sign recognition system with beta-correction. Machine Vision Applications 21 (2), 99–111.

Frueh, C., Zakhor, A., 2003. Constructing 3d city models by merging ground-based and airborne views. In: Computer Vision and Pattern Recognition. Proceedings. 2003 IEEE Computer Society Conference, June 2003, Vol. 2, pp. II–562.

Golovinskiy, A., Kim, V.G., Funkhouser, T., 2009. Shape-based recognition of 3d point clouds in urban environments. In: Proc. IEEE 12th International Conference on Computer Vision. IEEE, pp. 2154–2161.

Gonzalez-Aguilera, D., Crespo-Matellan, E., Hernandez-Lopez, D., Rodriguez-Gonzalvez, P., 2013. Automated urban analysis based on LiDAR-derived building models. IEEE Trans. Geosci. Remote Sensing 51, 1844–1851.

Gröger, G., Plümer, L., 2012. CityGML – Interoperable semantic 3D city models. ISPRS Journal of Photogrammetry and Remote Sensing 71, 12–33.

Haala, N., Brenner, C., 1999. Extraction of buildings and trees in urban environments. ISPRS Journal of Photogrammetry and Remote Sensing 54 (2), 130–137.

Holopainen, M., Vastaranta, M., Kankare, V., Hyyppa, J., Liang, X., Litkey, P., Xiaowei, Y., Kaartinen, H., Kukko, A., Kaasalainen, S., Jaakkola, A., Hyyppa, H., Vaaja, M., 2011. The use of ALS, TLS and VLS measurements in mapping and monitoring urban trees. Proc. Urban Remote Sensing Event (JURSE), 29–32.

Hosoi, F., Omasa, K., 2006. Voxel-based 3-D modeling of individual trees for estimating leaf area density using high-resolution portable scanning lidar. IEEE Trans. Geosci. Remote Sensing 44, 3610–3618.

Lehtomäki, M., Jaakkola, A., Hyyppa, J., Kukko, A., Kaartinen, H., 2010. Detection of vertical pole-like objects in a road environment using vehicle-based laser scanning data. Remote Sensing 2 (3), 641–664.

Lehtomäki, M., Jaakkola, A., Hyyppa, J., Kukko, A., Kaartinen, H., 2011. Performance analysis of a pole and tree trunk detection method for Mobile Laser Scanning Data. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 38 (Part 5/W12), 197–202.

Ling, C.X., 1995. Overfitting and generalization in learning discrete patterns. Neurocomputing 8 (3), 341–347.

Meagher, D., 1982. Geometric modeling using octree encoding. Computer Graphics and Image Processing 19 (2), 129–147.

Moskal, L.M., Zheng, G., 2012. Retrieving forest inventory variables with Terrestrial laser scanning (TLS) in urban heterogeneous forest. Remote Sensing 4, 1–20.

Optech, 2013. <http://www.optech.ca>. (accessed 1 August, 2013).

Pu, S., Rutzinger, M., Vosselman, G., Elberink, S.O., 2011. Recognizing basic structures from Mobile Laser Scanning Data for road inventory studies. ISPRS Journal of Photogrammetry and Remote Sensing 66 (6), S28–S39.

Puente, I., González-Jorge, H., Arias, P., Armesto, J., 2012. Land-based Mobile Laser Scanning Systems: a review. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 38 (Part 5-W12), 163–168.

Puente, I., González-Jorge, H., Riveiro, B., Arias, P., 2013. Accuracy verification of the Lynx Mobile Mapper system. Optics & Laser Technology 45 (1), 578–586.

Puttonen, E., Lehtomäki, M., Kaartinen, H., Zhu, L., Kukko, A., Jaakkola, A., 2013. Improved sampling for terrestrial and Mobile Laser Scanner point cloud data. Remote Sensing 5 (4), 1754–1773.

Ranzinger, M., Gleixner, G., 1997. GIS datasets for 3D urban planning. Computers, environment and urban systems 21 (2), 159–173.

Rutzinger, M., Höfle, B., Oude Elberink, S., Vosselman, G., 2011. Feasibility of facade footprint extraction from Mobile Laser Scanning Data. Photogrammetrie - Fernerkundung- Geoinformation 2011 (3), 97–107.

Sahin, C., Alkis, A., Ergun, B., Kulur, S., Batuk, F., Kilic, A., 2012. Producing 3D city model with the combined photogrammetric and laser scanner data in the example of Taksim Cumhuriyet square. Optics and Lasers in Engineering 50 (12), 1844–1853.

Shi, Y., Shibasaki, R., Shi, Z., 2008. Towards automatic road mapping by fusing vehicle-borne multi-sensor data. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 37 (Part B5), 867–872.

Tao, C.V., 2000. Mobile mapping technology for road network data acquisition. Journal of Geospatial Engineering 2 (2), 1–14.

Truong-Hong, L., Laefer, D.F., Hinks, T., Carr, H., 2013. Combining an angle criterion with voxelization and the flying voxel method in reconstructing building models from LiDAR data. Computer-Aided Civil and Infrastructure Engineering 28 (2), 112–129.

Vaaja, M., Hyyppa, J., Kukko, A., Kaartinen, H., Hyyppa, H., Alho, P., 2011. Mapping topography changes and elevation accuracies using a Mobile Laser Scanner. Remote Sensing 3 (3), 587–600.

Vanderhyde, J., Szymczak, A., 2008. Topological simplification of isosurfaces in volumetric data using octrees. Graphical Models 70 (1–2), 16–31.

Vosselman, G., Gorte, B.G., Sithole, G., Rabbani, T., 2004. Recognising structure in laser scanner point clouds. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 36 (Part 8/W2), 33–38.

Wu, B., Yu, B., Yue, W., Shu, S., Tan, W., Hu, C., Huang, Y., Wu, J., Liu, H., 2013. A voxel-based method for automated identification and morphological parameters estimation of individual street trees from Mobile Laser Scanning data. Remote Sensing 5 (2), 584–611.

Yokoyama, H., Date, H., Kanai, S., Takeda, H., 2011. Pole-like objects recognition from Mobile Laser Scanning Data using smoothing and principal component analysis. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 38 (Part 5/W12), 115–120.

Zhang, G.P., 2000. Neural networks for classification: a survey. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions 30 (4), 451–462.

Zhou, Q.-Y., Neumann, U., 2013. Complete residential urban area reconstruction from dense aerial LiDAR point clouds. Graphical Models 75 (3), 118–125.

Zin, T.T., Hama, H., Koh, S.S., 2007. Robust signboard recogniton in the presence of occlusion and reflection. International Journal of innovative Computing, Information and Control 3 (6A), 1321–1334.

Capítulo III

Detección de Superficies

[*Mobile Laser Scanner data for automatic surface detection based on line arrangement*]

Automation in Construction 58 (2015) 28–37

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Mobile Laser Scanner data for automatic surface detection based on line arrangement

C. Cabo *, S. García Cortés, C. Ordoñez

Department of Mining Exploitation, University of Oviedo, 33004 Oviedo, Spain

ARTICLE INFO

Article history:
Received 27 February 2015
Received in revised form 14 May 2015
Accepted 11 July 2015
Available online 25 July 2015

Keywords:
MLS (Mobile Laser Scanning)
Surface detection
Point cloud
Algorithm

ABSTRACT

An algorithm for automatic detection of planar and quasi-planar surfaces from Mobile Laser Scanning (MLS) data is proposed. The method uses line clouds for efficient data reduction of point clouds from MLS. The singular geometry of the MLS data on planar surfaces is used to transform the original point cloud into a more structured line cloud, which allows the simplification of the initial data and identification of surfaces by grouping lines. From each profile in the original dataset, strings of aligned points are identified, and a line cloud is defined by the endpoints of these strings. Lines are subsequently grouped following a set of parallelism, proximity and merging rules. The algorithm was tested using an urban dataset, and validated on 27 surfaces, by assessing the correctness and completeness of the point and line grouping. Correctness was, in all the surfaces, higher than 99%, and completeness was 90% on average.

1. Introduction

Laser scanning surveying techniques have been widely and increasingly used in many fields (such as forestry, or road and urban planning), in the last few decades [1–4]. The use of these techniques, typically produces a 3D point cloud, whose density, accuracy and spatial point distribution may significantly vary, depending on the platforms or surveying devices used.

From the early 90s the use of ALS (Airborne Laser Scanning) became widespread, as it provides spatial information of vast areas in a short period of time, with an adequate point density and accuracy for large object detection and extraction [5,6]. Later, the first Terrestrial Laser Scanners (TLS) appeared. They provide, in general, more accurate and dense point clouds than ALS, but they have some spatial limitations produced by their static nature, such as occlusions, or higher spatial heterogeneity within the point cloud. The data from ALS systems is usually obtained from an almost-nadiral perspective, which confers to the ALS point clouds a homogeneity that TLS data lacks [7]. However, for the same reason, and contrary to ALS, TLS systems are often able to register points on vertical surfaces.

In the 2000s, the first commercial Mobile Laser Scanners (MLS) arose [8–10]. These systems usually consist in (i) one or more LiDAR (Light Detection and Ranging) sensors, (ii) an IMU (Inertial Measurement Unit), and (iii) a GNSS (Global Navigation Satellite System) device, all of them deployed on a van or another type of vehicle [11–13]. MLS are able to avoid some of the said drawbacks of ALS and TLS systems: (i) point accuracy and density can be significantly higher than those of ALSs, (ii) point clouds often follow a certain pattern (i.e. produced by the movement of the sensors and the fixed relative perspective from the MLS vehicle), or the fact that (iii) some occlusions are avoided by using more than one sensor [14–17]. Nevertheless, MLS systems have some disadvantages and limitations. For instance: (i) point accuracy relies on the GNSS and the IMU, thus it is usually lower than that obtained with TLS systems, and (ii) the accessibility of the scanning targets or areas is limited by the vehicle or platform [18–20,9]. More recently, some other devices or variations of the aforementioned systems have appeared. For example, the Finnish Geodetic Institute has developed a Personal Laser Scanner (PLS) that goes beyond some of the accessibility limitations of the MLS systems [21,9].

Feature identification through visual recognition of a point cloud from a laser scan often constitutes a relatively simple task, in the sense of using adequate visualization software that even a non-trained user would be able to identify several features in the point cloud, when there is a prior knowledge from reality, e.g. a car, a building, or a tree. Such methods are widely used in order to test the accuracy of the systems and the performance of algorithms for automatic feature extraction or segmentation from LiDAR point clouds [20,22–26]. However, detection and recognition from visual inspection is a very time and resource consuming task.

Algorithms for feature detection have arisen and evolved along with LiDAR platforms and devices. Some algorithms/methods are general and therefore susceptible of being applied to any 3D point cloud [27–30], whereas, some other methods are specific for a target geometry, a point of view, or any other particular characteristic of a platform, device, or a special setting of them [23,31–34].

One of the main targets for automatic detection algorithms from 3D point clouds is surface detection and identification, since planar or ruled

* Corresponding author at: Dept. of Mining Exploitation, C/Independencia 13, 33004 Oviedo, Spain.
E-mail address: carloscabo.uniovi@gmail.com (C. Cabo).

http://dx.doi.org/10.1016/j.autcon.2015.07.005

surfaces are, together with linear and cylindrical features, the most used shapes for geometrically defining the majority of the objects represented in urban and road cartography and models [35]. Some studies have addressed algorithms and methods for surface identification and extraction from MLS data in the last few years. However, most of them analyse exclusively planar surfaces, or use plane fitting techniques for all the surfaces.

In 2011, Jochem et al. [36] published a method for the extraction of vertical facades from MLS point clouds based on the use of Hough transform. In the method, seed points are randomly selected, and using a region growing algorithm, the plane parameters are estimated. This study was focused on vertical wall detection, and the detected planes are reduced to 2D lines in a vertical projection. Neither completeness nor correctness were specifically assessed in this study, but the authors affirmed that completeness could reach 50–74%.

More recently, Fan et al. (2014) [37] used an algorithm for general man-made objects, including buildings. The original data was divided into three layers using the height above ground level (AGL). Three fixed values were set for AGL ([i]: below 2 m, [ii] between 2 and 5 m, and [iii] above 5 m), and the points within each layer were projected onto a horizontal raster. Finally, a set of rules was established in order to identify objects from the footprints in the three layers. Similar footprints in at least the first two layers were considered a possible vertical façade of a building, which was subsequently checked using its convex hull and neighbourhood. The performance of the method was analysed by checking the detection rate (70% for buildings) and a further classification of the detected features. Misdetections were due to occlusions and low point density in the horizontal layers that the algorithm set.

Although Yang and Dong [38] was specially focused on pole-like object segmentation, it addressed a method for planar surface segmentation that included the use of point intensities. Points were classified according to their attributes after a PCA (Principal Components Analysis) transformation for a further point cloud classification using SVMs (Support Vector Machines). Finally, segmentation was performed by using a set of geometric rules and a merging operator. Merging methods for planar surfaces were based on normalized cuts, taking into account Euclidean distance between patches and the angle between normal vectors. The performance of the method was analysed by evaluating the classification and segmentation precision (up to 96.8%) and recall (up to 93.4%).

Lari and Habib [39] showed algorithms aiming at identifying and segmenting planar and linear or cylindrical features/objects, taking into account local point densities. Points were classified using PCA, and subsequently clustered using plane fitting and adaptive cylinders. The detected features were finally extracted using a parameter-domain segmentation approach. The algorithm was tested using both real and simulated MLS, ALS and TLS datasets. As regards the simulated MLS data (i.e. a synthetically generated MLS point cloud), 93.3% of the planar surfaces were detected. For real test data, a threshold was established by setting an acceptable noise level (i.e. distance from each point to the fitted plane). In 45% of the surfaces, the noise level was lower than the threshold (established at 4 cm). Misdetections were due to the lack of density and PCA classification errors.

Some other studies (not specifically focused on surface extraction or MLS data processing) used the concept of 2D profiles. For instance, the algorithm shown in Sithole and Vosselman [40] used the intersections of two orthogonal sets of profile lines in order to classify ALS points into three different categories: (i) bare terrain, (ii) detached objects, and (iii) attached objects. Unconnected profile lines were obtained by applying a set of connectivity rules, for their subsequent grouping and cluster classification based on line group size, height and connectivity.

In 1994, Jian and Bunke [41] proposed a method for the segmentation of planar surfaces in range images of simple objects using line grouping (based on the work from Pavlidis and Horowitz [42]). More recently, Howarth et al. [43] used the same principles for surface extraction from image and range data.

Single scan lines from MLS data were analysed in by Lin and Hyyppä [44], using the k-segments defined by Verbeek et al. [45], for two-dimensional primitive fitting, and the use of similar methods for further three-dimensional geometrical analysis is suggested.

Lehtomäki et al. [23] suggested the concept of grouping consecutive scan lines for pole-like object detection. This study developed the idea of segmenting single scan lines, for a subsequent line grouping process based on shape and position attributes of the segments.

The objective of this work is to develop an algorithm for surface detection from MLS data, based on simple geometric principles that overcomes some of the limitations of the existing methods: (i) It must not be limited to the detection of plane surfaces. It is frequent that some of the main targets of surface detection algorithms from LiDAR data are not completely flat, e.g. some slightly curved building facades, road surfaces or walls. The proposed algorithm has to be able to deal with some non-flat surfaces (i.e. ruled and/or slightly curved surfaces). (ii) It must not be limited to vertical surfaces (as algorithms exclusively focused on vertical facades), but it has to be able to detect them, even in the cases of non-strictly-flat surfaces. (iii) It simplifies the point cloud into a smaller, meaningful and easy-to-deal-with line-based structure. This line structure is based on previous work related with 2D and scan lines profiling [41,43–45]. (iv) It has to be fully automatic, thus no training data is required, contrary to methods that use supervised classification techniques, and (v) the only data required by the algorithm are XYZ coordinates and time attribute of the original points, hence no other data, such as point intensities, are needed.

The proposed algorithm is based on an initial structured simplification and transformation of the point cloud into a line cloud, i.e. set of straight segments generated and organized from the original point cloud. The line cloud is stored in a group of vectors, with a very simple structure that avoids both duplicates and nonessential points. Lines are subsequently grouped following a set of simple geometric rules, and points and lines conforming the target surfaces are identified and labelled.

2. Methodology

MLS systems are typically based on rotating LiDAR sensors, which take measurements of points that could be grouped as lines when they hit regular surfaces. For instance (as shown in Fig. 1A) most of the existent MLS systems would create, if they were used in a cylindrical tunnel (following the directrix line), a string of points that could be interpreted as a helical line. In the same way, using the MLS system along an ideal rectangular-section tunnel or a very simply-shaped street would create a group of connected straight lines (see Fig. 1B–C). The spatial configuration of those lines depends on the setting parameters of the system, i.e. speed of the vehicle, sensor trajectory, scanner orientation, and sensor measurement and rotation rates [17,23].

Our method identifies planar or ruled surfaces from straight segments that can be extracted from the scanned profiles. Strings of points can be considered as lines (i.e. polylines whose nodes are the points from a profile). However, in general, when the laser beams hit some ruled or plane surfaces, the polylines that can be formed consist of a group of consecutive and aligned segments. These small segments, which join each point from a profile with the next one, can be grouped and, therefore, simplified by eliminating all the intermediate points, thus just keeping the first and last. In that way, all the points from the same profile and the same plane surface can be simplified and represented by a single straight segment.

Once the original point cloud is transformed into straight segments, i.e. the point cloud is transformed into a line cloud, they are grouped following parallelism and node-proximity criteria.

The process is carried out in four stages:

1. Identification of polylines in the point cloud. At this first stage, points are considered nodes of polylines and the possible gaps are identified in order to split different lines.

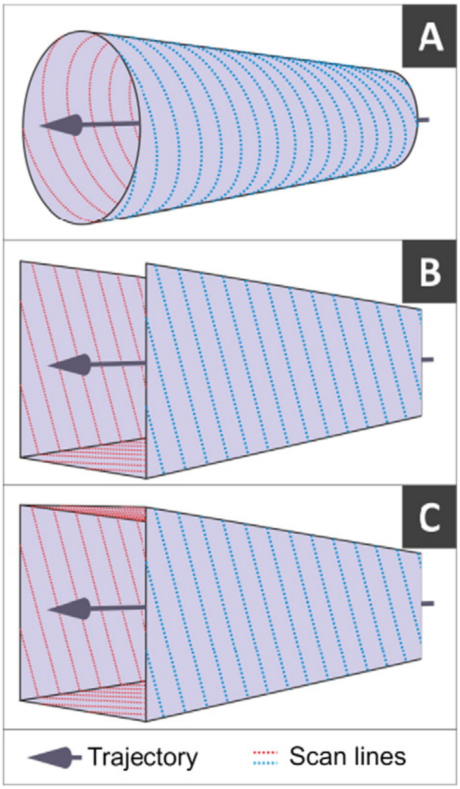


Fig. 1. Spatial configuration of the profiles from an MLS system on simple ideal surfaces/environments. A: Cylindrical tunnel. B: Simple corridor-type street. C: Rectangular-section tunnel.

2. Transformation and simplification of the polylines. Polylines are transformed into straight segments (line cloud).
3. Grouping processes. Straight segments are grouped in order to identify the different surfaces present in the point cloud.
4. Vertical surface identification. Normal vectors to the surface are obtained and analysed.

2.1. Identification of polylines

The start point of the algorithm is a set of points, taken from the same linear/profiling sensor, from which X, Y and Z coordinates and time are known. The time attribute allows knowing the order in which the points were collected.

Using the point order given by the time, a single and continuous polyline would be obtained by joining every single point in a dataset to its immediate consecutive. For instance, a single helical line would result from the ideal dataset represented in Fig. 1A. However, as shown in Fig. 2, some gaps due to the lack of continuity mainly produced by occlusions, or by the presence of open areas must be taken into account. In order to split the different polylines, distances between each point and its immediate consecutive are calculated. A distance threshold between consecutive points is set, and in cases where this limit is surpassed, the polyline is split. In that way, the number of polylines depends on the complexity of the structures found in each profile (see Fig. 2).

An example of a continuous polyline splitting can be found in Fig. 3. (i) Gaps are identified between points: (10–11), (11–12), (16–17) and (19–20). (ii) Polylines are created by the node strings: (1–10), (11), (12–16), (17–19) and (20–21). (iii) Polylines are split and simplified in subsequent stages.

2.2. Polyline simplification. Straight segments

At this stage, the polylines are simplified and transformed into straight segments using a three-dimensional version of the Douglas Peucker algorithm that reduces the number of nodes of the polylines by setting a maximum distance to the original line [46–48]. Polylines are delimited by gaps, and first and last points are kept. A straight line is set from the endpoints of each polyline and distances from all the intermediate points to this line are calculated. If all the distances are smaller than the threshold (i.e. maximum distance), the line is transformed into a single straight segment, which is defined by the original first and last nodes. If one or more nodes are further than the threshold, the furthest node (i.e. the node whose distance to the line is greatest) is set as an endpoint, thus the polyline is split in two: (i) from the first point to the new endpoint (i. e. pivot, break point, or new last point), and (ii) from the new endpoint to the original last point. This process is repeated in each polyline (including the split ones) until no distances greater than the threshold are detected.

For instance, in the first polyline in Fig. 3 (i.e. nodes from 1 to 10), distances from all the intermediate points to the straight segment 1–10 are calculated. In this case, more than one point is further than the threshold, hence the furthest of them (i.e. node 6) is set as an endpoint, and the original polyline is split into two: polyline 1A (nodes 1 to 6), and polyline 1B (nodes 6 to 10). Polyline 1A does not have any node whose distance to the line 1–6 is greater than the threshold, thus the string of nodes is transformed into a single straight segment (1–6) (red line in the figure). Polyline 1B is divided once again and the strings of nodes are transformed into two straight segments: (6–9) (green segment), and (9–10) (blue segment).

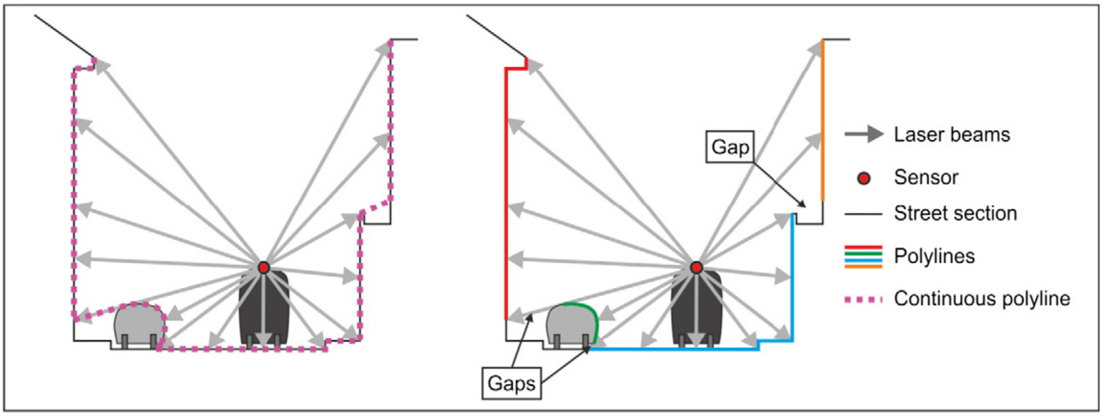
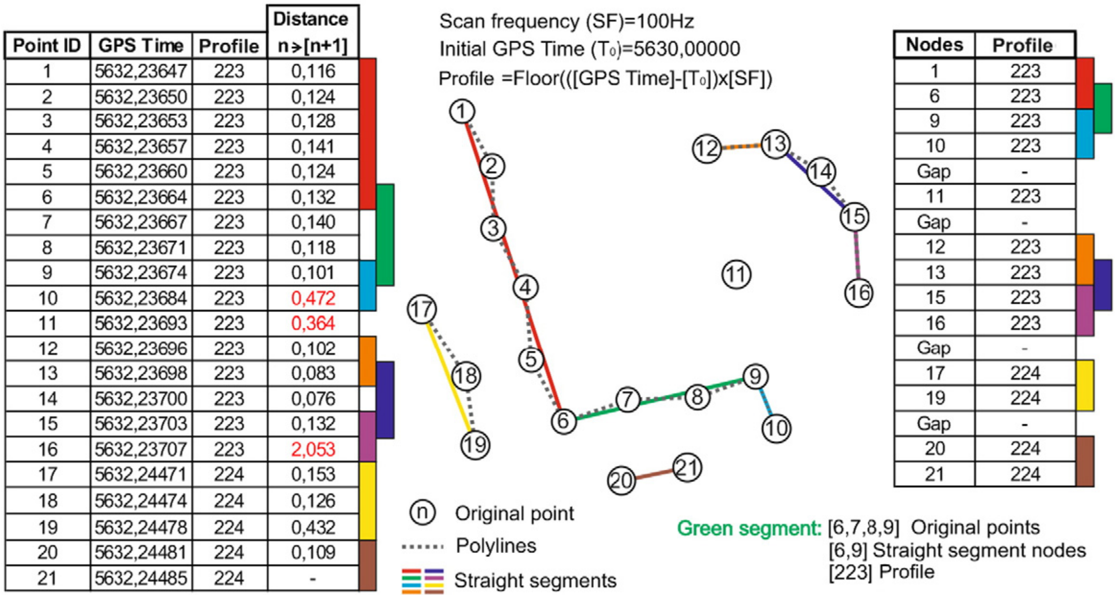


Fig. 2. Continuous polyline and polylines obtained after the gap-detection process.



Point ID	GPS Time	Profile	Distance n→[n+1]
1	5632,23647	223	0,116
2	5632,23650	223	0,124
3	5632,23653	223	0,128
4	5632,23657	223	0,141
5	5632,23660	223	0,124
6	5632,23664	223	0,132
7	5632,23667	223	0,140
8	5632,23671	223	0,118
9	5632,23674	223	0,101
10	5632,23684	223	0,472
11	5632,23693	223	0,364
12	5632,23696	223	0,102
13	5632,23698	223	0,083
14	5632,23700	223	0,076
15	5632,23703	223	0,132
16	5632,23707	223	2,053
17	5632,24471	224	0,153
18	5632,24474	224	0,126
19	5632,24478	224	0,432
20	5632,24481	224	0,109
21	5632,24485	224	-

Scan frequency (SF)=100Hz
Initial GPS Time (T_0)=5630,00000
Profile =Floor(([GPS Time]-[T_0])x[SF])

Nodes	Profile
1	223
6	223
9	223
10	223
Gap	-
11	223
Gap	-
12	223
13	223
15	223
16	223
Gap	-
17	224
19	224
Gap	-
20	224
21	224

Fig. 3. Example of polyline splitting, simplification and transformation into straight segments using Douglas Peucker algorithm. Colours on the right-hand side of the tables represent point/node membership to a segment.

The result of this process is a vector containing the indices of the endpoints of the straight segments. In the same way, four extra parameters are obtained at this stage for each straight segment: profile (i.e. index that labels the set of points captured by the scanner within each mirror revolution), length, tilt angle and azimuth. All of them are stored in independent vectors, and referred to the position of the first point of the straight segment.

2.2.1. Profile of each straight segment

A profile consists of all the points taken within a full turn of the sensor. The profile where point belongs is directly obtained from the time attribute if the scan frequency is known. Scan frequency (SF) represents the number of turns of the sensor head per time unit. Therefore, using the first point as start marker, the profile of a point can be easily obtained applying the equation shown in Fig. 3, where an example of profile calculation is performed for all the points. Note that this operation could be also performed by using the scan angle from the sensor.

It is possible that a segment has its nodes in different profiles, for that reason, and in order to avoid ambiguity, the profile of the first point is assigned to the whole segment.

2.2.2. Length, tilt angle and azimuth of the straight segments

The length, tilt angle and azimuth of all the straight segments are calculated using the coordinates of their nodes. In order to avoid duplicates, as shown in the left table in Fig. 3, nodes belonging to two straight segments appear only once. In that way, a node followed by a gap in the left table represents either the last node of a straight line, or an isolated point (i.e. not belonging to any segment), like the node 11 in the figure.

Every single node is checked and, if the following node is not a gap, XYZ coordinates are identified by the point ID, and extracted from the original point list.

Length and tilt angle are calculated from the XYZ coordinates, whereas azimuth is obtained from the XY coordinates exclusively.

2.3. Line grouping

Planar and some ruled surfaces are represented by a number of parallel straight segments (i.e. segments obtained through the simplification of the polylines). Those segments are usually in consecutive profiles, and at least one of the nodes of each simplified line is generally close to a parallel segment (in the following profile), that belongs to the same surface.

The method groups lines representing planar or ruled surface by selecting seed lines and seeking similar segments. The similarity criteria are based on node proximity and parallelism of the segments from contiguous profiles. The grouping process is divided into three steps: (i) selection of the initial seed line, (ii) selection of candidate lines, (iii) election of the line/s, and secondary seed lines. The process is repeated until all the lines are assigned to a surface or a maximum number of loops is reached.

An empty vector with the same length as the data vectors (e.g. azimuth vector) is created, and a code is assigned to the lines belonging to the same surface. In this way, at any time during the process, it is possible to know if any straight segment is already related to a surface (or a seed line).

2.3.1. Selection of the initial seed line

Seed lines are straight segments that are used in order to set the initial parameters of the grouping processes. The longest line not belonging to any surface is selected as a seed line. When the process starts, no straight segment is associated to any surface in the code vector, thus the initial seed line is the longest line. Once a seed line is selected, its attributes (i.e. profile, tilt angle, azimuth and node coordinates) are used towards the selection of the rest of the lines belonging to the same plane or ruled surface.

2.3.2. Selection of candidate lines

Candidate lines are selected for each seed line. These lines must: (i) belong to the following profile to the seed line, and (ii) be parallel to the seed line.

Parallelism is checked comparing tilt angle and azimuth values from the seed line to all of the lines in the following profile. Maximum discrepancy values for both parameters are established, and only the straight segments having a tilt angle and azimuth within the angular limits are considered parallel to the seed line, and therefore seen as candidate lines. See Fig. 4 for an illustration of candidate line selection from a seed line.

2.3.3. Election of candidate line/s and secondary seed lines

For two straight segments from consecutive profiles (e.g. a seed line and any of its candidate segments), parallelism is necessary, but not sufficient condition for a membership of the same group.

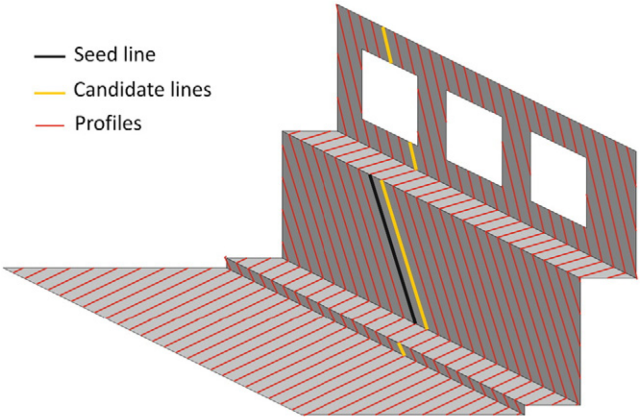


Fig. 4. A set of profiles from an MLS from a street scene, and an example of candidate segment selection with a seed line.

Straight segments from consecutive lines belonging to the same surface are required to be almost parallel (i.e. some angular differences may also appear because of deviations in the sensor trajectory, or in the case of non-planar surfaces. See Section 2.3.2). However, parallel lines within the same profile might be from different surfaces. For instance, as shown in Fig. 4, four candidate segments are selected from a seed line, but only one of them belongs to the same surface with it.

The election criterion is based on the proximity of the nodes of the candidate segments to the nodes of the seed line. Distance from the start-node of the seed line to the start-node of each candidate segment, and distances between end-nodes are respectively calculated.

Both the shortest distance between line start-nodes and between the end-nodes are selected, and if either of them is smaller than a pre-established threshold, the candidate line is elected and joined to the same surface as the seed line. Eight different cases of positive candidate election are considered, as explained graphically in Fig. 5:

Case 1. The two nodes from a single candidate line are the closest, and are within the distance threshold.

Case 2. There are no end-nodes close enough to the seed end-point, but there is a start-node that fulfills the proximity conditions.

Case 3. Similar to Case 2, but, in this case, the link is established between the end-points.

Case 4. Both nodes satisfy the conditions, but they belong to different segments. In this case, the two candidates are elected and joined to the surface.

When the grouping process starts, an initial seed line is selected. First, candidate lines are searched for in the following profile. Following the example in Fig. 6, the first profile to be analysed is S[n + 1] (being [n] the profile from the initial seed line). In S[n + 1] there is a segment that fulfills the conditions, thus the line is elected, and it becomes a secondary seed line. Trajectory or surface variations may produce a progressive lack of parallelism between the initial seed line and some straight segments located several profiles away. For that reason, and in order to absorb small angular variations, the seed line varies throughout the search, i.e. when a candidate line is elected, it becomes the new seed line (a secondary seed line). Parallelism and node proximity are similarly checked from any secondary seed line.

Straight segments on a surface can be split or divided by occlusions or presence of some objects (e.g. a door and a window in Fig. 6). Aiming at representing complex surfaces, these cases are referred in the algorithm by allowing double lines to be elected as candidate lines, and therefore in secondary seed lines. For instance, in Fig. 6, from the secondary seed line in Profile [n + 1], two segments are elected in Profile [n + 2] (see Case 4), and subsequently they form a secondary seed line. From this double seed line, another double seed line is elected (see Case 8 in Fig. 5), and so forth until there is no candidate that complies with the conditions. When the search in positive profiles (i.e. S[>n]) finishes, the process starts again in the opposite direction (i.e. starting in S[n − 1]) from the initial seed line.

Cases 5 to 8 address how a candidate line can be elected from a double seed line. In these cases, distances are exclusively calculated from the start node of one of the segments of the seed line, and from the end-node of the other one. Operatively, it works as if there was a single seed line, thus Cases 5 to 8 in Fig. 5 are analogous to Cases 1 to 4.

Even allowing for double lines, it is not always possible to represent some complex surfaces (e.g. see Fig. 7, Line group (1)). For that reason, a subsequent surface linking process is carried out. This process consists of assigning the same surface code to the groups of lines that share at least one common line. This is shown in Fig. 7:

Line group (1): All the blue segments are linked to the initial seed line (1).

Line groups (2) and (3): The longest segment not belonging to any surface is selected (i.e. Seed line (2), and then Seed line (3)), and all the possible links are established.
Line groups (1 + 2 + 3): The three line groups share lines, thus a single surface code is assigned to all of them.

2.4. Vertical surface identification

Non-completely-planar, but vertical surfaces are common among walls or façades on the sides of a curved street or road. Fig. 8 shows the surface of a slightly curved wall or façade, and the result of fitting a plane to the nodes of this line group.

The two parallelism thresholds, along with the Douglas Peucker tolerance, allow that some ruled surfaces, as the one in Fig. 8, are detected by the algorithm, even if the scanning plane of the sensor is tilted.

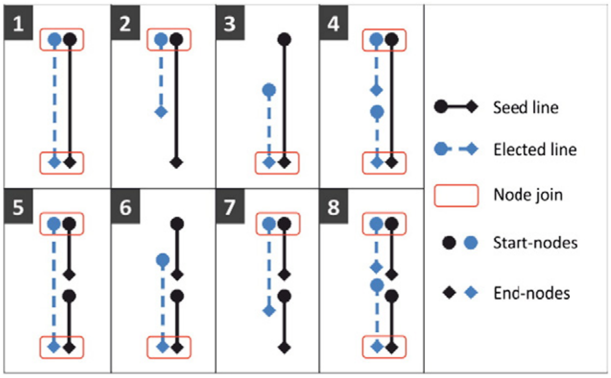


Fig. 5. Cases of candidate segment election.

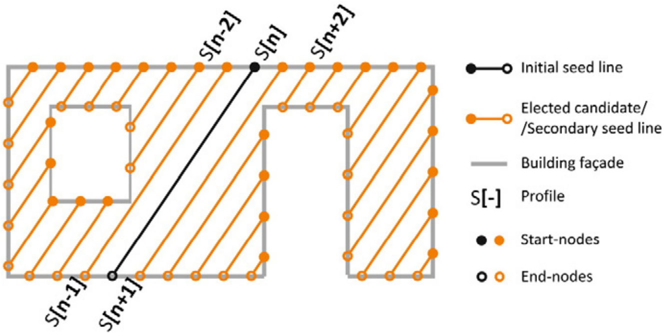


Fig. 6. Line grouping on a building façade.

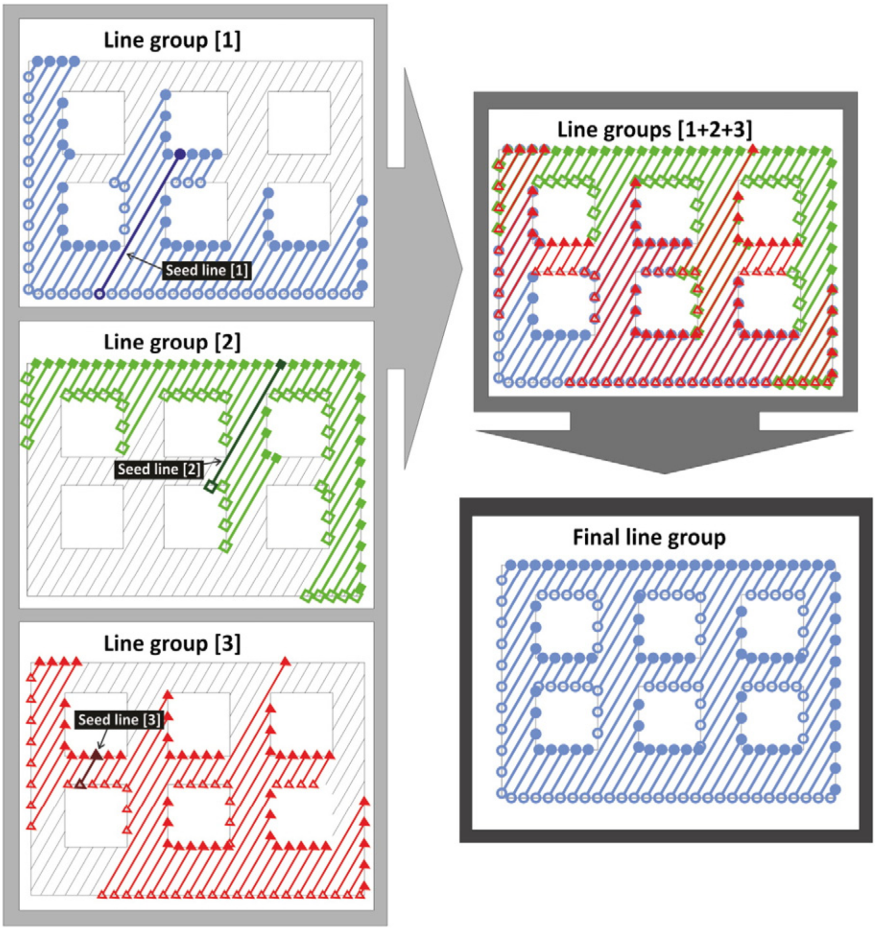


Fig. 7. Line group links for complex surface representation.

In order to determine verticality of the analysed surfaces, whether they are completely planar or not, least square plane fitting is individually applied to each node group. The surfaces are subsequently susceptible of being classified as (i) vertical, or (ii) non-vertical, by establishing an angular threshold for the normal vector of the fitted planes. If the tilt angle of a normal vector is small enough (i.e. smaller than the angular threshold), the surface is classified as vertical.

3. Experimenting on MLS data

3.1. Test data

The algorithm was tested using an MLS dataset from the Riegl VMX-250. This system integrates two LIDAR sensors (VQ-250), and an IMU/GNSS unit, and it is deployed on the rear top of a van. The VQ-250 is a rotational sensor that acquires points with a 360° field of view on planes set, in this case, at 45° to the horizontal and 45° to the trajectory (i.e. driving direction). See Table 1 for the system specifications [49].

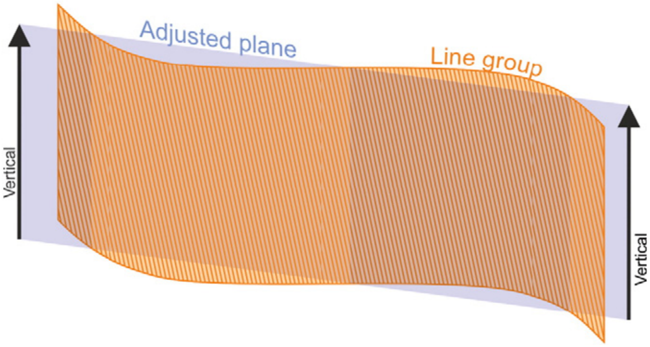


Fig. 8. Curved vertical surface.

The measurements were made along Stadtgraben Street, a 330 m long street in Horn (Austria). It is a slightly curved street, that has a 4% slope on average. Many urban features, such as houses, buildings, walls, street billboards, lampposts, traffic signs, fences and trees are present in the dataset, as well as pedestrians and vehicles.

The algorithm was applied to the whole dataset, although the validation was performed on a group of surfaces present in the street, most of which were vertical façades.

Two isolated façades (F1 and F2) and three groups of surfaces (A, B and C) were chosen in the street. Fig. 9 shows the results of the application of the algorithm to the dataset, and the location in the street of the features selected for validating the algorithm.

Table 1
Manufacturer specifications for the Riegl VMX-250.

Maximum range	200 m, 20%
Range precision	5 mm (1σ)
Absolute accuracy[a]	±5 cm (1σ)
Laser classification	IEC/CDRH class 1 eye-safe
Pulse repetition rate (PRR)	300 kHz/sensor
Scan frequency (SF)	100 Hz/sensor
Roll and pitch (IMU)	0.005° (1σ)
Heading (IMU)	0.015° (1σ)

[a] To meet the absolute accuracy, the GNSS data must be of sufficient quality.

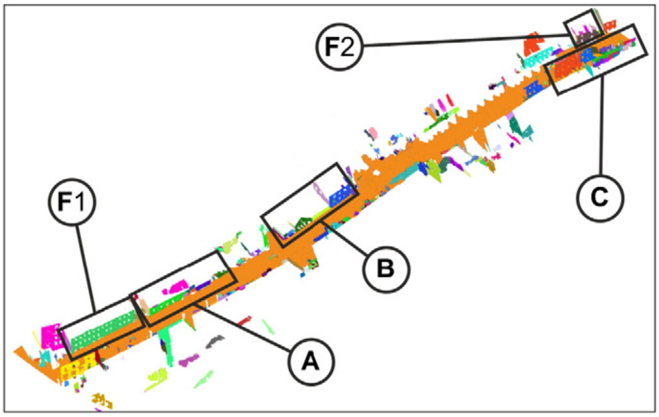


Fig. 9. Groups of lines in the dataset represented with random colours. Black rectangles enclose the surface groups A, B and C and the isolated façades F1 and F2 selected for validation.

3.2. Validation procedure

The performance of the algorithm was validated by comparing the number of points registered by the sensor from each surface, and the number of points assigned by the algorithm to a group representing that particular surface. Membership to the surface (i.e. identification of the points that belong to a certain feature) was obtained through interactive procedure, and was checked by three different operators. The procedure consisted of simultaneous visualization of the point cloud and the grouped lines, and identification one-by-one of (i) the points that, belonging to the actual surface, were not joined to the group, and (ii) the points that belong to the group, but were not on the surface. In this way, completeness and correctness values were calculated as follows:

- Correctness: Number of correctly grouped points (i.e. points on the actual surface, that belong to a group that defines the surface), divided by the total number of points belonging to the group. For instance: in the case of a façade and a group of lines (containing points) that covers it, correctness would represent the number of grouped points which are on the façade divided by the total number of points belonging to the group (i.e. both on the façade and not on it).
- Completeness: Number of correctly grouped points, divided by the total number of points detected by the sensor on the surface, including any non-grouped point on the surface.

Other methods such as area comparison (i.e. area registered by the sensor compared to the area covered exclusively by grouped points) were dismissed because of the difficulties of calculating the area enclosed by small groups of points. Estimating the area on a surface, corresponding to the points represented by a single line, that is not joined to any group, is not straightforward. Distance between consecutive lines is not always regular [17,23], neither is the distribution along the lines, which makes an accurate quantification of covered/non-covered areas impossible.

3.3. Algorithm settings

As has been explained in Section 2, some parameters are set forth. These parameters could be tuned for a specific purpose or dataset. However, we use the same parameters for the whole test dataset (i.e. Standard parameters), in order to generalize the algorithm performance validation. Table 2 contains the parameters used for the test and validation.

3.4. Validation surfaces

The performance of the algorithm was validated on 27 different surfaces. Two of them were isolated façades (F1 and F2), and the rest are clustered in three groups: Group A, with 6 different surfaces, Group B, with 9 surfaces, and Group C with 10 surfaces.

Table 2
Algorithm settings.

Polyline gap threshold	0.5 m
Douglas Peucker tolerance	5 cm
Azimuth difference tolerance	2°
Tilt angle difference tolerance	2°
Distance threshold between end-nodes	0.7 m

25 of the 27 surfaces were vertical features (i.e. as described in Section 2.4), such as walls, fences, building and house façades, large street billboards, or garage doors. However, there were two tilted surfaces (i.e. non-vertical): C5 and A3, which are roofs with a tilt angle of 20–30°. Not all the surfaces were flat. For instance, the vertical projection of the surface C1 has a 0.4 m sagittal with a chord of 7.5 m.

4. Results

The algorithm was applied to the whole dataset (more 12.7 million points) using the standard settings (see Table 2). The point cloud was transformed and reduced to 1 million nodes, performing a reduction rate of 92.1%. Table 3 summarizes the results for all the surfaces analysed.

Completeness ranges between 80% and 97%, with the only exception of Surface B4 (a fence in Group B). Surfaces B3 and B4 are two patches of a picket fence.

Correctness is, in all the surfaces, higher than 99.7%, and in 12 out of the 27 cases, there were no false positives. All the points grouped and not classified as “on the surface” are: (i) from window frameworks or downspouts, (ii) inside the outline of the surface, and (iii) sharing line with other points on the actual surface. The only exceptions to this are points from two lines from the front flat surface of one of the poles of the large billboard in Group A (i.e. Surface A5).

Figs. 10 and 11 show the original point clouds of the isolated façades F1 and F2, as well as (i) the correctly grouped lines, (ii) the non-grouped lines and (iii) the points off the surface, in different colours. Figs. 12–14 show the original point clouds and the correctly grouped lines using a different colour for each surface.

5. Discussion

5.1. Line cloud characteristics

MLS captures straight scan lines when the laser beam hits a plane or certain ruled surfaces. The algorithm proposed in this study transforms the original point cloud into a structured line cloud, where only the end-points of the straight segments are kept. This allows a substantial data reduction, if the original point cloud is not needed for further processing.

The initial point data contains four attributes (X, Y, Z and time) stored as real numbers, whereas the line cloud characterizes the segment nodes by 6 attributes (X, Y, Z, length, tilt angle and azimuth) plus an ID from the original point cloud, group code, and profile number as integers.

In a test case, the line cloud uses less than 8% of the original points, which, in the linear structure, are segment nodes. However, operatively, due to the codification attributes used for the linear structure, the amount of data required for the line cloud is 15% of the data in the original point cloud.

5.2. Algorithm settings

The algorithm settings (i.e. Standard parameters. See Section 3.3) were chosen independently on the MLS device and scan settings. It is highly unlikely for two consecutive points (from the same profile and on the same surface) to be further away than the polyline gap threshold (i.e. 0.5 m). However, due to the distance to the sensor and/or to the

Table 3
Correctness, completeness and complementary information from the validation surfaces.

Surface	Description	Area (m^2)	Points on the surface		Points off the surface	Correctness (%)	Completeness (%)
			Grouped	Ungrouped			
F1	Façade	212.3	236,175	14,081	214	99.9	94.4
F2	Façade	42.6	51,448	3769	94	99.8	93.2
A1	Lateral façade	34.3	15,986	990	27	99.8	94.2
A2	Façade	51.5	153,007	12,969	34	>99.9	92.2
A3	Roof	36.8	5859	260	0	100	95.8
A4	Garage door	4.2	7929	716	0	100	91.7
A5	Billboard	19.4	59,654	3784	30	99.9	94.0
A6	Wall	41.5	175,280	20,479	0	100	89.5
B1	Wall	12.2	6312	203	0	100	96.9
B2	Wall	9.6	16,593	2138	0	100	88.6
B3	Fence	7.5	2839	173	0	100	94.3
B4	Fence	6.0	5227	5334	0	100	49.5
B5	Façade + wall	95.6	143,344	7752	224	99.8	94.9
B6	Lateral façade	34.8	14,567	728	9	99.9	95.2
B7	Façade	45.0	72,374	7103	147	99.8	91.1
B8	Door	2.1	1648	280	0	100	85.4
B9	Lateral façade	3.6	1204	295	0	100	80.3
C1	Façade	131.6	95,019	8191	9	100	92.1
C2	Façade	44.1	37,297	2487	25	99.9	93.7
C3	Façade	29.3	9673	1752	97	99.0	84.7
C4	Lateral façade	10.7	9369	399	5	99.9	95.9
C5	Roof	24.4	9557	1129	0	100	89.4
C6	Façade	39.9	32,206	3272	75	99.8	90.8
C7	Garage doors	12.4	19,990	3204	28	99.9	86.2
C8	Façade	13.3	21,335	2413	3	>99.9	89.8
C9	Garage doors	12.0	17,342	572	0	100	96.8
C10	Lateral façade	20.4	20,199	1169	0	100.0	94.5

laser beam incidence angle, this value could be reached. In these cases, as a consequence of the lack of point density, the surfaces are too poorly described to be identified, and therefore dismissed. The distance between end-nodes is influenced by the aforementioned factors, and by some others, such as the distance between consecutive profiles or the shape of the surface edges. In this case, independently of the measurement settings, the established threshold (0.7 m) is only reached in very unfavorable conditions, and the possible links between the lines are dismissed.

The Douglas Peucker tolerance is influenced by the relative precision (relative location of the points in the same profile) and by the smoothness of the analysed surfaces. Although quantifying interaction between these two factors is not direct, the chosen value (0.05 m) is larger than the internal accuracy that can be obtained with the most commonly used MLS systems [50], and it could be changed if the smoothness of the target surfaces was different.

Many factors have an effect on the angles between lines found in consecutive profiles: sudden changes in the driving direction of the vehicle, planarity of the surfaces, or Douglas Peucker tolerance, among others with smaller influence. As for other parameters, establishing a valid value for any environment and MLS system is not straightforward. However, higher angular limits (i.e. larger than 2°) could be chosen with almost no change to the result (candidate election). Parallelism is only one of the conditions established for the lines to be considered from the same surface, and it is used as a filter before more compute-intensive steps (e.g. distance between start/end nodes). In this way, using small angular thresholds (i.e. 2° for both vertical and azimuth difference tolerances), often entails a reduction in the number of lines to be checked (i.e. calculating distances between end nodes).

5.3. Validation and comparison with other methods

All the wrongly assigned points belonged to lines where most of their points were on the actual surface. Therefore, the wrong assignment of points to a surface is due to errors during the line simplification process (see Section 2.2), and could be avoided by using a smaller Douglas Peucker (DP) tolerance (see Table 2). However, the use of a smaller tolerance may lead to oversegmentation in the line cloud, and the

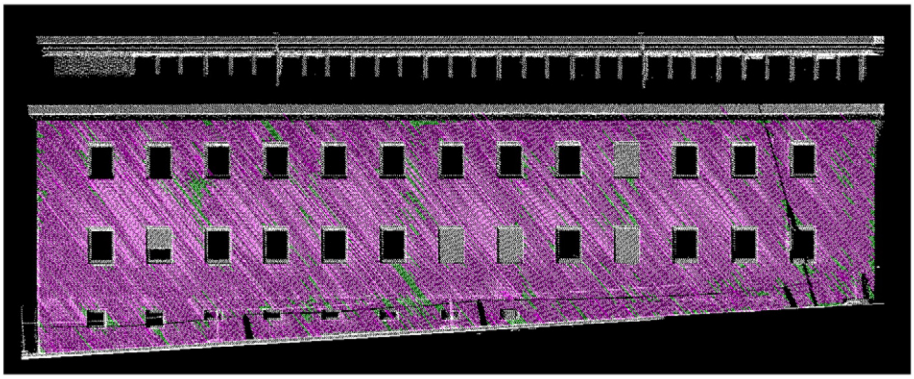


Fig. 10. Façade F1. Original point cloud and correctly grouped lines (magenta), non-grouped points and lines (green). The points grouped that do not belong to the actual surface are not visible at the scale of the figure.

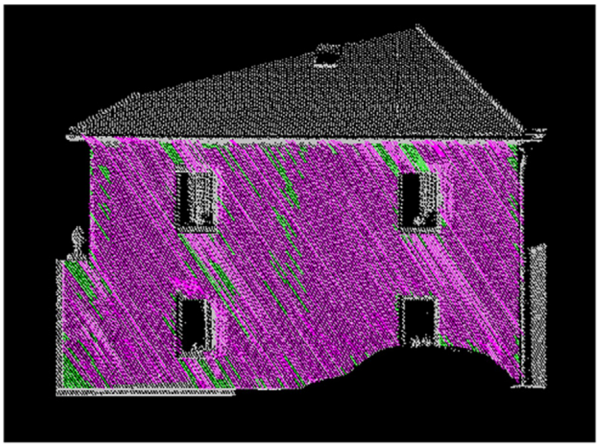


Fig. 11. Façade F2. Original point cloud and correctly grouped lines (magenta), non-grouped points and lines (green). The points grouped that do not belong to the actual surface are not visible at the scale of the figure.

impossibility of grouping lines from surfaces whose irregularities are higher than the tolerance.

Surfaces B3 and B4 are patches of a picket fence. Evenly spaced pickets cause the detection of points through the fence in Surface B4. However, there is a textile layer immediately behind Surface B3 that avoids the laser beam crossing the fence plane, thus the point and line clouds are similar to those from walls or façades.

Figs. 10 and 11 show (in green) non-grouped points and lines that belong to the surface. The presence of non-grouped lines is due to oversegmentation in the line cloud, which is produced either by the presence of (i) irregularities on the surface that are bigger than the DP tolerance, (ii) other features on the surfaces (e.g. ivy plants), or (iii) occlusions producing small blank patches near the edges of the surfaces. Nevertheless, in all the cases, the amount of correctly grouped points allows the precise interpretation and characterization of the surfaces. This fact can easily be seen in Figs. 10 and 11, observing the magenta lines, and in Fig. 12–14, comparing the original point cloud and the coloured line patches.

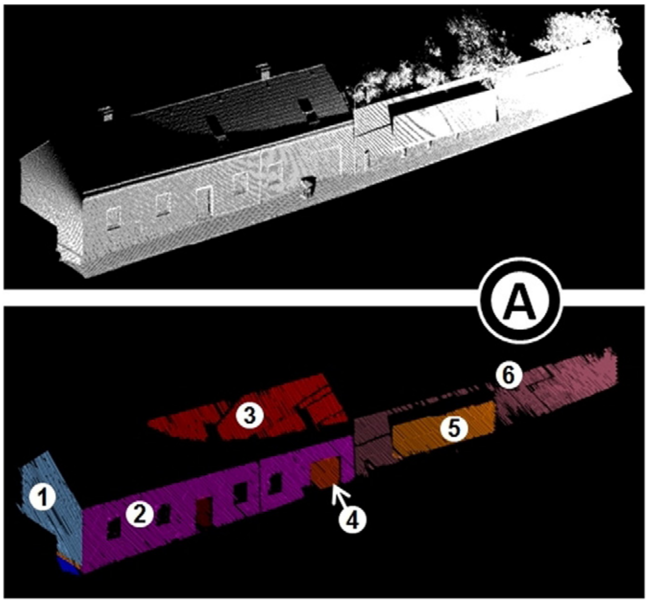


Fig. 12. Group A. Original point cloud (white dots) and line groups represented in random colours. Non-grouped lines and points grouped that do not belong to the surface, are not represented by reason of the scale.

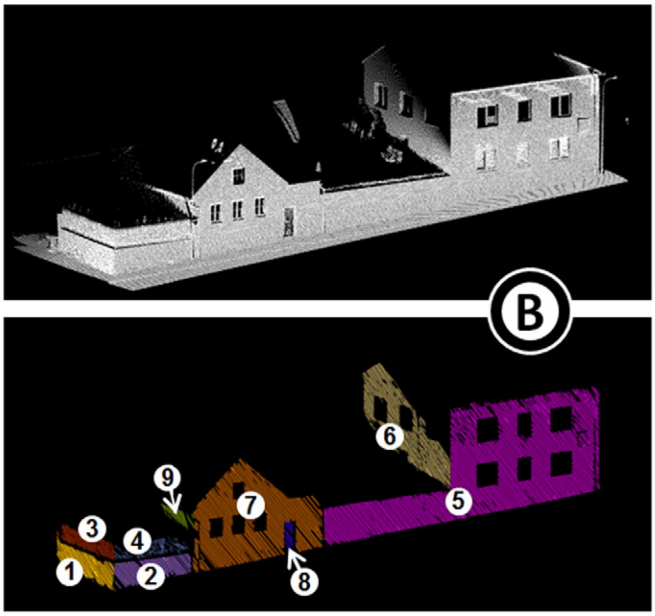


Fig. 13. Group B. Original point cloud (white dots) and line groups represented in random colours. Non-grouped lines and points grouped that do not belong to the surface are not represented by reason of the scale.

Comparing the performance of the proposed algorithm with previous studies [36–39] is not straightforward, as the targets and the way to assess their performance are different. Nevertheless, some performance indicators from previous studies (see Section 1) may be relevant when compared to the validation tests assessed in this work (see Section 3.2 and Table 3). This comparison suggests that the performance of the proposed algorithm is at least as good as the one of previous methods.

6. Conclusions

In this study, an algorithm for automatic detection of planar and quasi-planar surfaces from MLS datasets is proposed. This method transforms the original point cloud into a simple line cloud structure with much lower storage requirements. The lines are subsequently

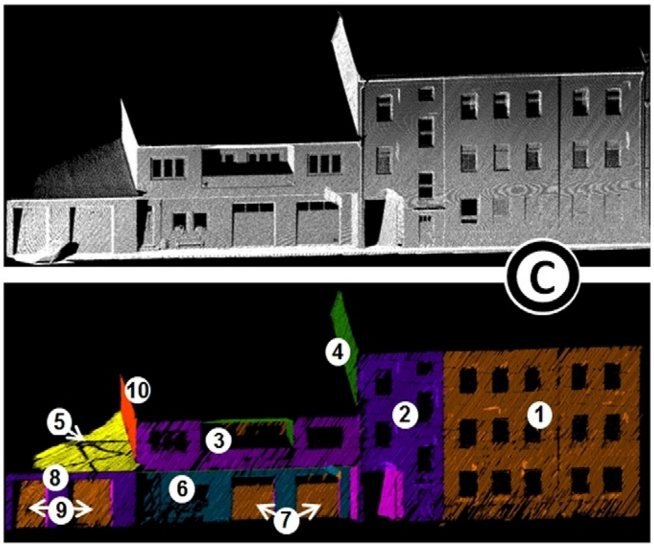


Fig. 14. Group C. Original point cloud (white dots) and line groups represented in random colours. Non-grouped lines and points grouped that do not belong to the surface are not represented by reason of the scale.

grouped, following a set of parallelism and node proximity rules, characterizing and conforming the surfaces registered by the sensor.

The method is fully automatic and does not require any training data, neither is it limited to vertical surfaces. Moreover, it is able to identify them even if not planar.

The performance of the algorithm was assessed with reference to 27 surfaces from an MLS dataset. Correctness was higher than 99% in all the surfaces, being 100% in 12 of them. Completeness averaged more than 90%, ranging from 80% to 97% in 26 out of 27 surfaces.

Finally, as regards the accomplishment of the initial objectives set for the algorithm (see Section 1) it can be concluded: (i) Planar and/or vertical surfaces are successfully detected by the algorithm In addition, non-vertical surfaces, or non-planar surfaces are equally detected. (ii) This study shows a method for identifying vertical surfaces, even if they are not planar. (iii) The algorithm is able to simplify the point cloud by transforming it into a line cloud structure, performing a significant data reduction if the original point cloud is no longer needed. (iv) The method is fully automatic, and no further data than XYZ coordinates and GNSS time records are required for the whole process.

Acknowledgments

The authors want to thank Riegl for providing the test data used in this paper. Carlos Cabo is in receipt of a “Severo Ochoa” PhD Grant funded by FICYT-Government of Principado de Asturias.

References

[1] D.F. Laefer, T. Hinks, H. Carr, L. Truong-Hong, New advances in automated urban modelling from airborne laser scanning data, Recent Patents Eng. 5 (3) (2011) 196–208.
[2] Y. Lin, A. Jaakkola, J. Hyyppä, H. Kaartinen, From TLS to VLS: biomass estimation at individual tree level, Remote Sens. 2 (2010) 1864–1879.
[3] M.A. Lefsky, W.B. Cohen, G.G. Parker, D.J. Harding, LiDAR remote sensing for ecosystem studies, Bioscience 52 (2002) 19–30.
[4] M. Vaaja, J. Hyyppa, A. Kukko, H. Kaartinen, H. Hyyppa, P. Alho, Mapping topography changes and elevation accuracies using a mobile laser scanner, Remote Sens. 3 (2011) 587–600.
[5] N. Haala, C. Brenner, Extraction of buildings and trees in urban environments, ISPRS J. Photogramm. Remote Sens. 54 (2–3) (1999) 130–137.
[6] H.G. Maas, G. Vosselman, Two algorithms for extracting building models from raw laser altimetry data, ISPRS J. Photogramm. Remote Sens. 54 (2–3) (1999) 153–163.
[7] T. Hilker, N.C. Coops, G.J. Newnham, M. van Leeuwen, M.A. Wulder, J. Stewart, D.S. Culvenor, Comparison of terrestrial and airborne LiDAR in describing stand structure of a thinned lodgepole pine forest, J. For. 110 (2012) 97–104.
[8] H. Zhao, R. Shibasaki, A vehicle-borne urban 3-D acquisition system using single-row laser range scanners, IEEE Trans. Syst. Man Cybern. 33 (2003) 658–666.
[9] A. Kukko, H. Kaartinen, J. Hyyppä, Multiplatform mobile laser scanning: usability and performance, Sensors 12 (2012) 11712–11733.
[10] H.J. Yoo, F. Goulette, J. Senpauroca, G. Lepère, Analysis and improvement of laser terrestrial mobile mapping systems configurations, Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. 38 (5) (2010) 633–638.
[11] L. Graham, Mobile mapping systems overview, Photogramm. Eng. Remote Sens. 76 (3) (2010) 222–228.
[12] G. Petrie, An introduction to the technology mobile mapping systems, Geoinformatics 13 (2010) 32–43.
[13] M. Vaaja, A. Kukko, H. Kaartinen, M. Kurkela, E. Kasvi, C. Flener, H. Hyyppä, J. Hyyppä, J. Järvelä, P. Alho, Data processing and quality evaluation of a boat-based mobile laser scanning system, Sensors 13 (9) (2013) 12497–12515.
[14] H. Kaartinen, J. Hyyppä, A. Kukko, A. Jaakkola, H. Hyyppä, Benchmarking the performance of mobile laser scanning systems using a permanent test field, Sensors 12 (2012) 12814–12835.
[15] J. Wang, F.X. Jin, Precision estimation of mobile laser scanning system, Surv. Rev. 42 (270–278) (2010).
[16] I. Puente, H. González-Jorge, P. Arias, J. Armesto, Land-based Mobile Laser Scanning Systems: a review, Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. 38 (5-W12) (2012) 163–168.
[17] C. Cahalane, T. McCarthy, C. McElhinney, Mobile mapping system performance—an initial investigation into the effect of vehicle speed on laser scan lines, Proc. Remote Sensing & Photogrammetry Society Annual Conference. From the sea-bed to the cloudtops, Cork, Ireland, Sep. 1–3, 2010 (8 pp.).
[18] K. Williams, M.J. Olsen, G.V. Roe, C. Glennie, Synthesis of transportation applications of mobile LiDAR, Remote Sens. 5 (9) (2013) 4652–4692.
[19] X. Liang, J. Hyyppa, A. Kukko, H. Kaartinen, A. Jaakkola, X. Yu, The use of a mobile laser scanning system for mapping large forest plots, IEEE Geosci. Remote Sens. Lett. 11 (9) (2014) 1504–1508.
[20] M. Holopainen, M. Vastaranta, V. Kankare, J. Hyyppa, L. Xinlian, P. Litkey, Y. Xiaowei, H. Kaartinen, A. Kukko, S. Kaasalainen, A. Jaakkola, H. Hyyppa, M. Vaaja, The use of ALS, TLS and VLS measurements in mapping and monitoring urban trees, Proceedings of the Urban Remote Sensing Event (JURSE), Munich, Germany, Apr. 11–13 2011, pp. 29–32.
[21] X. Liang, A. Kukko, H. Kaartinen, J. Hyyppä, X. Yu, A. Jaakkola, Y. Wang, Possibilities of a personal laser scanning system for forest mapping and ecosystem services, Sensors 1 (2014) 1228–1248.
[22] C. Cabo, C. Ordóñez, S. García-Cortés, J. Martínez, An algorithm for automatic detection of pole-like street furniture objects from Mobile Laser Scanners point clouds, ISPRS J. Photogramm. Remote Sens. 87 (2014) 47–56.
[23] M. Lehtomaki, A. Jaakkola, J. Hyyppa, A. Kukko, H. Kaartinen, Detection of vertical pole-like objects in a road environment using vehicle-based laser scanning data, Remote Sens. 2 (2010) 641–664.
[24] D. Barber, J. Mills, S. Smith-Voysey, Geometric validation of a ground-based mobile laser scanning system, ISPRS J. Photogramm. Remote Sens. 63 (1) (2008) 128–141.
[25] H.G. Maas, Methods for measuring height and planimetry discrepancies in airborne laser scanner data, Photogramm. Eng. Remote Sens. 68 (9) (2002) 933–940.
[26] S. Pu, M. Rutzinger, G. Vosselman, S.O. Elberink, Recognizing basic structures from mobile laser scanning data for road inventory studies, ISPRS J. Photogramm. Remote Sens. 66 (6) (2011) S28–S39.
[27] Z.Z. Kang, L.Q. Zhang, B.Q. Wang, Z. Li, F.M. Jia, An optimized BaySAC algorithm for efficient fitting of primitives in point clouds, IEEE Geosci. Remote Sens. Lett. 11 (6) (2014) 1096–1100.
[28] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, G. Taubin, The ball-pivoting algorithm for surface reconstruction, IEEE Trans. Vis. Comput. Graph. 5 (4) (1999) 349–359.
[29] K. Demarsin, D. Vanderstraeten, T. Volodine, D. Roose, Detection of closed sharp edges in point clouds using normal estimation and graph theory, Comput. Aided Des. 39 (4) (2007) 276–283.
[30] R. Schnabel, R. Wahl, R. Klein, Efficient RANSAC for point-cloud shape detection, Comput. Graph. Forum 26 (2) (2007) 214–226.
[31] S. Oude Elberink, G. Vosselman, Quality analysis on 3D building models reconstructed from airborne laser scanning data, ISPRS J. Photogramm. Remote Sens. 66 (2) (2011) 157–165.
[32] S. Filin, N. Pfeifer, Segmentation of airborne laser scanning data using a slope adaptive neighborhood, ISPRS J. Photogramm. Remote Sens. 60 (2) (2006) 71–80.
[33] Y. Yu, J. Li, H. Guan, C. Wang, J. Yu, Automated detection of road manhole and sewer well covers from mobile LiDAR point clouds, IEEE Geosci. Remote Sens. Lett. 11 (9) (2014) 1549–1553.
[34] A. Jaakkola, J. Hyyppä, H. Hyyppä, A. Kukko, Retrieval algorithms for road surface modelling using laser-based mobile mapping, Sensors 8 (2008) 5238–5249.
[35] G. Vosselman, B.G.H. Gorte, G. Sithole, T. Rabbani, Recognising structure in laser scanner point clouds, Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. 36 (8/W2) (2004) 33–38.
[36] A. Jochem, B. Höfle, M. Rutzinger, Extraction of vertical walls from mobile laser scanning data for solar potential assessment, Remote Sens. 3 (4) (2011) 650–667.
[37] H. Fan, W. Yao, L. Tang, Identifying man-made objects along urban road corridors from mobile LiDAR data, IEEE Geosci. Remote Sens. Lett. 11 (5) (2014) 950–954.
[38] B. Yang, Z. Dong, A shape-based segmentation method for mobile laser scanning point clouds, ISPRS J. Photogramm. Remote Sens. 81 (2013) 19–30.
[39] Z. Lari, A. Habib, An adaptive approach for the segmentation and extraction of planar and linear/cylindrical features from laser scanning data, ISPRS J. Photogramm. Remote Sens. 93 (2014) 192–212.
[40] G. Sithole, G. Vosselman, Automatic Structure Detection in a Point Cloud of an Urban Landscape, Proceedings of the Second GRSS/ISPRS Joint Workshop on Remote Sensing and Data. Fusion over Urban Areas, URBAN2003, Berlin, Germany, May 22–23 2003, pp. 67–71.
[41] T. Pavlidis, S.L. Horowitz, Segmentation of plane curves, IEEE Trans. Comput. 8 (C-23) (1974) 860–870.
[42] B. Howarth, M. Whitty, J. Katupitiya, J. Guivant, Extraction and grouping of surface features for 3D mapping, Proc. of the 2011 Australasian Conference on Robotics and Automation, Melbourne, Australia, Dec. 7–9, 2011 (6 pp.).
[43] Y. Lin, J. Hyyppa, K-segments-based geometric modeling of VLS scan lines, IEEE Geosci. Remote Sens. Lett. 8 (1) (2011) 93–97.
[44] J. Verbeek, N. Vlassis, B. Kröse, A k-segments algorithm for finding principal curves, Pattern Recogn. Lett. 23 (8) (2002) 1009–1017.
[45] X.Y. Jiang, H. Bunke, Fast Segmentation of range images into planar regions by scan line grouping, Mach. Vis. Appl. 7 (2) (1994) 115–122.
[46] D.H. Douglas, T.K. Peucker, Algorithms for the reduction of the number of points required to represent a digitized line or its caricature, Can. Cartographer 10 (2) (1973) 112–122.
[47] J. Hershberger, J. Snoeyink, Speeding up the Douglas–Peucker line-simplification algorithm, Proc. of the Fifth International Symposium on Spatial Data Handling, Charleston, USA, vol. 1 1992, pp. 134–143.
[48] K. Ebisch, A correction to the Douglas–Peucker line generalization algorithm, Comput. Geosci. 28 (8) (2002) 995–997.
[49] Riegl, http://www.riegl.com/ (accessed 15 January, 2015).
[50] I. Puente, H. González-Jorge, J. Martínez-Sánchez, P. Arias, Review of mobile mapping and surveying technologies, Measurement 46 (7) (2013) 2127–2145.

Capítulo IV

Detección de Superficie Asfaltada

[An algorithm for automatic road asphalt edge delineation from Mobile Laser Scanner data using the Line Clouds concept]

remote sensing

MDPI

Article

An Algorithm for Automatic Road Asphalt Edge Delineation from Mobile Laser Scanner Data Using the Line Clouds Concept

Carlos Cabo [1,*], Antero Kukko [2], Silverio García-Cortés [1], Harri Kaartinen [2], Juha Hyyppä [2] and Celestino Ordoñez [1]

[1] Department of Mining Exploitation, University of Oviedo, 33004 Oviedo, Spain; sgcortes@gmail.com (S.G.C.); cgalan.uniovi@gmail.com (C.O.)

[2] Finnish Geospatial Research Institute FGI, P.O. Box 15, FI-02431 Masala, Finland; antero.kukko@nls.fi (A.K.); harri.kaartinen@nls.fi (H.K.); juha.hyyppa@nls.fi (J.H.)

* Correspondence: carloscabo.uniovi@gmail.com

Academic Editors: Bailang Yu, Gonzalo Pajares, Xiaofeng Li and Prasad S. Thenkabail
Received: 10 July 2016; Accepted: 30 August 2016; Published: 7 September 2016

Abstract: Accurate road asphalt extent delineation is needed for road and street planning, road maintenance, and road safety assessment. In this article, a new approach for automatic roadside delineation is developed based on the line clouds concept. The method relies on line cloud grouping from point cloud laser data. Using geometric criteria, the initial 3D LiDAR point data is structured in lines covering the road surface. These lines are then grouped according to a set of quasi-planar restriction rules. Road asphalt edge limits are extracted from the end points of lines belonging to these groups. Finally a two-stage smoothing procedure is applied to correct for edge occlusions and other anomalies. The method was tested on a 2.1 km stretch of road, and the results were checked using a RTK-GNSS measured dataset as ground truth. Correctness and completeness were 99% and 97%, respectively.

Keywords: mobile laser scanning; 3D road extraction; scanning lines; road edges; smoothing window; road management; line clouds

1. Introduction

Road feature control plays a key role in traffic safety and infrastructure maintenance. Many departments and government agencies are introducing the mobile LiDAR system (mobile laser scanner; MLS) as a new and cost effective collection method to investigate and monitor the state of road infrastructure [1]. The main goal is to reduce the operational costs associated with a manual human inspection, while improving the quality and availability of the services.

The mobile laser scanning concept was introduced in the 1980s and early 1990s. One of the very first systems was developed by the Centre for Mapping at Ohio State University [2,3]. Since then, a lot of prototypes and commercial systems have been introduced, and the number of different systems, additional sensors, and configurations have exponentially increased. The basic components though, still remain unchanged: (i) mapping sensors (active and/or passive, visual or laser based); (ii) navigation and positioning sensors IMU (inertial measurement unit), GNSS (global navigation satellite system) devices; (iii) and a control unit for integration of data from each sub-system. All of these are firmly integrated on a van or another type of vehicle, but there is a clear trend to increase flexibility combining LiDAR with many different additional sensors. A fairly recent compilation of available MLS systems can be found in [4].

The RoamerR2 MLS system used for road data collection in this study has its origin as a research unit developed at the Finnish Geospatial Research Institute FGI but, in practice, is a highly operational unit for different surveying tasks [5]. RoamerR2 is a built on multi-platform idea, and it has been used in a multitude of 3D mapping tasks, from roads and urban mapping [6,7] to geomorphological studies [8] and precision forestry [9].

An MLS produces huge amounts of 3D points grouped in clouds, and register roadside information at very high densities. These data are very suitable for road information and inspection tasks, noise spread modeling, and other applications. Additionally, segmentation of 3D point clouds in urban scenarios allows coarse modeling of cities through the extraction of urban elements, like traffic lamps, trees, building facades, and pavement. Different filtering, segmentation, and classification approaches have been proposed from the very first days of airborne LiDAR and terrestrial laser systems [10,11]. Extraction of fundamental structures, like roads, are considered as necessary elements for digital ground models [12], but the 3D point cloud interpretation poses a major challenge in terms of calculation time, efficiency, and feature recognition [13], and only a few publications have dealt with delineation of road borders or curb detection exclusively from MLS cloud data.

Mainly, three different types of approaches arise from the published work on MLS point cloud processing: (i) methods based on scanning-line segmentation [14,15]; (ii) methods based on specific feature extraction like planar patches, trees, or poles [16–20]; and (iii) image segmentation approaches [21,22].

Road edge detection from MLS has been addressed by [6] using image processing methods over the intensity height image, combining it with existing GIS maps to estimate road locations. Image-processing techniques to detect curbs and street boundaries from MLS data after rasterizing the point cloud were applied in [23]. Image-processing with morphological operators was also conducted in [22] to detect street boundaries, amongst other features, in urban environments from MLS. Previously, other authors [24] had proposed a method to detect lanes from images considering a geometric model adaptable to different shapes.

Other algorithms are not based on image-processing. For instance, El-Halawany et al. [25] describes an algorithm based on computing normal vectors for detecting road curbs from MLS data. Those vectors are calculated by analyzing the local neighborhood of every point using k-D trees. In [26], given start and end points are used as initial conditions for an edge extraction algorithm based on deformable models. Smooth street boundaries are extracted using existing 2D vector maps that improve the modeling, but make the model sensitive to the accuracy of the GIS maps.

Another way to tackle segmentation from MLS data is to process scanning lines instead of unordered point clouds, as in [27], where the concept of line clouds was introduced and used to extract building façades. Curb detection was conducted in [15] using scanning lines and curb patterns. Zhao and Sibasaki [28] developed an algorithm where points in the scan lines were segmented into linear patches according to height variance, and then grouped. However, no in-depth analysis of the utility of their algorithm to detect the road surface was performed; instead, the study is focused on its capacity to discriminate between different objects. A two-step RANSAC algorithm was applied in [29] to each scan line to detect road edges. Then, 3D mathematical models were used to interpolate edge points. A different approach, using an adaptive alpha shape algorithm has been recently proposed [30]. Some other studies [6,31,32] take advantage of the intensity signal of the laser scanners to detect boundaries, although this procedure is not suitable when road markings do not exist, or when they are damaged.

All of the aforementioned algorithms for automatic detection or determination of road edges, surfaces, or shapes need the existence of road markings, curbs, and/or use additional data to the MLS geometric data (e.g., LiDAR intensity, or images). The main objective of this study is, therefore, to develop an automatic method for accurately delineating the road asphalt edge exclusively from geometric MLS data. The new algorithm will also be able to deal with adverse situations, such as (i) irregularities and discontinuities on the road surface (e.g., cracks, or different asphalt patches); and (ii) shallow or not very neat road edges.

In this article, Section 2 will cover the proposed methodology based in previous point cloud structuration as line groups, following the scanning-line approach. Section 3 compares the obtained results with real road edge data, measured using GNSS RTK (real-time kinematic) positioning. Finally conclusions will be addressed on Section 4.

2. Methodology

The proposed algorithm assumes a geometrical discontinuity (or break) at the edge of the road, and that this break reaches certain size. This discontinuity is due to the presence of a curb, wall, or simply the end of the asphalt at the edge of the road or street. However, at certain points (i.e., sections), the aforementioned discontinuity does not exist, or is not detectable from a MLS point cloud (even using visual inspection). The algorithm is designed towards a correct and continuous road edge detection even if, locally (i.e., at some sections), the edge is not detectable using the MLS source dataset.

The proposed method is based on (i) the transformation of the original point cloud into a line cloud, developed and reported in [27]; (ii) line grouping; (iii) the generation of contour lines from the line groups; and (iv) a two-stage line smoothing.

2.1. Data Requirements

From the datasets, (i) XYZ coordinates; (ii) time stamp; and (iii) a trajectory file are needed. However, if the points are time-wise sorted, a list containing the first point from each sweep can be used instead of the time stamp field.

The trajectory file is used for straightening the MLS path, which is part of one of the smoothing processes carried out by the algorithm. In regards to the trajectory data, only a list of coordinates of the position of the MLS along the path, linked to its time stamp, is required.

2.2. Line Clouds

Using the time stamp field and the scanning rate of the sensors, points are time-wise sorted and labeled with their sweep. As it is extensively explained in [27], in a theoretical situation, where the vehicle carrying the MLS system was driven along a circular-section tunnel, the result of joining all of the points of such dataset would generate a continuous helical line. In the case of an ideal corridor street, a succession of parallel U-shaped polylines would be generated. In doing this, a single polyline would be created by joining all of the points in a dataset. The algorithm splits this polyline following two criteria: (i) the polyline is divided in sweeps; and (ii) within each sweep, each polyline is split when the gap between two consecutive points exceeds a certain length. Figure 1 shows an example of an initial polyline split into several polylines.

Each of the resulting polylines may have points on several different features, or parts of them. For instance, the nodes from the blue line in the section of Figure 1B, are points measured on the road, on the pavement and on a wall, whereas the yellow line only represents a car.

At a certain scale, many of the nodes from the polylines might be redundant, as they could be represented by the end points of the straight segments that each polyline contains. This is more likely to happen when the laser beam finds planar or ruled surfaces [27]. In this way, all of the points from each polyline can be replaced by a set of straight lines, which are within a preset distance (tolerance) from the group of points that they replace. This simplification is carried out by using a three-dimensional version of the Douglas Peucker algorithm [33]. The process is tackled by checking the distance from all the points that each polyline contains, to the straight line that results from joining its first and last points. If there is any point further than the tolerance, the polyline is split at the furthest point, creating new polylines. The new polylines are analyzed using the same schema, and the subdivision carries on until there is no point further than the set tolerance, so the new polylines can be considered as straight lines. For instance, Figure 2 shows two cases of road profiles, the point cloud on these surfaces, and the straight segments.

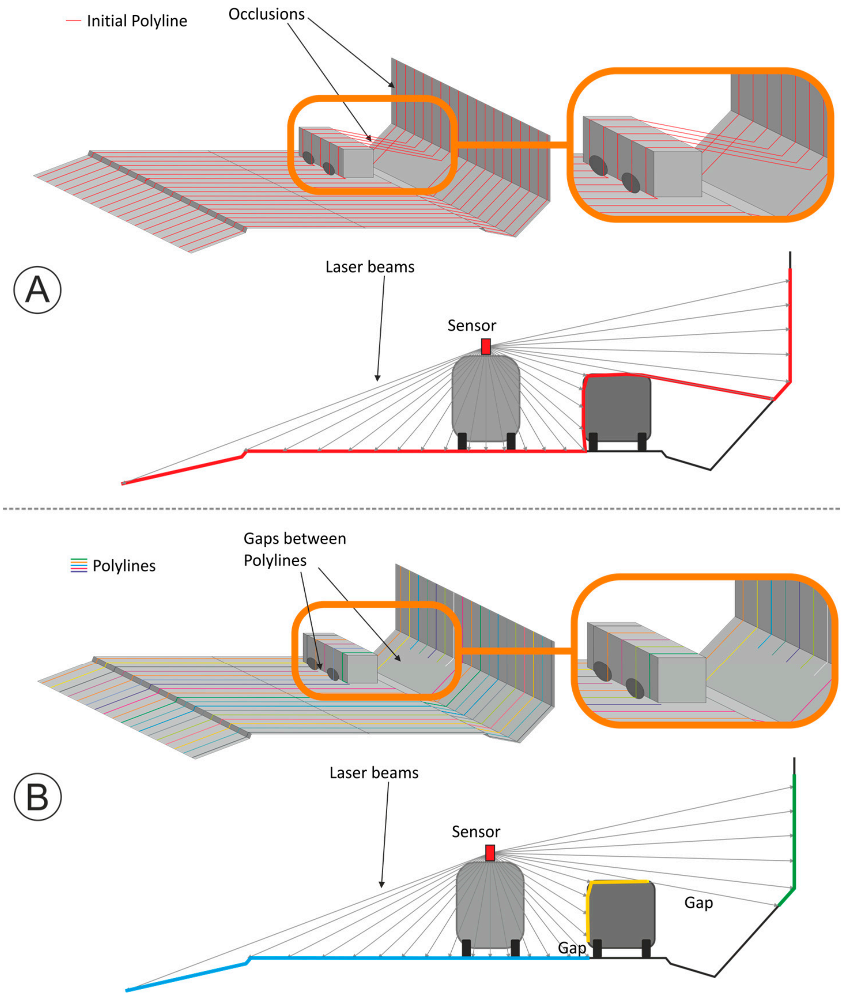


Figure 1. Result of MLS on a street corridor section. (**A**) perspective view and a section perpendicular to the trajectory of a single polyline as a result of joining all the consecutive points; (**B**) perspective view and a section perpendicular to the trajectory of randomly-colored polylines obtained from splitting the initial polyline at discontinuities or gaps.

In Figure 2A, a three-stage line simplification is shown. From polyline(n), a preliminary straight line (ST(n,1)) is created. There is, at least, one point that surpasses the tolerance set (i.e., maximum distance to ST(n,1)), thus, the line is split at the furthest point to ST(n,1). Two new lines are created: ST(n,1,1) and ST(n,1,2). All the points between the end-nodes of ST(n,1,1) are closer than the tolerance, therefore, ST(n,1,1) is not further divided. In ST(n,1,2) there is, at least, one point further than the tolerance, and it is divided into the definitive ST(n,1,2,1) and ST(n,1,2,2). Finally, the initial polyline(n) is divided and simplified into three consecutive straight segments, which are defined by only four end-nodes.

In both cases, in Figure 2B,C, the geometrical discontinuity at the road edge generates a line endpoint (endpoint shared by orange and green lines in both cases), which, in subsequent stages, will allow the identification of the road border.

The line simplification can be carried out using either all the points in the initial point cloud, or only a selection of points, that can be performed by excluding points using the view angle, and/or the relative height from the sensor (i.e., being that the road edge is the target of the method, it is not necessary to use points higher than 1 or 2 m from the road where the vehicle is traveling on).

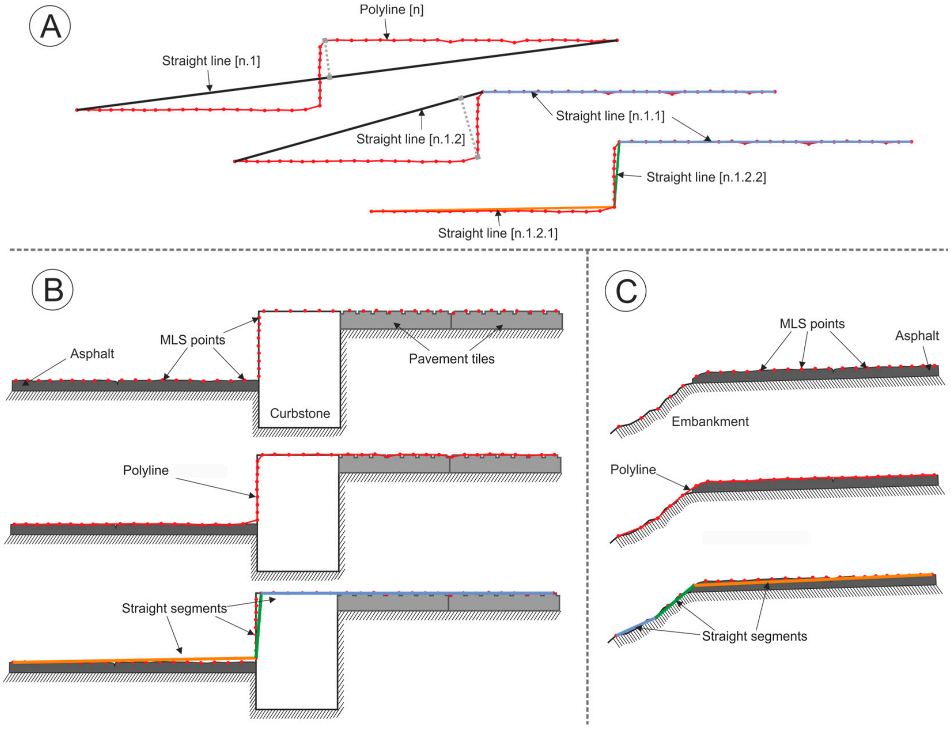
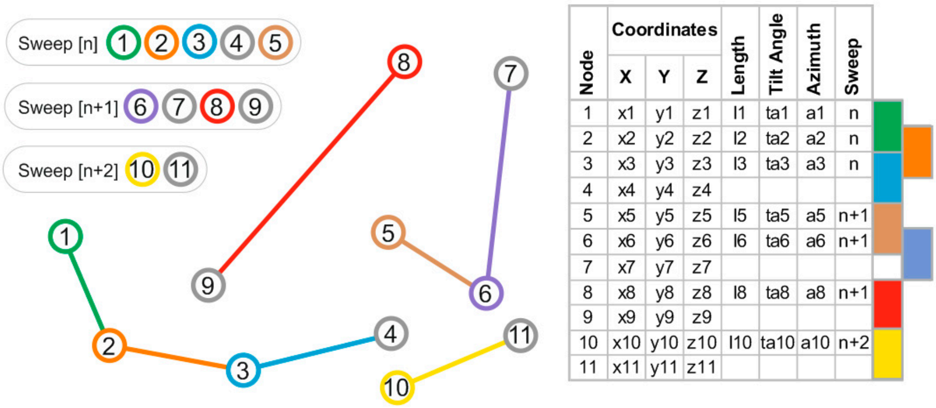


Figure 2. (**A**) transformation of a polyline into a set of straight lines. (**B**) and (**C**) Point cloud simplification on two different road edge profiles; (**B**) road edge on an embankment; (**C**) road edge in an urban environment (with curb/pavement).

Once the point cloud is transformed into a line cloud, three parameters are calculated and stored for each line: (i) length; (ii) tilt angle; and (iii) azimuth. After that, based on the tilt angle, those lines that clearly do not belong to the road surface are excluded, as it is assumed that the road surface does not contain heavily tilted lines (e.g., lines with a tilt angle bigger than 15°).

Finally, the point cloud is stored by listing the coordinates of all its nodes, and the attributes of each line (i.e., length, tilt angle, azimuth, and profile) are linked to the first node of each line. Figure 3 shows an example of line cloud storage. Note that the nodes that are not the first node from any line do not store any other attribute than XYZ coordinates.



Node	Coordinates			Length	Tilt Angle	Azimuth	Sweep
	X	Y	Z				
1	x1	y1	z1	l1	ta1	a1	n
2	x2	y2	z2	l2	ta2	a2	n
3	x3	y3	z3	l3	ta3	a3	n
4	x4	y4	z4				
5	x5	y5	z5	l5	ta5	a5	n+1
6	x6	y6	z6	l6	ta6	a6	n+1
7	x7	y7	z7				
8	x8	y8	z8	l8	ta8	a8	n+1
9	x9	y9	z9				
10	x10	y10	z10	l10	ta10	a10	n+2
11	x11	y11	z11				

Figure 3. Example of line cloud data storage of lines from three different sweeps: (n), (n+1) and (n+2). Points 1–11 are hypothetical consecutive points. The points are split into four different polylines: (1–2–3–4), (5–6–7), (8–9) and (10–11). The polylines are divided into seven straight lines, which are stored as shown in the table.

2.3. Line Grouping

The process of line grouping consists of the search and labeling of lines that belong to the same planar or quasi-planar surface. Following a similar procedure to the one described in [31], the lines are grouped based on their proximity and similarities. It is assumed that the lines from consecutive sweeps on planar or some ruled surfaces are almost parallel and, frequently, their initial or end nodes are close to each other.

Only the lines that reach a certain length are used in this process, avoiding short lines from a small group of original points. Three thresholds are set in order to analyze the parallelism and node proximity between the lines. Two of them are angular and establish the maximum difference in tilt angle and azimuth between two lines to be considered parallel. The third one refers to the maximum distance between equivalent nodes from different lines to be considered as possibly belonging to the same surface.

Initially, the longest line in the line cloud is used as a seed line (i.e., initial seed line), and all the lines from the following sweep that reach a certain length are preselected. Then, the lines whose tilt angle or azimuth differs from those of the seed line more than the threshold are removed from the preselection. After that, the distances from the first node of the seed line to the first nodes of the preselected lines are calculated and, equally, the distances between final nodes are checked. If at least one of the nodes from a preselected line is closer to its equivalent in the seed line, the line is selected and added to the same group as the seed line.

There are eight (8) possible cases of positive selection of a preselected line regarding the distances between equivalent nodes (see Figure 4):

- Cases 1 and 2: there is only one node closer than the threshold to its equivalent node in the seed line. This is a sufficient condition for the line to be selected.
- Case 3 and 4: there is a node within the search distance from each end of the seed line. In Case 3, both nodes are from the same preselected line, whereas, in Case 4, the nodes are from different lines. In Case 4, the two lines are selected.

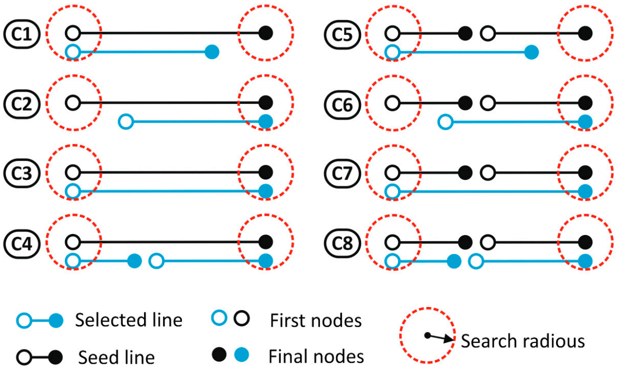


Figure 4. Possible cases (C1–C8) for positive line/s selection from seed line/s. In Cases (C1–C4), the seed line is a single line. In Cases (C5–C8), the selection is performed from double seed lines.

Once a line is selected, it becomes the new seed line, and the lines from the following sweep are compared to it. Lines from consecutive sweeps on the same planar or quasi-planar surface are usually almost parallel. However, lines on the same surface, but distant from the initial seed line, may not be parallel. This is due to changes in sensor orientation in relation to the surface and/or non-planarity of the surfaces that lead to gradual change in the azimuth and tilt angle of the lines from different sweeps on the same surface. In addition to this, the lines are often split by the presence of small artefacts on the surfaces, such as cracks, or any other local feature whose size is larger than the Douglas Peucker threshold. Even in the cases where this phenomenon only affects a small number of sweeps (e.g., the gap produced by a small hole in the asphalt), it could lead to a stop of the line grouping, as one of the nodes is placed in the middle of the surface. To avoid this, the selection of double lines is allowed (see Case 4 in Figure 4). When that happens, the two lines are used as a single seed line, and only the first node of the first line and the end node of the second line are used in the subsequent search and grouping step (i.e., within the following sweep). Cases 5 to 8 in Figure 4 are equivalent to Cases 1 to 4, although the selection is carried out from a double seed line.

The grouping process is carried out from the initial seed line in both directions. To start, beginning with the initial seed line in sweep (n), lines are sought for in the following sweep (i.e., sweep (n+1)). If a line (single or double) is selected, it becomes the new seed line, and the lines in sweep (n+2) are checked. The process continues until there is no line that fulfils the parallelism and node-proximity criteria. After that, the process is carried out in the opposite direction from the initial seed line (i.e., starting with sweep (n−1)). When the search finishes in both directions, another initial seed line is designed by choosing the longest line that does not yet belong to any group. When a line that belongs to a group is selected, the rest of the lines within the same group are joined to the new group. The process continues until (i) all the lines are assigned to a group; (ii) a certain amount of groups are created; or (iii) the following initial seed line does not reach a pre-set length.

Ideally, after the line grouping, all of the lines from the road surface should belong to the same group, but it is frequent that, because of the presence of small artefacts, cracks, irregularities on the asphalt, or different cross-slopes for each lane, the lines on the road surface are split in different groups (as shown in Figure 5B. These initial groups are joined following two criteria: (i) all of the groups whose lines are crossed by the projection of the trajectory (and are underneath it), and (ii) all of the groups that share a certain amount of nodes with any group selected using the previous criterion. Taking into account the initial assumption that there is a vertical discontinuity at the asphalt edge, only the groups from the road surface share nodes, as the lines that surpass a certain slope are not used for the line grouping.

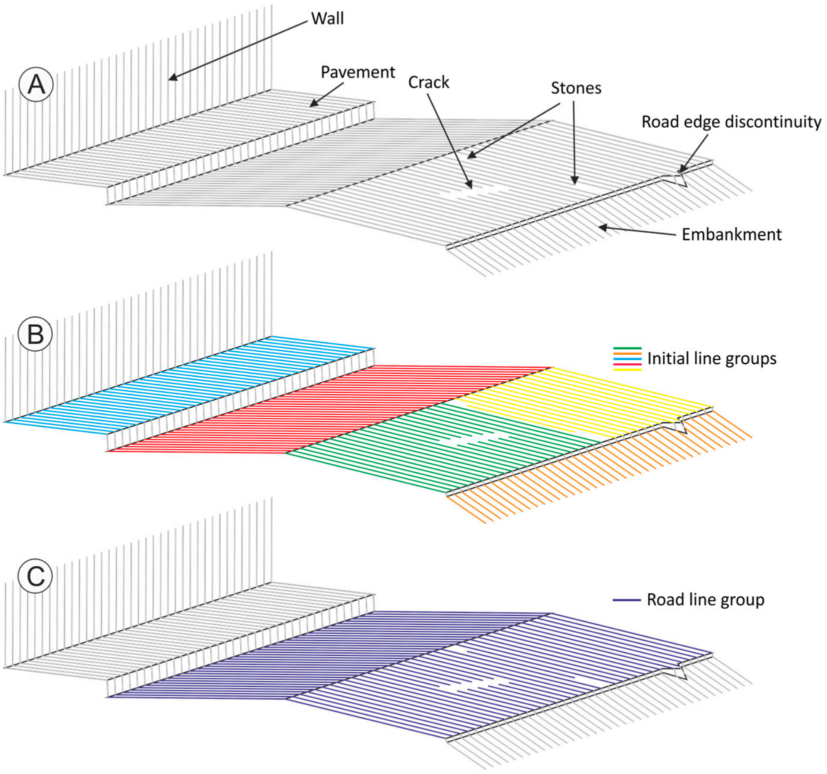


Figure 5. Line grouping on a road section. (**A**) Elements that produce disaggregation among the road surface groups; (**B**) Initial line groups; (**C**) Final group.

2.4. Road Edge Polylines

The road edge is initially extracted from the final group of lines. Initial polylines are created on both sides of the road by joining the extreme points of the line group at each sweep (see Figure 6, initial edge line).

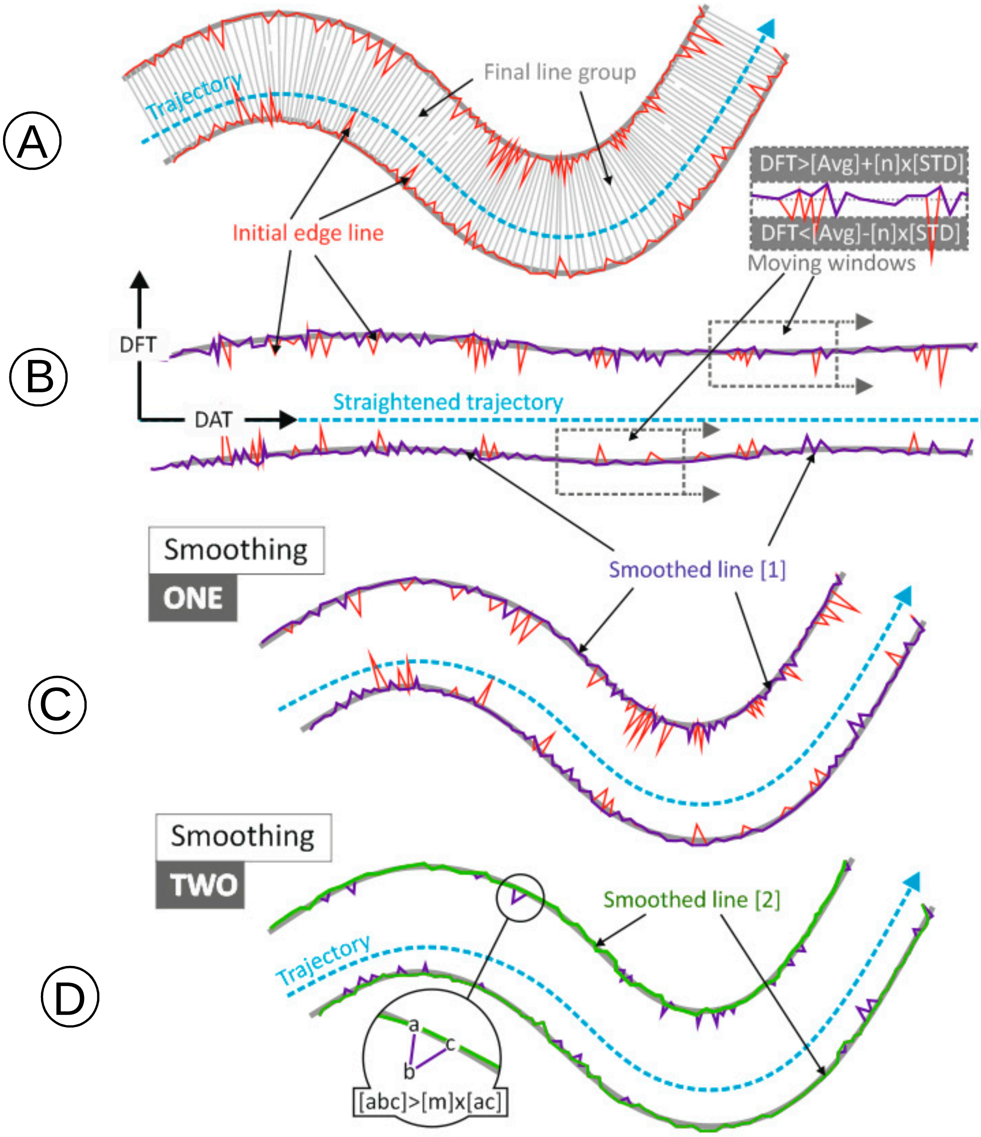


Figure 6. Example of line road edge polyline smoothing on a hypothetical road section. (**A**) final line group and initial edge polyline; (**B**) straightened trajectory, initial edge line transformed into (DFT, DAT) coordinates, and moving window; (**C**) result of the Smoothing one; (**D**) result of the Smoothing two.

It is usual that, at some sweeps, the extreme nodes do not match with the road edge, giving place to an uneven road edge polyline. This is often produced by the presence of small objects on the hard shoulder, such as some gravel stones, or any other debris whose size surpasses the Douglas Peucker tolerance.

Two different smoothing processes are carried out in order to avoid these irregularities. As most of the nodes are correct, the smoothing processes aim at detecting and eliminating isolated incorrect nodes, rather than averaging the position of both, correct and wrong, nodes of the initial polyline.

2.4.1. Smoothing 1: Moving Window along Straightened Trajectory

For the first smoothing process, the coordinates of the nodes from the initial road edge polylines are transformed into a system relative to the trajectory. This operation is equivalent to straightening the trajectory, and uses for each node (i) the distance from the trajectory (DFT); and (ii) the distance along the trajectory (DAT); see Figure 6.

Once the polylines are transformed, two moving windows that scroll along the entirety of the straightened trajectory are set forth using two parameters: (i) the window width; and (ii) the step that the window scrolls at each forward move. The units of these two parameters can be either number of sweeps, or distance.

The method analyzes the nodes that are within the limits of each position of the moving window. The nodes which are further than a preset number of times ((n) in Figure 6) the standard deviation from the average distance to the trajectory are detected and labeled. The windows are scrolled along the straightened trajectory and, at the end of the process, the exclusion of a specific node is determined by the number of times it was voted as an outlier by the moving window.

2.4.2. Smoothing 2

Even though the first smoothing is able to detect most of the outlier nodes, it is also frequent that some irregularities persist along the edge polylines. For instance, in some areas with a large proportion of outliers, the closest ones to the actual road edge can remain undetected, as the first smoothing process is average/deviation-dependent. To detect and remove these remaining outliers, the distances along the polyline from each node to its closest neighbors are checked and compared with the distances in a straight line, see Figure 6. Despite the road edges might be very intricate/irregular, these irregularities are usually different to, for instance, the peaks produced by isolated wrong nodes on the hard shoulder (as the example shown in Figure 6D). In the figure, the distance from node (a) to node (b) is compared with the distance along the path (abc) (i.e., along the polyline resulting from the first smoothing). If (abc) is more than [m] times bigger than (ac), the node (b) is removed.

3. Test Case

The algorithm performance and accuracy were tested on two MLS datasets from RoamerR2. The datasets contain points from a 2.1 km road section of the Ring III road that gives access to the Finnish Geospatial Research Institute (FGI), in Masala, Southern Finland. It is a non-straight interurban stretch of road, whose asphalt/road edge is not always very clear, as in some sections it is leveled or almost leveled with compact terrain. Cracks and different asphalt patches occur frequently on the road surface.

3.1. MLS System and Point Cloud Dataset Description

The RoamerR2 system consists of a GNSS-IMU positioning system and FARO Focus 3D 120S laser scanner (FARO Focus3D Features, Benefits & Technical Specifications, Lake Mary, FL, USA) for cross-track profiling of the road environment. The data acquisition and positioning sensors, NovAtel Flexpak6 GNSS receiver, 702GG antenna, and UIMU-LCI IMU (NovAtel Inc., Calgary, AB, Canada), are mounted on a single integration plate that could be mounted on any imaginable support structure, as shown in Figure 7.

In the RoamerR2 installation for this study the scanner was mounted on top of a car at 3.4 m elevation from the road surface below. Laser data was acquired at 95 Hz scan frequency and 244 kHz point measurement rate (scanner setting 1/16 at 3× noise compression) resulting in 2.4 mrad angular resolution of the data (i.e., 8 mm point spacing at nadir on the road surface). The LiDAR data was recorded on the scanner SD card, while the 200 Hz positioning data was stored on a tablet computer used to control the positioning system. The positioning data was post-processed with virtual GNSS reference base station data from Trimnet Service by Geotrim Oy, Finland, for accurate 3D trajectory of the system. Georeferenced point cloud data was then generated with proprietary tools using sensor bore-sight calibration and time stamp information.

The test dataset consist of two different point clouds: (i) Dataset 1, with 38 million points, and obtained with the vehicle moving from south to north; and (ii) Dataset 2, with 33 million points, and obtained with the vehicle moving in the opposite direction. The distance between consecutive points along the sweeps ranges between 8 mm and more than 20 cm, and the distance between sweeps along the trajectory is, on average, 10 cm.

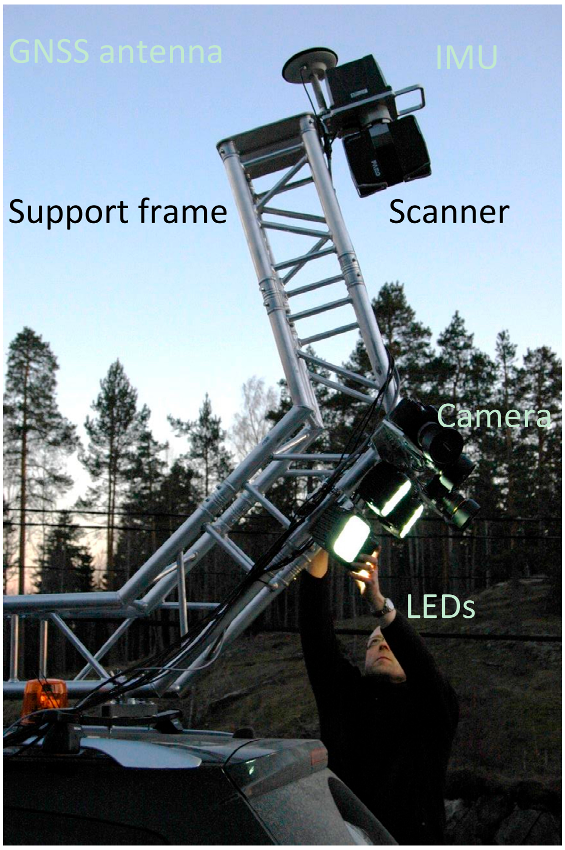


Figure 7. RoamerR2 mounted for road environment mapping using GlobalTruss frame for elevating the scanner to 3.4 m above the road surface beneath. The profile is tilted 15 degrees. LED lights could be used to light the road surface for image capture in the dark.

3.2. Ground Truth

The results from the algorithm were compared to the delineation of the road edge obtained through RTK-GNSS measurements.

The ground truth for this study consists of a polygonal line with 606 nodes on both sides of the test road section, and was collected with Trimble R8 RTK-GNSS device (Trimble Navigation Ltd., Sunnyvale, CA, USA) observing GPS and GLONASS satellites. The expected accuracy of the kinematic RTK data is 10 mm horizontally and 20 mm vertically. The point samples are unevenly spaced, as their density along the road was placed higher where the road edge was found more intricate.

3.3. Algorithm Settings

In this study, the algorithm parameters were set as shown in Table 1.

Although these parameters could be established as standard (i.e., susceptible to be used in most of the possible cases), some of them should be changed under some specific circumstances (e.g., extremely rough surfaces, or different MLS configurations). In that way, the standard Douglas Peucker tolerance (i.e., 1 cm. See Table 1) is suitable for road edge detection when asphalt roughness is under 1 cm on average, and the road edge step is higher than 1 cm. Note that for the Smoothing 1, the window width and scroll could be set off in distance units. In this test case, the window width is 4 m on average, and a node must be out of the standard deviation range in, at least, nine out of 20 positions of the moving window. As for the rest of standard parameters, moderate individual variations are barely perceptible in the final result.

Table 1. Settings applied to the test dataset.

Parameter	Value
Line clouds:	
Douglas Peucker tolerance	1 cm
Polylines split threshold	15 cm
Grouping:	
Maximum tilt angle (lines)	10°
Distance between line end nodes	65 cm
Minimum length for lines	70 cm
Minimum number of lines in a group	8
Maximum angular difference between lines:	
Tilt angle	6°
Azimuth	6°
Smoothing 1:	
Window width	40 profiles
Window scroll-step forward move	2 profiles
Number of standard deviations (as a limit)	1
Number of times node labeled as outlier	8
Smoothing 2:	
(m) parameter	$\sqrt{2}$

3.4. Metrics

Different metrics were defined in order to measure the accuracy and performance of the algorithm: (i) metrics based on areas (i.e., comparing the areas obtained by the algorithm with the ground truth); and (ii) metrics based on the distance between the road edge line from the algorithm and the ground truth.

3.4.1. Metrics Based on Areas

Two metrics compare the area obtained with the algorithm and the ground truth: (i) completeness; and (ii) correctness. Correctness expresses how much of the area detected by the algorithm corresponds to the actual road surface. Completeness shows how much of the real surface is detected by the algorithm.

Figure 8 shows how correctness and completeness are computed. Three polygons are created from (i) the RTK-GNSS ground truth (the orange patch in Figure 8); (ii) the road edge detected by the algorithm (black in Figure 8); and (iii) the intersection of the aforementioned polygons (i.e., the area that the two polygons have in common, shown in blue in Figure 8).

Correctness is computed as the proportion of the area detected by the algorithm which is inside of the actual road limits (i.e., ground truth polygon); the blue area divided by the black area in Figure 8. Completeness represents the proportion of the actual road surface that was satisfactorily detected (i.e., the blue area divided by orange area in Figure 8).

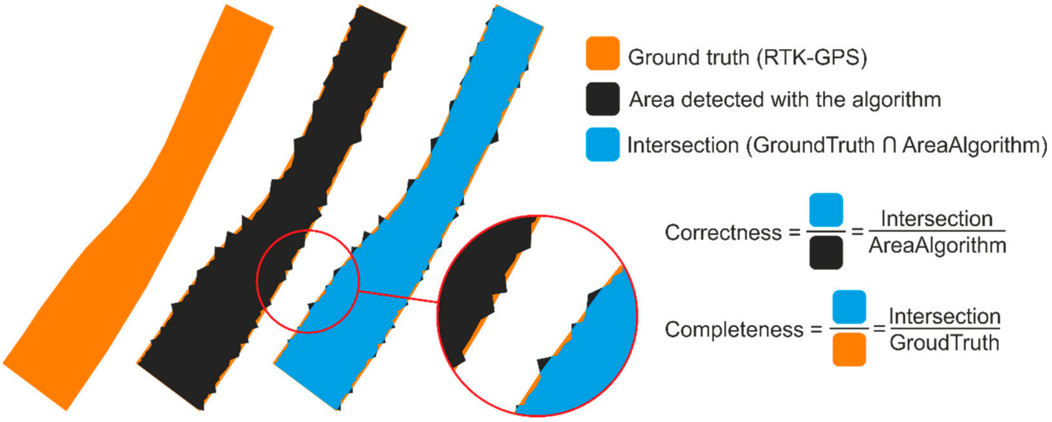

Ground truth (RTK-GPS)
Area detected with the algorithm
Intersection (GroundTruth ∩ AreaAlgorithm)
Correctness = = Intersection / AreaAlgorithm
Completeness = = Intersection / GroudTruth

Figure 8. Graphic representation of the ground truth, detected area, and how the correctness and completeness are calculated.

3.4.2. Metrics Based on Distances

The distance between the road edge detected by the algorithm and the actual border (i.e., the RTK-GNSS polylines) was also checked. This distance is measured along 426 evenly-spaced lines perpendicular to the trajectory. As shown in Figure 9, the distances from both sides of the road are checked at each perpendicular line. The distances are measured from the ground truth line towards the algorithm line, and they are stored as negative values if the algorithm line is closer to the trajectory than the RTK-GNSS line and positive otherwise.

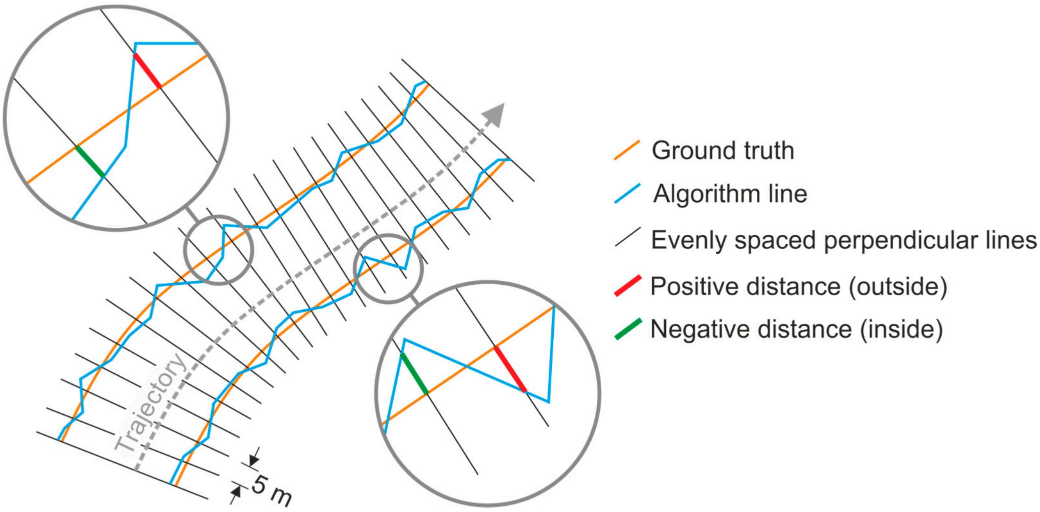

Ground truth
Algorithm line
Evenly spaced perpendicular lines
Positive distance (outside)
Negative distance (inside)
Trajectory
5 m

Figure 9. Measurement of distances (GroundTruth) → (AlgorithmLine). When the *Algorithm* line is closer to the trajectory than the *Ground Truth* line, distances along the perpendicular lines are stored as negative values, and as positive values otherwise.

3.5. Results

Correctness, completeness, and the distances along the perpendicular lines were calculated for (i) Dataset 1; (ii) Dataset 2; and (iii) the union of both datasets (i.e., the union of the polygons obtained from Dataset 1 and Dataset 2). Figure 10 shows a portion of the line cloud, the smoothed polyline and the RTK-GNSS line of a small road patch.

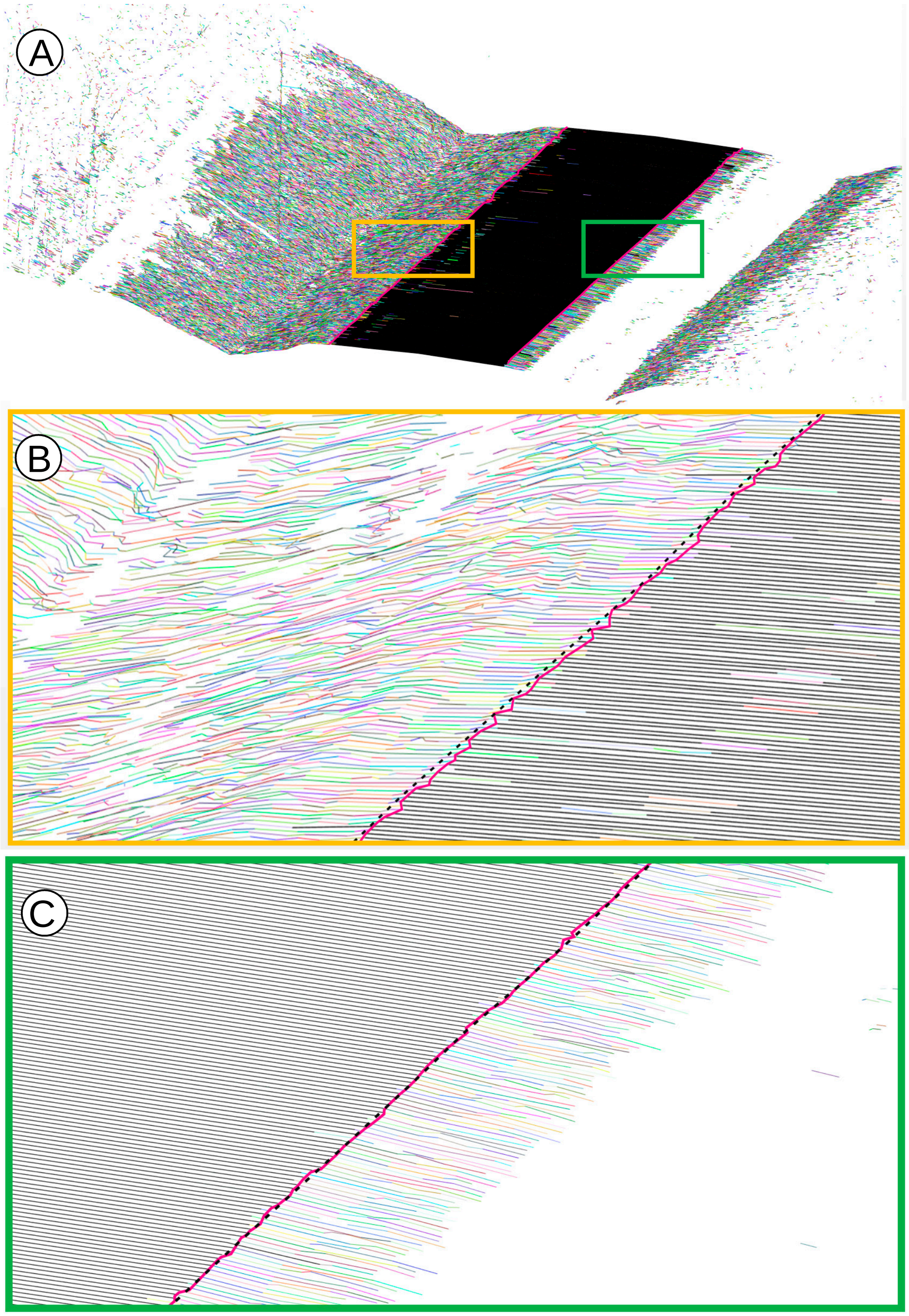


Figure 10. General view (**A**) and two zoomed details ((**B**) left edge and (**C**) right edge) of the line cloud in a small road section. The algorithm-smoothed polyline is in magenta, and the RTK-GNSS ground truth is represented by a black dotted line (only visible in the zoomed windows (orange and green rectangles)). The lines belonging to the final group are in black, and the ungrouped lines in random colors.

Figure 11 shows the smoothed polyline from the two test datasets and the union of them for three different sections (A, B, and C) of the test road.

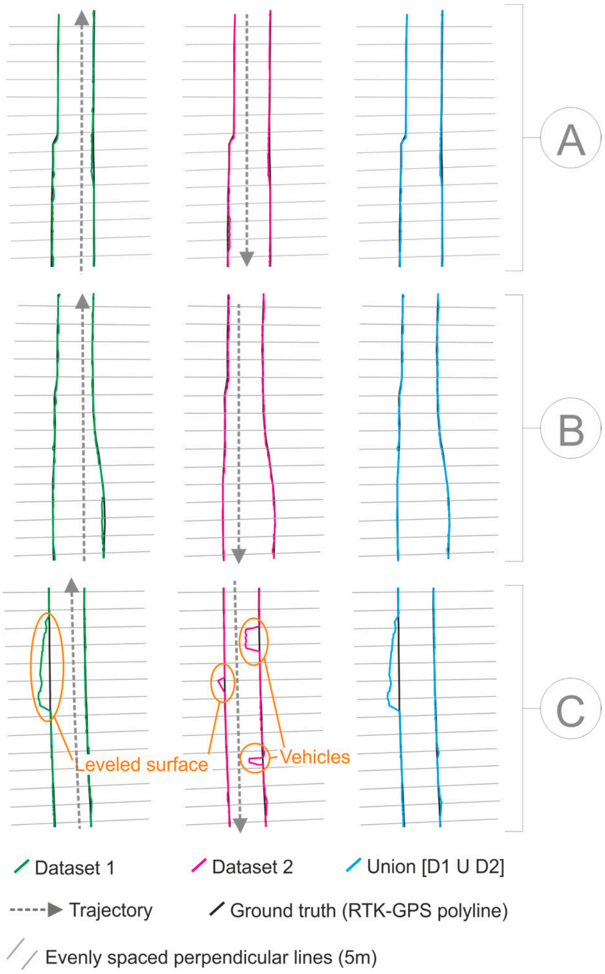

A
B
C
Leveled surface
Vehicles
Dataset 1
Dataset 2
Union [D1 U D2]
Trajectory
Ground truth (RTK-GPS polyline)
Evenly spaced perpendicular lines (5m)

Figure 11. Smoothed polylines from Dataset 1, Dataset 2, and the union of them in three different sections (**A**, **B**, and **C**) of the test stretch of road.

The three sections in Figure 11 are examples with some particularities: there is a sudden gauge widening in the left margin of A. In B, there is a smoothed widening in the right margin. The polyline from Dataset 1 in C reflects the presence of a compact terrain surface, which is almost leveled to the asphalt on the road. The polyline from Dataset 2 is affected by the presence of two vehicles moving in the opposite direction while the MLS was registering data.

The correctness and completeness values of the three polygons (i.e., Dataset 1 (D1), Dataset 2 (D2), and the union of (D1) and (D2)) are shown in the first two lines of Table 2.

Table 2. Area and distance performance metrics from the tests.

	Dataset 1		**Dataset 2**		**Union (D1)U(D2)**		
Correctness	98.94		99.07		98.35		(%)
Completeness	97.16		96.84		98.14		
Road side	West	East	West	East	West	East	
Average distance	−8.9	−11.7	−9.5	−10.5	−3.8	−6.9	(cm)
Median distance (Q2)	−11.4	−9.6	−8.9	−11.2	−7.4	−7.9	

The distances between the RTK-GNSS polyline and the algorithm lines were computed from the measurements on the 426 perpendicular lines. The last two rows from Table 2 show the average and median distances in both sides of the stretch of road. In order to compare the distribution of the distances in the test datasets, the boxplots are shown in Figure 12.

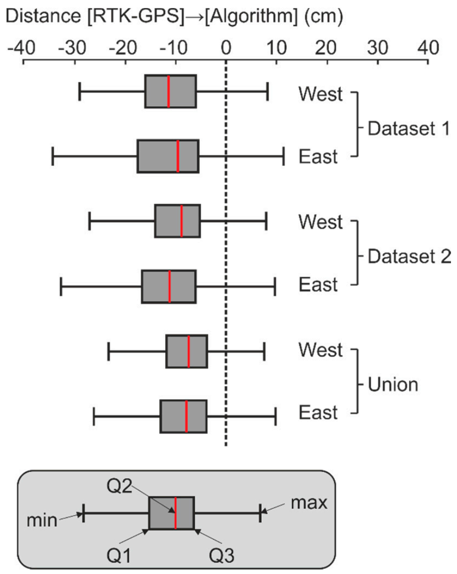


Figure 12. Boxplot of distance ((RTK-GNSS) → (Algorithm)) quantile distribution. Q1, Q2, and Q3 represent the quartiles, and min and max refer to 1.5 times the IQR (interquartile range) from the upper and lower quartiles respectively.

Tables 3–8 show the incidence of single or multiple parameter variations on correctness and completeness. These sensitivity tests were performed on a 300 m long stretch of the test road section.

Table 3. Completeness and correctness percentage for different values of the Douglas Peucker tolerance (DPT).

DPT(cm)	**0.5**	**0.6**	**0.7**	**0.9**	**1**	**1.1**	**1.3**	**1.5**	**2**	**2.5**	**3**	**5**
Correct.	99.97	99.97	99.85	99.82	99.87	99.84	99.60	99.26	97.60	92.39	88.46	81.56
Complet.	65.07	92.19	95.72	96.28	96.40	96.46	96.65	96.88	97.48	98.12	98.57	99.03

Table 4. Completeness and correctness percentage for different values of the polyline split threshold.

Polyline Split Threshold (cm)	**5**	**10**	**15**	**25**	**50**	**100**
Correctness	100	99.86	99.77	99.77	99.77	99.77
Completeness	85.15	95.64	96.40	96.40	96.40	96.40

Table 5. Completeness and correctness percentage for different values of minimum length for lines (MLL) and distance between line end nodes (DBLEN).

DBLEN	**15**	**25**	**35**	**45**	**55**	**65**	**75**	**85**	**95**	**105**	**115**	**125**
MLL(cm)	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**	**110**	**120**	**130**
Correct.	94.51	96.68	98.15	99.16	99.87	99.84	99.80	99.96	99.9	99.9	99.9	99.9
Complet.	97.0	98.1	97.8	97.08	97.6	96.46	96.15	95.7	95.55	95.00	94.88	94.55

Table 6. Completeness and correctness percentage for different values of the maximum angular difference in tilt and azimuth between lines.

Tilt	**1°**	**3°**	**5°**	**6°**	**7°**	**12°**
Azimuth	**1°**	**3°**	**5°**	**6°**	**7°**	**12°**
Correctness	99.80	99.77	**99.77**	99.77	99.77	99.77
Completeness	96.22	96.38	**96.40**	96.40	96.40	96.40

Table 7. Completeness and correctness percentage for different values of the minimum number of lines in a group to be joined to the main group.

Min. Number of Lines Per Group	1	4	8	16	40
Correctness	99.61	99.76	**99.77**	99.77	99.77
Completeness	96.53	96.43	**96.40**	96.37	96.25

Table 8. Completeness and correctness percentages for different values of the number of standard deviations as a limit for exclusion of nodes in Smoothing 1.

Number of Standard Deviations	0.1	0.5	1	2	3	6
Correctness	99.80	99.79	**99.77**	99.68	99.68	99.66
Completeness	91.77	96.20	**96.40**	95.92	95.81	95.71

For evaluating the effect of the variation of several (or all) parameters at the same time, 1000 sets of random values for all the variables (varying up to ±30% from the standard values) were created and used on the same 300 m long road section. This joint variation of all the parameters in ±30% did not produce completeness values below 95.1%, nor correctness values below 99.1%.

3.6. Discussion

The test results show that 99% of the area effectively detected by the algorithm was correct (i.e., corresponds to the actual surface), and that 97% of the road surface was correctly detected by the algorithm. If the union of the two datasets that register data from the same section (but with the MLS moving in the opposite direction) is used, both the proportion of correctly detected surface (correctness) and the proportion of the actual surface that was detected (completeness) go to 98%. See Table 2.

On the other hand, the algorithm tends to place the line around 10 cm to the inside of the road when using a single dataset, and 5–6 cm on average if using the union of the two datasets. This distance is larger on the most distant border from the trajectory (i.e., the west border when the MLS is moving from south to north [Dataset 1], and the east border in Dataset 2), see Table 2 and Figure 12. The difference between the two roadsides may be due to the effect of the lower incidence angle that produces a larger gap between consecutive points on the furthest road border. Moreover, the fact that the detected lines are, on average, 10 cm away from the RTK-GNSS line (when using a single dataset) could be fully explained by (i) the distance between points on a sweep, and (ii) the curvature of the asphalt border immediately before the edge.

- Using the MLS configuration described in Section 3.1, the distance between consecutive points on the same sweep ranges between 1.5 and 7–8 cm at the road border. These distances are most often 2 cm at the closest road edge and 4–5 cm at the furthest one, and depend on the slant distance to the sensor and the tilt angle of the surface. This explains part of the displacement of the edge line towards the center line of the road (i.e., from a few millimeters to several centimeters).
- The asphalt edge is not completely clear/sharp, but there is certain curvature on the asphalt profile immediately before the road edge. See Figure 2C. Using small Douglas Peucker tolerances (e.g., 1 cm, as the one used for the test), it is usual that the break of the line is produced towards the inside of the road. This effect could be reduced by using a larger Douglas Peucker tolerance, but this would lead to errors in shallow borders, as no break would be detected in the lines.

The surface misdetection (i.e., surface from the actual road that is not detected by the algorithm: (100%)–(completeness)) could be explained by the joint effect of (i) the fact that the detected road edge tends to be some centimeters inwards of the road; and (ii) the presence of some vehicles on the road while registering data (e.g., the two vehicles shown in Figure 11C). When using the union of two datasets, the completeness rises to 98%, as (i) the effect of the moving vehicles is eliminated (e.g., the gaps produced by the two vehicles in Figure 11 (C, Dataset 2) are removed when using the union of the two datasets); and (ii) the displacement of the detected line to the inside of the road is partially mitigated, as the polygon whose border stands out prevails. See Table 2.

The surface over-detection (i.e., surface detected by the algorithm that does not belong to the actual road: (100%)–(correctness)) is due to the absence of a clear/sharp asphalt edge in relatively long sections of the road side. These sections correspond to surfaces of compact terrain that are leveled, or almost leveled to the asphalt, such as the leveled surface in Figure 11.

The surface in Figure 11 is completely joined to the road polygon from Dataset 1, whereas in Dataset 2, a much smaller patch was joined. In this case, the surface is not completely leveled to the road, and the subtle step is only detected by the algorithm from Dataset 2, in which the west boundary is closer to the trajectory, and the gap between consecutive points is smaller. Using the data from Dataset 1, the step is too shallow if compared with the gap between points, thus, no break in the lines is detected there. The correctness of the union of the two datasets is slightly smaller than the one from Dataset 1 and 2 separately, as the leveled surfaces 'over-detected' in only one of them are added to the union.

It is also to be noted that decreasing the angular resolution (i.e., higher point measurement rate of 488 kHz or 970 kHz, or lower scan frequency) of the MLS system may improve the edge detection accuracy. A higher point density along the scanning lines could lead to better results in terms of (i) distances from the road edge polyline to the ground truth and (ii) completeness, as the displacement of the edge line to the inside of the road would be reduced. Nevertheless, there is no apparent reason for expecting different results with a different distance between lines along the trajectory, unless the road edge shape is very intricate, and the distance between sweeps is very large (e.g., 1 m). This may produce an over-smoothing of the road edge polyline, and the misdetection of some complex shapes or features.

The algorithm parameters are configurable, and their values directly affect the results and performance of the algorithm. In this study, a set of standard parameters are proposed for general use, and applied for the test; however, different values could be used for specific datasets and environments. Most of the values of the standard parameters could be modified individually by 40% or 50% without significant variations on the performance, as it is shown in Tables 3–8. In some cases, like the polyline split threshold (PST), larger variations in the parameter values could lead to larger misdetections or over-detections (see Table 4). The distance between consecutive points on the scan lines (i.e., sweeps) must be taken into account when setting up the polyline split threshold, as, if the PST is larger than the gap between points, they cannot be transformed into lines. For this reason, the PST must be larger than the maximum distance between consecutive points expected on the road surface. Nevertheless, individual large variations in most of the parameters are barely noticeable in terms of completeness and correctness. Douglas Peucker tolerance (DPT) is one of the most sensitive parameters, as it directly affects the configuration of the line cloud that is the basis for the rest of processes. A DPT value smaller than the irregularities on the road surface would produce over-segmentation of the line cloud and, therefore, would result in more misdetections (i.e., DPT values below 6 mm in Table 3), whereas a DPT larger than the height of the road edge step would avoid the line break needed for its identification and produce over-detection (e.g., DPT values above 2 cm in Table 3). On the other hand, single variations of the parameters that control the length of the lines, their parallelism, the minimum size of the groups, or the threshold for rejecting a node during the smoothing processes barely affect the performance of the algorithm (see Tables 5–8).

The parameters that control Smoothing 1 are specially interrelated. The window width controls the size of the irregularities in the polyline that the filter smooths, but the rest of the parameters must be set off accordingly (especially the scroll forward step and the threshold for a node to be labeled as an outlier). The (m) value from Smoothing 2 (see Section 2.4.2 and Table 1) controls the shape of the irregularities that are removed from the polyline. Using the standard value (i.e., $\sqrt{2}$) sharp peaks whose angle is smaller than 90° are removed when the distances (ab) and (bc) are similar, see Figure 6.

Moderate variations of several (or all) parameters at the same time (i.e., up to 30%), do not produce a significant variation of the performance of the algorithm, as completeness and correctness values do not decrease more than 1.3% and 0.7%, respectively, in any case.

Although there are several articles presenting new methods for automatically defining the shape of roads/streets from MLS data, comparing the results of the present study with previous work is not straightforward. For instance, some studies [6,24,31] focus on the extraction of road markings and, even though some of them report their performance in terms of completeness and correctness, these values are not comparable to the ones obtained in this study, as they do not try to detect the road edge. Several other studies present algorithms for automatic detection of curbs [15,22,23,25,32,34], and most of them assess completeness and correctness, but these refer to the length of the curbs detected. In contrast, Zhao et al. [28], Smadja et al. [29] and Guan et al. [35] show methods for automatically identifying road points from MLS data, but their performance is not assessed. Boyko and Funkhouser, and Bin et al. [26,30] are focused on the detection of the road surface, but limited to the presence of curbs. Misdetections and over-detections on the street surface are evaluated in Bin et al. [30], however, they are calculated based on the detection of points, rather than based on the surfaces. Completeness and correctness are reported in Boyko and Funkhouser [26] in a similar way to this study, although a 0.5 m × 0.5 m grid is used for surface comparisons. Completeness is 94%, and correctness is 86%. Kumar et al. [36] presented a method for road edge detection from MLS data, where elevation, reflectance/intensity, and pulse width, along with the trajectory, were used. The algorithm was tested in three small road sections, and the distances from the detected edges to the actual road edge were checked. In their experiments, the mean distances from the detected road edge to the ground truth ranged between 2 cm and 38 cm, depending on the type of road edge, the point density, the side of the road (i.e., closest or furthest border to the trajectory), and the kind and quality of the data available in each test dataset (i.e., elevation, reflectance/intensity, and pulse width).

4. Conclusions

In this study, an algorithm for the automatic delineation of road edges from MLS datasets was developed and tested. The method is based on the initial transformation of the original point cloud into a structured line cloud. The lines are grouped following a set of geometric criteria based on their parallelism and proximity, and the group containing the lines on the road is subsequently identified. An initial road edge polyline is obtained from the end nodes of the road line group, which is finally smoothed following a two-stage filtering.

The algorithm was tested on two datasets from RoamerR2 on a 2.1 km stretch of road. Similar results were obtained using both datasets separately: 99% surface correctness (i.e., proportion of detected surface that is within the actual road) and 97% surface completeness (i.e., proportion of the actual road that is detected by the algorithm). If the union of the two datasets is used, both completeness and correctness go to 98%.

The algorithm performance is affected by (i) point density along the scanning line; (ii) possible curvature of the asphalt border; (iii) presence of leveled surfaces of compact terrain on the sides of the road; and (iv) presence of vehicles on the road. The effect of all of them (except for the detection of leveled surfaces) is reduced by using the union of two datasets from the same road section and registered with the MLS moving in opposite directions.

For the further development of the road edge detection methodology there is an unused resource to be explored: laser intensity. That could give extra hints in placing the asphalt edge correctly, particularly in locations where the transition from the asphalt to the dirt/gravel is too smooth for geometric detection. There are also scanners operating at different wavelengths and, in particular, those responsive to moisture could be an answer to the edge localization task in the future study.

Acknowledgments: This study was made possible by financial funding from the Academy of Finland for "Centre of Excellence in Laser Scanning Research (CoE-LaSR) (272195)". Strategic Research Council at the Academy of Finland is acknowledged for financial support for "Competence Based Growth through Integrated Disruptive Technologies of 3D Digitalization, Robotics, Geospatial Information and Image Processing/Computing Point Cloud Ecosystem" (293389). Carlos Cabo is in receipt of a "Severo Ochoa" Grant provided by FICYT-Government of Principado de Asturias.

Author Contributions: Carlos Cabo developed the algorithm, performed the analysis of the test data and wrote most of the paper. Silverio García-Cortés and Celestino Ordoñez supervised the study. Antero Kukko, Harri Kaartinen and Juha Hyyppä provided the infrastructure/MLS systems, the test data, described the MLS devices, and supervised the algorithm.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Guan, H.; Li, J.; Yu, Y.; Chapman, M.; Wang, C. Automated road information extraction from mobile laser scanning data. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 194–205. [CrossRef]
2. Goad, C.C. The Ohio State University mapping system: The Positioning Component. In Proceedings of the 47th Annual Meeting The Institute of Navigation, (ION), Williamsburg, VA, USA, 10–12 June 1991; pp. 121–124.
3. Li, R. Mobile mapping: An emerging technology for spatial data acquisition. *Photogramm. Eng. Remote Sens.* **1997**, *63*, 1085–1092.
4. Puente, I.; González-Jorge, H.; Martínez-Sánchez, J.; Arias, P. Review of mobile mapping and surveying technologies. *Meas. J. Int. Meas. Confed.* **2013**, *46*, 2127–2145. [CrossRef]
5. Kukko, A.; Kaartinen, H.; Hyyppä, J.; Chen, Y. Multiplatform mobile laser scanning: Usability and performance. *Sensors* **2012**, *12*, 11712–11733. [CrossRef]
6. Jaakkola, A.; Hyyppä, J.; Hyyppä, H.; Kukko, A. Retrieval algorithms for road surface modelling using laser-based mobile mapping. *Sensors* **2008**, *8*, 5238–5249. [CrossRef]
7. Lehtomaki, M.; Jaakkola, A.; Hyyppä, J.; Lampinen, J.; Kaartinen, H.; Kukko, A.; Puttonen, E.; Hyyppä, H. Object classification and recognition from mobile laser scanning point clouds in a road environment. *IEEE Trans. Geosci. Remote Sens.* **2015**, *54*, 1226–1239. [CrossRef]
8. Calle, M.; Lotsari, E.; Kukko, A.; Alho, P.; Kaartinen, H.; Rodriguez-Lloveras, X.; Benito, G. Morphodynamics of an ephemeral gravel-bed stream combining Mobile Laser Scanner, hydraulic simulations and geomorphological indicators. *Z. Geomorphol.* **2015**, *59*, 33–57. [CrossRef]
9. Liang, X.; Hyyppä, J.; Kukko, A.; Kaartinen, H.; Jaakkola, A.; Yu, X. The use of a mobile laser scanning system for mapping large forest plots. *IEEE Geosci. Remote Sens. Lett.* **2014**, *11*, 1504–1508. [CrossRef]
10. Vosselman, G.; Maas, H.-G. *Airborne and Terrestrial Laser Scanning*; CRC: Boca Raton, FL, USA, 2010; p. 318.
11. Axelsson, P. DEM generation from laser scanner data using adaptive tin models. *Int. Arch. Photogramm. Remote Sens.* **2000**, *33*, 110–117.
12. Marcotegui, B.; Paristech, M.; Earth, V.; Hernández, J. Filtering of artifacts and pavement segmentation from mobile LiDAR data. *ISPRS Workshop Laserscanning* **2009**, *38*, 329–333.
13. Schwarz, K.; El-Sheimy, N. Mobile Mapping systems state of the art and future trends. *Int. Arch. Photogramm. Remote Sens.* **2004**, *35*, 10.
14. Manandhar, D.; Shibasaki, R. Feature extraction from range data. In Proceedings of the 22nd Asian Conference on Remote Sensing, Singapore, 5–9 November 2001.
15. Yang, B.; Fang, L.; Li, J. Semi-automated extraction and delineation of 3D roads of street scene from mobile laser scanning point clouds. *ISPRS J. Photogramm. Remote Sens.* **2013**, *79*, 80–93. [CrossRef]
16. Vosselman, G.; Gorte, B.; Sithole, G.; Rabbani, T. Recognising structure in laser scanning point clouds. In *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*; ISPRS: Vienna, Austria, 2004; pp. 33–38.
17. Biosca, J.M.; Lerma, J.L. Unsupervised robust planar segmentation of terrestrial laser scanner point clouds based on fuzzy clustering methods. *ISPRS J. Photogramm. Remote Sens.* **2008**, *63*, 84–98. [CrossRef]
18. Cabo, C.; Ordoñez, C.; García-Cortés, S.; Martínez, J. An algorithm for automatic detection of pole-like street furniture objects from Mobile Laser Scanner point clouds. *ISPRS J. Photogramm. Remote Sens.* **2014**, *87*, 47–56. [CrossRef]

19. Rabbani, T.; Dijkman, S.; Van den Heuvel, F.; Vosselman, G. An integrated approach for modelling and global registration of point clouds. *ISPRS J. Photogramm. Remote Sens.* **2007**, *61*, 355–370. [CrossRef]
20. Lehtomäki, M.; Jaakkola, A.; Hyyppä, J.; Kukko, A.; Kaartinen, H. Detection of vertical pole-like objects in a road environment using vehicle-based laser scanning data. *Remote Sens.* **2010**, *2*, 641–664. [CrossRef]
21. Yang, B.; Dong, Z. A shape-based segmentation method for mobile laser scanning point clouds. *ISPRS J. Photogramm. Remote Sens.* **2013**, *81*, 19–30. [CrossRef]
22. Serna, A.; Marcotegui, B. Urban accessibility diagnosis from mobile laser scanning data. *ISPRS J. Photogramm. Remote Sens.* **2013**, *84*, 23–32. [CrossRef]
23. Rodríguez-Cuenca, B.; García-Cortés, S.; Ordóñez, C.; Alonso, M.C. An approach to detect and delineate street curbs from MLS 3D point cloud data. *Autom. Constr.* **2015**, *51*, 103–112. [CrossRef]
24. Ruyi, J.; Reinhard, K.; Tobi, V.; Shigang, W. Lane detection and tracking using a new lane model and distance transform. *Mach. Vis. Appl.* **2011**, *22*, 721–737. [CrossRef]
25. El-Halawany, S.; Moussa, A.; Lichti, D.; El-Sheiny, N. Detection of road curb from mobile terrestrial laser scanner point cloud. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*; ISPRS: Calgary, AB, Canada, 2011.
26. Boyko, A.; Funkhouser, T. Extracting roads from dense point clouds in large scale urban environment. *ISPRS J. Photogramm. Remote Sens.* **2011**, *66*, S2–S12. [CrossRef]
27. Cabo, C.; García-Cortés, S.; Ordoñez, C. Mobile Laser Scanner data for automatic surface detection based on line arrangement. *Autom. Constr.* **2015**, *58*, 28–37. [CrossRef]
28. Zhao, H.; Shibasaki, R. Surface modelling of urban 3D objects from vehicle-borne laser range data. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2002**, *34*, 412–417.
29. Smadja, L.; Ninot, J.; Gavrilovic, T. Road extraction and environment interpretation from LiDAR sensors. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2010**, *38*, 281–286.
30. Bin, W.; Bailang, Y.; Chang, H.; Qiusheng, W.; Jianping, W. Automated extraction of ground surface along urban roads from mobile laser scanning point clouds. *Remote Sens. Lett.* **2016**, *7*, 170–179.
31. Kumar, P.; McElhinney, C.; Lewis, P.; McCarthy, T. Automated road markings extraction from mobile laser scanning data. *Int. J. Appl. Earth Obs. Geoinform.* **2014**, *32*, 125–137. [CrossRef]
32. Fang, L.; Yang, B.; Chen, Ch.; Fu, H. Extracting 3D road boundaries from mobile laser scanning point clouds. In Proceedings of the IEEE International Conference on Spatial Data Mining and Geographical Knowledge Services (ICSDM), Fuzhou, China, 8–10 July 2015; pp. 162–165.
33. Douglas, D.H.; Peucker, T.K. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartogr. Int. J. Geogr. Inf. Geovis.* **1973**, *10*, 112–122. [CrossRef]
34. Zhao, G.; Yuan, J. Curb detection and tracking using 3d-lidar scanner. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 30 September–3 October 2012; pp. 437–440.
35. Guan, H.; Li, J.; Yu, Y.; Wang, C.; Chapman, M.; Yang, B. Using mobile laser scanning data for automated extraction of road markings. *ISPRS J. Photogramm. Remote Sens.* **2014**, *87*, 93–107. [CrossRef]
36. Kumar, P.; Lewis, P.; McElhinney, C.P.; Boguslawski, P.; Mc-Carthy, T. Snake energy analysis and results validation for a mobile laser scanning data based automated road edge extraction algorithm. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2016**. [CrossRef]

Capítulo V

Discusión General

5.1 Estructuras de simplificación y homogenización. Voxelización y nubes de líneas

En esta tesis se desarrollan dos estructuras de simplificación y homogeneización de las nubes de puntos de escáneres láser móviles (MLS): voxelización, y nube de líneas. Ambas suponen la base para la creación de versiones reducidas de los datos originales, cuya función es la agilización de procesos y la formulación de abstracciones más sencillas.

Tanto la voxelización como las nubes de puntos son estructuras que agrupan puntos a partir de su inclusión en figuras geométricas: la voxelización utiliza ortoedros que incluyen todos los puntos de la nube original que están dentro de ellos, y las nubes de líneas utilizan segmentos rectos (denotados por sus nodos inicial y final), que incluyen todos los puntos que están a menos de una distancia umbral de cada línea. En ambos casos, se elimina la información que se considera redundante para un estudio en concreto o parte de él. En este sentido, al aplicar la voxelización a un grupo de puntos~~,~~ se están normalizando (o simplificando) sus coordenadas a un nivel que viene definido por el tamaño de los vóxeles. Es decir, al voxelizar se elimina el detalle a niveles inferiores del tamaño del vóxel en cada una de las dimensiones. El nivel de reducción dependerá de básicamente del tamaño de los vóxeles y de la densidad y distribución de los puntos. De esta forma, como se puede apreciar en los datos de la Tabla 3 del Capítulo II, este nivel de reducción llega a ser casi nulo cuando los tamaños de vóxel utilizados es tal, que es improbable encontrar más de un punto dentro del mismo vóxel.

En las nubes de líneas se analizan los puntos consecutivos alineados. Se almacenan únicamente los puntos extremos de cada segmento recto, y se prescinde de los puntos intermedios, al considerar que la información que aportan es redundante para el propósito por el que son simplificados. Así, se establece una distancia entre los puntos y la recta a la que podrían pertenecer, y se consideran irrelevantes separaciones punto-línea por debajo de ese umbral (achacables a la rugosidad del objeto, o a desviaciones de la medida LiDAR).

Las dos estructuras son reversibles. Es decir, se establecen los mecanismos necesarios durante su creación para que los puntos correspondientes a cada elemento estructural (vóxel o línea) sean directa e inmediatamente accesibles. De esta forma, la simplificación no supone una pérdida de información; simplemente proporciona una versión más ágil y con cierto contenido semántico (en el caso de las nubes de líneas). La recuperación de los puntos originales siempre es posible, y estos heredan las etiquetas adquiridas por los vóxeles o las líneas en los procesos clasificación o selección a los que hayan podido ser sometidos.

La voxelización aquí desarrollada se reduce a un conjunto de operaciones algebraicas, y la ordenación de un vector. Esto supone una reducción notable de los requerimientos computacionales para la creación de otras estructuras de ortoedros que incluyen relaciones jerárquicas (como los octrees y los kd-trees) (Vanderhyde and Szymczak, 2008; Elseberg et al., 2013). Por otro lado, son muchos los estudios que refieren el uso de la voxelización para simplificar las nubes de puntos que utilizan (Aijazi et al., 2013; Hosoi y Omasa, 2006; Moskal y Zheng, 2012; Truong-Hong et al., 2013; Wu et al., 2013). Sin embargo, el método propuesto en esta tesis permite la recuperación de los datos originales, por lo que no es necesario asumir pérdidas de detalle irreversibles.

Las nubes de líneas suponen una simplificación de la nube de puntos, y una estructuración de los datos que permite la formulación de abstracciones más sencillas en algunos casos. Esto último ocurre con los objetos sobre los que los puntos escaneados tienden a alinearse: en general, superficies planas o regladas.

5.2 Detección automática de elementos

5.2.1 Detección de Postes:

Para la detección de objetos con forma de poste, se voxeliza la nube de puntos, se realiza la detección de postes sobre la versión voxelizada, y se etiquetan los vóxeles que corresponden a cada poste detectado. Finalmente, los puntos correspondientes cada uno de los vóxeles seleccionados heredan sus etiquetas. La aplicación del algoritmo conlleva el uso de varios parámetros, siendo el tamaño de los vóxeles el más importante de todos ellos, ya que afecta directamente a la tasa de reducción de datos y, por tanto, a la agilidad de todos los procesos. El resto de parámetros se establecen como un número entero de veces el tamaño del vóxel.

Aunque sería posible variar los parámetros para cada caso de estudio, se propone un conjunto de parámetros estándar que son capaces de detectar la mayoría de postes en entornos urbanos y periurbanos. De esta forma, con la configuración estándar se buscan postes de una altura mínima de 1,2m, radio y sección máxima de 0,15m y 0,06m^2 respectivamente, y que tengan cierto aislamiento, de forma que no se encuentren (para cada sección) más de tres puntos entre el radio máximo interior (0,15m) y el radio exterior del anillo de aislamiento (0,45m).

Usando la configuración estándar, se detectan todos los postes presentes en los datos de prueba, con la única excepción de aquellos que se ven afectados por (i) grandes ocultamientos que no permiten ver la longitud mínima de poste establecida para considerarlos como tales (1,2m), y (ii) aquellos postes muy próximos a otros objetos, como contenedores, muros, o vegetación, que impiden cumplir con los criterios de aislamiento. La tasa de detección, que mide la proporción de postes existentes que son detectados, es 92,3%.

La tasa de corrección de la detección mide la proporción de elementos detectados como postes que realmente eran postes. Su valor medio en los datos de prueba es de 83.8%. Todos los postes detectados son correctos, a excepción de (i) varios elementos con forma de poste visibles a través de grandes cristaleras de escaparates, y (ii) varias ramas desnudas y casi verticales.

La comparación del algoritmo con estudios anteriores es compleja: (i) los objetivos de algunos de ellos no se corresponden con los de este estudio (Wu et al., 2013; Golovinskiy et al., 2009), (ii) algunos estudian de forma separada diferentes tipos de postes y se centran en la clasificación (Pu et al., 2011), y (iii) los datos de prueba son diferentes y no reproducibles.

Aunque no es posible establecer una comparación rigurosa de resultados, las tasas de detección y de corrección del algoritmo aquí desarrollado superan de forma significativa a las de los estudios previos (Pu et al., 2011; Golovinskiy et al., 2009; Yokoyama et al.,2011; Lehtomäki et al., 2010; Lehtomäki et al.,2011). Solo las tasas de detección y corrección de (Wu et al., 2013) son mayores; sin embargo, (i) solo detecta árboles, (ii) el método está diseñado para ser usado en terreno horizontal, y (iii) los árboles han de estar a la misma altura respecto la trayectoria del vehículo. Al margen de las tasas de detección, el algoritmo no se ve afectado por la presencia de grandes estructuras unidas a los postes. Por ejemplo, detecta sin problema postes unidos por ramas de árboles u otros elementos, superando así limitaciones de otros métodos como (Pu et al., 2011; Yokoyama et al.,2011). No necesita datos de entrenamiento como (Golovinskiy et al., 2009), ni tiene unos requerimientos específicos en cuanto a la geometría de la nube de puntos y el sistema de adquisición de datos, como (Lehtomäki et al., 2010; Lehtomäki et al.,2011).

5.2.2 Detección de Superficies:

Se genera una nube de líneas a partir de la nube de puntos, se agrupan las líneas siguiendo una serie de reglas de paralelismo y proximidad de nodos, se unen los grupos de líneas que pertenecen a la misma superficie, y se etiquetan las líneas de cada grupo final. Por último, se recuperan las los puntos de las líneas agrupadas, que heredarán sus etiquetas.

Aunque, como en el algoritmo de detección de postes, es posible variar los parámetros para cada nube de puntos que se evalúe, se propone un grupo de parámetros estándar, que es válido para detectar superficies en la mayor parte de las situaciones. Si bien, el algoritmo es capaz de detectar superficies con independencia de su inclinación, las pruebas están dirigidas a la extracción de superficies verticales. Es decir, fachadas de casas o edificios, muros, paneles, etc.

Se establecen los umbrales de salto entre puntos ("*gap*") y distancia máxima entre nodos, con criterios de mínima exclusión de líneas. Es decir, se considera muy poco frecuente que existan dos puntos consecutivos en la nube original que pertenezcan a la misma superficie y que estén separados más de 0,5m. De la misma forma, es muy poco habitual que la distancia entre nodos finales o iniciales de dos líneas consecutivas sea mayor que 0,7m. Los umbrales angulares de paralelismo se establecen en valores relativamente bajos, con el único fin de acotar la búsqueda entre las líneas de cada perfil y reducir el tiempo de procesado.

La incidencia de tolerancia Douglas Peucker está influida por (i) la precisión relativa de los puntos dentro de un mismo barrido, y (ii) la rugosidad de las superficies analizadas. Este valor ha de ser igual o mayor que las desviaciones (respecto a líneas rectas) que se esperan en los puntos sobre las superficies buscadas. La tolerancia Douglas Peucker puede y debe ser variada si se espera que el efecto conjunto de la rugosidad y la precisión relativa sea mayor que el valor estándar (0,05m).

El 99,9% de los puntos asignados a alguna superficie, realmente pertenecen a ella. Todos los puntos asignados erróneamente corresponden a líneas en las que la mayoría de sus puntos pertenecen a la superficie, por lo que la sobredetección puede achacarse a errores en la fase de simplificación (creación de la nube de líneas). Esto podría evitarse utilizando una tolerancia Douglas Peucker menor. Sin embargo, se podría producir una sobresegmentación de la nube de líneas, que dificultaría el agrupamiento.

Más del 90% de los puntos tomados por el MLS en cada una de las 27 superficies analizadas fueron detectados y agrupados por el algoritmo. La presencia de líneas no agrupadas se debe a la presencia de (i) irregularidades en la superficie superiores a la tolerancia Douglas Peucker, (ii) artefactos sobre las superficies (como, por ejemplo, hiedra), u (iii) ocultamientos que producen pequeños parches sin puntos cerca de los bordes de las superficies. Sin embargo, en todos los casos, el número de líneas (y puntos) detectadas permite la interpretación y caracterización rigurosa de superficies.

La comparación del funcionamiento y resultados del algoritmo con estudios previos es compleja, ya que los objetivos y la forma de cuantificar las tasas de acierto y error son diferentes en todos ellos. Sin embargo, algunos indicadores de rendimiento de otros métodos pueden ser relevantes si se comparan con las pruebas de validación realizadas en este estudio. Esta comparación sugiere que el rendimiento del algoritmo aquí desarrollado es al menos tan bueno como el de los estudios anteriores en cuanto a las tasas de detección y corrección de la detección, y que es más general, no limitándose a superficies estrictamente planas ni verticales (Jochem et al., 2011; Fan et al., 2014; Yang y Dong, 2013; Lari y Habib, 2014; Sithole y Vosselman, 2003)

5.2.3 Detección de la Superficie asfaltada:

Se transforma la nube de puntos en una nube de líneas, se agrupan las líneas, se unen los grupos sobre la misma superficie, y se selecciona la superficie sobre la que se proyecta la trayectoria del vehículo. Esta superficie corresponderá con la superficie asfaltada de la calle o carretera por la que circula el MLS. Se delinea el contorno del grupo de líneas seleccionado, y se somete a dos procesos de suavizado, para evitar perfiles dentados. En el caso de contar con datos de más de un sensor, se pueden aplicar estos procesos por separado a cada nube de puntos, para unir las líneas poligonales resultantes.

Al igual que en los dos algoritmos anteriores, los parámetros son configurables para casos especiales, pero se proponen unos valores estándar que se espera sean válidos para la mayor parte de las nubes de puntos. Como se puede apreciar en las Tablas 3-8 del Capítulo III, la mayor parte de los parámetros se podrían modificar individualmente en un 40% o 50% (o de forma conjunta hasta un 30%) sin cambios significativos en los resultados. Sin embargo, mayores variaciones podrían dar lugar a mayores errores de omisión y/o comisión.

Uno de los principales parámetros es la tolerancia Douglas Peucker, ya que afecta directamente a la configuración de la nube de líneas, que es la base del resto de los procesos. De esta manera, valores muy pequeños de esta tolerancia (menores que la rugosidad del asfalto) pueden producir errores de omisión, y valores muy grandes (mayores que la discontinuidad en el borde del asfalto) pueden producir sobredetecciónes (errores de comisión).

Los resultados de las pruebas realizadas al algoritmo muestran que el 99% del área delineada automáticamente se corresponde son la superficie asfaltada real y que el 97% de la superficie es detectada correctamente. Aplicando el algoritmo a dos nubes de puntos tomadas en sentidos de circulación diferentes, ambas tasas (detección y corrección de la detección) pasan a ser 98%.

Por otro lado, el algoritmo tiende a situar el borde del asfalto unos 10 cm hacia el interior de la carretera cundo se usa una sola nube de puntos, y 5-6cm cuando se utiliza la unión de las dos. Esta distancia es mayor en el borde más distante de la trayectoria del MLS. Esta diferencia podría ser debida al ángulo de incidencia (más bajo en distancias mayores sobre la misma superficie), que produce separaciones mayores entre puntos consecutivos en el mismo barrido. Las diferencias entre el borde detectado y el real (medido en campo con equipos RTK-GNSS [del inglés Real Time Kinematic – Global Navigation Satellite System]) podrían explicarse por la unión de dos factores: (i) la distancia entre puntos consecutivos en el mismo barrido, y (ii) la curvatura del borde del asfalto inmediatamente antes de su límite.

Aunque existen numerosos estudios que presentan métodos para definir automáticamente la forma de calles y carreteras a partir de datos MLS, comparar sus resultados supone una labor compleja. Por ejemplo, algunos estudios (Jaakkola et al., 2008; Ruyi et al., 2011; Kumar et al., 2014) están dirigidos a la extracción de marcas viales. Otros (Yang et al., 2013; Serna y Marcotegui, 2013; Rodríguez-Cuenca et al., 2015; El-Halawany et al., 2011; Fang et al., 2015; Zhao y Yuan, 2012) presentan algoritmos para la detección automática de bordillos, y presentan sus resultados en términos de longitud de bordillo detectado. Por el contrario, (Zhao y Shibasaki, 2002; Smadja et al., 2010; Guan et al., 2014) presentan métodos para la identificación automática de puntos de la carretera/calle, pero no cuantifican las tasas de detección y corrección. Boyko y Funkhouser (2011) y Bin et al. (2016) proponen métodos para la detección de la superficie de carretera, pero limitados a la presencia de bordillos. En Bin et al. (2016) se muestran resultados en términos de tasas de detección y de corrección de la detección, pero están calculados en base a la proporción de puntos detectados, y no a la superficie asfaltada. En Boyko y Funkhouser (2011), las tasas de detección están calculadas de forma similar a este estudio, aunque utilizan una malla de 0,5 x 0,5 m para hacer comparaciones de superficie. Detectan un 94% de la superficie asfaltada, y un 86% de la superficie que detectan realmente pertenece a la calle. Kumar et al. (2016) presentan un método que utiliza reflectancia/intensidad y datos del pulso reflejado, además de la trayectoria y las coordenadas de la nube de puntos. Dependiendo del tipo y cantidad de los datos, el tipo de carretera, y el lado de la vía, obtienen una línea de borde de asfalto con unas diferencias medias con la verdad terreno entre 2 y 38cm.

Capítulo VI

Conclusiones

6.1 Conclusiones

Una de las principales contribuciones de esta tesis es el desarrollo de dos estructuras para reducir los efectos de la gran cantidad de datos y la heterogeneidad de la distribución de los puntos de nubes de escáneres láser móviles: voxelización, y nubes de líneas. En base a estas dos estructuras, se desarrollan tres algoritmos de detección automática: (i) un algoritmo de detección automática de postes y los elementos situados sobre ellos (como señales, semáforos o farolas), (ii) un algoritmo de detección automática de superficies (como fachadas, muros o paneles), y (iii) un algoritmo para la determinación automática del límite de la superficie asfaltada en calles y carreteras. No obstante, estas estructuras pueden ser aplicadas a otros problemas de detección y extracción de objetos en nubes de puntos.

Las dos estructuras de simplificación (voxelización y nube de líneas) permiten la creación de versiones reducidas y homogeneizadas de la nube de puntos. La transformación de una nube de puntos en cualquiera de estas estructuras produce una versión simplificada sobre la que se pueden aplicar los algoritmos de detección automática de un modo más ágil, y permite~~n~~ la formulación de abstracciones más sencillas. Durante los procesos de detección automática, las estructuras simplificadas (vóxels y/o líneas) son etiquetadas y, posteriormente, se deshace la transformación para recuperar los puntos originales. Los puntos heredan las etiquetas de las estructuras simplificadas.

Los tres algoritmos de detección automática alcanzan los objetivos, que están basados en superar algunas de las limitaciones de los estudios previos, y en mejorar su eficacia.

Detección de postes:

Se desarrolla un algoritmo para la extracción automática de objetos con forma de poste a partir de datos MLS, y se prueba en cuatro zonas. Se lleva a cabo una simplificación inicial que consiste en la adaptación de la nube de puntos a un espacio de vóxeles regulares. Se aplica un análisis bidimensional para detectar las secciones de postes a partir de su forma y área. Finalmente, los postes son reconstruidos a partir de sus secciones.

El método se somete a pruebas en cuatro zonas en ambientes urbanos y periurbanos. Los resultados son evaluados en términos de tasa de detección (proporción de postes existentes en las zonas de prueba que son detectados por el algoritmo), y tasa de corrección de la detección (proporción de los elementos detectados por el algoritmo que realmente son postes). La tasa de detección total es de 92,3%, y la tasa de corrección de la detección es de 83.8%. Solo quedan sin detectar algunos postes ocultos tras grandes objetos, o muy cercanos a otros elementos.

El algoritmo de detección de postes es más general y mejora la eficacia de estudios anteriores: (i) Detecta postes unidos a otras estructuras o conectados a otros postes, (ii) es totalmente automático y no requiere el uso de datos de entrenamiento, (iii) no son necesarias asunciones iniciales a cerca de la posición de los postes, y (iv) es independiente de la geometría del escaneado y solo requiere las coordenadas de los puntos originales.

Detección de superficies:

Se propone un algoritmo para la detección de superficies planas y cuasi planas a partir de nubes de puntos MLS. Este método transforma la nube de puntos original en una estructura de nube de líneas con unos requerimientos de almacenamiento mucho menores. Las líneas son agrupadas siguiendo un conjunto de reglas de paralelismo y proximidad de nodos, describiendo y conformando las superficies registradas por el sensor.

La eficacia del algoritmo es evaluada en 27 superficies en una nube de puntos MLS. La tasa de detección (proporción de puntos agrupados que realmente están sobre la superficie) es mayor de 99% en todas las superficies, siendo 100% en 12 de ellas. La tasa de detección (proporción de puntos sobre las superficies, que el algoritmo agrupa correctamente) tiene un valor medio de más de 90%, con valores entre 80% y 97% en 26 de las 27 superficies.

Respecto al cumplimiento de los objetivos inicialmente propuestos: (i) Las superficies planas y algunas superficies regladas son detectadas con éxito por el algoritmo. (ii) El método es totalmente automático y no requiere datos de entrenamiento. (iii) No está limitado a superficies verticales. (iv) En todo el proceso no son necesarios más datos que las coordenadas y el tiempo de registro de cada punto.

Delineación del borde de asfalto:

Se desarrolla y prueba un algoritmo para la delineación del borde de carreteras a partir de datos MLS. EL método está basado en la transformación inicial de la nube de puntos en una nube de líneas estructurada. Las líneas se agrupan siguiendo una serie de reglas geométricas basadas en su paralelismo y proximidad, y se identifica el grupo de la superficie asfaltada.

Se obtiene un~~a~~ contorno inicial uniendo los nodos del grupo de líneas de la superficie asfaltada, que es finalmente suavizada siguiendo un filtrado de dos pasos. El algoritmo trabaja usando exclusivamente datos geométricos (coordenadas de los puntos, tiempo de registro de cada punto y trayectoria del MLS), y es capaz de afrontar la presencia de irregularidades y discontinuidades (como grietas, gravilla, o diferentes parches de asfalto).

El algoritmo fue probado con dos nubes de puntos del RoamerR2 en un tramo de carretera de 2,1 km. Se obtuvieron resultados similares utilizando las dos nubes por separado: una tasa de corrección de la detección (proporción de la superficie detectada que realmente pertenece a la carretera) de 99%, y una tasa de detección (proporción de la carretera que es detectada por el algoritmo) de 97%. Si se usa la unión de ambas nubes de puntos, la tasa de detección y la de corrección pasan a ser de 98%.

La eficacia del algoritmo se ve afectada por (i) la densidad de puntos a lo largo de la línea de escaneado, (ii) la posible curvatura del borde del asfalto, (iii) la presencia de superficies niveladas con el asfalto a los lados de la carretera, y (iv) la presencia de vehículos en la carretera. El efecto de todos ellos (excepto el de las superficies niveladas con la carretera) se reduce con el uso de la unión de las dos nubes de puntos del mismo tramo de carretera y registrados con el MLS circulando en sentidos opuestos.

6.2 Conclusions

In this PhD thesis, two data structures for reducing the effects of the large amount of data and the heterogeneity of the MLS point distribution are developed: voxelization, and line clouds. Based on these two structures, three algorithms for automatic feature detection are developed: (i) an algorithm for automatic detection of pole-like street furniture objects (e.g. traffic signs, traffic lights or lampposts), (ii) an algorithm for automatic surface detection (e.g. façades, walls or panels); and (iii) an algorithm for automatic delineation of road and street edges. In addition, these two structures could be applied for solving different feature detection problems.

The two simplification structures (i.e. voxelization and line clouds) allow the creation of reduced and homogenized versions of the point cloud. The transformation of a point cloud into any of these structures produces a simplified version of the data on which the feature detection algorithms can be applied in a more agile way, and allow the formulation of simpler abstractions. During these feature detection processes, the simplified structures (voxels and/or lines) are labelled, and then, the transformation is reversed in order to recover the original points. These points inherit the labels from the simplified structures.

The three algorithms for automatic feature detection meet the objectives, which are based on overcoming some of the drawbacks of previous studies, and improving their performance.

Pole-like objects detection:

An algorithm for automatic identification and extraction of pole-like objects from MLS data is developed and subsequently tested in four test sites. An initial simplification process is performed by fitting the original point cloud in a regular voxel space. A two-dimensional analysis is then applied in order to detect pole-like sections according to their shape and area. Finally, the pole-like objects are reconstructed in 3D.

The method is tested in four test sites from two different datasets in urban and peri-urban environments. The performance of the algorithm is assessed in terms of completeness (i.e. proportion of the poles in the test sites that are detected), and correctness (i.e. proportion of the detected features that are actual poles). An average completeness of 92.3%, and a correctness of 83.8% were achieved. Only some pole-like objects occluded by large features, or too close to other objects, remained undetected.

The pole detection algorithm is more general and improves the performance of the existing methods by: (i) Detecting pole-like objects joined to other features or connected to other poles, (ii) it is fully automatic and does not require the use of training data, (iii) no initial assumptions about the relative location of the poles are needed, and (iv) it is independent of the scanning geometry and it only needs the XYZ coordinates of the original points.

Surface detection:

An algorithm for automatic detection of planar and quasi-planar surfaces from MLS datasets is proposed. This method transforms the original point cloud into a line cloud structure with much lower storage requirements. The lines are subsequently grouped, following a set of parallelism and node proximity rules, describing and conforming the surfaces registered by the sensor.

The performance of the algorithm was assessed with reference to 27 surfaces from an MLS dataset. Correctness (i.e. proportion of grouped points that are on the actual surfaces) was higher than 99% in all the surfaces, being 100% in 12 of them. Completeness (i.e. proportion of points on the actual surfaces that are correctly grouped) averaged more than 90%, ranging from 80% to 97% in 26 out of 27 surfaces.

As regards the accomplishment of the initial objectives set for the surface detection algorithm: (i) Planar and non-completely-planar surfaces (i.e. some ruled surfaces) are successfully detected by the algorithm. (ii) The method is fully automatic and does not require any training data. (iii) It is not limited to vertical surfaces. (iv) No further data than XYZ coordinates and time stamp are required for the whole process.

Road surface delineation:

An algorithm for the automatic delineation of road edges from MLS datasets was developed and tested. The method is based on the initial transformation of the original point cloud into a structured line cloud. The lines are grouped following a set of geometric criteria based on their parallelism and proximity, and the group containing the lines on the road is subsequently identified.

An initial road edge polyline is obtained from the end nodes of the road line group, which is finally smoothed following a two-stage filtering. The algorithm works using exclusively geometric data (i.e. point coordinates, time stamp, and MLS trajectory), and is able to deal with irregularities and discontinuities on the road surface (e.g. cracks, gravel, or different asphalt patches).

The algorithm was tested on two datasets from RoamerR2 on a 2.1 km stretch of road. Similar results were obtained using both datasets separately: 99% surface correctness (i.e., proportion of detected surface that is within the actual road) and 97% surface completeness (i.e. proportion of the actual road that is detected by the algorithm). If the union of the two datasets is used, both completeness and correctness go to 98%.

The algorithm performance is affected by (i) point density along the scanning line; (ii) possible curvature of the asphalt border; (iii) presence of leveled surfaces of compact terrain on the sides of the road; and (iv) presence of vehicles on the road. The effect of all of them (except for the detection of leveled surfaces) is reduced by using the union of two datasets from the same road section and registered with the MLS moving in opposite directions.

Bibliografía

Aijazi, A. K., Checchin, P., & Trassoudaine, L. (2013). Segmentation based classification of 3D urban point clouds: A super-voxel based approach with evaluation. *Remote Sensing*, *5*(4), 1624-1650.

Alho, P., Vaaja, M., Kukko, A., Kasvi, E., Kurkela, M., Hyyppä, J., Hyyppä, H., & Kaartinen, H. (2011). Mobile laser scanning in fluvial geomorphology: Mapping and change detection of point bars. *Zeitschrift für Geomorphologie*, Supplementary Issues, 55(2), 31-50.

Amann, M.C., Bosch T.L., Marc M.R., Rioux M. (2001) Laser ranging: a critical review of usual techniques for distance measurement. *Optical Engineering*, 40 (1), 10-19

Anderson, J., Martin, M. E., Smith, M. L., Dubayah, R. O., Hofton, M. A., Hyde, P., Peterson, B., Blair, J., & Knox, R. G. (2006). The use of waveform lidar to measure northern temperate mixed conifer and deciduous forest structure in New Hampshire. *Remote Sensing of Environment*, 105(3), 248-261.

Bagrow, L. (2010). *History of cartography*. Transaction Publishers.

Ballard, D. H. (1981). Generalizing the Hough transform to detect arbitrary shapes. *Pattern recognition*, *13*(2), 111-122.

Barber, D., Mills, J., & Smith-Voysey, S. (2008). Geometric validation of a ground-based mobile laser scanning system. *ISPRS Journal of Photogrammetry and Remote Sensing*, *63*(1), 128-141.

Blair, J. B., Rabine, D. L., & Hofton, M. A. (1999). The Laser Vegetation Imaging Sensor: a medium-altitude, digitisation-only, airborne laser altimeter for mapping vegetation and topography. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54(2), 115-122.

Boyko, A.; Funkhouser, T. Extracting roads from dense point clouds in large scale urban environment. ISPRS J. Photogramm. Remote Sens. 2011, 66, S2–S12.

Brenner, A. C., Zwally, H. J., Bentley, C. R., Csatho, B. M., Harding, D. J., Hofton, M. A., Minster, J.-B., Roberts, L.A., Saba, J., Thomas, R., & Yi, D. (2003). Derivation of range and range distributions from laser pulse waveform analysis for surface elevations, roughness, slope, and vegetation heights. *Algorithm Theoretical Basis Document*, 4, 26-32.

Bufton, J. L. (1989). Laser altimetry measurements from aircraft and spacecraft. *Proceedings of the IEEE*, 77(3), 463-477.

Carabajal, C. C., & Harding, D. J. (2001). Evaluation of geoscience laser altimeter system (GLAS) waveforms for vegetated landscapes using airborne laser altimeter scanning data. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 34(3/W4), 125-130.

Demantke, J., Mallet, C., David, N., & Vallet, B. (2011). Dimensionality based scale selection in 3D lidar point clouds. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *38*(Part 5), W12.

Douillard, B., Underwood, J., Kuntz, N., Vlaskine, V., Quadros, A., Morton, P., & Frenkel, A. (2011, May). On the segmentation of 3D LIDAR point clouds. In*Robotics and Automation (ICRA), 2011 IEEE International Conference on* (pp. 2798-2805). IEEE.

El-Halawany, S., Moussa, A., Lichti, D. D., & El-Sheimy, N. (2011, August). Detection of road curb from mobile terrestrial laser scanner point cloud. In *Proceedings of the ISPRS Workshop on Laserscanning, Calgary, Canada* (Vol. 2931).

Elseberg, J., Borrmann, D., & Nüchter, A. (2013). One billion points in the cloud–an octree for efficient processing of 3D laser scans. *ISPRS Journal of Photogrammetry and Remote Sensing*, *76*, 76-88.

Falkowski, M. J., Smith, A. M., Hudak, A. T., Gessler, P. E., Vierling, L. A., & Crookston, N. L. (2006). Automated estimation of individual conifer tree height and crown diameter via two-dimensional spatial wavelet analysis of lidar data.*Canadian Journal of Remote Sensing*, *32*(2), 153-161.

Fan, H., Yao, W., & Tang, L. (2014). Identifying man-made objects along urban road corridors from mobile LiDAR data. *IEEE Geoscience and Remote Sensing Letters*, *11*(5), 950-954.

Fang, H. T., & Huang, D. S. (2004). Noise reduction in lidar signal based on discrete wavelet transform. *Optics Communications*, *233*(1), 67-76.

Fang, L., Yang, B., Chen, C., & Fu, H. (2015, July). Extraction 3D road boundaries from mobile laser scanning point clouds. In *Spatial Data Mining and Geographical Knowledge Services (ICSDM), 2015 2nd IEEE International Conference on* (pp. 162-165). IEEE.

Fernandez, J. C., Singhania, A., Caceres, J., Slatton, K. C., Starek, M., & Kumar, R. (2007). An overview of lidar point cloud processing software. *GEM Center Report No. Rep_2007-12-001, University of Florida*.

Flood, M., & Gutelius, B. (1997). Commercial Implications of Topographic Terrain Mapping Using Scanning Airborne Laser Radar. Photogrammetric Engineering and Remote Sensing, 63(4), 327-332.

Fowler, A., & Kadatskiy, V. (2011, May). Accuracy and error assessment of terrestrial, mobile and airborne lidar. In *Proceedings of American Society of Photogrammetry and Remote Sensing Conference (ASPRP 2011), Milwaukee, WI, USA* (Vol. 15).

Glennie, C., & Lichti, D. D. (2010). Static calibration and analysis of the Velodyne HDL-64E S2 for high accuracy mobile scanning. *Remote Sensing*, *2*(6), 1610-1624.

Golovinskiy, A., Kim, V. G., & Funkhouser, T. (2009, September). Shape-based recognition of 3D point clouds in urban environments. In 2009 IEEE 12th International Conference on Computer Vision (pp. 2154-2161). IEEE.

Gräfe, G. (2007). High precision kinematic surveying with laser scanners. *Journal of Applied Geodesy jag*, 1(4), 185-199.

Gräfe, G. (2007). Kinematic surveying with static accuracy. In 8th Conference on Optical 3-D Measurement Techniques (pp. 142-149).

Guan, H., Li, J., & Yu, Y., (2013). 3D urban mapping using a Trimble MX8 mobile laser scanning system: a validation study. In: MMT 2013, 1–3 May 2013.

Guan, H., Li, J., Yu, Y., Wang, C., Chapman, M., & Yang, B. (2014). Using mobile laser scanning data for automated extraction of road markings. *ISPRS Journal of Photogrammetry and Remote Sensing*, *87*, 93-107.

Guan, H., Li, J., Cao, S., & Yu, Y. (2016). Use of mobile LiDAR in road information inventory: A review. *International Journal of Image and Data Fusion*, *7*(3), 219-242.

Harding, D. J., Lefsky, M. A., Parker, G. G., & Blair, J. B. (2001). Laser altimeter canopy height profiles: Methods and validation for closed-canopy, broadleaf forests. *Remote Sensing of Environment*, 76(3), 283-297.

Heritage, G., & Large, A. (Eds.). (2009). *Laser scanning for the environmental sciences*. John Wiley & Sons.

Hosoi, F., & Omasa, K. (2006). Voxel-based 3-D modeling of individual trees for estimating leaf area density using high-resolution portable scanning lidar. *IEEE transactions on geoscience and remote sensing*, *44*(12), 3610-3618.

Hough. P. (1962). *U.S. Patent No. 3,069,654*. Washington, DC: U.S. Patent and Trademark Office.

Huang, J., Menq, C.-H. (2001) Automatic data segmentation for geometric feature extraction from unorganized 3-D coordinate points. *IEEE Transactions on Robotics and Automation*, 17 (3), pp. 268-279.

Hyde, P., Dubayah, R., Peterson, B., Blair, J. B., Hofton, M., Hunsaker, C., Knox, R., & Walker, W. (2005). Mapping forest structure for wildlife habitat analysis using waveform lidar: Validation of montane ecosystems. *Remote sensing of environment*, 96(3), 427-437.

Hyyppä, J., Hyyppä, H., Leckie, D., Gougeon, F., Yu, X., & 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., Jaakkola, A., Chen, Y, & Kukko, A. (2013, September). Unconventional LIDAR mapping from air, terrestrial and mobile. In *Proceedings of the Photogrammetric Week*(pp. 205-214).

Jaakkola, A., Hyyppä, J., Hyyppä, H., & Kukko, A. (2008). Retrieval algorithms for road surface modelling using laser-based mobile mapping. *Sensors*, *8*(9), 5238-5249.

Jacob, C., & Dahl, E. H. (2006). *The sovereign map: Theoretical approaches in cartography throughout history*. University of Chicago Press.

Jochem, A., Höfle, B., & Rutzinger, M. (2011). Extraction of vertical walls from mobile laser scanning data for solar potential assessment. *Remote sensing*, *3*(4), 650-667.

Jutzi, B., Eberle, B., & Stilla, U. (2003). Estimation and measurement of backscattered signals from pulsed laser radar. In International Symposium on Remote Sensing (pp. 256-267). International Society for Optics and Photonics.

Kikuta, H., Iwata, K., & Nagata, R. (1986). Distance measurement by the wavelength shift of laser diode light. *Applied optics*, 25(17), 2976-2980.

Koláčný, A. (2013). Cartographic information—a fundamental concept and term in modern cartography. *The cartographic journal*.

Kremer, J., & Hunter, G. (2007). Performance of the streetmapper mobile lidar mapping system in real world projects. In Photogrammetric week (Vol. 7, No. 2007, pp. 215-225).

Kruth, J. P., & Kerstens, A. (1998). Reverse engineering modelling of free-form surfaces from point clouds subject to boundary conditions. *Journal of Materials Processing Technology*, *76*(1), 120-127.

Kukko, A. (2009). Road environment mapper—3D data capturing with mobile mapping. Licentiate's Thesis, Helsinki University of Technology, Espoo, Finland.

Kukko, A., & Hyyppä, J. (2007). Laser scanner simulator for system analysis and algorithm development: A case with forest measurements. In ISPRS Workshop on Laser Scanning (pp. 234-240).

Kukko, A., Kaartinen, H., Hyyppä, J., & Chen, Y. (2012). Multiplatform mobile laser scanning: Usability and performance. *Sensors*, *12*(9), 11712-11733.

Kumar, P., McElhinney, C. P., Lewis, P., & McCarthy, T. (2014). Automated road markings extraction from mobile laser scanning data. *International Journal of Applied Earth Observation and Geoinformation*, *32*, 125-137.

Kumar, P., Lewis, P., McElhinney, C. P., Boguslawski, P., & McCarthy, T. (2016). Snake Energy Analysis and Result Validation for a Mobile Laser Scanning Data-Based Automated Road Edge Extraction Algorithm.

Lari, Z., & Habib, A. (2014). An adaptive approach for the segmentation and extraction of planar and linear/cylindrical features from laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing*, *93*, 192-212.

Lefsky, M. A., Cohen, W. B., Acker, S. A., Spies, T. A., Parker, G. G., & Harding, D. (1999). Lidar remote sensing of biophysical properties and canopy structure of forest of Douglas-fir and western hemlock. *Remote Sensing of Environment*, 70(3), 339-361.

Lehtomäki, M., Jaakkola, A., Hyyppä, J., Kukko, A., & Kaartinen, H. (2010). Detection of vertical pole-like objects in a road environment using vehicle-based laser scanning data. *Remote Sensing*, *2*(3), 641-664.

Lehtomäki, M., Jaakkola, A., Hyyppä, J., Kukko, A., & Kaartinen, H. (2011). Performance analysis of a pole and tree trunk detection method for mobile laser scanning data. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci*, *38*, 197-202.

Lehtomäki, M., Jaakkola, A., Hyyppä, J., Lampinen, J., Kaartinen, H., Kukko, A., & Hyyppä, H. (2016). Object classification and recognition from mobile laser scanning point clouds in a road environment. *IEEE Transactions on Geoscience and Remote Sensing*, 54(2), 1226-1239.

Leonardis, A., Gupta, A., & Bajcsy, R. (1995). Segmentation of range images as the search for geometric parametric models. *International Journal of Computer Vision*, *14*(3), 253-277.

Liu, G. H., Wong, Y. S., Zhang, Y. F., & Loh, H. T. (2003). Error-based segmentation of cloud data for direct rapid prototyping. *Computer-Aided Design*, *35*(7), 633-645.

Mallet, C., & Bretar, F. (2009). Full-waveform topographic lidar: State-of-the-art. ISPRS Journal of photogrammetry and remote sensing, 64(1), 1-16.

Meagher, D. (1982). Geometric modeling using octree encoding. *Computer graphics and image processing*, *19*(2), 129-147.

Mendenhall, S. (2011). Mobile Laser Scanning. *CE News*, *23*(5).

Morrone, M. C., & Burr, D. C. (1988). Feature detection in human vision: A phase-dependent energy model. *Proceedings of the Royal Society of London B: Biological Sciences*, *235*(1280), 221-245.

Moskal, L. M., & Zheng, G. (2011). Retrieving forest inventory variables with terrestrial laser scanning (TLS) in urban heterogeneous forest. *Remote Sensing*, *4*(1), 1-20.

Nelson, R., Krabill, W., & MacLean, G. (1984). Determining forest canopy characteristics using airborne laser data. *Remote Sensing of Environment*, 15(3), 201-212.

Nuchter, A., Lingemann, K., & Hertzberg, J. (2007, August). Cached kd tree search for ICP algorithms. In *3-D Digital Imaging and Modeling, 2007. 3DIM'07. Sixth International Conference on* (pp. 419-426). IEEE.

Palojärvi, P. (2003). Integrated electronic and optoelectronic circuits and devices for pulsed time-of-flight laser rangefinding. Ph.D. dissertation, University of Oulun, 56p.

Paschotta, R. (2008). Encyclopedia of laser physics and technology (Vol. 1). Berlin: Wiley-vch.

Pu, S., Rutzinger, M., Vosselman, G., & Elberink, S. O. (2011). Recognizing basic structures from mobile laser scanning data for road inventory studies.*ISPRS Journal of Photogrammetry and Remote Sensing*, *66*(6), S28-S39.

Puente, I., González-Jorge, H., Arias, P., & Armesto, J. (2011). Land-based mobile laser scanning systems: A review. In: Lichti, D.D. & Habib, A.F. (Eds.) ISPRS Workshop Laser Scanning, Calgary, Canada. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 163-168.

Puente, I., González-Jorge, H., Martínez-Sánchez, J., & Arias, P. (2013). Review of mobile mapping and surveying technologies. *Measurement*, *46*(7), 2127-2145.

Puente, I., González-Jorge, H., Riveiro, B., & Arias, P. (2013). Accuracy verification of the Lynx Mobile Mapper system. *Optics & Laser Technology*, *45*, 578-586.

Rabbani, T., & Van Den Heuvel, F. (2005). Efficient hough transform for automatic detection of cylinders in point clouds. *ISPRS WG III/3, III/4*, *3*, 60-65.

Rabbani, T., Van Den Heuvel, F., & Vosselmann, G. (2006). Segmentation of point clouds using smoothness constraint. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *36*(5), 248-253.

Raguram, R., Frahm, J. M., & Pollefeys, M. (2008, October). A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus. In *European Conference on Computer Vision* (pp. 500-513). Springer Berlin Heidelberg.

Riegl. www.riegl.com. (Accessed1 November, 2016).

Rodriguez-Cuenca, B., Garcia-Cortes, S., Ordóñez, C., & Alonso, M. C. (2015). An approach to detect and delineate street curbs from MLS 3D point cloud data. *Automation in Construction*, *51*, 103-112.

Roggero, M. (2002). Object segmentation with region growing and principal component analysis. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 34(3/A), 289-294.

Rutzinger, M., Elberink, S. O., Pu, S., & Vosselman, G. (2009). Automatic extraction of vertical walls from mobile and airborne laser scanning data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *38*(Part 3), W8.

Ruyi, J., Reinhard, K., Tobi, V., & Shigang, W. (2011). Lane detection and tracking using a new lane model and distance transform. *Machine vision and applications*, *22*(4), 721-737.

Sarkar, B., & Menq, C. H. (1991). Smooth-surface approximation and reverse engineering. *Computer-Aided Design*, *23*(9), 623-628.

Schawlow, A. L., & Townes, C. H. (1958). Infrared and optical masers. *Physical Review*, 112(6), 1940-1949.

Schnabel, R., Wahl, R., & Klein, R. (2007, June). Efficient RANSAC for point-cloud shape detection. In *Computer graphics forum* (Vol. 26, No. 2, pp. 214-226). Blackwell Publishing Ltd.

Schnabel, R., Wessel, R., Wahl, R., & Klein, R. (2008, February). Shape recognition in 3d point-clouds. In *The 16-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision* (Vol. 8).

Schott, J. R. (2007). *Remote sensing*. Oxford University Press.

Schreier, H., Lougheed, J., Tucker, C., & Leckie, D. (1985). Automated measurements of terrain reflection and height variations using an airborne infrared laser system. International Journal of Remote Sensing, 6(1), 101-113.

Serna, A.; Marcotegui, B. (2006) Urban accessibility diagnosis from mobile laser scanning data. G. Sun, Z., Bebis, G., & Miller, R.. On-road vehicle detection: A review. *IEEE transactions on pattern analysis and machine intelligence*, *28*(5), 694-711.

Sithole, G., & Vosselman, G. (2003, May). Automatic structure detection in a point-cloud of an urban landscape. In *Remote Sensing and Data Fusion over Urban Areas, 2003. 2nd GRSS/ISPRS Joint Workshop on* (pp. 67-71). IEEE.

Sithole, G., & Vosselman, G. (2004). Experimental comparison of filter algorithms for bare-Earth extraction from airborne laser scanning point clouds.*ISPRS journal of photogrammetry and remote sensing*, *59*(1), 85-101.

Smadja, L., Ninot, J., & Gavrilovic, T. (2010). Road extraction and environment interpretation from Lidar sensors. *IAPRS*, *38*, 281-286.

Solodukhin, V., Zukov, A., & Mazugin, I. (1977). Possibilities of laser aerial photography for forest profiling. *Lesnoe Khozyaisto* (Forest Management), 10, 53-58.

Talaya, J., Bosch, E., Alamus, R., Bosch, E., Serra, A., & Baron, A., (2004) GEOMOBIL: the mobile mapping system from the ICC. Proceedings of 4th International Symposium on Mobile Mapping Technology, Kinming, China.

Tarsha-Kurdi, F., Landes, T., & Grussenmeyer, P. (2007, September). Hough-transform and extended ransac algorithms for automatic detection of 3d building roof planes from lidar data. In *Proceedings of the ISPRS Workshop on Laser Scanning* (Vol. 36, pp. 407-412).

Taylor, D. R. F. (2013). Cartography for knowledge, action and development: retrospective and prospective. *The Cartographic Journal*.

Teledyne Optech. < http://www.teledyneoptech.com/>. (Accessed 1 Novenber, 2016).

Truong-Hong, L., Laefer, D. F., Hinks, T., & Carr, H. (2013). Combining an angle criterion with voxelization and the flying voxel method in reconstructing building models from LiDAR data. *Computer-Aided Civil and Infrastructure Engineering*, *28*(2), 112-129.

Vanco, M., Brunnett, G., & Schreiber, T. (2000). A direct approach towards automatic surface segmentation of unorganized 3D points. In *Proceedings spring conference on computer graphics* (pp. 185-194).

Vanderhyde, J., & Szymczak, A. (2008). Topological simplification of isosurfaces in volumetric data using octrees. *Graphical Models*, *70*(1), 16-31.

Vosselman, G., & Maas, H. G. (Eds.). (2010). *Airborne and terrestrial laser scanning*. Whittles Publishing.

Wagner, W., Ullrich, A., Melzer, T., Briese, C., & Kraus, K. (2004). From single-pulse to full-waveform airborne laser scanners: potential and practical challenges. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 35, B3, 201-206.

Wagner, W., Ullrich, A., Ducic, V., Melzer, T., & Studnicka, N. (2006). Gaussian decomposition and calibration of a novel small-footprint full-waveform digitising airborne laser scanner. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(2), 100-112.

Wehr, A., & Lohr, U. (1999). Airborne laser scanning—an introduction and overview. *ISPRS Journal of photogrammetry and remote sensing*, *54*(2), 68-82.

Woodford, O. J., Pham, M. T., Maki, A., Perbet, F., & Stenger, B. (2014). Demisting the Hough transform for 3D shape recognition and registration. *International Journal of Computer Vision*, *106*(3), 332-341.

Wu, B., Yu, B., Yue, W., Shu, S., Tan, W., Hu, C., ... & Liu, H. (2013). A voxel-based method for automated identification and morphological parameters estimation of individual street trees from mobile laser scanning data. *Remote Sensing*, *5*(2), 584-611.

Wu, B., Yu, B., Huang, C., Wu, Q., & Wu, J. (2016). Automated extraction of ground surface along urban roads from mobile laser scanning point clouds. *Remote Sensing Letters*, *7*(2), 170-179.

Yang, B., Fang, L., & Li, J. (2013). Semi-automated extraction and delineation of 3D roads of street scene from mobile laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, *79*, 80-93.

Yokoyama, H., Date, H., Kanai, S., & Takeda, H., 2011. Pole-like objects recognition from Mobile Laser Scanning Data using smoothing and principal component analysis. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 38 (Part 5/W12),* 115–120.

Yu, Y., Li, J., Guan, H., Jia, F., & Wang, C. (2015). Learning hierarchical features for automated extraction of road markings from 3-D mobile LIDAR point clouds. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *8*(2), 709-726.

Zhao, H., & Shibasaki, R. (2002). Surface Modelling of Urban 3D Objects from Vehicle-Borne Laser Range Data. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, *34*(3/A), 412-417.

Zhao, G., & Yuan, J. (2012, September). Curb detection and tracking using 3D-LIDAR scanner. In *2012 19th IEEE International Conference on Image Processing* (pp. 437-440). IEEE.

Abreviaturas

LiDAR -> Light Detection and Ranging

LS -> *Laser Scanning*

ALS -> *Aerial Laser Scanner*

TLS -> *Terrestrial Laser Scanner*

MLS -> *Mobile Laser Scanner*

RANSAC -> *RANdom SAmple Consensus*

LASER -> *Light Amplification by Simulated Emission of Radiation*

ToF -> *Time of Flight*

PS -> *Phase Shift*

FOV -> *Field of View*

MMS -> *Mobile Mapping System*

GNSS -> *Global Navigation Satellite System*

IMU -> *Inertial Measurement Unit*

ACP *-> Análisis de Componentes Principales*

RTK -> *Real Time Kinematic*

Financiación

Esta tesis doctoral ha sido financiada por el programa Severo Ochoa de ayudas predoctorales para formación en investigación y docencia del Plan de Ciencia, Tecnología e Innovación del Principado de Asturias (PCTI) (Referencia BP12-163).

La estancia breve de investigación en el *Finnish Geodetic Institute* (Kirkkonummi, Finlandia), entre septiembre y diciembre de 2014, fue financiada por las ‘Ayudas para estancias breves’ de este mismo programa.

Factor de Impacto de las Publicaciones

Publication 1

Cabo, C., Ordoñez, C., García-Cortés, S., & Martínez, J. (2014). An algorithm for automatic detection of pole-like street furniture objects from Mobile Laser Scanner point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, *87*, 47-56.

Impact Factor 2015:

- IF: 4.188
- Category: Remote Sensing
- Ranking: Q1 (2/28)

Publication 2

Cabo, C., Cortés, S. G., & Ordoñez, C. (2015). Mobile Laser Scanner data for automatic surface detection based on line arrangement. *Automation in Construction*, *58*, 28-37.

Impact Factor 2015:

- IF: 2.442
- Category: Civil Engineering
- Ranking: Q1 (12/126)

Publication 3

Cabo, C., Kukko, A., García-Cortés, S., Kaartinen, H., Hyyppä, J., & Ordoñez, C. (2016). An Algorithm for Automatic Road Asphalt Edge Delineation from Mobile Laser Scanner Data Using the Line Clouds Concept. *Remote Sensing*, *8*(9), 740.

Impact Factor 2015:

- IF: 3.036
- Category: Remote Sensing
- Ranking: Q1 (5/28)