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

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).

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.

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

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.

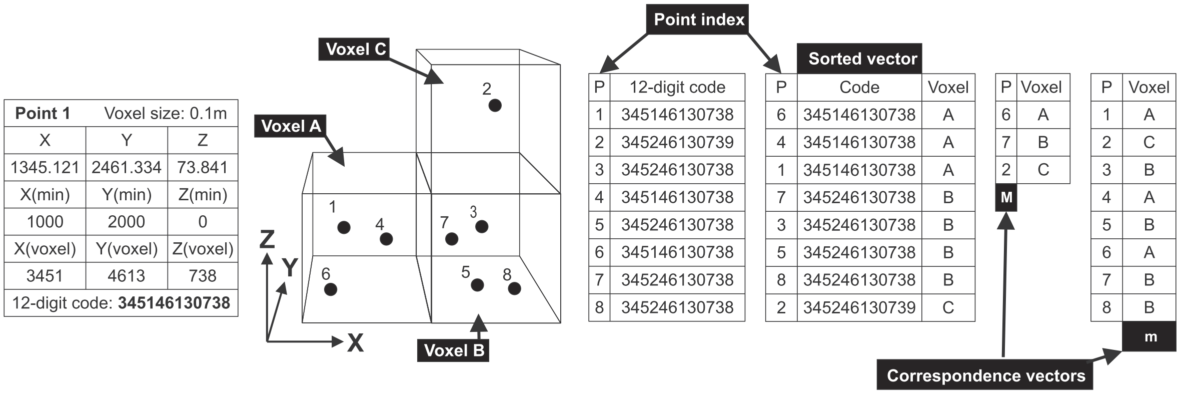
2.2.1. Segmentation of the connected horizontal elements

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.

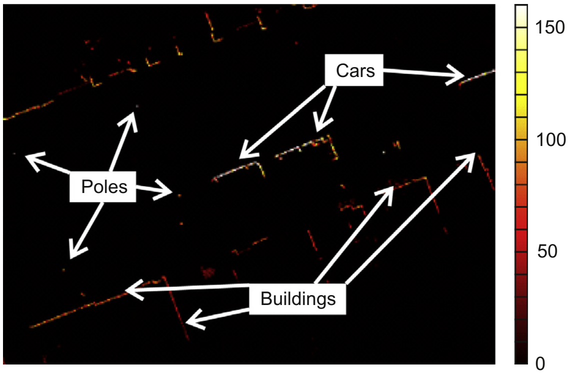


Fig. 2. 2D horizontal section of the tridimensional grid. Color represents the number of points stored in each voxel. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

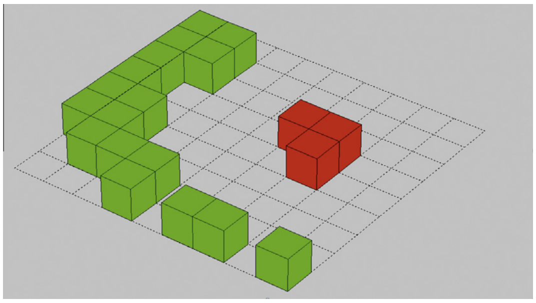


Fig. 3. Segmentation of horizontal sections by area. Red: voxels from a pole section. Green: voxels from building sections. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

(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

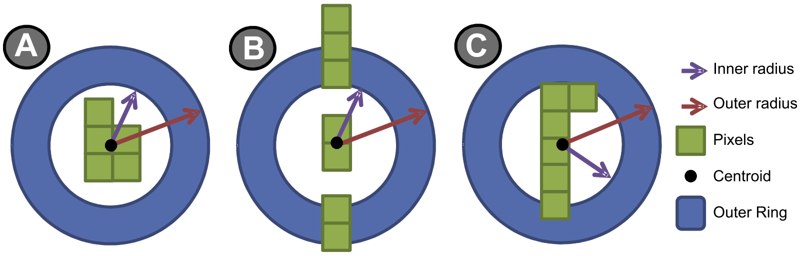


Fig. 4. Diagram of the outer ring search in a horizontal section. Cases: (a) Pole section, (b) a section of a wall with occlusions and (c) false positive detected by the presence of voxels in the ring.

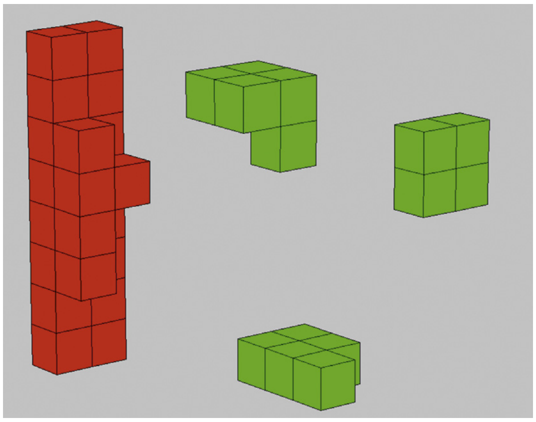


Fig. 5. Vertical union of filtered horizontal sections. Red: Group that exceeds the minimum height (Pole representation). Green: Isolated sections. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

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).

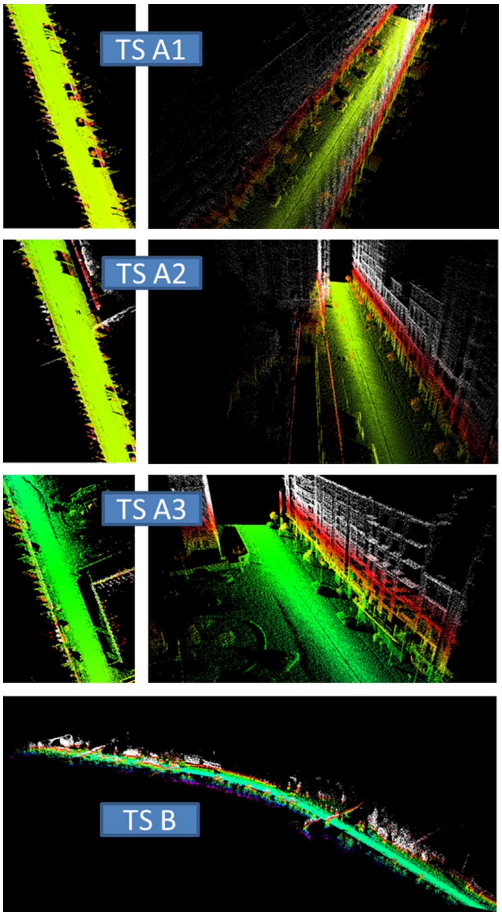


Fig. 6. Footprint and perspective view of the test sites.

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.

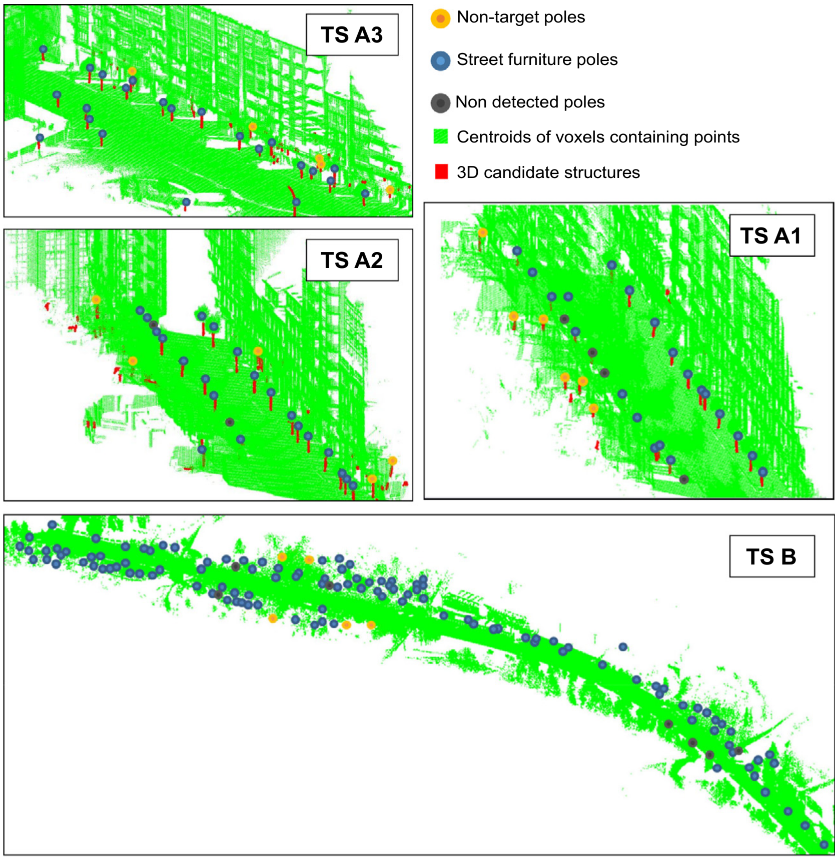


Fig. 7. Results of the application of the algorithm in the four test sites.

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

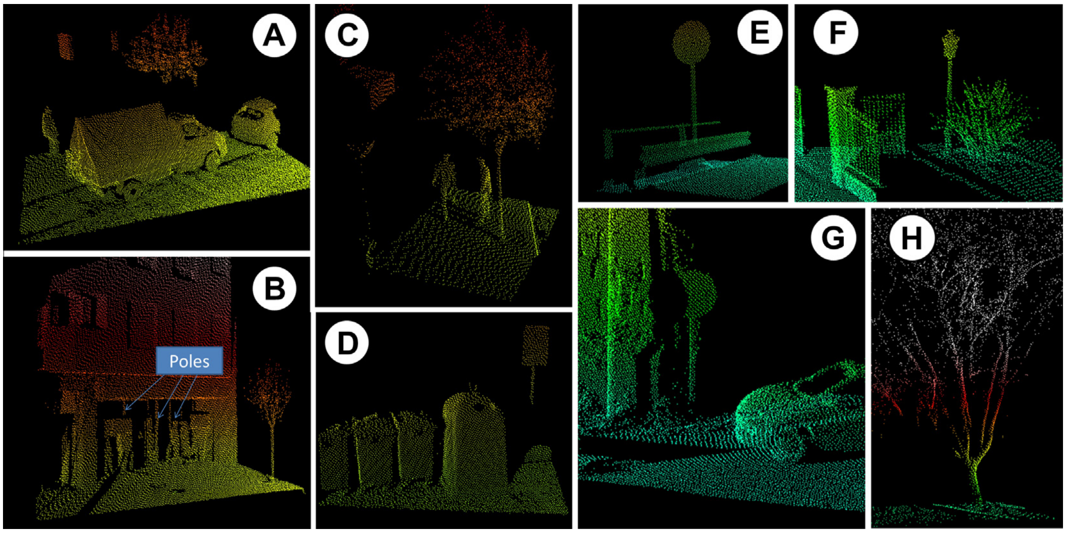


Fig. 8. A: Total occlusion of a pole (TS-1). B: Features detected inside the buildings through big windows in Dataset A. C: Pedestrian close to a tree (TS-A1). D: Partial occlusion from bins (TS-A2). E: Sign post close to the guardrail in TS-B. F: Lamp post surrounded by a bush. G: Sign post close to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

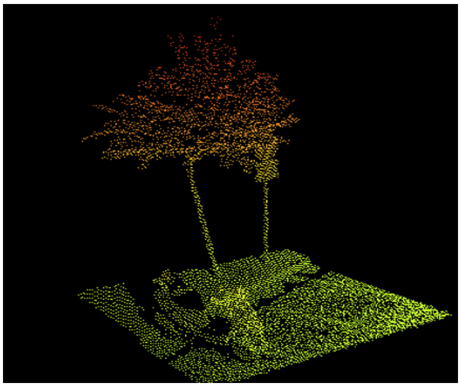


Fig. 9. Slanting pole-like object detected in TS-A1.

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.