Research paper

Automatic extraction of road features in urban environments using dense ALS data

CrossMark

Mario Soilán[a,*], Linh Truong-Hong[b], Belén Riveiro[a], Debra Laefer[b,c]

[a] Department of Materials Engineering, Applied Mechanics and Construction, School of Industrial Engineering, University of Vigo, 36310, Spain
[b] Urban Modelling Group, School of Civil, Structural and Environment Engineering, University College Dublin, Belfield, Dublin 4, Ireland
[c] Urban Modeling Group, Center for Urban Science and Progress, New York University, Brooklyn, New York, 12011, USA

ARTICLE INFO

ABSTRACT

This paper describes a methodology that automatically extracts semantic information from urban ALS data for urban parameterization and road network definition. First, building façades are segmented from the ground surface by combining knowledge-based information with both voxel and raster data. Next, heuristic rules and unsupervised learning are applied to the ground surface data to distinguish sidewalk and pavement points as a means for curb detection. Then radiometric information was employed for road marking extraction. Using high-density ALS data from Dublin, Ireland, this fully automatic workflow was able to generate a F-score close to 95% for pavement and sidewalk identification with a resolution of 20 cm and better than 80% for road marking detection.

1. Introduction

According to Biljecki et al. (2015), a three-dimensional (3D) city model can be used in more than 100 applications, including emergency response planning, infrastructure inventory and maintenance, and virtual tourism. Therefore, building 3D city models that can be efficiently updated and semantically labeled are essential in the context of a smart city. While many effective strategies have been introduced for distinguishing buildings from vegetation and paved areas (e.g. Idrees et al., 2013; Vo, 2016), even in a Big Data context (Aljumaily et al., 2016, 2017), relatively little has been robustly and scalably achieved for the semantic labeling of smaller road features such as curbs and road markings. Yet, these are needed for systematic cataloguing and management. Obtaining these smaller road features is the topic of this paper.

2. Background

Comprehensive documentation of the urban road network is essential for applications such as transportation management and auto-navigation. Broadly speaking, remote sensing based research concerning road network extraction and modelling can be divided in two broad categories based on input data: (1) imagery and (2) Light Detection and Ranging (LiDAR). A wide range of approaches has been applied to extract road surfaces (Wang et al., 2016) such as

classification-based methods (Kirthika and Mookambiga, 2011), knowledge-based methods (Shen et al., 2008), mathematical morphology (Barrile and Bilotta, 2016; Alshehhi and Marpu, 2017) and active contour models (Anil and Natarajan, 2010). These methods may document either the complete road network (Poullis, 2014) or different topographical parameters of road axes, such as road centerlines (Miao et al., 2013) and horizontal alignment (Holgado-Barco et al., 2015). The rapid growing popularity of imagery obtained from unmanned aerial vehicles (Chen et al., 2016a, 2016b, 2016c) is fueling such approaches. While imagery is inexpensive, deriving 3D models requires intermediary steps that may be problematic in urban areas because of parallax problems (e.g. Byrne et al., 2017b) or prove computationally expensive with reduced accuracy (Byrne et al., 2017a). As such, LiDAR (a laser-based technology) offers a cost-effective alternative data stream.

When deployed on a mobile platform, laser scanners are able to gather 3D road network data with spatial resolution of thousands of points per square meter (Puente et al., 2013), for model reconstruction (Yang et al., 2013; Chen and He, 2015), curb mapping (Kumar et al., 2017; Rodríguez-Cuenca et al., 2015; Zhou and Vosselman, 2012) or network asset management via classification and subsequent inventorying of features such as traffic signs (Yu et al., 2016) or road markings (Yu et al., 2015). The procedure adopted for the traffic signage involved reflectivity based clustering for extraction and then bag-of-visual-phrases representation and a deep Boltzmann machine feature encoder (Yu et al., 2016). For road markings, heuristic decisions and

deep learning model were used for road marking identification (Yu et al., 2015). Subsequently, Yang et al. (2017) classified road facilities such as street lamps, guardrails or power lines from mobile laser scanner data by combining features at different scales (point, segment and object scales) with contextual features. The high spatial density of the 3D data (averaging 2000 pts/m² on the road surface) allowed for the detection and classification of small road objects but at a high cost with respect to data collection, processing, and storage.

In contrast, ALS systems acquire less dense 3D information (up to only a few hundred points per square meter, Vo et al., 2015a, 2015b) and are limited by the nadir orientation of the equipment (Hinks et al., 2009). From such data Zhao et al. (2011) extracted road centerlines from road images generated from airborne LiDAR data by assuming that point intensity satisfied a multivariate Gaussian distribution and dividing terrain objects into 4 classes. Points that belonged to the road class were selected manually, road segments were found using a radius-rotating intersection detection method, and road centerlines were fitted using total least squares. Subsequently, Hu et al., (2014) successfully developed a road centerline detection methodology by first separating ground and non-ground points and then using spatial clustering using a mean shift to detect road center points, with tensor voting to enhance linear features and a Hough transform to defines centerline primitives. To the same end, Hui et al. (2016) employed skewness balancing (for intensity-based filtering), a rotating neighborhood (to remove narrow roads), and hierarchical fusion and optimization (for elimination of parking lots and bare earth). Instead of extracting only the road centerline, Zhu and Mordohai (2009) extracted the whole road network by first detecting multiple, horizontally close, ground planes, and subsequently projecting ground points intensities onto to 2D images. They extracted road features on that image using boundary and interior features and then generated hypotheses from those features. The combination of the hypotheses generated a road likelihood map that was further used to find a set of road segments via a minimum cover algorithm.

Although ALS technology is constantly improving, there exist some limitations such as the low density in comparison with TLS systems, thus small elements of the road network may be missing, hindering progress towards road asset inventory applications. With more efficient flight path planning and a lower flight altitude, this drawback can be partially alleviated (He et al., 2017). This paper explores the possibility of using dense ALS data for extracting different elements of the urban road layout that are typically assessed using terrestrial equipment. The fact that terrestrial surveys aiming to cover large urban areas take longer and need more data storage capacity than an aerial survey motivates this work, which explores the possibility of using dense ALS data for extracting different elements of the urban road layout that are typically assessed using terrestrial equipment. Specifically, the main parts of the road (curbs, sidewalk and pavement) and road markings. The main contributions of this paper are the usage of ALS data for (1) Extracting a curb map that allows the classification of sidewalk and pavement area, and (2) Detecting road markings on the ground relying on the reflective properties of the painting materials.

3. Methodology

The proposed method aims to extract urban ground-level, semantic information in the form of road edges, sidewalks, and road markings from ALS data through five steps (Fig. 1): (1) point cloud preprocessing, (2) elevation-based segmentation, (3) ground extraction, (4) road curb extraction, and (5) Road features detection.

3.1. Step 1. point cloud preprocessing

Let $P_{raw} = (x, y, z, I, t)$ be a raw three-dimensional (3D) point cloud, where (x, y, z) are the 3D coordinates acquired by the positioning system representing a set of points in space, and (I, t) are,

respectively, the intensity value and the time stamp for each 3D point. First, intensity values are normalized to be proportional to the surface reflectance. This enables fusing multiple ALS datasets (Kaasalainen et al., 2005; Vain et al., 2010), when necessary. According to Höfle and Pfeifer (2007) (Eq. (1)), the reflectance depends on a reference range (R_s), the distance between the laser scanner and the object (R), the angle of incidence (a − angle between surface normal and laser beam −), the instrument, and atmospheric conditions with respect to attenuation η_{atm}:

$$I_{norm} = I \frac{R^2}{R_s} \frac{1}{\eta_{atm} \cos \alpha} \tag{1}$$

A factor not explicitly addressed in this equation is the equipment. Herein, the trajectory data were used to compute the range and incidence angle for each point in the point cloud. The normalized range was set to 300 m, and the atmospheric attenuation was neglected since there was only a single flight conducted over only 4 h (see Section 4.1 for dataset information). As each scanned area is measured from multiple angles, this normalization homogenizes the intensity values of the raw point cloud (Fig. 2a).

Next, a saliency analysis was conducted on the point cloud $P_n = (x, y, z, I_{norm}, t)$ to segment points whose normal vectors are close to the z-axis (i.e. belonging to a horizontal surface) from those who are not. For this, an algorithm by Wang et al. (2015) was modified. Instead of using the distance between the normal vector of each point and a dominant normal vector to compute a saliency map, the following modifications were made:

- Normal vectors were obtained within a voxel space using a cubic cell grid (Soilán et al., 2017). For each voxel, a single point was defined as the centroid of the points in the cell, thereby creating a de facto downsampling of the data. This accelerates calculation of normal vectors and reduces noise.
- The k-means algorithm (Lloyd, 1982) was modified by changing the cluster selection constraint to select the dominant normal vector. Instead of taking the largest cluster of five to define the dominant normal vector, the centroid of the cluster closest to the z-axis is used for the ground plane.

The saliency map is computed by projecting the distance between each normal vector and the dominant normal vector onto a hyperbolic tangent function space. This allows separation of salient and non-salient points (Wang et al., 2015), Fig. 2b. Façades, poles, and lateral sides of vehicles contain salient points, while roofs, balconies, and ground segments do not. The saliency analysis returns the point indices i_s and i_{ns}, for point positions within P_{norm}. Let $\mathscr{S}(P, i)$ be a function that selects a subset of points with indices i in a 3D point cloud P. The salient and non-salient point clouds can be computed as $P_s = \mathscr{S}(P_n, i_s)$ and $P_{ns} = \mathscr{S}(P_n, i_{ns})$, respectively. To save computational resources, only the indices are stored, applying $S(P, i)$ when it is needed.

3.2. Step 2. elevation-based segmentation

Saliency analysis facilitates façade extraction using a height filter within a 2D raster image, where $P_s = \mathscr{S}(P_n, i_s)$ the salient point cloud and $\mathscr{R}(P, g_r)$ a function that computes a 2D raster structure, which is created defining a square cell grid with size g_r in the XY plane and projecting the point cloud P onto it. The raster structure contains the following information:

- Index of every cell in the raster grid that contains at least 1 point; given an NxM grid, the cell in a position $(x, y)|x, y \rangle 0, x \leq N, y \leq M$ is defined as $y + M(x - 1)$.
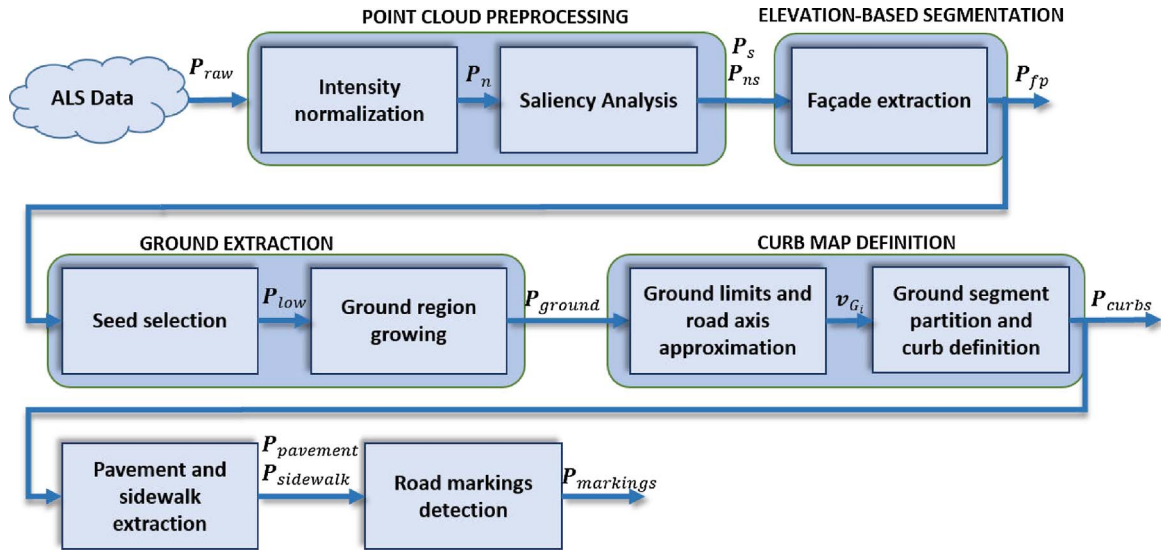- Indices of the points in P within each cell, herein referred to as the cell index.

POINT CLOUD PREPROCESSING   ELEVATION-BASED SEGMENTATION

Fig. 1. Workflow.

Raw Intensity   **Normalized Intensity**
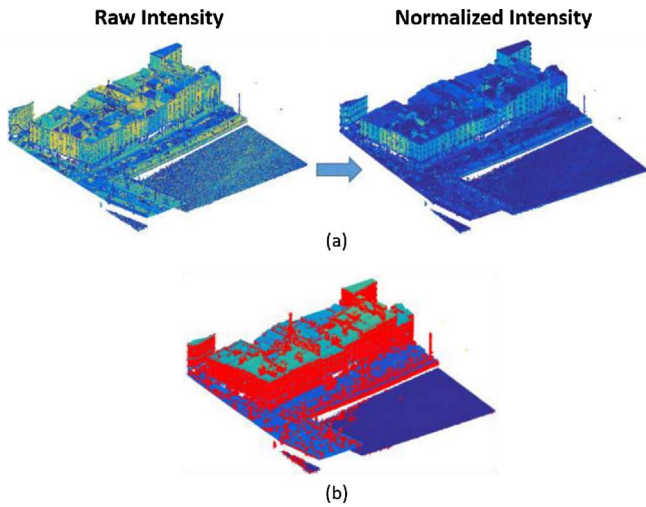

(a)

(b)

Fig. 2. Point cloud preprocessing. (a) Intensity normalization. Point cloud colour based intensity values. Intensity values of the raw point cloud are homogeinized after the normalization process. (b) Saliency analysis. The point cloud is divided using the saliency of each point. P_s, point cloud of salient points, is colored in red while P_{ns}, point cloud of non-salient points, is colored in different tones of blue according to the height of the point. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

- Intensity based image stored in a 2D array, defined as the average point intensity of each cell; visualizable as a grayscale image.
- Height based image stored in a 2D array, defined as the maximum height difference of points within each cell.
- Density based image stored in a 2D array, defined as the number of points in each cell.

A 2D raster structure $R_{ns} = \mathscr{R}(P_{ns}, g_{r1})$ is computed, using the height based image (Fig. 3a) for façade extraction by binarizing it using Otsu method (Otsu, 1979), that is, assuming that the image has two classes of pixels and calculating a binarization threshold that maximizes the inter-class variance. Then, the indices of pixels, i_{fp}, which are set to "true" in the binary image, are used to select the points in P_{ns} belonging to vertical structures (e.g. façades and poles). Subsequently, a Euclidean distance clustering is performed to organize the points by grouping nearby points together. Next, a contextual merging process connects points of the same façade that belong to different clusters due to oc-clusions and openings. If the points of a cluster share all their raster

indices with another cluster, then both clusters are assumed to belong to the same vertical structure and are, thus, merged to finally become a set of clusters $\mathscr{C}_{fp} = \{i^1_{fp}, i^2_{fp} \ldots i^n_{fp}\}|i^i_{fp} \subset i_{fp}$. The segmented points of the entire point cloud can be computed as $P_{fp} = \mathscr{S}(P_s, i_{fp})$, while those of each individual cluster as $P^i_{fp} = \mathscr{S}(P_{fp}, i^i_{fp})$ (Fig. 3b).

3.3. Step 3. ground extraction

Ground extraction was conducted based on a voxel grid-based seg-mentation in which the ground segment is obtained via region growing (as per Douillard et al., 2011), where adjacent voxels whose vertical mean and variance differences are less than certain thresholds (em-pirically defined as $d_\mu = 7.5 cm$ and $d_\sigma = 0.05$), and seed points are selected from within P_{ns} relying on contextual knowledge, as lower points in P_{fp} represent the bottoms of façades and pole-like structures, which are assumed to connect with the ground.

Thus, the point with the minimum z-coordinate for each cell com-puted within the raster structure $R_{fp} = \mathscr{R}(P_{fp}, g_{r1})$ is stored. The z co-ordinates of these selected points are then gathered in a 50-bin histo-gram. The largest bin b_i and its two adjacent bins (b_{i-1}, b_{i+1}) are selected, and points with indices i_{seed} with z coordinates between the limits of the selected histogram bins are extracted. This ensures selec-tion of points in P_{fp} that connect with the ground (some points may be poor candidates due to ground-level occlusions). Finally, a KNN algo-rithm is applied to find the five nearest neighbors (as established em-pirically) in P_{ns} for each point of $P_{low} = \mathscr{S}(P_{fp}, i_{seed})$, thereby obtaining a group of seed points (Fig. 4a).

Let $\mathscr{V}(P, g_v)$ be a function that computes a cubic voxel grid struc-ture with grid size g_v for a point cloud P. The voxel structure $V_{ns} = \mathscr{V}(P_{ns}, g_{v1})$ is computed, and voxels containing seed points are selected as per the aforementioned ground segmentation method.

The resulting voxel indices relating to the ground surface can be transformed into a set of point indices i_{ground}. The ground point cloud (Fig. 4b) can be extracted as $P_{ground} = \mathscr{S}(P_{ns}, i_{ground})$.

3.4. Step 4. road curb extraction

Urban ground typically consists of pavement (road surface) and sidewalks separated by curbs, which form the road edge. However, the separation is not necessarily continuous because of ramps and drive-ways, which preclude simply using a change in elevation to establish the road edge. Instead, the proposed curb extraction algorithm has four main steps: (1) finding ground limits through a raster structure, (2)
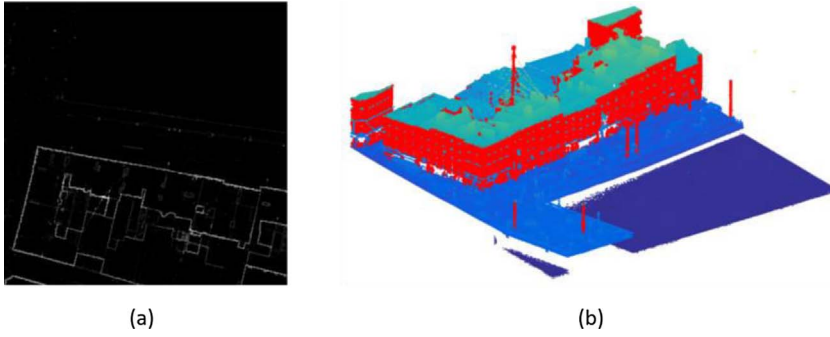
Fig. 3. Elevation-based segmentation. (a) height based image computed within the raster structure R_{ns} of non-salient point cloud P_{ns}, which highlights building outlines; (b) elevation-based segmentation outputs building façades and vertical, pole-like objects (red). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

approximating the road axis, (3) partitioning the ground segment, and (4) fusing heuristic and supervised learning methods for a segment-wise analysis based on geometrical features.

3.4.1. Step 4a. ground limits detection

The first step is to detect the ground segment limits. Usage of global thresholds to detect elements such as curbs may incur in large error rates because of local geometric differences. For that reason, a local analysis is preferred. This is done by approximating the road axis (assumed to be parallel to the primary road direction) by using the ground limits.

For this, let $P_{ground} = \mathscr{S}(P_{ns}, i_{ground})$ be the ground point cloud and $R_{ground} = \mathscr{R}(P_{ground}, g_{r2})$ a raster structure of P_{ground}. Using the density-based image, an occupancy binary image I_g can be created where cells with at least one point are set to "true" (Fig. 5a). Then, a spherical structuring element with 5 pixels (equivalent to 1 m for $g_{r2} = 0.2m$) of radius, H, is defined. Finally, a set of morphological operations are applied to I_g. A closing operation, $I_{g2} = (I_g \oplus H) \ominus H$, closes holes in the image caused by occlusions in the ground. A morphological erosion, $I_{g3} = (I_{g2} \ominus H)$, trims the ground boundaries, and an XOR operation $I_{limits} = I_{g2} \oplus I_{g3}$ retrieves the edges of the ground. As the correspondence between pixels in I_{limits} and points in P_{ground} is known via R_{ground} properties, which stores point indices for each pixel (i_{limits}), a point cloud with its ground limits can retrieved as $P_{limits} = \mathscr{S}(P_{ground}, i_{limits})$ (Fig. 5b).

3.4.2. Step 4b. road axis approximation

The point cloud P_{limits} is inherently unorganized. To define a set of vectors that approximate locally the road axis, let $\mathscr{L} = \{L_1, L_2, ...L_n\}$, be a set of polylines where each polyline L_i is defined as a group of 3D vertices, $L_i = \{G_1, G_2, ...G_n\} | G_i = (x, y, z)_i$, and v_{G_i} the director vector of the corresponding segment. $\mathscr{L}$ and v_{G_i} are computed following the pseudocode in Algorithm 1. An iterative process select the point in P_{limits} with less neighbors, G_i, to start a polyline, and defines the next vertex G_{i+1} as the further neighbor in a sphere with 5 m of radius. The vector v_{G_i} is then defined as the eigenvector corresponding to the largest eigenvalue after applying Principal Component Analysis (PCA) to
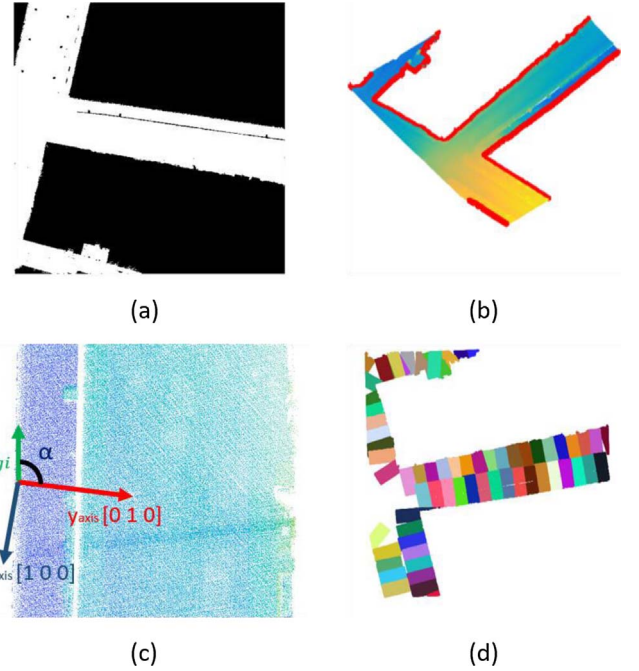

Fig. 5. Ground segment processing. (a) Occupancy binary image computed within the raster structure R_{ground}, (b) Ground limits (coloured in red) used to approximate road axis, (c) P_{ground} transformed by rotating points so road axis v_{gi} is aligned with local y axis (unit vector [0 1 0], which points to the North). (d) Ground partition with each section randomly coloured for visualization. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the points within the defined neighborhood. The point G_{i+1} is then used to find the next vertex, and the process continues iteratively until there are no more neighbors. Then, if there exist points in P_{limits} that have not been used for the previous polylines, the whole process is repeated for a new polyline.
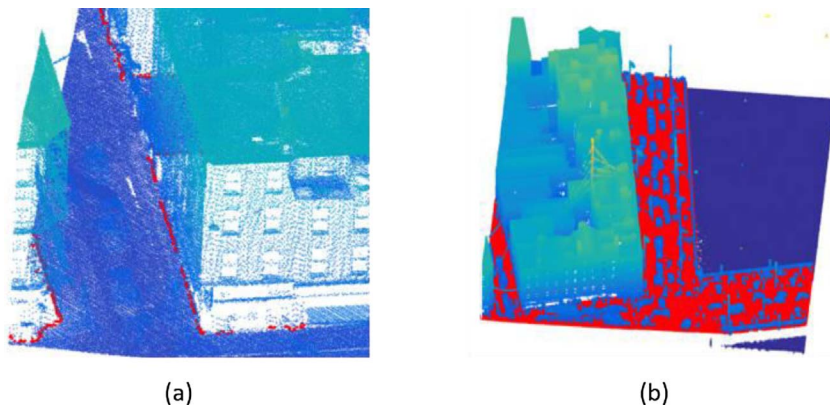

Fig. 4. Ground extraction. (a) Lower points of vertical elements, colored in red, are used for selecting seed points. (b) Extractable ground segment (red) after region growing. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Algorithm 1. Groundlimits definition.

Data: Road limit point cloud $\boldsymbol{P}_{limits}$.
Outputs: lines: A set of polylines (cell array).
         vectors: A set of vectors corresponding to each segment (cell array).
Functions: **-** rangesearch(*points1*, *points2*, *dist*) outputs the index of the points in points1 within a sphere of radio dist for each point in points2.
        - Sort(array, condition) sorts an array given a condition.
        - pca(points) computes Principal Component Analysis of input points.
Outputs eigenvectors and eigenvalues.
id = rangesearch($\boldsymbol{P}_{limits}(x, y, z)$, $\boldsymbol{P}_{limits}(x, y, z)$, 5);
id_sort ← sort(id, numel(id));   %% Indices ordered by number of neighbors
line_index ← zeros(column vector with as many rows as points in $\boldsymbol{P}_{limits}$).
line_number ← 0;
while *NOT* every $(x, y, z) \in \boldsymbol{P}_{limits}$ is assigned to a line
  line_starts ← first element in id_sort such that line_index = 0
  line_propagation ← true;
  cnt ← 1;
  line_number++;
  first ← true;
  while (line_propagation == true)
    If (first)
      points_in_line ← id_sort(line_starts); %%Indices of the points within a line.
      first ← false;
    end
    If (points_in_line have been already assigned to a line)
      line_propagation ← false;
    end
    lines{line_number}(cnt,:) ← [points_in_line(first), points_in_line(last)];
    [eigvec, eigval] = pca($\boldsymbol{P}_{limits}$(points_in_line));
    vectors{line_number}(cnt,:) ← eigvec(1,:); %%Eigenvector of largest eigenvalue.
    cnt++;
    line_index(points_in_line) ← line_number;
    points_in_line ← id(points_in_line(last)) %%Last point of a line, first point of
          the next one.
    points_in_line(line_index != 0) ← []; %%Delete already assigned points.
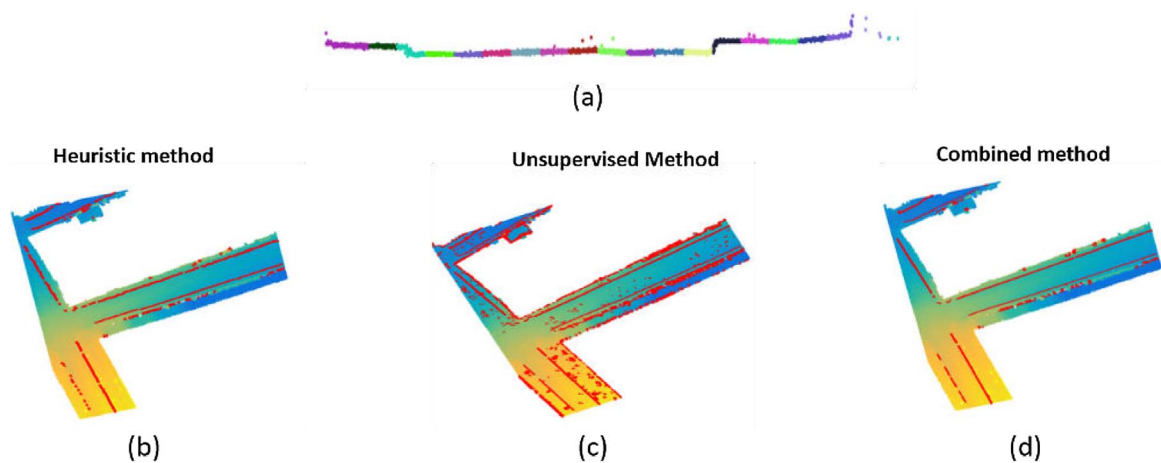 **end**
end
return lines, vectors

Fig. 6. Curb detection. (a) Cross-section of ingle subsection of $\boldsymbol{P}_{gi}^l$. Each colour represents one cell group, (b) Curb map obtained using the heuristic method. (c) Curb map obtained using k-means algorithm. (d) Combined results with curbs shown in red. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
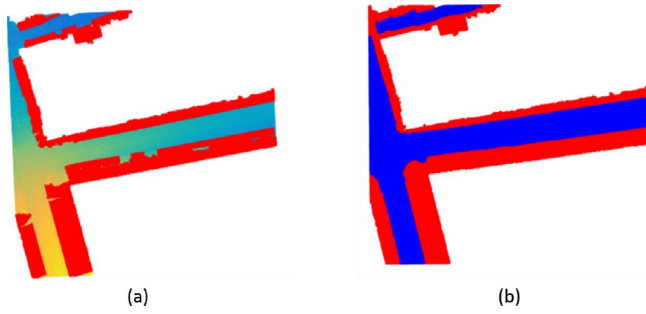
Fig. 7. Sidewalk and pavement extraction. (a) Approximated sidewalk obtained using only geometric features and curb map. (b) After application of radiometric features: pavement (blue) and sidewalk (red). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

3.4.3. Step 4c. ground segment partition

After approximating the road axis, P_{ground} can be subdivided with the principal axes of each section corresponding to the road's longitudinal and transversal axes. For that purpose, a coordinate transformation is done to P_{ground} for each segment G_i obtained in Step 4b. A transformation matrix T is defined such that

$$P_{ground} = [T] \cdot P^t_{ground} \qquad (2)$$

where $[T] = \begin{bmatrix} R_{3x3} & t_{3x1} \\ 0_{1x3} & 1 \end{bmatrix}$ is composed of a rotation matrix R_{3x3} and a translation matrix t_{3x1}. The rotation matrix is defined as a rotation about the z-axis with the angle a that forms the vector v_{G_i} and the y-axis:.

$$[R_{3x3}] = \begin{bmatrix} cosa & -sina & 0 \\ sina & cosa & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (3)$$

and t_{3x1} are the (x, y, z) coordinates of G_i. Note that y-axis is the unit vector [0 1 0] that, according to the global coordinate system of the point cloud P_{raw}, points to the North.

Solving Eq. (2) gives a point cloud centered on G_i, with a y-axis corresponding to the approximate road axis v_{G_i}. A single section is extracted from P^t_{ground} by selecting only those point indices i_{gi} with y-coordinate values of 0–5 m (the upper bound distance is the radius used to compute G_i) and x-coordinate values between −10 m and 10 m, assuming that the sidewalk with will be less than 10m wide. Hence, for each vector v_{G_i}, its indices i_{gi} define each road section in its original coordinates, $P_{gi} = \mathscr{S}(P_{ground}, i_{gi})$; in its transformed coordinates $P^t_{gi} = \mathscr{S}(P^t_{ground}, i_{gi})$. The complete partition of a ground segment can be seen in Fig. 5c.

3.4.4. Step 4d. curb definition

Each section P^t_{gi} is heuristically processed to extract curb points. First, P^t_{gi} is divided into 1m long subsections along the y-axis. Then each subsection is projected onto the XZ plane (a vertical plane whose normal vector is the y-axis) and further divided into cells along the x-axis (Fig. 6a). The following curb features are then computed for each cell $\mathscr{C}_i = \{x, y, z\}|y = 0, x_{min} < x < x_{max}, i = 1...n$.

- $\mu(i)$: Mean of the z-coordinates of the points in $\mathscr{C}_i$.
- $\sigma(i)$: Variance of the z-coordinates of the points in $\mathscr{C}_i$.
- $d_\mu(i)$: Difference between vertical means of $\mathscr{C}_i$ neighboring cells.

$$d_\mu(i) = \begin{cases} |\mu(C_i) - \mu(C_{i+1})| & if\ i = 1 \\ |\mu(C_{i-1}) - \mu(C_i)| & if\ i = n \\ |\mu(C_{i-1}) - \mu(C_{i+1})| & otherwise \end{cases}$$

- $d_\sigma(i)$: Difference between vertical variances of $\mathscr{C}_i$ neighboring cells.

$$d_\sigma(i) = \begin{cases} |\sigma(C_i) - \sigma(C_{i+1})| & if\ i = 1 \\ |\sigma(C_{i-1}) - \sigma(C_i)| & if\ i = n \\ |\sigma(C_{i-1}) - \sigma(C_{i+1})| & otherwise \end{cases}$$

Here, a set of conditions is defined for deciding whether a cell of points contains a curb. Any cell that contains a curb should have high vertical variance $\sigma(i)$, small difference between neighboring cell variances $d_\sigma(i)$ and high difference between neighboring cell heights $d_\mu(i)$. Parameters were selected based on empirical analysis:

cell $\mathscr{C}_i$ contain a curb if $\sigma(i) > 0.025\ a$ nd $d_\mu(i) > 0.075m$ and $d_\sigma(i) < 0.015$.

Cells that fulfill those conditions are selected from which potential curb points can be extracted. While extracted points can be retrieved in the ground point cloud P_{ground} and in the complete point cloud P_n based exclusively on stored indices, this heuristic procedure is insufficient for consistent detection, especially where they are curves. Similar errors may appear whenever the road axis and the y-axis of P^t_{gi} are not fully parallel. To overcome this drawback, an unsupervised classification approach is added as a subsequent step.

For each road section P^t_{gi}, two local features are computed for each point $p \in P^t_{gi}$: (1) vertical range $r(p)$ and (2) vertical variance $\sigma(p)$ of the points within a spherical neighborhood of p. The neighborhood has a radius of 0.2 m, which, for each curb point, will contain points on both sides of the road edge. A two-class classification for both features is applied using a k-means algorithm, in which every subsection of $[r(p), \sigma(p)]$ points is assumed to contain two classes. Points whose features belong the class with the larger range and variance mean are selected as potential curbs.

Now there is a combination of two different curb maps obtained by: (i) the heuristic process (with indices i_{ch} within P^t_{gi}, Fig. 6b) and (ii) an unsupervised classification algorithm (with indices i_{ck} within P^t_{gi}, Fig. 6c). The combination is shown in Fig. 6d. The process is conducted in a section-wise fashion. First, the intersection between both curb maps $i_{ci} = i_{ch} \cap i_{ck}$ is obtained. If there are no points in the intersection (as will occur in curved sections), i_{ci} is defined as $i_{ci} = i_{ck}$. Subsequently, a region growing process is applied using the points with indices i_{ci} as seeds points and i_{ck} as a growing region via iterative neighbor searching. The process stops when no more points can be reached. A new set of indices, i_{cc} (related with P_{gi} and therefore with P_{ground}), is finally generated. The output i_{cc} is a set of curb point indices.

3.5. Step 5. road features detection

3.5.1. Step 5a. pavement and sidewalk extraction

For pavement and sidewalk extraction, let $P_{curbs} = \{P_{c1} \cup ...P_{cj}...\cup P_{cn}\} = \mathscr{S}(P_{ground}, \{i_{c1} \cup ...i_{cj}...\cup i_{cn}\})$, $j = 1...n$ be the curb map and $R_{ground} = \mathscr{R}(P_{ground}, g_{r2})$ the raster structure of the ground points. First, a coarse description of the sidewalk is obtained using the curb map and the following binary images:

- I_g = occupancy image of P_{ground} (Fig. 5b), where every cell of R_{ground} that contains at least one point of P_{ground} equals "true".
- I_{lim} = occupancy image of P_{limits}, where every cell of R_{ground} that contains at least one point of P_{limits} equals "true".
- I_{curb} = occupancy image of P_{curbs}.

A region growing process is applied based on the following rule: ground limits are allowed to grow until they find a curb (see Algorithm 2). This is conduced section-wise and results in a coarse description of the pavement (Fig. 7a); sections without curb points are excluded from analysis. Next pixel indices of the binary image I_{as} (i.e. an approximation of the sidewalks) are retrieved.

When the sidewalk and pavement are comprised of different materials (e.g. concrete vs asphalt), then radiometric properties of the point cloud can be used to increase geometrical accuracy. Material reflectance difference in the form of intensity readings can be seen on the
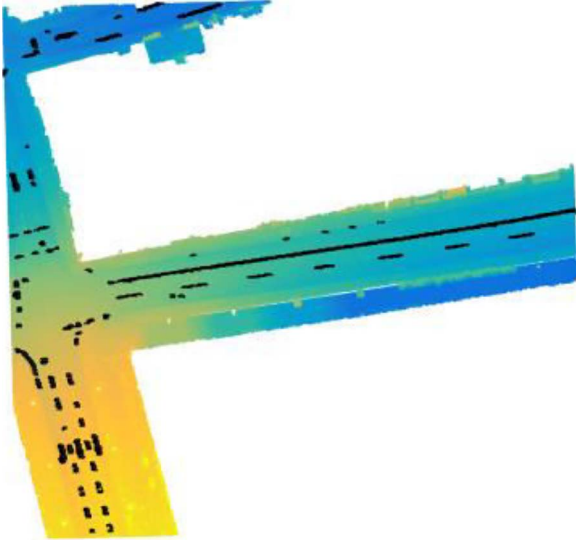
Fig. 8. Road marking detection. Points detected as road markings are highlighted on the ground point cloud.

intensity based image of R_{ground}. To obtain two pixel classes based on intensity values, a k-means algorithm is applied, and two binary images I_{low} and I_{high} based on intensity cluster values with respect to the image's mean intensity. However, reflectance is itself not a robust separator. To achieve that a binary operation $I_p = I_{low} \wedge (\neg I_{as})$ is used to remove pixels that belong to the sidewalk, as obtained using the curb map. Hence, I_p contains mainly the pavement. As all the images are directly related with the raster structure R_{ground}, the surface of each pixel can be measured as $g_{r_2}^2$, and an area filter can be applied to remove connected components for surface smaller than $10m^2$ (assuming pavement to be a large, fully connected surface), thereby obtaining a filtered binary image I'_{low}. To fill holes (mainly caused by small reflective elements on the road surface) in the binary image, a spherical structuring element H with a radius of 5 pixels is defined, and a closing operation is done so that the binary image $I_{pavement} = I'_{low} \oplus H$ defines the pavement, and $I_{sidewalk} = I_g \wedge (\neg I_{pavement})$ defines the sidewalk. Point indices $\boldsymbol{i}_{pavement}$ and $\boldsymbol{i}_{sidewalk}$ can be extracted from both images, and the correspondent segmented point clouds are defined as $\boldsymbol{P}_{pavement} = \mathscr{S}(\boldsymbol{P}_{ground}, \boldsymbol{i}_{pavement})$ and $\boldsymbol{P}_{sidewalk} = \mathscr{S}(\boldsymbol{P}_{ground}, \boldsymbol{i}_{sidewalk})$ respectively (Fig. 7b).

Algorithm 2. Coarse definition of the sidewalk.

Data: I_g, I_{lim}, I_{curb}
Output: I_{as}
Functions: neighbors(I): Find the 8-neighbourhood of each true pixel in I.
       bwconncomp(I): Find the connected components of binary image I.
         intersect(a, b): Find the common elements between arrays a and b.
conn = bwconncomp(I_{lim})
for each connected component i
growing_image ← I_{lim}(connected component i);
growing ← true;
  while growing == *true*
  neigh = neighbors(growing_image);
  valid_neighbors = intersect(growing_image, I_g) %%Neighbors within the road section
  growing_image(valid_neighbors) ← true %%Growing step
  if (intersect(growing_image, I_{curb}) is not an empty array) OR (growing_image is equal to
        growing_image_prev)
    growing = false;
  end
  growing_image_ant = growing_image;
  end
end
return I_{as} ← growing_image

3.6. Step 5b. road markings detection

Once the road surface is extracted, its intensity-based image can be obtained using R_{ground}. Since road markings are painted with reflective materials to ensure visibility under low lighting conditions, their reflectance can be assumed to be much higher than the pavement's. Reflectance difference can be highlighted by applying a standard deviation filter on the pavement intensity image in a 3-by-3 pixel neighborhood. The resulting image I_σ highlights pixels where the reflectance difference is large. These pixels are selected via a k-means algorithm (separating high ($I_{\sigma h}$) from low ($I_{\sigma l}$) using the standard deviations in a process analogous to Section 2.5.1). The binary operation $I_{ph} = I_{high} \wedge I_{\sigma h}$ selects points on the pavement with high reflectance and a high standard deviation. Finally, an area filter, similar to that described in Section 2.5.2 deletes connected components with less than two pixels, thereby obtaining a binary image $I_{markings}$. Point indices $\boldsymbol{i}_{markings}$ are extracted so a point cloud containing the detected road markings can be computed as $\boldsymbol{P}_{markings} = \mathscr{S}(\boldsymbol{P}_{ground}, \boldsymbol{i}_{markings})$ (Fig. 8).

4. Validation

4.1. Study data

The aforementioned described technique was applied to a high density urban LiDAR data set for a portion of Dublin, Ireland (Laefer et al., 2017) (Fig. 9a). The flight campaign was planned using approaches described in Hinks et al. (2009) to adapt ALS to the complexity of urban areas by improving the visualization of small details of the environment and maximizing data coverage of building façades, which involved a 67% overlap in straight tracks oriented with a 45° offset with respect to the predominant street axes, to minimize self-shadowing effects. Captured from a rotary platform at approximately 300 with a 30° scanning angle, most location are captured six times from unique sky locations (Hinks et al., 2015). All flightlines are shown in Fig. 9a. For this study, a pair of $100m^2$ sections were selected, with typical horizontal point densities 230 points/m^2. These are referenced as $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ (Figs. 9b-c) and reflect a complicated urban area with a bridge over the Liffey River, a variety of buildings of different height, a mixture of street types including a cobbled street. The data and metadata are fully available at (Laefer et al., 2017).
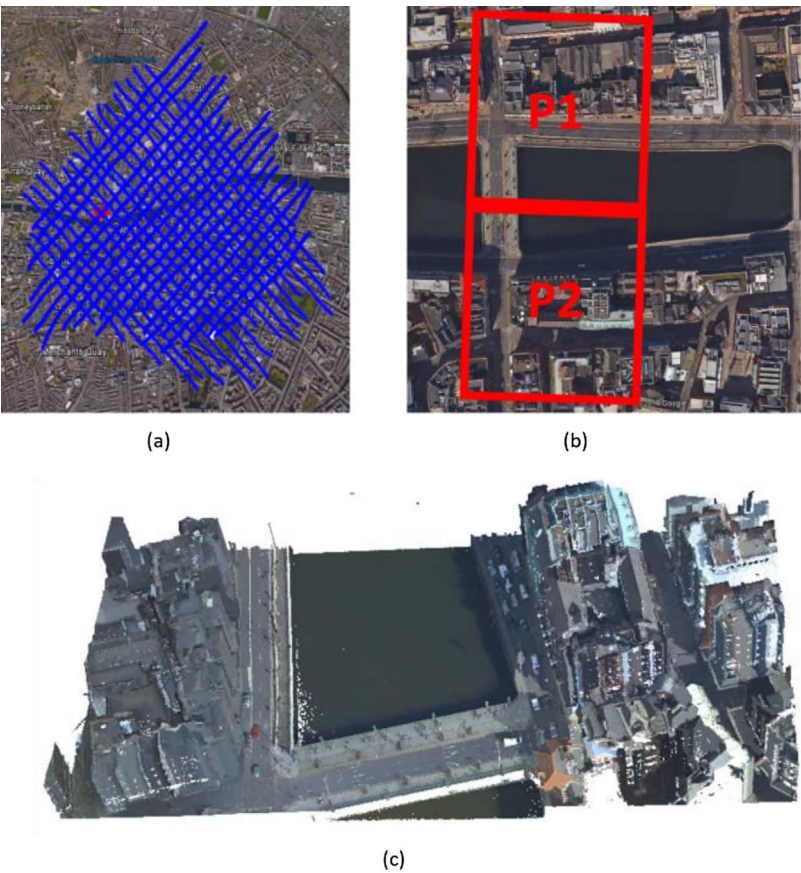
(a)

(b)

Fig. 9. Case study. (a) Complete flight path trajectory over Dublin city centre. (b) Areas of study. (c) Section of point cloud for the study areas.
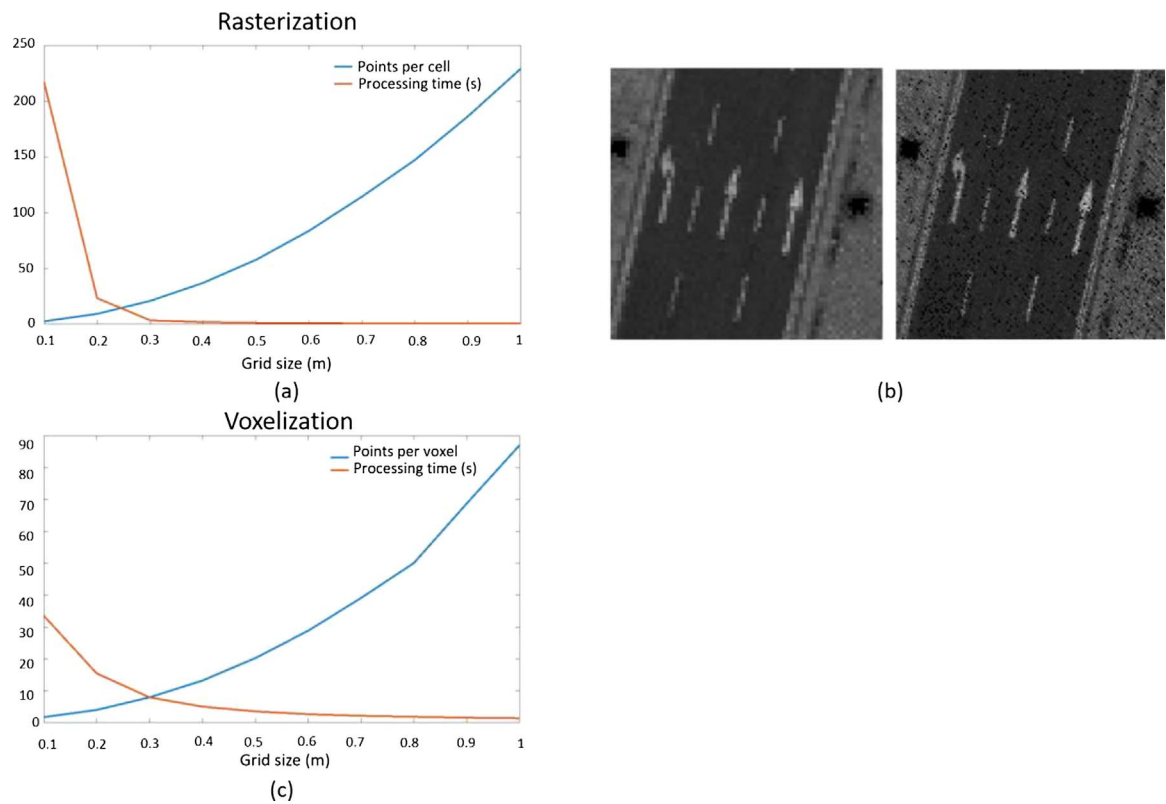

(c)

(a)

(b)

(c)

Fig. 10. Parameter selection. (a) Execution times and points per cell for raster generation of different grid sizes. (b) Intensity based raster image for 0.2 m (left) and 0.1 m (right). (c) Execution times and points per voxel for voxel creation of different grid size.
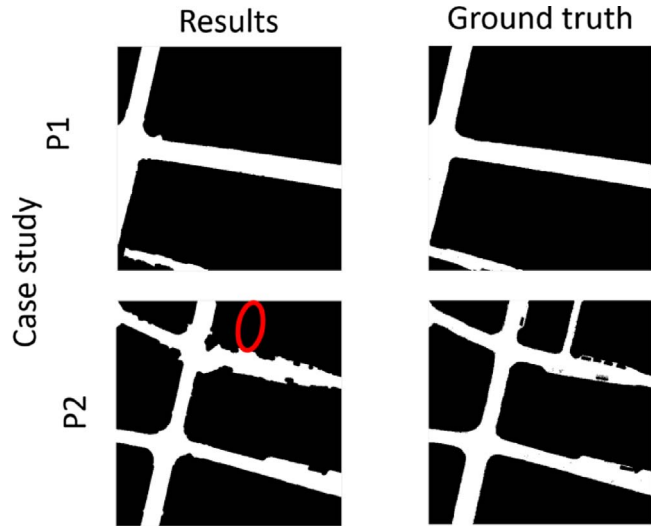
Fig. 11. Pavement detection results for both case study point clouds, together with the manually generated ground truth which is used for comparison.

Table 1
Results for sidewalk area detection.

	Precision	Recall	Fscore
P_1	0.939	0.990	0.964
P_2	0.895	0.896	0.920
Average	0.917	0.943	0.942

4.2. Parameter selection and ground truth generation

To generate the 2D raster and 3D voxel structures, three parameters were selected. The first two (g_{r1}, g_{r2}) define the cell size of the 2D raster grids used throughout the methodology scheme. Both parameters were selected attending to two criteria: Execution time and point density. Fig. 10a shows the processing time required for the creation of a raster structure $R_n = \mathcal{R}(P_n, g_r)$ together with the average number of points per cell, using cell grids ranging from $g_r = 0.1m$ to $g_r = 1m$. While the execution time decreases exponentially with the grid size, the number of points per cell increases quadratically. In general terms, grid sizes smaller than 0.1m are not efficient in terms of execution time and causes the raster images to lose quality, as the number of empty cells increases (Fig. 10b). Grid size g_{r1} defines R_{ns} and R_{fp} grid sizes for building façade segmentation (Step 2) and ground segmentation (Step 3), respectively. Selecting $g_{r1} = 0.3m$ represents a balance between computational efficiency and qualitative results. Similarly, g_{r2}, which defines the R_{ground} grid size used for the definition of curb map, sidewalk, pavement and road markings (Steps 4–5) where the raster structure is repeatedly utilized was wet to 0.2m. The term is also used in the standard deviation filter in Step 5a, which requires the raster structure to capture local information with higher resolution.

The third term, g_{v1} defines the voxel size of V_{ns}, the 3D cubic grid used for ground extraction (Step 3) and employs the same raster grid size parameter criteria. A similar analysis was, therefore performed, defining a voxel structure $V_n = \mathcal{V}(P_n, g_v)$ with g_v ranging from $g_v = 0.1m$ to $g_v = 1m$ (Fig. 10c). For Step 3, the optimal voxel size was $g_{v1} = 0.3m$.

To quantify the performance of the proposed method, ground truth was generated for the pavement area, sidewalk area, and road markings from the raster structure R_{ground} using manual pixel-by-pixel manual labelling. This was possible given their clear visual distinction in the intensity based image of R_{ground}. The evaluation metrics used were Precision (Eq. (4)), Recall (Eq. (5)), and F-score (Eq. (6)):

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$F_{score} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{6}$$

For this, let I_r be the binary image corresponding to the manually generated reference and I_m the corresponding binary image of the road surface generated from the proposed method. True positives (TP), false positives (FP) and false negatives (FN) are defined as Eqs. (7)–(9), respectively:
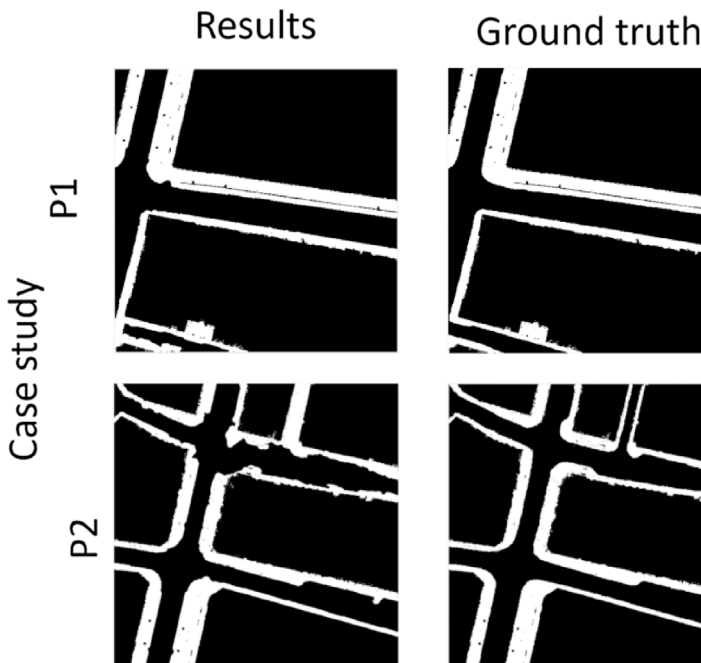
$$TP = \sum (I_r \wedge I_m) \tag{7}$$

Fig. 12. Sidewalk detection results for both case study point clouds, together with the manually generated ground truth which is used for comparison.

Table 2
Results for pavement area detection.

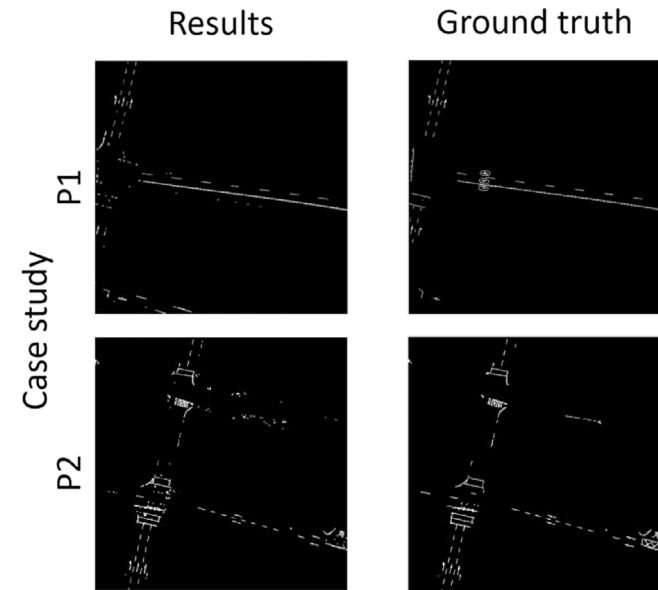|	Precision	Recall	Fscore
P_1	0.991	0.943	0.967
P_2	0.956	0.917	0.936
Average	0.973	0.93	0.950

Fig. 13. Road marking detection results for both case study point clouds, together with the manually generated ground truth which is used for comparison.

Table 3
Results for road marking detection.

	Precision	Recall	Fscore
P_1	0.786	0.763	0.773
P_2	0.858	0.871	0.864
Average	0.822	0.817	0.818

$$FP = \sum \left((\neg I_r) \Lambda I_m \right) \tag{8}$$

$$FN = \sum \left(I_r \Lambda (\neg I_m) \right) \tag{9}$$

4.3. Sidewalk and pavement detection

Fig. 11 shows the outcome for the sidewalk detection performance metrics quantified in Table 1. Fig. 12 and Table 2 show the equivalent for the pavement detection performance.

In both cases average results were high quality, with F-scores around 95%. Notably the relatively high false positives and false negatives in P_2 appear on one street (Fig. 11) composed of cobbled stones (Essex Street). Elsewhere performance was consistently robust.

4.4. Road marking detection

Road markings detection is shown in Fig. 13 and quantified in Table 3 demonstrated success exceeding 80%.

5. Discussion

The robustness of the proposed method is influenced by different

Table 4
Comparison with state of the art methods.

Method	Precision	Recall	Fscore	Point Density (pts/m^2)
Yu et al. (2015)	0.91	0.93	0.925	Unknown
Soilán et al. (2017)	0.961	0.917	0.939	2185.5
Proposed	0.822	0.817	0.818	204.4

error sources, which may be cumulative. Sidewalk and pavement segmentation depends on geometric and radiometric features, both of which may contain errors. Furthermore, the technique relies upon the reflectance of the pavement being distinguishable from that of the sidewalk. If the materials of both pavement and sidewalk were similar, the algorithm would have to rely only on the curb map. As aerial data densities continue to improve the ability to capture such small features from the air will also improve. The success of this technique presented herein demonstrates the capability to do this with data densities in the range of hundreds of points per square meter, as opposed the current approaches that typically use thousands of points per square meter.

In fact, prior to the research herein, road marking detection has been done exclusive with much denser terrestrial (RIEGL VMX-450) or mobile (LYNX Mobile Mapper) based laser scans, with data densities one to two orders of magnitude higher. Notably only a 12–13% better score has been reported with such work (e.g. Table 4) despite the much higher costs related to the collection, storage and processing of these substantially denser datasets, thus showing the strong capabilities of the work proposed herein (Table 4).

6. Conclusions

This paper introduces a method for detection of road networks and their markings using aerial laser scanning. The procedure starts by segmenting building facades to provide contextual information for robust ground segment via region growing within a voxel space. Then, the ground segment is hierarchically divided into different segments with unique semantic meanings. Next, a section-wise approach is used to obtain a curb map, which is computed combining a heuristic approach and an unsupervised learning algorithm. Finally, curb information, together with the radiometric properties of the ground materials, are used for sidewalk and pavement segmentation, while road markings are detected based on reflectance levels.

When applied to a pair of $100m^2$ sections Dublin, Ireland, F-scores around 95% were obtained for pavement and sidewalks and 80% for road markings. These were within 13% of results reported for competing techniques using datasets an order of magnitude denser, thus showing the long term scalability potential of this technique as aerial laser scanning data sets continue to improve in quality. This fact, together with the algorithms that detect curbs and segment pavement and sidewalk, is the main contribution of this work.

References

Aljumaily, H., Laefer, D.F., Cuadra, D., 2016. Big-Data approach for three-Dimensional building extraction from aerial laser scanning. J. Comput. Civ. Eng. 30, 4015049.

http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0000524.

Aljumaily, H., Cuadra, D., Laefer, D.F., 2017. Urban point cloud mining based on density clustering and MapReduce. J. Comput. Civ. Eng. 31. http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0000674.

Alshehhi, R., Marpu, P.R., 2017. Hierarchical graph-based segmentation for extracting road networks from high-resolution satellite images. ISPRS. J. Photogramm. Remote Sens. 126, 245–260. http://dx.doi.org/10.1016/j.isprsjprs.2017.02.008.

Anil, P.N., Natarajan, S., 2010. A Novel Approach Using Active Contour Model for Semi-Automatic Road Extraction from High Resolution Satellite Imagery 2010. Second Int. Conf. Mach. Learn Comput, pp. 263–266. http://dx.doi.org/10.1109/ICMLC.2010.36.

Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S., Çöltekin, A., 2015. Applications of 3D city models: state of the art review. ISPRS Int. J. Geo-Inf. 4, 2842–2889. http://dx.doi.org/10.3390/ijgi4042842.

Byrne, J., Laefer, D.F., O'Keeffe, E., 2017a. Maximizing feature detection in aerial unmanned aerial vehicle datasets. J. Appl. Remote Sens. 11 (2), 25015. http://dx.doi.org/10.1117/1.JRS.11.025015.

Byrne, J., O'Keeffe, E., Lennon, D., Laefer, D.F., 2017b. 3D Reconstructions Using Unstabilised Video Footage from an Unmanned Aerial System. J. Imaging 3 (2), 15. http://dx.doi.org/10.3390/jimaging3020015.

Chen, D., He, X., 2015. Fast automatic three-dimensional road model reconstruction based on mobile laser scanning system. Opt. − Int. J. Light Electron. Opt. 126, 725–730. http://dx.doi.org/10.1016/j.ijleo.2015.02.021.

Chen, S., Laefer, D., Mangina, E., 2016a. State of technology review of civilian UAVs. Recent Pat. Eng. 10, 160–174. http://dx.doi.org/10.2174/1872212110666160712230039.

Chen, S., Laefer, D., Mangina, E., 2016b. State of technology review of civilian UAVs. Recent Pat. Eng. 10, 160–174. http://dx.doi.org/10.2174/1872212110666160712230039.

Chen, S., Laefer, D., Mangina, E., 2016c. State of technology review of civilian UAVs. Recent Pat. Eng. 10, 160–174. http://dx.doi.org/10.2174/1872212110666160712230039.

Douillard, B., Underwood, J., Kuntz, N., Vlaskine, V., Quadros, a., Morton, P., Frenkel, a., 2011. On the segmentation of 3D lidar point clouds. Proc. − IEEE Int. Conf. Robot. Autom. 2798–2805. http://dx.doi.org/10.1109/ICRA.2011.5979818.

Höfle, B., Pfeifer, N., 2007. Correction of laser scanning intensity data: data and model-driven approaches. ISPRS J. Photogramm. Remote Sens. 62, 415–433. http://dx.doi.org/10.1016/j.isprsjprs.2007.05.008.

He, Y., Song, Z., Liu, Z., 2017. Updating highway asset inventory using airborne. LiDAR 104, 132–141. http://dx.doi.org/10.1016/j.measurement.2017.03.026.

Hinks, T., Carr, H., Laefer, D.F., 2009. Flight Optimization Algorithms for Aerial LiDAR Capture for Urban Infrastructure Model Generation. J. Comput. Civ. Eng. 23, 330–339.

Hinks, T., Carr, H., Gharibi, H., Laefer, D.F., 2015. Visualisation of urban airborne laser scanning data with occlusion analysis. ISPRS J. Photogramm. Remote Sens. 104, 77–87. http://dx.doi.org/10.1016/j.isprsjprs.2015.01.014.

Holgado-Barco, A., González-Aguilera, D., Arias-Sanchez, P., Martinez-Sanchez, J., 2015. Semiautomatic extraction of road horizontal alignment from a mobile. LiDAR Syst. Comput. Civ. Infrastruct. Eng. 30, 217–228. http://dx.doi.org/10.1111/mice.12087.

Hu, X., Li, Y., Shan, J., Zhang, J., Zhang, Y., 2014. Road centerline extraction in complex urban scenes from LiDAR data based on multiple features. IEEE Trans. Geosci. Remote Sens. 52, 7448–7456. http://dx.doi.org/10.1109/TGRS.2014.2312793.

Hui, Z., Hu, Y., Jin, S., Yevenyo, Y.Z., 2016. Road centerline extraction from airborne LiDAR point cloud based on hierarchical fusion and optimization. ISPRS J. Photogramm. Remote Sens. 118, 22–36. http://dx.doi.org/10.1016/j.isprsjprs.2016.04.003.

Idrees, M., Mohd Shafri, H.Z., Saeidi, V., 2013. Imaging spectroscopy and light detection and ranging data fusion for urban features extraction. Am. J. Appl. Sci. 10, 1575–1585. http://dx.doi.org/10.3844/ajassp.2013.1575.1585.

Kaasalainen, S., Ahokas, E., Hyyppä, J., Suomalainen, J., 2005. Study of surface brightness from backscattered laser intensity: calibration of laser data. IEEE Geosci. Remote Sens. Lett. 2, 255–259. http://dx.doi.org/10.1109/LGRS.2005.850534.

Kirthika, A., Mookambiga, A., 2011. Automated road network extraction using artificial neural network. IEEE-Int. Conf. Recent Trends Inf. Technol. 2011, 1061–1065. http://dx.doi.org/10.1109/ICRTIT.2011.5972323.

Kumar, P., Lewis, P., McElhinney, C.P., Boguslawski, P., McCarthy, T., 2017. Snake energy analysis and result validation for a mobile laser scanning data-Based automated road edge extraction algorithm. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 10, 763–773. http://dx.doi.org/10.1109/JSTARS.2016.2564984.

Laefer, D.F., Abuwarda, S., Vo, A.-V., Truong-Hong, L. and Gharibi, H. (2017). "2015 Aerial Laser and Photogrammetry Survey of Dublin City Collection Record" https://geo.nyu.edu/catalog/nyu_2451_38684.

Lloyd, S.P., 1982. Least squares quantization in PCM. IEEE Trans. Inf. Theory 28, 129–137. http://dx.doi.org/10.1109/TIT.1982.1056489.

Miao, Z., Shi, W., Zhang, H., Wang, X., 2013. Road centerline extraction from high-resolution imagery based on shape features and multivariate adaptive regression splines. IEEE Geosci. Remote Sens Lett. 10, 583–587. http://dx.doi.org/10.1109/LGRS.2012.2214761.

Otsu, N., 1979. Threshold selection method from grey-level histograms. IEEE Trans. Syst. Man Cybern.

Poullis, C., 2014. Tensor-Cuts: a simultaneous multi-type feature extractor and classifier and its application to road extraction from satellite images. ISPRS J. Photogramm. Remote Sens. 95, 93–108. http://dx.doi.org/10.1016/j.isprsjprs.2014.06.006.

Puente, I., González-Jorge, H., Martínez-Sánchez, J., Arias, P., 2013. 2013. Review of mobile mapping and surveying technologies. Meas. J. Int. Meas. Confed. http://dx.doi.org/10.1016/j.measurement.2013.03.006.

Rodríguez-Cuenca, B., García-Cortés, S., Ordóñez, C., Alonso, M.C., 2015. An approach to detect and delineate street curbs from MLS 3D point cloud data. Autom. Constr. 51, 103–112. http://dx.doi.org/10.1016/j.autcon.2014.12.009.

Shen, J., Lin, X., Shi, Y., Wong, C., 2008. Knowledge-Based road extraction from high resolution remotely sensed imagery. Congr. Image Signal Process. 4, 608–612. http://dx.doi.org/10.1109/CISP.2008.519.

Soilán, M., Riveiro, B., Martínez-Sánchez, J., Arias, P., 2017. Segmentation and classification of road markings using MLS data. ISPRS J. Photogramm. Remote Sens. 123, 94–103. http://dx.doi.org/10.1016/j.isprsjprs.2016.11.011.

Vain, A., Xiaowei, Y., Kaasalainen, S., Hyyppä, J., 2010. Correcting airborne laser scanning intensity data for automatic gain control effect. Geosci. Remote Sens. Lett. IEEE 7, 511–514. http://dx.doi.org/10.1109/LGRS.2010.2040578.

Vo, A.-V., Truong-Hong, L., Laefer, D.F., Bertolotto, M., 2015a. Octree-based region growing for point cloud segmentation. ISPRS J. Photogramm. Remote Sens. 104, 88–100. http://dx.doi.org/10.1016/j.isprsjprs.2015.01.011.

Vo, A., Laefer, D.F., Tiede, D., Oleire-oltmanns, S., Baraldi, A., Shimoni, M., Moser, G., Member, S., Tuia, D., Member, S., 2016b. Processing of extremely high resolution LiDAR and RGB data: outcome of the IEEE GRSS data fusion contest — part B: 3 -D contest. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 9, 1–16. http://dx.doi.org/10.1109/JSTARS.2016.2581843.

Vo, A.V., 2016. Spatial Data Storage and Processing Strategies for Urban Laser Scanning.

Wang, H., Luo, H., Wen, C., Cheng, J., Li, P., Chen, Y., Wang, C., Li, J., 2015. Road boundaries detection based on local normal saliency from mobile laser scanning data. IEEE Geosci. Remote Sens. Lett. 12, 2085–2089.

Wang, W., Yang, N., Zhang, Y., Wang, F., Cao, T., Eklund, P., 2016. A review of road extraction from remote sensing images. J. Traffic Transp. Eng. English Ed. 3, 271–282. http://dx.doi.org/10.1016/j.jtte.2016.05.005.

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 J. Photogramm. Remote Sens. 79, 80–93. http://dx.doi.org/10.1016/j.isprsjprs.2013.01.0.

Yang, B., Dong, Z., Liu, Y., Liang, F., Wang, Y., 2017. Computing multiple aggregation levels and contextual features for road facilities recognition using mobile laser scanning data. ISPRS J. Photogramm. Remote Sens. 126, 180–194. http://dx.doi.org/10.1016/j.isprsjprs.2017.02.014.

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 J. Sel. Top. Appl. Earth Obs. Remote Sens. 8, 709–726.

Yu, Y., Li, J., Wen, C., Guan, H., Luo, H., Wang, C., 2016. Bag-of-visual-phrases and hierarchical deep models for traffic sign detection and recognition in mobile laser scanning data. ISPRS J. Photogramm. Remote Sens. 113, 106–123. http://dx.doi.org/10.1016/j.isprsjprs.2016.01.005.

Zhao, J., You, S., Huang, J., 2011. Rapid extraction and updating of road network from airborne LiDAR data. IEEE Appl. Imag. Pattern Recognit. Work 1–7. http://dx.doi.org/10.1109/AIPR.2011.6176360.

Zhou, L., Vosselman, G., 2012. Mapping curbstones in airborne and mobile laser scanning data. Int. J. Appl. Earth Obs. Geoinf. 18, 293–304. http://dx.doi.org/10.1016/j.jag.2012.01.024.

Zhu, Q., Mordohai, P., 2009. A minimum cover approach for extracting the road network from airborne LIDAR data. In: ICCV Workshops. IEEE 12th International Conference on Computer Vision Workshops 2009. pp. 1582–1589. http://dx.doi.org/10.1109/ICCVW.2009.5457423.