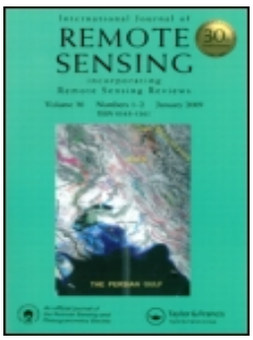


International Journal of Remote Sensing

ISSN: 0143-1161 (Print) 1366-5901 (Online) Journal homepage: http://www.tandfonline.com/loi/tres20

Automated extraction of street-scene objects from mobile lidar point clouds

Bisheng Yang , Zheng Wei , Qingquan Li & Jonathan Li

To cite this article: Bisheng Yang , Zheng Wei , Qingquan Li & Jonathan Li (2012) Automated extraction of street-scene objects from mobile lidar point clouds, International Journal of Remote Sensing, 33:18, 5839-5861, DOI: 10.1080/01431161.2012.674229

To link to this article: http://dx.doi.org/10.1080/01431161.2012.674229

Published online: 30 Mar 2012.

International Journal of Remote Sensing
Vol. 33, No. 18, 20 September 2012, 5839–5861

Taylor & Francis
Taylor & Francis Group

Automated extraction of street-scene objects from mobile lidar point clouds

BISHENG YANG*†, ZHENG WEI*†, QINGQUAN LI† and JONATHAN LI†‡
†State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China
‡Department of Geography and Environmental Management, Faculty of Environment, University of Waterloo, Waterloo, Canada N2L 3G1

(*Received 16 September 2010; in final form 29 July 2011*)

Mobile laser scanning or lidar is a new and rapid system to capture high-density three-dimensional (3-D) point clouds. Automatic data segmentation and feature extraction are the key steps for accurate identification and 3-D reconstruction of street-scene objects (e.g. buildings and trees). This article presents a novel method for automated extraction of street-scene objects from mobile lidar point clouds. The proposed method first uses planar division to sort points into different grids, then calculates the weights of points in each grid according to the spatial distribution of mobile lidar points and generates the geo-referenced feature image of the point clouds using the inverse-distance-weighted interpolation method. Finally, the proposed method transforms the extraction of street-scene objects from 3-D mobile lidar point clouds into the extraction of geometric features from two-dimensional (2-D) imagery space, thus simplifying the automated object extraction process. Experimental results show that the proposed method provides a promising solution for automatically extracting street-scene objects from mobile lidar point clouds.

1. Introduction

A laser scanning or lidar system provides an efficient solution for capturing spatial data in a fast, efficient and highly reproducible way. It has been widely used in many fields, such as cultural heritage documentation, reverse engineering, three-dimensional (3-D) object reconstruction and digital elevation model (DEM) generation, as it can directly obtain the 3-D coordinates of objects (Biosca and Lerma 2008). Laser scanning can be divided into three categories, namely, airborne laser scanning (ALS), terrestrial laser scanning (TLS) and mobile laser scanning (MLS) or mobile lidar. ALS has been successfully used for DEM generation (Kraus and Pfeifer 1998, Axelsson 2000, Sithole and Vosselman 2004) and reconstruction of building rooftops (Haala and Brenner 1999, Maas and Vosselman 1999, Overby *et al*. 2004, Akel *et al*. 2009). However, it has difficulties in capturing points of the facades of buildings. As mobile mapping technology has made great progress (Schwarz and El-Sheimy 2007, Tao and Li 2007, Toth 2009), mobile lidar allows the rapid and cost-effective capture of 3-D

*Corresponding authors. Email: bshyang@whu.edu.cn (B. Yang); weizheng0628@163.com (Z. Wei)

International Journal of Remote Sensing
ISSN 0143-1161 print/ISSN 1366-5901 online © 2012 Taylor & Francis
http://www.tandfonline.com
http://dx.doi.org/10.1080/01431161.2012.674229

data from large street sections, including dense point coverage of building facades (Haala *et al.* 2008a).

Mobile lidar systems such as the vehicle-borne laser measurement system (VLMS) (Manandhar and Shibasaki 2002, Zhao and Shibasaki 2005), StreetMapper (Hunter *et al*. 2006, Kremer and Hunter 2007, Haala *et al*. 2008b, StreetMapper 2010), LYNX (Optech 2009, Zampa and Conforti 2009) and FGI Roamer (Kukko *et al*. 2007, Jaakkola *et al*. 2008) have been actively studied and implemented in the past decade. Kremer and Hunter (2007) evaluated the accuracy and performance of the StreetMapper system in some real-world projects. A mobile lidar system can capture high-accuracy, high-density points at a precision and resolution exceeding those available through aerial photogrammetry when using stationary terrestrial lidar would be time-consuming (Zampa and Conforti 2009). A detailed review of current mobile lidar systems was presented by Barber *et al.* (2008). However, when compared with the advances in mobile lidar systems, automated algorithms and software tools for efficiently extracting 3-D street-scene objects of interest from mobile lidar point clouds rather fall behind, due to huge data volumes and the complexity of urban street scenes. To extract street-scene objects or detailed features of building facades, mobile lidar point clouds need to be classified into different categories (e.g. buildings and trees), which is a key step for accurate identification and 3-D reconstruction of street-scene objects.

Different from the approaches for ALS data processing, methods for processing mobile lidar data have to deal fully with 3-D point clouds (Biosca and Lerma 2008). Due to the non-unique correspondence between (X, Y) coordinates in the horizontal plane and the Z coordinate, the algorithms for filtering and classifying ALS data, such as the triangulated irregular network (TIN)-based filtering method (Axelsson 2000), have difficulties in handling mobile lidar data because of data dimensionality.

Several researchers have proposed the use of prior knowledge in extracting buildings from mobile lidar point clouds. Manandhar and Shibasaki (2001, 2002) used the information from individual scan lines, geometric structures and densities of points to classify mobile lidar points into roads and buildings. Abuhadrous *et al.* (2004) classified points into buildings, roads and trees by histograms of Z and Y directions in one profile. These approaches need prior knowledge of profiles and scan lines for classifying mobile lidar point clouds. Li *et al.* (2004) proposed a method of density of projected points (DoPP) to extract buildings. However, this method has difficulties in setting a reasonable threshold of density for the extraction of building boundaries.

Other researchers have proposed the use of ancillary data, such as range imagery (Brovelli and Cannata 2004) or digital images captured by CCD cameras (Becker and Haala 2007), for classification and object extraction from mobile lidar point clouds. Mrstik and Kusevic (2009) fused airborne lidar data with mobile lidar data to add building roofs, reduce shadows and increase coverage. They also coloured the mobile lidar point clouds by fusing the information of the mounted line scan camera. However, ancillary data are not always available. On the other hand, the integration of geometric and semantic data shows promise for object extraction and reconstruction from mobile lidar point clouds. A bottom-up process for window extraction was proposed by Pu and Vosselman (2009). They classified the point clouds into different geometric features and applied semantic feature recognition to extract ground, wall, door and roof protrusion and intrusion. Then, the method of hole-based window extraction was adopted to further identify windows. Becker (2009) aimed at data-driven facade reconstruction by combining the bottom-up and top-down

strategies with automatically inferred rules of facade elements. Their approach first searches for holes on the facade to model windows and doors using cell decomposition, and then refines the geometry of the facade elements based on semantic data. For incomplete data, a facade grammar is automatically inferred and then used for the generation of facade structures. The two methods mentioned above can extract building facades and even more detailed features (e.g. windows) from mobile lidar point clouds. However, these methods have difficulties in extracting the point clouds of individual buildings from mobile lidar point clouds that include trees, buildings and so on, because of huge data volume, redundancy and occlusion. Other related studies including road surface modelling from mobile lidar data can be found in the works of Kukko *et al.* (2007) and Jaakkola *et al.* (2008). To this end, an efficient solution for automatically extracting the point clouds of street-scene objects from mobile lidar point clouds is urgently needed.

Inspired by the integration of images and point clouds for extraction of features in the field of ALS, the authors of this article present a method for automated extraction of street-scene objects from mobile lidar point clouds. The proposed method first generates the geo-referenced feature image of mobile lidar point clouds and then extracts the boundaries of street-scene objects (e.g. buildings) by applying the method of image segmentation and contour tracing on the geo-referenced feature image generated. Once the boundaries of street-scene objects are extracted, the 3-D points associated with street-scene objects can easily be isolated and extracted from mobile lidar point clouds. The proposed method is able to:

- generate the geo-referenced feature image that preserves implicit geometric features in mobile lidar point clouds;
- extract the boundaries of street-scene objects from the geo-referenced feature image; and
- extract the point clouds of street-scene objects from 3-D point clouds by the 2-D boundaries extracted.

2. Method for extraction of street-scene objects

It is rather difficult to directly extract street-scene objects from mobile lidar point clouds because of noise in the data, huge data volume and no explicit or ancillary data (e.g. intensity image). Although clustering-based methods (Biosca and Lerma 2008) can extract 3-D boundaries and local geometric features of street-scene objects, the efficiency is relatively low and the results are uncertain. We propose a method for generating a geo-referenced feature image that is capable of maintaining implicit geometric features in mobile lidar point clouds. Hence, the existing operators for extracting line or key point features from imagery can be adopted to extract the boundaries of street-scene objects from the feature image generated. Figure 1 elaborates the flow chart of the proposed method. The proposed method encompasses two key components, namely feature image generation and street-scene object extraction.

In light of the flow chart of the proposed method, feature image generation is the first step in street-scene object extraction. Although the geo-referenced feature image can preserve the spatial distribution and the density of mobile lidar points, the feature image is also able to reflect the varied characteristics of different street-scene objects, which are represented as grey values in the feature image. To fulfil the above requirements, the grey values of each pixel and the resolution of the geo-referenced feature image have to be determined.

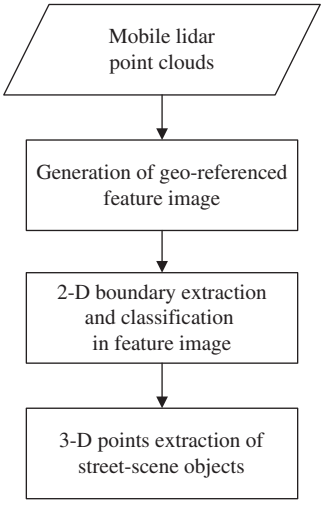


Figure 1. Flow chart of the proposed method.

2.1 *Generation of the geo-referenced feature image*

The proposed method first divides the mobile lidar points into grids on the XOY plane, using (X, Y) coordinates of points. According to the coordinates of the lidar points, the minimum bounding box can easily be determined. Let $(X_{\min}, Y_{\min})$ and $(X_{\max}, Y_{\max})$ be the coordinates of the left-bottom and right-top corners of the geo-referenced feature image, respectively, and the cell size of the grids (C_{csg}) be the spatial resolution. Thus, the width (W) and the height (H) of the geo-referenced feature image can be calculated as

$$\begin{cases} W = (X_{\max} - X_{\min})/C_{\text{csg}} \\ H = (Y_{\max} - Y_{\min})/C_{\text{csg}}. \end{cases} \tag{1}$$

Once the width and height of the geo-referenced image are determined, the mobile lidar point clouds can be located in each cell (see figure 2). To generate a geo-referenced feature image, the grey value of each cell has to be calculated.

The spatial distribution of mobile lidar point clouds implies the local geometric features of street-scene objects. Hence, the grey values of the geo-referenced feature image should reflect the spatial distribution of mobile lidar point clouds. The grey value of each cell of the geo-referenced feature image is determined by the inverse-distance-weighted (IDW) interpolation method that is widely used for DEM interpolation (Bater and Coops 2009). As the points in each cell are not 2.5-D but 3-D, the weight associated with each point should be carefully determined in order to reflect the spatial distribution. To fulfil the above requirement, we define the following rules to determine the weight associated with each point.

Rule 1: a point has a greater weight than another point if it is higher above the point of least height than the other point.

Rule 2: a point has a lighter weight than another point if the distance between the point and the centre of the cell is greater than that between the other point and the centre of the cell.

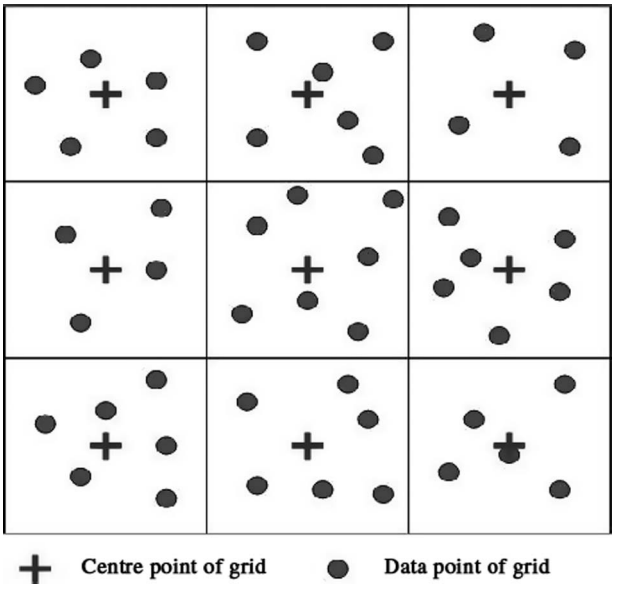


Figure 2. Sketch map of cells and data points.

These two rules indicate that the weight of each point is determined by two factors, namely planar distance and height difference. Once the weight of each point is determined, the grey value of each cell is calculated by the IDW interpolation method as follows:

$$F_{ij} = \left(\sum_{k=1}^{n_{ij}} W_{ijk} Z_{ijk}\right) \Big/ \left(\sum_{k=1}^{n_{ij}} W_{ijk}\right), \tag{2}$$

where F_{ij} is the weighted average of the feature value of the cell with coordinates (i, j), which will be displayed as grey values in the image, n_{ij} is the number of scanning points in the cell, W_{ijk} and Z_{ijk} are the weight and height of the kth point in this cell ($0 < k < n_{ij}$), respectively, and the weight, W_{ijk}, can be calculated by

$$W_{ijk} = \alpha W_{ijk}^{\mathrm{XY}} + \beta W_{ijk}^{\mathrm{H}}, \tag{3}$$

where α and β are the coefficients of weights, which need to be tuned, W_{ijk}^{XY} is the weight of the kth point in the cell, calculated using only the planar distance in the cell, and W_{ijk}^{H} is the weight of the kth point in the cell, calculated using only the height difference in the cell. They can be determined as follows:

$$\begin{cases} W_{ijk}^{\mathrm{XY}} = \sqrt{2} C_{\mathrm{csg}} \Big/ \left(D_{ij}^{k} + \delta\right) \\ W_{ijk}^{\mathrm{H}} = H_{ij}^{k} (h_{\min(ij)} - Z_{\min}) / (Z_{\max} - h_{\max(ij)} + \delta), \\ \alpha + \beta = 1.0 \end{cases} \tag{4}$$

where D_{ij}^{k} is the planar distance from the kth point in the cell with coordinates (i, j) to the centre of the cell, H_{ij}^{k} is the height difference from the kth point in the

cell to the lowest point of the cell, $h_{\min(ij)}$ and $h_{\max(ij)}$ are the lowest and highest elevations in the cell, respectively, $Z_{\max}$ and $Z_{\min}$ are the lowest and highest elevations in the whole scanning area, respectively, and δ is a positive infinitesimal value to make the denominator of the equation be nonzero. The weight W_{ijk}^{H} of the point calculated by the formula mentioned earlier is to emphasize high-lying objects and objects with a large standard deviation of height values. D_{ij}^{k} and H_{ij}^{k} are calculated by

$$\begin{cases} D_{ij}^{k} = \sqrt{\left(x_{ij}^{k} - x_{0}^{ij}\right)^{2} + \left(y_{ij}^{k} - y_{0}^{ij}\right)^{2}} \\ H_{ij}^{k} = Z_{ijk} - h_{\min(ij)} \end{cases}, \tag{5}$$

where (x_{ij}^{k}, y_{ij}^{k}) are the coordinates of the kth point in the cell with coordinates (i, j) and (x_{0}^{ij}, y_{0}^{ij}) are the coordinates of the centre of the cell.

The aforementioned equations demonstrate that the grey value of each cell of the geo-referenced feature image is co-determined by the planar distance values between the points and the centre of cell and the height differences between the points and the lowest height point of the cell. Figure 3 illustrates a simulation result of the weights of points W_{ijk} with the values of both α and β set to 0.5.

As shown in figure 3, a larger weight will be assigned to the points of higher elevation when the points have the identical coordinates of (x, y). Hence, a geo-referenced feature image is generated with the specified values of α and β. It is clear that the grey values of the geo-referenced feature image will change with different values of α and β. We will discuss the functionalities of α and β for the generation of the geo-referenced feature image in the experimental studies.

2.2 *Extraction of street-scene objects*

In light of the generation of the geo-referenced feature image, the areas with points of higher elevations have larger grey values in the geo-referenced feature image. Hence,

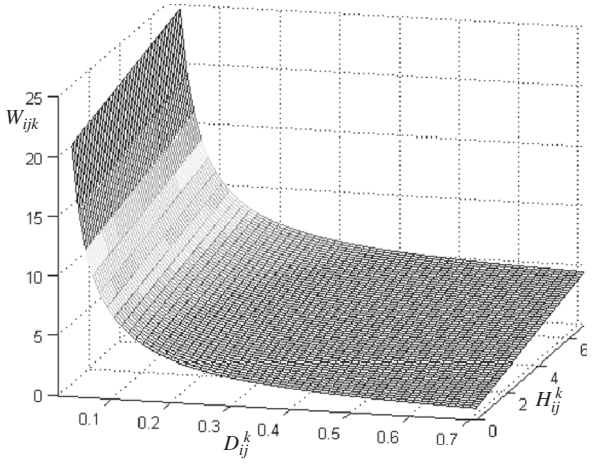
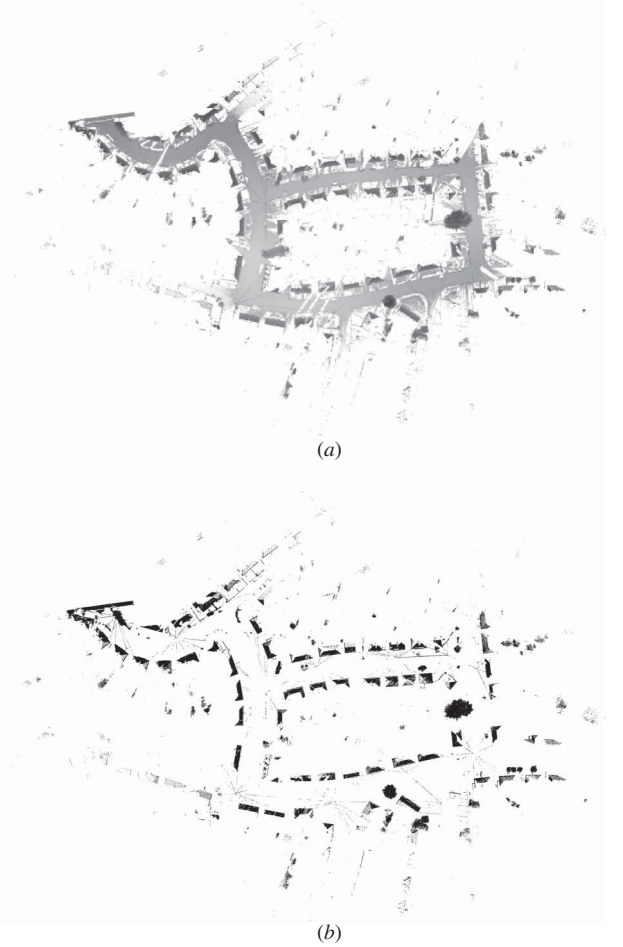


Figure 3. Simulation result of the weights of points.

the areas of buildings and trees have larger grey values compared with their neighbouring areas (e.g. ground and roads). A threshold of grey value can be specified to extract the areas of larger grey values according to the histogram analysis on the geo-referenced feature image. In this study, we selected the method of discrete discriminant analysis (Goldstein and Dillon 1978) to automatically specify the threshold because of its robustness. The method of discrete discriminant analysis first finds the maximum and minimum grey values of the feature image. Then, it calculates the between-cluster variance under an assumed threshold. Finally, it finds the threshold with the highest value of between-cluster variances as the optimal threshold, which is used to classify the pixels into different categories.

Figure 4 illustrates the segmentation results of the geo-referenced feature image generated from a data set of mobile lidar point clouds. To eliminate the noise in the segmented geo-referenced feature image and to extract street-scene objects (e.g. buildings and trees), we define the following operators.



(*a*)

(*b*)

Figure 4. (*a*) Generated geo-referenced feature image with $C_{csg} = 0.25$ m, $\alpha = 0.2$ and $\beta = 0.8$. (*b*) Segmentation of the geo-referenced feature image.

2.2.1 Boundary extraction. In the field of image processing, many methods have been proposed to extract contours in 2-D image space, including the most widely used Canny edge detector (Canny 1986) and the active contour model (Kass *et al*. 1988). As we deal with a segmented binary feature image (the background is black) here, we implement contour extraction in a simple way. A pixel is changed into white if it is a black pixel and its eight neighbouring pixels are black as well. In this way, the black pixels whose eight neighbouring pixels are not all black will be labelled as boundary pixels.

In order to trace contours, a contour-tracing algorithm proposed by Pavlidis (1982) was applied to identify each contour as a sequence of edge points. Figure 5 shows the result of contour extraction and tracing from a geo-referenced feature image.

2.2.2 Size constraint. As illustrated in figure 5, contours of noise with small sizes occur in the image. Considering the sizes of trees and buildings, the extracted boundary will be eliminated if its size is less than a threshold. The threshold is a parameter indicating the perimeter of a contour. Once the C_{csg} of one geo-referenced feature image is specified, the threshold indicates the length characteristics of the outlines of the geometric objects. Hence, this threshold should be tuned with a different C_{csg}. As all the contours are closed, we count the length of each contour as its size in our study. Figure 6 illustrates the result of size constraint with a threshold of 100 pixels.

2.2.3 Shape constraint. Generally, trees show an ellipse or circle shape and buildings show a rectangular shape. The compactness of the polygon shape can be applied to distinguish buildings and trees. We use the following equation from Touya (2007) to calculate the compactness (C_{sp}) of each contour shape.

$$C_{\mathrm{sp}} = 4\pi \frac{S_{\mathrm{contour}}}{P^2_{\mathrm{contour}}}, \tag{6}$$

Figure 5. The result of contour extraction and tracing.

Figure 6. Filtering shapes by size constraint (threshold = 100 pixels).

where S_{contour} and P_{contour} are the area and perimeter of a contour, respectively. According to equation (6), C_{sp} is 1.0 if the contour shows a circle shape. If the contour shows an elongation shape, C_{sp} is near 0.0. As a tree crown shows an approximate circle shape and a building shows an approximate rectangle shape, trees and buildings can be classified according to the shape constraint. Figure 7 shows the contours of extracted trees and buildings classified by the shape constraint.

Once the generated geo-referenced feature image is segmented and the above-mentioned three operators are invoked, the boundaries of street-scene objects (e.g. buildings and trees) can be extracted. Hence, the spatial extension of the extracted street-scene objects can be easily calculated according to the extracted boundaries associated. Therefore, the point clouds corresponding to each extracted object can

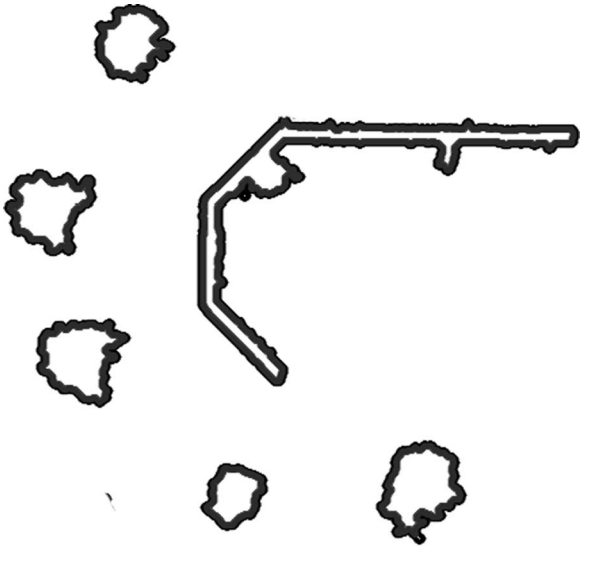


Figure 7. Example of classifying shapes by shape constraint.

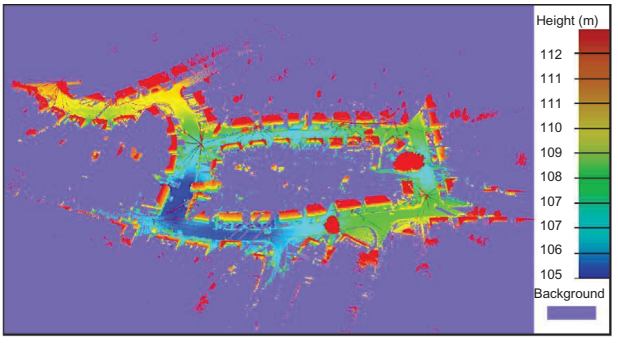
be extracted from the mobile lidar point clouds according to the spatial extension calculated.

Generally, the perimeter of a tree crown is larger compared with that of the tree trunk. The perimeters of buildings at different heights are usually kept at a constant value. In addition, the mean perimeter of a building in the Z direction is usually much larger than that of a tree. Hence, buildings and trees show different profiles in the Z direction, and this can be used to further classify buildings and trees from point clouds.

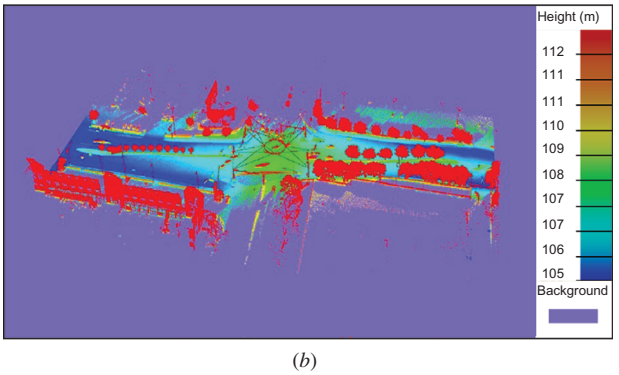
Given an unknown object with points, the area of a profile at a specified height (A_p) can be calculated with all the points lying in profile in the Z direction. These areas of profiles at different heights are used to compute the mean value (μ_A) of profile areas in the Z direction. Therefore, a threshold of μ_A can be set to distinguish buildings and trees. If μ_A is smaller than the specified threshold, this object will be classified as trees. Otherwise, this object will be considered a building.

3. Experimental studies

Two data sets captured by Optech's LYNX Mobile Mapper (Optech 2009), were selected for assessing the performance of the proposed method for extraction of different street-scene objects. Data set 1 consists of about 9 million points covering a 400 m × 350 m street scene, while there are 5.1 million points covering an area of about 140 m × 240 m for data set 2. The spatial span of point clouds is about 1–5 cm. Figure 8 illustrates the perspective views of the two data sets.



(*a*)



(*b*)

Figure 8. Perspective views of the two data sets: (*a*) data set 1 and (*b*) data set 2.

3.1 *Geo-referenced feature image generation*

In the proposed approach, a geo-referenced feature image is generated from mobile lidar point clouds before the street-scene object extraction operator is invoked. For the geo-referenced feature image generation, two parameters should be determined, namely, spatial resolution, also sometimes designated as C_{csg}, and distance weight, α (height difference weight, $\beta = 1 - \alpha$). It is clear that the two parameters have a vital effect on the visual effects of the geo-referenced feature image. To evaluate such effects, we generated a set of geo-referenced feature images with different values of the two parameters. As the spatial span of the mobile lidar point clouds is about 1–5 cm, we generated the geo-referenced feature image with cell sizes of 0.5 and 0.25 m, respectively. For each cell size, we specified the distance weights, α, as 0.0, 0.2, 0.4, 0.6, 0.8 and 1.0, respectively. Given $\alpha + \beta = 1$, the corresponding height difference weights, β, are 1.0, 0.8, 0.6, 0.4, 0.2 and 0.0, respectively. Figure 9 shows the geo-referenced feature images of data set 1 with the above values of the parameters. The left six images and right six images have cell sizes of 0.5 and 0.25 m, respectively.

As shown in figure 9, the areas of buildings (rectangles in figure 9), trees (ellipse in figure 9) and power lines (star in figure 9) have higher grey values compared with other areas (e.g. roads and ground). Moreover, the areas of buildings, trees and power lines have prominent boundaries that are well preserved in the feature images. Simultaneously, the differences between different street-scene objects are more notable. It implies that the generated geo-referenced feature image is able to well depict geometric features of street-scene objects and is suitable for extracting them. The results also show that the value of α has a great effect on the preservation of geometric features of street-scene objects in the geo-referenced feature image when a spatial resolution is specified. As illustrated in figure 9(*a*) ($\alpha = 0.0$, $\beta = 1.0$), the street-scene objects of higher elevations (e.g. power lines) in the geo-referenced feature image are visually prominent compared with those in the other geo-referenced feature images. This indicates that the α value is sensitive to the calculation of grey values. When the parameter α has a smaller value, the generated feature image enlarges the differences between the grey values. Hence, the parameter α should be given a smaller value (e.g. 0.2) for generating the geo-referenced feature image that is helpful for extracting street-scene objects that have higher elevations, such as power lines and roofs.

To comprehensively evaluate the effect of the α value on the feature image generation, we analysed the pixel-level differences of the feature images with the same cell size and different α values. For the six geo-referenced feature images generated with identical cell sizes, the total number of changed pixels, $N_{changed}$, between the two images and the sum of the grey values of the changed pixels, $\sum R_{ratio}$, was calculated for identical values of C_{csg}. Hence, the pixel-level difference (P_{pld}) between the two images is calculated by

$$P_{pld} = \sum R_{ratio} \Big/ N_{changed}. \tag{7}$$

Figures 10(*a*) and (*b*) illustrate the pixel-level differences of the geo-referenced feature images with cell sizes of 0.5 and 0.25 m, respectively.

The numbers 1, 2, 3, 4, 5 and 6 on the horizontal axis in figure 10 refer to the six groups of geo-referenced feature images generated with six parameter pairs ($\alpha = 0.0$, $\beta = 1.0$; $\alpha = 0.2$, $\beta = 0.8$; $\alpha = 0.4$, $\beta = 0.6$; $\alpha = 0.6$, $\beta = 0.4$; $\alpha = 0.8$, $\beta = 0.2$; $\alpha = 1.0$, $\beta = 0.0$), respectively. The vertical axis represents the value of the pixel-level difference

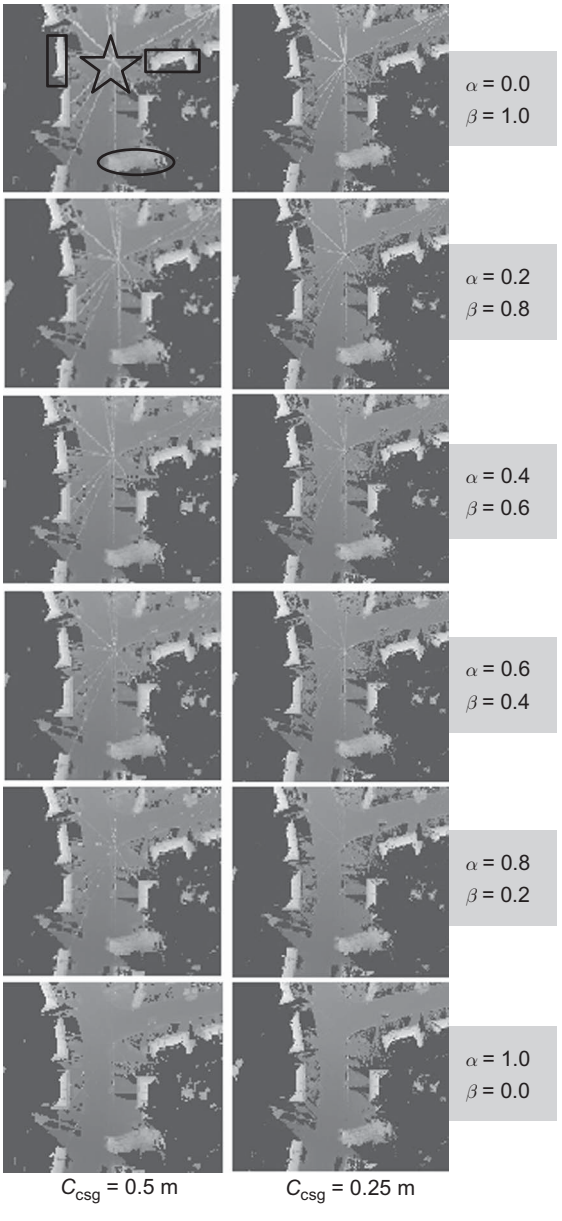


Figure 9. Data set 1: geo-referenced feature images with different α and C_{csg}. The rectangles, ellipse and star indicate buildings, trees and power lines, respectively.

(P_{pld}). It can be seen that the pixel-level differences between the feature images increase with differences in α values. The five red stars show the pixel-level differences between the feature image pairs of 1 and 2, 2 and 3, 3 and 4, 4 and 5 and 5 and 6, respectively. As the differences of α values between the pairs of six feature images are identical, it shows that the pixel-level differences between the geo-referenced feature images decline with the increase of the parameter α. This indicates that the height differences of points have a more important effect on the grey value determination of the geo-referenced feature images than the planar.

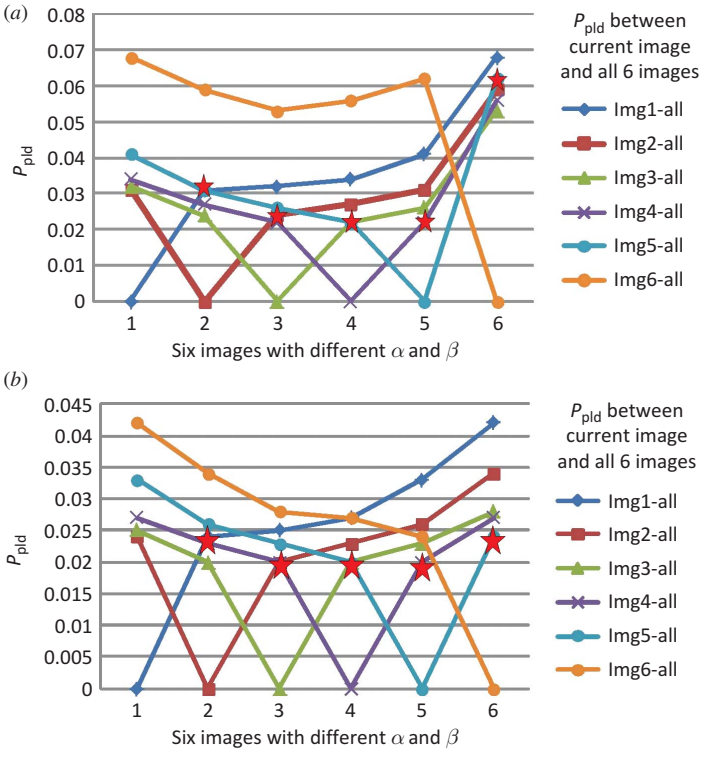


Figure 10. Data set 1: pixel-level differences (P_{pld}) of the geo-referenced feature images with (*a*) $C_{csg} = 0.5$ m and (*b*) $C_{csg} = 0.25$ m. The details of the parameter pairs corresponding to the numbers 1–6 are described in §3.1.

By comparing figures 10(*a*) and (*b*), we also found that the pixel-level difference between the geo-referenced feature images is proportional to the increase of the C_{csg} when the α value is fixed. Moreover, minor pixel-level differences between the feature images vary when a smaller cell size is specified. This indicates that the α value has a minor effect on the qualities of the geo-referenced feature image generated when a smaller cell size is specified. The feature image 6 was generated with the parameter of $\alpha = 1.0$, that is the grey values of the feature image 6 were calculated only by the weight of planar distance. It can be seen that the pixel-level difference between the feature image pair 5 and 6 has a sudden jump. This indicates that the grey values of the feature image should be co-determined by the weights of height difference and planar distance. The zero value of P_{pld} in figures 10(*a*) and (*b*) shows the zero pixel-level difference between each image and itself.

3.2 *Extraction of street-scene objects*

In light of the generation of the geo-referenced feature image and the analysis of the pixel-level difference between the geo-referenced feature images, the geo-referenced feature image is able to enlarge the differences of grey values if a small α value and a small cell size are specified. To validate the effectiveness of the proposed algorithm for street-scene object extraction, feature images of data sets 1 and 2 were generated

with the parameters of $\alpha = 0.2$, $\beta = 0.8$ and $C_{csg} = 0.25$ m. This configuration of parameters is beneficial for emphasizing high-lying objects.

The generated feature images were first segmented by the method of discrete discriminant analysis to maintain the point clouds of street-scene objects. Then, a contour extraction and tracing operator was invoked to extract the boundaries of street-scene objects. In our experiment, the thresholds of the size constraint were set to be 100 and 80 pixels for extracting street-scene objects in data sets 1 and 2, respectively. Then, the thresholds of the shape constraint were specified as 0.4 and 0.6 to classify buildings and trees from the extracted street-scene objects in data sets 1 and 2, respectively. Generally, the mean diameter of a tree from the tree crown to the tree trunk in a street-scene environment is less than 5 m. With this prior information, it is reasonable to set the mean area μ_A of tree profiles in the Z direction to be 20 m^2, which is calculated as $\pi \times 2.5^2$. Therefore, for Z-direction profile analysis, the threshold of μ_A was specified as 20 m^2 for both data sets. The spatial extension of each object can thus be calculated according to the corresponding boundary extracted. Hence, the point clouds corresponding to each object can be extracted from the mobile lidar point clouds according to the spatial extension calculated. Figures 11 and 12 show the results of street-scene objects extracted from data sets 1 and 2, respectively. The extracted point clouds of buildings can be used for further building facade reconstruction.

To evaluate the sensitiveness of the size constraint and the shape constraint, data set 1 was tested with the size constraints of 80, 100 and 120 pixels. Data set 2 was

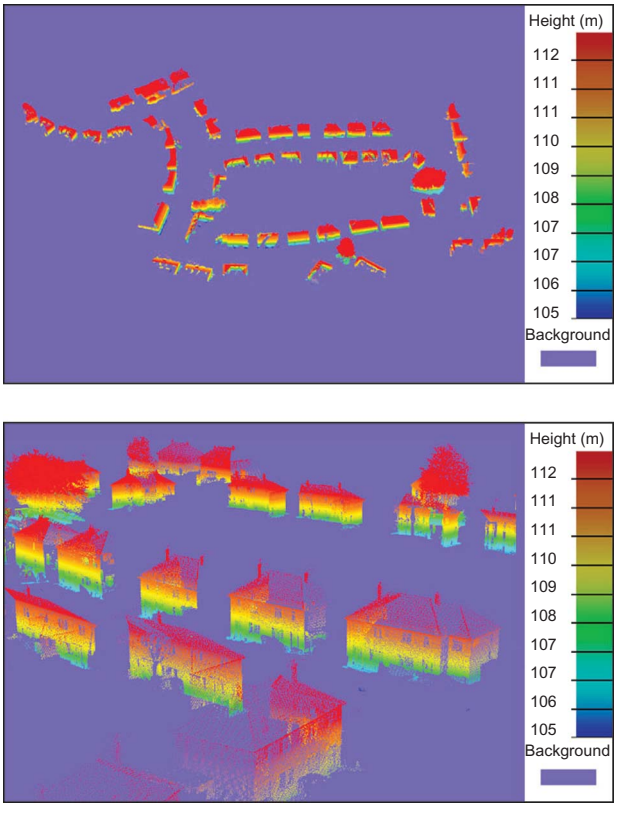


Figure 11. Data set 1: (*a*) top view and (*b*) perspective view of all the extracted objects.

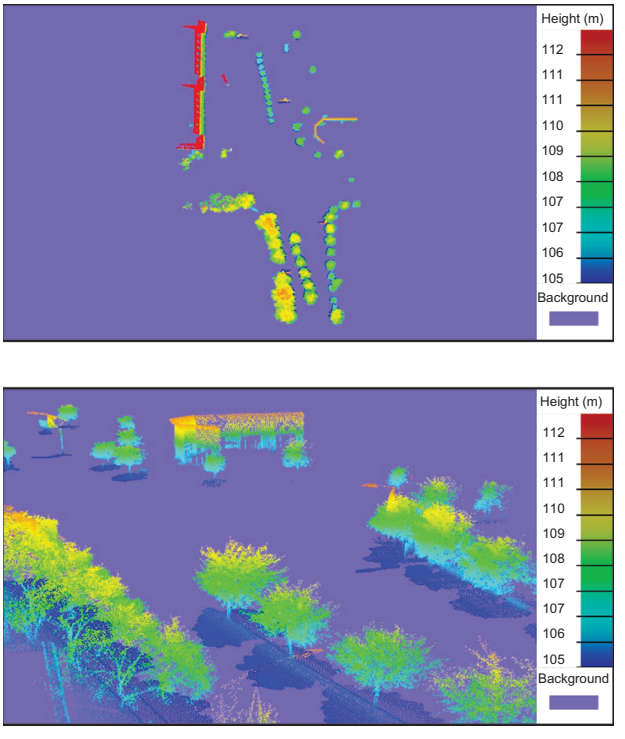


Figure 12. Data set 2: (*a*) top view and (*b*) perspective view of all the extracted objects.

tested with the size constraints of 60, 80 and 100 pixels. Different thresholds of the shape constraint were specified corresponding to the size constraint for both data sets. The percentage of missing buildings or trees and the percentage of falsely classified buildings or trees were defined to evaluate the proposed method.

The percentage of missing buildings or trees (E_m) is the ratio of the number of missing or undetected buildings or trees (N_m) to the total number of ground-truth buildings or trees (N_T). The missing buildings or trees result from considering buildings or trees as non-buildings or non-trees. As a result, the percentage of missing buildings or trees can be calculated as

$$E_m = N_m/N_T. \tag{8}$$

The percentage of falsely classified buildings or trees (E_f) corresponds to the ratio between the number of falsely classified buildings or trees (N_f) and the total number of classified buildings or trees (N_c). The falsely classified buildings or trees result from considering non-buildings or non-trees as buildings or trees. Likewise, the percentage of falsely classified buildings or trees can be calculated as

$$E_f = N_f/N_c. \tag{9}$$

Table 1 lists the E_m and the E_f of both data sets by the constraints of size and shape and by the size constraint and Z-direction profile analysis and lists the correctly classified buildings (denoted by B) and trees (denoted by T) by the shape constraint and the

Table 1. Percentages of missing and falsely classified buildings or trees for both data sets (B = buildings, T = trees).

	Total number of ground-truth buildings or trees (N_T)		Constraints on size and shape		Classification result by shape constraint						Correctly classified objects by shape constraint			
					Number of classified buildings or trees (N_c)		Percentage of missing buildings or trees (E_m)		Percentage of falsely classified buildings or trees (E_f)		Number of correctly classified buildings or trees		Percentage of correctly classified buildings or trees	
	B	T	Size (pixels)	Shape (C_{sp})	B	T	B	T	B	T	B	T	B	T
Data set 1	60	14	80	0.40	74	18	20.0	57.1	35.1	66.7	48	6	80.0	42.9
				0.45	80	12	11.7	64.3	33.8	58.3	53	5	88.3	35.7
				0.50	88	4	1.7	78.6	33.0	25.0	59	3	98.3	21.4
				0.55	92	0	0.0	100.0	34.8	0.0	60	0	100.0	0.0
			100	0.40	64	15	16.7	64.3	21.9	66.7	50	5	83.3	35.7
				0.45	69	10	10.0	71.4	21.7	60.0	54	4	90.0	28.6
				0.50	75	4	1.7	78.6	21.3	25.0	59	3	98.3	21.4
				0.55	79	0	0.0	100.0	24.1	0.0	60	0	100.0	0.0
			120	0.40	58	9	11.7	85.7	8.6	77.8	53	2	88.3	14.3
				0.45	62	5	5.0	85.7	8.1	60.0	57	2	95.0	14.3
				0.50	66	1	0.0	92.9	9.1	0.0	60	1	100.0	7.1
				0.55	67	0	0.0	100.0	10.4	0.0	60	0	100.0	0.0
Data set 2	2	47	60	0.40	50	57	0.0	38.3	96.0	49.1	2	29	100.0	61.7
				0.50	68	39	0.0	44.7	97.1	33.3	2	26	100.0	55.3
				0.60	77	30	0.0	48.9	97.4	20.0	2	24	100.0	51.1
				0.70	90	17	0.0	68.1	97.8	11.8	2	15	100.0	31.9
			80	0.40	46	40	0.0	40.4	65.7	30.0	2	28	100.0	59.6
				0.50	59	27	0.0	46.8	96.6	7.4	2	25	100.0	53.2
				0.60	62	24	0.0	51.1	96.8	4.2	2	23	100.0	48.9
				0.70	72	14	0.0	70.2	97.2	0.0	2	14	100.0	29.8
			100	0.40	42	34	0.0	42.6	95.2	20.6	2	27	100.0	57.4
				0.50	51	25	0.0	48.9	96.1	4.0	2	24	100.0	51.1
				0.60	53	23	0.0	53.2	96.2	4.3	2	22	100.0	46.8
				0.70	63	13	0.0	72.3	96.8	0.0	2	13	100.0	27.7

(*Continued*)

Table 1. (*Continued*.)

	Classification result by Z-direction profile analysis						Correctly classified objects by Z-direction profile analysis			
	Number of classified buildings or trees (N_c)		Percentage of missing buildings or trees (E_m)		Percentage of falsely classified buildings or trees (E_f)		Number of correctly classified buildings or trees		Percentage of correctly classified buildings or trees	
	B	T	B	T	B	T	B	T	B	T
Data set 1	64	16	5.0	21.4	10.9	31.3	57	11	95.0	78.6
	60	15	6.7	21.4	6.7	26.7	56	11	93.3	78.6
	56	13	11.7	35.7	5.4	30.8	53	9	88.3	64.3
Data set 2	6	52	0.0	2.1	66.7	11.5	2	46	100.0	97.9
	4	48	0.0	4.3	50.0	6.3	2	45	100.0	95.7
	3	46	0.0	8.5	33.3	6.5	2	43	100.0	91.5

Z-direction profile analysis. The total number of ground-truth buildings (N_{Tb}) and trees (N_{Tt}) were manually counted for both data sets. It can be seen that data set 1 contains more buildings (60 buildings and 14 trees) and data set 2 contains more trees (2 buildings and 47 trees). As listed in table 1, the E_{m} and the E_{f} of data sets 1 and 2 were high when only the size and the shape of constraints were used. It can be seen that the threshold of the shape constraint has a large effect on the E_{m} and the E_{f}. The percentages of missing buildings (E_{mb}) and falsely classified trees (E_{ft}) decline with the increase of the threshold of the shape constraint according to the shape constraint. However, the percentages of missing trees (E_{mt}) and falsely classified buildings (E_{fb}) show the opposite trend. This dilemma leads to difficulties in specifying the threshold of the shape constraint. The percentage of missing trees (E_{mt}) in data set 2 is 70.2% with a shape constraint of 0.7 and a size constraint of 80 pixels. This is because many trees and buildings have almost the same outlines, especially when the trees are dense. However, the results were substantially improved when an additional Z-direction profile analysis was used. It can be seen that the percentage of missing trees (E_{mt}) using Z-direction profile analysis increases with the increase of the threshold of the size constraint, while the percentage of falsely classified buildings (E_{fb}) declines. As the threshold of μ_{A} is related to the spatial extension of an object, the percentages of missing buildings and falsely classified trees will be higher with a larger threshold of μ_{A}. The larger the threshold of μ_{A} that is specified, the more buildings will be classified as trees, which leads to the increase of the percentages of missing buildings and falsely classified trees.

3.3 *Comparison with DoPP algorithm and FCM clustering*

To demonstrate the efficiency and effectiveness of the proposed algorithm implemented in C^{++} language, the DoPP algorithm and the fuzzy C-means (FCM) clustering (Biosca and Lerma 2008) were compared in our experiment.

The DoPP algorithm divides the mobile lidar points into cells with a certain size in the XOY plane. Then, the threshold of the minimum number of points in one cell is specified for finding those cells which have the number of points more than the specified threshold. The points located in those cells are considered to belong to building facade points. The FCM clustering method (Biosca and Lerma 2008) consists of two steps, namely the classification of data points into surface categories and the classification of points lying on planar features into different planes according to the feature vector, which was calculated by the distance from a point to its best-fitting plane along its normal direction. In our study, the feature vector of a point was used to classify the point clouds into planar surfaces, undulating surfaces and the rest.

The proposed method, the DoPP algorithm and the FCM clustering method were tested on a computer with 1 Gb RAM and a CPU of 1.8 GHz. Figure 13(*a*) shows the buildings extracted with the proposed method. Figure 13(*b*) shows the extracted facade points using a DoPP algorithm with the cell size of 0.25 m and the minimum number of points of 150 in one cell. Figure 13(*c*) shows the classification results of the FCM clustering method, where the whole point clouds were classified into planar surfaces, undulating surfaces and the rest. The number of iterations required for the convergence was set to be 500. The points in blue are planar features, the points in green are undulating surfaces and the other points are shown as red dots. It can be seen that our proposed method achieved a better result.

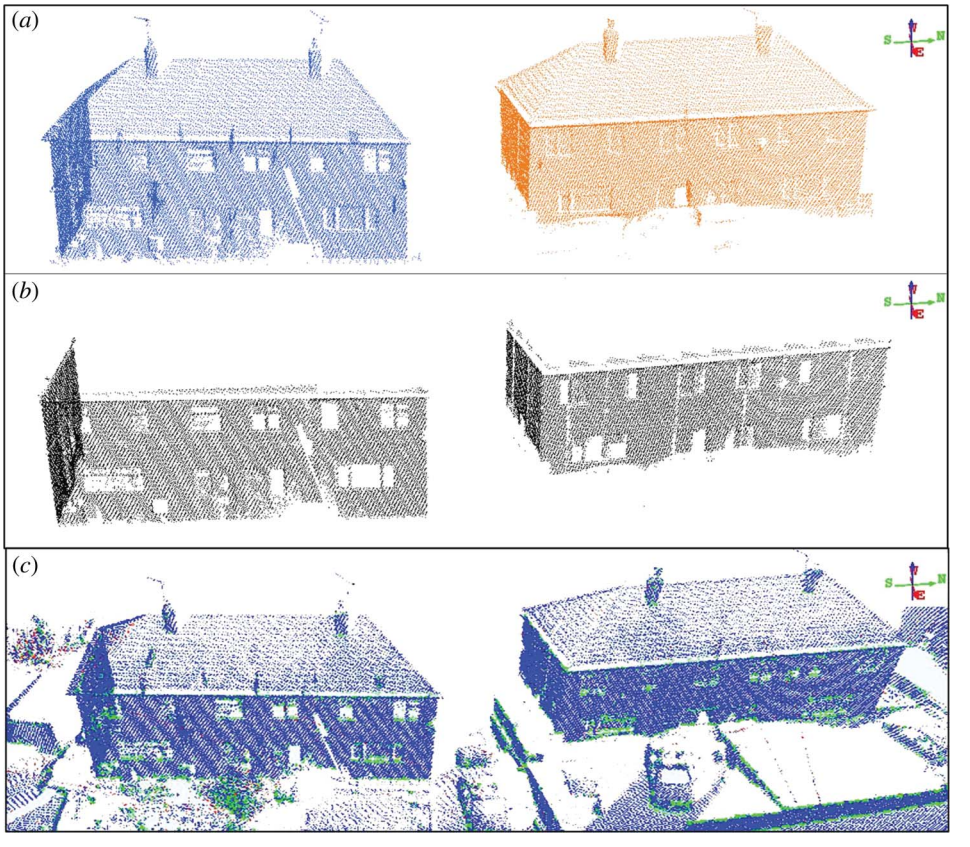


Figure 13. Object extraction by (*a*) the proposed method, (*b*) the DoPP algorithm and (*c*) FCM clustering.

The time complexity of an algorithm is commonly expressed using big O notation. For geo-referenced feature image generation in the proposed method, it takes $O(N)$ time to compute the feature values of grid cells, where N is the number of data points. Given the large volume of mobile lidar data, the time cost in the image processing procedure is negligible. According to the principle of the DoPP algorithm, it also takes $O(N)$ time to extract grid cells of building facades. Likewise, it takes $O(kN\log_2 N + N)$ time to run k-nearest neighbour searching and compute the feature vector of each point for the FCM clustering method. For the further process of implementing FCM clustering, its time complexity is $O(NcI)$, where c is the number of clusters and I is the number of iterations required for convergence. Table 2 lists the time costs of the three methods.

It can be seen that our proposed method has almost the same time efficiency as that of the DoPP algorithm. They both have better time performance than the FCM clustering method. However, the DoPP algorithm only extracts facade points as a whole rather than as an individual building facade. Therefore, classifying the extracted facade points into different buildings needs further tasks. As shown in figure 13(*c*), the FCM clustering method classified points into different categories according to the feature vector defined. However, it is difficult to separate ground points from building points because of similar planar feature vectors of ground and building facades.

Table 2. Time efficiency comparison of the three methods.

Method	Procedure ($C_{csg} = 0.25$ m)	Processing time (s)
DoPP algorithm	Counting points in grids	2.095
	Building facade extraction	26.140
FCM clustering	Feature vector calculation of each point	1823.52
	Classification of all points using FCM	609.66
The proposed method	Generation of feature image	3.701
	2-D object classification	0.140
	3-D street-scene object extraction	24.180

Note: DoPP, density of projected points; FCM, fuzzy C-means.

Our proposed method is capable of classifying the extracted points into different objects (e.g. buildings). It also demonstrates that the proposed method maintains the local geometric features of street-scene objects. For instance, the point clouds of building roofs and tree crowns are well preserved in the extracted point clouds, which is particularly useful for further building facade reconstruction with high levels of detail.

4. Discussion

The extraction of street-scene objects from mobile lidar point clouds is quite a time-consuming job due to huge data volumes and a lack of explicit geometric features. This article proposes a promising solution for efficiently extracting point clouds of street-scene objects from mobile lidar point clouds. The main contributions of the proposed method are the generation of the geo-referenced feature image first followed by extracting street-scene objects from the feature image. Thus, the proposed method transforms the extraction of street-scene objects from 3-D point clouds to object extraction from 2-D imagery. We have presented a formula to calculate the grey values of the geo-referenced feature image with a specified cell size of the grid (e.g. 25 cm). Image segmentation and contour extraction techniques were applied to extract the boundaries of street-scene objects in image space. Hence, the point clouds of the street-scene objects (e.g. buildings and trees) can be extracted in 3-D according to the boundaries associated.

The calculation of the grey values of the geo-referenced feature image demonstrates that the generated feature image is of vital importance for efficiently extracting street-scene objects from mobile lidar point clouds. Moreover, two parameters, namely the cell size of the feature image and the weights of points, determine the qualities of the feature images generated. The cell size of the grid can be obtained from the acquisition parameters, such as the scanning frequency and average span between the scanned points. The weights of points of each cell consist of height differences with the minimum height of the whole area and the planar distances to the centre of the cell. The analysis of pixel-level differences between the generated feature images shows that the height differences of points play a more important role for the weights of points compared with the planar distance values. That is why we assign a larger proportion to the height differences part for the determination of the weights of points. This is also helpful for enlarging the grey-value differences of the geo-referenced feature image, which is useful for extracting the boundaries of high-lying street-scene objects. On the other hand, an α value is easily tuned to generate the geo-referenced feature images

with a given cell size for different purposes. For example, the geo-referenced feature image can be generated with a low α value for enlarging the grey values of the area of higher elevations, which is beneficial for the extraction of power lines, as shown in figure 9 (dotted in star).

In our method, the threshold of the size constraint differs from objects to be extracted and is affected by the cell size of grids. It may be difficult to automatically adjust the threshold of the size constraint for different street-scene objects without prior knowledge. On the other hand, the proposed method calculates the compactness of each extracted object as a shape constraint to classify buildings and tree crowns. It can be seen that the shape constraint has a large effect on the two types of errors. Moreover, when compared with the size constraint, the shape constraint is more difficult to specify. Hence, many non-trees and non-buildings were misclassified as trees and buildings. Nevertheless, the percentages of missing buildings or trees and falsely classified buildings or trees were greatly reduced by 50% and 70%, respectively, with an additional Z-direction profile analysis. This is because the profiles of buildings and trees in the Z-direction are different. With prior knowledge of the diameter of the tree crown, the threshold of the mean area of tree profiles was easily specified for the Z-direction profile analysis.

5. Conclusions

Automated extraction of street-scene objects from mobile lidar point clouds remains a major unsolved problem in the use of mobile lidar mapping systems. A key issue is to identify geometric features (e.g. the boundary of a building) from 3-D lidar point clouds. This article has presented a novel solution for efficiently extracting street-scene objects (e.g. buildings and trees) from point clouds collected by Optech's LYNX Mobile Mapping System. Our method consists of three key steps: (1) determination of the weights of points, (2) generation of a geo-referenced feature image and (3) image segmentation towards object extraction from the feature image. The geo-referenced feature image generated from mobile lidar point clouds is able to represent the spatial distribution of scanned points and preserve the local geometric features of street-scene objects. The experimental results demonstrate that our method is capable of generating the geo-referenced feature image, in which the geometric features of street-scene objects are well preserved with the two parameters of the cell size and point weights. Moreover, these two parameters can easily be tuned for different purposes. In addition, our method demonstrates high computational efficiency for the extraction of point clouds of street-scene objects, which is beneficial for further reconstruction of building facades. Our further studies will focus on vector-based 3-D construction of street-scene objects based on the point clouds of the extracted building.

Acknowledgements

The work presented in this article was substantially supported by NSFC (No. 41071268), the Outstanding Talented Project from the Ministry of Education of PR China (NCET-07-0643) and the Fundamental Research Funds for the Central Universities (No. 3103005). The LYNX Mobile Mapper data sets were provided by Opech Inc. The authors thank the anonymous reviewers for their insightful comments and suggestions.

References

Abuhadrous, I., Ammoun, S., Nashashibi, F., Goulette, F. and Laurgeau, C., 2004. Digitizing and 3D modeling of urban environments and roads using vehicle-borne laser scanner system. In *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, 28 September–2 October 2004, Sendai, Japan (Piscataway, NJ: IEEE Press), vol. 1, pp. 76–81.

Akel, N.A., Filin, S. and Dovtsher, Y., 2009, Reconstruction of complex shape buildings from lidar data using free form surfaces. *Photogrammetric Engineering & Remote Sensing*, **75**, pp. 271–280.

Axelsson, P., 2000, DEM generation from laser scanner data using adaptive TIN models. *International Archives of the Photogrammetry Remote Sensing and Spatial Information Sciences*, **33**, pp. 110–117.

Barber, D., Mills, J. and Smith, V.S., 2008, Geometric validation of a ground-based mobile laser scanning system. *ISPRS Journal of Photogrammetry and Remote Sensing*, **63**, pp. 128–141.

Bater, C.W. and Coops, N.C., 2009, Evaluating error associated with lidar-derived DEM interpolation. *Computers & Geosciences*, **35**, pp. 289–300.

Becker, S., 2009, Generation and application of rules for quality dependent façade reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, **64**, pp. 640–653.

Becker, S. and Haala, N., 2007, Combined feature extraction for façade reconstruction. In *Proceedings of the ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007*, 12–14 September 2007, P. Rönnholm, H. Hyyppä and J. Hyyppä (Eds.), Espoo, Finland (Laser Scanning 2007 and SilviLaser 2007 Organizing Committee (ISPRS)), pp. 44–49.

Biosca, J.M. and Lerma, J.L., 2008, Unsupervised robust planar segmentation of terrestrial laser scanner point clouds based on fuzzy clustering methods. *ISPRS Journal of Photogrammetry & Remote Sensing*, **63**, pp. 84–98.

Brovelli, M.A. and Cannata, M., 2004, Digital terrain model reconstruction in urban areas from airborne laser scanning data: the method and the example of the town of Pavia (Northern Italy). *Computers & Geosciences*, **30**, pp. 325–331.

Canny, J.F., 1986, A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **8**, pp. 679–698.

Goldstein, M. and Dillon, W.R., 1978, *Discrete Discriminant Analysis* (New York: Wiley).

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

Haala, N., Peter, M., Cefalu, A. and Kremer, J., 2008a, Mobile lidar mapping for urban data capture. In *VSMM 2008 – Conference on Virtual Systems and MultiMedia Dedicated to Digital Heritage*, 20–25 October 2008, D. Pitzalis (Ed.), Limassol, Cyprus (VSMM 2008 Organizing Committee), pp. 101–106.

Haala, N., Peter, M., Kremer, J. and Hunter, G., 2008b, Mobile lidar mapping for 3D point cloud collection in urban areas – a performance test. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, **37**, pp. 1119–1124.

Hunter, G., Cox, C. and Kremer, J., 2006, Development of a commercial laser scanning mobile mapping system – StreetMapper. In *2nd International Workshop on the Future of Remote Sensing*, 17–18 October 2006, Antwerp, Belgium.

Jaakkola, A., Hyyppä, J., Hyyppä, H. and Kukko, A., 2008, Retrieval algorithms for road surface modeling using laser-based mobile mapping. *Sensors*, **8**, pp. 5238–5249.

Kass, M., Witkin, A. and Terzopoulos, D., 1988, Snakes: active contour models. *International Journal of Computer Vision*, **1**, pp. 321–331.

Kraus, K. and Pfeifer, N., 1998, Determination of terrain models in wooded areas with airborne laser scanner data. *ISPRS Journal of Photogrammetry and Remote Sensing*, **53**, pp. 193–203.

Kremer, J. and Hunter, G., 2007, Performance of the StreetMapper mobile LIDAR mapping system in 'real world' projects. In *Photogrammetric Week '07*, 2007 (Heidelberg: Wichmann), pp. 215–225.

Kukko, A., Andrei, C.O., Salminen, V.M., Kaartinen, H., Chen, Y., Ronnholm, P., Hyyppä, H., Hyyppä, J., Chen, R., Haggren, H., Kosonen, I. and Capek, K., 2007, Road environment mapping system of the Finnish Geodetic Institute-FGI ROAMER. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, **36**, pp. 241–247.

Li, B.J., Li, Q.Q., Shi, W.Z. and Wu, F.F., 2004, Feature extraction and modeling of urban building from vehicle-borne laser scanning data. In *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 12–23 July 2004, O. Altan (Ed.), Istanbul, Turkey (The Organizing Committee of XXth ISPRS Congress), pp. 934–940.

Manandhar, D. and Shibasaki, R., 2001, Vehicle-borne laser mapping system (VLMS) for 3-D GIS. *Geoscience and Remote Sensing Symposium*, **5**, pp. 2073–2075.

Manandhar, D. and Shibasaki, R., 2002, Auto-extraction of urban features from vehicle-borne laser data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, **34**, pp. 650–655.

Mass, H. and Vosselman, G., 1999, Two algorithms for extracting building models from raw laser altimetry data. *ISPRS Journal of Photogrammetry & Remote Sensing*, **54**, pp. 153–163.

Mrstik, P. and Kusevic, K., 2009, Real time 3D fusion of imagery and mobile lidar. In *ASPRS 2009 Annual Conference*, 9–13 March 2009, Baltimore, MD.

Optech Inc., 2009, LYNX Mobile Mapper. Available online at: http://www.optech.ca/pdf/LynxDataSheet.pdf (accessed 10 December 2009).

Overby, J., Bodum, L., Kjems, E. and Iisoe, P.M., 2004, Automatic 3D building reconstruction from airborne laser scanning and cadastral data using Hough transform. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, **35**, pp. 296–301.

Pavlidis, T., 1982, *Algorithms for Graphics and Image Processing* (Potomac, MD: Computer Science Press).

Pu, S. and Vosselman, G., 2009, Knowledge based reconstruction of building models from terrestrial laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing*, **64**, pp. 575–584.

Schwarz, K.P. and El-Sheimy, N., 2007, Digital mobile mapping systems – state-of-the-art and future trends. In *Advances in Mobile Mapping Technology*, V. Tao and J. Li (Eds.), pp. 3–18 (London: Taylor & Francis).

Sithole, G. and Vosselman, G., 2004, Experimental comparison of filter algorithms for bare-earth extraction from airborne laser scanning point clouds. *ISPRS Journal of Photogrammetry & Remote Sensing*, **59**, pp. 85–101.

StreetMapper, 2010, StreetMapper | Mobile Laser Mapping. Available online at: http://www.streetmapper.net (accessed 23 April 2010).

Tao, V. and Li, J. (Eds.), 2007, *Advances in Mobile Mapping Technology* (London: Taylor & Francis).

Toth, C.K., 2009, R&D of mobile lidar mapping and future trends. In *ASPRS 2009 Annual Conference*, 9–13 March 2009, Baltimore, MD.

Touya, G., 2007, A road network selection process based on data enrichment and structure detection. In *10th ICA Workshop on Generalization and Multiple Representation*, 2–3 August 2007, Moscow.

Zampa, F. and Conforti, D., 2009, Mapping with mobile lidar. *GIM International*, **23**, pp. 35–37.

Zhao, H. and Shibasaki, R., 2005, Updating digital geographic database using vehicle-borne laser scanners and line cameras. *Photogrammetric Engineering & Remote Sensing*, **71**, pp. 415–424.