

A FAST AND ROBUST ALGORITHM FOR ROAD EDGES EXTRACTION FROM LIDAR DATA

Kaijin Qiu^a, Kai Sun^a, Kou Ding^a, Zhen Shu^a

^a Ledor Spatial Information Technology Co., Ltd. Building No.12, HUST Scien &Tech Park, East Lake Hi-Tech Zone, Wuhan, China - (choukaijin, sunkai, dingkou, shuzhen)@ledor.com.cn

Commission V, WG V/4

KEY WORDS: Mobile Mapping Systems (MMS), LiDAR data, Point clouds, Road edges extraction, Rough plane, Refined plane

ABSTRACT:

Fast mapping of roads plays an important role in many geospatial applications, such as infrastructure planning, traffic monitoring, and driver assistance. How to extract various road edges fast and robustly is a challenging task. In this paper, we present a fast and robust algorithm for the automatic road edges extraction from terrestrial mobile LiDAR data. The algorithm is based on a key observation: most roads around edges have difference in elevation and road edges with pavement are seen in two different planes. In our algorithm, we firstly extract a rough plane based on RANSAC algorithm, and then multiple refined planes which only contains pavement are extracted from the rough plane. The road edges are extracted based on these refined planes. In practice, there is a serious problem that the rough and refined planes usually extracted badly due to rough roads and different density of point cloud. To eliminate the influence of rough roads, the technology which is similar with the difference of DSM (digital surface model) and DTM (digital terrain model) is used, and we also propose a method which adjust the point clouds to a similar density to eliminate the influence of different density. Experiments show the validities of the proposed method with multiple datasets (e.g. urban road, highway, and some rural road). We use the same parameters through the experiments and our algorithm can achieve real-time processing speeds.

1. INTRODUCTION

In recent years, laser terrestrial Mobile Mapping Systems (MMS) have been developed for the digital mapping. The applicability of MMS are various (road control, urban security, virtual world, entertainment, etc.) due to its performance, contactless, high precision and cost efficient 3D data acquisition capability. Laser scanners onboard terrestrial based moving vehicles are an important part of MMS, and widely used in collecting high quality 3D data. The 3D data can assist road authorities in effective management of road networks and ensure maximum safe driving condition for road users. Extracting road from the 3D data is a key step to implement these applications. Obviously, the road edge is a fundamental feature and its efficient extraction is a challenging research topic due to various road and its boundaries (e.g. Figure 1).



Figure 1. Various road and its boundaries

We present a fast and robust algorithm which has been developed for extracting road edges in multiple road environments from terrestrial mobile light detection and ranging (LiDAR) data. Section 2 describes the research background and previous road edges extraction methods from LiDAR data. Section 3 describes

our road edge extraction algorithm and how to solve different practical problems. We test our road edge extraction algorithm on different types of roads and the experimental results are listed and analysed in Section 4. Conclusions are discussed in Section 5.

2. BACKGROUND

Road extraction have been examined by many studies from airborne and terrestrial LiDAR data. The methods are mostly based on the identification of planar or smooth surfaces and their attributes, e.g. height differences, surface normal vector variations, and intensities. Clode et al. (2004) presented a hierarchical classification technique to classify the LiDAR points into road or non-road from airborne laser scanner data. The algorithm dealt with using as much of the recorded laser information as possible including height and intensity. Yuan et al. (2008) proposed a road-surface abstraction algorithm which used fuzzy cluster method to cluster points and abstracted straight lines by their location and slope angle. A current referenced horizontal could be acquired by comparing several continuous scan lines and then the algorithm abstracted obstacles on road-surface area. Lam et al. (2010) achieved road extraction in 3D by modelling the road as a dynamic system of connected planes, computed by plane fitting of local point clouds. Smadja et al. (2010) presented a two-step method, which was based on the local road shape. Different uses of the RANSAC algorithm were employed, for road sides rough estimation in the first place, then for unlikely candidates elimination. Zhang (2010) proposed a LiDAR-based road and road-edge detection method. The road candidate region was identified by filtering techniques, and the candidate region was determined by pattern recognition techniques. The identified 3D road edge points were then projected on a ground plane to estimate the road kerbs. McElhinney et al. (2010) detailed a LiDAR based road edge

extraction algorithm which is applicable to both urban and rural road sections with geospatial data. Ibrahim and Lichti (2012) presented an approach to the detection and extraction of the road curb and the street floor from unorganized 3D point clouds captured by Mobile terrestrial laser scanners. The pipeline consisted of a sequence of five steps: organizing the input irregular 3D point cloud and nearest neighbour search; density based segmentation to extract the ground segment; morphological analysis to refine the ground segment; derivative of Gaussian to detect the curb; solving the travelling salesman problem to form a closed polygon of the curb and point-in-polygon test to extract the street floor. Kumar et al. (2013) presented an algorithm to extract left and right road edges from terrestrial mobile LiDAR data. The algorithm was based on a combination of two modified versions of the parametric active contour or snake model.

However, most of the above road extraction approaches have been developed for urban road environments, and these methods fail to provide an efficient and robust solution due to poor performance and parameter-sensitivity. In the next section, we describe our road edge extraction algorithm which attempts to address the limitations of these approaches.

3. ROAD EDGE EXTRACTION ALGORITHM

The proposed method consists of a sequence of steps. The framework of this method is illustrated in Figure 2. There are four key steps: Partitioning point clouds according to the GPS time of points and vehicle trajectory; Pre-processing point clouds to denoise and eliminate the influence of rough roads; Extracting candidate road edge points by detecting different plane using RANSAC algorithm; Refining road edge by global consistency. The following sections describe the steps in details.

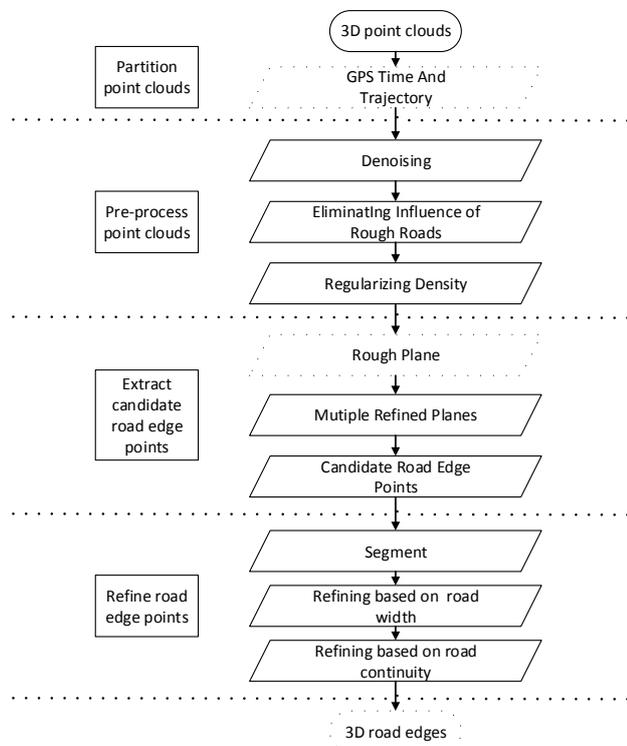


Figure 2. Framework of extracting road edge

3.1 Point Clouds Partition

Terrestrial LiDAR point clouds can't be processed all at once due to a huge number of points and very large area. Pu et al. (2011) partitioned point clouds according to vehicle trajectory data. Although vehicle trajectory data provide a feasible solution for extracting the approximate road areas, the road edge extraction may cause problems when the vehicle several times passes the same road or neighbouring several different roads. We propose to partition point clouds into consecutive sections based on the vehicle trajectory and Navigation Satellite Timing and Ranging Global Position (GPS Time). The length and width of a section and the time difference between the consecutive points are manually specified (e.g. 200 m long and 50 m wide). Figure 3 shows that the proposed method divides the point clouds into a set of sequential sections.

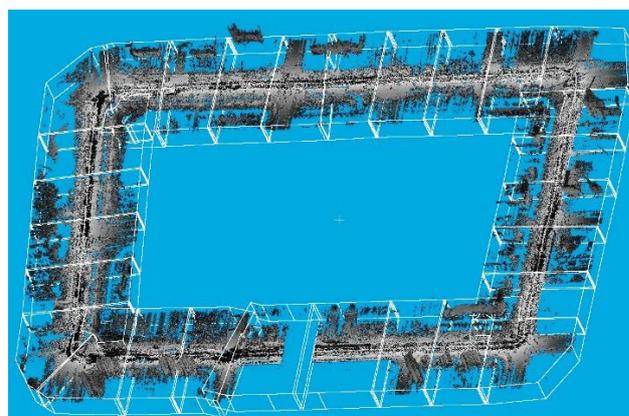


Figure 3. Partitioning terrestrial LiDAR point clouds into overlapping sequential sections

3.2 Pre-processing

Terrestrial LiDAR point clouds may contain too much points, which affect performance. These points include outlier points and non-uniform density points. We adopt the method proposed by Rusu et al. (2008), which uses point neighbourhood statistics to filter outlier data. Further, the algorithm of down-sampling is used to adjust the point clouds to a similar density to eliminate the influence of different density.

3.3 Candidate Road Edge Points Extracting

Smadja et al. (2010) assumed that there exists a main polynomial corresponding to the road, and the RANSAC algorithm with two thresholds is used to approximate the road by two parabolas. The first parabola indicates the expected outliers rate in the points set, which is directly related to the iterations number, and the second one specifies the distance above which a point is considered as an outlier. To our knowledge, the road may be flat or not, and so two thresholds are insufficient to deal with variety of road conditions. We improved the method and the proposed method has three key steps:

- Eliminating the influence of rough road
- Approximating the road by a flat instead of parabola
- Using RANSAC algorithm for multiple thresholds instead of two

3.3.1 Eliminating the influence of rough road: To eliminate the influence of rough roads, a technology which is similar with the difference of Digital Surface Model (DSM) and Digital Terrain Model (DTM) is used. The technology has three key steps:

- Divide the point clouds into small grid
- Calculate the minimum value along z-axis for each grid
- Each point clouds in grid minus the minimum z value

Figure 4 shows results of eliminating rough road.

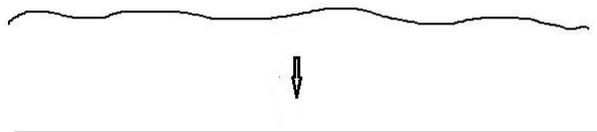


Figure 4. Eliminating the influence of rough road

3.3.2 Rough Segmentation: As shown in figure 5(a), the rough segmentation has four key steps. Firstly, we fit a plane Π_0 from the partitioned data based on RANSAC algorithm. If the distance $D_i(P_i, \Pi_0)$ of a point P_i to the plane Π_0 is less than a threshold α , the point P_i is called inlier, otherwise called outlier. The conditions can be described as

$$D_i(P_i, \Pi_0) < \alpha \quad (1)$$

The set of inliers is called rough plane. Secondly, multiple refined planes Π_i are extracted from rough plane using the same method with different thresholds β_i . Thirdly, rough road edge points could be extracted by combination of rough plane and each refined plane. We search inliers near an outlier. If there exist an inlier, and then the outlier is considered to be a road edge point. Therefore, we can get multiple rough road edges due to multiple refined planes. At last, the best candidate road edge points are extracted by comparing the energy E_s about smoothness for different rough road edge points. The energy can be described as

$$E_s = \sigma(e_i, t_i) \quad (2)$$

where σ = variance
 e_i = candidate road edge point
 t_i = nearest trajectory point to e_i

The segmentation are presented on figures 5(b), 5(c), 5(d) and 5(e). It can be seen that the road edge points are well extracted, although some points could be wrong classified. The proposed method is quite robust and almost no additional assumptions.

3.4 Refined Road Edge Extracting

The detected candidate road edge points are discontinuous and do not contain topology. It is necessary to track and refine these points. The proposed method uses the geometric patterns of road to refine the road edge points, which has two key steps:

- Refining based on stable road width
- Refining based on road continuity

3.4.1 Refining based on road width: We construct an equation to refine candidate road edge points based on road width. The equation is described as

$$\begin{cases} \max N(W_j) \\ \text{subject to } D(L_i, R_i) = W_j \end{cases} \quad (3)$$

where L_i, R_i = left and right candidate point

W_j = road width
 $D(L_i, R_i)$ = the Euclidean distance
 $N(W_j)$ = the number of point

We have three steps to solve the equation. Firstly, the candidate road edge points could be divided into two parts by vehicle trajectory, namely, the left set S_L and right set S_R . Secondly, each point L_i in S_L could find a nearest trajectory point P_i . Thirdly, we find a point R_i in S_R , which must satisfy two conditions: (1) distance of R_i to line $L_i P_i$ is less than a threshold; (2) R_i is the nearest point to P_i . The distance between L_i and R_i is considered to be a temporary road width. Lastly, the largest number of the temporary road width is the actual road width, and the corresponding points are road edge points. The results are presented on figure 6.

3.4.2 Refining by Road Continuity: We construct an equation to refine candidate road edge points based on road continuity. The equation is described as

$$\begin{cases} \max L(P_0, P_n) \\ \text{subject to } K(P_i, P_{i+1}) < \gamma \end{cases} \quad (4)$$

where γ = threshold
 $K(P_i, P_{i+1})$ = slope of continuous two points
 $L(P_0, P_n)$ = the sum of the Euclidean distances of all consecutive points

We have four steps to solve the equation. Firstly, the candidate road edge points are classified into segments by cluster extraction based on the Euclidean distance. Secondly, we fit multiple line segments in each candidate road edge segments. Thirdly, the line segments can be connected based on direction and position. Lastly, the longest line segments after connecting is considered to be the best road edges, and the corresponding points are the best road edge points. The results are presented on figure 7.

4. EXPERIMENTS

4.1 Datasets

We selected four types of road, which covered highway, urban, rural and national primary roads. The processed data was acquired using Leador's MMS. As shown in figure 8, the road consisted of grass-soil edges, kerb edges and fence edges. The scene contains many low fences and cars that are geometrically similar to curbs. Furthermore, the scene contains numerous grass strips, trees, poles, buildings and electric wires, which make the road extraction and delineation more complicated.

4.2 Parameters

We selected these road sections to demonstrate the ability of our algorithm to extract distinct road edges with fixed parameters. To extract rough plane the parameter α is 0.5m, and to extract refined plane the parameter β_i is 0.01m, 0.02m, 0.03m, 0.04m, 0.05m, 0.06m, 0.07m and 0.08m respectively.

4.3 Results and discussion

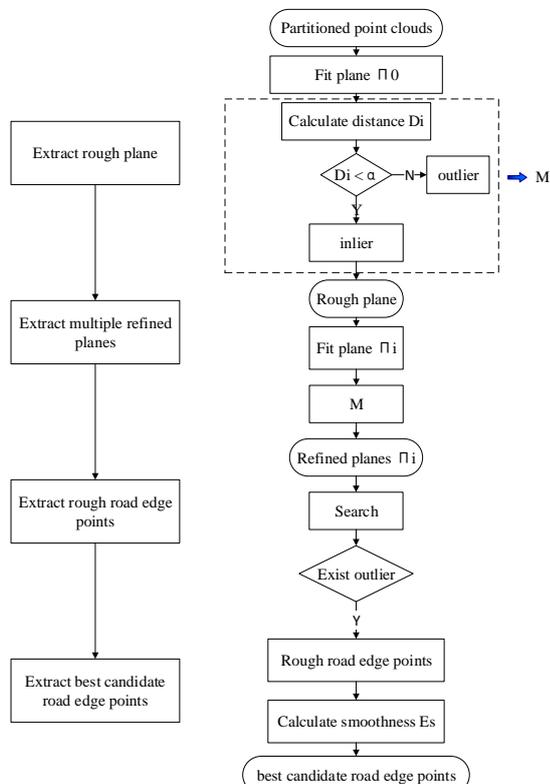
In order to validate the quality of the proposed road edges extraction algorithm, we computed the completeness and correctness quality measures, which are widely used for the evaluation of road extraction algorithms (e.g. Heipke et al., 1997; Clode et al., 2004; Kumar et al., 2013): Buffer zones around the manually digitised left and right edges, which are considered as ground truth in each road section and then the automated extracted left and right edges are labelled as inside or outside the buffer zone (Heipke et al., 1997; Zhou and Vosselman, 2012;

Kumar et al., 2013). The buffer width parameter was set to 0.2m for application. The completeness and correctness values were calculated for the automated extracted road edges in each test data as shown in Table 1.

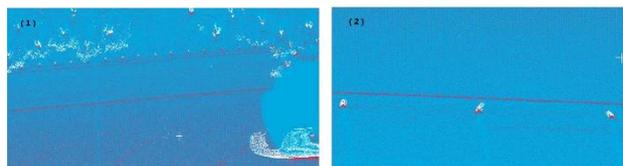
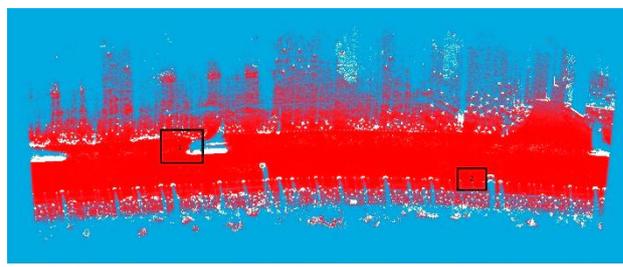
Road	Completeness		Correctness		Time-consuming (ms/m)
	left (%)	Right (%)	Left (%)	Right (%)	
Urban 1	96.1	96.7	99.4	99.6	150
Urban 2	94.5	95.2	99.1	99.2	
Urban 3	56.3	95.3	70.9	99.3	
Rural 1	96.0	97.5	99.2	99.3	
Rural 2	93.2	95.5	99.0	99.1	
Rural 3	83.4	87.5	96.5	96.2	
Highway 1	97.1	96.8	99.8	99.6	
Highway 2	97.3	96.4	99.9	99.9	
Highway 3	97.2	96.8	99.7	99.7	
National primary	86.7	86.1	90.1	90.0	

Table 1. Completeness and correctness values were calculated for the automated extracted road edges

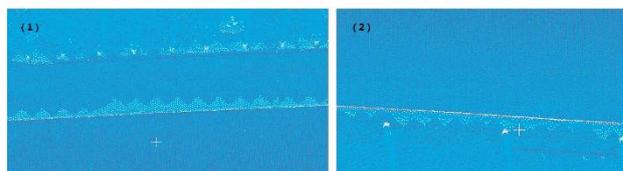
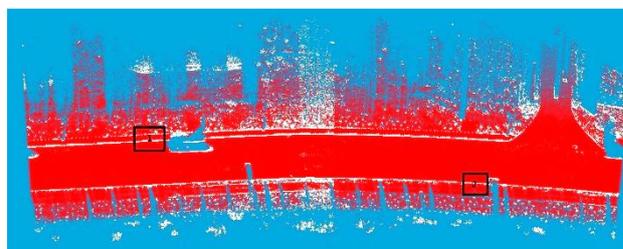
The proposed algorithm was able to successfully extract the left and right road edges in these datasets. In the case of urban 3, the completeness values is less than 70%. The primary reason for this was that too many vehicles parked by the roadside as shown in figure 9. In the case of national primary, the completeness values is less than 90%. This was due to almost no height differences at some road edges as shown in figure 10. We also collect the computation time. The single-threaded average performance is about 150 milliseconds per meters on our platform (3.60Ghz Core i7 CPU, 16.0GB of RAM, C++ implementation), and multi-threaded parallel processing could improve the performance.



(a). Rough segmentation flowchart



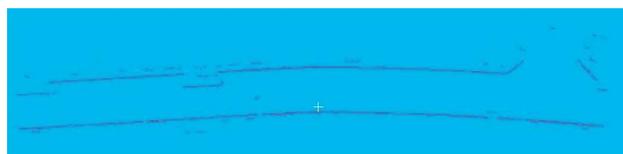
(b). Rough plane and details (red points are inliers, white points are outliers)



(c). One of refined plane and details (red points are inliers, white points are outliers)



(d). One of rough road edge points



(e). Best candidate road edge points

Figure 5. Rough segmentation

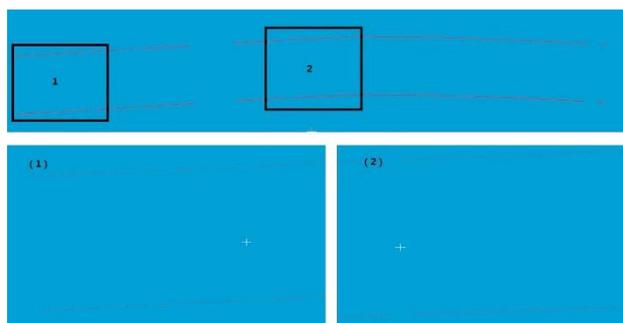


Figure 6. Refining based on road width and details

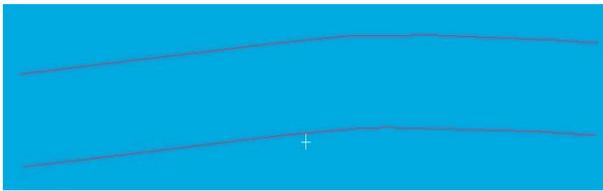


Figure 7. Refining based on road continuity



Figure 8. LiDAR data of various road edges

5. CONCLUSIONS

We have proposed an efficient method to extract road edges from the dense and huge volume point clouds. Our algorithm was based on the assumption that the LiDAR data elevation can be used to distinguish the road surface from the road edges. The successful extraction of road edges from the multiple types of road validates our proposed algorithm.

Although the proposed method can extract 3D road edges from the test datasets, the road conditions are critical for the performance of the proposed method. Particularly, obscure road edges and too many obstructions by roadside have an effect on the detection of road edges, and it is difficult for the proposed method to deal with these situations.

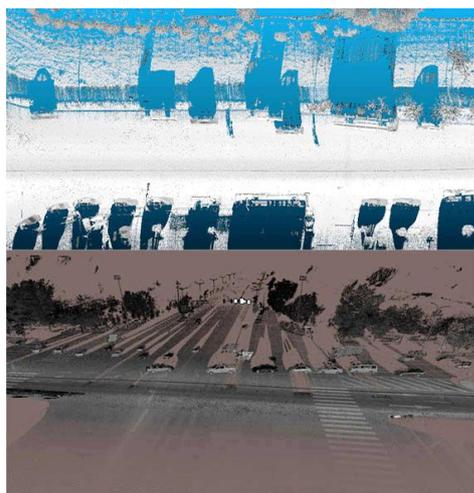


Figure 9. Cars parked at the roadside

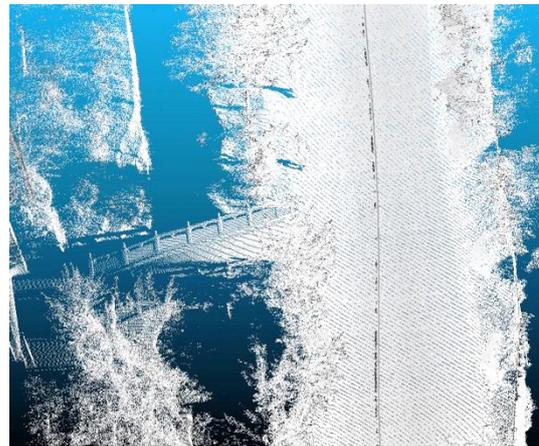


Figure 10. Obscure road edges

REFERENCES

- Clode, S., Kootsookos, P., Rottensteiner, F., 2004. The automatic extraction of roads from LiDAR data. In: *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 35 (Part B3), pp. 1–6.
- Yuan, X., Zhao, C.X., Cai, Y.F., Zhang, H., Chen, D.B., 2008. Road-surface abstraction using lidar sensing. In: *Proc. 10th International Conference on Control, Automation, Robotics and Vision*, 17–20 December, Hanoi, pp. 1097–1102.
- Lam, J., Kusevic, K., Mrstik, P., Harrap, R., Greenspan, M., 2010. Urban scene extraction from mobile ground based LiDAR data. In: *Proc. 5th International Symposium on 3D Data, Processing, Visualization & Transmission*, 17–20 May, Paris, pp. 1–8.
- Smadja, L., Ninot, J., Gavrilovic, T., 2010. Road extraction and environment interpretation from LiDAR sensors. In: *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 38 (Part 3A), pp. 281–286.
- Zhang, W., 2010. LiDAR based road and road-edge detection. In: *Proc. Intelligent Vehicles Symposium*, 21–24 June, San Diego, pp. 845–848.
- McElhinney, C.P., Kumar, P., Cahalane, C., McCarthy, T., 2010. Initial results from european road safety inspection (EURSI) mobile mapping project. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 38 (Part 5), 440–445.
- Ibrahim, S., Lichti, D., 2012. Curb-based street floor extraction from mobile terrestrial LiDAR point cloud. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 39 (Part B5), 193–198.
- Kumar, P., McElhinney, C. P., Lewis, P. and McCarthy, T., 2013. An automated algorithm for extracting road edges from terrestrial mobile LiDAR data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 85, pp. 44–55.
- Pu, S., Rutzinger, M., Vosselman, G., Oude Elberink, S., 2011. Recognizing basic structures from mobile laser scanning data for road inventory studies. *ISPRS Journal of Photogrammetry and Remote Sensing* 66 (6), S28–S39.

Rusu, R.B., Marton, Z.C., Blodow, N., Dolha, M., Beetz, M., 2008. Towards 3D Point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56, pp. 927–941.

Heipke, C., Mayer, H., Wiedeman, C., Jamet, O., 1997. Evaluation of automatic road extraction. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 32 (Part 3/W3), 47–56.

Zhou, L., Vosselman, G., 2012. Mapping curbstones in airborne and mobile laser scanning data. *International Journal of Applied Earth Observation and Geoinformation* 18, 293–304.