

## Article

# Point Cloud Data Retrieval from 3D Geospatial Database for Automated Road Median Extraction

Pankaj Kumar <sup>1,†\*</sup>, Paul Lewis <sup>2</sup> and Conor P. McElhinney <sup>3</sup>,

<sup>1</sup> Geomatics Division, Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA), Castelldefels, Barcelona, Spain; pankaj.kumar@cttc.es

<sup>2</sup> National Centre for Geocomputation (NCG), Maynooth University, Co. Kildare, Ireland; paul.lewis@mu.ie  
<sup>3</sup> Amazon Web Services (AWS), Dublin, Ireland; conormc@gmail.com

\* Correspondence: pankaj.kumar@cttc.es; Tel.: +34-936-452-900

† Current address: Geomatics Division, Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA), Castelldefels, Barcelona, Spain

**Abstract:** Laser scanning systems make use of Light Detection and Ranging (LiDAR) technology to acquire accurately georeferenced sets of dense 3D point cloud data. The information acquired using these systems produces better knowledge about the terrain objects which are inherently 3D in nature. The LiDAR data acquired from mobile, airborne or terrestrial platforms provides several benefit over conventional sources of data acquisition in terms of accuracy, resolution and attributes. However, the large volume and scale of LiDAR data have inhibited the development of automated feature extraction algorithms due to the extensive computational cost involved in it. Moreover, the heterogeneously distributed point cloud, which represents objects with varying size, point density, holes and complicated structures pose a great challenge for data processing. Currently, geospatial database systems do not provide a robust solution for efficient storage and accessibility of raw data in a way that data processing could be applied based on optimal spatial extent. In this paper, we present Global LiDAR and Imagery Mobile Processing Spatial Environment (GLIMPSE) system that provides a framework for storage, management and integration of 3D LiDAR data acquired from multiple platforms. The system facilitates an efficient accessibility to the raw dataset, which is hierarchically represented in a geographically meaningful way. We utilise the GLIMPSE system to automatically extract road median from Airborne Laser Scanning (ALS) point cloud. In the first part of this paper, we detail an approach to efficiently retrieve the point cloud data from the GLIMPSE system for a particular geographic area based on user requirements. In the second part, we present an algorithm to automatically extract road median from the retrieved LiDAR data. The developed road median extraction algorithm utilises the LiDAR elevation and intensity attributes to distinguish the median from the road surface. We successfully tested our algorithms on two road sections consisting of distinct road median types based on concrete and grass-hedge barriers. The use of GLIMPSE improved the efficiency of the road median extraction in terms of fast accessibility to ALS point cloud data for the required road sections. The developed system and its associated algorithms provide a comprehensive solution to the user's requirement for an efficient storage, integration, retrieval and processing of large volumes of LiDAR point cloud data. These findings and knowledge contribute to a more rapid, cost-effective and comprehensive approach to surveying road networks.

**Keywords:** Airborne Laser Scanning; Geospatial Database; Data Retrieval; Road Median; Attributes

## 1. Introduction

Accurate 3D information about the location of real-world objects along with their geometrical and structural properties supports a wide range of services and evidence based decision making. Light Detection And Ranging (LiDAR) enables 3D surveying of real world environments by measuring the time of return of emitted light pulses. The applicability of laser scanning systems, which use

LiDAR technology, continues to prove their worth due to the rapid, continuous and cost effective 3D data collection capability. The information obtained through these systems contributes to the production of useful knowledge which can be used to develop more efficient approaches for managing urban infrastructures and natural resources [1]. Their 3D data acquisition capability produces better knowledge about terrain objects which are inherently 3D in nature. Laser scanning systems facilitate the acquisition of an accurate, precise and georeferenced set of dense LiDAR point cloud data. Other benefit of these systems is the high level of automation involved in data acquisition process and their ability to acquire data beneath tree's canopy [2].

The 3D data acquired from Mobile Laser Scanning (MLS) and Airborne Laser Scanning (ALS) systems differ in terms of intrinsic accuracy, resolution and object scale. This difference is incurred primarily due to the distance of the scanner to the target objects, scan direction and view angle [3]. Laser scanning system can facilitate reliable and accurate acquisition of 3D spatially referenced data about road median, which is a narrow strip of land that separates traffic on opposite sides of the road. The road median is one of the fundamental feature, whose correct identification is a prerequisite to obtain precise information about road and other physical objects along it. The acquired 3D LiDAR data can be used to locate, measure and classify the road median in a timely cost-effective manner in order to facilitate their maintenance. This information can assist road authorities in effective management of road networks and to ensure maximum safety conditions for road users [4].

LiDAR data records a number of attributes including elevation, intensity, pulse width, range and multiple echo information, all of which can be used for precise extraction of road median. It is a rich source of 3D georeferenced information, whose volume and scale have inhibited the development of automated algorithms [5]. The large volume of data produced by modern-day laser scanning systems lead to time-consuming and computational expensive processes for automated features extraction. For example, Riegl VQ-250 MLS generates 300,000 points per second resulting in approximately 20GB of data per hour. Significant challenges exist when dealing with these volumes of LiDAR data, from its storage through all stages to features extraction. The raw LiDAR data files usually contain long strips of points which prohibit efficient data processing in terms of memory allocation and algorithm design [6]. The acquired point cloud is heterogeneously distributed representing objects with varying size, point density, holes and complicated structures, which poses a great challenge for data processing [7].

While an efficient storage system for very-large 3D point cloud data is one of the key aspects for information extraction. Equally important should be the ability of such a system to segment and merge point clouds that are spatially coincident. However, the industrial standard has been to cluster LiDAR data into spatial grids, which are then stored as binary files in a survey-based file structure. This survey-based approach, that prevails in most software suites, has proven to be a major constraint in a meaningful analysis of the LiDAR data [8,9]. These software suites do not provide efficient access to the raw data at different or irregular spatial dimensions. It becomes challenging to access LiDAR data for a particular geographical area across numerous surveys that invariably accumulate as a massive dataset. The efficiency of automated processing algorithm depends upon the raw LiDAR data being spatially optimised for the algorithm. The data storage system should facilitate an efficient accessibility to the raw datasets in a way that data processing could be applied based on optimal spatial extent and point density/resolution. This will improve the efficiency of automated algorithms in terms of computation and reduced processing of the dataset.

In this paper, we present a **Global LiDAR and Imagery Mobile Processing Spatial Environment (GLIMPSE)** system, that provides a framework for storage, management and integration of LiDAR data acquired from terrestrial, mobile and aerial platforms, in the context of a automated pipeline towards the development of a road median extraction algorithm. We describe an optimal construction of a very large scale spatial database and how these datasets can be hierarchically represented in a geographically meaningful way, that facilitates the spatial segmentation of data based on user requirements. Finally, the segmented data is visualized in a WebGL enabled web-application viewer that allows a user to interact with it. The prototyped system provides a comprehensive framework

for integration, scalability, fast retrieval and processing of point cloud data to extract features. We developed a method for efficiently retrieving the ALS point cloud from the GLIMPSE system based on user specified spatial extent and then retrieved data is further processed to automatically extract the road median. When imported in the GLIMPSE prototype, the ALS data is optimised into a spatial hierarchy. In the first part, we present an approach to retrieve point cloud data for a particular geographical area from a large volume of raw data stored in the GLIMPSE system. In the second part, we process the retrieved LiDAR data to automatically extract the road median. The developed road median algorithm is based on the assumption that the LiDAR elevation and intensity attributes can be utilised to distinguish median from the road surface.

In Section 2, we review different geospatial database approaches used for managing very-large voluminous LiDAR datasets. We present a detailed review of various methodologies developed for extracting road objects from ALS and MLS datasets. Following the review, we list various limitations existing in current LiDAR data management and processing approaches, which have been addressed through our research work. In Section 3, we describe the GLIMPSE system and how spatial hierarchies are built, then we provide details on the spatial data segmentation and fusion processes involved in it. In Section 4, we provide a step-wise description of our automated algorithms to retrieve point cloud data from the GLIMPSE system and then to extract the road median from the retrieved data. We test our algorithms on two road sections consisting of distinct road median types in Section 5. In Section 6, we validate the road median extraction results and discuss them. Finally, we conclude the paper in Section 7.

## 2. Related Work

The typical workflow while handling large volumes of LiDAR data has been to process the raw data into specific format, such as a Digital Elevation Model (DEM) [9]. DEMs have been quite popular for representing surface from 3D LiDAR data in either raster or Triangulated Irregular Network (TIN) format. These 2.5D representations are useful in simplifying the complexity of LiDAR point cloud, but do not fully exploit the 3D capability. In some cases, the largest features (i.e. ground or buildings) present in the point cloud are manually removed in order to reduce their complexity and processing time [10]. These constraints have led to the use of Spatial Database Management Systems (SDBMSs) for the storage, management and retrieval of LiDAR data. The standard has been to grid the raw LiDAR data into spatially equal tiles in SDBMS and then spatial boundaries of these files are used to access raw files through spatial query operations. LiDAR point cloud datasets are stored in most SDBMSs as either point/multipoint or other geometric data types [11]. The management of data with point type is difficult to handle, however the multipoint type makes the retrieval process simpler by storing collection of points in a single record but it does not include all the features. The representation of LiDAR data with 2D geometric types (i.e. polygon, box) again do not fulfil the requirements of 3D space.

An efficient 3D SDBMS must provide a capability to apply various spatial operations and functions, embedded in its query language, to 3D data type [12]. A suitable spatial indexing is crucial in a database as it is used to accelerate data retrieval and operations performed on it. It organizes the database space in a particular manner so that no complete table is required to be examined in order to complete particular queries [13]. Most SDBMSs perform spatial indexing based on R-tree [14] and quad-tree [15] techniques. The R-tree structure uses Minimum Bounding Rectangle (MBR) to divide an object space in a tree-shaped hierarchical structure, while its 3D extension consists of Minimum Bounding Boxes (MBB). The quad-tree data structure decomposes space in a hierarchical structure such that each node representing bounding box consists of maximum four child nodes, however in its 3D extension, octree, each node can have up to eight child nodes. Many traditional DBMSs have been extended with functionalities to manage and store 3D spatial data [13]. Oracle Spatial has been developed as a modification of Oracle 4 while PostGIS is an extension of another popular open-source DBMS named PostgreSQL. They both support different 3D spatial data types (such as point, line,

133 polygon and combination of them) and provide 3D spatial indexing mechanism that enables fast data  
134 retrieval.

135 Several other approaches have been developed for managing the large volume of LiDAR point  
136 cloud data. [16] presented a scalable approach to interpolate grid DEM from large LiDAR dataset  
137 based on quad-tree segmentation. In their approach, the point cloud data was partitioned into a set of  
138 quad-tree segments and then each segment was interpolated using points within the segment and its  
139 neighbourhood. The approach was tested on 390 million points (around 20GB) to interpolate DEM  
140 in about 53 hours, which other GIS software suites were unable to process. [17] applied octree and  
141 local KD tree based approach to manage and visualize large LiDAR dataset, while [18] optimized a  
142 workflow for processing airborne LiDAR data within GIS-based environment. [19] presented a study  
143 with the use of spatial extensions in IBM's DB2 database to manage high-resolution airborne LiDAR  
144 data. They also experimented with a single partitioned database on a supercomputer resource and  
145 multi-partitioned database across several nodes that function together as a single database engine,  
146 in order to deal with large volume dataset. In [20], a method was proposed for distributed data  
147 organisation and parallel data retrieval from huge volume of airborne LiDAR data. The distribution  
148 strategy took into account the spatial relationship in between the dataset, while an improved data  
149 retrieval speed led to fast analysis, visualization and processing of the point cloud. [21] implemented  
150 an octree data structure to store and compress 3D point cloud data. They further demonstrated its  
151 usage for an easier exchange of file format, fast data visualization and an efficient plane detection  
152 algorithm. [22] presented a point cloud management system based on groups of points that provided  
153 a perspective for meta-data, concurrency, integration with other geospatial datasets, filtering and  
154 fast processing. The proposed system was tested with several billion points acquired from aerial,  
155 terrestrial LiDAR and stereo-vision. With an expansive growth in cloud computing services, there has  
156 been an increased deployment of various distributed web-based LiDAR management applications.  
157 NSF-funded OpenTopography is one such significant application that provides a web-based access  
158 to high resolution LiDAR topography data along with derivative products and online processing  
159 tools [23]. Another such application is Dielmo's LiDAR-Online that provides a web-based platform to  
160 visualize, access and process LiDAR data [24].

161 LiDAR has matured to an accurate technology which can be employed for reliable extraction  
162 of various features along road networks. The extracted roads can be represented as homogeneous  
163 areas or pairs of parallel lines corresponding to edges, depending upon the spatial resolution of the  
164 input dataset. The methods developed for segmenting roads from ALS datasets are mostly based  
165 on utilizing their attributes to distinguish road areas from other objects. [25] reported their work  
166 on the segmentation of ALS data into road and non-road objects based on elevation and intensity  
167 attributes, while [26] detected kerbstones based on the detection of small height jumps caused by them  
168 in the ALS data. Their road extraction results were influenced by the presence of parking, private  
169 roads and parked cars in the surveyed areas. The integration of high resolution optical imagery or  
170 2D topographic map data with aerial LiDAR data for road extraction has also been reported [27–29].  
171 However, the road extraction accuracy might be affected from positioning errors inherited in maps  
172 and occlusion arising from building and tree shadows in optical imageries. In most recent works, [30]  
173 presented road detection approach in which ALS data was filtered to estimate ground points and then  
174 road candidates were identified based on local distribution of intensity histogram. [31] proposed a  
175 method to extract road centrelines using ALS data. Their method was based on filtering the ground  
176 points and then estimating road points by applying an optimal intensity threshold. The estimated  
177 points were finally refined by removing narrow roads and attached areas to extract the network of road  
178 centrelines. [32] detected the roads in forested mountainous areas using ALS data. In their approach, a  
179 supervised classification was applied to Digital Terrain Model (DTM) and then a graph was built over  
180 candidate regions to locate the roads. Finally, the roads were characterised to estimate their width and  
181 slope parameters using an object-based image analysis. Several methods have also been reported for  
182 extracting road edges, in particular, from MLS data. These works are particularly focused on extracting



kerb edges in an urban environment, where there is a sufficient height or slope difference in between the road and kerb points [33–37]. In rural conditions, the road comprises of grass-soil surface, in which case the edges are not as easily defined by slope or elevation changes alone. The approaches developed for extracting rural road edges from MLS data are based on integrated use of its elevation, intensity and pulse width attributes which were utilized to distinguish the road from grass-soil surface [4,38,39]. Apart from these, several other methods have been proposed for extracting road markings [40,41], road poles and towers [42,43], road surface roughness [44–46] and surrounding tree objects [7,47] from MLS data.

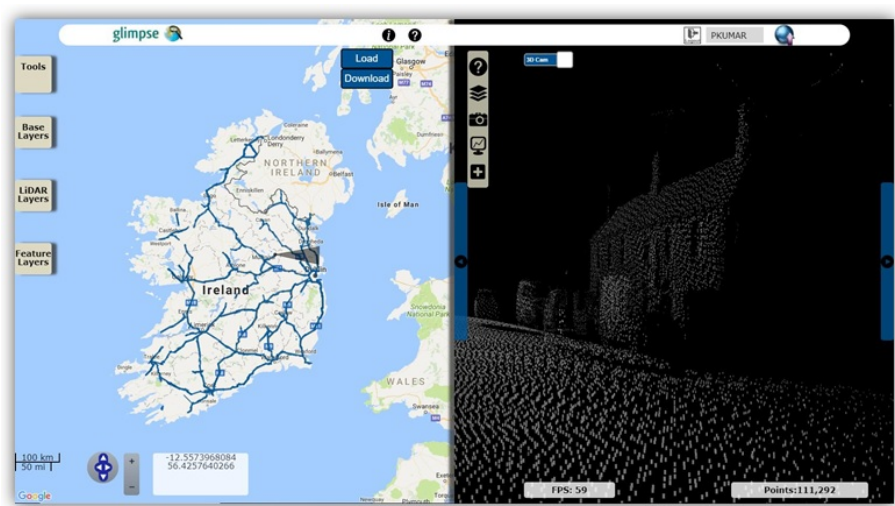
One of the major constraints in the approaches developed for extracting road objects from LiDAR data is the computational intensive, iterative and time consuming processes involved in them. This is due to the massive size and un-organised nature of LiDAR datasets that limits their meaningful analysis for extracting relevant information. Such huge volume datasets give rise to significant challenges for data visualization, efficient data analysis and rapid data processing. The users are prohibited from exploiting the full range of opportunities that LiDAR data offers. There has been very limited use of any LiDAR data management platform in the road features extraction processes that could have provided spatially optimised accessibility and fast data processing to the users. This, in turn, would have been beneficial in terms of improved efficiency and computational capabilities of automated algorithms. Some SDBMSs offer capabilities for storage, management and retrieval of LiDAR data but fail to support an efficient analysis of such vast dataset. The existing data types and spatial indexing techniques appear to be insufficient to handle large volumes of LiDAR data. There has been very few systems where the LiDAR datasets acquired from terrestrial, mobile and aerial platforms, could be integrated into a single data management solution. There is a need for a more robust and comprehensive data management framework that could provide spatially optimised, unrestricted and integrated access to the LiDAR points and its attributes. This would facilitate an efficient data analysis and fast data processing in order to extract road features from LiDAR data in an automated and operational way. Towards this goal, we describe the GLIMPSE system in the next section.

### 3. 3D Geospatial Database

Empirical experience with both ALS and MLS geospatial data has shown that the primary obstacles in the processing of these datasets is their considerable size and the inability to easily constrain them based on point attributes. Leading on from this is the preparation and extraction difficulties, when using these data for bespoke requirements. For example, in the case of extracting a road median, the process would be significantly constrained by the survey-processing methodology that prevails in industry standard software suites. These suites provide no context for spatial optimisation of the data loaded from many different surveys. Thus, data segmentation for road median detection cannot be easily implemented through an optimal and empirically informed spatial approach.

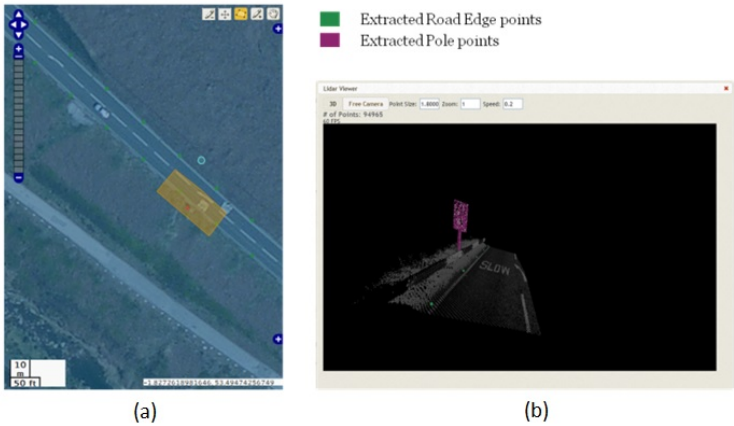
However, approaching this problem with a point-cloud fusion and spatial-constraint perspective, it is possible to optimise the LiDAR data being output to algorithms that specialise in feature extraction process. This can be achieved through procedures that leverage the power of a platform such as PostGIS and its numerous, integrated, spatial API's. The geo-referenced raw LiDAR is stored in a database where optimised spatial indexes can be generated in order to facilitate efficient querying of the data. Consequently, optimally located LiDAR data, across numerous surveys, can be output in a user required spatial context. In that case, the road median algorithm, can be operated on a reduced target data set relative to the original survey but, also, at higher point densities as spatially coincident point clouds can be segmented and fused in the same database operations.

Towards this objective, a prototype cloud-application has been developed called GLIMPSE that currently hosts 4.2 terabytes of ALS and MLS data comprising 29 billion records. These datasets cover multiple MLS and ALS surveys for 90% of the national road network in Ireland. The LiDAR data coverage is displayed as blue transparent polygons in the Google maps UI along with WebGL viewer in the GLIMPSE platform, as shown in Figure 1. GLIMPSE builds a 2D spatial hierarchy of the 3D



**Figure 1.** The GLIMPSE platform with LiDAR coverage represented in blue in the Google maps UI along with WebGL viewer.

LiDAR data uploaded into the system. The importance of this step is that it gives a spatial context, where optimally located LiDAR data can be segmented and fused for output to a feature extraction processing algorithm, as is the case with the road median extraction. The system enables a user to segment the raw data using a spatial tool and then the processed results are visualised through the GLIMPSE WebGL viewer, as shown in Figure 2.



**Figure 2.** User defined (a) LiDAR section polygon represented in orange and (b) processed results in the WebGL viewer.

In Section 3.1, we detail how spatial hierarchies are built and defined in the GLIMPSE system, while in Section 3.2, we describe the optimal approach to data segmentation, fusion and retrieval. Through these sections, the cloud-application based User Interface (UI) in the GLIMPSE system is also detailed that enables interaction, understanding and visualisation of the platforms objectives.

### 3.1. GLIMPSE Spatial Hierarchy

We implemented a 2D spatial hierarchy, where all the spatial data sources from MLS, ALS and terrestrial surveys are stored at different levels of spatial detail. This stage does not change the original spatial detail of the LiDAR but allows a user to evaluate and thus process the available LiDAR in approximate coverage zones. Two examples are presented in Figure 3, in which the LiDAR coverage is

shown in the Google maps UI as blue transparent polygon. In Figure 3(a), the coverage of ALS survey

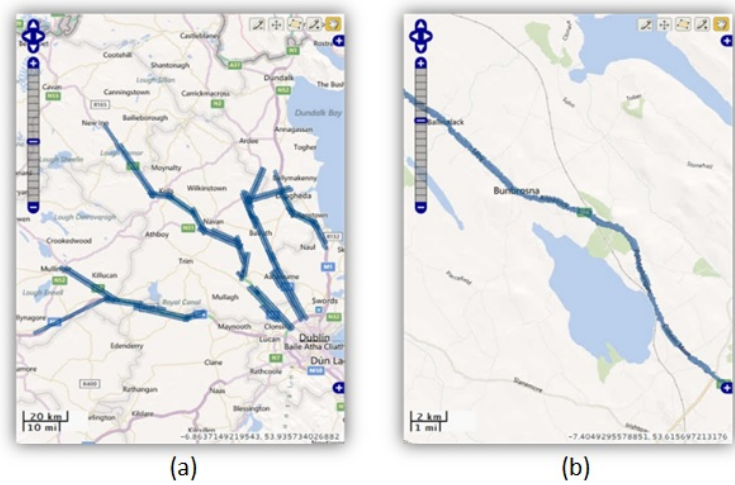


Figure 3. Approximated coverage zone for (a) ALS and (b) MLS datasets.

containing 1,700 million points is shown, while Figure 3(b) shows the LiDAR coverage of MLS survey consisting of 400 million points. Within the GLIMPSE UI, the user can use standard GIS point and polygon creation tools to intersect a planar view of the available data in this area, which is presented in Section 3.2. This is also true for automated process that can be scripted into the GLIMPSE platform.

The spatial hierarchy procedure has been implemented as a bespoke implementation of the PostGIS2 ST\_ConcaveHull method. This approach was chosen as it presented significant performance improvements over the default PostGIS implementation. This was important, not only because the quantities of LiDAR being processed are so large, but also because the three different LiDAR data source types have very different point density properties. For example, MLS data can have a point density in excess of 4,000 points per  $m^2$ , where point densities reduce as a function of distance orthogonal to the survey vehicles direction of operation. Effectively this LiDAR data source produces a very dense and detailed route corridor point cloud. On the other hand ALS data, which can also be presented in a corridor like fashion representing the aerial vehicles flight path, the point densities are very much uniform across the scan area and typically in the order of 10 points per  $m^2$ .

The first stage in our process is to snap all the LiDAR data to a spatial grid. A sub-step in the first stage involves the application of a sub-sampling threshold to the LiDAR. This sub-sampling is applied differently depending on the LiDAR data source; in the case of MLS data, areas with high point densities, close to the survey vehicle, are sub-sampled with a higher threshold than areas with lower point densities. The second stage is to generate concave hulls for all the sub-sampled and gridded data. Finally, a spatial union is performed on all the concave hulls for the LiDAR data being processed.

This final stage can take the form of a number of different spatial unions such that the highest accuracy concave hull is a direct 2D spatial representation of a single table of raw LiDAR. This idea follows through to unions that define all the data in different spatial configurations; local, regional, national, etc. The spatial union can also be applied in a number of other ways such as modelling the data using the survey based approach that exists in typical commercial software suites.

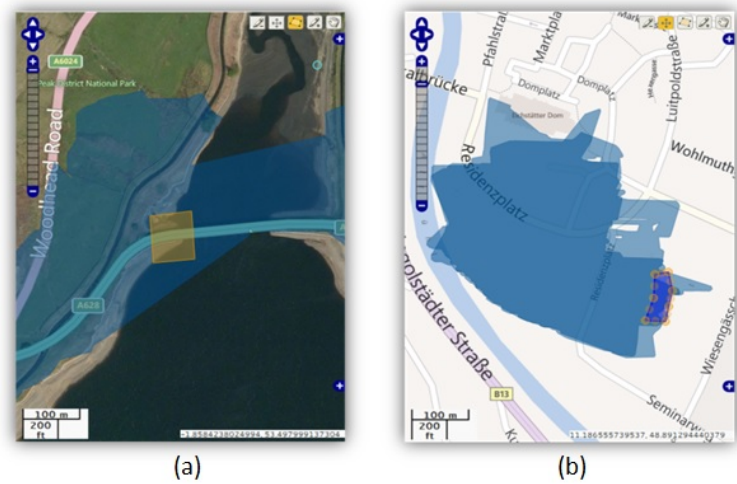
### 3.2. GLIMPSE Spatial Segmentation & Fusion

Having approached each stage of this framework pipeline with a spatial constraint perspective to the fore, it is possible at this stage to optimize the LiDAR data being output. This can be achieved, once again, through procedures that leverage the power of the PostGIS platform through its numerous, integrated, spatial functionalities. Due to the spatial-indexing and spatial-hierarchies that have been

generated for the LiDAR, it is relatively easy to highlight use cases which generate bespoke LiDAR subsets from our 29 billion points stored in a 4.2 TB database.

These use-cases could be in any relevant form where the process being initiated needs access to LiDAR. Consequently, this could be an automated processing algorithm, such as the road median extraction algorithm, where subsets of LiDAR can be spatially optimized and fused such that the algorithm handles only relevant LiDAR in the system. In Figure 3, we can see a sample of the GLIMPSE UI which highlights how this can happen. Alternatively, this example use-case could just as easily be a LiDAR awareness approach where LiDAR can be quickly and easily segmented for a user to view the outputs such that they are suitable for a given process or requirement.

In this example both point or polygon spatial-constraint geometries can be created by a user; polygons have been created in this case as can be seen in Figure 4. This operation provides



**Figure 4.** User initiated sectioning of (a) and (b) LiDAR datasets with polygon tools for feature extraction processing.

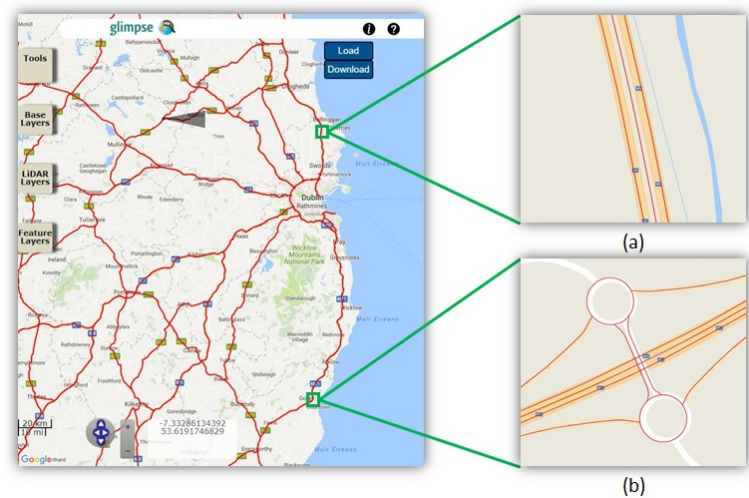
the geographical context for the LiDAR query where the user can choose between a 2D or 3D implementation. Either choice will result in a 3D LiDAR point cloud being returned. However, in the 3D case the extra altitude (Z) parameter is added to the spatial query and can be used to bind the point-cloud in the Z domain, i.e. a road median extraction algorithm will not require data outside the boundary of the road surface. These optimizations can significantly reduce the number of points returned which subsequently decreases the query time, required processing time and the rendering time.

Extending on from this example is the possibility of numerous different user or process driven use-cases that may require the spatially segmented LiDAR data to be returned based on attribute constraints as well. For every independent LiDAR attribute, including the table's primary key, an index exists such that searching any LiDAR table becomes a function of the index being searched and not a dependent process on the table's primary key. Thus, in the previous example we could further constrain the returned point cloud data by requesting only points whose intensity values fall within a certain range, which allows us to continue to leverage the power of the database. Using the optimizations that are inherent to the PostgreSQL database, a standard spatial query will optimize its search based on the spatial index and the primary key of the table. For queries that extend such a search, based on other attributes, the primary key element is easily replaced with the index of the chosen constraint. In the main, runtimes on these attribute constrained queries will be comparable in the single constraint case, although as the number of attribute constraints increases runtimes would logically extend. In the next section, we describe our point cloud retrieval and road median extraction algorithms.



4. Algorithm

We developed algorithms to efficiently retrieve ALS data from the GLIMPSE system based on user specified spatial extent and then to automatically extract road median from the retrieved data. The use of GLIMPSE system facilitates the segmentation and fast retrieval of point cloud data for a particular geographical area. In the first part, we describe a method to efficiently retrieve ALS data from the GLIMPSE system based on user specifications while in the second part, an algorithm to automatically extract road median from the retrieved ALS data is presented. We use road vector polylines in our algorithm as a secondary data source, which were obtained from the National Roads Authority (NRA) in Ireland. These road polylines were imported in the GLIMPSE system, while representing centre, left-lane and right-lane of the dual carriageways in Ireland as shown in Figure 5. Each road polyline



**Figure 5.** Road vector polylines represented in red in the GLIMPSE system, with inset pictures representing (a) N01 and (b) N11 road sections in detail.

was associated with Route ID, Road Number, Direction, Section Number and Lane Representation.

4.1. Point Cloud Retrieval

Our point cloud retrieval algorithm is developed based on the assumption that a polygon spatial tool in the GLIMPSE system can be utilised to segment the ALS data, as shown in Figure 6. The



**Figure 6.** GLIMPSE polygon spatial tool.

spatial tool requires height and width parameters along with heading and position (i.e. latitude and longitude) information at the centre of polygon for enabling it to segment the LiDAR data. The user specifies the cross-section length (i.e. height),  $l_c$  and width,  $w_c$  dimensions of polygon along with the required total length,  $l_t$ . The central position and heading parameters of each polygon segment are estimated using road vector polylines based on user clicked location. A workflow of the point cloud retrieval algorithm is shown in Figure 7.

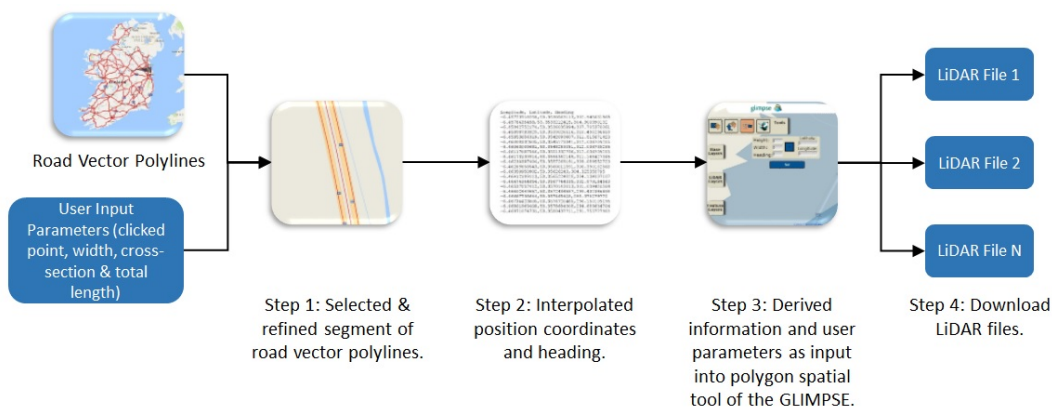


Figure 7. Point cloud retrieval algorithm.

The user explores the road polylines incorporated in the Google maps UI of GLIMPSE system and can click any point near the preferred road section for which the ALS data is required. In Step 1 of our algorithm, the road polyline point nearest to the user clicked location is selected using a kd-tree approach. The k-d tree is a data structure which is used to organise the points in a space with k-dimensions. Each node of a tree specifies an axis and splits the set of points based on whether their coordinate along that axis is greater or less than a node value [48]. The splitting is done based on the first dimension axis and then each level down the division is done based on the next dimension. The splitting points are selected to ensure that the cells do not become long and thin. In our approach, all the road polyline points, representing the centre of the dual carriageway, are split in three dimension and then the polyline point nearest to the user clicked location is selected. The selected polyline point is used to further select the consecutive polyline points based on total length specified by the user.

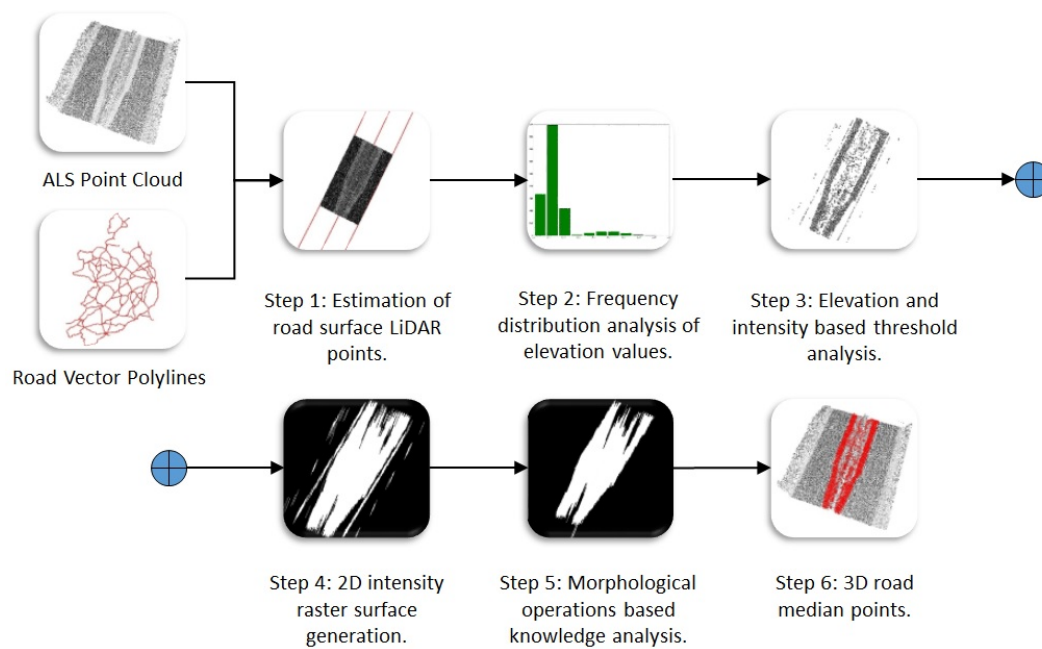
In step 2 of our algorithm, the central position coordinates of each cross-section length are linearly interpolated from the selected road polyline points. The heading at each centre location is derived by first estimating the slope angle from the interpolated position coordinates and then estimating the angle with respect to the North direction. In step 3 of the point cloud retrieval algorithm, the position coordinates and heading information along with user specified length and width dimensions are input into the polygon spatial tool of the GLIMPSE system. The input information facilitates the segmentation of ALS data using the polygon tool and allows the user to download the segmented data from the GLIMPSE system in step 4 of our algorithm. For example, if the user wants to access the ALS data of 2km road section with 100m cross-section length, then 20 data files will be downloaded from the GLIMPSE system.

4.2. Road Median Extraction

The ALS data files accessed using the point cloud retrieval algorithm are further used to extract the road median. Our road median extraction algorithm is based on the fact that LiDAR elevation and intensity attributes can be efficiently utilised to extract the median [49]. We input *n* number of ALS data files and road vector polylines in our road median extraction algorithm. The input data files are batch processed to extract continuous road median along the tested road section. The dimensions (i.e. length, width, height) of data section in each file are based on user input parameters in the point cloud retrieval algorithm. A workflow of the road median extraction algorithm is shown in Figure 8. In the following sections, we describe various processing steps involved in our algorithm.

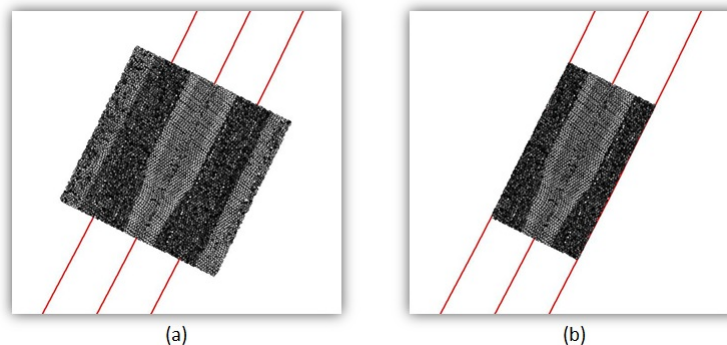
4.2.1. Road Surface Estimation

In Step 1 of our algorithm, we use the road vector polylines to estimate the LiDAR points belonging to the road surface. We consider the polylines along the left-lane and the right-lane, and



**Figure 8.** Road median extraction algorithm.

find their intersection with the convex hull fitted to the input LiDAR points. The intersected boundary points are then used to remove the outside points, while inner points are retained which belong to the road surface. Thus, the use of road polylines is efficient in terms of reducing the search space in LiDAR data for extracting the road median. The process of estimating road surface points using road polylines is shown in Figure 9.

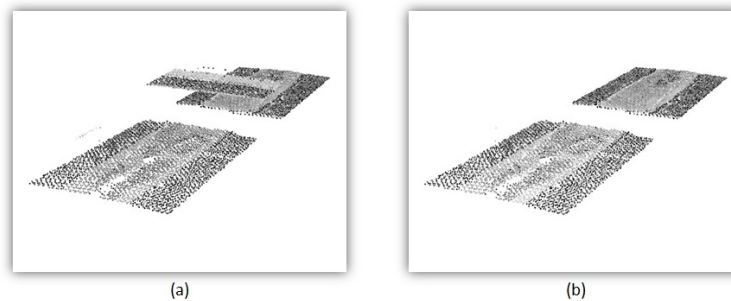


**Figure 9.** The input (a) road section points and (b) road surface points estimated using polylines represented in red.

#### 4.2.2. Frequency Distribution Analysis

The road sections consist of highways crossing above them at some locations, which are required to be removed in order to get a correct estimation of the road median. In Step 2 of our algorithm, these crossing highways are removed based on frequency distribution of the elevation values obtained from the road surface LiDAR points. We assume that along the road section, a large number of LiDAR points will belong to its surface, while in comparison less points will correspond to any highway crossing above it. Based on this assumption, frequency distribution of elevation values is estimated and then elevation value with maximum frequency is identified as  $h_r$ , which belongs to the road surface. The points with elevation values more than  $h_r + \epsilon$  or less than  $h_r - \epsilon$  are removed, while in between points

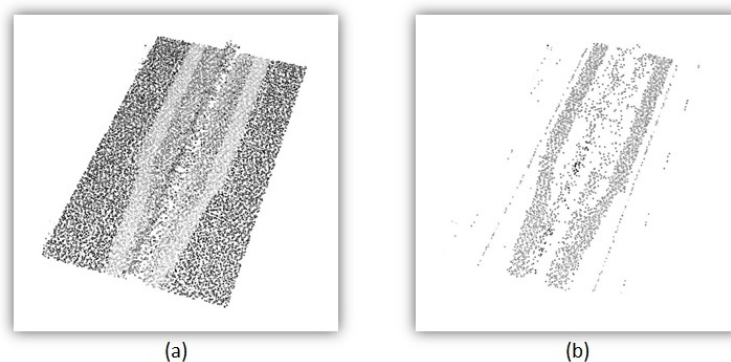
are retained for further processing. The value of  $\epsilon$  parameter is estimated empirically and fixed for all the road sections in such a way that it could be useful in removing the crossing highways. In this way, crossing highways above the road sections are removed by detecting the road surface points with maximum elevation frequency. An example of removing crossing highway above the road section is shown in Figure 10.



**Figure 10.** The road surface (a) with highway crossing above it, which is then (b) removed using frequency distribution analysis.

#### 4.2.3. Threshold Analysis

In Step 3 of our algorithm, we apply threshold to elevation and intensity attributes in order to get an initial estimation of the road median. The elevation and intensity values are normalised with respect to their minimum and maximum values, and converted to an 8-bit data type. This enables a two way transformation between the 8-bit values and their original LiDAR values, which in turn will allow for the use of a single threshold value for all road sections. We apply  $T_{elev}$  and  $T_{int}$  threshold parameters to road surface LiDAR points and get initial road median points, as shown in Figure 4. The values of these threshold parameters are estimated empirically and fixed for all the road sections, which allows for their fully automated application.



**Figure 11.** Threshold values applied to the (a) road surface points in order to get an initial estimation of the (b) road median.

#### 4.2.4. 2D Raster Surface Generation

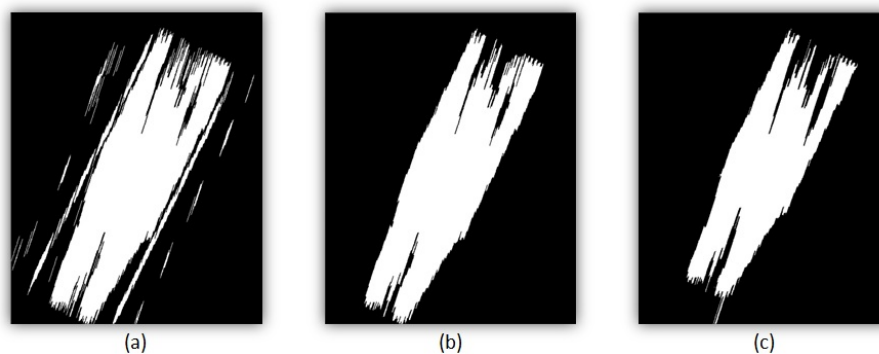
In Step 4 of our algorithm, we generate 2D intensity raster surfaces from the initial estimated road median points using a cell size,  $c$  parameter. The value of  $c$  is selected based on an average spacing of LiDAR points. The value of each cell in the raster surface is estimated as the average of the intensity values of the LiDAR points that fall within the 2D boundary of the cell.



#### 4.2.5. Morphological Operations

The initial estimated road median may be incomplete and contain other road surface elements that are introduced through the use of thresholding. To overcome this, we use morphological operations based knowledge analysis of road median in Step 5 of our algorithm [40]. This analysis involves three processes. In the first process, the intensity raster surface is converted into a binary image and then the morphological dilation operation is applied in which a structuring element is placed over the image cells. The purpose of dilation is to use the structuring element to grow cells with a value of 1 in order to fill in any holes. A structuring element consists of a binary matrix that represents the selected shape and size. A central element of the matrix represents an origin and the elements with a value of 1 describe a neighbourhood of the structuring element. The origin of the structuring element is positioned over each cell in the binary raster surface to dilate that cell along the neighbourhood of the structuring element.

We select a linear shaped structuring element to dilate the cells due to linear pattern of the road median. The length,  $l$ , of the linear element is selected empirically and is fixed for all the road sections, which allows us to automate the morphological operation in our algorithm. The linear element is used with a  $\theta$  angle that is calculated from the mean heading of the road polyline points. This angle is useful in dilating the road median cells along the longitudinal direction. The use of dilation operation fills the holes and completes the shapes of road median in the input binary image. An example of the dilation of input binary image using linear element is shown in Figure 12(a).



**Figure 12.** Morphological operations based knowledge analysis: (a) dilated image, (b) road surface cells removed and (c) eroded image.

In the second process, we group cells into objects in the dilated image using connectivity. If a cell has a value of 1 then it is connected to the cells whose values are 1 and are directly above, below, left or right of that cell. We calculate the length and average width values of each object in the dilated image. Objects whose length and average width values are less than length threshold,  $T_L$  and width threshold,  $T_W$  are considered as other road surface elements and are removed from the image, as shown in Figure 12(b). These length and width threshold values are estimated based on knowledge about standard dimensions of the road median. In the third process, we apply an erosion operation to the dilated image in order to retain the original boundary shape of the road median. In an erosion operation, cells are removed from the road median cells using a structuring element. The linear shaped structuring element used for dilation is also applied to erode the road median cells, as shown in Figure 12(c). In this way, the combined use of morphological operations and knowledge about the dimensions of the road median is able to complete its shape and remove other road surface elements.

#### 4.2.6. 3D Road Median Points

In final Step 6 of our algorithm, we extract the 3D road median points from the 2D output. The original 3D LiDAR points which are contained within the 2D road median cell boundaries are extracted. In the next section, we present the test results of our algorithm on the road sections.

## 5. Experimentation

The dataset acquired using an ALS system along dual carriageway roads in Ireland was uploaded into the GLIMPSE system. In the first part, we applied our point cloud retrieval algorithm to efficiently access ALS data from the GLIMPSE system based on our specific requirements. We, as a user, explored the road polylines imported in the GLIMPSE system and clicked the points near two preferred road sections of dual carriageway. In each section, other input parameters were provided as  $l_c = 50\text{m}$ ,  $w_c = 50\text{m}$  and  $l_t = 1000\text{m}$ . The first 1km section consisted of road median with narrow concrete barrier, as shown in Figure 13(a), while the second 1km section contained road median with wide grass-hedge concrete barrier, as shown in Figure 13(b).



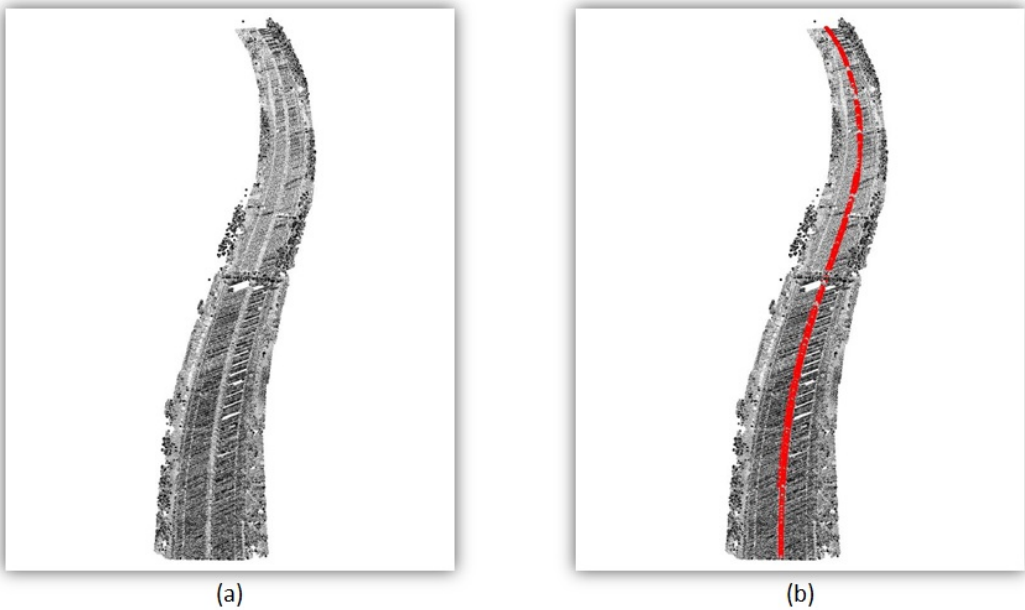
**Figure 13.** Digital images of the (a) first road section with narrow concrete barrier and the (b) second road section with wide grass-hedge barrier. (Geographic locations: (a)  $53^{\circ}21'11''\text{N } 6^{\circ}27'26.1''\text{W}$  (b)  $53^{\circ}26'9.8''\text{N } 6^{\circ}12'37.7''\text{W}$ ) (Images Courtesy: Streetview, Google)

The selected road polyline points and input parameters were used to estimate the central position and heading information of each cross-section length along the two road sections. Finally, the derived information facilitated the downloading of ALS data files from GLIMPSE in accordance with the specified parameters. In each 1km road section,  $n = 20$  number of data files were accessed with each corresponding to 50m cross-section length.

In the second part, the accessed data files were batch processed to extract the road median along the tested road sections using empirically estimated parameters. The value of  $\epsilon$  was selected as 4m, which was found to be useful in removing the highways crossing above the road sections. The threshold parameters,  $T_{elev}$  and  $T_{int}$  were applied as 200 and 100 respectively, to get an initial estimation of road median in both the sections. The cell size,  $c = 0.1$  was used to generate 2D intensity raster surface from initial estimated road median points. The length,  $l$  of linear element was used as 50, while angle  $\theta$  was calculated from the mean heading of the road polyline points. The length threshold,  $T_L$  was applied as 100 in both the road sections, however due to different width size of median, we applied different values of  $T_W$  in them. The value of  $T_W$  was provided as 5 in the first road section, while in the second section it was applied as 10. The final extracted road median in the first and second road sections are shown in Figures 14 and 15 respectively. In the next section, we present the validation of experimental results and discuss them.

## 6. Results & Discussion

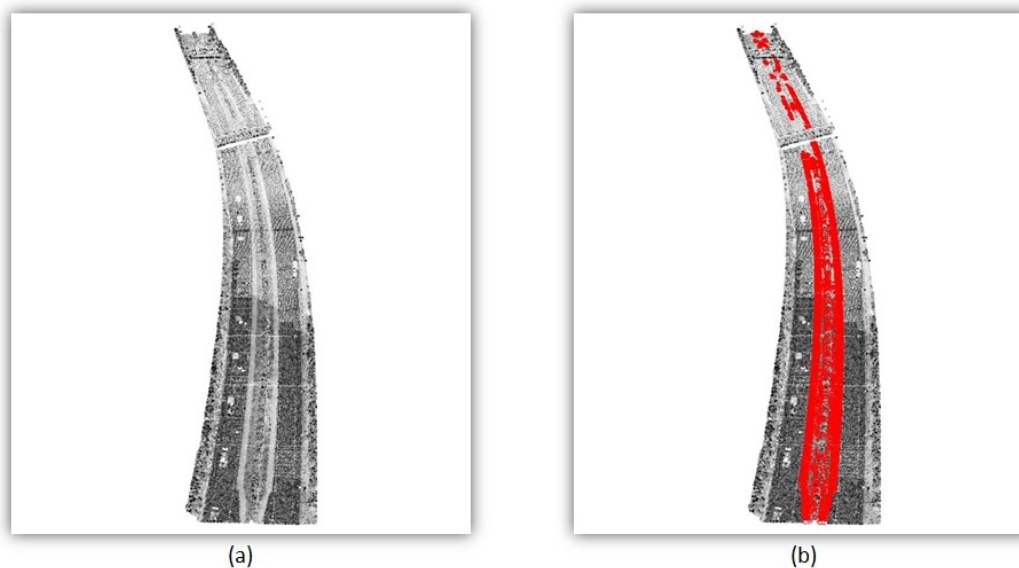
Our algorithms were able to successfully extract the road median in the tested road sections based on a specified spatial extent. We, as a user, specified input parameters in the point cloud retrieval algorithm to efficiently access the ALS data for a particular geographical area from a large volume of data stored in the GLIMPSE system. The retrieved datasets were further processed to automatically extract the road median along the tested road sections. We validated the extracted road median in the first and second road sections using a manually digitised road median. Our validation approach is based on an estimating orthogonal proximity of extracted road median points along left and right



**Figure 14.** The (a) ALS point cloud along the first road section and the (b) extracted road median points represented in red.

edges from their corresponding manually digitised road median points [39]. In our approach, we utilised the centre road polyline point which was selected at 2m intervals along the road section in each iteration. The extracted and digitised road median points were rotated toward the horizontal X-axis based on heading angle of the selected road polyline point. This horizontal rotation was done to simplify the process of estimating orthogonal distance. After rotation, linear splines were fitted to both the 2D-extracted and digitised road median points. The purpose of the spline fitting was to facilitate the interpolation of Y-axis values of the extracted and digitised road median points with respect to the known X-axis value of the selected road polyline point. Finally, we calculated an orthogonal Euclidean distance between the extracted and digitised road median points, which were rotated toward the horizontal X-axis. The positive or negative value indicated the inside or outside position of the extracted road median point with respect to the digitised road median point. This process was iterated for all the road polyline points selected at a 2m interval. Thus, the orthogonal distance was used to estimate the accuracy of extracted median in the two tested road sections. Box plots for the accuracy values of extracted road median points along left and right edges in the first and second road sections are shown in Figures 16 and 17 respectively. We also carried out statistical analyses of the accuracy values of extracted road median points along left and right edges in the first and second road sections as shown in Table 1.

Our road median extraction algorithm was able to successfully extract the road median in the first and second road sections. In the first road section, the accuracy values were found to be better than in the second road section. The minimum-maximum range values were lowest in the first section, while its accuracy values along the left and right edges were within a  $\pm 2\text{m}$  tolerance. In the second road section, 10.16% and 1.39% accuracy values along the left and right edges respectively were more than  $\pm 2\text{m}$  tolerance. Similarly, the mean, median and RMSE values in the first road section were found to be better than in the second road section. This was due to higher reflectivity obtained from the narrow concrete barrier in the first road section than from the grass-hedge concrete barrier in the second section. The non-continuous shapes of some road median also contributed to lower accuracy values in the second road section as shown in Figure 18. In both the road sections, the road median points along the right edge displayed better results than along the left edge. In the first and second



**Figure 15.** The (a) ALS point cloud along the second road section and the (b) extracted road median points represented in red.

road sections, the accuracy values of road median points along the right edge were higher within  $\pm 0.2\text{m}$ ,  $\pm 0.5\text{m}$ ,  $\pm 1\text{m}$  and  $\pm 2\text{m}$  tolerances than along the left edge.

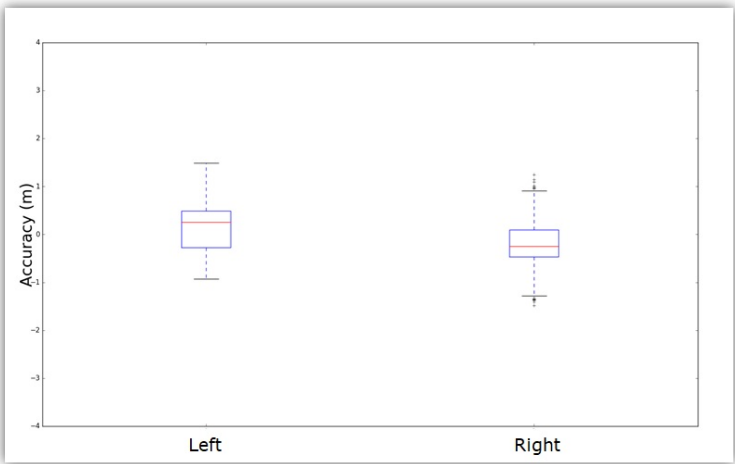
Our automated algorithm extracted the road median with an average RMSE values of 0.55m and 0.92m in the first and second road sections respectively. This low accuracy was attributed to the fact that road median at some locations along the tested road sections were missed, while at other locations, the adjacent lane markings were included in the output as false positives. High reflectivity from the lane markings led to their inclusion in the initial estimated road median during threshold applied to the intensity attribute. Further, their close proximity to road median did not enabled their removal during morphological operations. The value of LiDAR intensity attribute depends upon incidence angle of the laser pulse, the distance from the laser scanner and the illuminated surface. The normalisation of intensity attribute with respect to these factors will provide true reflectance values from the targeted objects. The use of such normalised intensity values in our algorithm will improve the quality of extracted road median. The tested road sections were also associated with highways crossing above them at some locations, which were efficiently removed based on frequency distribution analysis of elevation values. This analysis was done based on an assumption that large number of points will belong to the road surface in comparison with crossing highways. However, in case of wider highway, the large number of LiDAR points will belong to it and this will lead to the removal of road section beneath it. In the morphological operations, we applied different values of width threshold due to different width of the median in the tested road sections.

We analysed the computational performance of our two algorithms by estimating the total time taken to retrieve the data from the GLIMPSE system and then to finally extract the road median. In the first and second road sections with each 1km length and 50m cross-section length, it took approximately 37 minutes and 119 minutes respectively to retrieve and process the dataset. This analysis was performed on a computer with Intel Core i5-6600 processor @3.30GHz, 8GB RAM and a 64-bit operating system. The second road section consisted of wider road median due to which it took more time to process the dataset in comparison with the first road section.

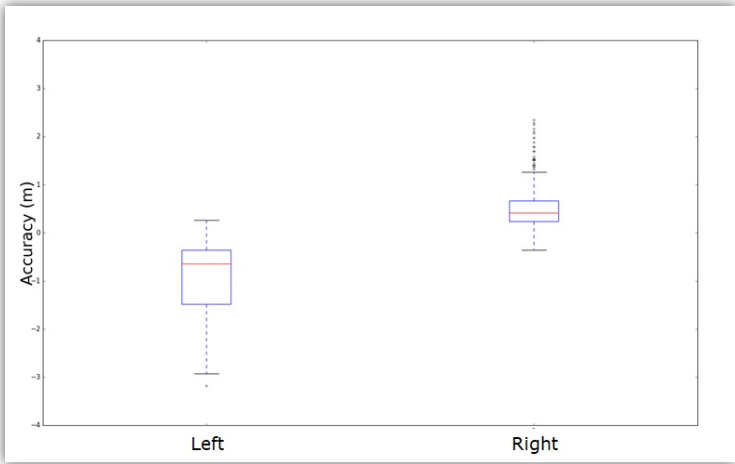
## 7. Conclusion

In this paper, we presented the GLIMPSE system that provides a comprehensive framework for storage, integration and scalability of large volume of LiDAR data. It facilitates a hierarchical





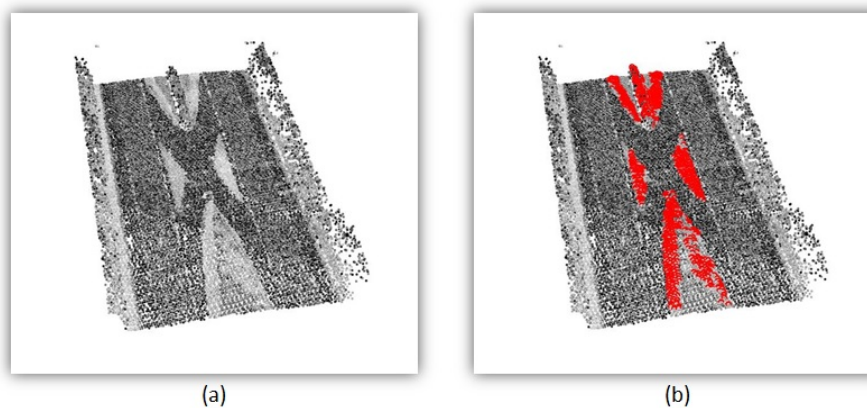
**Figure 16.** Box plot for the accuracy values of extracted road median points along left and right edges in the first road section.



**Figure 17.** Box plot for the accuracy values of extracted road median points along left and right edges in the second road section.

**Table 1.** Statistical analysis of the accuracy values of extracted road median points along left and right edges in the first and second road sections.

	First Section		Second Section	
	Left	Right	Left	Right
minimum (m)	-0.93	-1.48	-3.18	-0.36
maximum (m)	1.49	1.24	0.26	2.34
lower adjacent (m)	-0.93	-1.28	-2.93	-0.36
upper adjacent (m)	1.49	0.91	0.26	1.26
25th percentile (m)	-0.27	-0.47	-1.49	0.24
75th percentile (m)	0.49	0.09	-0.36	0.67
mean (m)	0.17	-0.20	-0.93	0.51
median (m)	0.25	-0.25	-0.64	0.41
inside ±0.2m (%)	15.66	23.69	9.47	19.40
inside ±0.5m (%)	57.63	69.08	38.57	60.28
inside ±1m (%)	92.17	94.18	63.05	89.38
inside ±2m (%)	100	100	89.84	98.61
outside ±2m (%)	0	0	10.16	1.39
RMSE (m)	0.58	0.51	1.16	0.67



**Figure 18.** The (a) ALS point cloud with some non-continuous road median along the second road section and their (b) extraction represented in red.

representation of large scale data in a geographical meaningful way, that enables the user to spatially segment the data based on its requirement. The use of such a system provides a framework for the fast retrieval of point cloud data based on an optimal spatial extent which, in turn, improves the efficiency of automated algorithms in terms of computational cost and reduced processing. We developed the methods to efficiently retrieve point clouds from the GLIMPSE system and to automatically extract the road median from the retrieved data. In our point cloud retrieval algorithm, the road vector polylines are used as secondary data source to estimate the parameters required for spatial segmentation of ALS point cloud in the GLIMPSE based on user input information. In case of MLS dataset, the navigation points can also be utilised as secondary data source, which are usually procured during the data acquisition process. The GLIMPSE system provides a single platform for storage, integration, retrieval and processing of large volume of LiDAR point cloud in a computationally efficient manner.

Our road median extraction algorithm was developed based on the assumption that LiDAR data provides elevation and intensity values, which can be utilised to distinguish the median. The use of road polylines enables to estimate the points belonging to the road surface, which in turn facilitates more accurate extraction of road median. We then perform the frequency distribution analysis of elevation values, which provides for the removal of the highways crossing above the road sections. The morphological operations based knowledge analysis is useful in completing the shape of the road median and removing other road surface elements, that are introduced through the use of thresholding. The algorithms were tested on two road sections to extract the concrete and grass-hedge based road medians. The successful extraction of road median from these multiple road section environments based on user requirement validates our two algorithms. The developed workflow can assist authorities in ensuring rapid and timely maintenance of road features along the route corridor environment.

In future work, the road median algorithm will be tested on road sections with more distinct medians. The elevation and intensity threshold values applied to get an initial estimate of the road median, were estimated empirically. However, a more robust and automated approach will need to be developed in order to get the threshold values. We will also focus on the normalisation of the intensity attribute, which will improve the quality of road median extraction. The size of input data sections and cell size of raster surfaces impact the efficiency of our algorithm in terms of computational cost. These parameters are required to be efficiently analysed to find their optimal values. Future work will also focus on the integration of LiDAR and imagery data acquired from multiple platforms in the GLIMPSE system, which will then be utilised to develop other road feature extraction algorithms.

**Acknowledgments:** Research presented in this paper was initially funded by the Irish Research Council Enterprise Partnership scheme and Strategic Research Cluster grant 07/SRC/I1168 awarded by Science Foundation Ireland

under the National Development Plan. This research also received financial support through Science Foundation Ireland grant SFI 13/IF/I2782. The research was continued at Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA), Barcelona. The authors gratefully acknowledge this support.

**Author Contributions:** Pankaj Kumar collaborated with Paul Lewis and Conor P. McElhinney in developing the concept and performing the experiments. Pankaj Kumar and Paul Lewis contributed in preparing this manuscript.

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

**Abbreviations**

The following abbreviations are used in this manuscript:

LiDAR	Light Detection and Ranging
MLS	Mobile Laser Scanning
ALS	Airborne Laser Scanning
GLIMPSE	Global LiDAR and Imagery Mobile Processing Spatial Environment
DEM	Digital Elevation Model
TIN	Triangulated Irregular Network
SDBMS	Spatial Database Management System
MBR	Minimum Bounding Rectangle
MBB	Minimum Bounding Boxes
DTM	Digital Terrain Model
UI	User Interface
NRA	National Roads Authority

**References**

1. Darnel, C. Using LiDAR to solve industry challenges. *Geoconnexion International Magazine* **2012**, *11*, 18–19.
2. Kumar, P. Road features extraction using terrestrial mobile laser scanning system. Ph.d., National University of Ireland Maynooth (NUIM), 2012.
3. Rutzinger, M.; Elberink, S.J.O.; Pu, S.; Vosselman, G. Automatic extraction of vertical walls from mobile and airborne laser scanning data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1-2 September, Paris, France **2009**, XXXVIII-3, 7–11.
4. Kumar, P.; McElhinney, C.P.; Lewis, P.; McCarthy, T. An automated algorithm for extracting road edges from terrestrial mobile LiDAR data. *ISPRS Journal of Photogrammetry and Remote Sensing* **2013**, *85*, 44–55.
5. Kumar, P.; Lewis, P.; McElhinney, C.P. Parametric analysis for automated extraction of road edges from mobile laser scanning data. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 28-30 October, Kuala Lumpur, Malaysia **2015**, II-2/W2, 215–221.
6. Chen, Q. Airborne LiDAR data processing and information extraction. *Photogrammetric Engineering & Remote Sensing* **2007**, *73*, 109–112.
7. Yang, B.; Dong, Z.; Zhao, G.; Dai, W. Hierarchical extraction of urban objects from mobile laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing* **2015**, *99*, 45–57.
8. Lewis, P.; McElhinney, C.P.; Schon, B.; McCarthy, T. Mobile mapping system LiDAR data framework. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* **2010**, XXXVIII-4, 135–138.
9. Lewis, P.; McElhinney, C.P.; McCarthy, T. LiDAR data management pipeline; from spatial database population to web-application visualization. *3rd International Conference on Computing for Geospatial Research and Applications*, 1-3 July, Washington, USA **2012**.
10. Cabo, C.; Ordonez, C.; Garcia-Cortes, S.; Martinez, J. An algorithm for automatic detection of pole-like street furniture objects from mobile laser scanner point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing* **2014**, *87*, 47–56.
11. Ming, G.; Yanmin, W.; Youshan, Z.; Junzhao, Z. Research on database storage of large-scale terrestrial LiDAR data. *International Forum on Computer Science-Technology and Applications*, 25-27 December, Chongqing, China **2009**, pp. 19–23.
12. Bruenig, M.; Zlatanova, S. 3D Geo-DBMS. *Directions Magazine* (Available: <http://www.directionsmag.com/entry/3d-geo-dbms/123603>) **2004**.

- 595 13. Schon, B.; Bertolotto, M.; Laefer, D.F.; Morrish, S.W. Storage, manipulation and visualization of LiDAR  
596 data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 25-28  
597 Febraury, Trento, Italy **2009**, XXXVIII-5, 8.
- 598 14. Guttman, A. R-trees: a dynamic index structure for spatial searching. *ACM SIGMOD International*  
599 *Conference on Management of Data*, 18-21 June, Boston, USA **1984**, 14, 47-57.
- 600 15. Samet, H. *Foundations of multidimensional and metric data structures*; Morgan Kaufmann Publisher, 2006; p.  
601 1024.
- 602 16. Agarwal, P.K.; Arge, L.; Danner, A., From Point Cloud to Grid DEM: A Scalable Approach. In *Progress in*  
603 *Spatial Data Handling: 12th International Symposium on Spatial Data Handling*; Riedl, A.; Kainz, W.; Elmes,  
604 G.A., Eds.; Springer Berlin Heidelberg, 2006; pp. 771-788.
- 605 17. Hua, L.; Zhengdong, H.; Qingming, Z.; Peng, L. A database approach to very large LiDAR data  
606 management. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information*  
607 *Sciences*, 3-11 July, Beijing, China **2008**, XXXVII, 463-468.
- 608 18. Stal, C.; Maeyer, P.D.; Wulf, A.D.; Nuttens, T.; Vanclooster, A.; Weghe, N.V.D. An optimized workflow  
609 for processing airborne laser scan data in a GIS-based environment. *The International Archives of the*  
610 *Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3-4 November, Berlin, Germany **2010**,  
611 XXXVIII, 163-167.
- 612 19. Nandigam, V.; Baru, C.; Crosby, C. Database design for high-resolution LiDAR topography data.  
613 *International Conference on Scientific and Statistical Database Management*, 30 June-2 July, Heidelberg, Germany  
614 **2010**, 1, 151-159.
- 615 20. Hongchao, M.; Wang, Z. Distributed data organization and parallel data retrieval methods for huge laser  
616 scanner point clouds. *Computers & Geosciences* **2011**, 37, 193-201.
- 617 21. Elseberg, J.; Borrmann, D.; Nüchter, A. One billion points in the cloud - an octree for efficient processing of  
618 3D laser scans. *ISPRS Journal of Photogrammetry and Remote Sensing* **2013**, 76, 76-88.
- 619 22. Cura, R.; Perret, J.; Paparoditis, N. A scalable and multi-purpose point cloud server (PCS) for easier and  
620 faster point cloud data management and processing. *ISPRS Journal of Photogrammetry and Remote Sensing*  
621 **2017**, 127, 39-56.
- 622 23. NSF. OpenTopography - LiDAR access facility. <http://opentopo.sdsc.edu/datasets?listAll=true> **2017**.
- 623 24. Dielmo. LiDAR-Online. <http://www.lidar-online.com/tools/maps/?lang=en> **2017**.
- 624 25. Clode, S.; Kootsookos, P.; Rottensteiner, F. The automatic extraction of roads from LiDAR data. *The*  
625 *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 12-23 July,  
626 Istanbul, Turkey **2004**, XXXV, 231-236.
- 627 26. Vosselman, G.; Liang, Z. Detection of curbstones in airborne laser scanning data. *The International Archives*  
628 *of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1-2 September, Paris, France **2009**,  
629 XXXVIII, 111-116.
- 630 27. Hu, X.; Tao, C.V.; Hu, Y. Automatic road extraction from dense urban area by integrated processing of  
631 high resolution imagery and lidar data. *The International Archives of the Photogrammetry, Remote Sensing and*  
632 *Spatial Information*, 12-23 July, Istanbul, Turkey **2004**, XXXV, 320-324.
- 633 28. Mumtaz, S.A.; Mooney, K. A semi-automatic approach to object extraction from a combination of image  
634 and laser data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information*  
635 *Sciences*, 3-4 September, Paris, France **2009**, 38, 53-58.
- 636 29. Elberink, S.J.O.; Vosselman, G. 3D information extraction from laser point clouds covering complex road  
637 junctions. *The Photogrammetric Record* **2009**, 24, 23-36.
- 638 30. Li, Y.; Yong, B.; Wu, H.; An, R.; Xu, H. Road detection from airborne LiDAR point clouds adaptive for  
639 variability of intensity data. *Optik - International Journal for Light and Electron Optics* **2015**, 126, 4292-4298.
- 640 31. Hui, Z.; Hu, Y.; Jin, S.; Yevenyo, Y.Z. Road centerline extraction from airborne LiDAR point cloud based on  
641 hierarchical fusion and optimization. *ISPRS Journal of Photogrammetry and Remote Sensing* **2016**, 118, 22-36.
- 642 32. Ferraz, A.; Mallet, C.; Chehata, N. Large-scale road detection in forested mountainous areas using airborne  
643 topographic lidar data. *ISPRS Journal of Photogrammetry and Remote Sensing* **2016**, 112, 23-36.
- 644 33. Boyko, A.; Funkhouser, T. Extracting roads from dense point clouds in large scale urban environment.  
645 *ISPRS Journal of Photogrammetry and Remote Sensing* **2011**, 66, S2-S12.
- 646 34. Zhou, L.; Vosselman, G. Mapping curbstones in airborne and mobile laser scanning data. *International*  
647 *Journal of Applied Earth Observation and Geoinformation* **2012**, 18, 293-304.



35. Serna, A.; Marcotegui, B. Urban accessibility diagnosis from mobile laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing* **2013**, *84*, 23–32.
36. Yang, B.; Fang, L.; Li, J. Semi-automated extraction and delineation of 3D roads of street scene from mobile laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing* **2013**, *79*, 80–93.
37. Wang, H.; Luo, H.; Wen, C.; Cheng, J.; Li, P.; Chen, Y.; Wang, C.; Li, J. Road boundaries detection based on local normal saliency from mobile laser scanning data. *IEEE Geoscience and remote sensing letters* **2015**, *12*, 2085–2089.
38. McElhinney, C.P.; Kumar, P.; Cahalane, C.; McCarthy, T. Initial results from european road safety inspection (EURSI) mobile mapping project. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 21-24 June, Newcastle, UK **2010**, XXXVIII, 440–445.
39. Kumar, P.; Lewis, P.; McElhinney, C.P.; Boguslawski, P.; McCarthy, T. Snake energy analysis and result validation for a mobile laser scanning data based automated road edge extraction algorithm. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **2017**, *10*, 763–773.
40. Kumar, P.; McElhinney, C.P.; Lewis, P.; McCarthy, T. Automated road markings extraction from mobile laser scanning data. *International Journal of Applied Earth Observation and Geoinformation* **2014**, *32*, 125–137.
41. Guo, J.; Tsai, M.J.; Han, J.Y. Automatic reconstruction of road surface features by using terrestrial mobile lidar. *Automation in Construction* **2015**, *58*, 165–175.
42. Teo, T.A.; Chiu, C.M. Pole-like road object detection from mobile LiDAR system using a coarse-to-fine approach. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **2015**, *8*, 4805–4818.
43. Yan, W.Y.; Morsey, S.; Shaker, A.; Tulloch, M. Automated extraction of highway light poles and towers from mobile LiDAR data. *Optics & Laser Technology* **2016**, *77*, 162–168.
44. Kumar, P.; Lewis, P.; McElhinney, C.P.; Abdul-Rahman, A. An algorithm for automated estimation of road roughness from mobile laser scanning data. *The Photogrammetric Record* **2015**, *30*, 30–45.
45. Guan, H.; Li, J.; Yu, Y.; Chapman, M.; Wang, H.; Wang, C. Iterative tensor voting for pavement crack extraction using mobile laser scanning data. *IEEE Transactions on Geoscience and Remote Sensing* **2015**, *53*, 1527–1537.
46. Diaz-Vilarino, L.; Gonzalez-Jorge, H.; Bueno, M.; Arias, P.; Puente, I. Automatic classification of urban pavements using mobile LiDAR data and roughness descriptors. *Construction and Building Materials* **2016**, *102*, 208–215.
47. Rutzinger, M.; Pratihast, A.K.; Elberink, S.O.; Vosselman, G. Detection and modelling of 3D trees from mobile laser scanning data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 21-24 June, Newcastle, UK **2010**, XXXVIII, 520–525.
48. Maneewongvatana, S.; Mount, D.M. Analysis of approximate nearest neighbor searching with clustered point sets. *First Workshop on Algorithm Engineering and Experimentation*, 15-16 January, Baltimore, US **1999**, p. 20.
49. Kumar, P.; Lewis, P. Automated extraction of road median from airborne laser scanning data. *The 38th Asian Conference on Remote Sensing*, 23-27th October **2017**, pp. 1–9.