LIDAR-BASED VEHICLE SEGMENTATION

Á. Rakusz – T. Lovas – Á. Barsi

Department of Photogrammetry and Geoinformatics
Budapest University of Technology and Economics
H-1111 Budapest, Műegyetem rkp. 3.
Hungary
{rakusz.adam, lovas.tamas, barsi.arpad}@fmt.bme.hu

Commission II, WG II/2

KEY WORDS: LIDAR processing, Object extraction, Algorithm comparison, Point cloud segmentation

ABSTRACT:

The paper focuses on a particular aspect of feature extraction from LiDAR data. To support transportation flow data estimation, points reflected back from vehicles should be extracted from a LiDAR cloud. A simple thresholding can certainly provide a good starting point to solve this task, but in order to achieve a robust solution there are several other tasks that should be addressed. First, the road itself should be identified (actually continuously followed) to define the search window for the vehicles. Then, the surface of the road must be modeled to obtain true elevation of the vehicle (which is measured in the normal direction of the surface). Once the LiDAR points representing a vehicle have been obtained, at minimum the vehicle orientation should be determined such as travel direction. This paper introduces a technique to accomplish the above mentioned tasks. The road is followed by the guidance of an initial coarse centerline description. Then a preprocessing phase takes place, the point cloud is segmented to get the vehicle blobs. The segmentation is based on standard image processing methods, such as histogram thresholding or edge detection techniques, both methods are currently under consideration. In the next step, vehicle outlines are created using statistical parameters, such as standard deviation of height values or height "texture" measures. The robustness of the process has been improved by using Delaunay-triangulation to test slope measures. The newly developed method has been implemented in Matlab environment and provides visualization tools for diagnostic purposes. The obtained results have proven that our algorithm performs well in effectively extracting vehicles from LiDAR data that can contribute to the complex task of traffic flow information evaluation.

1. INTRODUCTION

LiDAR stands for Light Detection And Ranging. Regarding the data acquisition concept it is similar to radar, except it operates with laser light. Flown with a helicopter or fixed wing aircraft, eye-safe laser pulses are sent to the ground and their reflections are recorded. Accurate distances are then calculated to the points on the ground and therefore elevations can be determined for not only the ground surface but the buildings, roads, vehicles, vegetation and even something as thin as e.g. power lines (Barsi et al. 2003), (Tovari 2002).

LiDAR technology provides a point cloud, in which all points have three coordinates and – in most cases – intensity values. The laser pulse reflects from the closest object, known as first pulse, (in this paper we have used only this).

The greatest difference between LiDAR and other distance measurement methods is the data structure. We have points along a narrow strip, where we don't know exactly where the beam is reflected from, therefore, we cannot add any attribute information to these points. In addition, when we use LiDAR for transportation purposes, we have to adjust our calculation by shortening against flight direction and elongating along it. Therefore, our task is to develop methods for selecting, separating and classifying points using different approaches. One objective was to extract vehicle points for classification in order to support transportation flow data estimation.

In LiDAR processing there are two completely different possible approaches. If we don't want to lose information, we have to use sampled 3D data points. But it is much easier if we handle the data set as an image, after interpolating it to a regular square grid, where the intensity comes from the height of the point, certainly calibrated. In this paper we have used the original sampled data and the resampled, interpolated form of it (image). The visual control with images is also easier.

The methods were tested with two different datasets, with diverse point density. One of them was acquired in July 2000 over the State Route US 35 (East of Dayton, OH), whilst the other is about Toronto, Canada, in winter 2004. (See flight data in Table 1) (Toth et al. 2003).

	Flight 1	Flight 2
Flying Height (AGL):	470 m	660 m
Average Ground Speed:	56.6 m/sec	58.5 m/sec
Heading:	290 degrees (North-West)	250 degrees
Scan Frequency:	50 Hz	46 Hz
Field of View (Half Angle):	6 degrees	20 degrees
Laser Repetition Rate:	10 kHz	70 kHz
Point density	1.5 points/m^2	2.4 points/m^2
Area	Route 35, Dayton, Ohio	Toronto, Canada

Table 1. Flight parameters

In the Ohio data set, 20-30 points are reflected from a passenger car (and 40-60 from a truck) traveling along the flight direction.

In the opposite direction, this value is under 10 (Lovas et al. 2004).
Figure 1 shows a cross section of the LiDAR strip, which is a view about the cross section at the centerline.

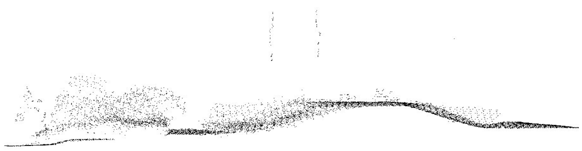

Figure 1. Cross section of the road with surroundings

In this paper we present three, different methods for vehicle segmentation (Figure 2).

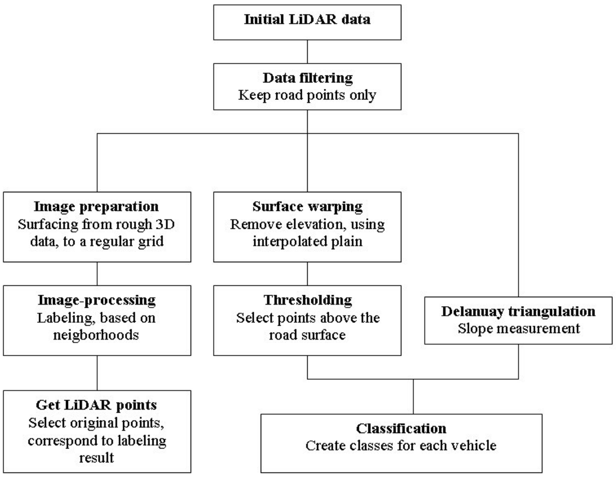

Figure 2. Data processing flowchart

2. DATA FILTERING

In Figure 1 not only the road but also the surroundings (vegetation, ground work, landmarks, transmission line and vehicles) can be seen. First, we can easily detach the points not belonging to vehicles. If the position of the centerline of the road, and the number of lanes and their width are known, the usable swath can be obtained.
If the centerline is not given, we can develop a semi-automatic algorithm that is based on the cross sections. Roads are usually located on embankments. We have to mark one cross section and the road direction, then using the calculated parameters from the sample section (height of the trapeze, angles and lengths, road slope from centerline), and the basic properties of the road (angle of slope - both for the long and cross direction, curve radius). Then the same data for the next cross-section should be calculated, close to the last one (e.g., 10 meters). Combining this with the original dataset we can decide whether the calculation is right or not. If the calculations are correct, and the matching is good, the middle position and the parameters of the given cross-section can be recorded. If not, the same calculation with the same parameters in a different position should be performed (rotating by a small angle around the middle point of the last recorded cross section). If in this position a properly matching cross section cannot be found, this has to be ignored, and a shorter distance from the last recorded one have to be used.

3. THRESHOLDING

In order to perform vehicle extraction, we have to separate all points above the average road height in a local environment. We cannot accomplish that without knowing the road level at every position of the vehicles or other objects (e. g., vegetation). Using a zone with a little bit smaller width than the sampling density, we can ensure that only one point can fall inside. A polyline connecting these points and the centerline represent the road surface. This should not be very accurate because we use only the first pulse reflected from the tops; the lowest part of the vehicle is the engine hood, which is higher above the road than the distance between the points (Figure 3.). All points above that surface possibly belong to a vehicle. This new set is the basis for our further examinations.

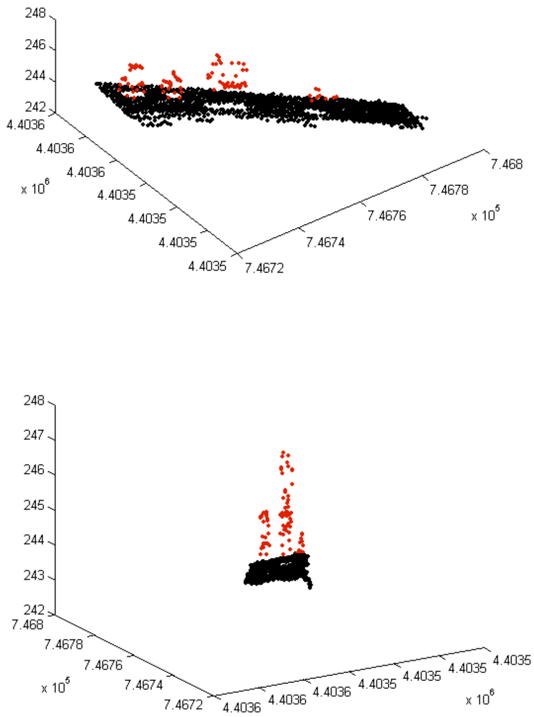

Figure 3. Test data in 3D: rear and front view

In case of sloping roads the same height could represent a road and also a vehicle. In order to identify vehicles more easily we have to compensate for the slope of the road. The centerline of the road is given or being calculated only horizontally. In Figure 4, the sampled point heights are shown along the centerline. The long section of the road is shown, where the sloping angles are different, but can be approximated with lines segments (marked in red). Decreasing all point height to the value of the regression line, at the point's horizontal position this goal can be achieved (Pitas 2000)

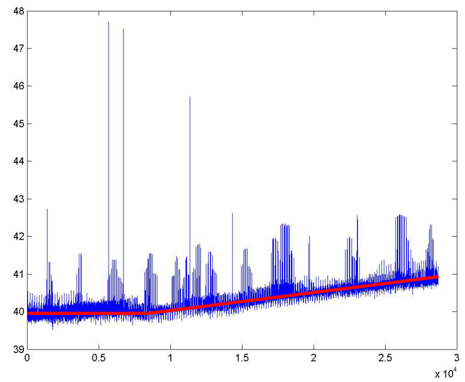

Figure 4. Point heights along the long-section and the interpolated elevation of the centerline

4. DELAUNAY-TRIANGULATION

The other way to get vehicle points is using Delaunay-triangulation. Figure 5 shows the calculated triangles on the test area.

The grayscale image is color-coded for the triangle slopes; hence this can be the basis of vehicle segmentation. This method is accurate, precise, and less sensitive for errors.

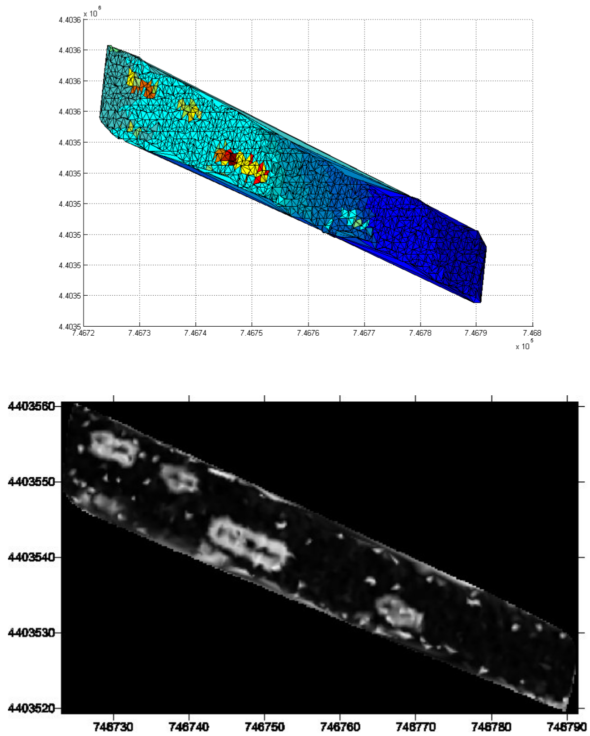

Figure 5. Result of the Delaunay-triangulation and the grayscale image, representing triangles slope

Using Delaunay-triangulation for the initial data set, the non-regular network of triangles can be created; this method minimizes data loss caused by interpolation, however, it is hard to handle this network. The creation of these triangles is automatic, therefore, the special characteristics of the road, vehicles and landmarks cannot be considered.

The method presented here is based on triangulation. The gradients of the triangles are very dissimilar, but there are only slight differences between the heights of points in neighboring triangles.

First, we calculate the slope angle of each triangle to a reference plane that is practically the approximate position of the road surface; this value in radians is linked to the center of gravity of the triangles. This is still insufficient information to separate vehicles because triangles contain also points reflected from the surface of the road, therefore, resulting in very different slope angles. In the test data, the horizontal distance between points is generally less than the distance from the road surface and the points reflected back from vehicles. The perimeter of a triangle gives information about the slope of it; if both the slope and the perimeter of a triangle is big, then the triangle most likely belongs to the boundary of a vehicle. Using the centerline of the road and lane parameters, triangles at the sides of lanes could be filtered out.

A vehicle is found, if the bordering triangles are tend to have slope angles in the same direction, like the walls of a tent, and there is a space between these “walls”. The vehicle envelope can be achieved, fitting a polygon on the points found (only vertically), one for higher ones, one for the lower ones, which are on the same level as the road. Separating points for these two polygons could be achieved using heights; inside the higher level polygon -as a fence- including points fitting on polygon we could get the vehicle points. The extensions of vehicles could also help filter out some wrong triangles. (See Figure 3 for vertical, Figure 4 for horizontal triangle errors) (Sederberg et al. 1985).

5. CLASSIFICATION

The segmentation of vehicles from this pre-processed data is based on the remaining triangles. First, the distances between each pair of observation points are calculated. These can be used to compute the hierarchical cluster information based on the single linkage algorithm. Then, we could define clusters for each vehicle using the resulting cluster-tree.

For each cluster representing a vehicle we make a principal component analysis, to get eigenvectors, which stretches a coordinate system. The length of the vectors carries information about the dimensions of the vehicle, while the orientation has about the moving direction.

In the introduction we mentioned the potential of using image-processing techniques for segmentation.

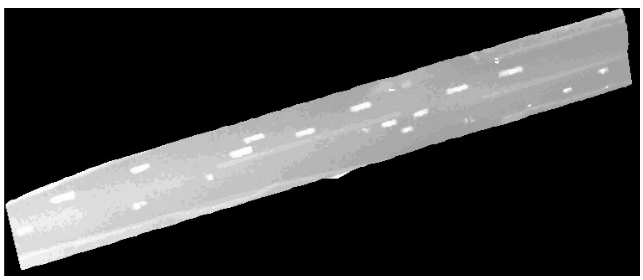

Figure 6. Intensity image of the surface

The intensity of pixels is derived from their height, after creating the covering surface (Figure 6.). The network is regular rectangular, the pixel size should be less than the minimal distance between the points in the point cloud. We used bilinear interpolation for calculating the pixel values.

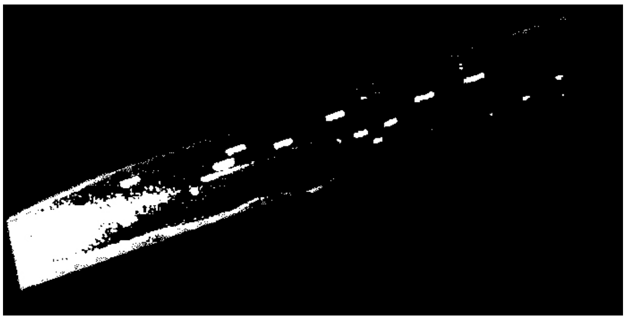

Figure 7. Binary image

It is easier to use a binary image for classification, therefore, the original image has to be converted, using an appropriate threshold value. As described above, because of the elevation of the road, the same pixel value could represent both vehicle and road points. (It can be seen in Figure 5 and 6.)
In Figure 7, the labeling method can be easily applied to get connected components, which based on the investigation of the neighboring pixel values (4 or 8 neighbors), and sort them into classes; each separate class gets a unique identifier.

6. CONCLUSIONS

This paper demonstrated a way to extract vehicles from a LiDAR point cloud to provide data for vehicle classification.
The comparison results of the introduced three methods are shown in Table 2.

	Route 35, Dayton, Ohio	Toronto, Canada
Thresholding	4/4	13/14
Triangulation	4/4	14/14
Image labeling	4/4	12/14

Table 2. Segmentation methods comparison
(Extracted vehicle/present vehicles)

In Figure 8, the results obtained for one vehicle are shown. Because of the automatic Delaunay triangulation, which operates without any constraints, one point is missing, but it provides the most precise boundary of the vehicle out of the 3 algorithms. The fastest method is the thresholding-based technique.

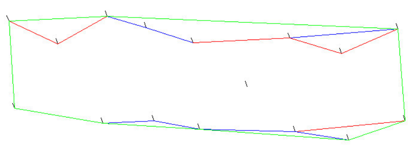

Figure 8. Vehicle points with polygon boundary
(Red - Triangulation, Green - Labeling, Blue - Thresholding)

The results are promising, but further work is needed for refined segmentation. In addition, using denser point cloud is expected to result in better point selection.

Acknowledgements

The authors would like to thank to Woolpert LLC and Optech International for providing the LiDAR datasets, and to Charles K. Toth, Center for Mapping, The Ohio State University, for his extremely useful contribution during the research.

References

Barsi, Á., Detrekői, Á., Lovas, T., Tóvári, D., 2003. Data collection by airborne laserscanning (in Hungarian). *Geodézia és Kartográfia*, Vol. LV, No. 7, pp. 10-17

Tóvári, D., 2002. Analysis of airborne laserscanner data.

Toth, C. K., Barsi, A., Lovas, T., 2003. Vehicle recognition from LiDAR data. *International Archives of Photogrammetry and Remote Sensing,* Vol. XXXIV, Part 3/W13, pp. 162-166

Lovas, T., Barsi, A., Toth, C. K. 2004: Detecting Moving Targets in Laser Scanning, Proc. ASPRS Annual Conference, May 23-27, in press

Pitas, I. (2000): Digital Image Algorithms and Applications, John Wiley & Sons, Inc.

Sederberg, T. W. and Anderson, D. C.,1985 „steiner Surface Patches," IEEE Computer Grapichs and Applications, May 1985