Next Generation Map Making: Geo-Referenced Ground-Level LIDAR Point Clouds for Automatic Retro-Reflective Road Feature Extraction

Xin Chen Matei Stroila Ruisheng Wang

Brad Kohlmeyer Narayanan Alwar Jeff Bach

NAVTEQ Corporation
425 West Randolph Street
Chicago, IL 60606 USA
xin.chen@navteq.com

ABSTRACT

This paper presents a novel method to process large scale, ground level Light Detection and Ranging (LIDAR) data to automatically detect geo-referenced navigation attributes (traffic signs and lane markings) corresponding to a collection travel path. A mobile data collection device is introduced. Both the intensity of the LIDAR light return and 3-D information of the point clouds are used to find retro-reflective, painted objects. Panoramic and high definition images are registered with 3-D point clouds so that the content of the sign and color information can subsequently be extracted.

Categories and Subject Descriptors

I.4 [**Artificial Intelligence**]: Image Processing and Computer Vision

General Terms

algorithm

Keywords

LIDAR, geo-reference, retro-reflective, lane marking, sign, road, ground-level

1. INTRODUCTION

Accurately modeling real-world data in three-dimension (3-D) has broad applications in surveying, asset inventory, autonomous robot navigation and map making. For example, navigation systems provide visual guidance and instructions using map data, including navigation-related attributes such as traffic signs and pavement markings. The collection and extraction of such geographic data is labor intensive and time consuming. A mobile data collection system and automatic feature extraction software are desirable.

With the rapid development of LIDAR sensors and global positioning systems (GPS), large scale, accurate and dense 3-D point clouds can be captured. Recent research by Munoz et.al. [8][9][7] processed 3-D point clouds for automated scene understanding. However, to our knowledge, there is little work that combines the intensity of LIDAR light return with 3-D coordinates to extract lane markings and signs. Figure 1 shows 3-D point cloud visualization with and without intensity (color is used to represent the elevation). Apparently, the higher intensity return of the retro-reflective paint indicates lane markings and signs.

Lane markings and signs are one of the most important attributes on the roadway for navigation. In contrast to traditional methods using the 2-D imagery modality to automatically detect signs [4] and lane markings for Advanced Driver Assistance System (ADAS) [6] and vision-based intelligent vehicles [3], we identify these objects from the LIDAR 3-D point clouds using intensity return values. The advantages of this approach are (1) Direct acquisition of 3-D coordinates once the objects are detected; (2) less redundancy for processing than video-based object detection; (3) invariant to lighting conditions and shadows; (4) robust for irregular shapes (e.g. partial occluded signs, sharp lane curves, etc.) (4) more contrast from the background.

In Section 2, we will discuss geo-referenced LIDAR data collection on a vehicle-based platform moving at the posted speed limit. Sign extraction from LIDAR is presented in Section 3, followed by lane marking extraction from LIDAR in Section 4. We conclude in Section 5.

2. DATA COLLECTION

Data is collected by NAVTEQ using the equipment shown in Figure 2. The data collection apparatus features a 360 degree LIDAR sensor (Velodyne HDL-64E), panoramic camera (Ladybug 3), high definition cameras, GPS, Inertial Measurement Unit (IMU) and Distance Measurement Instrument (DMI). The LIDAR sensor operates on 64 lasers mounted on upper and lower blocks of 32 lasers each and the entire

17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS) '09 Seattle, WA USA

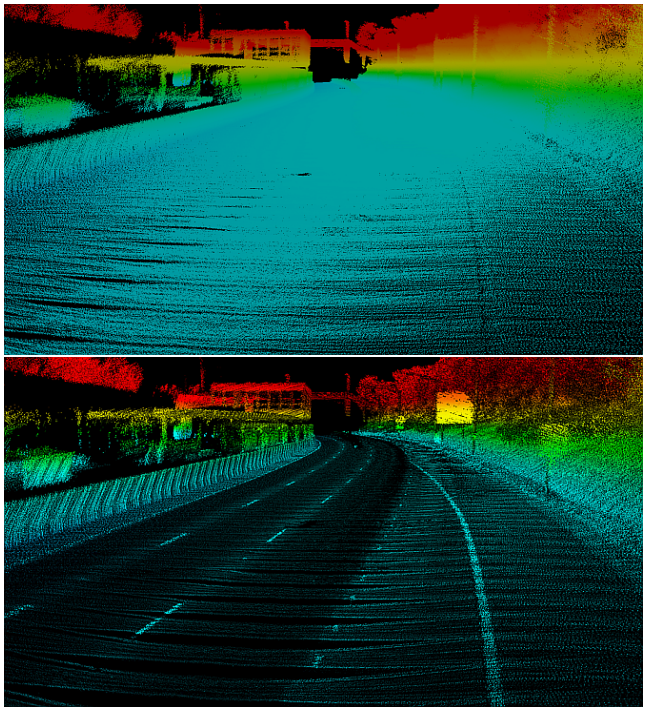

Figure 1: Lidar point clouds with (bottom) and without (above) intensity of the light return. (An elevation-based color map is used for an easier visual perception.)

unit spins. The spin rate during the data collection is 600 RPM. The unit inherently delivers a 360-degree horizontal field of view (FOV) and a 26.8 degree vertical FOV. The rotation angle of the spin associated with the 3-D point is recorded for data processing. This sensor provides usable returns of up to 120 meters and generates over 1.3 million points per second. The LIDAR sensor is mounted towards the back of the vehicle and rotates around the axis highlighted by the red dashed line in Figure 2. The vehicle drives at the posted speed limit from urban roads to interstate highways while collecting the data. Imagery data is also collected and geo-referenced.

GPS, IMU and DMI, with positioning software, generate accurate and reliable 3-D information including roll, pitch, heading direction, latitude, longitude and altitude of the data collection system, even in the area where the satellite signal is occasionally not reliable. We transform each raw LIDAR 3-D point's local 3-D coordinate (i.e. relative to the sensor) to global coordinates, i.e. latitude, longitude and altitude so that each 3-D point is geo-referenced. Figure 3 shows the 3-D point clouds collected using this data collection system and rendered on top of the maps and satellite imagery.

3. SIGN DETECTION

In this section we present a method to detect retro-reflective signs using LIDAR point clouds with intensity. The intensity is a measure of the retro-reflectivity property of the scanned surface. The main assumption is that the signs do not have significant paint degradation.

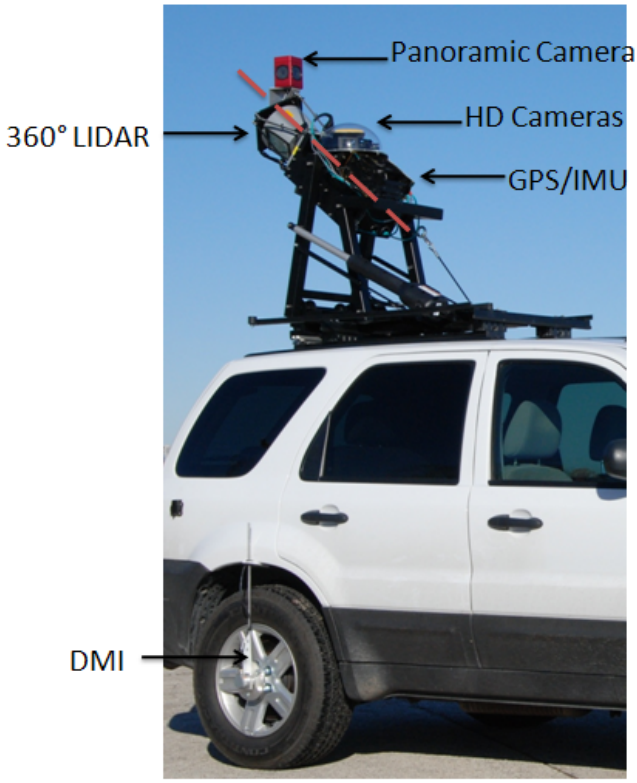

Figure 2: Data collection vehicle mounted with 360 degree LIDAR, panoramic camera, high definition cameras, IMU/GPS and DMI

3.1 Algorithm

The point cloud has the following data attributes per point: 3-D coordinates, intensity, distance to sensor, sensor angle and time stamp. The detection of the retro-reflective surfaces (see Figure 4) consists of the following steps (we specify in parentheses the parameter values used in our prototype implementation):

Data filtering: We first use heuristics to reduce the data size and possible false positives. We filter the point clouds based on the distance to sensor (less than 25 meters), sensor angle (200 degree interval centered at the straight-up direction), and intensity (more than 180 on a 0-255 scale). The intensity threshold is a very important parameter that depends on the LIDAR sensor calibration. We set this number by experimentation. We compute the time histogram of the filtered data and cluster the time interval bins with more than a threshold number of points (100) per histogram bin. We then spatially analyze each time cluster as follows.

Data clustering: Given a cluster of points, we put these points in a 3-D grid structure according to the data size and fixed interval length (0.2 meters). Each grid cell contains a certain number of points. We assign binary values to the grid cells based on the number of points that they contain: if larger than a threshold (3), 1, else, 0. Then, we compute the connected components of the resulting binary 3-D image. Each connected component represents a cluster of points with high intensity.

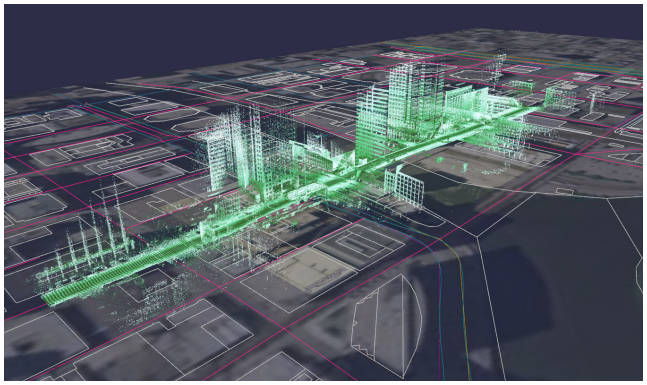

Figure 3: Geo-referenced 3-D point clouds rendered on map and satellite imagery

Geometry fitting: We fit a plane to this cluster of points using the RANSAC algorithm [5] to remove outliers. We eliminate false positives by filtering out the planes for which the plane normal orientation is not close to the driving direction (cosine of the angle less than 0.5). The points that fit this geometry well are projected to the plane. We further compute the convex hull of the resulting planar point clouds using Akl's and Toussaint's algorithm [2], and then, a minimum rectangle is fit to the convex hull using Toussaint's algorithm [10]. We use the CGAL implementation of the aforementioned algorithms, [1].

Since the point clouds are registered with the captured images, the bounding boxes of the detected retro-reflective surfaces are used to crop the registered images (see Figure 5). The corresponding content (speed limit, advisory, turn restriction, guide and other traffic signs) can be automatically classified based on object recognition.

3.2 Performance

On a laptop with an Intel Core 2 Duo T7800 @ 2.6 GHZ processor, and with 3.5 GB RAM memory, using one core only, 2.5 million points captured in an 80 meter driving distance are processed in average of 7 seconds. We tested the prototype on 7 miles of highway. Out of 104 traffic signs of interest the program detected 102 ($\approx$ 98% recognition). The two missed signs have significant paint degradation, so the corresponding points have low intensity values. Typical false positives are billboards, passing cars, and signs that are not traffic related. We have very few false positives (5, $\approx$ 4.9%) since we filter out retro-reflective surfaces that are too far from the sensor, too close to the road surface, or that are not transversally oriented (close to) to the road.

4. LANE MARKING RECONSTRUCTION

In this section we present a fully automatic method to 3-D lane marking extraction from LIDAR point clouds. Our method does not require a lane model, and can extract all visible 3-D lane markings from LIDAR through a one-button click. All the points are transformed from WGS84 to Universal Transverse Mercator (UTM) coordinates so that metric distance is directly available. We break the entire point clouds representing a travel path into slices, each of which is 36 meters along the trajectory, and process them individually. The outputs of 3-D lane markings include 3-D coordinates along the lane marking's central lines, lane marking types, and topological relationships among detected lane markings etc. The approach involves two stages: road surface detection and lane marking extraction.

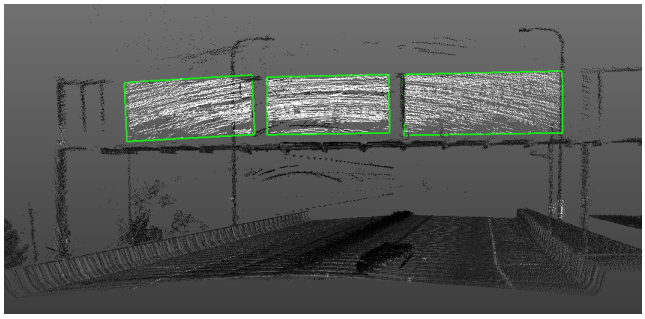

Figure 4: Lidar point clouds and detected retro-reflective surfaces

Figure 5: Rendering of the bounding boxes of the detected retro-reflective surfaces using the registration of the images with the point clouds.

4.1 Road Surface Detection

The road surface is detected by finding the road boundaries including raised curbs and barriers and border lines between paved and unpaved areas. There is significant difference of elevation distribution on each side of the boundary. The paved road surface tends to have consistent and locally uniform elevation values.

First, all the points that are 30 meeters farther away (on the ground) from the trajectory are discarded, because they are unlikely to belong to the same road.

The ground elevation e_0 is estimated by averaging the elevations e_i from the lasers directly striking the ground as shown in Equation 1

$$e_0 = \frac{1}{n} \sum_{|ang_{e_i} - ang_{grnd}| < \epsilon} e_i \qquad (1)$$

where ang_{grnd} is the rotation angle of the laser approximately intersecting with the driving direction on the ground. ang_{e_i} is the rotation angle of the laser e_i. ϵ is an angle range that includes several 3-D points used to estimate the ground elevation and n is the number of lasers in the range. All

points with elevations 0.25 meter lower or higher than e_0 are discarded as non-ground points.

For each scan line of the LIDAR scanner, i.e. one full cycle of rotation, the left and right boundaries (lb and rb, respectively) are identified by finding the first local change (indicated by standard deviation of the elevations of 10 adjacent points in one scan line) of elevation from the center to the left and to the right. All of the points to the left of the lb and to the right of rb are discarded as non-road points. This classification of road and non-road points repeats for each scan line.

Figure 6 shows the original, 3-D point cloud data and the detected road surface highlighted in red. Note that there is a gap in the road surface due to the occlusion of the other vehicles that block the LIDAR scan.

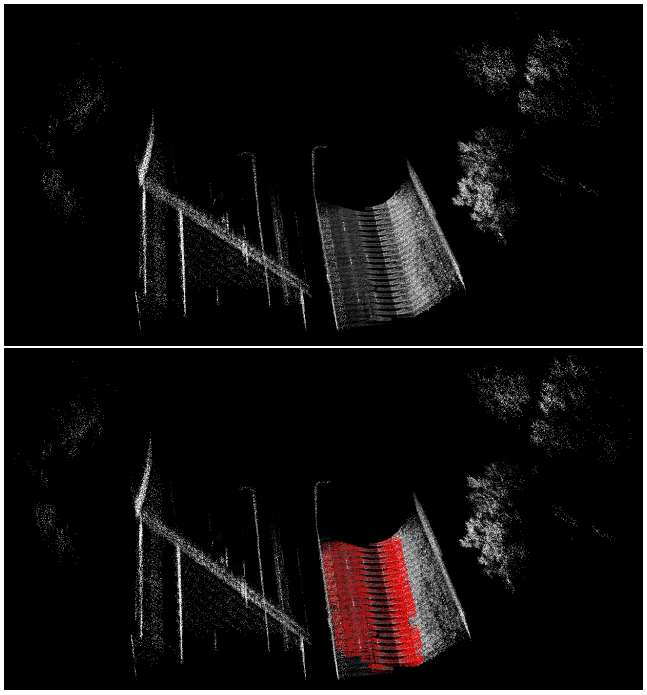

Figure 6: Original LIDAR 3-D point clouds (above) and the detected road surface highlighted red (bottom)

4.2 Lane Marking Extraction

Candidate lane marking localization Figure 7 shows the intensity values of the LIDAR light return of one scan line on the road surface. The three peaks correspond to the lane markings. For each scan line we select the peaks as candidate lane marking points using adaptive thresholding, i.e. the threshold is invariant to absolute values of LIDAR light return. The results are shown in Figure 8.

Hough Transform Clustering We generate 2-D binary images so that image processing algorithms can be employed to extract lane markings. The 3-D candidate lane marking points are employed to form a 2-D image by removing the elevation coordinates. The value of each pixel is 1 if it corresponds to a lane marking point and 0 otherwise. Hough Transform is then applied to the road reflectance images to detect all candidate lane markings as shown in Figure 9. One

Figure 7: The intensity values of LIDAR light return of one scan line of the road surface

disadvantage of Hough Transform is that the program has to specify the number of lines to be detected. In our case, we set a large number (140) to guarantee that all potential lane markings will be extracted and in particular dashed lane markings. Nevertheless, many false detections caused primarily by image noise contained road shoulder areas.

Refinement of candidate lane markings To eliminate false detections and retain correct lane markings, we employ trajectory constraint and geometry check, with the assumption that the lane markings are roughly oriented along the vehicle's trajectory direction. The vehicle's trajectory and the orientation of each detected line are computed first. Since the computation of the trajectory is not very accurate, we set a range around the trajectory's direction (i.e. 5 degree) as a trajectory constraint. The orientation of detected lines outside of this range will be considered as false positive and eliminated. This step removes most of the false detections, though some still remain. For each remaining detected line, we employ geometry check to filter out false lane markings. The distance between two lines and the voting number from Hough Transform are used as two criteria to retain correct lane markings and remove the false ones. The results of such refinement of potential lane markings is shown in Figure 10.

3-D lane marking generation Results from previous steps provide rough locations for each lane marking. To compute accurate locations of lane markings, we create a bounding box around each lane marking and apply RANSAC curve fitting to localize each lane marking accurately within the 2-D image. The types of lane marking can also be identified by examining the distance interval between point clusters. Currently we differentiate two types of lane markings: solid and dashed. To better represent lane markings, we select a point every 10 centimeters along fitted lines. The final coordinate of each point is an average of points within a 10 by 10 centimeters rectangular box surrounding that point. The 2-D points are then traced back to 3-D LIDAR point clouds to obtain the 3-D coordinates of lane markings. Figure 11 shows the extracted 2-D lane markings.

4.3 Performance

Our current implementation uses MATLAB. We tested a data set collected from 600 meters of highway travel featuring over 23 million LIDAR points on a laptop with an Intel

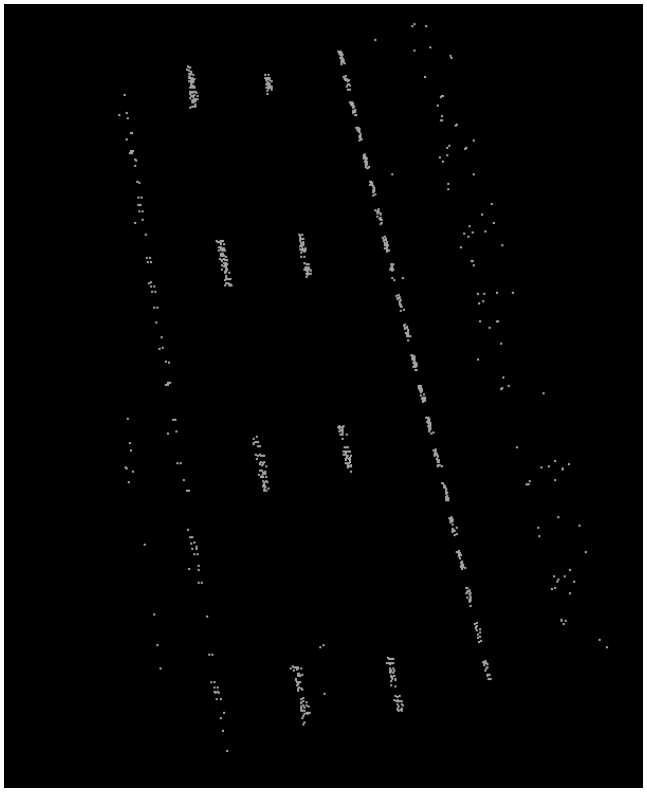

Figure 8: The detected candidate lane marking points in 3-D

Core 2 Duo CPU P9500 2.53 GHZ processor, with 3.48 GB RAM memory (only one core was used). The total processing time for this data set is about 3.1 minutes. The lane markings are well extracted as shown in the Figure 12.

5. CONCLUSION

In this paper, we present a novel method to automatically identify traffic signs and lane markings, one of the most important road navigation attributes, from mobile ground-level LIDAR collection. The 3-D world coordinates as well as the intensity return values of LIDAR light are used for segmenting the region of interest and for geometry modeling. With pixel-level registered imagery data, we can subsequently identify the color information and the content of the signs. We have processed data on a large scale and achieved satisfactory accuracy and performance.

We are currently working on the following potential challenging situations: (1) occlusions from other vehicles and pedestrians, particularly for lane marking extraction, which results in inaccurate road boundary detection and gaps between lane markings; and (2) poor quality of the paint which makes the regions of interest less distinctive.

6. REFERENCES

[1] CGAL, Computational Geometry Algorithms Library. http://www.cgal.org.

[2] S. G. Akl and G. T. Toussaint. A fast convex hull algorithm. *Inform. Process. Lett.*, 7(5):219–222, 1978.

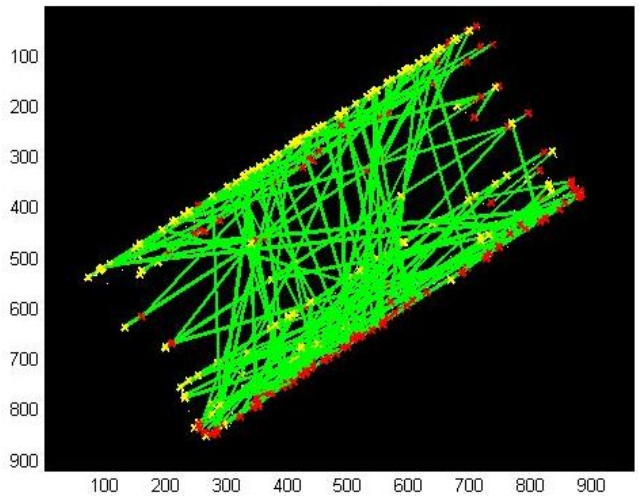

Figure 9: Candidate lane marking detection by Hough Transform

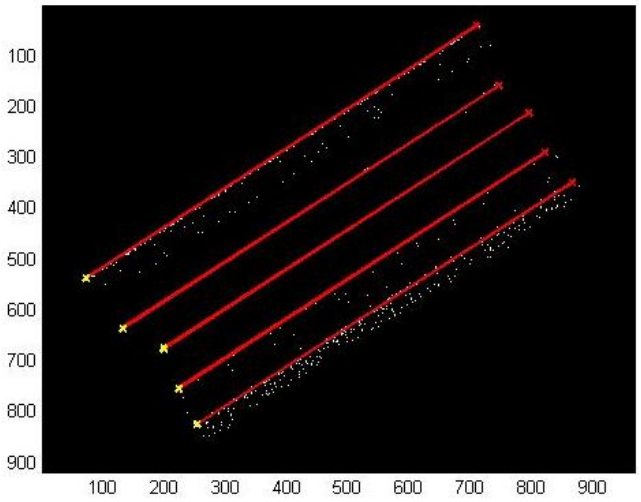

Figure 10: Lane markings after refinement

[3] M. Bertozzi, A. Broggi, A. Fascioli, and R. F. A. Vision-based intelligent vehicles: State of the art and perspectives. *Robotics and Autonomous Systems*, 32:1–16, 2000.

[4] X. Chen, M. Stroila, J. Lynch, N. Alwar, B. Kohlmeyer, and J. Bach. Towards next-generation map making. *1st IEEE International Workshop on Media Information Analysis for Personal and Social Applications in conjunction with IEEE International Conference on Multimedia and Expo (ICME)*, July 2009.

[5] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[6] J. McCall and M. Trivedi. Video-based lane estimation and tracking for driver assistance: survey, system, and

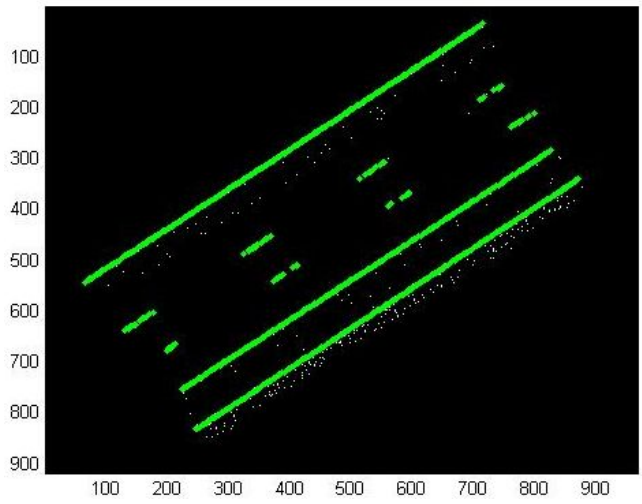

Figure 11: 2-D lane markings after RASANC curve fitting

evaluation. *IEEE Transactions on Intelligent Transportation Systems*, 7(1):20–37, March 2006.

[7] D. Munoz, J. Bagnell, N. Vandapel, and M. Hebert. Contextual classification with functional max-margin markov networks. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2009.

[8] D. Munoz, N. Vandapel, and M. Hebert. Directional associative markov network for 3-d point cloud classification. *Fourth International Symposium on 3D Data Processing, Visualization and Transmission*, June 2008.

[9] D. Munoz, N. Vandapel, and M. Hebert. Onboard contextual classification of 3-d point clouds with learned high-order markov random fields. *IEEE International Conference on Robotics and Automation*, May 2009.

[10] G. T. Toussaint. Solving geometric problems with the rotating calipers. *Proc. IEEE MELECON '83*, pages A10.02/1–4, 1983.

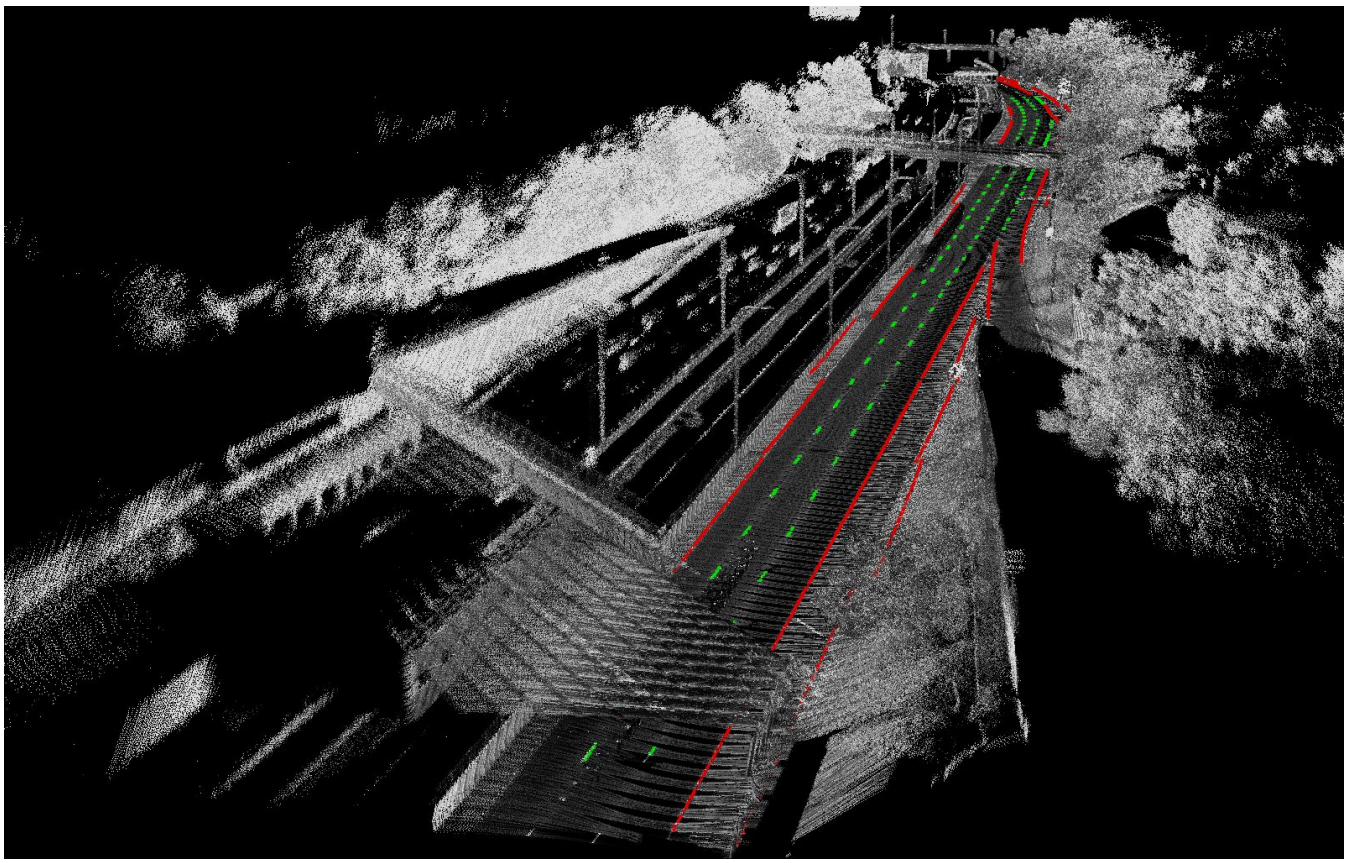

Figure 12: Extracted 3-D lane markings