2017 IEEE Intelligent Vehicles Symposium (IV)
June 11-14, 2017, Redondo Beach, CA, USA

Roadway Feature Mapping from Point Cloud Data: A Graph-Based Clustering Approach

Mohammad Billah, Arash Maskooki, Farzana Rahman and Jay A. Farrell

Abstract— **Connected and automated vehicle applications are facilitated by Enhanced Digital Maps (EDMs) of the roadway environment. Due to the high numbers of roadway miles and signalized intersections, there is significant research interest in the automatic extraction of such maps from georectified LiDAR data. Most existing methods convert the LiDAR point cloud to a set of images for feature extraction and mapping. This rasterization step loses information that could be retained if new methods were developed that work directly on the LiDAR point could for feature extraction and mapping, without rasterization.**

This article presents one such approach that operates on a road surface point cloud, processing small patches at a time using a locally adaptive version of Otsu's method to discard low intensity reflections while retaining reflections from roadway markings. The main new aspect of the approach is a graph-based clustering algorithm implemented directly on the point cloud. A cluster growing method is used to group similar road markings into the same group to enable detection of the stop bars and lane edges. Finally, a SAE-J2735 map message is created from the extracted roadway features.

I. INTRODUCTION

Intelligent driver assistance systems such as Advanced Driver Assistance Systems (ADAS) and Connected Vehicles (CV) benefit from Enhanced Digital Maps (EDMs) containing roadway features (e.g., lane edges and stop bars). Application performance will be enhanced as EDM accuracy and number of stored features both improve. Currently, EDM accuracy is at the 1-10 m level, with future accuracy desired to be at the decimeter level. Enhancing the number, type, automation and accuracy of EDM's are all active research topics.

Recently, there has been a rising interest in mobile terrestrial laser scanning (MTLS) [1]–[3], which has been adopted by both the commercial and academic sectors [4]. MTLS approaches capture dense, high precision point clouds in short periods of time. The MTLS including a Global Navigation Satellite System (GNSS) receiver and Inertial Measurement Unit (IMU) for high precision platform positioning [5], [6] along with a LiDAR, and possibly cameras, are mounted on a vehicle and driven through the roadway network.

There has been significant recent research on extracting roadway features from point cloud data, e.g., [7]–[11] . A common step in these approaches is the conversion of the point cloud to a 2D image. In the point cloud, each point is stored at a high accuracy determined by the georectification process (centimeter), LiDAR measurement accuracy (millimeter), and file storage precision (sub millimeter); however, the number of points and hence the file size is very large. Many of the points may be irrelevant to the roadway map. Conversion of selected relevant portions of the point cloud to images focuses the feature extraction process on the roadway and allows utilization of well understood image processing techniques for feature extraction. There are inherent tradeoffs when converting a 3D point cloud to a 2D image. Decreasing the pixel size increases the required amount of memory and processing power; however, increasing the pixel size (i.e. decreased resolution) reduces the ability of the extraction algorithms to effectively resolve features and decreases the georectified accuracy of the extracted features. Also, when more than one point cloud element maps to a pixel, then the question arises as to which value to assign to that pixel (e.g., min, max, mean, median). The selection of this one pixel value from the multiplicity of point cloud values discards valuable information for feature extraction. Yu et. al. [12] extracted and classified road markings using multi-segment thresholding, Euclidean distance based clustering and a deep learning approach. However, it is unclear how the algorithm parameters affect its performance.

This paper presents a new method to extract roadway features directly from road surface point clouds using graph-based clustering techniques. The roadway features considered in this paper are stop bars and lane edges. Additionally, this paper discusses EDM feature storage in the SAE-J2735 map message format [13].

II. PROBLEM STATEMENT

Our goal is to extract features painted on the road surface. We assume a preprocessor has used a test $T_S(p_i)$ to extract a road surface point cloud $\mathbb{S}$ which is a subset of the overall point cloud dataset $\mathbb{P}$:

$$\mathbb{S} = \{p_i \in \mathbb{P} \mid T_S(p_i) = 1,\, i = 1, \cdots, N\}. \qquad (1)$$

The preprocessor has also extracted the road edge points $\mathbb{Q}$. Due to space limitations, this preprocessor is not discussed herein, but it is straightforward [14]. Examples of $\mathbb{P}$ and $\mathbb{S}$ are shown in Fig. 1. Each $p_i = (x_i, y_i, z_i, I_i)$, where (x_i, y_i, z_i) are coordinates of point p_i are in world frame. The laser intensity reflected from p_i is I_i. The number of points in $\mathbb{S}$ is N. A dot notation is used to refer to the components of p_i (e.g. we refer to x_i component of p_i as $p_i.x$).

The road surface point cloud $\mathbb{S}$ can be conceptualized as two subsets: features points $\mathbb{F}$ and non-feature points $\mathbb{O}$ such

*This work was supported by a grant from the Cooperative Transportation Systems Pooled Fund Study subaward number GS11191-147223.

Billah and Rahman are with the department of Electrical and Computer Engineering, University of California, Riverside, 92521, U.S.A. {mbillah, frimi}@ece.ucr.edu

Farrell is faculty of Electrical and Computer Engineering, University of California, Riverside, 92521, U.S.A. farrell@ece.ucr.edu

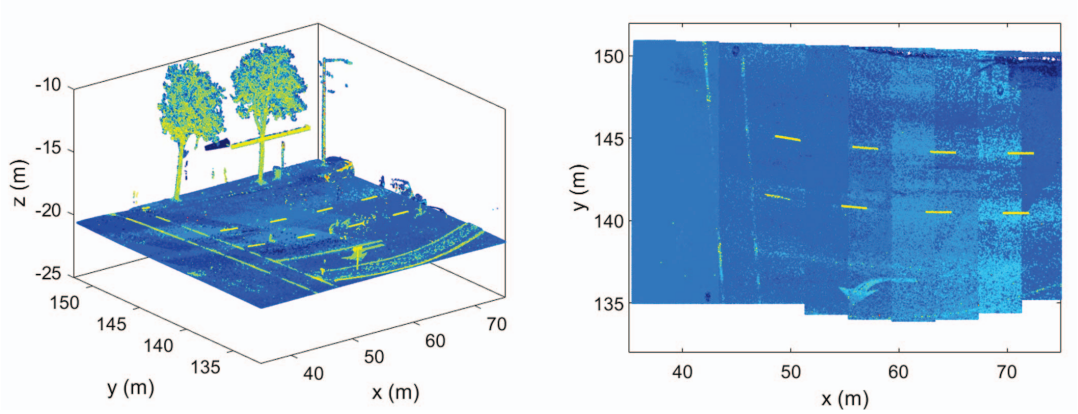

Fig. 1. An example point cloud $\mathbb{P}$ (left) and road surface $\mathbb{S}$ extracted from $\mathbb{P}$ (right).

that $\mathbb{S} = \mathbb{F} \cup \mathbb{O}$ and $\mathbb{F} \cap \mathbb{O} = \emptyset$. We assume that $\mathbb{F}$ consists of M features: $\mathbb{F} = \bigcup_{m=1}^{M} f_m$ where feature $f_m \subset \mathbb{S}$ from a single feature. Features correspond to markings painted on the road surface with high reflectance paint, so that their reflected intensity I should be higher than the pavement background. Each feature f_m has minimum and maximum dimensions and nominal separation from other features as defined in [15]; therefore, point reflections from each feature are expected to be clustered closely together with similar high intensity, and separated from points reflected from other features by regions of relatively low intensity.

III. GRAPH BASED CLUSTERING APPROACH

The feature map is extracted from $\mathbb{S}$ in four steps:

A. Locally Adaptive Otsu's Method: The goal of this step is to reduce the cardinality of the set $\mathbb{S}$ by removing unwanted points $\mathbb{O}$ while preserving feature points $\mathbb{F}$.

B. Clustering: The goal of this step is to remove the remaining points of $\mathbb{O}$ while points from $\mathbb{F}$ are assigned to distinct clusters.

C. Feature Extraction and Classification: Clusters belonging to each roadway feature are merged and classified.

D. Map Message Extraction: An SAE-J2735 map message with feature descriptors is output for $\mathbb{S}$.

A. Locally Adaptive Otsu's Method

For a rectangular patch of road surface with length L ($L \approx 10m$), the percentage of the surface that is painted is low. Since the point density required for feature extraction is high, and the LiDAR reflections are from the entire surface, the cardinality of the set $\mathbb{S}$ is very large, while the percentage of reflections from the features is low. Therefore, discarding low intensity points can greatly enhance computational efficiency.

Reflections intensity depends on angle-of-incidence, distance, surface material, and marking quality [16]. Therefore, a global intensity-threshold value does not effectively eliminate unwanted surface points while retaining feature points. Therefore, a locally adaptive method is implemented.

The first step uses a loose global threshold t to remove many (e.g., 80%) of the low intensity points while retaining

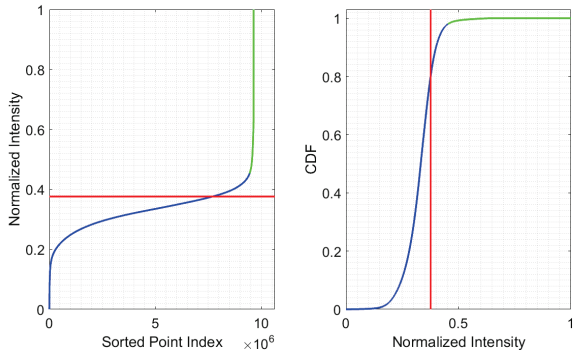

Fig. 2. Sorted normalized intensity of $\mathbb{S}$ (left). Intensity CDF (right). Blue dots are ultimately classified into $\mathbb{O}$. Green dots into $\mathbb{F}$.

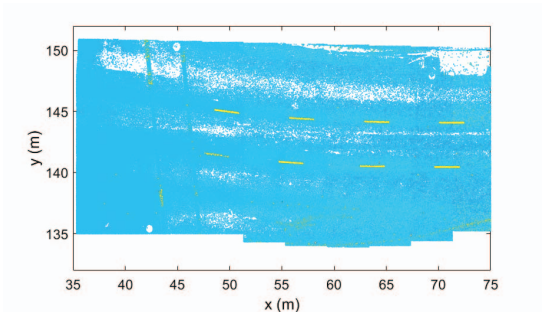

Fig. 3. Road surface point cloud $\mathbb{S}'$ after removing 80% of low intensity points using a global threshold.

feature points with high probability:

$$\mathbb{S}' = \{p_i \in \mathbb{S} \mid p_i.I > t\}. \quad (2)$$

Increasing t eliminates more elements of $\mathbb{O}$ while increasing the probability of losing elements in $\mathbb{F}$. Fig. 2 shows an example intensity cumulative distribution function. Fig. 3 shows an example $\mathbb{S}'$.

Since surface material and marking quality vary locally, the road is split into ℓ smaller segments $\mathbb{R}'_k$ such that $\mathbb{S}' = \cup_{k=1}^{\ell} \mathbb{R}'_k$. Each $\mathbb{R}'_k$ has side length L small enough so that the road surface and marking quality are assumed to be constant.

Otsu's method [17] is a computationally efficient tech-

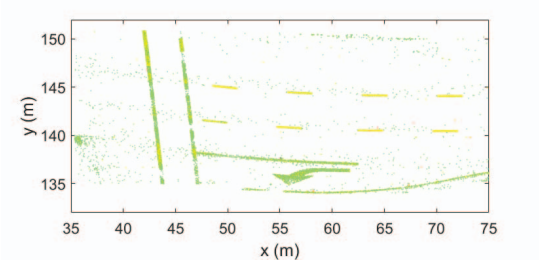

Fig. 4. Road surface point cloud after locally adaptive Otsu's method.

nique to divide each set $\mathbb{R}'_k$ into two subsets based on local intensity thresholding. It selects a threshold t_k to minimize the intra-subset intensity variance while maximizing the intensity separation between the two subsets. The resulting point cloud after locally applying Otsu's method is

$$\mathbb{S}'_k = \{p_i \in \mathbb{R}'_k \mid p_i.I > t_k\}. \tag{3}$$

An example is shown in Fig. 4. Otsu's method assumes that each $\mathbb{R}'_k$ has a bimodal density. To satisfy this assumption, each $\mathbb{R}'_k$ must be large enough to ensure that it contains reflections from both lane markings and unmarked road surface. Separation specifications for road surface features are defined in standard references [15].

Processing all segments using Otsu's threshold yields,

$$\mathbb{S}'' = \bigcup_{k=1}^{\ell} \mathbb{S}'_k \tag{4}$$

B. Graph Based Clustering

This section describes an approach that creates a graph based on distance and intensity similarity. This graph is then used to build clusters intended to correspond to features. This process works without rasterization.

Let $G = (\mathbb{V}, \mathbb{E})$ be a graph with vertices

$$\mathbb{V} = \{v_i \in \mathbb{S}'_k | i = 1, \cdots, K\} \tag{5}$$

For each vertex $v_i \in \mathbb{V}$ define a neighborhood

$$\mathbb{N}_i = \{v_j \in \mathbb{V} \mid d(v_i, v_j) < \epsilon, j = 1, \cdots, Q\} \tag{6}$$

where $d(v_i, v_j)$ is the Euclidean distance and ϵ is the neighborhood radius. The radius ϵ is a user-defined parameter that should correspond to the minimum feature size [15].

The set of weighted edges is

$$\mathbb{E} = \left\{(e_{ij}, w_{ij}) \mid i = 1, \cdots, K;\ j = 1, \cdots, \bar{K}\right\} \tag{7}$$

where e_{ij} is the edge connecting $v_i, v_j \in \mathbb{V}$. Weighted edges will only be defined between a node v_i and its $\bar{K}$ most similar vertices in neighborhood $\mathbb{N}_i$. The weight of edge e_{ij} is

$$w_{ij} = |v_i.I - v_j.I| \tag{8}$$

This edge weight is a measure of intensity dissimilarity with higher values indicating greater dissimilarity. Thus there is both a distance and a similarity constraint imposed for an edge to exist. The distance constraint defining neighborhood

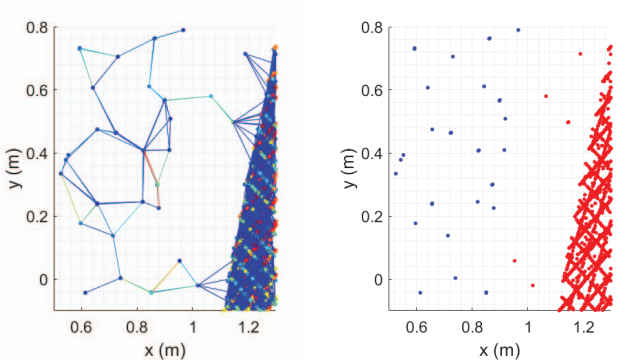

Fig. 5. (Left) Nodes and edges after creating a graph for a small patch of point cloud. The color of the nodes and the lines represent node intensity and edge weight. The colormap is from blue to red, blue being the lowest and red the highest. (Right) After segmentation, the red nodes represent points in $\mathbb{F}$ and the blue nodes represents points in $\mathbb{O}$.

$\mathbb{N}_i$ limits the computation required to find the most similar points and matches the Euclidean distance between connected nodes to the minimum feature size. The parameter $\bar{K}$ determines the graph sparsity, which affects computational complexity.

The next step is to create and merge clusters using the approach suggested in [18]. A cluster C_i is the set of vertices of a connected sub-graph of G. At the beginning of clustering, each vertex is assigned to its own cluster.

Let (C_i, E_{C_i}) be a connected subgraph of G where $C_i \subset \mathbb{V}$ and $E_{C_i} \subset \mathbb{E}$ containing the edges connecting the vertices in C_i. The Minimum Spanning Tree of (C_i, E_{C_i}), denoted $MST(C_i, E_{C_i})$, is a subset of the edges in E_{C_i} that connects all the vertices of C_i while having the minimum possible total edge weight. The internal dissimilarity of C_i, denoted $\mu(C_i)$, is defined as the largest weight of $MST(C_i, E_{C_i})$:

$$\mu(C_i) = \max_{e \in MST(C_i, E_{C_i})} w_{ij}$$

For a cluster with a single node, the cluster dissimilarity is zero. The dissimilarity between two connected clusters C_1 and C_2 is defined as:

$$\mu(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2} w_{ij} \tag{9}$$

where e_{ij} is an edge connecting the two clusters. If two clusters are disconnected, then $\mu(C_1, C_2) = \infty$.

The clusters are merged if their dissimilarity is smaller than the minimum internal dissimilarity $\lambda(C_1, C_2)$:

$$\lambda(C_1, C_2) = \min\left(\mu(C_1) + \tau(C_1), \mu(C_2) + \tau(C_2)\right)$$

where $\tau(C) = \frac{\kappa}{|\mathbb{C}|}$, $\kappa > 0$ affects the final size of the clusters and $|\mathbb{C}|$ denotes the cardinality of C. The edge weights are sorted smallest to largest and processed in that order. After all the edges are tested, the segmentation is complete. Fig. 5 show an example graph before and after cluster merging.

C. Feature Extraction and Classification

After cluster merging, clusters with small cardinality (less than 100 points) are discarded. Then roadway features (e.g. lane dividers and stop-bar) are extracted from the remaining clusters using the steps described below.

1) Splitting Large Nonlinear Clusters: The clustering algorithm may put the points from the stop-bar and nearby lane dividers in one cluster as the road markings are often close together. Therefore, the points in these large clusters may not form a line. Such clusters are divided into multiple linear clusters using RANSAC [19]. Next, RANSAC is applied to extract a line (with high cardinality) from clusters that are wider than the maximum feature width. The points that belong to this line are removed from the cluster. This method is repeated until the number of remaining points is sufficiently small and they are discarded. The points extracted in each iteration are assigned to a new cluster. At this point, all remaining clusters contain points that fall along lines. Each linear cluster is associated with a length, width and orientation that are defined as the length of the major and minor axes and angle of the major axis with the x-axis for the 0.90 confidence ellipse of the cluster points. Any cluster further that $60m$ from the intersection center is removed.

2) Stop-bar Detection: All the clusters that are oriented approximately perpendicular to the direction of the MTLS travel are combined into one cluster C_S. For various reasons, different segments of the stop-bar may have been detected as distinct clusters that are now all in C_S. RANSAC is applied to C_S to extract lines. For each detected line, points that belong to that line are removed from C_S and saved. The process is repeated. Each of these iterations detects one potential stop-bar. Ingresses to intersections that have a crosswalk will have a second line parallel to the stop-bar and closer to the intersection center. If only one line is detected then the detected line is considered as the stop-bar for that ingress. If more than two potential stop-bars are detected, a warning message is generated. When multiple lines are detected, the line that is closest to the center would be the crosswalk edge and the line that is second closest to the intersection center is declared to be the stop-bar. After the stop-bar is defined, cluster C_S is discarded. Any other clusters between the stop-bar and intersection center are removed as they do not contain any roadway features.

3) Discarding Unwanted Large Clusters: The road markings required for the map message have certain length and width [15]. Clusters that have width $> w$ are removed. This gets rid of the clusters containing right and left turn arrows and other wide clusters containing unwanted points.

4) Lane Edge Point Addition and Cluster Merging: The goal of this step is to merge the clusters for each lane divider into a distinct group. First, the edge points $\mathbb{Q}$ are added to the set as a cluster. Then all the remaining clusters are processed as follows. First, one cluster C_i is selected - perhaps the top-left most cluster. The selected cluster is assigned a group number g_1. Then, the closest ungrouped cluster C_j in the direction of travel that is within a distance r_g is merged into cluster g_1.This process is repeated until no additional clusters can be merged into that cluster. Then the next ungrouped cluster is assigned to a new group g_2. The above merging process is repeated for g_2. This process is repeated until all the clusters all grouped.

D. Map Message Extraction

After the clusters are grouped, map features are extracted. Each group of the clusters represents one lane edge, so the lane edges are found by extrapolating the points of each group so that they intersect with the stop-bar. Then the center nodes of each lane is calculated using the lane edge points such that the first node is on the stop-bar. These nodes define the SAE-J2735 map message.

IV. EXPERIMENTAL RESULTS

Fig. 1 shows a sample point cloud $\mathbb{P}$ and the road surface $\mathbb{S}$ extracted from the point cloud.

In this example, $\mathbb{S}$ contains around 1 million points. The majority of these points are low intensity surface points carrying no valuable information. The intensity of all the points and the Cumulative Distribution Function (CDF) of the intensities are shown in Fig. 2. The left figure shows a

Algorithm 1 Clustering

Parameters:
$\bar{K}$ = 4 ▷ Number of connected neighboring points
ϵ = 0.15 m ▷ Neighborhood radius
κ = 300 ▷ Parameter of segmentation algorithm

Input:
$\mathbb{S}'_k$ ▷ Point cloud after locally adaptive Otsu's

procedure CREATE_GRAPH($\mathbb{S}'_k$)
  $V_n = \mathbb{S}'_k$ ▷ Vertices
  for all $v_i \in V_n$ **do**
    for all $v_j \in V_n \ni d(v_i, v_j) < \epsilon$ **do**
      calculate w_{ij} using (8)
    end for
    for $\bar{K}$ smallest w_{ij} **do**
      $E(i,j) = w_{ij}$
    end for
    remove v_i from V_n
  end for
  return $G = (\mathbb{V}, \mathbb{E})$
end procedure

procedure CLUSTER_GRAPH(G)
  G = CREATE_GRAPH($\mathbb{S}'_k$)
  sort edges $e_{ij} \in \mathbb{E}$ smallest to largest
  assign each $v_i \in \mathbb{V}$ to its own cluster C_i
  for all e_{ij} in sorted $\mathbb{E}$ **do**
    calculate $\mu(C_i, C_j)$ using (9)
    calculate $\lambda(C_i, C_j)$ using (10)
    if $\mu(C_i, C_j) < \lambda(C_i, C_j)$ and $C_i \neq C_j$ **then**:
      merge C_i and C_j
    end if
  end for
end procedure

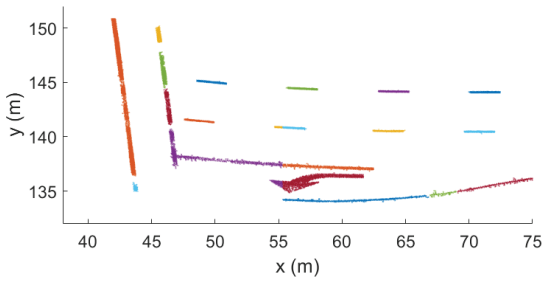

Fig. 6. Road surface point cloud after clustering.

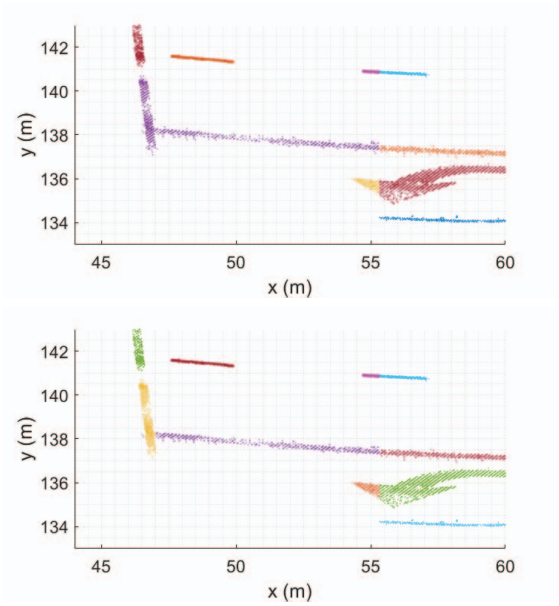

Fig. 7. (Top) Shows a large cluster before splitting in purple color. (Bottom) Shows two clusters with perpendicular orientation after splitting the large cluster.

red line indicating the intensity threshold defined in eqn. (2). From the intensity CDF shown on the right, less than 20% of the road surface point cloud is above that intensity. This 20% with the highest intensity is kept. The rest are discarded.

Fig. 3 shows the point cloud after the global threshold. As seen in Fig. 3, the features are still surrounded by unwanted surface points although the size of the point cloud $\mathbb{S}'$ is reduced by 80%. In comparison with the image on the right in Fig. 1, Fig. 3 appears more pale due to its smaller cardinality (e.g., it contains many white pixels). Next $\mathbb{S}'$ is split into the sets $\mathbb{R}'_k$; Otsu's method is applied in each segment; then the segments are merged to form $\mathbb{S}''$ as defined in eqn. (4). Comparison of Fig. 3 to Fig. 4 shows that local adaptive application of Otsu's method (with $L = 8m$) has improved clarity and removed the vast majority of the points reflected from the road surface, but did not remove the points corresponding to the lane markings.

The next step is to perform the clustering approach from Section III-B to $\mathbb{S}''$. The approach is summarized in Algorithm 1 with parameter values. For computational efficiency, $\mathbb{S}''$ is processed in slices with length approximately $20m$ to keep the number of points below 50000. Fig. 5 shows a small

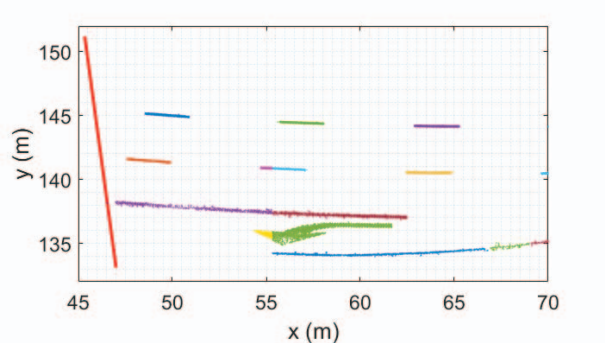

Fig. 8. After detecting stop-bar and discarding clusters that are large and outside region of interest.

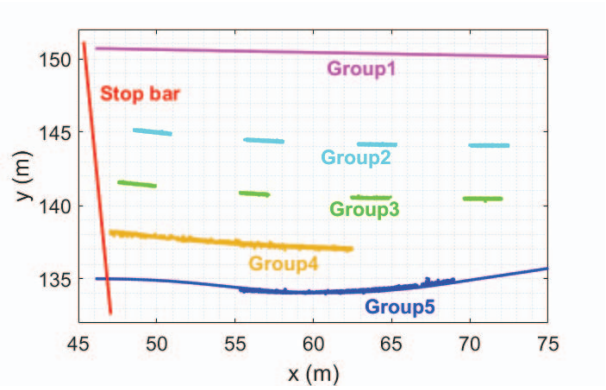

Fig. 9. After grouping the clusters.

portion of the resulting graph on the left and small patch of the clustered point cloud on the right. Fig. 6 shows the results of applying clustering to S''. Colors are assigned to individual clusters with cardinality larger than 100. Smaller clusters are discarded and not plotted. Feature points ($\mathbb{F}$) are separated from background points ($\mathbb{O}$) well.

Next features are extracted using the method described in Section III-C. Fig. 7 shows a large nonlinear cluster (purple) before splitting and two linear clusters (purple and yellow) afterwards. Next, the stop-bar is detected and all the clusters between the stop-bar and intersection center are discarded as they contain no additional valuable information. Fig. 8 shows the detected stop-bar as a red line along with the remaining portions of the point cloud.

After the stop-bar is detected and the large unwanted clusters are discarded (those that have width larger than the maximum feature width, i.e., $w = 0.5m$), the edge points are added to the set as a cluster. Then the clusters are grouped using the method described in Section III-C. Fig. 9 shows the result after grouping the clusters. Each color represents a different group.

After the stop-bar and the lane edges are detected, the SAE-J2735 nodes and message are computed and output. The SAE-J2735 map message includes nodes for each lane centerline that start at the stop-bar. The lane edges are calculated by extrapolating the grouped clusters. Then the lane center nodes are calculated midway between the lane

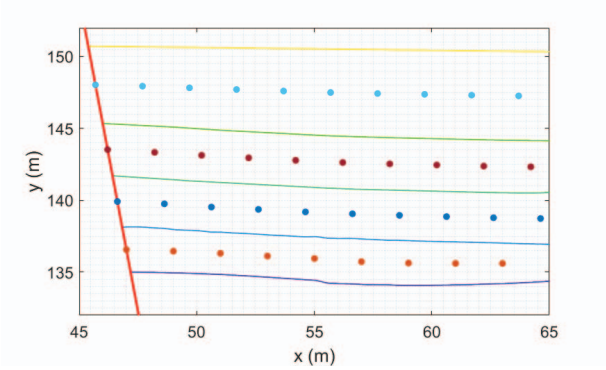

Fig. 10. Estimated J2735 nodes.

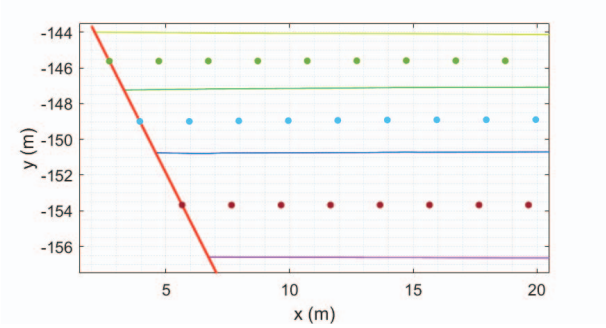

Fig. 11. Additional example of estimated J2735 nodes.

edges. Fig. 10 shows the SAE-J2735 nodes as dotes and the stop-bar and the calculated lane edges as solid lines. Fig. 11 and Fig. 12 show additional example outputs after applying the proposed algorithm.

V. CONCLUSION

This paper proposed a roadway feature extraction algorithm using LiDAR road surface data, which includes (1) a locally adaptive intensity thresholding method, (2) a graph generation and clustering algorithm, (3) a feature extraction algorithm, and (4) a J2735 map message output function. A main advantage of the algorithm is that it extracts features directly from the point cloud without rasterization.

Experimental results are included to demonstrate the approach. The discussion includes explanations for the physical meaning of the parameters and their effect on computational load and algorithm performance.

This method is effective for extracting roadway features (i.e., stop-bar and lane edges) from the large volume of LiDAR data. The process is automated and the parameter value depends on the roadway feature characteristics so they do not require any manual tuning.

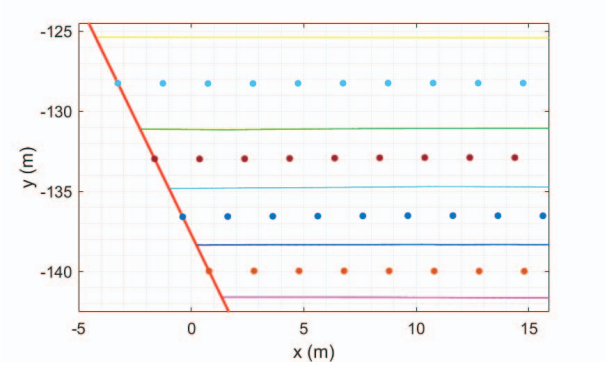

Fig. 12. Additional example of estimated J2735 nodes.

ACKNOWLEDGMENT

The authors would like to acknowledge Advanced Highway Maintenance & Construction Technology Research Center (AHMCT) for providing 3D point cloud data.

REFERENCES

[1] C. Darnel, "Using LiDAR to solve industry challenges," *Geo: Geoconnexion Int. Mag.*, vol. 11, no. 1, pp. 18–19, 2012.
[2] N. Haala, M. Peter, A. Cefalu, and J. Kremer, "Mobile LiDAR mapping for urban data capture," *14th Int. Conf. on Virtual Syst. and Multimedia*, pp. 95–100, 2008.
[3] V. Ussyshkin, "Mobile laser scanning technology for surveying application: from data collection to end-products," *FIG Working Week*, 2009.
[4] D. Barber, J. Mills, and S. Smith-Voysey, "Geometric validation of a ground-based mobile laser scanning system," *ISPRS J. of Photogramm. and Rem. Sens.*, vol. 63, no. 1, pp. 128–141, 2008.
[5] C. V. Tao and J. Li, *Advances in mobile mapping technology*. CRC Press, 2007, vol. 4.
[6] A. Vu, J. A. Farrell, and M. Barth, "Centimeter-accuracy smoothed vehicle trajectory estimation," *IEEE Intelli. Transp. Syst. Mag.*, vol. 5, no. 4, pp. 121–135, 2013.
[7] C. Crawford, "Minimising noise from LiDAR for contouring and slope analysis," Available at http://desktop.arcgis.com/en/arcmap/10.3/manage-data/las-dataset/lidar-solutions-minimizing-noise-from-lidar-for-contouring-and-slope-analysis.htm, 2009.
[8] B. Yang, L. Fang, Q. Li, and J. Li, "Automated extraction of road markings from mobile LiDAR point clouds," *Photogramm. Engineering & Rem. Sens.*, vol. 78, no. 4, pp. 331–338, 2012.
[9] H. Guan, J. Li, Y. Yu, C. Wang, M. Chapman, and B. Yang, "Using mobile laser scanning data for automated extraction of road markings," *ISPRS J. of Photogramm. and Rem. Sens.*, vol. 87, pp. 93–107, 2014.
[10] P. Kumar, C. P. McElhinney, P. Lewis, and T. McCarthy, "Automated road markings extraction from mobile laser scanning data," *Int. J. of Applied Earth Obs. and Geoinf.*, vol. 32, pp. 125–137, 2014.
[11] H. Guan, J. Li, Y. Yu, M. Chapman, and C. Wang, "Automated road information extraction from mobile laser scanning data," *IEEE T. on Intelli. Transp. Syst.*, vol. 16, no. 1, pp. 194–205, 2015.
[12] Y. Yu, J. Li, H. Guan, F. Jia, and C. Wang, "Learning hierarchical features for automated extraction of road markings from 3-d mobile LiDAR point clouds," *IEEE J. of Sel. Topics in Applied Earth Observations and Rem. Sens.*, vol. 8, no. 2, pp. 709–726, 2015.
[13] *Dedicated short range communications (DSRC) message set dictionary*, SAE International, Tech. Rep. J2735_200911.
[14] J. A. Farrell, M. Todd, and M. Barth, "Best practices for surveying and mapping roadways and intersections for connected vehicle applications," Available at http://escholarship.org/uc/item/4f88m75k, 2016.
[15] "Manual on uniform traffic control devices," Available at http://mutcd.fhwa.dot.gov/.
[16] T. Ogawa and K. Takagi, "Lane recognition using on-vehicle LiDAR," *IEEE Intelli. Vehicles Symp.*, pp. 540–545, 2006.
[17] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285-296, pp. 23–27, 1975.
[18] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. of Comp. Vis.*, vol. 59, no. 2, pp. 167–181, 2004.
[19] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Comm. of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.