Author manuscript, published in Proc. IEEE Int'l Conf. on Cognitive Infocommunications, pp. 515 - 520, Budapest, Hungary 2013

On board 3D Object Perception in Dynamic Urban Scenes

Attila Börcs, Balázs Nagy and Csaba Benedek

Distributed Events Analysis Research Laboratory,
Institute for Computer Science and Control of the Hungarian Academy of Sciences
H-1111, Kende utca 13-17 Budapest, Hungary, E-mail: firstname.lastname@sztaki.mta.hu

***Abstract*—In urban environments, object recognition and road monitoring are key issues for driving assistance systems or autonomous vehicles. This paper presents a LIDAR-based perception system which provides reliable detection of 3D urban objects from point cloud sequences of a Velodyne HDL-64E terrestrial LIDAR scanner installed on a moving platform. As for the output of the system, we perform real-time localization and identification of typical urban objects, such as traffic signs, vehicles or crosswalks. In contrast to most existing works, the proposed algorithm does not use hand-labeled training datasets to perform object classification. Experimental results are carried out on real LIDAR measurements in the streets of Budapest, Hungary.**

I. Introduction

The reliable perception of the surrounding environment is an important task in outdoor robotics. Robustly detecting and identifying various urban objects are key problems for autonomous driving, and driving assistance systems. Future mobile vision systems promise a number of benefits for the society, including prevention of road accidents by constantly monitoring the surrounding vehicles or ensuring more comfort and convenience for the drivers. Laser range sensors are particularly interesting for these tasks since in contrast to conventional camera systems they are highly robust against illumination changes or weather conditions, and typically provide a larger field of view. Moreover, LIDAR mapping systems are able to rapidly acquire large-scale 3D point cloud data for real-time vision, with jointly providing accurate 3D geometrical information of the scene, and additional features about the reflection properties and compactness of the surfaces.

A number of approaches are available in literature for solving object recognition problems in point clouds of a 3D laser scanners. In [1], a framework has been proposed for object classification and tracking. The basic idea is to use an octree based Occupancy Grid representation to model the surrounding environment, and simple features e.g. length ratios of object bounding boxes for object classification. In that method three different object classes are considered: pedestrians, bicycles and vehicles.

In our case the observed environment consists of complex urban scenarios with many object types such as trees, poles, traffic signs, occluded wall regions, thus simple features may not be robust enough for efficient object classification, due to varying appearance of the considered objects' geometry throughout an entire city.

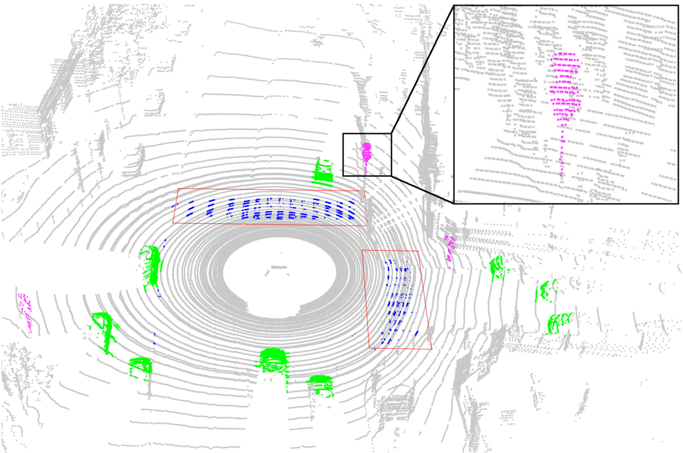

Fig. 1: The result of the proposed object detection algorithm. Recognized object classes denoted by the following color codes: green - *vehicle*, magenta - *traffic sign*, blue surrounded by a red rectangle - *crosswalk*.

A group of existing object classification techniques [2],[3] use robust features, based on shape and contextual descriptors. In [2], a set of clustering methods is presented for various types of 3D point clouds, including dense 3D data (e.g. Riegl scans) and sparse point sets (e.g. Velodyne scans). The authors of [3] propose a system for object recognition, which clusters nearby points to form a set of potential object locations in a hierarchical approach. Then, they segment points near the estimated locations into foreground and background sets with a graph-cut algorithm. Finally they build a feature vector for each point cluster and label the feature vectors using a classifier trained on a set of manually labeled objects. However, the above approaches do not perform in real time.

Object recognition tasks from unstructured point clouds are often performed via machine learning techniques with predefined training samples [4], [5], [6], [7]. In [5] a 3D object detection system was proposed for robots, using objects from Google's 3D Warehouse. They train the proposed system for performing navigation tasks in urban and indoor environments. First object detection is obtained by calculating various point

This work is connected to the i4D project funded by the internal R&D grant of MTA SZTAKI, and it was partially supported by the the Government of Hungary through a European Space Agency (ESA) Contract under the Plan for European Cooperating States (PECS). The third author also acknowledges the support of the Hungarian Research Fund (OTKA #101598)

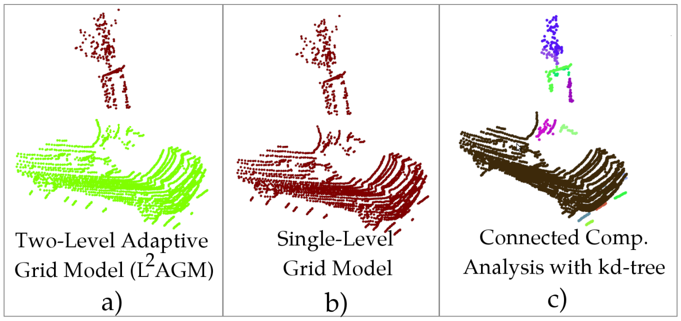
Two-Level Adaptive Grid Model (L2AGM)
a)
Single-Level Grid Model
b)
Connected Comp. Analysis with kd-tree
c)

Fig. 2: Comparison of three different object separation methods for a case of nearby objects (e.g. sapling and vehicle in the demonstrated scenario). On the left: the proposed L^2AGM grid model separates correctly the two objects. In the middle: the conventional single-level grid model width 8-connected neighbourhood merge them together, because merging criteria fails due to low grid resolution (used 60 cm cell size). On the right: kd-tree based connected component analysis often over-segments the objects due to inhomogeneous density of the point cloud.

cloud descriptors. Second, a ray casting algorithm was proposed to obtain additional features from the 3D Warehouse objects. Third the detection is performed in the descriptor space. However, difficulties in object extraction are not detailed here, and a hand-labeled training dataset is required in advance.

In this paper we address the problem of detecting and classifying different types of urban objects such as traffic signs, vehicles and crosswalks in real-time without any hand-labeled training datasets. The processed data comes from a terrestrial Rotating Multi-Beam (RMB) LIDAR scanner (Velodyne HDL-64E) which is able to provide 360° point streams with a frequency of 5-15 Hz. The main challenge regarding this task is that the appearance of an urban object in the RMB range data can drastically change as a function of distance from the sensor, while we must expect artifacts of self-occlusion or occlusion by other object, measurement noise, inhomogeneous point density and mirroring effects. Manual evaluation is particularly unreliable, because human visual system is not accustomed to interpreting unorganized points sets [8]. Thus, automatic object detection and classification is a crucial need for dealing this problem.

In this work we particularly focus on challenging 3D scenarios, where nearby vehicles or traffic signs must be separated and identified. The key ideas of our approach are the following: 1) We propose a novel object segmentation method (called Two-Level Adaptive Grid Model - L^2AGM), which can robustly separate various 3D objects in terrestrial point clouds collected from urban environment, while maintains low computational complexity enabling real-time performance. 2) We present efficient features for object classification based on laser intensity responses and object geometry. Details of the proposed method are introduced in Sec. II, and experiments on the real data are provided in Sec. III.

II. Proposed Object Perception Framework

The proposed method consists of four main steps: *First*, the individual LIDAR point cloud scans are segmented into different semantic regions. *Second*, urban objects are separated with a novel Two-Level Adaptive Grid Model (L^2AGM). *Third*, features are extracted concerning both geometrical appearance and laser intensity responses of the objects. *Fourth*, street object are classified as vehicle, traffic sign or crosswalk using efficient feature combinations.

The segmentation process assigns to each measured point a class label from the following set: (i) *clutter* (ii) *ground*, (iii) *tall structure objects* (walls, roofs, lamps posts, traffic lights etc.), (iv) *short street objects* (vehicles, pedestrians etc.). In this section, we address the discrimination of these four classes.

A. Point Cloud Segmentation

In our system, point cloud segmentation is achieved by a grid based approach [9],[10]. In the literature various robust approaches are proposed for planar *ground* modeling such as RANSAC. However in terrestrial point clouds often significant elevation differences (up to a few meters) can be experienced due to slope between the opposite sides of the observed roads and squares. In these cases, planar ground estimation leads inaccurate ground segmentation, and yields significant errors in the extracted object shapes, e.g. bottom parts can be cut off, or the objects may drift over the ground. In contrast to planar fitting based solutions, we apply a locally adaptive terrain modeling approach, detailed as follows.

We fit a regular 2D grid S with W_S rectangle side length onto the $P_{z=0}$ plane, where $s \in S$ denotes a single cell. We used a W_S value between 50cm and 80cm. Smaller grid size is not viable due to the resolution; smaller cells would not have enough points in them to calculate reliable statistical information. On the other hand, larger cell size can result in larger number of falsely classified points, since within a large cell, multiple objects can occur. Near-the-center grid cells may include hundreds of points, while the point density rapidly decreases as a function of the distance form the sensor. We assign each $p \in \mathcal{P}$ point of the point cloud to the corresponding cell s_p, which contains the projection of p to $P_{z=0}$. Let us denote by $\mathcal{P}_s = \{p \in \mathcal{P} : s = s_p\}$ the point set projected to cell s. $z_{\max}(s)$, $z_{\min}(s)$ and $\hat{z}(s)$ are the maximum, minimum and average of the elevation values within $\mathcal{P}_s$ and $\mathcal{L}(c) \in \mathcal{L} = \{l_{(clutter)}, l_{(roof)}, l_{(ground)}, l_{(tall\,obj.)}, l_{(short\,obj.)}\}$ denotes cell class.

We use point height information for assigning each cell to the corresponding cell class. Before that, we detect and remove cells that belong to clutter regions, thus we will not visit these cells later and save processing time. We consider a cell cardinality criteria, which classifies any cell to *clutter* $\mathcal{L}(c) = l_{(clutter)}$, which contains less points than a cardinality threshold (typically 4-8 points). After clutter removal, ground detection is achieved by an elevation difference criteria within each cell. All the points in a cell are classified as *ground* $\mathcal{L}(c) = l_{(ground)}$, if the difference of the minimal and maximal point elevations in the cell is smaller than a threshold (used 25cm), moreover the average of the elevations in neighboring cells does not exceeds an allowed height range. A cell belongs to the class of *tall structure objects* (e.g. traffic signs, building walls, lamp post etc.), denoted by $\mathcal{L}(c) = l_{(tall\, obj.)}$, if either the maximal point height within the cell is larger than a predefined value (used 140cm), or the observed point height difference is larger than a threshold (used 310cm). The rest of the points in the cloud are assigned to class *short street object* $\mathcal{L}(c) = l_{(short\, obj.)}$ belonging to vehicles, pedestrians, mail boxes, billboards etc.

B. Object Detection

In this section we propose a method for automatic separation of 3D blobs from LIDAR point cloud sequences, providing accurate detection of urban objects. The main bottlenecks of object separation techniques are the following: First, efficient point neighbourhood modeling is necessary. The main challenge here is to obtain point neighbours as fast as possible. Particularly in robot perception systems it is a crucial criteria to obtain this step in real-time. Second, robust merging criteria is needed for merging a certain 3D point and its neighbours into the same blob as long as they belong to the same object.

Although various established techniques do exists, such as grid based 4-connection neighbourhood [11] and kd-tree based approach [12], these methods often give us insufficient results on raw Velodyne LIDAR point clouds for two reasons: 1) As demonstrated in Fig. 2 (b) simple grid based object segmentation methods with small grid resolution (i.e. large cell size) are capable of working in real time, but often fail according to object merging criteria, if we try to separate nearby objects e.g. vehicles in crowded parking lot or pedestrians who pass by each other. On the other hand, by increasing the gird resolution (i.e. decreasing the cell size), more accurate object segmentation can be achieved, however this step slows down the algorithm, and may falsely cut off regions of a given object. 2) Due to the strongly inhomogeneous density of the Velodyne LIDAR point clouds, kd-tree based solutions unnecessarily over partition the 3D space and split the desired objects, moreover this approach does not work in real-time, and the optimal parameters strongly depend on the object geometry (see Fig. 2 (c)).

Our key idea is to create an extended grid based approach (Fig. 2 (a)) called Two-Level Adaptive Grid Model (L^2AGM). In contrast to a simple grid based segmentation, this method uses a coarse cell grid for fast detection, and a dense cell grid to ensure robust separation of nearby objects. This two-level grid structure allows us to detect objects in real time, as well as prevent to over partition the desired object. Moreover in contrary to the kd-tree based approaches, the optimal parameter setting of the proposed method only depends on the point cloud density, and does not influenced by the geometry of the given object. The model construction consists of two main steps:

First using the coarse cell grid (60 cm resolution), and the initial segmentation from section II-A, we consider the *short object* and *long object* cell classes as *foreground*, while we label the other classes as *background*. Our intention is to find connected 3D blobs within the foreground regions, by merging the first-level grid cells together so that they represent different street objects. We use $\psi(s, s_r) = |Z_{max}(s) - Z_{max}(s_r)|$ merging indicator, which measures the local elevation difference between cell s and its neighbouring cell s_r, where $r \in N_s^\nu$ and N_s^ν the $\nu \times \nu$ neighbourhood of s (used 3×3). If the ψ indicator is smaller than a predefined value, then s and s_r belongs to the same 3D object.

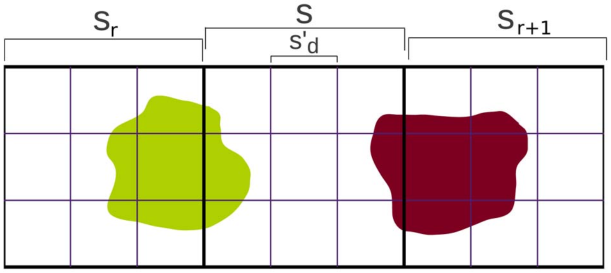

Fig. 3: Nearby objects not separable on the first-level cells s, because the center cell s contains both objects, and s_r;s_{r+1} neighbour cells contains each object independently due to resolution issues. However examining the point density of the sesond-level cells s'_d these objects can be separated from each other.

Second the merging criteria in the *first* step often yields insufficient results for nearby objects, because the grid resolution (used W_s=60cm cell side length) is not detailed enough to separate them (see Fig. 3). To save computational time, we keep the coarse grid resolution from the initial classification, but we handle this issue by creating a second-level grid with a higher resolution. This denser grid partitioning step is only executed in *tall* and *short object* regions of the 3D scenario, thus when we go trough the second-level grid, the computational time does not increase significantly. The cell s is subdivided into smaller cells $s'_d | d \in 1, 2 \cdots, 1/\xi^2$, with cell side length $W_{s'_d} = \xi W_s$, where $0 < \xi \leq 1/2$ is a scaling factor (used $1/3$). Thereafter, we prescribe a point density based merging criterion on the second-level grid by measuring the point cardinality in each cell s'_d. We expect several points within each sub-cell of a given object, and a strongly varying point density (i.e. varying point cardinality in cells $s'_d | d \in 1, 2 \cdots, 1/\xi^2$) in regions splitting various objects. If this high-low-high density change does not exit for a given super cell s, and the elevation difference based merging criterion is also fulfilled, then we finalize the merging process by connecting the cell s and its neighbouring cell s_r. Otherwise, we do not connect the two cells together, but we subdivide the center cell s between its pairwise neigbouring cell s_r and s_{r+1}, which are perpendicular to the direction of the density gradient (see Fig. 4).

C. Feature Selection and Object Recognition

Our method can distinguish three object classes: *traffic sign*, *vehicle* and *crosswalk*. The class *traffic sign* represents objects which have a notably low spatial variance according to depth and width, and a high spatial variance considering the object height. We expect points with high intensity responses in the upper part of the object, and low intensity for the bottom part, since road signs especially give us high intensity response caused by strongly reflective surfaces made from shiny materials. Objects are classified as *vehicles*, if they have a large spatial variance in width and depth, and small variance in height. We assign *crosswalk* class to objects, which are located

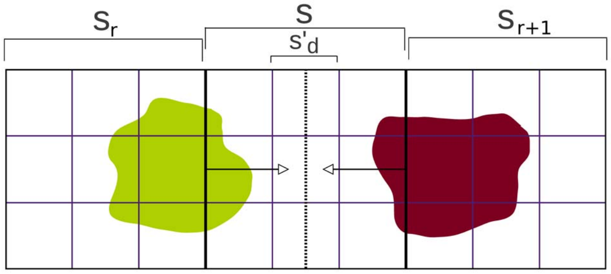

Fig. 4: If the high-low-high density change is exists in center cell s, than we adjust the first-level grid structure to the separated objects by subdivide s between its pairwise neighbouring cell s_r and s_{r+1}, with the perpendicular direction of the density gradient (denoted by dotted line). Note that, this step also works with left-right and bottom-up pairwise cell neighbourhood depending on the direction of the density gradient.

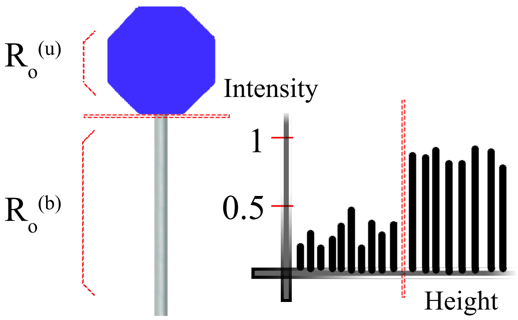

Fig. 5: f_i measures the mean intensity values of different object regions as a function of elevation (height). The feature finds an elevation threshold based on local object geometry (denoted by red horizontal line), and measures the intensity in the upper range $\mathcal{R}_o^{(u)}$ and the lower range $\mathcal{R}_o^{(b)}$. We expect high intensity ratio in case of traffic signs, and low intensity ratio otherwise.

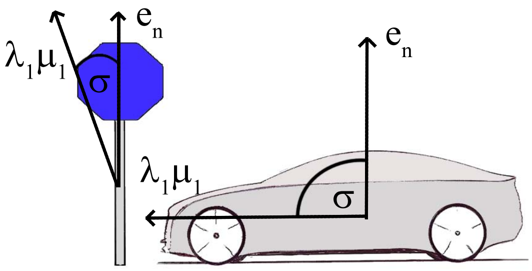

Fig. 6: f_{so} calculates the the orientation and the length of the main variance $\lambda_1\mu_1$ within the detected object by covariance analysis, and measure the angle different σ between $\lambda_1\mu_1$ and the up vector e_n. The angle σ is low in case of tall-elongated object, such as traffic sign or pedestrians, and high in case of vehicles.

on the ground, and give high intensity values on both directions regarding to the two main axes of the ground.

Let us denote by $\mathcal{R}_o$ the set of a 3D points, which are belongs to an object candidate. For each extracted object candidate, several geometric and laser intensity attributes are computed in order to distinguish the three object classes. The *Intensity ratio* fi, *Spatial Orientation* f_o and the *Point Cardinality* f_{pc} features calculated on the obtained 3D object candidates from section II-B, and the *Planar Intensity* f_{pi} derived from the ground region of the initial segmentation from section II-A.

- *Intensity ratio* f_i represents the ratio of the observed mean intensities in two different height ranges within an object. For traffic signs, we expect high intensities in upper height range, and low intensities in lower height range (see Fig. 5).

  $$f_i = \mathbf{1}\left(\frac{\frac{1}{|\mathcal{R}_o^{(u)}|}\sum_{p\in\mathcal{R}_o^{(u)}}\{i(p)\}}{\frac{1}{|\mathcal{R}_o^{(b)}|}\sum_{p\in\mathcal{R}_o^{(b)}}\{i(p)\}} > \sigma_i\right),$$

  where $\mathcal{R}_o^{(u)}$ and $\mathcal{R}_o^{(b)}$ denotes the set of object points which are higher or lower than an elevation threshold, $|\mathcal{R}_o^{(\cdot)}|$ is the cardinality of $\mathcal{R}_o^{(\cdot)}$, σ_i is the intensity ratio threshold, and $i(p)$ is the intensity value of the object point p.
- *Spatial orientation* f_{so} allows the distinction between *vehicles* and traffic signs (Fig. 6). This feature corresponds to the angle difference between the main orientation of the object and the up vector $v_{up} = (0,0,1)$. In order to obtain main orientation, we calculate the three eigenvalues $\lambda_1 > \lambda_2 > \lambda_3$ with the corresponding eigenvectors $\mu_1 > \mu_2 > \mu_3$ of the $\sum_{\mathcal{R}_o}$ covariance matrix. The μ_1 eigenvector shows the largest spatial variance according to the object geometry. We assume that the angle between μ_1 and v_{up} is large in case of vehicles, and small in case of tall-elongated objects (e.g. pedestrians or traffic signs):

  $$f_{so} = \mathbf{1}\left(\arccos\left(\frac{\lambda_1\mu_1}{\|\lambda_1\mu_1\| \cdot \|v_{up}\|}\right) < \sigma_{so}\right),$$

  where σ_{so} is an angle threshold between μ_1 and e_n.
- *Point cardinality* f_{pc} measures the size of the point cloud which belongs to the detected object:

  $$f_{pc} = \mathbf{1}\left(|\mathcal{R}_o| > \sigma_{pc}\right),$$

  where σ_{pc} is the cardinality threshold.
- *Planar intensity value* f_{pi} is dedicated to crosswalks. This feature corresponds to high intensity regions of the ground. We calculate the intensity histograms in the direction of the two main axes of the $z = 0$ plane, and find the mutual maximum range of the obtained intensity histograms (see Fig. 7). The response is logical false ($f_{pi} = 0$) in case of small road signs, and logical true ($f_{pi} = 1$) in case of crosswalks.

After feature extraction $\mathcal{F}_{vh}$, $\mathcal{F}_{ts}$, $\mathcal{F}_{cw}$ class data terms are derived for object descriptor combination. These three

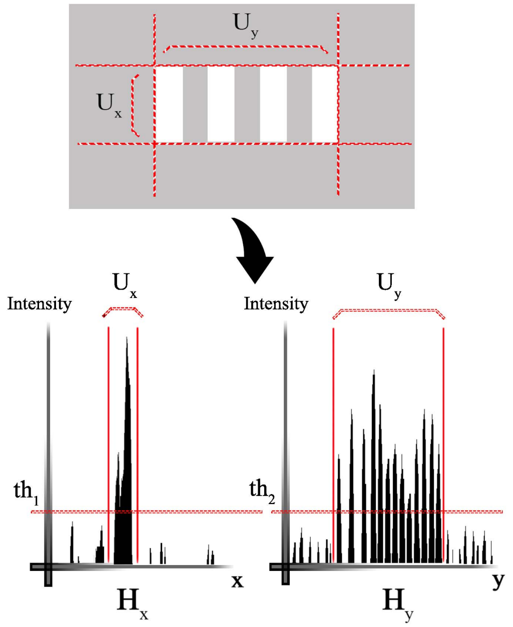

Fig. 7: The f_{pi} feature obtain the intensity histograms regarding to the two main axes of the ground. We can estimate the side lengths of the crosswalk by thresolding the intensity values of $\mathrm{H_x}$ and $\mathrm{H_y}$.

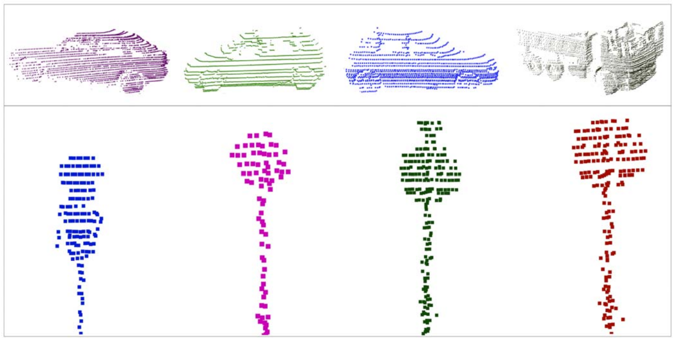

Fig. 8: Qualitative evaluation of the proposed object recognition algorithm. On the top row: Recognized objects with *vehicle* object class. On the bottom row: Objects which are belong to *traffic sign* object class.

data terms corresponds to *vehicle*, *traffic sign* and *crosswalk* classes, and assign corresponding object class to the feature combinations.

$$\mathcal{F}_{vh} = \mathbf{1}(((1 - f_{so}) \cdot f_{pc} \cdot (1 - f_i)) = 1)$$
$$\mathcal{F}_{ts} = \mathbf{1}((f_i \cdot (1 - f_{pc}) \cdot f_{so}) = 1)$$
$$\mathcal{F}_{cw} = \mathbf{1}(f_{pi} = 1)$$

Since the above descriptors ensure that for each $\mathcal{R}_o$ object candidate, at most one of the $\mathcal{F}_{vh}$, $\mathcal{F}_{ts}$ or $\mathcal{F}_{cw}$ features have the value "1" , we can classify the objects in a straightforward way: object R_o is vehicle if $\mathcal{F}_{vh} = 1$, traffic sign if $\mathcal{F}_{ts} = 1$ and crosswalk if $\mathcal{F}_{cw} = 1$. Otherwise, we do not assign any label to the object, marking it unrecognized.

III. Experiments

We have tested the proposed approach on real point cloud sequences obtained by a Velodyne HDL-64E laser scanner in the streets of Budapest, Hungary. Our system framework runs in real-time on standard CPU[1], with a processing time of around 33msec/frame, which is lower than the update frequency of the Velodyne HDL-64E. Some qualitative results are shown in Fig. 1 and 8 (best viewed in color), confirming the usability of our method.

IV. Conclusions

This paper has proposed a novel object detection method which is using two grid levels for robust separation of nearby objects. Thereafter we show an efficient object recognition algorithm using object geometry and laser intensity based feature combinations. We managed to separate 3D objects by a Two-Level Adaptive Grid Model (L^2AGM). The optimal parameters of the proposed method do not depend on the object's geometry. For robust separation of nearby object the proposed model can utilize two merging criteria on different grid levels, based on point elevation difference and point density changes. Moreover, in contrast to most existing works, the proposed object recognition step of our framework takes advantage of the laser intensity response of the Velodyne sensor, and does not use hand-labeled training datasets to perform object classification.

Our future work includes an extensive quantitative evaluation of the proposed framework dealing with several types of objects on different point cloud scenarios, and consider the possibility of fusing terrestrial 3D data with airborne or spaceborne RGB data (such as satellite or aerial images) for achieve more accurate object detection and recognition.

References

[1] A. Azim and O. Aycard, "Detection, classification and tracking of moving objects in a 3D environment.," in *IEEE Intelligent Vehicles Symposium (IV)*, Alcalá de Henares, Spain, 2012, pp. 802–807.

[2] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A. Frenkel, "On the segmentation of 3D Lidar point clouds," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, 2011, pp. 2798 –2805.

[3] A. Golovinskiy, V. G. Kim, and T. Funkhouser, "Shape-based recognition of 3D point clouds in urban environments," Kyoto, Japan, 2009.

[4] M. Samples and M. R. James, "Learning a real-time 3D point cloud obstacle discriminator via bootstrapping," in *Workshop on Robotics and Intelligent Transportation System*, Anchorage, Alaska, 2010.

[5] K. Lai and D. Fox, "Object recognition in 3D point clouds using web data and domain adaptation," *I. J. Robotic Res.*, vol. 29, no. 8, pp. 1019–1037, 2010.

[6] A. J. Quadros, J. Underwood, and B. Douillard, "An occlusion-aware feature for range images," in *IEEE International Conference on Robotics and Automation (ICRA)*, St. Paul, USA, 2012, pp. 4428–4435.

[7] C. Liang-Chia, H. Hoang Hong, N. Xuan-Loc, and W. Hsiao-Wen, "Novel 3-D object recognition methodology employing a curvature-based histogram," *I. J. Robotic Res.*, vol. 10, pp. 1019–1037, 2013.

[1] The test environment had an Intel Core i7 processor and 8 GB of DDR3 RAM.

[8] A. Velizhev, R. Shapovalov, and K. Schindler, "An implicit shape model for object detection in 3D point clouds," in *ISPRS Congress*, Melbourne, Australia, 2012.

[9] A. Börcs, O. Józsa, and C. Benedek, "Object extraction in urban environments from large-scale dynamic point cloud dataset," in *IEEE International Workshop on Content-Based Multimedia Indexing (CBMI)*, Veszprém, Hungary, 2013.

[10] O. Józsa, A. Börcs, and C. Benedek, "Towards 4D virtual city reconstruction from Lidar point cloud sequences," in *ISPRS Workshop on 3D Virtual City Modeling*, vol. II-3/W1 of *ISPRS Annals Photogram. Rem. Sens. and Spat. Inf. Sci.*, pp. 15–20. Regina, Canada, 2013.

[11] C. Fulgenzi, A. Spalanzani, and C. Laugier, "Dynamic obstacle avoidance in uncertain environment combining pvos and occupancy grid," in *in Proc. IEEE Int. Conf. on Robotics and Automation, 2007*, Rome, Italy, pp. 1610–1616.

[12] M. Himmelsbach, A. Müller A. T. Lüttel, and H.-J. Wünsche, "LIDAR-based 3D Object Perception," in *Proceedings of 1st International Workshop on Cognition for Technical Systems*, Munich, Germany, 2008.